

An XML Framework for Grid-Enabled Decision Support Systems

Matti Heikkurinen¹, Tapio Niemi²,
Vesa Sivunen¹ and Petra Sohlman³

Matti.Heikkurinen@cern.ch, tapio@cs.uta.fi,
Vesa.Sivunen@cern.ch, Petra.Sohlman@valtiokonttori.fi

¹ Helsinki Institute of Physics, CERN Offices, CH-1211 Geneva, Switzerland

² Department of Computer and Information Sciences,
FIN-33014 University of Tampere, Finland

³ State Treasury, Insurance Administration, Sörnäisten rantatie 13,
P.O. Box 14, FIN-00531 Helsinki, Finland

Abstract

The aim of this paper is to present a framework for decision support systems. The framework combines components from Grid computing and XML to OLAP technologies in a novel way. Our hypothesis is that with these components one can gain additional benefits from the traditional industrial management optimisation methods by making data gathering faster, including data across organisational boundaries and getting more computing power for analysis. The goal is to help the decision maker to formalise and answer to problem at hand by using a collection of heterogeneous, geographically distributed data sources with the aid of OLAP methods.

1 Introduction

The corporate environment is increasingly dependent on the speed and accuracy of inter-organisational decision making. The necessary data sources may include operational databases of the company, data files in the intranet, data in the public Internet - but also real time streaming data, or data sets orders of magnitude greater than supported by the database systems today. Due to the increasingly complex global supplier network structures, ownership and control of these sources may not be limited to a single enterprise. Coinciding with this increased complexity, Internet based B2B systems and Internet banking applications reduce the inherent latency of the marketplace transactions. The ability to flexibly use aggregation of data sources for analysis and knowledge discovery is becoming one of the key areas of interest in corporate computing systems. OLAP (on-line analytical processing) is one of the most promising methods in analysis of business data, supported by major database system vendors. However, deployment of OLAP systems has been slow due to the deficiencies in design methods, need for manual work in data collection and integration, complicated query methods, and excessive computing power requirements. To alleviate the shortfalls of the design methods, we will propose new design principles for OLAP cubes. Applying new design methods and novel Grid tools, our aim is to automate a major part of the data collection and integrations tasks and address performance issues in cube construction and query processing. The framework proposed is based on well known components and it will offer a generic platform, with standard interfaces and high level security due to the use of XML and Grid technologies.

Reliable and up-to-date data and information, together with the necessary knowledge needed to interpret it, are probably the most important resources in any kind of decision making. On the technical level, solutions based on various data management components applying statistical or other knowledge discovery methods on the data offer some partial solutions to the data management and refinement problems. However, design criteria of these systems are not well understood, nor is the way and extent they provide actual value in the economic sense. Research in this area is thus necessarily a combination of computer science, business economics, and statistics.

In many processes lots of data is stored into databases or other data storage systems for later retrieval and analysis. The storing and retrieving data have been the main issues in the database research, with less effort dedicated to analysing large amounts of data - at least until recently. Gathering as much information as possible from volumes of data is a central problem in many application areas. In many scientific analysis tasks,

data should be combined from different locations and different kinds of data storages, either due to the various inherent limitations of database systems used, or due to the geographical spread of the organisation or organisations producing the data. Working on these kinds of logically connected data sets requires lots of computing power. In addition, data transfer is one of the critical components of the systems. Distributing data processing close to data storage is necessarily needed to reduce the network traffic, since it is impossible to transfer terabytes of data efficiently. It is when trying to address these challenges, where existing methodologies and commercial software solutions tend to fail due to the limitations of the software used or the available hardware infrastructure. For example, in digital format a mammogram requires storing of 100-200MB of data. US National Digital Mammogram archive aims at eventually storing yearly up to 28 petabytes of these images and associated metadata from some 2000 medical facilities. Analysing a distributed database of this size clearly exceeds the capacities of database solutions available nowadays or in the near future. (see: http://www.infoworld.com/article/02/10/18/021104fegrid_1.html.)

