

An Application for WebDAV-based Authoring of Databases - WebDAD

Bitu Shadgar
University of Bristol, U.K
Department of Computer Science
Merchant Venturers Building a
0044 117 9545252
B.Shadgar@bristol.ac.uk

Ian Holyer
University of Bristol, U.K
Department of Computer Science
Merchant Venturers Building a
0044 117 9545148
Ian.Holyer@bristol.ac.uk

ABSTRACT

We propose a new approach to accessing databases through the Web that is based on the WebDAV protocol (Web-based Distributed Authoring and Versioning). This paper introduces an extensible framework for WebDAV-based authoring of databases, whether relational or object-oriented, called WebDAD. WebDAD is a mapping of SQL database operations onto an HTTP URL namespace, and WebDAV protocol operations. This is the first peer-reviewed presentation on mapping SQL to WebDAV, and provides an insight into the similarities and differences between the WebDAV and SQL database approaches for managing data. The WebDAV protocol has been chosen as a foundation for WebDAD, since it is the first protocol for distributed authoring. Also, WebDAV advantages such as metadata and access control make it a better candidate of being the layout protocol for database transactions rather than HTTP.

Keywords

WebDAV, HTTP, Databases, SQL.

1. INTRODUCTION

Today almost all well known Web-based database applications are based on the HTTP protocol [1], and especially the Post method. Although HTTP is a very strong protocol for browsing and searching distributed and heterogeneous information, it is not strong enough to support the authoring and managing of that information over the Internet [4]. In fact, the idea of authoring databases based on the WebDAV protocol is built on two reasons. Firstly, despite Post being a very helpful method to transfer all kinds of database transaction from Web clients to Web servers, it is not always a safe method. In particular the Post method tunnels every kind of SQL query into the Web servers, without concerning the query's type whether it is a retrieval or a storing SQL query. Since WebDAV defines more methods regarding the authoring of the Web resources, it lets us represent the operations more accurately [3]. In other words, WebDAV methods provide the ability of expressing SQL queries in the form of a sequence of methods as it is discussed in WebDAD. Hence, the Web server and intermediaries are aware of the nature of each SQL query whether it is safe or not. Secondly, WebDAV advantages such as metadata is well suited to present database metadata. Regarding the fact that there is no standard language in order to extract the database metadata and it is vendor dependent, WebDAV is a better candidate of being the layout protocol for database transactions rather than HTTP. Indeed, the WebDAV access control presents controls on databases via WebDAV permissions, and improves that access control in some databases. In the following, we extend a WebDAV support application to support the database data model. Indeed, a primitive subset of SQL statements, which can be shown to be relatively complete in the context of the WebDAD framework is specified. Each SQL statement is translated based on the taxonomy of the WebDAV methods. Although WebDAV is the best layout for authoring, however WebDAV methods are mostly intended to support file system resources and as an aside, some shortages are observed to support other Web resources such as databases.

The next section presents the WebDAD structure and section three offers concluding remarks and related work.

2. WEBDAD FRAMEWORK

In this section, we present the fundamental principles of our proposed authoring framework. The WebDAD data model considers each record as a separate resource, and each table as a collection within WebDAV [10]. In an relational database, we propose to consider each record as a separate resource, and each table as a collection within WebDAV. Hence, they are accessible using the following URI (Uniform Resource Identifier) [2,8].

```
http://host[:port]/rdb/schema/table
http://host[:port]/rdb/schema/table/record
```

Although, all SQL statements of SQL99 can be represented in form of WebDAV methods. We select a primitive subset of SQL statements which covers the most important SQL statements such as the Create Table, Alter Table, Delete, Drop Table, Insert, Select, Update, Grant and Revoke statements [7]. In each connection between client and server WebDAD sends an SQL statement to the database along with the corresponding WebDAV methods to the WebDAV repository. Table1 lists each SQL query with its relative WebDAV methods.

WebDAV as an extension of HTTP is a stateless protocol. However, it is necessary for WebDAD to have a mechanism of some kind to treat the sequence of WebDAV methods relevant to each SQL statement as one single transaction. Therefore, we propose to make use of the Batch method. It still keeps the protocol stateless. Although the Batch method has been refused as a new WebDAV method [5,6], but since WebDAD uses Batch method to represent a given sequence of WebDAV methods as one transaction regarding to each class of SQL statement, intermediaries are able to inspect the methods of a Batch request in order to determine the semantics of a request, and thereby its associated access control and/or authentication requirements. In case of any error, the Batch method will report the error message related to the most recent method which fails.

SQL Statement	Relative WebDAV Methods
Create Table	Mkcol, PropPach
Alter Table	Mkcol PropPach
Drop Table	Delete (collection)
Select	Search, Put
Insert	[Select ,] Put
Update	Select (where clause) [, Select (set clause)]
Delete	[Select ,] Delete (resource)
Grant	Acl
Revoke	Acl

Table 1. SQL statements and relative WebDAV methods.

