文章编号:1009-3087(2015)05-0123-07 DOI:10.15961/j.jsuese.2015.05.018

基于 PROV 的 ETL 起源信息统一表达机制

柯 洁[1],董红斌[2*],梁意文[1],谭成予[1],艾 勇[3]
(1.武汉大学 计算机学院,湖北 武汉 430072;2.武汉大学 国际软件学院,湖北 武汉 430079;
3.中南民族大学 计算机科学学院,湖北 武汉 430074)

摘 要:在异构环境下,目前数据起源研究主要基于 OPM 模型来表示数据在 ETL 中的来源过程,存在着起源概念不统一、词汇使用混乱以及无法提供标准化访问等问题。基于 W3C 的 PROV 模型,提出了 ETL 起源信息的统一表达机制。该机制首先对 ETL 过程的起源概念及其关系进行了统一描述。然后,针对 ETL 过程特殊的语义表达需求,建立了多粒度的 ETL 起源词汇表。最终,建立在 RDF 之上的标准化查询机制提高了起源信息的可访问性。

关键词:ETL;数据起源;互操作性;PROV;OPM
中图分类号:TP391.1 **文献标志码**:A

A PROV Based Representation of Data Provenance for ETL Process

KE Jie[1], *DONG Hongbin*[2*], *LIANG Yiwen*[1], *TAN Chengyu*[1], *AI Yong*[3]
(1. Computer School, Wuhan Univ., Wuhan 430072, China; 2. International School of Software, Wuhan Univ., Wuhan 430079, China;
3. College of Computer Sci., South-Central Univ. for Nationalities, Wuhan 430074, China)

Abstract: In heterogeneous environment, data provenance information in ETL is represented on the basis of OPM. However, there is still a lack of consensus on conceptual representation of ETL provenance information, usage of provenance vocabulary and a consolidated access mode. A unified provenance representation mechanism, which was based on PROV, was proposed for ETL. Firstly, it presented a concept representation mechanism for ETL, which demonstrated primary provenance concepts and their relationships. Secondly, it constructed a multi-granularity vocabulary to fulfill the requirement of expressing provenance information on different abstraction levels. Finally, a standard access mode was proposed in which provenance information was organized into two levels, the bottom one was described with RDF, and the above level was formed based on query of the former.

Key words: ETL; data provenance; interoperability; PROV; OP

数据起源(data provenance)是对数据产生、传输或影响的实体和过程的描述[1]。在数据仓库中,由于起源信息能够重现数据抽取(extract)、转换(transform)并加载(load)到数据仓库的整个过程,因此被广泛地应用于数据仓库中的数据质量评估、ETL 转换过程优化等方面[2]。

随着企业分布式系统的广泛应用,数据的多源异构性不断上升。承载信息的数据源往往分布在异构的系统平台,并以不同的结构类型存在(例如结构化关系表、无结构化的平面文件等)[3]。在这种环境下,起源信息可能会以不同的数据模型和语言来进行描述。而且,起源信息的产生和消费是 2 个独立的环节,产生起源信息的系统往往并不最终利用这些信息,而由其他系统或第三方来对起源信息进行分析[4]。因此,在异构环境下,实际应用中要求起源信息必须具有一定互操作性。

起源的互操作性(interoperability)指各系统在起源信息的概念表达、语义描述以及访问方式上达成的一致,目的是促进起源信息在系统间的交换和利用[4]。虽然多数系统能够提供内部数据处理的

收稿日期:2015-01-11
基金项目:国家自然科学基金面上项目资助(61170306)
作者简介:柯 洁(1986—),男,博士生. 研究方向:数据起源. E-mail:madfrog.jk@gmail.com
* 通信联系人 E-mail:hbdong@whu.edu.cn

