Towards OLAP Security Design – Survey and Research Issues

Torsten Priebe, Günther Pernul *
Department of Information Systems
University of Essen
Universitätsstr. 9, 45141 Essen
Germany
{priebe,pernul}@wi-inf.uni-essen.de

ABSTRACT

With the use of data warehousing and online analytical processing (OLAP) for decision support applications new security issues arise. The goal of this paper is to introduce an OLAP security design methodology, pointing out fields that require further research work. We present possible access control requirements categorized by their complexity. OLAP security mechanisms and their implementations in commercial systems are presented and checked for their suitability to address the requirements.

Traditionally data warehouses were queried by high level users (executive management, business analysts) only. As the range of potential users with data warehouse access is steadily growing, this assumption is no longer appropriate and the necessity of proper access control mechanisms arises. However, a data warehouse is primarily built as an open system. Especially exploratory OLAP analysis requires this open nature; security controls may hinder the analytical discovery process.

Keywords

Data warehouse, OLAP, security, access control, design.

1. INTRODUCTION

The relevance of data warehouses and online analytical processing (OLAP) for an organization's decision support system has rapidly grown over the past few years. At the same time a quite good sensitivity for information security and privacy has evolved. However, not many approaches have been made to bring these two fields together. Data warehouses by their very nature create a security conflict [8]. On the one hand, the goal is to make all necessary data accessible as easy as possible. On the other hand, this data is usually very valuable and sensitive. The concept of security is very broad (covering ethical and social issues, moral issues, privacy and legal issues as well). In this paper we focus mainly on the technical issues laying an accent on authorization and access control.

We explore these security issues in the context of the GOAL project which aims at studying the integration of geographical information systems (GIS) and data warehouse technology. In this project we are responsible for data integrity and security. Two pilot applications (A1 and A2) have been defined for the project. Application A1 is concerned about visitor admissions to the castles and other monuments in Southern Bohemia, Czech Republic. Application A2 deals with the drinking water distribution and consumption in the Western Bohemian region close to the city Sokolov, CZ. Throughout this paper we use application A1 as a case study. It turns out to be quite suitable for discussing security questions. The castle management is organized in a hierarchical way. All but two of the castles are managed by a regional institute, but still keep a certain degree of independence. They are only willing to provide their data for integration in a central data warehouse if they trust the installed security measures. Different user groups with only restricted access to the data warehouse have been defined. However, the developed concepts should be applicable for most other OLAP applications as well.

The rest of this paper is organized as follows: the remainder of section 1 presents general data warehouse security issues and identifies OLAP access control as our core topic. The necessity of OLAP security design is motivated and a methodology is introduced. We present possible access control requirements categorized by their complexity in section 2. Section 3 deals with OLAP security mechanisms and their problems on a vendor independent conceptual level. In section 4 we present and compare the implementations in different commercial systems. Finally, section 5 concludes the paper and sketches the path of our future research work. Details on the data model of GOAL application A1 (our scenario) can be found in the appendix.

1.1 General Security Considerations

Obviously a lot of communication takes place in a data warehouse system, creating the need for proper communication security measures. The data load process (transferring the source data from operational databases to the data warehouse) defines new re-

* This work is supported in part by the European Union through INCO-Copernicus grant no. 977091 (project GOAL – Geographic Information Online Analysis).

quirements for a network infrastructure. Independent (possibly distributed) source databases have to be consolidated over a network. As the data may be highly sensitive it is essential to protect it from eavesdropping and similar secrecy threats. For the communication between the front-end applications and the OLAP server (or the data warehouse in 2-tier environments) usually a client/server connection will be utilized, possibly to remote sites. Even though the information on this channel is most likely aggregated and less complete, it may be highly security critical. The use of the Internet or other possibly insecure networks for the above mentioned connections makes suitable security measures necessary. As only some tools support encrypted communication on application level, virtual private network (VPN) technology might be appropriate.

Authentication and audit are other security measures that have to be installed in a data warehouse environment. A clear proof of a user's identity is needed in order to apply appropriate security restrictions and to avoid access by unauthorized users. Corresponding user identification and authentication mechanisms check the authenticity of the pretended identity, either inside the front-end tools, or by making use of authentication mechanisms provided by modern operating systems or tools allowing a "single sign-on" strategy. The essential decision for auditing in data warehouses is to put it in the right place in the architecture. Using only the auditing capabilities of an underlying (relational) DBMS holding the data warehouse will not satisfy the needs, as logging the access to tables of the star/snowflake schema (or similar objects) will not expose the actual multidimensional queries performed (especially in MOLAP based systems). The conclusion is that auditing should also be performed on the multidimensional level of an OLAP engine (i.e. at the same level where authorization semantics are defined).

