A Flexible Data Structure for Heterogeneous Information Integration

Xin-Yang WANG¹,a*, De-Yu QI¹, Wei-Wei LIN¹ and Na-Qin ZHOU¹

¹Research Institute of Computer Systems, South China University of Technology, Guangzhou 510006 China
awxyyuppie@139.com
*Corresponding author

Keywords: Common data model, Data integration, GDM algebraic operation, GriData.

Abstract. A uniform mechanism to express and operate various data is essential for the development of big data and internet of things, and the common data model provides fundamental supports for the integration of heterogeneous data. By comparing with the advantages and disadvantages of several primary common data models, this paper proposes a new common data model called GriDoc Data Model (GDM). Through the introduction of concepts, such as Paragraph, Section, GriData and GriDoc, the model establishes a flexible mechanism to express different kinds of heterogeneous data. On the basis of the GDM model concept, this paper defines the basic algebraic operations and presents the process of integrating and querying GDM object data.

Introduction

The amount of information is growing in exponential speed as the result of human society production activities, especially the developments of cloud computing and big data technologies prompt data to expand with unprecedented dramatic speed in size, structure and format. It is necessary to integrate all kinds of heterogeneous data with a uniform mechanism.

Generally, the integration of heterogeneous data can be implemented from two aspects of data structure and semantic. For example, the OEM model[1, 2] is for structured, semi-structured and un-structured data and is a simple self-description object model, which allows object nesting. Its variant model the OIM model [3-5] is also for structured and semi-structured data. However, OEM and its variants mix data value with data schema, leading to data repetition and waste of memory. The data models have to be modified accordingly with the change of data source, and their ability to express semantic relations among data is also very limited. Due to the characteristics of content self-description, platform independence, separation of content and display, and good expansibility, etc., XML is also suitable for describing heterogeneous data sources. A lot of XML-based data integration models are available, such as the XML tree data model in [6]; XML DTD is adopted as structural description for data interchange in the SIMS system[7]; the XML-based data integration model XIDM [8-10] is applied in the Panorama system; the SUDAI [11-13] model uses ODM as the description tool and XML as the media to store data description results.

However, there are some problems to describe data semantics with XML. For example,
identical concept is often expressed by different words and meanwhile the same word may indicate different meanings. Thereby, XML has obvious defects in solving semantic conflicts. There are also some other information techniques of data structure. The hypergraph-based common data model HDM [14, 15] realizes schema mappings among data models of ER, UML, relational database and ORM.

The semantic-based heterogeneous information description model ODM [16] and its relevant model language ODL [16] are put forward in the MOMIS[17] system. ODM overcomes the drawbacks that exist in OEM and OIM, saves memory space, improves the retrieval efficiency, and is especially suitable for integrating dynamic and complex heterogeneous data. However, the ODL language is not convenient enough for ordinary people to describe data source and lacks good versatility since it is not a network exchange language.

From the existing heterogeneous data integration models, we know that data model characteristics, such as data description ability, semantic analysis ability, loose coupling relationship with physical data, and certain degree of flexibility, determine the performance of common data model. This paper puts forward a new common data model called the GriDoc Data Model (GDM). The model is capable of describing structured, semi-structured and non-structured data, and has better data description ability. GriDoc objects have rich relationships among each other, and thereby the model also has strong semantic description capability. Meanwhile, the GDM model has superior performance on decoupling with underlying physical data, high reusability, data sharing and flexibility of data operation. The following chapters will further depict these properties.

**GDM Common Data Model**

Definition 1. In GDM model, the minimum data organization unit that has definite data structure is called Paragraph. If denoted by \( P \), Paragraph is a quintuple \( P=<GID, n, t, ref, c> \), where \( GID \) (Global ID) is a global unified identification in GDM model, and is unique; \( n \) denotes Paragraph name; \( t \) denotes the data type represented by Paragraph; \( ref \) is a Boolean variable that indicates the Paragraph data is value or reference; \( c \) is the value of Paragraph data.

