Using Nested Tables for Representing and Querying Semistructured Web Data*

Irna M.R. Evangelista Filha[1], Altigran S. da Silva[1,†], Alberto H. F. Laender[1], and David W. Embley[2]

[1]Department of Computer Science
Federal University of Minas Gerais
31270-901 Belo Horizonte MG Brazil
{imre,alti,laender}@dcc.ufmg.br

[2]Department of Computer Science
Brigham Young University
Provo, Utah 84602 USA
embley@cs.byu.edu

Abstract. This paper proposes an approach for representing and querying semistructured Web data, which is based on nested tables allowing internal nested structural variations. Our motivation is to reduce the complexity found in typical query languages for semistructured data and to provide users with an alternative for quickly querying data obtained from multiple-record Web pages. We show the feasibility of our proposal by developing a prototype for a graphical query interface called QSByE (*Querying Semistructured data By Example*), which implements a set of QBE-like operations that extends typical nested-relational-algebra operations to handle semistructured data.

1 Introduction

In recent years, the demand for technologies to manage data available on the Web has increased considerably. Motivated by such a demand, several researchers have proposed adapting database techniques to efficiently and flexibly deal with the particularities of semistructured Web data. Unfortunately, the inherent freedom common in semistructured data (e.g., XML) inhibits the deployment of metaphors and paradigms like the ones that have been used extensively in typical data management tasks.

We propose in this paper the use of nested tables for representing and querying semistructured Web data. We show that nested tables can be nicely adapted for this kind of data and can provide a simple and intuitive representation close to record-based database representations, since they naturally accommodate hierarchical data. The distinction, and main contribution, of our proposal is that it also allows the representation of variations and irregularities typical of semistructured data.

We also introduce a set of query operations for semistructured data represented as an internally varying nested table and describe QSByE (*Querying*

*This work was partially supported by Project SIAM (MCT/CNPq/PRONEX grant number 76.97.1016.00) and by CNPq (grant number 467775/00-1). The first and second authors are supported by scholarships from CAPES. The fourth author is supported by NSF (grant number IIS-0083127).

†On leave from the University of Amazonas, Brazil.

Semistructured data By Example) [3], a query interface that implements such operations. QSByE combines features of QBE (*Query By Example*) [6] with typical features of query languages for semistructured data, such as type coercion and path expressions [1]. Particularly, QSByE makes it possible to handle data extracted from Web pages by a tool called DEByE (*Data Extraction By Example*) [4].

2 Nested Tables with Structural Variants

In this section we propose a generalization of regular nested tables in which we allow a column to have two or more distinct substructures called *variants*.

As an example, consider the nested table illustrated in Figure 1, which contains data extracted from an Amazon.com Web page that shows products related to Paul McCartney available in several "stores". To simplify our example, we consider only the stores *Popular Music*, *Books*, and *Auctions*. Observe that the information about products from each store is different. The information consists of Title, Artist, and AudioType for popular music; Title, Authors, and BookType for books; and Item, Bid, and Time for auctions. As a consequence, the internal structures of the data in the column Info are distinct for each of the rows.

Store	Info
Popular Music	Title: *Wingspan (Hits & History)*; Artist: *Paul McCartney*; AudioType: *Audio CD*
Books	Title: *Blackbird Singing ...*; Authors: *Paul McCartney, Adrian Mitchell*; BookType: *HardCover*
Auctions	Item: *PAUL MCCARTNEY At ...*; Bid: *$ 26.99*; Time: *1 days, 20:15:57*

Fig. 1. Example of a Nested Table Allowing Structural Variants.

Despite the relative simplicity of dealing with semistructured data in the form of nested tables, it is easy to see that such a representation is not as expressive as general semistructured data models or XML. However, in this work we are mainly concerned with representing data extracted from a specific type of Web page that is said to be a *data rich*, *ontological narrow*, *multiple-record* Web page. Examples of such pages are found in Web sites such as bookstores, electronic catalogs, travel agencies, and classified ads and include pages composed of data whose overall structure is naturally hierarchical, but exhibits a modest degree of variation [4].

3 Query Operations and the QSByE Interface

In this section we present a set of operations for querying semistructured data represented as nested tables, which extends the operations proposed by Colby [2]. We also illustrate how these operations work in the QSByE interface. These operations, selection, projection, nest and unnest, are briefly described in what follows.