1.2 Access Control

Access control on the back-end side involves controlling the access to the data warehouse and the source databases by the extract/transform/load processes and the access to these procedures (invoking as well as administration). In [5] a role-based authorization model for the administrative processes in a data warehouse is presented identifying two categories of roles. *Developers* write the extraction, integration and transformation scripts. They need access primarily to metadata, not to the data itself. *Operations personnel* invoke the corresponding processes. They do not need permissions to access the data directly, just to run the trusted programs. However, when problems arise, developers and operations personnel might need additional access to some data, e.g. to decide about data cleaning strategies or to fix failures. One might grant additional permissions, if it is made sure that the use of such permissions is extensively monitored by auditing.

On the front-end side more novel access control issues arise. Traditionally data warehouses were queried by high level users (executive management, business analysts) only, which was used as an excuse for OLAP vendors not to provide support for fine grained access control. This assumption is, however, no longer appropriate. The range of potential users of analysis tools querying a data warehouse is steadily growing, up to customers and partners, or castle visitors in GOAL. Protecting the sensitive data from unauthorized access more and more becomes an issue, which leads to the necessity of proper access control policies for end-user access to the data warehouse. Not every user should be able to access all data.

Front-end applications include static reporting (running and creating/modifying reports), OLAP, and data mining/KDD. In *static reporting* applications, where users only use predefined static queries, access control can be defined on a per report basis. On the other hand it is very difficult to apply security to *data mining*. Data mining is aimed at finding new patterns in the data; the result (and thus its sensitivity) is not known beforehand. However, some policies (such as partitioning the data warehouse) can be implemented using a data mart-like technology. The major front-end tools for data warehouses are *OLAP* applications, providing interactive ad-hoc analysis of multidimensionally structured data. A data warehouse is primarily built as an open system. The goal is to make all necessary data accessible as easy as possible. Especially exploratory OLAP analysis requires this open nature; security controls may hinder the analytical discovery process.

We have identified communication security, user identification and authentication, auditing, and access control as important security issues. As we focus on access control in (ad-hoc) OLAP applications we will use the term OLAP security in this narrow meaning throughout the rest of this paper.

1.3 OLAP Security Design

Deriving the access control policies from the operational data sources is very difficult although some research efforts are made in this area [16]. Data from different systems (with different policies) will be consolidated. Also, the users of the operational systems are not the same as the users of the data warehouse. However, the main problem is, that the relational model is predominate in operational systems while OLAP systems make use of the non-traditional multidimensional model. Access control schemes do not map easily. Protection is not defined in terms of tables, but dimensions, hierarchical paths, granularity levels. The need for proper OLAP security design arises.

We already mentioned that the design of OLAP access restrictions has to be performed with care, as they might hinder the analysis or even produce wrong results. Additionally the tools' security capabilities are highly proprietary and the syntax of their security constraints is not feasible for design and documentation of the access restrictions. In order to approach the topic from the application side, the classical database design methodology (requirement analysis, conceptual, logical, and physical design) should be applied to OLAP security as well. Figure 1 is adapted from [1] who suggest a similar model for regular database security. The important difference, however, is the multidimensional conceptual data model and the OLAP security mechanisms that significantly differ from the capabilities of relational database management systems. The phases marked bold in the diagram will be (to some degree) covered in this paper, the design phases themselves are subject to our future work (see also our conclusion in section 5).

A multiphase methodological approach allows security policies to be separated from security mechanisms. This separation yields advantages such as [1]:

- The capability of defining access control rules and reasoning about them independently of their implementation (with no burdens about implementation details)

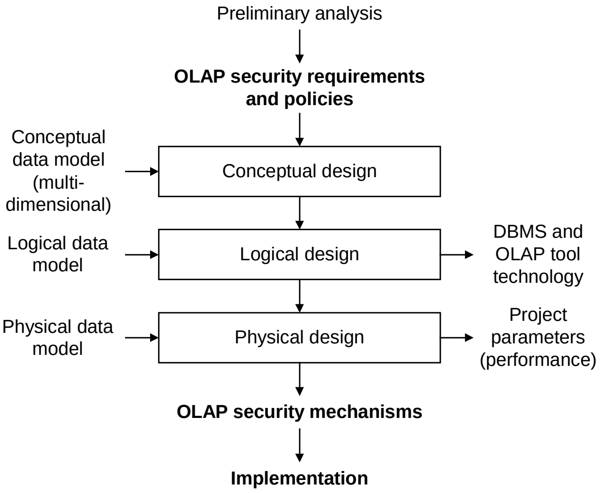
Preliminary analysis
OLAP security requirements and policies
Conceptual data model (multi-dimensional)
Conceptual design
Logical data model
Logical design
DBMS and OLAP tool technology
Physical data model
Physical design
Project parameters (performance)
OLAP security mechanisms
Implementation

