Computer Science and Application 计算机科学与应用, 2018, 8(3), 314-322
Published Online March 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.83036

Hans汉斯

Research on Heterogeneous Data Integration Based on JSON

Zhihong Liang[1], Gehao Lu[2*], Yuxiang Huang[2], Mingming Qin[1], Shuqiong Sun[2]

[1]School of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming Yunnan
[2]School of Software, Yunnan University, Kunming Yunnan
Email: *zhliang@swfu.edu.cn

Received: Mar. 10th, 2018; accepted: Mar. 22nd, 2018; published: Mar. 28th, 2018

Abstract

With the advent of big data, Internet plus era, Internet technology is used in more and more fields to complete the daily work and produce huge amounts of data. How to effectively realize the sharing of data, solve the heterogeneity of syntax and semantics of heterogeneous data integration is the problem to be solved in heterogeneous data. On the basis of traditional data integration scheme, combined with middleware technology, JSON related technology and Ontology related technology, an improved heterogeneous data integration scheme is proposed. In this scheme, the query processor module adds query optimizer and selective replicator to improve query efficiency. In the field of geographic information system, a complete design scheme is proposed and implemented, and the feasibility of this method is verified. The example shows that the heterogeneous data integration middleware based on JSON can realize the resource sharing of heterogeneous data well and solve the problem of heterogeneous data.

Keywords

Heterogeneous Data, Data Integration, JSON, Middleware

基于JSON的异构数据集成的研究

梁志宏[1]，陆歌皓[2*]，黄宇翔[2]，秦明明[1]，孙书琼[2]

[1]西南林业大学大数据与智能工程学院，云南 昆明
[2]云南大学软件学院，云南 昆明
Email: *zhliang@swfu.edu.cn

收稿日期：2018年3月10日；录用日期：2018年3月22日；发布日期：2018年3月28日

文章引用: 梁志宏, 陆歌皓, 黄宇翔, 秦明明, 孙书琼. 基于JSON的异构数据集成的研究[J]. 计算机科学与应用, 2018, 8(3): 314-322. DOI: 10.12677/csa.2018.83036

摘　要

随着大数据、互联网+时代的到来，越来越多的领域采用互联网技术完成日常的工作并产生了海量的数据信息，如何有效的实现海量数据的共享，解决异构数据的语法和语义的异构性是异构数据集成中亟需解决的问题。在传统的数据集成方案的基础上，结合中间件技术，JSON相关技术、Ontology相关技术，提出一种改进的异构数据集成方案。在该方案中，查询处理器模块，加入查询优化器和选择性复制器，提高查询效率。再针对地理信息系统领域，提出了完整的设计方案，并进行了实现，验证了这种方法的可行性。通过实例表明，基于JSON的异构数据集成中间件能够很好的实现对异构数据的资源共享，解决异构数据问题。

关键词

异构数据，数据集成，JSON，中间件

1. 引言

随着大数据、互联网+的时代的到来，越来越多的领域采用互联网技术完成日常的工作。同时也产生了海量的数据信息，如何有效的实现海量数据的共享是目前数据集成方案需要解决的问题。同时由于软件企业的环境选择不同，所以在企业中有很多厂商的硬件平台、操作系统、网络协议、DBMS 上很多差异性，为了改善“信息孤岛”的情况，企业希望通过数据集成方式建立信息交流的桥梁。这就是异构系统的集成，其中关键技术是数据交换和信息在分布式异构系统中的集成。如何构建一个共同的“虚拟数据库”来方便数据的共享，实现数据信息的最大价值，是目前亟需解决的问题。

本文是在目前比较流行的集成技术研究的基础上，参照过去的解决异构数据的方案，同时参考了目前比较流行的数据集成系统，来设计基于 JSON 的数据交换格式同时引入中间件技术的数据集成架构。实现消除信息孤岛，给用户提供统一的数据查询平台，屏蔽异构，从而实现用户的透明性访问[1]。本文还引入查询优化的算法设计，从而使用户在查询处理数据时，效率大大提高。对查询进行优化，在数据集系统中简化操作，降低查询响应时间。

