

时态 JSON 数据模型及查询语言处理

胡章兵,左良利

(南京航空航天大学 计算机科学与技术学院,江苏 南京 211106)

摘要:JSON 作为新一代的数据交换格式,因其轻量级,易解析,高效率等特点在数据交换领域变得越来越受欢迎。但是,传统的 JSON 文档不能反映自身的历史演变进程,而同时又希望能够检索任意时间点的文档内容。因此,能够反映文档内容随时间变化的时态模型变得十分必要和有价值。由于 JSON 和 XML 的功能非常类似,并且时态 XML 已经得到了众多学者的广泛研究,因此通过借鉴时态 XML 的研究成果可以为时态 JSON 研究工作提供很多帮助和启示。通过在非时态 JSON 模型中加入时间属性,提出时态 JSON 数据模型。该时态模型记录了 JSON 文档元素随时间变化的历史数据,再对非时态查询语言进行时态扩展支撑,就可以得到在任意时间点的文档快照,检索出查询语句的结果。最后,给出了模型的模式定义和时态模型到时态文档的映射算法。

关键词:时态 XML 模型;时态 JSON 数据模型;查询语言;模式;映射算法

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2019)10-0141-05

doi:10.3969/j.issn.1673-629X.2019.10.028

Modeling Temporal Information with JSON and Query Language

HU Zhang-bing, ZUO Liang-li

(School of Computer Science and Technology, Nanjing University of Aeronautics and
Astronautics, Nanjing 211106, China)

Abstract:As a new generation of data exchange format, JSON is becoming more and more popular in the field of data exchange because of its lightweight, easy to parse and high efficiency. However, traditional JSON document cannot reflect their own historical evolution, while at the same time hope to retrieve the document content at any point in time. As a result, a temporal model that reflects changes in document content over time becomes necessary and valuable. Because JSON and XML have extreme similar functions, and temporal XML has been widely studied by many scholars, it can provide a lot of help and inspiration for the study of temporal JSON by referring to the research results of temporal XML. A temporal JSON data model is proposed by adding temporal attributes to the non-temporal JSON model. It records the historical data of JSON document elements varying with time, then supports the temporal expansion of the non-temporal query language to obtain a snapshot of the document at any point in time and retrieve the results of query statement. Finally, the schema definition of the model and the mapping algorithm from temporal model to temporal document are given.

Key words:temporal XML model; temporal JSON data model; query language; schema; mapping algorithm

0 引言

随着网络技术的发展和普及,各个领域的数据都呈现指数级的增长。中国云计算大会网站 2018 年发布的数据量增长报告显示,2020 年的互联网数据量将是目前的 44 倍。正是由于各个领域的数据量不断增长和软硬件计算能力的提升,云计算、大数据、机器学习等智能技术得以迅猛发展和应用。由此,利用这些智能技术来挖掘出在大量数据的背后所隐含的发展趋势和规律就显得极具有价值和必要^[1]。

时间属性作为自然界数据表达的一种重要衡量维度,挖掘出在大量数据发展过程中的时态规律,逐渐得到学术界和工业界的广泛关注和研究。时态数据挖掘算法(temporal data mining)就是在这一背景下逐渐被研究和应用的成果^[2]。时态数据挖掘算法是对观测到的时间属性数据进行分析,然后发现未知的知识和以时间数据拥有者可以理解且对其有价值的方式来总结时间知识。由于现实世界总是按照时间不断发展变化的,大多数信息都包含有时间属性,例如股票的波动,

收稿日期:2018-11-12

修回日期:2019-03-13

网络出版时间:2019-04-24

基金项目:国家自然科学基金(61370075)

作者简介:胡章兵(1992-),男,硕士研究生,CCF 会员(A2334G),研究方向为智能数据处理。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190424.1055.070.html>