The aim of this article is to address the limitations of the current solutions by using XML, Grid security and computing, and OLAP (on-line analytical processing) together in a novel way. In the numerous Grid efforts the ease and flexibility of sharing data between scientific researchers has proven to be one of the central requirements in the support of distributed collaborations. Solutions developed to address this issue are, for example, Grid-enabled database interfaces, replica management systems, and various virtual organisation support systems augmenting the basic security solutions. At the same time, Grid computing in itself offers an vision of ubiquitous, on demand access to cheap computing resources, effectively bypassing the limitations imposed by the current hardware infrastructure.

The research on the economics of the Grid has so far concentrated on issues related to the exchange of these computing resources, where the central problem has so far been devising a way to assign a price for the various uses of computing resources. However, using the Grid technologies as a supporting infrastructure and OLAP methods as analysis tools in situations where there already exists some method for performing monetary transactions bypasses the problem of the general Grid economics. The central hypothesis is that one could gain additional benefits from the traditional industrial management optimisation methods by:

1. Making the data gathering phase faster by using the Grid components to include data directly from the sources, which do not necessarily conform to any database schema.

2. Including data from several databases, possibly owned by different organisations and implemented using tools from different vendors.
3. Using OLAP technologies to give more intuitive view to the operational data in the organisation under study.
4. Using Grid technologies to harness more computing power for generating and analysing various scenarios developed based on the OLAP analysis.

Due to the limitations of the current business infrastructure (network bandwidth, storage and computing capacities), it is unlikely that in the near future the commercial OLAP systems will be developed to the point where the combined benefits of these approaches can be realised. Moreover, it seems that the amount of stored data is increasing more rapidly than the capacity of the systems. Due to the constraints imposed by the markets in the foreseeable future, these kinds of initiatives related to the handling of large amounts of data will fall into the realm of basic research in the computer science. Our focus will be in establishing an experimental and theoretical feedback loop between the methods and components of computer science and the needs of business economics.

2 Technology and terminology

2.1 OLAP Background

On-Line Analytical Processing (OLAP) has gained popularity as a method to support decision making in situations where raw data on measures, such as sales or profit, needs to be analysed at different levels of statistical aggregation. In OLAP, queries are made against structures called OLAP cubes.

The OLAP cube is a multidimensional database in which the dimension attributes (i.e. co-ordinates) determine the measure values. For example, the dimensions are time, location, product, and the measures are sales and profit. The dimensions usually have a hierarchical structure, which enables the user to analyse the data in different levels of details. For example, the analysis can be performed at day level or at month level. In the latter case, the monthly data is aggregated from daily data.

The design of the OLAP cube is based on knowledge of the application area and the types of queries that the users are expected to pose. In optimal case, a well-designed OLAP cube can offer the user some of the flexibility of a pure database solution, with the possibility of posing infinitely complex

queries, combined with the user support of the application specific knowledge encoded into the system that is the strength of dedicated expert systems.

The contents of OLAP databases are typically collected from other data repositories, such as operational databases. For a well-defined and targeted system, where the information needs are well known, it may be straightforward to collect the right data at the right time. However, there is more and more data generally available, and also the information needs develop. Consequently, it gets more and more difficult to anticipate the needs of OLAP users. This leads to a situation where it is increasingly difficult to know in advance what data is required - and when - for the desired analysis tasks.

As commercial use of OLAP technology gains in sophistication, OLAP cube design techniques will become more important because of the increased frequency with which an enterprise will require the redesign its OLAP cubes. There are a number of reasons for this phenomenon. In practice, companies may often change their organisation. Thus, realignment of data warehouse schema is needed quite frequently. Especially required dimensions of data and their hierarchies can change regularly, actually even more frequently than the real-life organisation [11]. The users may also want to speculate about the effects of, for example, changing the way their company has arranged its organisation. Furthermore, needs to analyse data in ways which were not anticipated at the time when the OLAP cubes were designed may emerge as a result of this testing of scenarios.