2. JSON 及其技术

2.1. JSON

作为一种轻量级的数据交换格式，JSON (Java Script Object Notation)逐渐的出现在现代 WEB 技术中。随着 JSON 被广泛的使用，人们对于 JSON 的研究也越来越多。同时，JSON 是现在比较流行的 JavaScript 的一个子集。JSON 在 WEB 应用中有很多优点，如传输速率快，支持语言多，书写简单等。由于它的优势，现在广泛的应用在开发中。

2.2. JSON Schema

在 XML 应用中，其中 XML Schema (XML 模式)很好的约束了 XML 文档。同样的，以 JSON Schema

(JSON 模式)对文本的命名规则，属性大小等进行限定后的文档才是有效的。而这些统一要求的约束或规范就是 JSON Schema [2]。经过 JSON Schema 规范过合理性文档，其中的一些信息都有统一的特性，扩展了 JSON 的自我描述性，并且内容规范明了，对于开发程序来说和计算来说都可以轻松的获取文本信息的[3]。

2.3. JSON 文档的查询技术

JSONiq 查询技术支持 where 语句，let 语句，for 语句等。这些支持使得 JSONiq 查询技术在数据集成中，有效的对异构数据源进行查询，同时使得数据查询变得比较简单。JSONiq 查询技术的强大还不仅如此，它还支持重叠和非重叠窗口。

JSONiq 支持 JSON 数据查询以及数据的自我更新技术，在今后异构数据集成中，数据查询处理模块发挥很大的作用。使用 JSONiq 查询技术作为全局查询语言，提高系统的灵活性和高效性。

3. JSON 中间件技术集成方案

3.1. 基于中间件的数据集成模型

主要的数据集成方法有 FDBS、数据仓库法、中介系统法。FDBS 在处理少量数据时具有优势，可以实现其自治性。数据仓库法实现用户的访问并为用户提供良好的策略服务。查询处理性能要求比较高。但是由于公用数据库都是采用统一的仓库模式的数据，使得对于一些最新的数据信息，不能及时的更新。再加上昂贵的数据仓库系统、一些自带的数据转换工具的数据库的搭建，数据仓库法在数据集成中并不是很理想的解决方案。中介系统通过引入中介器来完成对数据的查询访问，使得用户无需对底层复杂的数据源进行处理，实现对数据的透明性访问。

在数据集成方面，引入了数据库中间件，它也是位于 C/B 之间的中介接口软件。客户端层是应用层，是一些应用程序和接口。而服务器端则是各种异构数据源的集成。在数据集成中，中间件和一系列平台组成的中间层它提供了一系列的 API 程序接口[4]。实现对数据源的透明访问。所以，相比较而言，基于中间件的数据集成方案具有很大的优势，基于中间件的异构数据库集成模型如图 1 所示。

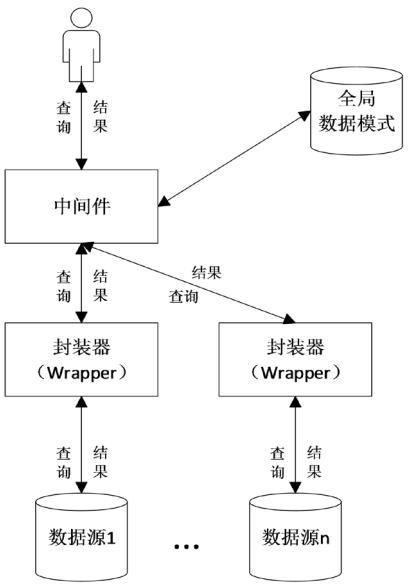

Figure 1. Integration model based on Middleware
图 1. 基于中间件的集成模型

3.2. JSON 集成中间件的改进方案

通过对传统的数据集成方法的研究学习，结合目前在数据集成主要解决的问题，同时分析了传统数据集成方案的不足。通过对选择性复制的理念的研究，再结合新的技术如 JSONiq 和 JSON-LD、Ontology 技术，提出了新的数据集成方案。该方案的体系结构设计的目的是解决语法、语义异构的问题。同时提供一个统一的查询平台。使用 RESTfull Web 服务代替传统的基于 SOAP 服务。作为一个整体约束 JSON 集成框架。一方面，这种架构简化了系统的架构同时统一了界面。解决了互操作的问题。再加上选择性复制器，提高系统查询处理的高效性。详细的 JSON 集成中间件体系结构设计如图 2 所示。