Figure 1. OLAP security design methodology (adapted from [1])

- The possibility to compare different access control policies, or different mechanisms for the same policy. This is especially useful in the heterogeneous world of OLAP tools.
- The capability of designing mechanisms supporting different policies. This advantage becomes a strong need when policies change as a consequence of changes in the organization.

2. OLAP SECURITY REQUIREMENTS AND POLICIES

Different applications lead to very different requirements (i.e. possible policies) for OLAP access control. The result of the requirements analysis is high-level guidelines that can be translated in a rule format, suitable for formalization and subsequent design phases [1]. However, a proper foundation for multidimensional conceptual security modeling is not available. In this paper we thus use plain English to describe possible requirements. These requirements will be formalized in future by defining more formalized a security constraint language (SCL).

Figure 2 defines basic and advanced access control requirements[1] which will be discussed in the following giving examples for possible access rules based on user groups that have been identified in GOAL [10] (for details on the data model see the appendix).

2.1 Basic Requirements

Hiding whole cubes is a very straightforward requirement. In the context of GOAL a corresponding access rule might specify that certain users can only view data from the *Services* cube and not from the *People* cube.

If a certain dimension mirrors the structure of the users (e.g. object managers exist for each object represented in the *Geography* dimension), it might be necessary to hide certain slices of a cube. Hiding certain measures similar to hiding slices. In fact, there is no conceptual difference if the set of measures is interpreted as a flat dimension. An example from GOAL would be a user group who can just see the *Count* measure of the *Services* cube.

Detail data is often considered more sensitive than summarized data. It thus might be necessary to restrict access to data below a particular dimension detail level (i.e. hiding levels of detail). This can range from only the leaf level to hiding an entire dimension (in which case the user will only be able to query aggregates over all members of that dimension). Consider as an example an access control policy where data may not be accessed on finer granularity than *Month* on the *Time* dimension.

2.2 Advanced Requirements

If detail data is considered particularly sensitive as mentioned above, and additionally users are responsible for certain members of a dimension (i.e. cube slices), this leads to policies where it is necessary to hide levels of detail in certain slices of a cube. Consider the following access control policy for object managers: "An object manager can get any information concerning his object, only restricted data for other objects."

When designing such policies it makes a substantial difference whether the dimension on which the slice is defined and the dimension whose detail levels are to be hidden are identical or not (see section 3). For example, if the *Tour* detail level will be hidden for specific objects, only one dimension is involved. As an example for hiding levels of detail in certain slices of a different dimension, assume an object manager that is allowed to view daily data for his own object, but only monthly data for the others.

In dynamic or data driven policies access rules are not defined on certain structure elements of the multidimensional data model. Access permissions depend on the data itself (i.e. dimension member properties or fact measures). An example is a user group that can access objects with more than 5,000 visitors per month only (say, these objects are considered to be of public interest). As this can change with every data load, we call the policy dynamic.

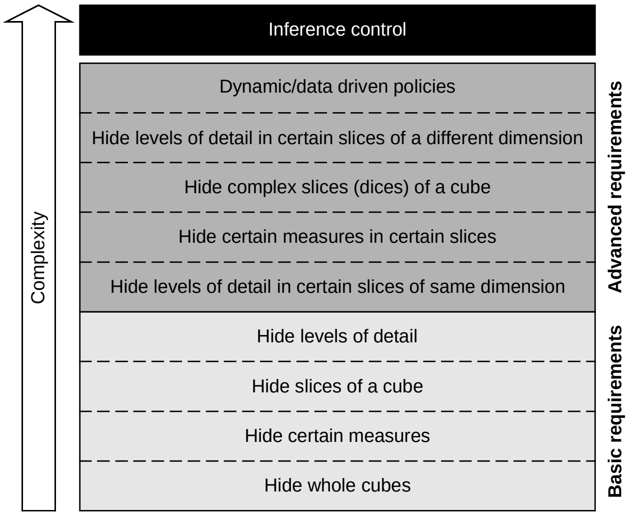
Inference control
Dynamic/data driven policies
Hide levels of detail in certain slices of a different dimension
Hide complex slices (dices) of a cube
Hide certain measures in certain slices
Hide levels of detail in certain slices of same dimension
Hide levels of detail
Hide slices of a cube
Hide certain measures
Hide whole cubes
Complexity
Advanced requirements
Basic requirements

Figure 2. Different OLAP access control requirements

[1] The wording in the diagram ("Hide ...") implies the explicit definition of prohibitions rather than permissions. However, the same concepts are applicable for positive authorization.

