Model of Storage of ERP Protocols in EEG/ERP Portal

Vaclav Papez, Roman Moucek
Department of Computer Science and Engineering
University of West Bohemia
Pilsen, Czech Republic

Abstract— **The EEG/ERP (Electroencephalography/Even-Related Potentials) Portal serves for storage and management of ERP experiments. The protocols describing these experiments are designed and implemented using various software applications and/or hardware devices. Some protocols are kept in XML files and stored in the EEG/ERP portal database as LOB (Large Object) data types. Since it is important for researchers to directly find details about design of the ERP protocols in the related XML files and it is also hard to use LOB data for generating the semantic web output, storage types provided by the Oracle 11g database system were investigated and compared. Therefore a new model of storage of ERP protocols is introduced. This model is finally simplified after analysis of results of performance tests.**

Keywords – electroencephalography; event related potentials; neuroinformatics; EEG/ERP portal; EEG/ERP experiments; experimental protocol; Oracle; relational model; XML Type; binary XMLType; structured XMLType; unstructured XMLType; performance test; semantic web

I. INTRODUCTION

The Electroencephalography / Even-Related Potentials (EEG/ERP) Portal is one of the main projects of the neuroinformatics research group at the Department of Computer Science and Engineering, University of West Bohemia. Activities of the group include development and integration of software tools and hardware devices for EEG/ERP research, proposals and modifications of signal processing methods, and design and implementation of EEG/ERP experiments. The EEG/ERP portal has been developed to store and manage a huge amount of data and metadata related to EEG/ERP experiments. Currently the portal provides the following features [6]:

- Management of EEG/ERP data / metadata
- Management of EEG/ERP experimental protocols
- Management of data related to tested subjects
- Sharing of knowledge and work in groups
- Signal processing methods
- Content management system
- Full text search

Design of ERP experiments is quite a challenging task, which should follow some general principles described e.g. in [8]. Experimental protocols are designed and implemented using various software applications and/or hardware devices. Some protocols are finally described in XML (Extensible Markup Language) file(s) and interpreted by related applications. The advantage of the XML format is its openness, transparency and especially the opportunity to query it for details of the protocol. Since these details are very important for the interpretation of the EEG/ERP experiment, it is worth querying them quickly and easily.

The EEG/ERP portal stores XML files, which describe ERP experiments, in the relational database Oracle 11g. Since this database system introduced new solutions for storing XML files, our aim was to find the best way to store and query this type of file in the database and finally improve the database performance. Then the goal of this paper is to present, analyze and compare traditional and innovative solutions for the storage of ERP protocols as XML files in the Oracle 11g relational database and provide recommendations, which would lead to faster and more comfortable processing of these protocols. We hope that the results will be beneficial not only for community experimenters and developers, but also for people interested in effective storage and processing of XML files.

The paper is organized as follows. Section 2 introduces various formats of ERP protocols. Alternatives for effective storage of ERP protocols, comparison of traditional methods for storing data in XML with the new ones using XML Type and model of storage of ERP protocols in the EEG/ERP portal are presented in Section 3. The fourth section describes design and results of performance tests. Conclusions, recommendations, and the final model of storage of ERP protocols are presented in Section 5.

II. FORMATS OF ERP PROTOCOLS

ERP experimental protocols are very variable and they can be designed and implemented using various software applications and/or hardware devices. The EEG/ERP portal works with protocols of any type, format and length. In our laboratory many protocols are implemented as short XML files, usually with a well-known structure.

978-1-4673-1184-7/12/$31.00 ©2012 IEEE

Authorized licensed use limited to: Universidad del Valle. Downloaded on October 17,2024 at 10:59:54 UTC from IEEE Xplore. Restrictions apply.

In general, we can divide ERP protocols according to their implementation into two following groups:

- XML protocols in the standard XML file format
  - with schema – the structure of the file is known in advance; it is described by XML Schema Definition (XSD) or Document Type Definition (DTD) languages
  - without schema – the structure of the file is undefined
- Non – XML protocols in text or binary file(s) formats

Because non-xml protocols have an unpredictable structure, they are simply stored in the database as character or binary large objects (CLOB, and BLOB types) and are not further discussed.