起源信息,但是,由于缺乏互操作性使这些信息很难在系统间进行共享和使用。例如,在不同的系统中,对于代理和活动之间的关系可能会采用起源词汇表OPMV[5]中的 wasControlledBy 来描述,也有可能表示为 PROV-O[6]中 wasAssociatedWith。对这些不同形式的描述、词汇语义的错误理解很可能会导致技术人员在数据质量评估时产生错误的判断,提升ETL 中起源信息的互操作性成为一个亟待解决的问题。

数据仓库中,对起源互操作性的研究仍处于起步阶段。早期的研究关注于数据仓库环境下起源信息的追踪,较少涉及各系统间的互操作性,OPM(open provenance model)标准[7]的发起使得该问题开始受到了关注。Freitas 等[8-9]基于 OPM 首次提出了 ETL 过程的起源信息表达机制,该表达机制分为3层。底层基于 OPM 的工作流表示,中间为扩展于 OPMV 的词汇表 cogs,最上层则为领域相关的表示层。

然而,目前对于起源互操作性的研究工作仍存在着不足。首先,ETL 运行时的操作逻辑并没有得到清晰的表示。Freitas 等[8-9]直接将 OPM 的抽象结构对应到 ETL 中,这种做法无法准确表示处理的内部结构,例如,缺少对操作中数据的输入输出、参数的输入以及模式映射关系的表达。其次,在多粒度的起源表达上,虽然文献[8-9]中列举了较多的词汇,但是对这些词汇的粒度层次关系并没有相关阐述,词汇的使用也较为混乱。最后,在起源信息的查询访问上,相关的工作也没有提供具体的措施。

PROV[10]是由 W3C 起源工作小组提出的数据起源标准,其提出的一个主要目的就是为了提升起源信息的互操作性,为作者的研究工作提供了契机。该模型从抽象层次上统一了起源过程涉及的实体(Entity)、活动(Activity)以及代理(Agent)之间的关系,并基于语义 Web 标准提供了起源信息在系统间交换的标准格式,可以作为不同系统间起源建模的标准和已有起源词汇相互映射的桥梁。然而,PROV 并非一个完整的起源模型,针对特定领域(如 ETL 过程)需要依据实际情况对其进行扩充。

针对 ETL 起源信息在互操作性上的不足,作者提出了基于 PROV 的起源信息统一表达机制。首先,该机制描述了 ETL 过程中的主要概念以及相互之间的关系。其次,针对起源信息在多粒度层次上准确表达的需求,该机制对 PROV 词汇表进行了扩展,统一了 ETL 不同粒度上的起源词汇表达。最后,针对起源信息不同层次的访问需求,将起源信息组织划分为2个层次,底层采用 RDF 描述语言形成基本起源图,并在此基础上建立了标准化查询以支持高粒度的起源信息访问。

1 相关工作

1.1 数据仓库中起源信息的互操作性

在数据仓库早期的起源研究中,互操作性并不是研究的重点,研究者主要关注于起源信息的追踪技术[11-13]。企业内外部环境的变化(例如同社交网络的结合)以及数据的高速增长使得数据的多源异构性不断上升,起源信息的互操作性开始受到研究者的关注。但目前相关的研究并不多见,Freitas 等[8-9]重点关注 ETL 中起源信息的互操作性,基于 OPM 模型建立了 ETL 起源表达机制以及一套起源词汇表 cogs。Freitas 等[8-9]将起源的表达划分成3个层次,底层完全基于 OPM 的工作流机制,中间层则是基于 cogs 词汇,最高层为领域相关的表达。对于 ETL 处理的内部结构,例如,对于操作涉及的输入和输出、参数等相关因素,其概念的界定以及之间的关系应如何表达,Freitas 等[8-9]并没有明确的进行阐述,这也使得其词汇的使用较为混乱。

1.2 PROV 起源模型

在起源研究领域,PROV 已经成为了最新的标准。PROV[10]基于 OPM 模型发展而来,目标是成为开放环境下(例如 Web 环境)起源建模的标准模型,以保证起源信息的互操作性。PROV 模型与具体的应用领域无关,是一个高层次的起源模型,其从抽象层次上描述了实体(Entity)、活动(Activity)以及代理(Agent)之间的关系,用以指明哪些用户、利用哪些条件、参与了哪些活动、产生了什么样的后果,主要被用于描述数据在系统中的处理过程,这些概念的相互关系如图1所示。