2.3 Inference Control

OLAP systems are particularly subject to inference problems as they rely on summary or aggregate data. Due to the access on such aggregate data, smart (so called tracker) queries might reveal data that is not accessible directly.

Information inference has already been identified in statistical database security research. The traditional problem was to protect data on individuals while aggregate queries are allowed. A similar problem arises in OLAP through parallel classifications [18,20]. In practice dimensions are not as orthogonal as in theory. If a certain detail level is hidden, but a parallel classification exists through other (unrestricted) dimensions, the hidden detail values can possibly be revealed by a single authorized query. An approach to address such *single-query inferences* is to deny queries involving less than a certain number of records (query-set size control [1], also called *smallness indicator* by [20]). However, it has been shown that a combination of allowed (aggregate) tracker queries can be used to infer hidden detail data [1,4]. Such *multi-query inferences* are a lot harder to address.

Further research in this area might be worthwhile, even though the required efforts are probably not feasibly for most "real life" projects. [20] present an indicator based approach to address single-query inference. However, it does not provide a means against trackers that intelligently combine multiple queries. One interesting approach is the use of data mining tools to detect potential inference problems in the query audit logs [22].

3. OLAP SECURITY MECHANISMS

In general the requirement analysis and conceptual security design phases should focus on semantic aspects without yet considering implementation details. However, in order to develop a useful design methodology, it is necessary to keep the capabilities of concrete systems in mind. In this section we present OLAP security mechanisms on a vendor independent level. The next section (shortly) covers concrete implementations in commercial systems.

Security mechanisms in general provide means to define which security subjects (users, groups, roles) may or may not access certain security objects (i.e. sensitive data) applying a particular access type (usually read access in OLAP), see also [6]. The closure assumption specifies whether everything is forbidden unless explicitly allowed (*closed world*) or vice versa (*open world*). Due to the usually high-level users of OLAP applications an open world policy might be appropriate. In fact all evaluated commercial systems (see section 4) follow an open world policy. Another aspect is the applicability of the *ownership* principle. If security objects are owned by security subjects, this leads to a decentralized policy. As the "owners" of warehouse data should be the owners of the corresponding operational data, this would require a means of deriving the access control policies from the operational data sources. However, as mentioned before, it is hard to decide about the ownership of aggregated data from different sources/owners. A (centralized) administrator based policy is the alternative.

3.1 Security Enforcing Architecture Component

One way to categorize the different security mechanisms is by the architecture component used to enforce it. Candidates are:

- The *relational DBMS* holding the data warehouse (usually a star/snowflake schema in ROLAP or mixed environments). Access control is applied by creating multiple physical databases (data marts) or by using SQL views. Access restrictions expressed in the multi-dimensional model have to be translated to the relational model.
- The *OLAP server*, if available. This should be the preferred approach as the multidimensional model is used and the client/server environment provides the necessary protection against bypassing the security mechanism.
- The *front-end* application. The multidimensional model is used to express the restrictions, but preventing unauthorized access by bypassing the system might be difficult. While the last two approaches only provide an implicit limitation of the analysis tool (certain data is not provided by the OLAP server or DBMS), this approach can be extended to explicitly limiting the user interface [7,11]. Import/export functions or certain visualization components can be disabled for certain users.

However, the available possibilities depend on the deployed architecture. Obviously an OLAP server based security solution cannot be used in a 2-tier fat-client ROLAP environment and the use of SQL views on the relational data warehouse level is not suitable for MOLAP systems.

3.2 General Approach

Another way to classify the available security mechanisms is to distinguish view and rule based approaches.

Views are declaratively defined subsets of the entire system. The analysis tool is restricted to authorized queries by limiting the navigation capabilities (hiding structural metadata elements or dimension members). The idea is to hide the existence of sensitive data and not only the data itself. Different users (or user groups/roles) see different cubes which are all built from the same source data. The concept is transparent for the end user as no queries are denied. However, this transparency is also a danger as missing data might falsify trends and analysis results.

