INTERNATIONAL JOURNAL OF ENHANCED RESEARCH IN MANAGEMENT & COMPUTER APPLICATIONS
VOL. 2, ISSUE 1, JAN. 2013 ISSN NO: 2319-7471

Derivation Rule Dependency and Data Provenance Semantics

Choon Lee[1], Don Kerr[2]

[1] Department of Business IT, College of Economics and Business, Kookmin University, Korea

[2] Faculty of Business, University of Sunshine Coast, Australia

cylee@kookmin.ac.kr, dkerr@usc.edu.au

ABSTRACT

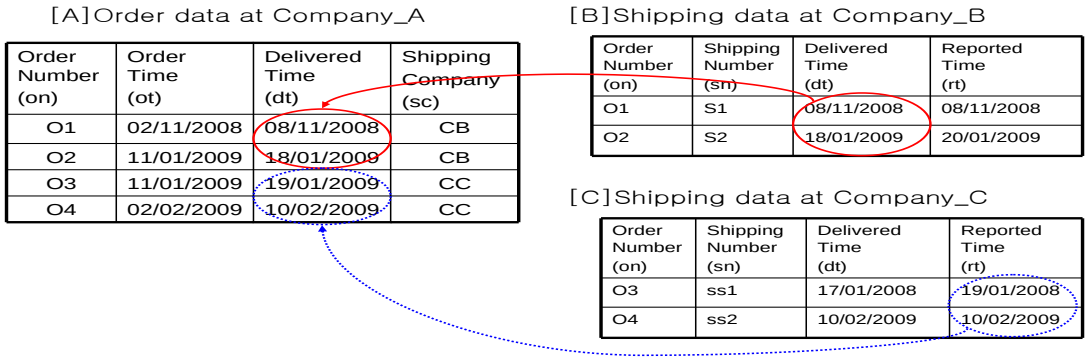
This paper proposes a derivation rule dependency (DRD) to represent data provenance semantics. Data provenances are mostly for tracing data lineage and data creation processes. We propose to treat data provenance semantics as derivation dependencies meta-data. This study is a kind of conceptual one. We are in the process of building a prototype that manages and apply it to real world examples. Further, we plan to test effectiveness of data provenance meta-data management using an agent-based simulation model.

1. INTRODUCTION

Information sharing has been a major issue in current information systems researches. As information systems are getting more complicated, users are provided with data that are not their own creation. It means that users are exposed to data that they are unfamiliar with and may misuse them. These phenomena are inevitable in complex inter- and/or intra-corporate information systems. To reduce these kinds of pitfalls, data provenances have been proposed to be provided with data. I.e., data are not provided alone but with their provenances. Data provenances help users understand data and prevent their misuse [5][7].

Data provenances have been treated as an important issue in areas of engineering and/or scientific researches. To use scientific data correctly, it is necessary to understand environments that those data are created [11][12]. . For example, we need to trace sources of protein data to interpret them correctly for coronary artery disease analysis [2].

As compared to scientific data, data provenance has not been a big issue for business data. However, business data are not immune to provenance semantics. For example, orders are shipped via multiple shipping companies. Let us assume that, as shown in [Figure 1], one shipping company reports it as a time that a customer receive an ordered product, and the other report it as a time that all administrative works have been finished after delivering ordered products. Then it might be erroneous to mix these [delivered time]s to measure order delivery. To prevent this kind of an error, order delivered times are to be managed based on their provenance semantics.



[Figure 1] A derivation rule dependency example

In sum, to integrate information across internal departments and/or external organizations, data provenances are to be defined and managed as meta-data. For this purpose, we propose derivation rule dependencies that represent data provenances among database attributes.

2. Derivation Rule Dependency (DRD)

This study proposes to formalize provenance semantics based on data derivation rules. Derivation rules specify input attributes that are used to derive target attribute.

Derivation Rule Dependency: An Attribute Y is said to be derivationally dependent on attributes X1,…,Xn if there exist a process and/or function that produces Y's values from X's values, which shall be denoted as

X1, …, Xn =R=> Y,

where, Y is derived(target, output) attribute,
X1, …, Xn are deriving(source, input) attributes.

However, in most cases, values of an attribute are derived by multiple rules. For example, in <Figure 1>, an attribute [Derived Time] for [Order Number] O1 is derived differently from the one for [Order Number] O3. In a nut shell, derivation rules are conditional on values of other attributes, which shall be called derivation conditions.