Finally, in a geographically distributed corporate environment, the need for up-to-date data means that the OLAP cube needs to be constructed almost in real time [1]. Geographical distribution adds challenges related to the management of the user rights, fault tolerance of the computation over the wide area networks and limited amount of available bandwidth.

2.2 Grid

Our design applies Grid technologies. Foster and Kesselman describe the Grid as follows: “The Grid is a software infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources.” [4] An essential components of our framework is the Grid Security Infrastructure (GSI), which allows secure connections to potentially all computers in the Grid. GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol [9]. The design uses HTTP/HTTPS protocol to access remote databases. Access to common http and https ports (80, 443) is normally allowed through possible firewalls. The user authentication is based on a common certificate, thus separate user IDs or passwords are not needed.

The best known GSI implementation is a part of Globus software. Globus (<http://www.globus.org>) is a de-facto standard for distributed computing platforms. Globus incorporates a large body of secure data transfer and remote program execution tools.

2.2.1 Spitfire and TrustManager

Spitfire is one of the European Data Grid Projects (EDG) [10]. It offers a Java servlet that accepts database request and displays the results in XML. TrustManager is EDG's authentication software that can be used in connection with Spitfire. The TrustManager analyses the users rights to execute queries based on his user (Grid) certificate.

2.2.2 Mobile Analyzer

The computing facilities and remote processing of the system will be implemented using the Mobile Analyzer technology [6]. Mobile Analyzer is Java software that easily enables the execution of Java code in remote computers. The basic idea of Mobile Analyzer is that the user provides Java classes that are executed remotely. The system has facilities to retrieve results and status information back from the computing servers. Like Spitfire, Mobile Analyzer uses a certificate based user authorisation system. The architecture of Mobile Analyzer is shown in Figure 1. The Proxy server serves the requests of clients and dispatches jobs to proper analysis servers. Analysis servers are computing power and data storage for the jobs. The Mobile Analyzer framework enables us to easily perform the data processing where the data is stored or to distribute computing in order increase performance via parallel processing.

2.2.3 VOMS

The EDG has introduced two concepts called Virtual Organisation (VO) and Resource Providers (RP). This has been done mainly because it is not practical to look after authentication or authorisation information for everyone at every site. The VO is an abstract entity grouping users, institutions and Resources in a single administrative domain. The RP facility offers resources to other parties. Virtual Organisation Membership Service (VOMS) creates a certificate, which proves the identity and that the entity (like a user) has a membership and certain other attributes of a VO [5].

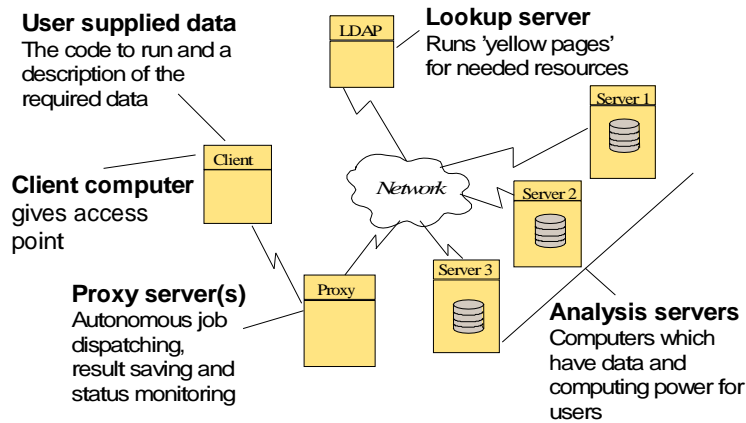


Figure 1: Mobile Analyzer

2.3 XML

XML (eXtensible Markup Language) is “the universal format for structured documents and data on the Web” [3]. As such, XML is a meta language – a language for describing other languages – which lets one design customised markup languages for different types of documents, using a declaration syntax defined in recommendation [2].