III. Storage of XML protocols

Since it is important for researchers to find details about design of the ERP protocol in the related XML file, these detailed pieces of information should be easily accessible in the database. To achieve this, it is necessary to use a traditional technique, it means to parse the ERP experiment and store it to relational tables, or to use innovative techniques provided by the Oracle database system. This section describes these techniques and their combination. Finally, a suitable approach for the storage of ERP protocols is proposed.

A. Relational Model

The schema of each frequently repetitive protocol can be transformed into relational tables. The application logic has to validate the XML file, parse the data and insert them into tables. Then it is possible to query the protocol by SQL (Structured Query Language). When the user needs to create the original XML protocol again, he/she has to select the proper data from the database and build the XML file via the application logic (for example JDBC – Java Database Connectivity and StAX – Streaming API for XML for JAVA can be used). Storing a new schema requires modification of the database structure. The following example shows a simple ERP protocol (Fig. 1) and its representation by the relational data model that describes the schema (Fig. 2).

B. XML Type Storage

XMLType is an extended data type first introduced in Oracle 9i. It can be used in three different ways depending on the structure of the related XML document. There are three types of XML Type storages – the structured, unstructured and binary storage. Analogously to large objects types (CLOB, BLOB) only one XML Type column per table is permitted.

1) Structured XML Type Storage

The structured XML Type (also called Object Relational Type) is designed as the storage of frequently repeated XML data centric documents with the same schema. The XSD schema is required for creating the table; it must be specified in the related script. Schemas are stored separately (the storage is

```
<?xml version="1.0" encoding="windows-1250"
standalone="yes"?>

<imagelist count="250" random="yes" txt="output.txt"
delay="700" report="single" pause="300">

        <img src="p300_img\1a.jpg" count="50"
mark="one"/>
        <img src="p300_img\1b.jpg" count="50"
mark="one"/>

        <img src="p300_img\8.jpg" count="50"
mark="eight"/>
        <img src="p300_img\1c.jpg" count="50"
mark="one"/>
        <img src="p300_img\1.jpg" count="50"
mark="one"/>

</imagelist>
```

Figure 1. ERP Protocol in XML

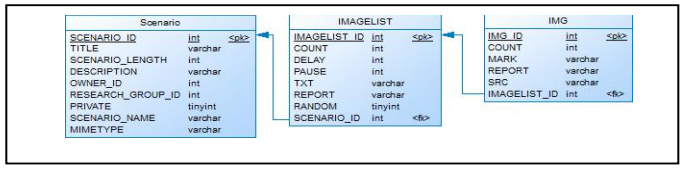

Figure 2. Relational model of XML schema

provided by Oracle) and have to be registered. The registration is not trivial and is worth using an application for this step, for example OracleSchemaRegister [4]. The registered schema is parsed and new types and nested tables are created. These objects represent schema for internal purposes (mapping of the XML document to the nested tables) of the Oracle database system and allow users to select data from the stored XML document while maintaining high performance. Each query to the XML file consists of the standard SQL part that selects the requested XML file, and the XPath part that extracts requested data from the selected XML file. Nested tables are created entirely for an internal purpose, querying them is impossible.

In the database just one table per schema is required (three tables for three schemas are presented in Fig. 3 - SCENARIO_TYPE_SCHEMA_1 to SCENARIO_TYPE_ SCHEMA_3).

During registering the schema a lot of objects (types and internal tables) are created and then used in the database system. Each of these objects has its own object identifier (OID); the schema file has its own OID as well. An unpleasant side effect of this solution is non-trivial migration of the database; standard migration usually crashes because of OIDs conflict (Oracle 11g is not able to generate a new OID for schemas in the repository during migration of the database).

The structured storage is recommended to store frequently repeated protocols of the same type (schema).

2) Unstructured XML Type Storage

The unstructured XML Type is most similar to the traditional CLOB type (it is also called XML Type CLOB). It is enriched by the possibility of querying the XML document by XPath. The schema is not required to store the XML document. This storage is very simple but it has the worst performance (starting Oracle 11g and XMLIndex performance

Authorized licensed use limited to: Universidad del Valle. Downloaded on October 17,2024 at 10:59:54 UTC from IEEE Xplore. Restrictions apply.

is higher, but still worse than in the case of using other storages). Moreover, we have to know the structure of the selected protocol during querying explicitly because various XML documents can be stored in one column of the table. This storage is useful when the user does not have or cannot create a schema. This storage is recommended by Oracle for document centric XML files.