At the moment of presenting this paper it is hard or even impossible to replace SQL queries with only the WebDAV methods, since the WebDAV methods are not able to express all aspects of some statements. For example, current WebDAV methods can not express all aspects of the Insert statement, since there is no way of checking data types by WebDAV methods. However, some statements like Create Table can be expressed completely by the WebDAV methods i.e. Mkcol and PropPach methods. Therefore in our implementation, SQL statements are temporary embedded in Batch methods. When the Batch method is sent to the server, WebDAD knows exactly which statements have to be processed by the database. Figure1 shows an example of a Batch method in order to create a table.

```

BATCH /rdb/schema HTTP/1.1
Host: host
Content: text/xml; charset= "utf-8"
Content-Length: xxxx

<?xml version= "1.0" encoding= "utf-8" ?>
<batch oneTransaction= "true" isDB= "true"
    DBStatement= "Create Table SP ( S# NOT NULL varchar
        P# NOT NULL varchar(3), QTY NOT NULL :
        PRIMARY KEY (S#, P#),
        FOREIGN KEY (S#) REFERENCES S,
        FOREIGN KEY (P#) REFERENCES P)"
    statementClass= "CREATE_TABLE">
<request>
    <header> The header of Mkol(SP) </header>
    <body> The body part of Mkol(SP) </body>
</request>
<request>
    <header>
        The header of PropPatch(SP, hashtable-of-properties
    </header>
    <body>
        The body part of PropPatch(SP, hashtable-of-propert
    </body>
</request>
</batch>

```

Figure 1. A sample of Batch method to create a table.

In addition, the contents of the request tags declare what methods are necessary to handle the metadata and access control features. These two kinds of processing are done simultaneously and during the process the database and the WebDAV repository will exchange their different parts of data or metadata to accomplish the user request in the best way (Figure2).

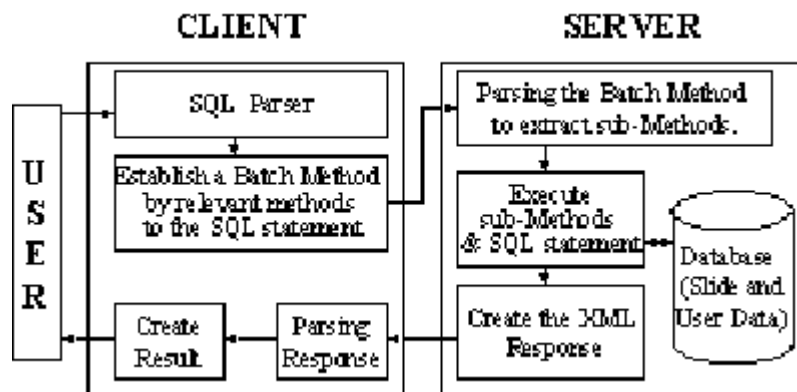


Figure 2. The architecture of WebDAD application.

3. CONCLUSION

The objective of this research work is to highlight and bring to the attention of the WebDAV group that file systems are not the only WebDAV resource to be considered. But also WebDAV is a proper layout protocol for authoring of other Web resources, such as databases, with some consideration. Although there are a lot of applications to support authoring of file systems based on WebDAV [9], WebDAD is the first application intended to provide a WebDAV-based support for authoring databases.

Eventually, we would expect all database manipulation to be carried out using standard request methods.

4. ACKNOWLEDGEMENTS

The first author is on the scholarship funded by the Iranian Ministry of Science, Research and Technology. The authors also thank the WebDAV and Jakarta Slide working group for their support.

5. REFERENCES

1. Marty Hall, 2000, Core Servlets and Java Server Pages, Prentice-Hall.
2. R. Elmasri and S.B. Navathe, 2000, Fundamentals of Database Systems, 3rd ed., Addison Wesley.
3. Y. Goland, E. Whitehead, A. Faizi, S.R. Carter and D. Jensen, 1999, RFC 2518, HTTP Extensions for Distributed Authoring - WEBDAV.
4. R. Fielding, J. Gettys, J. Mogul and T. Berners-Lee, 1997, RFC 2068, Hypertext Transfer Protocol.
5. Interest in standardizing Batch methods?, <http://lists.w3.org/Archives/Public/w3c-dist-auth/2002JanMar/000-1.html>.
6. Proposal: WebDAV and Transactions, <http://lists.w3-.org/Archives/Public/w3c-dist-auth/2002JulSep/0172.-html>
7. Kline, K. with Kline, D., 2001, SQL In a Nutshell, O'Reilly, US.
8. T. Berners-Lee, R. Fielding and L. Masinter, 1998, RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, <http://www.ietf.org/rfc/rfc2396>.
9. WebDAV Projects and Software, <http://www.webdav.org/projects>.
10. B. Shadgar, I. Holyer, WebDAV-based Distributed Authoring of Databases (WebDAD), Austrian Computer Society, Advances in Information and Communication Technology, page 529.