When the amount of the XML data increases and the use of relational databases is definitely not decreasing, the need to process database requests involving both kinds of data is becoming more and more important. This denotes a need for systems that are able to handle heterogeneous databases. Our aim is to provide a common front end (middleware) for both XML and relational data [8].

3 System Architecture

The goal is to develop OLAP methods and components to help the user in the following task: A decision-maker has a problem at hand to be formalised and answered using a collection of heterogeneous, geographically distributed data sources with the aid of OLAP methods.

Our aim is not to develop a new OLAP server but only a collection of methods and tools that can be used when constructing an OLAP cube into an OLAP server. However, these tools must be capable to do something very similar to OLAP servers: data must be selected and aggregated quite often before uploading it into a server. The implementation of a very basic

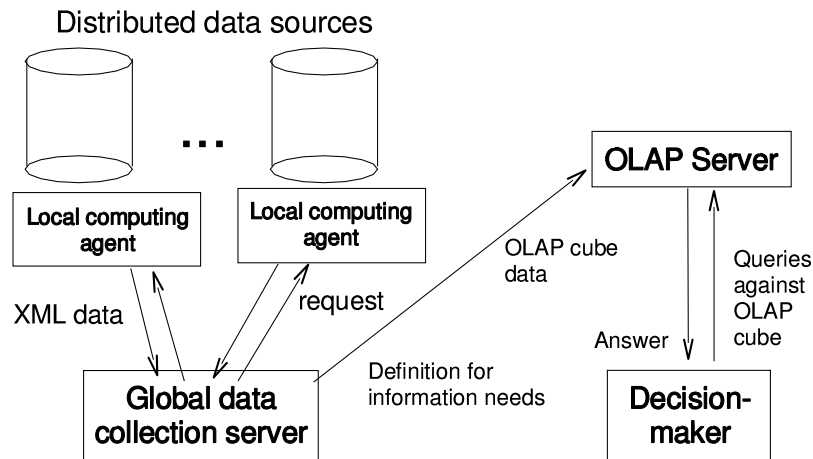


Figure 2: The General Architecture of the System

prototype of this kind of system is described [7]. In general, we aim to solve the problem by studying and developing a framework illustrated in Figure 2. The research can be divided into the sub tasks described in the following paragraphs.

3.1 Cube Construction from Heterogeneous Data Sources

The current practice in OLAP is to construct an OLAP cube that contains all information stored in the data warehouse of a company. Sometimes, however, the data volumes can be so large that it is very hard and expensive to prepare an OLAP cube for a potentially wide selection of OLAP queries in advance. It appears that collecting the right data on demand might be a better alternative in many applications. This way the data is also up-to-date, as it is collected when it is needed. To speed up the cube construction process, distributed computing will be applied. Grid techniques will be applied to secure issues, computing, and data access.

A further problem may well be that the data repositories or data warehouses involved in data collection are often heterogeneous, yet their information should be integrated in the OLAP database. XML appears to be a suitable solution for this problem. All kind of data is easy to represent using XML and it is quite straightforward to transform an XML language to another XML language. Our idea is to use an XML language to present OLAP data. An example of the OLAP cube data represented by an XML language is shown in Figure 3. We use another XML document to define to

which database and server the required data is stored. Other XML languages used in different databases can be transformed to this common presentation by using XSLT transformations. XML is also readable by most of the OLAP servers.

```
<olap_cube name="trade">
  <fact_table>
    <fact_row value="200" product="fine paper"
export_country="Finland"
import_country="UK" year="1988"/>
    <fact_row value="256" product="stainless steel"
export_country="Finland"
import_country="USA" year="1989"/>
  </fact_table>
  <product>
    <product_row product_name="fine paper"
sub_group="paper" main_group="forest"/>
    <product_row product_name="stainless steel"
sub_group="steel" main_group="metal"/>
  </product>
  .
  .
</olap_cube>
```

Figure 3: A part of the example OLAP cube in XML

3.2 Defining Users' Information Needs

It is not straightforward for the user to define what data is needed in the analysis. The problem at hand can be from an exact question (e.g. What was the profit in May 2002?) to a very general one (e.g. How to improve productivity?). Our aim is to develop an user friendly XML-based query language for defining the data needed in the analysis and also for querying OLAP data. A simple version of this query method is shown in Figure 4.