3.3. 与传统的数据集成方法的比较

1) 数据交换格式

传统的数据集成方案大都是基于 XML 的数据交换格式，虽然在数据集成方面的相关研究比较多，但随着海量数据的不断增大，再加上 XML 自身冗长的标签特性，使得在处理大量地理信息时数据传输效率底下，等待时间长，响应慢等问题。JSON 由于自身轻量级的特性很好的解决了这一问题[5]。

2) Ontology 的引入

最近比较流行的 Ontology 技术的引入，很好的解决的数据集成问题中语义的异构。由于 Ontology 可以精确的对领域知识进行描述，而数据集成就是面向某一领域的，如地理信息系统的集成是面向地理信息领域的集成。

3) JSON 相关技术的引入

JSON 查询技术，在查询处理器模块支持 where 语句，let 语句，for 语句，方便有效的对异构数据源进行查询，使得数据查询变得比较简单。同时该支持自我更新技术，提高系统的灵活性和高效性。

4. 基于 JSON 集成中间件的方案设计与实现

4.1. 查询处理器模块设计

当应用层客户提交查询请求时，这时的请求是全局模式下的请求，被传到中间层经过翻译再传到底层的数据层。同样的过程，查询的结果按照相反的步骤提交给用户。如图 3 是查询处理器模块的基本处理流程。

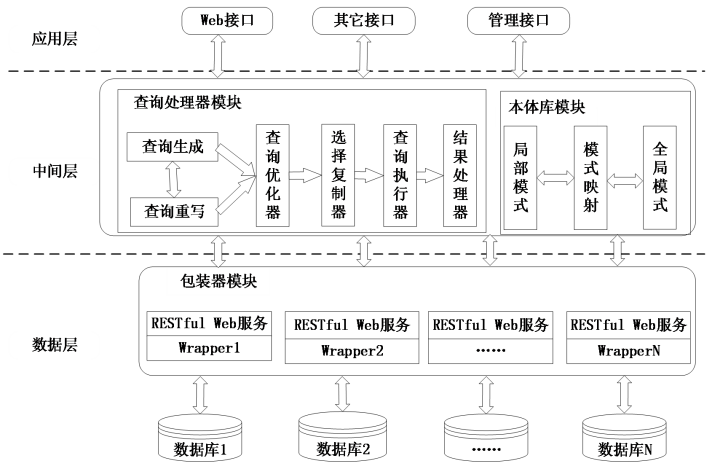

Figure 2. JSON integrated middleware architecture
图 2. JSON 集成中间件体系结构

1) 查询生成

户提交查询请求，对查询请求语法上的验证。在进行语法验证时，采用使用 JSON Schema 模式验证，最后形成全局查询语句。

2) 查询重写

查询重写，就是对查询的在次处理，也就是全局查询子查询的过程。

本文提出了一个简单的算法如下：

输入：Qg (全局查询)