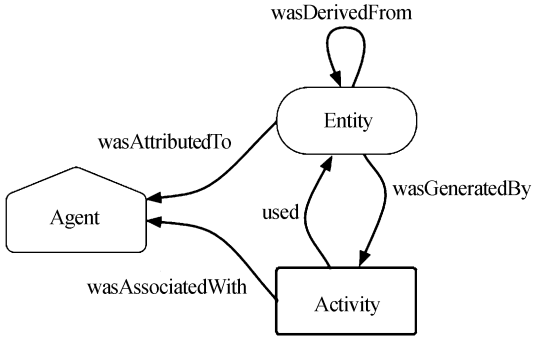

图1 W3C PROV 模型的基本结构

Fig. 1 An overview of the structure of PROV

作为一个抽象模型,PROV 并不针对具体的应用环境,而只是对于各领域下数据起源过程的一般化表达。因此,如果要将 PROV 应用到特定的环境中,必须根据具体情况进行相应的扩展,否则会带来诸如语义不准确等问题。在 PROV 模型产生以后,将其应用到具体的领域成为了研究的热点,目前应用的领域主要集中于科学工作流[14] 和 Web[15] 环境。

2　ETL 起源信息互操作性的需求

下面主要讨论在 ETL 起源互操作性上所要到达的目标。图 2 展示了在分布式环境下,ETL 处理过程的示例,图 2 中,椭圆形表示数据、矩形表示操作,实线部分为系统 s 中运行的 ETL 处理实例 wf,虚线部分为系统 s_ex 中的实例 wf_ex。wf_ex 运行结束后,其产生的数据 ex_d2 被 wf 所采用,操作 op2 则利用数据 d2 和 ex_d2 产生输出 d3。现要追踪数据 d3 的来源,其来源于 2 个实例 wf 和 wf_ex 的处理过程。如果 2 个系统中的起源信息采用不同的表达方式,在起源信息缺乏互操作性的情况下,d3 的起源追踪过程将在连线 u_1 处中断。

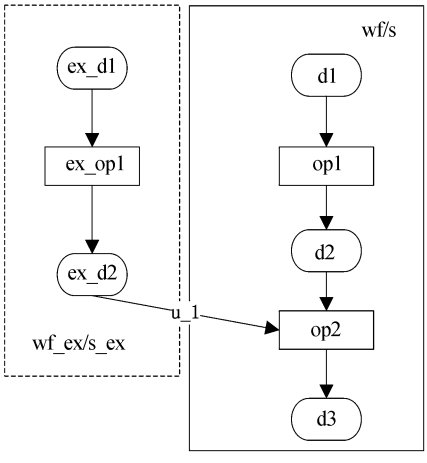

图 2　分布式环境下 ETL 执行流程

Fig. 2　ETL workflow in distributed environment

结合以上示例,作者认为起源互操作性的需求可以归纳为以下几点:

1)ETL 过程概念表达上的统一

PROV 描述了数据起源的通用过程,但仅凭这些抽象表示并不能准确表达 ETL 中的起源过程。例如,在利用 PROV 时,used 关系会被同时用于表示操作中的数据输入和用户输入的参数,从而无法区分这 2 种性质不同的关系。因此,理清 ETL 中的相关概念以及之间的关系是研究的一个主要目标。

2)起源信息的粒度层次

PROV-O[6] 提供了较为全面的词汇表达,可以同 Dublin Core 之间进行映射,并且还允许不同的层次的起源描述在一张图中共存,但没有规定起源信息的粒度究竟以何种原则划分,将在 3.2 节探讨该问题。