3) Binary XML Type Storage

The binary XML Type is a new storage introduced in Oracle 11g. The XML protocol is stored in a special post-parsed binary format. Moreover, the user can have various protocols with different schemas in one column, but the protocols can be still validated. If the schema is present, querying of the XML document is faster than in the case of the unstructured storage. This storage is the best solution in the case we have many protocols with different schemas, or we do not have a lot of protocols derived by the same schema or variability of schemas quickly grows. Oracle recommends this storage for document centric XMLs.

4) Comparison of XML Type Storages

The following table shows a comparison of the storages presented above.

TABLE I. COMPARISON OF XML TYPES STORAGES (S – STRUCTURED; U – UNSTRUCTURED; B – BINARY; Y – YES; N – NO; L – LIMITED)

Task	*S*	*U*	*B*
More than one XML type per table permitted	N	N	N
Schema required before table is created	Y	N	N
Pre-parsed schema	Y	N	N
XML Type data is post-parsed	N	N	Y
XPath query	Y	Y	Y
More than one schema per table permitted	N	N	Y
Universality	N	L	Y
Suitability for one XML schema	Y	N	Y
Suitability for multiple XML schema	N	N	Y
Suitability for non-schema XMLs	N	Y	Y

C. Model of storage in EEG/ERP portal

Each previously mentioned solution for storing the XML document is not generally applicable to a random set of ERP protocols. The best database performance is achieved by using a combination of more storage types.

Any ERP protocol regardless of its format has the same set of metadata. These metadata are stored in the relational table Scenario (Fig. 3) in the EEG/ERP portal [1]. Non-XML protocols are stored in the table using the standard BLOB type. The table for XML protocols with unknown or undefined schema uses the binary storage. Finally, the structured storage is used for the rest of protocols. The following diagram shows a final data model that was proposed in [2].

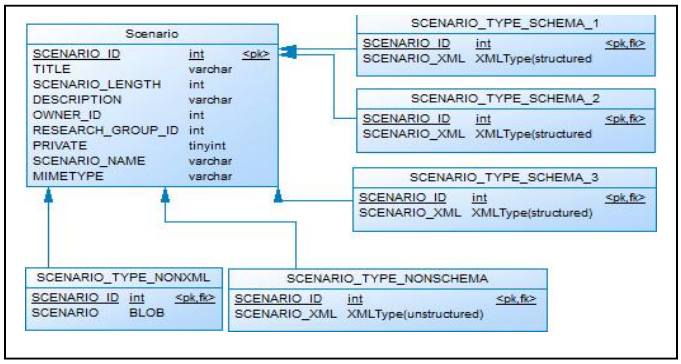

Figure 3. Model of storage of ERP protocols in the EEG/ERP portal [2]

The implementation of this model provides the best overall performance of this part of the database but not the best sustainability. Let suppose the following use case.

A new experiment according to a new protocol is performed. The experimenter wants to upload data to the EEG/ERP portal. At first, he/she discovers that his/her schema of the protocol is missing so he/she needs to register a new one. This step requires changes in the structure of the database (a new table for a new schema is created). It is impossible to delegate it to users; this step has to be done by the administrator. The experimenter has to wait until the new schema is registered. He/she has two options:

- send only the schema to the database and after its registration upload data and metadata of the experiment
- insert all data and metadata along with the schema of the experiment in the database; data and metadata will be accessible after schema registration

Therefore, the user prefers to upload a single XML file without schema because he/she does not need to wait. It is possible to avoid these problems by using a binary storage only because decline in performance is acceptable (see Section 4).

IV. PERFORMANCE TESTS

Short performance tests were realized under the following assumptions. There were thousands to tens of thousands stored protocols. Both protocol files (ten elements nested to the second level) and queries were short and simple. With these assumptions three queries were created for each storage type (a simple query, a query with the JOIN clause and the string comparison and a query with the JOIN clause and the integer comparison).