输出：Qs (子查询)，Qs = {Qs1，Qs2，…… Qsn}

算法的具体描述：

a) 首先输入 Qg，对 Qg 进行子句分离。在分离的过程中参照 FLWOR 表达式中的 F、R、W。最后保存分离结果。代码如下图 4。

b) Where 句子的处理：依据 where $x.project_type eq 1 的查询条件的布尔值。

c) for 子句的处理：依据$x in collection 赋值变量，由通过 JSON-LD 描述的两者的映射规则。

d) return 子句的处理：return {“num”:$x.project_num……}将返回的结果进行结构的构造，并对这些内容进行条件约束。

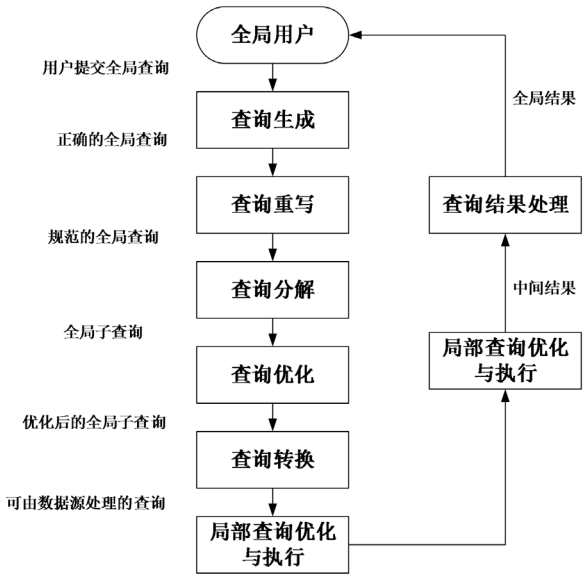

Figure 3. Basic processing flow chart
图 3. 基本处理流程图

```
for $x in collection("Global.jsonld.project Info")
where $x.project_type eq 1
return {"num":$x.project_num,
 "name":$x.project_name,
"type":$x.project_type,
"money":$x.project_money, }
```

Figure 4. FRW code
图 4. FRW 代码

3) 查询 CBO (Cost-Based Optimization)优化器

在对数据查询处理模块时，需要考虑两个问题。一是如何做到用户提交全局查询后，中间件返回的查询结果时，这个过程中响应时间尽可能的短。二是如何使得查询处理数据这块的代价和最小[6]。

4) 选择性复制器

选择性复制器就是有选择的对部分查询数据，复制到指定的存储空间，以方便后期的其它用户对数据的查询使用，来减少数据查询时间。

5) 查询执行器

它的作用是负责查询计划的执行。Wrapper 对各种异构数据源进行包装处理，以方便数据的查询。而查询执行器发出请求后，并将经过处理的结果返还给查询执行器。由查询执行器则参考优化后的查询计划进行查询的执行，最后请求工作。

6) 查询处理器

把 JSON 源文件以及 JSON Schema 文件，通过结果处理器，最后形成新的文件返回给客户端。这个文件就是基于 JSON 的结果文件。

4.2. 包装器模块设计

Wrapper 主要由两个模块组成：查询转换器和结果转换器。Wrapper 主要对底层数据源进行包装处理过程的结构如图 5 所示。

Wrapper 主要由查询转换器和结果转换器构成：1) 查询转换器：中间层将分解后的子查询，发送给向 Wrapper。Wrapper 接收到 JSONiq 子查询后，调用查询转换器对 JSONiq 子查询处理为针对底层数据源的 JSONiq 子查询。如底层数据源是来自 SQL 数据库，则转换器把子查询转换为 SQL 可以识别的子查询。2) 结果转换器：它接收底层 SQL 对查询处理的结果，再对并把这些查询处理结果进行转换，最后生成 JSON 数据格式的文件。

4.3. 本体库模块设计

Ontology 库的设计包括两方面。一是模式集成问题的设计，如：Global Schema 到 Export Schema、Export Schema 到 Local Schema 映射。二是在数据集成中，异构性的解决，这里主要解决的是语义上的异构问题。

Ontology (本体)是概念模型的明确的规范说明[7]。上述的这个定义是对 Ontology 经典的定义。由于 Ontology 可以精确的对领域知识进行描述，而数据集成就是面向某一领域的。如地理信息系统的集成是面