Paragraph can represent both data value and attribute value. By treating data value as a special attribute, it is feasible to express any data attribute with Paragraph uniformly, with Paragraph name representing attribute name and Paragraph value representing attribute value. All these attributes are organized by another GDM data object, i.e. Section, defined as Definition 2.

Definition 2. In GDM model, the data organization unit composed of one or more data structure elements is called Section. If denoted by \( S \), \( S \) is a quadruple \( S=<GID, n, D, R> \), where \( GID \) (Global ID) is the global unified identification; \( n \) denotes Section name; \( D \) is the list of GDM objects in \( S \), i.e. \( D=\{d_i | d_i=P_j \text{ or } d_i=S_k, \text{ and } S_k \neq S; i,j,k=1,2,3...n; n \geq 0\} \); \( R \) is the relation list of GDM objects in \( S \), i.e. \( R=\{LR\}, LR=\{<d_i, d_j> | d_i, d_j \in D, 1 \leq i, j \leq n\} \).

The Section data structure is shown in Fig.1.
From Definition 2, it is known that Section is a compound data structure. A Section contains one or more atomic Paragraphs, or one or more Sections except itself, or both. So a Section can include any type and any number of GDM objects (except itself). As is shown in Fig.1(b), the Section $S_1$ contains $S_2$, $S_3$ and $P_4$, while $S_2$ and $S_3$ contain $P_1$, $P_2$ and $P_1$, $P_3$ respectively. When taking $S_2$ and $S_3$ as its children, $S_1$ automatically includes the true child nodes and indirect child nodes (see Definition 4) of $S_2$ and $S_3$, and has good data reusability; since $P_1$ is shared by $S_2$ and $S_3$ at the same time, the model has a good data sharing property.

Definition 3. In GDM model, Paragraph and Section are collectively called GriData (i.e. GDM object). If GriData is denoted by $G$, then $G = \{g_i | g_i=P$ or $g_i=S\}$.

There are various relations among GriDatas.

Definition 4. For $\forall g_p, g_q \in G$, and $g_q \neq P$, if $g_p$ is one of sub elements in $g_q$, then $g_p$ is called the true child node of $g_q$; if $g_p$ is the true child node of some true child node in $g_q$, then $g_p$ is called the indirect child node of $g_q$; similarly, true child nodes of $g_q$’s indirect child nodes are also collectively called indirect child nodes of $g_q$, and so on.

Definition 5. For $\forall g_p, g_q \in G$, and $g_q \neq P$, if $g_p$ is a true child node of $g_q$, then we say $g_q$ properly includes $g_p$, denoted as $g_p \subset g_q$; if $g_p$ is an indirect child node of $g_q$, then we say $g_q$ indirectly includes $g_p$, denoted as $g_p \supset g_q$; otherwise, they are accordingly denoted as $g_p \not\subset g_q$ and $g_p \not\supset g_q$.

Definition 6. For $\forall g_p \in G$, all the true child nodes of $g_p$ constitute the true child node set, denoted by $sub_{dir}(g_p)$; all the indirect child nodes of $g_p$ constitute the indirect child node set, denoted by $sub_{indir}(g_p)$; all the true child nodes and indirect child nodes uniformly form the child node set of $g_p$, denoted by $sub(g_p)$, and $sub(g_p)=sub_{dir}(g_p)+sub_{indir}(g_p)$.

There is sequence among child nodes in GriData, if $g_p$ is the $i$-th child node of $g_q$, it is denoted as $g_p=g_q(i)$.

Definition 7. For $\forall g_p, g_q, g_r \in G$, $g_r \neq P$, $g_p=g_r(i)$, $g_q=g_r(j)$ ($i,j \geq 1$), if $j=i+1$, then $g_r$ is called the direct precursor node of $g_q$, denoted by $g_p \rightarrow g_q$, and accordingly $g_q$ is the direct descendant node of $g_p$; if $j=i+1$, then $g_p$ is called the indirect precursor node of $g_q$, denoted by $g_p \supset g_q$, and accordingly $g_q$ is the indirect descendant node of $g_p$.