超市的交易,天气的变化,文档的编辑,含时间的实验数据等。因此,在数据知识领域,如何对现有的随时间变化的数据进行时态建模,一直是学者们的热点研究方向。

1 相关工作

文中的主要工作与时态数据库和时态 XML 非常相似。接下来分别从这两个领域介绍其相关工作。

1.1 时态数据库

首先,在时态数据库发展过程中,如何在传统的数据库模式中引入时态信息得到了众多学者的广泛研究。自 20 世纪 80 年代,E. F. Codd 提出的关系数据库模型得到广泛应用之后,Jensen 等就随即提出了基于关系表模式的时态关系数据库^[3],并给出了时态数据库的概念定义和词汇表。James Clifford 通过将时间属性添加到关系表模型,提出了历史关系数据库模型^[4],当需要跟踪来自数据的所有更改并具有建模现实的完整历史时,其时间标记数据值可以体现数据在时间维度上的历史变化过程。在文献^[5]中,将时间关系模型主要分为两类:非分组记录(ungrouped)和分组记录(grouped)。非分组记录模型在记录后面添加表示该记录的有效时间属性(有效开始时间 v_{start} 和有效结束时间 v_{end})。而分组记录是根据记录在时间上的发展变化过程对记录中的每个属性进行时间分组标记。表现形式的区别用一张员工表展示,如图 1 所示。

name	salary	title	dept	v_{start}	v_{end}
tom	50 000	engineer	d1	v1	e1
tom	60 000	engineer	d1	v2	e2
tom	60 000	Sr engineer	d2	v3	e3
tom	60 000	leader	d2	v4	e4

(a) ungrouped record

name	salary	title	dept
tom	v1 50 000 e1	v1 engineer e2	v1 d1 e2
	v2 60 000 e4	v3 Sr engineer e3	v3 d2 e4
		v4 leader e4	

(b) grouped record

图 1 非分组记录和分组记录

随着单时态数据库的不断深入研究,其在时态表达方面的缺点就逐渐显现出来。不论是单时态有效时间数据库还是单时态事务时间数据库,都仅能够表达数据在某一个时间维度的信息。由此,双时态数据库逐渐成为学者们研究的热点方向。Knolmayer 等给出了双时态关系数据库的模型定义和具体实现^[6],其在关系数据库模型中同时加入有效时间(数据在现实世界中真实有效的时间区间)和事务时间(数据在存储介质世界的有效时间区间),得到了能够在现实世界维度和物理介质维度都能体现数据在时间线索上的发展变化历史模型。此外,Dayal 等实现了一个基于面向对象的时态数据库系统^[7],在面向对象模型中加入有效时间维度,扩展了面向对象数据库数据对象在时态语义表达的功能。Zheng 等提出了一个基于图数据库的时态图模型^[8]。

1.2 时态 XML

关系型数据库模型和查询语言 SQL (structure query language) 虽然能够很好地表达和处理结构化数据,但是现有的很多应用领域需要处理半结构化的数据,但关系数据模型和 SQL 对半结构化数据模型表达的灵活性差,因此半结构化模型和查询语言处理逐渐成为数据库,网络数据传输,元数据等领域的研究热点。特别的,随着现代计算机应用(例如社交网络,协作 web 信息系统等)的迅速发展,要求网络上的数据交换具有较高的传输效率。XML (extensible markup language) 自 20 世纪 80 年代被万维网联盟(W3C)拟定,因其自描述,易理解等优点已经成为了 W3C 的推荐标准。但是由于非时态 XML 模型既不能支持时态语义表达功能,也不能体现 XML 文档在随时间更迭过程中的数据变化过程等缺点,时态 XML 逐渐吸引了众多学者的研究。Amagasa 等针对非时态 XML 的 XPath 模型,通过在每条边上添加有效时间戳,以表示子节点是否有效存在^[9]。另外,Wang 等在其管理系统中,为了支持多个客户端编辑 XML 文档进行协同工作,利用时态 XML 模型来管理文档在协同编辑过程中的版本变化^[10]。同时,Grandi 为了高效地管理 XML 格式的法律文献,实现了一个时态 XML 模型,通过时态模型中的发表时间,有效时间,效力时间,事务时间四个时间维度表示法律文献的演变历史^[11]。同样,为了更好地突出模型在时态表达和模型一致性等方面的特性,双时态 XML 在许多应用领域也日益成为研究对象。Wang 等提出了在双时态 XML 领域中较为普遍的实现方案,通过在标签属性中加入有效时间区间和事务时间区间来表达数据在双时态维度的体现^[12]。另外,汤娜等讨论了在双时态 XML 查询中 now 语义失真的研究和扩展^[13]。一般的,在由时态

