ELSEVIER

Data & Knowledge Engineering 39 (2001) 105–142

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

Schemas for web data: a reverse engineering approach

Sourav S. Bhowmick [a,*], Wee Keong Ng [a], Sanjay Madria [b]

[a] *Center for Advances Information in Systems, School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore*

[b] *Department of Computer Science, University of Missouri-Rolla, Rolla 65409, USA*

Received 24 July 2001; received in revised form 24 July 2001; accepted 24 July 2001

Abstract

In this paper, we show how to generate schemas of a set of HTML or XML documents retrieved from the web in the context of our web warehousing system called WHOWEDA (*W*are*H*ouse *O*f *WE*b *DA*ta). *Web schemas* are used to bind a *web table* that contains a collection of interlinked web documents called *web tuples*. These schemas specify the metadata, content and structural properties (in the form of *predicates*) shared by the web documents and hyperlinks in the web table. They also summarize the hyperlink structure of these documents using the notion of *connectivities*. Web schemas are generated in three stages. In the first stage, a *simple* or *complex* web schema is generated from the user's query (*coupling query*). In the next stage, the *complex* web schema is decomposed into a set of *simple* web schemas. These two stages are performed without inspecting the data instances, i.e., web tuples. Finally, in the last stage the set of *simple* web schemas are *pruned* by inspecting the hyperlink structure of the web tuples. We also discuss the formal algorithm for generating a set of simple web schemas from a coupling query.

Keywords: Web schemas; Web table; Web warehouse; Coupling query; Web tuples

1. Introduction

The exponential growth of the web in the last few years had a significant impact on the traditional techniques used for data management during the last few decades. This has compelled the database community to reuse traditional techniques wherever possible to manage web data. Unfortunately, due to the very nature of web data, it is not always possible to reuse conventional techniques effectively. This has led the database community to rethink and reuse existing techniques in a new way to address the current challenges. In this paper, we describe a novel technique for generating schemas of web data. We introduce the notion of *web schema* to model instances of warehouse data and show how it is generated in the context of our web warehousing system, called WHOWEDA (*W*are*h*ouse *O*f *We*b *Da*ta) [3]. As will be seen, this issue is more challenging than the corresponding problem for relational schema due to the irregularity and incompleteness

* Corresponding author.
E-mail addresses: assourav@ntu.edu.sg (S.S. Bhowmick), awkng@ntu.edu.sg (W.K. Ng), madrias@umr.edu (S. Madria).

0169-023X/01/$ - see front matter
PII: S0169-023X(01)00036-2

of data in the world wide web. Beyond its use to define the structure of a set of data in the warehouse, a web schema serves two important functions. It helps user in query formulation and aids the query processor for efficient execution of query [4].

There has been increasing research activities in generating schemas for semistructured data [2,6,7,9,10]. For instance, in [7,8], the authors provide a structural summary that allows a semistructured database system (or a user of one) to quickly extract information about label paths in the database. In [10] a work on the extraction of implicit structure in semistructured data modeled in the style of [1] as directed, labeled graph is presented. Our approach differs from these works in the following ways.

1.1. Content, metadata and structural summary

Traditionally, a schema provides a structural summary of the data it binds. Query formulation and evaluation can be performed efficiently if some of the content and metadata properties shared by the web documents and hyperlinks are highlighted in the schema. HTML tags are not used for describing the data segment enclosed in it and hence structural summary of HTML pages is not very useful in subsequent query evaluation and formulation. Furthermore, capturing summary of the hyperlink structure of a set of web documents also helps us to formulate meaningful queries in the warehouse. Consequently, a web schema provides two types of information: first, it specifies some of the common properties shared by the documents and hyperlinks in the *web table* with respect to not only their structure, but also their metadata and content. Second, a web schema(s) summarizes the hyperlink structure of these documents. For instance, given a set of documents, the web schema(s) may specify that the title of all these documents contain the keyword "genetic disorder". It may also specify that these documents belong to the web site at `www.ninds.nih.gov` and contain the tags `symptom`, `treatment` and `drugs` inside the tag `disease`. Also, a web schema may specify that a set of documents containing the keyword "genetic disorder" are directly linked to a set of documents having the tags `drugs` and `side effects` via a set of hyperlinks whose label contain the keyword "drugs". In Section 3.1, we describe how the content, metadata and structural summary is incorporated in a web schema.

1.2. Reverse approach for schema generation

We take a *reverse* approach in generating a web schema. The standard database paradigm in schema generation involves first creating a schema to describe the structure of the database and then populating that database through the interface provided by the schema. A schema is then used to decide whether some new data fits the schema or whether a query is legal against the set of data. Hence, a schema is defined before the query. In our approach, a web schema is defined from a *coupling query*. A coupling query is specified by a user and is used to populate the web warehouse by retrieving relevant data from the web that matches the query. The results of such query is a set of directed graphs called *web tuples* and are stored in a *web table*. We justify this reverse approach now. If a web schema is defined by a user ahead of time, the structure, content and rigidity of a web schema depends on the following factors: first, the information a user wishes to retrieve from the web. Second, the user's level of knowledge of the content and structure of the web site(s) containing the relevant data. However, this conventional approach is not feasible because of the following reasons: first, it is unrealistic to assume from the user complete