Incorporating derivation conditions, derivation rule dependency is revised as follows:

Conditional Derivation Rule Dependency(CDRD)

A derivation rule among target(derived) attribute Y and source(deriving) attributes X1, …, Xn, is expanded to include conditions that the rule is applied. Including conditions that derivation rules are applied, a derivation rule dependency is denoted as follows:

X1, …, Xn =R=> Y | Q

where, Y is derived attribute,
X1, …, Xn are deriving attributes,
Q is a condition that a derivation rule is applied.

In derivation rule dependencies, conditions specify objects that data derivation rules are applied to. That is, it specifies records that the rule is applied to. For example, in [figure 1], the rule indicated by a solid line is stated by the following derivation rule dependency:

[B].[dt] =R=> [A].[dt] | Q1

Q1 specifies records whose [delivered time] are derived from company B's [delivered time]. The following SQL specifies those records whose shipping company is 'Company_B';

Q1 : Select * From A Where [A].[sc] = 'CB';

In the sense that conditions specify records to which rules are applied, they are abbreviated to include WHERE clauses of SQL, as shown in the following:

[B].[dt] =R=> [A].[dt] | ([A].[sc] = 'CB')

Based on the same rationale, the rule indicated by a dotted line is stated by the following derivation rule dependency:

[C].[rt] =R=> [A].[dt] | ([A].[sc] = 'CC')

This study proposes that there exist multiple derivational rules for an attribute. Thus, even though it is not specifically mentioned, a derivation rule dependency means a conditional derivation rule dependency.

Recursive Derivation Rules

In a derivation rule, we assumed that target data are derived from source data. That is, new data are created by derivation processes. However, in some cases, data are just updated rather than created. For example, a sales price might be calculated by multiplying a percentage change to the previous sales price.

This kind of data update happens when the derived attribute name is identical to one of deriving attribute names. In this case, data derivation is a kind of recursive process. Recursive derivation processes are easily incorporated into the proposed conditional derivation rule dependency as follow:

X1, …, Xn, Y =R=> Y | Q

where, Y is target attribute,
X1, …, Xn are source attributes,
Q is a condition that a data derivation rule is applied.

Properties of Derivation Rule Dependency:

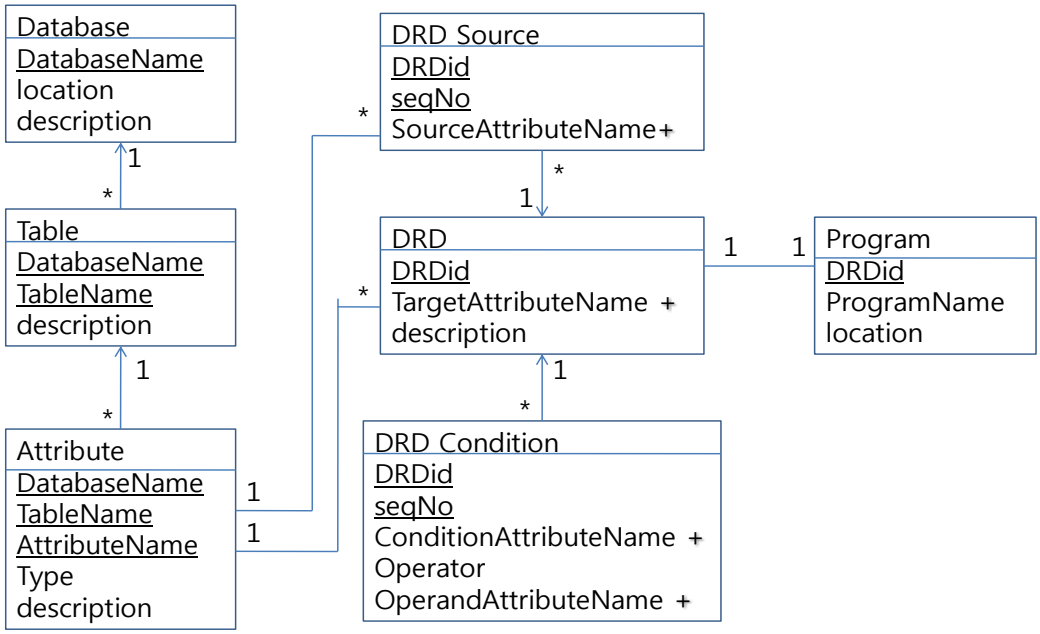
A derivation rule dependency connects target data to source data. Thus, if we combine each dependency, they form a directed acyclic graph (DAG) with recursive arcs. Nodes are database attribute names and arcs are derivation rule dependencies. In the sense that derivation rules are depicted as a DAG, they satisfy following properties:

(a) Transitivity
If X =R=> Y|Q1 and Y =R=> Z|Q2, then X =R=> Z|Q1∧Q2
(b) Decomposition:
If X =R=> YZ|Q1, then X =R=> Y|Q1, X =R=> Z|Q1
(c) Pseudo transitivity:
If X =R=> Y|Q1, and YZ =R=> W|Q2, XZ =R=> W| Q1∧Q2
(d) Union:
If X =R=> Y|Q1, X =R=> Z|Q1, then X =R=> YZ|Q1

Derivation rule dependencies satisfy a decomposition and union axioms. Thus, from now on, we assume that derivation rule dependencies are defined for a single attribute, which shall be called an atomic derived rule dependency.

3. A Meta-database Schema for Derived Rule Dependencies

Derivation rules dependencies provide indispensable metadata for information users. Users might have better understanding of data provenance semantics via derivation rule dependencies. We propose a meta-database schema to include derivation rule dependencies as meta-data. In the schema, derivation processes are mapped to derivation rule dependency. This study assumes that derivation rules may contain arbitrary types of functions. Thus, they are managed separately from derivation rule dependencies.
[Figure 2] shows a conceptual schema to store derivation rule dependency. As shown in the figure, derivation rule dependencies are stored as relations among attributes. In addition, conditions are specified for each derivation rule dependency.



(Note:
+ : An Attributename is a concatenation of Dabasename, Tablename, and AttributeName.
Databasenames and Tablenames are not included for the sake of simplicity)

[Figure 2] Meta-database Schema for Derivation Rule Dependency

Using the proposed meta-database schema, derivation rule dependencies in [Figure 1] is shown as follows:

[A]Order data at Company_A

Order Number (on)	Order Time (ot)	Delivered Time (dt)	Shipping Company (sc)
O1	02/11/2008	08/11/2008	CB
O2	11/01/2009	18/01/2009	CB
O3	11/01/2009	19/01/2009	CC
O4	02/02/2009	10/02/2009	CC

[B]Shipping data at Company_B

Order Number (on)	Shipping Number (sn)	Delivered Time (dt)	Reported Time (rt)
O1	S1	08/11/2008	08/11/2008
O2	S2	18/01/2009	20/01/2009

[C]Shipping data at Company_C

Order Number (on)	Shipping Number (sn)	Delivered Time (dt)	Reported Time (rt)
O3	ss1	17/01/2008	19/01/2008
O4	ss2	10/02/2009	10/02/2009

[D]DRD

DRD id (id)	Target Attribute Name (ta)
D1	[A].[dt]
D2	[A].[dt]

[E]DRD source

DRD id	Sequence (sq)	Source Attribute Name (sa)
D1	1	[B].[dt]
D2	1	[C].[rt]

[F]DRD Condition

DRD Id (id)	Sequence (sq)	Condition Attribute Name (ca)	Operator (op)	Operand (od)	Operand Type (tp)
D1	1	[A].[sc]	=	'CB'	value
D2	1	[A].[sc]	=	'CC'	value

[Figure 3] Derivation Rule Dependencies Metadata (Example)

Querying Derivation Rule Dependencies

A derivation rule dependency represents data lineage at schema level. However, it also includes conditions that provenance rules are applied. Thus, it can be instance-level semantics as well as schema-level provenance.

As shown in [Figure 3], derivation rule dependencies are stored with data in a database. Thus we can query them using SQL. This approach has been proved to ease management of metadata [9].

4. Application of DRD to Data Management

Database design and derived rule dependency

Database schemas in [Figure 4] convey the same amount of information. <Schema A> is a kind of normalized one. And, <Schema B> includes a redundant attribute - [Shipping Company].

From a data management viewpoint, <Schema A> is preferred to <Schema B> because the former is better normalized. However, if we shift to a user's view on provenance semantics, <Schema A> does not provide any information to differentiate derivation rules for [Delivered Time]. In other words, it is not easy to describe that [Shipping Company] apply different rules for [Delivered Data]. As compared to <Schema A>, <Schema B> includes [Shipping Company] which is a condition attribute for derivation . Thus it is more easily understood that [Delivered Time] is created by different rules by different [Shipping Company].