3)起源信息的访问

W3C PROV 工作小组要求对于某一个系统中起源信息,一经发布,用户就应该可以依据唯一的标识符(例如 URI)或查询服务来获取对应的信息记录,从而支持其在不同系统之间的共享。但用户对起源信息的访问不仅限于简单的查询,还包括不同角度、粒度层次的访问需求,起源信息应当以合理的方式组织从而提高可访问性。

3　ETL 过程的起源表达机制

基于起源信息互操作性的需求,本节主要描述如何利用 PROV 来表示 ETL 过程中的起源信息。

3.1　ETL 起源过程的概念表达

ETL 本质上是数据从数据源向数据仓库流动的过程,处理过程包括数据的抽取、清洗、转换以及加载。这些处理业务过程按照某种形式(例如 Microsoft SQL Server 2008 Integration Services 以 XML 的形式)被提前定义,可以定时或周期性地执行。运行时系统间会存在如图 2 所示的数据流动。在系统的内部,具体的处理逻辑则由一系列操作组成,待处理的数据依次从上一个操作流向下一个操作节点,形成一个有向无环图。这一过程同工作流的执行过程相似,因此,将 ETL 业务流程看成是工作流的过程。

在 ETL 的运行过程中,每个操作由某个用户负责执行,并运行于特定的软件平台上。这些操作除了接收数据的输入和产生输出外,部分操作还会接收用户输入的参数,并依照预定义的模式映射关系来运行。图 2 中以操作为中心,基于 PROV 模型对 ETL 工作流中的概念以及相互间的关系进行了表示,主要包括 3 类信息:

1)ETL 工作流定义信息

ETL Workflow 表示 ETL 工作流的过程定义,包括 ETL 的数据源描述、主要处理步骤等,由工作流开发人员预先定义。工作流的定义以文件的形式存在(例如 XML),定义了 ETL 过程中的主要步骤,用以约束整个工作流的执行。采用PROV中的计划(Plan)

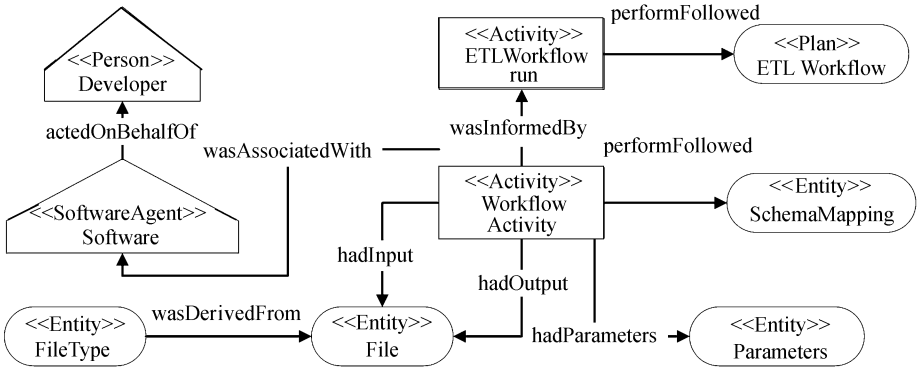

图3 基于 PROV 的 ETL 起源表示

Fig. 3 PROV based representation of ETL provenance

类型表示工作流定义,其相关描述信息还包括工作流定义的创建者、创建时间、定义文件的类型等信息。

2)ETL 运行时信息

运行时信息为 ETL 起源信息的主体。ETLWorkflow run 为依据工作流定义产生的运行实例。一次运行实例可能运行 ETL Workflow 定义的全部或者部分步骤,并且一些步骤可能会被重复运行。例如,数据的清洗过程中可能会存在不断的改进,直到数据符合清洗的标准。工作流实例在 ETL Workflow 的指导下运行,这种计划的约束作用在 PROV 中采用的是对代理和活动之间 wasAssociatedWith 关系进行描述来实现。在 ETL 工作流中,这种约束作用可以更为直接地体现计划和工作流实例之间的关系。因此,采用扩展的关系 performFollowed 进行表示。