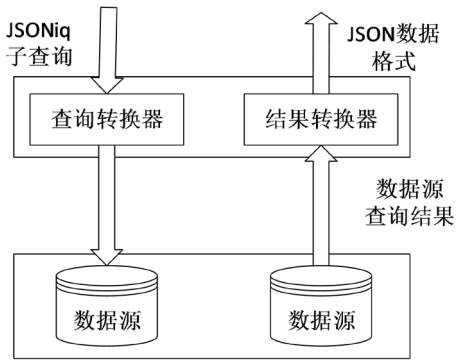

Figure 5. The design of Wrapper
图 5. Wrapper 的设计

向地理信息领域的集成，而医院管理系统的集成就是面向医疗信息领域的集成。举个简单例子来说，Ontology 的概念。有三种描述：1) 苹果。2) apple。3) 水果。三个描述都是在描述水果，这些描述可以称为 Ontology。这就是 Ontology 形式化的特点。因为 Ontology 可以明确的规定某一领域的范围，因此 Ontology 还有明确的特征。如地理信息 Ontology，代表着地理信息这一领域。同时也表现着一种共享的概念模型。所以，Ontology 的能准确的描述领域知识，从而消除语义异构的问题[8]。

5. 基于地理信息集成应用实例

5.1. 实验应用平台搭建

JSON 集成中间件的数据集成框架，以 JSON 作为数据交换格式，同时引入 Ontology 概念，解决了异构数据语义上的异构。实现环境如表 1 所示：

服务器端语言：PHP；

Web 服务器：Ngnix。

5.2. 地理信息系统异构问题的解决

随着地理信息系统的不断发展发展，如何通过对海量的数据的集成，从而实现对数据的高效访。如地理信息系统中三个数据源：分别是数据源 DB1、DB2、DB3。其关系如表 2~4。

Table 1. Database and operating platform configuration such as table
表 1. 数据库及操作平台配置如表

	数据源 DB1	数据源 DB2
IP 地址	192.168.3.8	222.198.33.15
数据库服务器	SQL Server	MYSQL
操作系统	Window7	XP

Table 2. DB1's traffic status table
表 2. DB1 的交通状况表

字段	类型	Null	注释
Name	Varchar(20)	YES	交通名称
Number	Int(8)	PK	交通编号
Level	Int(10)	YES	交通层次
Money	Money(100)	YES	高速费用
Condition	Varchar(100)	YES	交通状况
Mark	Varchar(20)	YES	道路名称

Table 3. DB2's road facilities
表 3. DB2 的道路设施表

字段	类型	Null	注释
Capacity	Int(8)	PK	通行能力
Name	Varchar(20)	YES	道路名称
Open_time	Int(10)	YES	开通时间
R_grade	Money(100)	YES	道路等级

Table 4. DB3's traffic road facilities
表 4. DB3 的交通道路设施表

字段	类型	Null	注释
Name	Varchar(20)	YES	交通名称
Money	Money(100)	YES	维修费用
Tra_time	Int(10)		道路设备投放时间
Suggestion	Varchar(100)	YES	道路设备投入意见
Updation	Int(100)	YES	道路设备更新时间

表 2~4 的数据由于数据库的不同，存在异构问题。如：语法异构和语义异构。通过对异构数据的、对语法异构的研究学习，可以知道异构数据类型的异构。为了消除语法上的异构问题，使用 JSON Schema 模式分别对 DB1 和 DB2 进行描述，描述的同时也是映射过程。类似的方法用 JSON Schema 模式描述来描述异构数据源 DB2 和 DB3，解决语义间的异构。

要解决语义异构的问题，一种方法是利用 JSON 关联数据对数据源 Ontology 概念上的描述[9]。另外一种是通过对两张地理信息系统中关系表结构和语义进行分析，构造出(全局模式)中介模式下的关系表。其中 Traffic (交通)，Road (道路状况)，Facility (道路设施)。

5.3. 地理信息系统查询处理模块实现

查询处理器：当应用层客户提交查询请求时，中间层接受查询请求，并交由查询处理器处理。查询处理器处理数据的过程，实际上就是对查询进行分解为针对数据源的子查询。最后把查到的结果提交给用户。

5.4. 全局模式的实现

全局模式设计和局部模式的生成一样，首先我们需要建立一个全局 Ontology。达到语义化的描述的目的。为了形象的描述问题，设地理信息集成领域，由于全局 Ontology 在地理信息领域中，可给地理信息领域，提供统一的语意描述[10]。建立全局地理信息 Ontology，从领域到语义层次上的转变。具体步骤如下：