The protocols were generated using random data; they were equivalent to the previously mentioned example (Fig. 1). Execution of queries was realized via SQLDeveloper. Each query was executed four times and the average execution time was calculated. No indexes were used. The following figures show average execution times in seconds (vertical axis) for above mentioned queries, depending on the number of protocols (horizontal axis). Blue lines represent the structured

Authorized licensed use limited to: Universidad del Valle. Downloaded on October 17,2024 at 10:59:54 UTC from IEEE Xplore. Restrictions apply.

storage; red lines represent the binary storage and green lines represent the data in relational tables (XML Type is not used).

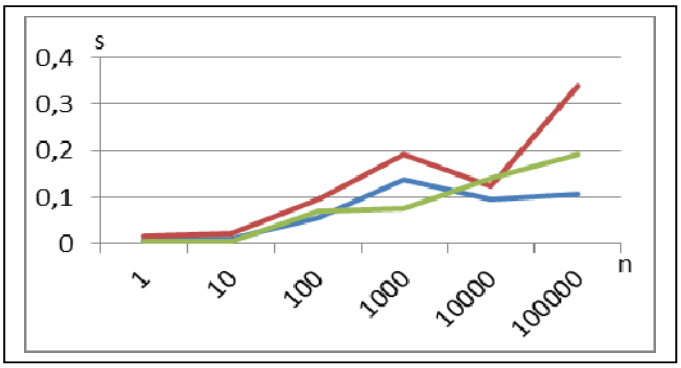

Figure 4. Results of the performance test for a simple query (number of protocols – horizontal axis, average execution time – vertical axis)

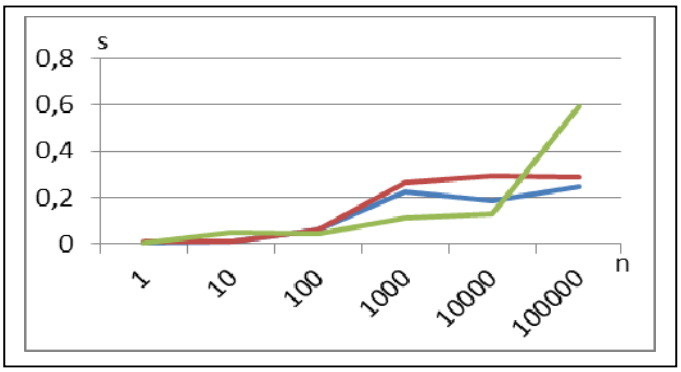

Figure 5. Results of the performance test for a query containing JOIN and the string comparison (number of protocols – horizontal axis, average execution time – vertical axis)

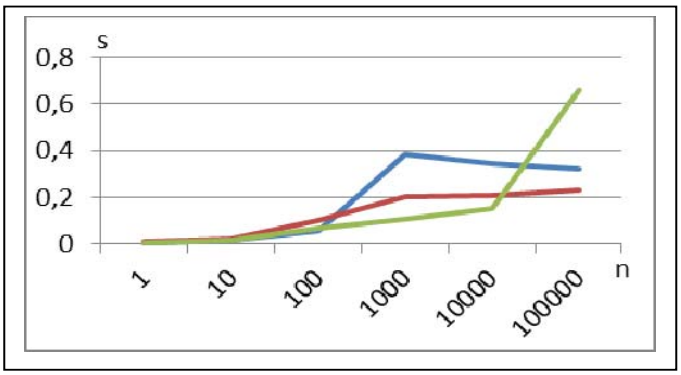

Figure 6. Results of the performance test for a query containing contains JOIN and the integer comparison (number of protocols – horizontal axis, average execution time – vertical axis)

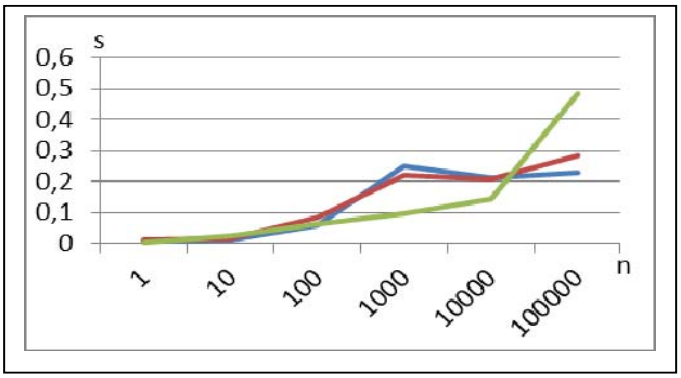

Figure 7. Results of the performance test across all queries (number of protocols – horizontal axis, average execution time – vertical axis)

The part of the database using the structured storage has the best performance as it was expected. However, differences between the structured and binary storage is not more than 10 %.