3.3 Design Methods for OLAP Cubes

There are various different ways, in which the users may want to arrange their OLAP cube, depending on the nature of the queries they intend to pose. Different designs vary significantly in terms of efficiency and practicality. For example, for a given set of raw data, one design may produce an extremely sparse cube, in the sense that many of the data items are missing or zero because of the nature of the data. The sparsity itself can be handled using

```

<query_definition>
  <selection_constraints>
    <constraint name="year" value="1980,1990"/>
    <constraint name="import_country.continent" value="3,5"/>
    <constraint name="product.main_group" value="0"/>
  </selection_constraints>
  <roll_up_operations>
    <operation name="import_country.continent"/>
    <operation name="product.main_group"/>
  </roll_up_operations>
</query_definition>

```

Figure 4: A sample XML query for OLAP cube construction

storage structures optimised for sparse data but sparsity together storing precalculated data can cause the need of the storage space to increase very rapidly. Thus, the sparsity problem must be handled on the level of logical design. Some other designs can be problematic for query formulation as they may, for instance, produce incorrect aggregations.

Real time systems and data streams set special requirements for design methods. Real time data analysis is becoming more important in future. Privacy protection is also an important issue in data analysis.

We will meet these challenges by developing design methods and components to ensure a good cube structure and to take care of disclosiveness problem.

4 Conclusions

We have described a framework in which we plan to produce new methods and components for OLAP design and OLAP-based data analysis. These include a new design principles for multidimensional databases, and distributed cube construction and query processing for OLAP cubes. The framework is largely built on the basis of the commonly used Grid and XML components. We believe that our system will provide experimental and theoretical knowledge about a novel combination of methods and processes for increasing the effectiveness of business intelligence systems.

References

- [1] M. Akinde, M. Bhlen, T. Johnson, Lakshmanan L., and Srivastava D. Efficient OLAP query processing in distributed data warehouses. *Information Systems*, 28, 2003.
- [2] The World Wide Web Consortium. Extensible markup language (xml) 1.0 (second edition), W3C recommendation 6 October. Available on: <http://www.w3.org/TR/2000/REC-xml-20001006>, 2000.
- [3] The World Wide Web Consortium. Extensible markup language (XML). Available on: <http://www.w3.org/XML/>, 2002.
- [4] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3), 2001.
- [5] Security Coordination Group. Datagrid security design. Technical report, European DataGrid Project, 2003.
- [6] J. Karppinen, T. Niemi, and M. Niinimäki. Mobile analyzer - new concept for next generation of distributed computing. In *The 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, (CCGrid 2003)*, 2003. A poster.
- [7] T. Niemi, M. Niinimäki, J. Nummenmaa, and P. Thanisch. Applying Grid technologies to XML based OLAP cube construction. Technical report, CERN Open Preprint series, 2002. Available on: <http://doc.cern.ch/archive/electronic/cern/preprints/open/open-2003-004.pdf>.
- [8] T. Niemi, M. Niinimäki, and V. Sivunen. Integrating distributed heterogeneous databases and distributed grid computing. In *Proceedings of the 5th International Conference on Enterprise Information Systems*, volume 1, pages 96–103. ICEIS Press, 2003.
- [9] The Globus project. Commodity Grid Kits. Available on: <http://www-unix.globus.org/cog/>, 2002.
- [10] Project Spitfire. Project Spitfire. Available on: <http://spitfire.web.cern.ch>, 2001.
- [11] T. Zurek and M. Sinnwell. Data warehousing has more colours than just black & white. In *Proceedings of the 25th international conference of very large data bases*. Morgan Kaufmann, 1999.