Journal of Database Management
Volume 33 • Issue 1

A Temporal JSON Data Model and Its Query Languages

Ruizhe Ma, University of Massachusetts Lowell, USA
Zhangbing Hu, Nanjing University of Aeronautics and Astronautics, China
Li Yan, Nanjing University of Aeronautics and Astronautics, China*

ABSTRACT

As a ubiquitous form of data in the natural world, time has been widely used in many domains such as the military, finance, medical treatment, and environment. Temporal data models that are used to model the dynamic development process of data are constantly proposed, such as temporal database and temporal XML. With the rapid development and popularization of the web, the requirement of dealing with diverse data efficiently is becoming more important. As a new generation of data exchange language, JSON (JavaScript Object Notation) has been widely used because it is lightweight, has fast parsing, and has high transmission efficiency. In this paper, the authors propose a novel temporal JSON data model. Based on the proposed temporal JSON data model, they further propose two temporal JSON query languages, t-JSONPath and t-JSONiq, which are the temporal extensions to JSONPath and JSONiq, respectively. With the proposed temporal JSON data model, they demonstrate our temporal JSON query languages with typical temporal query examples and evaluate their query response times.

KEYWORDS

JSON, Temporal Data, Temporal JSON Model, Temporal Query Languages

INTRODUCTION

As a new generation of data exchange language, JSON (JavaScript Object Notation) has the advantages of being lightweight, having fast parsing speed, and high transmission efficiency compared with XML (eXtensible Markup Language) (Lin *et al.*, 2012). Nowadays, JSON has been widely used in various application fields. A distributed evolutionary computing system, for example, is designed and implemented in (Merelo-Guervós *et al.*, 2008), which is based on the usability and efficiency of asynchronous JavaScript and JSON to improve the utilization of computing resources in distributed systems. In addition, the JSON network interface is based on its concise grammar format and efficient transmission, which can simplify message communication between heterogeneous systems (e.g., ACT-R and EPIC[1]). With Big Data's development, NoSQL (not only SQL) databases have been widely studied and applied. The document databases in NoSQL databases (e.g., MongoDB and CouchDB), for example, are designed and implemented based on JSON (Boicea, Radulescu, & Agapin, 2012). Moreover, JSON adopts a language-independent text

DOI: 10.4018/JDM.299556
*Corresponding Author

format, which makes it easier for ones to read and write and at the same time retains the grammatical habits similar to the C programming language. These characteristics have made JSON an ideal data exchange language.

Time widely exists in various practical applications. Typically, data generated by applications are closely associated with time: data may be available or correct at a certain time point or a given time interval (Dyreson *et al*., 1994). In addition, some historical data need to be recorded. From a philosophical point of view, time is ubiquitous. For example, for Bunge, a state of anything is a list of its properties at a particular moment in time. Hence, every application would require the consideration of time, explicitly or implicitly. Therefore, it is essential to manage temporal data (Nascimento, Sellis, & Cheng, 2015; Claramunt, Schneider, & Wong, 2017; Zheng, Li, & Shim, 2018). A core issue in temporal data management is establishing a temporal data model (Nascimento, Sellis, & Cheng, 2015). Some temporal data models have been proposed for representing the temporal semantics of data. Traditionally, temporal data management is mainly based on temporal relational databases (Tansel and Tin, 1998; Tansel *et al*., 1993; Chomicki, 1994; Clifford *et al*., 1995; Clifford, Croker, & Tuzhilin, 1996). With the emergence of new application paradigms and data models, several temporal data models are proposed beyond the temporal relational databases. In the context of the Web, for example, temporal XML models are developed (Wang, Zhou, & Zaniolo, 2004; Rizzolo and Vaisman, 2008; Snodgrass et al., 2008), and temporal RDF (Resource Description Framework) models are proposed (Gutiérrez, Hurtado, & Vaisman, 2007; Wang and Tansel, 2019). Furthermore, to deal with Big Data with time (Cuzzocrea, 2015), temporal NoSQL database models are devised (Hu & Dessloch, 2015).

Nowadays, JSON has been extensively applied due to its advantages of being lightweight, having fast parsing speed, and high transmission efficiency compared with XML. In (Brahmia *et al*., 2016), to manage temporal data in NoSQL databases, a framework called Temporal JSON Schema (τJSchema) is proposed, which is inspired by the τXSchema framework defined for XML data (Snodgrass *et al*., 2008). Note that the temporal JSON model and temporal query language are not investigated in (Brahmia *et al*., 2016).

In this paper, we concentrate on modeling temporal data with JSON and investigate temporal JSON query languages for the temporal JSON model. First, we propose a temporal JSON data model by extending the traditional JSON data model. Based on the proposed temporal JSON model, we further propose two temporal JSON query languages *t*-JSONPath and *t*-JSONiq. There are a number of works in temporal databases and XML, but to the best of our knowledge, this is the first effort to propose a temporal JSON data model and its query languages. The main contributions of this paper are summarized as follows:

(1) We propose a formal temporal JSON data model to record the past, current, and even future state of JSON data.
(2) Based on the proposed temporal JSON data model, we extend two mainstream JSON query languages called JSONPath and JSONiq and then obtain temporal JSON query languages *t*-JSONPath and *t*-JSONiq by extending temporal operations.
(3) We evaluate the proposed temporal JSON query languages *t*-JSONPath and *t*-JSONiq with experiments, which apply different types of temporal queries.

The remainder of this paper is organized as follows. The next section presents a brief overview of related work in temporal databases and temporal XML. The third section proposes our temporal JSON data model and the expression form of the temporal JSON document. The fourth section proposes the temporal query languages *t*-JSONPath and *t*-JSONiq. The fifth section presents the experimental evaluations of our temporal query languages. The final section concludes this paper.

Related Content

Conceptual Data Modeling Patterns: Representation and Validation
Dinesh Batra (2005). *Journal of Database Management (pp. 84-106).*
www.irma-international.org/article/conceptual-data-modeling-patterns/3333

Semantic Heterogeneity in Multidatabase Systems: A Review and a Proposed Meta-Data Structure
Te-Wei Wangand Kenneth E. Murphy (2004). *Journal of Database Management (pp. 71-87).*
www.irma-international.org/article/semantic-heterogeneity-multidatabase-systems/3321

TEDI: Efficient Shortest Path Query Answering on Graphs
Fang Wei (2012). *Graph Data Management: Techniques and Applications (pp. 214-238).*
www.irma-international.org/chapter/tedi-efficient-shortest-path-query/58612

Cross-Correlation Measure for Mining Spatio-Temporal Patterns
James Ma, Daniel Zeng, Huimin Zhaoand Chunyang Liu (2013). *Journal of Database Management (pp. 13-34).*
www.irma-international.org/article/cross-correlation-measure-for-mining-spatio-temporal-patterns/86282

GeoBase: Indexing NetCDF Files for Large-Scale Data Analysis
Tanu Malik (2014). *Big Data Management, Technologies, and Applications (pp. 295-313).*
www.irma-international.org/chapter/geobase/85460