1) 先构建一个全局地理信息 Ontology，从而形成一个全局地理星系系统的虚拟查询视图。

2) 完成全局虚拟视图的语义化的描写。

3) 通过全局地理信息系统的虚拟查询视图，从而在数据查询处理模块式，使得用户查询不用再去针对底层的数据源。

5.5. 基于地理信息系统案例结果

1) 查询执行

a) 查询语句如下：在交通道路以及设施集成平台下，查询设施使用时间超过两年的，主干路的信息。

b) select name, tra_time from Road Facility where grade > “2”；查询结果的把上一步的查询结果作为下步查询的判断条件，所以这次输入的应该还是 name 组装成 mark Str 字符串序列。

c) SQL 语句实现：select name, type, money, level, mark from Total where platform in 信息序列 and level = “主干路”。

图 6 是查询处理后的结果。

```
{"message_info":
  [
    {"title":"昆曲高速资讯","condition":"最进几天昆明方向车流量大，请避开高峰期出
行","number":"G50",
   "level":"主干路", "mark":"KunRoad", "grade":"3"},
    {"title":"昆曲磨速资讯","condition":"规划里程692，通车里程
461km","number":"G8511",
"level":"主干路", "mark":"KunMoRoad", "grade":"4"},
    {"title":"滇中环线资讯", "condition":"在建设过程中，是“十三五”云南重点高速公
路建设","number":"0", "level":"主干路", "mark":"HuanZhongRoad", "grade":"3"},
    {"title":春城路资讯", "condition":"4月12日，下午13：00-24：00施工，请谨慎驾驶
","number":"174500",
"level":"主干路", "mark":"ChunChengRoad", "grade":"4"},

.....
  ]
}
```


Figure 6. Result code after query processing
图 6. 查询处理后的结果代码

6. 结束语

本文通过对异构数据的研究及目前异构数据源面临的主要问题，再对传统的数据集成方法进行对比研究。后来在传统的异构数据集成的基础上，引入中间件技术。提出基于 JSON 的异构数据集成方案，通过地理信息实例，很好的完成了数据集成，同时解决了异构数据语法异构和语义异构的问题。数据集成时的数据交换格式采用 JSON 数据格式而不是传统的 XML 作为交换格式。JSON 相关技术有 JSON Schema，JSONiq 技术，JSON-LD 技术。

参考文献

[1] 朱峰. 基于 JSON 的互联网异构数据整合的应用研究[D]: [硕士学位论文]. 南京: 南京邮电大学, 2016.

[2] 张佳. 基于 SOA 的异构数据集成解决方案[J]. 电子技术与软件工程, 2014(10): 223-227.

[3] Jane, L. and Steven, M. (2012) Build an App with Mongo DB. NSTL. TN. 226.

[4] Richardson, L., Amundsen, M. and Ruby, S. (2013) RESTful Web APIs. O’Reilly Media.

[5] 高静, 段会川. JSON 数据传输效率研究[J]. 计算机工程与设计, 2011, 32(7): 2267-2270 .

[6] Lang, J., Liu, Y.B. and Xiong, S.Y. (2010) Data Integration Method Based on SOA Software Architecture. *Jisuanji Yingyong/Journal of Computer Applications*, **30**, 2370-2373. https://doi.org/10.3724/SP.J.1087.2010.02370

[7] Ives, Z.G. and Lu, Y. (2000) XML Query Languages in Practice: An Evaluation. *Proceedings of the* 1*st International Conference on Web-Age Information Management*, June 2000, Shanghai.

[8] Bertino, E. and Catania, B. (2001) Integrating XML and Databases. *IEEE Internet Computing*, **5**, 84-88. https://doi.org/10.1109/4236.939454

[9] Ahmed, R., DeSmedt, P., Du, W., *et al.* (1999) The Pegasus Heterogeneous Multidatabase Systems. *Computer*, **24**, 19-27. https://doi.org/10.1109/2.116885

[10] 李俊, 李勇. 联邦式异构数据库应用系统的集成框架和实现技术的研究[J]. 计算机应用研究, 2001, 18(4): 19-22.