XML 模型到时态 XML 文档的转化中,时态文档通过时态属性或时态标签来表示数据的时间信息。属性的时间过程同样用属性元素来表示。

2 非时态 JSON 数据模型

XML 虽然得到了广泛的应用和研究,但是随着需求的不断增加和复杂,XML 的缺点也逐渐显露出来,如冗余度高,数据插入修改困难,处理大量数据时效率低下,客户端浏览器解析困难等。2006 年, Douglas Crockford 把 JSON(JavaScript Object Notation)提交给 Internet Engineering Task Force (IETF),JSON 因相比 XML 易读写,同时也易于机器解析生成等特点得到了广泛应用。自 JSON 数据格式提出以来,针对 JSON 和 XML 在实用性、传输效率、安全性等方面进行了深入的讨论研究和实验^[14-16],都一致认为 JSON 在传输效率和浏览器解析等方面都比 XML 更加的实用和高效。因此,数据交换格式在 WEB 领域中逐渐从 XML 转为 JSON。2009 年,权重民等利用 JSON 实现了一种高效、安全访问远程数据库的方式^[14]。2018 年,王东兴等提出基于 JSON 的 GeoJSON 在异构地理信息数据集成中的应用^[16]。但是在非时态 JSON 得到广泛应用时,针对 JSON 的理论研究和模型定义却鲜有成果。

2.1 JSON 语法及文档示例

JSON 语法定义,JSON 文档是由键值对组成的字典,其中的值又可以是一个 JSON 文档,从而 JSON 模型允许任意级别的嵌套。完整的 JSON 规范定义了七种类型的值:分别是字符串,数字,对象,数组, true, false 和 null。文中给出了一个来自某论坛网站用户信息的简单 JSON 文档示例,如下:

```
{
  "name": { "firstname": "Tom", "lastname": "Doe" },
  "age": 18,
  "hobbies": [ "fish", "tennis" ]
}
```

2.2 JSON 文档树

根据 JSON 文档的语法定义,可以用树型结构来描述 JSON 文档的信息,如图 2 所示。

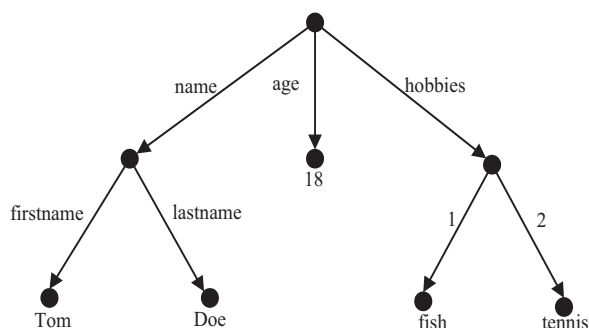


图 2 JSON 树

2.3 JSON 树模型

定义 1(节点):假设 J 是一个 JSON 文档,其中 $V(J)$ 是文档 J 的节点集,则 $V(J)$ 有六种节点:根节点、字符串节点、数字节点、布尔类型节点、数组节点和对象节点,分别标记为 $r, v_s(J), v_n(J), v_b(J), v_a(J), v_o(J)$ 。为了简化讨论, null 暂不做考虑。满足:

$$V(J) = r \cup v_s(J) \cup v_n(J) \cup v_b(J) \cup v_a(J) \cup v_o(J)$$

定义 2(边): $E(J)$ 是文档 J 的所有边的集合,其中每条边可以表示为 (p, c) , 其中 $p = r, c \in v_e(J)$ 或 $p \in v_e(J), c \in v_e(J)$ 或 $p \in v_e(J), c \in v_i(J)$ 或 $p \in v_e(J), c \in v_a(J)$ 或 $p \in v_a(J), c \in v_i(J)$ 。