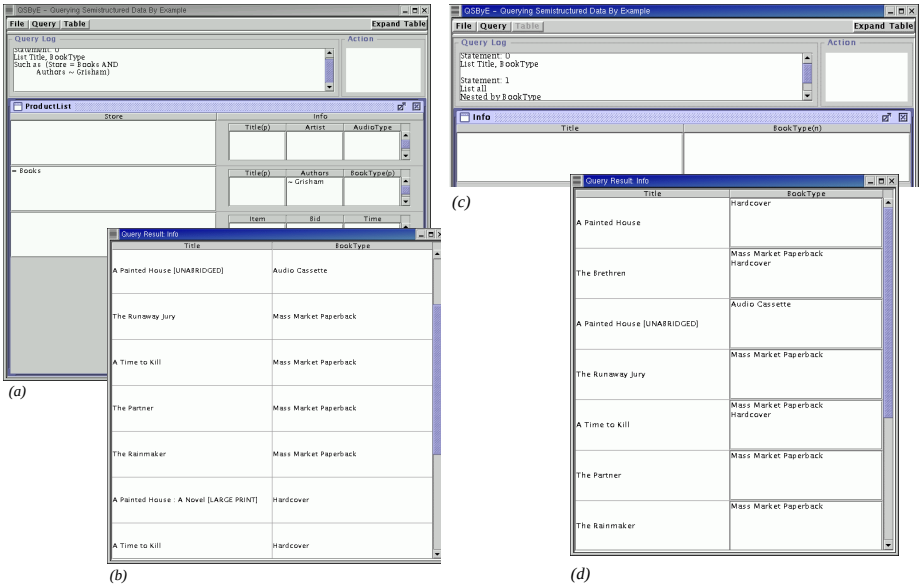

Fig. 2. Specification and Result of the Example Query.

The ***selection*** operation allows selection conditions to be specified at any level of nesting. A selection condition is a boolean expression defined over the value in a column or over the existence of a column in an internal subtable (i.e., a *structural condition*). The ***projection*** operation is similar to the projection operation defined in the relational algebra. We can say that this operation horizontally reduces a table by keeping only the columns specified by the user. Similar to what was proposed by Thomas and Fischer [5], the ***nest*** operation has two distinct semantics. When applied to a single column, this operation groups the values in the column that are associated with equal values occurring in other columns. Otherwise, it creates a new internal table that groups the values of these columns and, consequently, generates a new level of nesting in the table. The ***unnest*** operation is the inverse of the nest operation. It also has two distinct semantics, according to [5], and it is only valid for columns containing list of atoms or an internal table. If it is applied to a list of atoms, it will "ungroup" its elements, i.e., it will split them into different rows of the table. Otherwise, if it is applied to an internal table it will eliminate a level of nesting.

These operations were implemented in QSByE (*Querying Semistructured data By Example*) [3], a query interface that adopts a variation of the QBE paradigm [6] to provide a suitable environment for casual users to query semistructured Web data. We illustrate how queries are formulated with QSByE by means of an example of a query over the data in the nested table of Figure 1.

Suppose a user is interested in issuing the following query: *List the title and the type of the books written by John Grisham. Rearrange the result by nesting the values of the column BookType.* To formulate this query, the user first selects a repository (resultant from a data extraction using DEByE) containing the

data of interest. This immediately provides the structure of the stored data in the form of a nested skeleton, as illustrated in Figure 2(a). Then, the user specifies the selection conditions on the columns Store and Authors to retrieve only books written by John Grisham. This is shown in Figure 2(a). Notice that the condition on Authors has been specified using the $\sim$ operator that denotes approximate comparison by pattern matching. To show only the title and type of the books, the user specifies a projection operation by clicking on the header of the corresponding columns. This is also shown in Figure 2(a). Figure 2(b) shows an excerpt of the query result.

To satisfy the query specification, it is necessary to rearrange the result table of Figure 2(b), by nesting the values of the column BookType. This is accomplished as follows. By clicking on the "Query" button (Figure 2(a)), the user makes QSByE to replace the current skeleton table by the skeleton of the result table of Figure 2(b), as shown in Figure 2(c). Then, the user clicks on the "Table" button, selects the nest operation and clicks on the column BookType. QSByE then marks the header of this column with an "(n)" (see Figure 2(c)). The table resulting from this operation is shown in Figure 2(d).

4 Conclusions

We have presented a generalization of nested tables for semistructured data in which distinct rows are allowed to contain data with distinct structures. We have shown that these nested tables can be used to represent and query semistructured data extracted from the Web. We extended nested relational algebra operations to manage the variable nestings we defined, and we implemented QSByE to allow a user to issue queries in a QBE-like manner. Preliminary experimentations with QSByE have demonstrated that with a small amount of training even unskilled users can use the interface to query semistructured data extracted from Web pages.

References

[1] Abiteboul, S., Quass, D., McHugh, J., Widom, J., and Wiener, J. The Lorel Query Language for Semistructured Data. *International Journal on Digital Libraries 1*, 1 (1997), 68–88.

[2] Colby, L. S. A Recursive Algebra and Query Optimization for Nested Relations. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data* (Portland, Oregon, 1989), pp. 273–283.

[3] Evangelista-Filha, I. M. R., Laender, A. H. F., and Silva, A. S. Querying Semistructured Data By Example: The QSByE Interface. In *Proceedings of the International Workshop on Information Integration on the Web* (Rio de Janeiro, Brazil, 2001), pp. 156–163.

[4] Laender, A. H. F., Ribeiro-Neto, B., and da Silva., A. S. DEByE – Data Extraction by Bxample. *Data and Knowledge Engineering 40*, 2 (2002), 121–154.

[5] Thomas, S. J., and Fischer, P. C. Nested Relational Structures. *Advances in Computing Research 3* (1986), 269–307.

[6] Zloof, M. M. Query-by-Example: A Data Base Language. *IBM Systems Journal 16*, 4 (1977), 324–343.