Example 1. A table Student in relational database is shown in Fig.2(a), and it’s expressed with GDM model in Fig.2(b) and 2(c).
In Fig. 2(b), the table and every record are denoted by Sections, and every record contains many fields, shown in Fig. 2(c). Each field is also a Section object and contains attributes Value, Type, etc, treating the value of a field as a special attribute. These attributes are atomic and represented by the Paragraph object. Take the field SID for example, it is integer type, and its value is 001. In Fig. 2(c), Paragraph Value represents the value of SID field, where ID is its global ID; n is the name of the Paragraph- “value”; t denotes the data type that Paragraph Value represents, since SID is integer, t=“integer”; ref means whether the value of the Paragraph is reference type or not, if ref=“true”, then c is the “address” of referenced value, otherwise, c is the value of Paragraph Value, here since ref=“false”, thus c=001. Paragraph Type denotes the data type of SID field, and the meanings of GID and n are as previously mentioned; since a data type is usually expressed with string, t=“String”, and the SID field is integer type, thus ref=“false” and c=“integer”.

It is necessary to note that in Paragraph Value t=“integer” and in Paragraph Type c=“integer” both represent the SID field data type. The repetition is indispensable because the former is used for data reading and manipulation, and the latter is for data description and expression. The advantage of this mechanism is especially obvious in integrating non-structured data, as is shown in Example 2.

Example 2. File1, File2 and File3 are of various types and distributed over different network nodes. They are integrated into the GriDoc File through virtual views, shown in Fig. 3.
During the heterogeneous data integration process, a series of operations on GDM data objects are needed. The following will define basic algebraic operations of GDM objects and offer GDM object manipulation functions on the basis of the algebraic operations.

**GDM Object Algebraic Operation**

The GDM object algebraic operation has a characteristic of closeness, namely the results of GDM object algebraic operation are not influenced by data format heterogeneity.

Suppose there are two GriData objects \( g_p=(r_p, V_p, E_p) \) and \( g_q=(r_q, V_q, E_q) \), the GDM object algebraic operations are defined as follows.

1. **GDM Object Union**
   
   \[
   g = g_p + g_q = (r, V, E)
   \]
   
   where \( r = g, \quad V = \{ v_i | \forall v_i \in G, (v_i \in V_p) \lor (v_i \in V_q) \}, \quad E = \{ e_i | e_i = v_i, v_j, \forall v_i, v_j \in G, (v_i \in V_p \land v_j \in V_q) \lor (v_i \in V_q \land v_j \in V_p) \}, \quad e_i \in ((E_p - <r_p, v_i>) \lor (E_q - <r_q, v_j>)) \lor (v_i \rightarrow v_j \lor v_j \rightarrow v_i) \lor (<r, v_i> \lor <r, v_j>)) \}, \]
   
   i.e. the union of \( g_p \)'s and \( g_q \)'s true children sets, denoted by \( g_p + g_q \).

2. **GDM Object Difference**
   
   \[
   g = g_p - g_q = (r, V, E)
   \]
   
   where \( r = g, \quad V = \{ v_i | \forall v_i \in G, v_i \in (V_p \land v_j \not\in g_q) \lor (V_q \land v_j \not\in g_p) \}, \quad E = \{ e_i | e_i = <v_i, v_j>, \forall v_i, v_j \in G, (v_i \in V_p \land v_j \in V_q) \lor (v_i \in V_q \land v_j \in V_p) \land (f(v_i) = \text{true} \land f(v_j) = \text{true}) \}, \quad e_i \in ((E_p \lor E_q) \land (v_j \rightarrow v_i \lor v_i \rightarrow v_j)) \}, \]
   
   i.e. the difference of \( g_p \)'s and \( g_q \)'s true children sets, denoted by \( g_p - g_q \).