定义 3(文档):JSON 文档 $J = (V(J), E(J), r)$ 。

3 时态 JSON 数据模型

随着时间的更迭,用户的数据不断发生变化。该用户在 2017 年 12 月 1 号创建,基本信息如图 2 所示。此后,在 2017 年 12 月 15 号其“firstname”变更为“Alen”,在 2018 年 1 月 1 号加入“vip”属性,并且成为该论坛的 VIP 用户,在 2018 年 1 月 1 号新添了爱好“yoga”,在 2018 年 1 月 5 号删除爱好“tennis”,在 2018 年 1 月 23 号,其年龄由 18 增长为 19。

由于非时态 JSON 模型不能很好地表达数据对象在时间上的变化过程,通过对非时态 JSON 模型进行时态扩展得到了时态 JSON 模型。

3.1 时态 JSON 模型

定义 4(边):将有效时间属性加入到边的定义中,以此反映这条边尾部节点的有效时间区间,得到 $TE = ((p, c), t)$ 。其中 (p, c) 与非时态模型的定义一致,而 t 表征该边的时态信息, t 是一个时间区间,由有效开始时间和有效结束时间构成 $[v_{start}, v_{end}]$,如:

$[2017-12-01, 2018-01-15]$ 。2017 年 12 月 1 号到 2018 年 1 月 15 号是该边尾部对应节点的有效时间范围。

定义 5(文档一致性):若一个节点有很多孩子节点,其有效时间为 t ,连接双亲节点和孩子节点的边分别为:

$((p, c), t_1), ((p, c), t_2), \dots, ((p, c), t_n)$, 则:

$$(1) \bigcup_{1 \leq i < n} t_i \subseteq t;$$

$$(2) t_1 \cap t_2 \cap \dots \cap t_n = \emptyset \text{ (数组除外)}。$$

根据用户信息的变化过程,可以用图 3 所示的时态模型表示文档的演变进程。

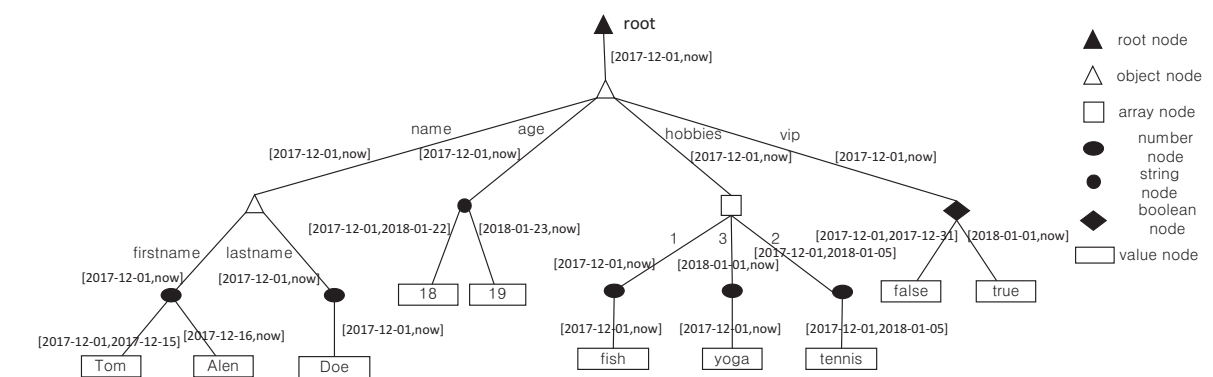


图 3 时态 JSON 模型

3.2 JSON Schema

JSON Schema 是用来规范 JSON 文档的属性结构的,JSON Schema 本身也是一个 JSON 文档。JSON 文档的 schema 如图 4 所示。