Up to approximately one hundred protocols the performance of all three storages is similar. For hundreds and thousands of protocols the execution time for storages using the XML Type grows more rapidly than for the relational schema. For more than ten thousand protocols the execution time grows very fast in the case of using the relational schema. Moreover, the execution time slightly decreases for the range of thousand to ten thousand protocols. It means that implicit optimization is better in the case of using the XML Type.

V. CONCLUSION AND RECOMMENTATION

Current protocol storage was evaluated and new solutions with Oracle`s XML type technology were proposed. The performance tests were done with surprising results - the binary XML storage had nearly the same performance as the structured type and both were better than relational storage. Considering the performance tests and advantages and disadvantages of described storages, a new schema for storing ERP experiments was proposed. This proposal reduces the schema to a very simple form; the changes do not affect the performance of the related part of the database. The table for metadata is the same, non-XML protocols are stored using the BLOB type and the binary storage is used for storing XML protocols. Both XML files with and without the schema can be stored there. Fig. 8 shows the final model of storage of ERP protocols in the EEG/ERP portal.

The future work focuses on migration problems and in particular on investigation if the XML storage can be suitably used to the automatic transformation of the relational database to the Resource Description Framework (RDF) [7]. A new idea of obtaining a semantic web description (in RDF and OWL languages) from the schema (XSD) arose in the course of solving this problem.

Authorized licensed use limited to: Universidad del Valle. Downloaded on October 17,2024 at 10:59:54 UTC from IEEE Xplore. Restrictions apply.

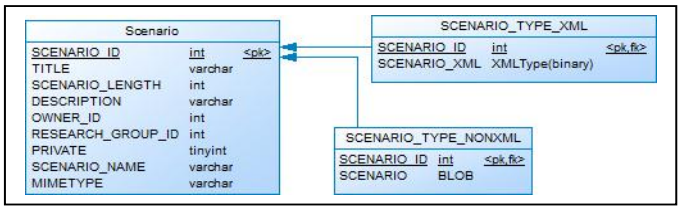

Figure 8. Final model of storage of ERP protocols in the EEG/ERP portal

ACKNOWLEDGMENT

The work was supported by the UWB grant SGS-2010-038 Methods and Applications of Bio- and Medical Informatics.

REFERENCES

[1] P. Jezek and R. Moucek, "Database of EEG/ERP experiments" in International Conference on Health Informatics, vol. 3 Valencia, pp. 222-227, 2010

[2] J. Koren, "Using XML in Oracle database", unpublished

[3] J. Koren, "EEG/ERP Portal – Methods for storing structured data", Bachelor thesis, Pilsen, 2011

[4] M. Kral, "Experimentalni lekarsky informacni system - zpracovani heterogennich nestrukturovanych dat", Diploma thesis, Pilsen, 2010

[5] R. Moucek, P. Jezek and V. Papez, "Semantic web technologies in EEG/ERP domain - software solution" in Internation Conference on Health Informatics, vol. 4 Rome, pp. 618-621, 2011

[6] R. Moucek, P. Jaros, P. Jezek and V. Papez, "Software infrastructure for EEG/ERP research" in Conference on Knowledge Engineering and Ontology Development, Paris, 2011

[7] V. Papez, "Neuroinformatic database and semantic web", Diploma thesis, Pilsen, 2010

[8] S. J. Luck, "An Introduction to the Event-Related Potential technique", Cambridge, MIT press, 2005

[9] Oralce, Oracle XML Developer's guide, retrieved May 2011 from http://docs.oracle.com/cd/B28359_01/appdev.111/b28369/xdb03usg.htm

Authorized licensed use limited to: Universidad del Valle. Downloaded on October 17,2024 at 10:59:54 UTC from IEEE Xplore. Restrictions apply.