一个操作 Workflow Activity(包括用户手工操作、自定义程序操作或是 ETL 工具提供的预定义操作)相当于对 ETL Workflow 定义中步骤的一个实例,一次运行实例由多个操作构成。每个操作由 ETL 实例负责启动和结束,因此它们之间的关系采用 PROV 中的 wasInformedBy 来表示。采用扩展的 hadInput、hadOutput 关系分别表示操作 Workflow Activity与数据的输入、输出(概念上表示为File)之间的关系,用 hadParameters 表示操作与参数之间的关系。此外,对于一些操作步骤,预定义的模式映射关系会以计划的形式来约束操作的行为,同样采用扩展的 performFollowed 关系来表示。

3)用户参与信息

软件代理 SoftwareAgent 表示运行 ETL 实例的软件环境,比如运行软件平台、ETL 执行引擎等。软件在开发人员 Developer 的指导下运行。相互之间的关系采用 PROV 的 actedOnBehalfOf 来表示。工作流的定义 ETL Workflow 也会有相应的人员参与定义,但是这一点不在讨论范围之内,不做重点讨论。

3.2 多粒度起源词汇表

在企业中,管理人员往往更为关注数据处理的全貌,需要高层次的起源信息,而技术人员则需要了解更多的细节性的信息。例如,对于某一项异常数据,技术人员可能会着重查看在数据清洗阶段某个开发人员具体进行了哪些操作、这些操作都产生了什么样的结果。

不同粒度层次起源信息表达需要不同层次的起源词汇。PROV 模型提供了较为丰富的词汇表来对起源过程进行抽象表达,包括 30 个类和 50 个关系。对于 ETL 过程,表 1 展示了部分词汇的扩展。

表1 多粒度的起源词汇表(部分)

Tab. 1 Multi-granularity provenance vocabulary(part)

实体类型	实体类型描述	活动类型	活动类型描述	关系类型	关系类型描述
ETLWorkflow	ETL 工作流定义	rowMerge	合并行	hadInput	数据输入
table	关系表	rowAggregate	聚合行	hadOutput	数据输出
flatFile	平面文件	rowSplit	拆分行	hadParameters	参数输入
excel	Excel 表格	rowDelete	删除行	performFollowed	依据计划执行

ETL 过程中的核心实体为各种类型的数据对象。这些数据对象包括结构化的关系表、XML 文件、Excel 表单以及 json 格式文件等,也包括半结构化平面文件、日志文件等。此外,实体还涉及 ETL

工作流的定义文件、模式映射文件、视图等。

在 PROV 中,活动指作用于实体上,并产生新的实体或使实体的状态发生改变的行为。ETL 过程活动主要包括数据抽取、清洗、转换以及加载 4 个子过程。在每一个子过程中,具体的操作不尽相同。例如,在数据清洗阶段,操作包括删除行、删除列、填充列等;在转换阶段,操作包括行合并、行拆分、列合并、列拆分等操作。图 4 为扩展的 PROV Activity 词汇层次结构。词汇所表示的操作粒度从上往下依次减小。ETL run 表示工作流中的所有操作,作者对相应的实体、活动和对应的关系进行了扩展。词汇在数据抽取、清洗、转换和加载共 4 个操作下进行细分。需要说明的是,数据清洗和转换并没有严格的逻辑界限,因此图 4 中操作会出现交叉的情况。此外,还对关系词汇进行了扩展,主要有数据的输入、输出、参数输入等,在此不再赘述。