3. **GDM Object Selection**
   
   If let \( f \) be selection condition, then
   
   \[
   g = \sigma_f (g_p) = (r, V, E)
   \]
   
   Where \( r = g, \quad V = \{ v_i | \forall v_i \in G, v_i \in (V_p \land f(v_j) = \text{true}) \lor (V_q \land f(v_j) = \text{true}) \}, \quad E = \{ e_i | e_i = <v_i, v_j>, \forall v_i, v_j \in G, (v_i \in V_p \land v_j \in V_q) \lor (v_i \in V_q \land v_j \in V_p) \land (f(v_i) = \text{true} \land f(v_j) = \text{true}) \}, \quad e_i \in ((E_p \lor E_q) \land (v_j \rightarrow v_i \lor v_i \rightarrow v_j)) \}.

4. **GDM Object Projection**
   
   In GriData object \( g_p=(r_p, V_p, E_p) \), given a group of absolute paths \( path_1, path_2, \ldots, path_n \) with respect to \( g_p \), where \( path_k = g_{p_1}/g_{p_2}/\ldots/g_{p_k}, 1 \leq k \leq n \), if the projection of \( g_p \) on \( path \) is denoted by the symbol \( \Pi_{[path_1, \ldots, path_n]}(g_p) \), then
   
   \[
   g = \Pi_{[path_1, \ldots, path_n]}(g_p) = (r, V, E)
   \]
   
   where \( r = g, \quad V = \{ v_i | \forall v_i \in G, 1 \leq k \leq n, (v_i \in (\{path_k\} \land \{r_p\})) \lor (v_i \in \text{end}(path_k)) \}, \quad E = \{ e_i | 1 \leq k \leq n, e_i \in (path_k - <r_p, r_1>) \lor <r, r_1> \lor E_{\text{end}(path_k)} \}.

5. **GDM Object Join**
   
   Suppose there are two GriData objects \( g_1=(r_1, V_1, E_1) \) and \( g_2=(r_2, V_2, E_2) \), \( path_1 \) and \( path_2 \) are absolute paths relative to \( g_1 \) and \( g_2 \) respectively. Let \( f \) be the join condition,
joining all the child nodes of \( \text{end}(\text{path}_2) \) in \( g_2 \) as the child nodes of \( \text{end}(\text{path}_1) \) in \( g_1 \) can be denoted as

\[
g = g_1 \times_{\substack{\text{path},\text{path}_1,\text{f} \\ g_2}} \, g_2 = (r, V, E)
\]

Where \( r=g \),
\[
V = \{ v_i | \forall v_i \in G, \quad (v_i \in ((V_1 - r_1) \lor \text{sub}(\text{end}(\text{path}_2)))) \land f(\text{sub}(\text{end}(\text{path}_1))), \\
\text{sub}(\text{end}(\text{path}_2))) = \text{true} \land (\forall \text{path}_n \in g_1, \ \text{path}_n \in g_2, \ \text{begin}(\text{path}_n) = \text{end}(\text{path}_1) \land \text{begin}(\text{path}_n) = \text{end}(\text{path}_2) \land \text{path}_n \neq \text{path}_n) \},
\]
\[
E = \{ e_i | \forall v_i, v_j, v_k, v_l \in G, \quad v_i \subset g_1, \quad v_j \subset \text{end}(\text{path}_2) \land v_j \subset \text{end}(\text{path}_1), \quad v_k \in \text{sub}(v_j), \quad v_l \subset \text{end}(\text{path}_1), \quad \text{then} e_i \in (E_1 - <r_1, v_i>) \lor <r, v_i> \lor (v_j \subset \text{end}(\text{path}_1) \lor v_k \subset \text{end}(\text{path}_1) \lor v_l \in E_v) \lor (v_j \rightarrow v_i \lor v_j \rightarrow v_l) \}.
\]