<Schema A>

Order

Order Number (on)	Order Time (ot)	Delivered Time (dt)	Shipping Number (sn)
O1	02/11/2008	08/11/2008	S1
O2	11/01/2009	18/01/2009	S1
O3	11/01/2009	19/01/2009	ss1
O4	02/02/2009	10/02/2009	ss2

PK: [Order Number]

Shipping

Shipping Number (sn)	Shipping Company (sc)	Delivered Time (dt)	Reported Time (rt)
S1	CA	08/11/2008	08/11/2008
S2	CA	18/01/2009	20/01/2009
ss1	CB	17/01/2008	19/01/2008
ss2	CB	10/02/2009	10/02/2009

PK: [Shipping Number]

<Schema B>

Order

Order Number (on)	Order Time (ot)	Delivered Time (dt)	Shipping Company (sc)	Shipping Number (sn)
O1	02/11/2008	08/11/2008	CA	S1
O2	11/01/2009	18/01/2009	CA	S1
O3	11/01/2009	19/01/2009	CB	ss1
O4	02/02/2009	10/02/2009	CB	ss2

PK: [Order Number]

Shipping

Shipping Number (sn)	Shipping Company (sc)	Delivered Time (dt)	Reported Time (rt)
S1	CA	08/11/2008	08/11/2008
S2	CA	18/01/2009	20/01/2009
ss1	CB	17/01/2008	19/01/2008
ss2	CB	10/02/2009	10/02/2009

PK: [Shipping Number]

[Figure 4] A Database Schema with Condition Attributes

Data Provenance Semantics Metadata Management

Data provenances have been proved a major component of meta-data. Thus, there have been many researches to incorporate data provenances into data management. However, they suffer following pitfalls to be used in business applications.

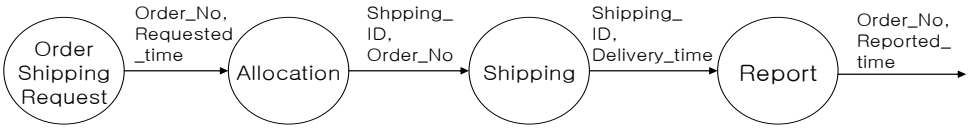
- Most data are assumed to be created and/or copied from source data. Thus, database updates have not been actively treated.
- Data transformation processes are included as a part of meta-data to reproduce derived data from sources. Thus, it makes data provenance meta-data management too complicated to be used by casual data users.

This study assumes that data lineage is defined for a subset of records and incorporates derivation rule dependencies rather than transformations themselves. In this sense, it can ease management of provenances meta-data.

5. Related Studies

Data provenance semantics have been studied extensively in recent years. Much of this previous work is related to workflows. Workflows show data creation procedures by tracing steps and flows of data.

For example, the following shows order shipment workflows. It shows business processes that data go through to process works pictorially. By recording inputs and outputs for each workflow step, we can trace data creation procedures and infer data provenance semantics.



[Figure 5] Order Shipment Workflow – an Example

Data provenance semantics are defined for data instance objects as well as database schema objects. The latter, which is called schema-level provenance, records provenance semantics for each schema object; thus, data provenance semantics are same for all records. The former, which is called instance-level provenance, records provenance semantics for each data instance like order number O1's delivery time; thus, each data set has its own semantic metadata set [8].

A derivation rule dependency represents data lineage at schema level. However, it is different from schema-level provenance semantics in the sense that it includes conditions that provenance rules are applied. Thus, it can be instance-level provenance semantics as well schema-level provenance semantics.

Database annotation is another approach to record data provenance semantics. Annotations are done either for schema objects or for data instances. In the former, annotations are place to schema objects; i.e., data provenance metadata is recorded as an annotation for each schema object. In the latter, annotations are place to record instances; i.e., data provenance metadata is recorded as an annotation for each data item in a database record. In this case, a database includes data annotations as well as data [1][2].

These two approaches to data provenance semantics are different to each other in an architectural view. An annotation approach is based on relational database architecture. Even though annotations are stored in different space from database tables, they are included as a part of a database.

Recent researches on virtual data grid systems include data lineage and data transformation relationships as a part of virtual data catalogue. These systems are based on languages designed to represent derivation procedures [11].