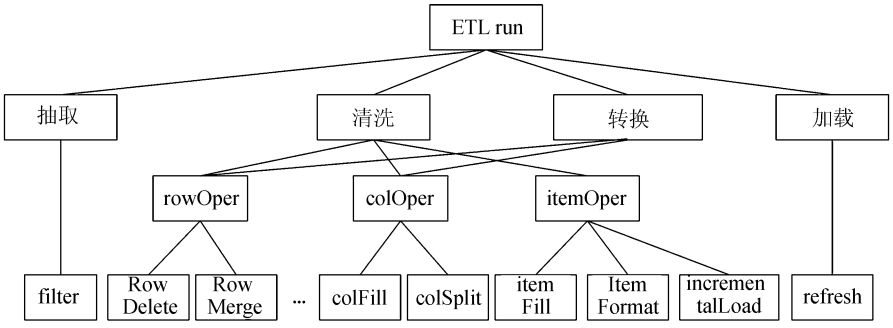

图 4　PROV Activity 词汇的多粒度扩展

Fig. 4　Extended vocabulary of PROV Activity

为了说明扩展词汇表的应用,以下采用资源描述框架 RDF(已被作为 PROV 的标准描述语言中的一种)来对图 1 中的 op2 操作进行描述。具体的描述信息如下:

```
% prefix part was ignored
:op2
  a prov:Activity;
  prov:startedAtTime   "2012-04-15T13:00:00—
                        04:00"^^xsd:dateTime;
  prov:hadInput        :d2,:ex_d2;
  prov:hadParameters   :p1;
  prov:hadOutput       :d3;
  prov:performFollowed :m1;
  prov:wasInformedBy   :wf_run;
  prov:wasAssociatedWith :dev;
  prov:endAtTime       "2012-04-15T14:00:00—
                        04:00"^^xsd:dateTime;
:d2
  a prov:Entity;
  prov:type            "table";
  prov:atLocation      "http://192.168.1.123/wf/
                        op/d2"^^xsd:string;
:ex_d2
  a prov:Entity;
  prov:type            "table";
  prov:atLocation      "http://192.168.1.133/wf_ex/
                        op/ex_d2"^^xsd:string;
:d3
  a prov:Entity;
  prov:type            "table";
  prov:atLocation      "http://192.168.1.123/wf/
                        op/d3"^^xsd:string;
:m1
  a prov:Entity;
  prov:type            "XML";
  prov:atLocation      "http://192.168.1.123/wf/
                        op/m1"^^xsd:string;
:dev
  a prov:Person;
  staffNo              "376-980"^^xsd:string
```

3.3　起源信息的查询及访问

起源信息的查询访问也是互操作性的一个重要方面。在 PROV 的标准中,PROV-AQ[15] 提供了起源信息访问的最佳实践,指出起源信息作为一种信息资源其应该能够通过 URI 唯一地标识,用户可以通过对 URI 地址解引用或通过访问服务的 URI 来访问相应的信息。并且还指出,用户主要从 3 种角度来关注起源信息,包括以用户为中心、以数据为中心和以过程为中心。但并没有提供具体的查询访问实现机制。

作者认为对于 ETL 过程,用户对于这 3 个方面起源信息的查询可进一步转化为类似于如下的问

题:

1)以用户为中心:用户 u 参与了哪些具体的数据产生过程、导致数据目前状态都有哪些用户参与;

2)以数据为中心:数据对象 d 是由哪些数据产生的;

3)以过程为中心:具体到某一个数据对象,其产生的完整过程是什么样的,包括参与的人员、操作以及涉及的数据等。

为了满足用户对起源信息的访问,将 ETL 起源信息的组织划分为 2 层。首先,底层采用图 3 所示的表达机制,具体的描述语言可以采用 RDF,以此形成基本的起源图。其次,针对第 2 节提到的较高层次的起源信息访问,系统基于 SPARQL 提供查询服务,并将服务的 URI 暴露给用户,查询的示例如下所示:

```
@prefix prov:    <http://www.w3.org/ns/prov#>
#以代理为中心的查询
SELECT ?activityName?startTime?endTime
FROM <prov-resource-URI>
WHERE {
    ?specificalActivity    prov:wasAssociatedWith
                           "dev";
    prov:Activity?activityName;
    prov:startedAtTime?startTime;
    prov:endedAtTime?endTime;
}
ORDER BY ASC [?startTime];
#以数据为中心的查询
@prefix prov:http://www.w3.org/ns/prov#
SELECT ?linkAction
FROM <prov-resource-URI>
WHERE {
    ?action hadOutput:d2.
    ?action:wasInformedBy+ ?linkAction.
}
```