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Member",
  "type": "object",
  "properties": {
    "name": {
      "id": "http://jsonschema.com/name", "type": "string"
    },
    "age": {
      "id": "http://jsonschema.com/age", "type": "number"
    },
    "hobbies": {
      "id": "http://jsonschema.com/hobbies", "type": "array",
      "items": { "type": "string" }
    },
    "require": [ "name", "age", "hobbies" ]
  }
}
```

图 4 JSON Schema

但是,随着 JSON 文档的变化,JSON 文档的 Schema 也同样会发生变化。例如,文档加入了属性“vip”之后,原本的 Schema 不能够表达 JSON 文档的结构了。但是由于 Schema 也是 JSON 文档,因此采用同样的方法,可以用类似图 3 所示的时态 JSON 模型来表征 JSON Schema 的变化。

3.3 模型到文档的映射算法

Algorithm: Temporal Model Translation
Input temporal model root node R
Output temporal document
1. $D \leftarrow \text{getDocument}(R)$
2. $\text{getDocument}(N)$
3. if(N is object) :
4. $\text{getDocument}(N)$
5. else :
6. If(N is array) :
7. for each edge :
8. $p_1 \leftarrow \text{createKVpair}(N.\text{attribute}, N.\text{text})$

9. $p_2 \leftarrow \text{createKVpair}(\text{"validtime"}, \text{time})$
10. else :
11. for each edge :
12. $p_1 \leftarrow \text{createKVpair}(N.\text{attribute}, N.\text{text})$
13. $p_2 \leftarrow \text{createKVpair}(\text{"validtime"}, \text{time})$
14. }
15. $\text{createKVpair}(\text{key}, \text{value})$ {
16. $\text{kv.set}(\text{key})$
17. $\text{kv.set}(\text{value})$
18. return kv
19. }

根据上述提出的时态模型到时态文档的映射算法,可以得到如图 5 所示的时态 JSON 文档。

```
{
  "name": [ { "name": { "firstname": { "firstname": "John",
    "validtime": [2017-12-01, 2017-12-15] },
    "firstname": "Alen", "validtime": [2017-12-16, now] } } ],
  "lasttime": [ { "lastname": "Doe", "validtime": [2017-12-01, now] } ],
  "validtime": [2017-12-01, now] ],
  "age": [ { "age": 18, "validtime": [2017-12-01, 2018-01-22] },
    { "age": 19, "validtime": [2018-01-23, now] } ],
  "hobbies": [ { "hobbies": [ "fish", "tennis", ], "validtime": [2017-12-01, 2018-01-01] },
    { "hobbies": [ "fish", "tennis", "yoga", ], "validtime": [2018-01-01, 2018-01-05] },
    { "hobbies": [ "fish", "yoga", ], "validtime": [2018-01-05, now] } ],
  "vip": [ { "vip": false, "validtime": [2017-12-01, 2017-12-31] },
    { "vip": true, "validtime": [2018-01-01, now] } ],
  "validtime": [2017-12-01, now] ]
}
```

图 5 时态 JSON 文档

3.4 时态文档查询语言

非时态 JSON 查询语言目前还没有标准规范,但

是对于非时态 JSON 文档,已经有 JSONPath,JSONip, N1QL 等处理方法。文中时态文档的查询语言综合上述几种方式的特点进行时态扩展。首先,在 JSONPath 表达式中添加时态属性扩展支持,可以满足大部分检索需求,例如普通的 JSONPath 从 2.1 节文档示例中查询用户的名字,可以用如下 JSONPath 表达式:

```
$ . name. firstname 或 $ [ "name" ] [ "firstname" ]
```

但是要检索出在图 3 时态文档中的名字,需要对其进行时态扩展,结合 JSONPath 和 JSONip,提出了时态查询语言。例如要检索出在 2017 年 12 月 5 号该用户的名字,可以用如下表达式:

```
let $ name = collection. find( " name" )
return {
  " firstname" = $ name( " firstname" ) [ @ from eq 2017-12-05
and @ end eq 2017-12-05 ]
}
```

其中,“@ from”属性表示开始时间,“@ end”表示结束时间,“eq”表示相等,用“=”表示作用一样。另外,如果要检索该用户注册为 VIP 的有效时间,可以用自定义 time() 函数获取,表达式如下:

```
let $ vip = collection. find( " vip" ) [ @ value = true ]
return time( $ vip)
```

4 结束语

目前,针对 JSON 时态信息建模和时态查询语言处理的理论研究非常少,文中提出了时态 JSON 数据模型。根据时态 JSON 模型,对传统的查询语言做了简单的时态扩展,但目前仅能支持一些相对简单的查询,后续会进一步提出类似 TempSQL 能完成连接、分组、排序等更高级的时态查询处理。另外,还提出了一个由时态模型到时态文档的映射算法,解决了从模型到文档的映射方法,后续会根据提出的时态模型和针对映射后的文档解决其在存储方面的问题,因为存储的性能好坏直接影响查询性能。

参考文献:

- [1] HAN Jiawei. Data mining: concepts and techniques[M]. [s. l.]: Morgan Kaufmann Publishers Inc., 2005.
- [2] POST A R, HARRISON JR J H. Temporal data mining[J]. Clinics in Laboratory Medicine, 2008, 28(1): 83-100.
- [3] JENSEN C S, CLIFFORD J, GADIA S K, et al. A glossary of temporal database concepts[J]. ACM SIGMOD Record, 1992, 21(3): 35-43.
- [4] CLIFFORD J, CROKER A. The historical relational data model (Hrdm) and algebra based on lifespans[C]//IEEE third international conference on data engineering. Los Angeles, CA, USA: IEEE, 1987: 528-537.
- [5] SNODGRASS R T. Temporal databases[J]. Computer, 1992, 19(9): 35-42.
- [6] KNOLMAYER G F, MYRACH T. Concepts of bitemporal database theory and the evolution of web documents[C]//Proceedings of the 34th annual Hawaii international conference on system sciences. Maui, HI, USA: IEEE, 2001: 10.
- [7] WUU G T J, DAYAL U. A uniform model for temporal object-oriented databases[C]//Eighth international conference on data engineering. Tempe, AZ, USA: IEEE, 1992: 584-593.
- [8] ZHENG Linjiang, ZHOU Longhui, ZHAO Xin, et al. The spatio-temporal data modeling and application based on graph database[C]//2017 4th international conference on information science and control engineering. Changsha, China: IEEE, 2017.
- [9] AMAGASA T, YOSHIKAWA M, UEMURA S. A data model for temporal XML documents[C]//Proceedings of the 11th international conference on database and expert systems applications. [s. l.]: Springer, 2000: 334-344.
- [10] WANG Fusheng, ZHOU Xin, ZANIOLO C. Temporal information management using XML[C]//International conference on conceptual modeling. [s. l.]: Springer, 2008: 858-859.
- [11] GRANDI F, MANDREOLI F, TIBERIO P, et al. A temporal data model and management system for normative texts in XML format[C]//Proceedings of the 5th ACM international workshop on web information and data management. [s. l.]: ACM, 2003: 29-36.
- [12] WANG Fusheng, ZANIOLO C. XBiT: an XML-based bitemporal data model[C]//International conference on conceptual modeling. [s. l.]: Springer, 2004: 810-824.
- [13] 汤娜, 陈罗武, 刘瑞君, 等. 双时态 XML 查询中 now 语义失真的研究与扩展[J]. 计算机科学, 2008, 35(6): 233-235.
- [14] 权重民, 彭昕昀. 利用 JSON 实现 Android 高效、安全访问远程数据库的一种方式[J]. 韶关学院学报, 2011, 32(12): 16-20.
- [15] LIN Boci, CHEN Yan, CHEN Xu, et al. Comparison between JSON and XML in applications based on AJAX[C]//International conference on computer science & service system. Nanjing, China: IEEE, 2012: 1174-1177.
- [16] 王东兴, 朱翊. GeoJSON 在异构地理信息数据集成中的应用[J]. 测绘与空间地理信息, 2018, 41(6): 148-150.