Recently, many studies have proposed to treat data lineage meta-data as a part of database, thus querying them using an ordinary query language like SQL [9][12]. Our study is mostly related to these studies in the sense that we also treat derivation dependencies meta-data with data. Ours is more concerned with business application databases and multiple derivation rules applied to data as compared to these studies.

6. Conclusion and Future Work

There have been clear needs for data lineage and data provenance semantics. In this paper, we presented a derivation rule dependency (DRD) to represent data provenance semantics. DRD is based on assumptions that: (1) values of a single attribute might be derived by different rules, and (2) data lineage processes are too complex to incorporate them into data provenance semantics.

Our DRDs define structural relationship among database attributes. Thus, they are stored as a database meta-data and can be queried using an ordinary query language like SQL. By being treated as meta-data, data provenance semantics are incorporated into ordinary database operations.

This study is an initial step to incorporate derivation rules into database system. We are in process of building a prototype. Further experimental researches are to be done using the system. We hope that the experimental results expand our understanding of provenance semantics and utility of derived rule dependencies.

7. Acknowledgement

This work is supported by the "Korean Research Foundation Grant" (KRF-2009-013-B00021).

8. References

[1]. Moreau, L., Freire, J., Futrelle, J., McGrath, R.E., Myers, J., Pauson, P., "The Open Provenance Model", Dec. 2007.

[2]. Buneman, P., Chapman, A.P., Cheney, J., "Provenance Management in Curated Databases" , Proc. of ACM SIGMOD '06, pp 539-550, June 2006.

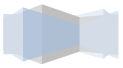
[3]. Bhagwat, D., Chiticariu, Tan, W., Vijayvargiya, G., "An Annotation management system for relational databases", The VLDB Journal(2005), 14(4), 373-396.

[4]. Lee, C., A Knowledge Management Scheme for Meta-data: an Information Structure Graph, Decision Support Systems, 36, 341-354, 2004.

[5]. Heinis, T., & Alonso, G., "Efficient Lineage Tracking For Scientific Workflows" , Proc. of ACM SIGMOD '08, pp 1007-1018, June 2008.

[6]. Zhang, M., Zhang, X., Zhang X., & Prabhakar, "Tracing Lineage Beyond Relational Operators" , VLDB '07, pp 1116-1127, 2007.

[7]. Cui, Y., Widom, J., "Tracing the Lineage of View Data in a Warehousing Environement", ACM Transactions on Database System, 25(2), pp 179-227, June 2000.



[8]. Cui, Y., Widom, J., "Lineage Tracing for General Data Warehouse Transformation", VLDB Journal(2003), 12, pp 41-58, 2003.

[9]. Srivastava, D. & Velegrakis, Y., "Intensional Associations Between Data and Metadata", SIGMOD '07, pp 401-412, 2007.

[10]. Moreau, L., Groth, P., Miles, S., Vazquez-Salceda, J., Ibbotson, J., Jiang S., Munroe, S., Rana, O., Schriber, A., Tan, V., & Varga, L., "The Provenance of Electronic Data, Communications of ACM", 51(4), pp 52-58, 2008.

[11]. Rosental, A., & Sciore, E., "Propagating Integrity Information among Interrelated Databases, Integrity and Internal Control in Information Systems", pp 5-18, 1998

[12]. de Santana, A., & de Carvalho Moura, A., "Metadata to Support Transformations and Data & Metadata Lineage in a Warehousing Environment", Lecture Notes in Computer Science, Volume 3181, pp 249-258, 2004.

[13]. Simmhan, Y., & Plale, B., & Gannon, D., "Query Capabilities of the Karma Provenance Framework, Concurrency and Computation: Practice and Experience", 20, pp441-451, 2008.

[14]. Woodruff, A., & Stonebraker, M., "Supporting Fine-Grained Data Lineage in a Database Visualization Environment", IEEE 13th International Conference on Data Engineering, Birmingham, UK, 1997.

[15]. Foster, I., Vockler, J., Wilde, M., & Zhao, Y., Chimera, "A Virtual Data System for Representing, Querying, and Automating Data Derivation", Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM '02), 2002.

[16]. Velegrakis, Y., Miller, R., & Mylopoulos, J., "Representing and Querying Data Transformation", Proceedings of the 21st International Conference on Data Engineering (ICDE 2005), 2005.