4 结 论

企业内部的异构环境对 ETL 过程的起源信息追踪提出了新的需求,起源信息必须能够支持不同系统间的互操作性,从而支持完整起源信息的追踪。在 W3C PROV 模型的基础之上,建立了 ETL 过程起源信息的统一表达机制,包括起源概念及其关系的统一描述、多粒度的词汇表以及标准化的查询机制。该机制为 PROV 模型在 ETL 领域的应用,在下一步的工作中,将从 2 个方面开展研究:

1)依据该表达机制搭建原型系统,对该机制的表达能力进行验证。

2)基于多系统间起源信息的共享,建立适用于 ETL 起源的约束和推理机制,实现异构环境下 ETL 起源信息的应用。

参考文献:

[1]What is Provenance[EB/OL].[2012-06-30].http://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance.

[2]Simmhan Y L,Plale B,Gannon D. A survey of data provenance techniques[R/OL].http://www.cs.indiana.edu/l/www/ftp/techreports/TR618.pdf.

[3]Jarke M,Jeusfeld M A,Quix C J,et al. Data warehouse architecture and quality:Impact and open challenges[M]//Seminal Contributions to Information Systems Engineering. Berlin:Springer,2013:183-189.

[4]Provenance Interoperability[EB/OL].[2011-10-17].http://www.w3.org/2011/prov/wiki/Interoperability.

[5]Zhao J. The open provenance model vocabulary[R]. New York:ACM,2012.

[6]Lebo T,Sahoo S,McGuinness D,et al. Prov-O:The PROV ontology[R/OL].[2013-04-30].http://www.w3.org/TR/2013/REC-prov-o-20130430.

[7]Moreau L,Clifford B,Freire J,et al. The open provenance model core specification(v1.1)[J]. Future Generation Computer Systems,2011,27(6):743-756.

[8]Omitola T,Freitas A,Curry E,et al. Capturing interactive data transformation operations using provenance workflows[M]//The Semantic Web:ESWC 2012 Satellite Events. Berlin:Springer,2012:29-42.

[9]Freitas A,Kämpgen B,Oliveira J G,et al. Representing interoperable provenance descriptions for ETL workflows[M]//The

Semantic Web:ESWC 2012 Satellite Events. Berlin:Springer,2012:43 – 57.

[10] Moreau L, Missier P, Belhajjame K, et al. Prov-dm: The PROV data model[R/OL]. [2013 – 04 – 30]. http://www.w3.org/TR/2013/REC – prov – dm – 20130430.

[11] Cui Y, Widom J. Lineage tracing for general data warehouse transformations[J]. The VLDB Journal:The International Journal on Very Large Data Bases,2003,12(1):41 – 58.

[12] Fan Hao. Data lineage tracing in data warehousing environments[M]//Data Management:Data, Data Everywhere. Berlin:Springer,2007:25 – 36.

[13] Wang T, Dai C. Tracing Data Provenance Based on Inverse Mechanism in ETL[C]//Proceedings of 2012 International Conference on Information Technology and Management Science (ICITMS 2012). Berlin:Springer,2013:553 – 561.

[14] Missier P, Dey S, Belhajjame K, et al. D-PROV:Extending the PROV provenance model with workflow structure [C]//Proceedings of the 5th USENIX Workshop on the Theory and Practice of Provenance(TaPP'13). Berkeley:USENIX Association,2013, Article No. 9.

[15] Baillie C, Edwards P, Pignotti E. Quality assessment, provenance, and the web of linked sensor data[M]// Provenance and Annotation of Data and Processes. Berlin:Springer,2012:220 – 222.

[16] Klyne G, Groth P, Moreau L, et al. PROV-AQ:Provenance access and query[R/OL]. [2013 – 04 – 30]. http://www.w3.org/TR/2013/NOTE – prov – aq – 20130430.

(编辑　赵　婧)