(6) GDM Object Cutting

Given a group of absolute paths \( \text{path}_1, \text{path}_2, \ldots, \text{path}_n \), where \( \text{path}_k = g_{p_k}g_{k_1}g_{k_2} \ldots g_{kn}, \ 1 \leq k \leq n \), the operation of cutting all the child nodes of these paths is denoted with \( \Theta_{\{\text{path}_1,\ldots,\text{path}_n\}}(g_p) \), there is

\[
g = \Theta_{\{\text{path}_1,\ldots,\text{path}_n\}}(g_p) = (r, V, E)
\]

where \( r=g \),
\[
V = \{ v_i | \forall v_i \in G, \ 1 \leq k \leq n, v_i \in (V_1 - r_1 - \text{sub}(\text{end}(\text{path}_3))) \}, \quad E = \{ e_i | \forall v_j \in G, \ 1 \leq k \leq n, v_j \in \text{sub}(\text{end}(\text{path}_3)), \ \text{then} e_i \in (E_1 - <\text{end}(\text{path}_3), v_i> - E_v) \}.
\]

When integrating heterogeneous data of massive amount, the GDM object cutting operation is always used for gathering a part of the data or cutting a portion of the data off to improve the efficiency.

**Example of Data Integration and Query**

In addition to the table *Student* in Example 1, suppose there are another two data source tables *Course* and *Grade*, which record the information of courses and student grades respectively. The structures of the tables and their relevant GDM models are shown in Fig.4.

<table>
<thead>
<tr>
<th>CID</th>
<th>Name</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Maths</td>
<td>3</td>
</tr>
<tr>
<td>102</td>
<td>Physics</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig.4(a) table *Course*

<table>
<thead>
<tr>
<th>SID</th>
<th>CID</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>101</td>
<td>83</td>
</tr>
<tr>
<td>001</td>
<td>102</td>
<td>94</td>
</tr>
<tr>
<td>002</td>
<td>101</td>
<td>89</td>
</tr>
</tbody>
</table>

Fig.4(c) table *Grade*

**Fig.4(b) the GDM model of *Course***

**Fig.4(d) the GDM model of *Grade***

Figure 4. Tables *Course* and *Grade* and their relevant GDM models.
Example 3. Query all the selected course names and grades of student No. 001. The query process can be divided into the following three steps:

1. $G' = \sigma_{\text{Grade.SID}=\text{"001"}}(\text{Grade})$
2. $G' = G | x | G'/\text{grade}/\text{CID}, \text{Course}/\text{course}/\text{Name}, \text{Grade}/\text{CID}=\text{Course}/\text{ID}$
3. $\text{Result} = \Pi_{G'}/\text{grade}/\text{CID}/\text{Name}, G' / \text{grade}/\text{Score}$

Where grade and course are the i-th record respectively in tables Grade and Course.

The above example shows that GDM model is capable of implementing complex querying from different data sources, which is of great significance for heterogeneous data integration.

**Conclusions**

By comparing existing data models, this paper puts forward a new uniform data model, i.e. the GDM model. The model expresses various data intuitively and flexibly on the basis of concepts, such as Paragraph, Section, GriData, GriDoc, et al. This paper then defines basic algebra operations of GDM objects, which provides essential theory support for GDM object manipulations. A comprehensive example of integrating and querying GDM object data is provided, showing that GDM model has favorable ability for heterogeneous data integration and operation.

Further researches on issues such as schema translation, heterogeneous data semantic integration, data query and system framework of the GDM model will be discussed in our subsequent works.

**Acknowledgements**

Many thanks are given to those who offer assistance to complete the research work, and special gratitude are to the reviewers and editors for their diligent work and relevant suggestions.

This work is also supported by the National Natural Science Foundation of China (Grant Nos. 61070015 and 61402183), Guangdong Natural Science Foundation (Grant Nos. 10351806001000000, S201203006242 and S20130400124), Guangdong Provincial Scientific and Technological Projects (Grant Nos. 2014B090901028, 2016A010101007, 2016B090918021, 2014B010117001, 2014A010103022 and 2014A010103008), Guangdong frontier and key technical innovation projects (Grant No. 2014B010110004), Guangzhou major special research collaborative innovation projects (Grant No. 2060404) and Guangdong research cooperation projects (503036363071).

**References**

[7] Junyan Z. Research of Information Source Description Model Based on Heterogeneous Data Exchange [D]: South China Normal University; 2009.