Views can be created using SQL views on the star/snowflake schema or multidimensional views on the OLAP engine level if supported. Although the requirements are actually expressed in multidimensional terms (see above), a common notation for views on this level is not available. In the relational world views are expressed as results of relational queries. However, this approach cannot be directly translated into the multidimensional world. A SQL query on a relation results in a relation itself, but a multidimensional query performed on a hypercube does not necessarily result in a hypercube. Additionally there is no common query language (although first approaches are made, such as Microsoft's MDX).

A *rule* based approach does not present (structurally) different cubes to the users. All users know of the existence of the entire cube (with all dimensions and measures), they are only not allowed to view all of it. Access rules (usually in form of Boolean expressions) define what a user may see and what he may not see. These rule expressions can, however, become very complex and hard to maintain (see section 4).

In rule based authorization systems OLAP reports might be denied or include blank cells. Certain complex policies can only be expressed using rule based approaches with cell-level granularity. To inform the user about falsified results, cells that have been left blank for security reasons are usually explicitly marked (e.g. with "N/A").

As both (view and rule) approaches have their advantages a combination of both concepts is desirable. Such *hybrid* approaches allow to specify complex (cell-level) rule constraints as well as view definitions to filter metadata elements and dimension members in order to provide a certain degree of end-user transparency.

3.3 Problems Related to Hiding Information in Cubes

The multidimensional nature of OLAP data and the extensive use of aggregation arises several problems when hiding information in cubes. If certain cube slices are hidden, e.g. if certain objects are hidden from a region manager's view, the data on a *Region* level will either be falsified (if only visible objects are included), or, if it is unmodified, tracker queries to infer hidden data might become possible. These courser granularity levels should thus also be hidden, i.e. the top hierarchy level ("All") will have a new semantic.

Hiding levels of detail in certain slices of the same dimension (e.g. hiding the *Tour* level only for certain objects) creates facts with different base granularities (see figure 3). This is sometimes referred to as a frayed dimension; many OLAP systems do not support this. However, "fake" dimension members (shown gray in the figure) might be usable, replacing the hidden dimension members, e.g. the individual tours of a certain object, with a single one representing their aggregate. In ROLAP this can be accomplished by means of (quite complex) SQL views.

Complex security requirements, such as hiding levels of detail in certain slices of a different dimension, have been presented. These can only implemented using rule-based approaches and will possibly result in denied or partially performed queries (including blank or "N/A" cells). If partial execution is used, the semantic of the totals in such reports is unclear, as it is not defined whether the hidden cells are included or not.

For example, say, a regional manager with access to daily data for his object but only to monthly data for the others starts with a query on ticket sales by month and object for all castles. This query is allowed and fully performed. He then issues a drill-down operation on the *Time* dimension to daily data. If the query is not denied, the results will be incomplete (daily data for some objects is not accessible). But what about the totals in the report? They can either remain unchanged and reflect the "real" totals (i.e. over all objects), which leaves the report in an inconsistent state. Or they can be the sums over the actually displayed values, which means that the sums have changed from one query to the other even though the queried domain (namely all objects) has remained the same. In fact, both approaches can be found in today's commercial systems (sometimes the result even depends on the syntactical formulation of the query).

4. IMPLEMENTATION IN COMMERCIAL SYSTEMS

In this section we present the access control capabilities of some commercial OLAP systems and present the use of SQL views in ROLAP. Due to space limitations we can only give a short overview at this point. For a summarized security feature comparison of the evaluated products and their capabilities to implement the requirements identified in section 2, see table 1.

4.1 ROLAP Based Tools (SQL Views)

In projects with ROLAP based tools relational views are used if the product does not support sufficient access control on OLAP server level. The use of SQL views is similar to building dependent data marts of a data warehouse. However, SQL views on the relational data warehouse level can become very complex and hard to maintain. Another problem arises when precalculated (materialized) aggregates exist. These have to be filtered correspondingly. Additional mechanisms have to be applied for metadata filtering in order to hide certain structure elements.

Basic requirements can usually be implemented easily with relational views. Measures can be hidden by applying vertical filtering (hiding of columns) on the fact table. In order to hide cube slices horizontal filtering has to be applied to the fact and/or dimension tables. Depending on the architecture filtering only the dimension tables may be sufficient as it limits the queries possible through the analysis tool. In practice extra ACL (access control list) columns in these tables can be used to simplify the view maintenance [11].

Complex policies are more challenging. Hiding levels of detail in certain slices of the same dimension creates facts with different base granularities. However, not all systems support fact table partitions (that could be simulated by views) at different granularities. Alternatively, a "fake" dimension member (see above) can be used. More complex policies are possible, but dangerous as filtering out the corresponding facts might lead to falsi-

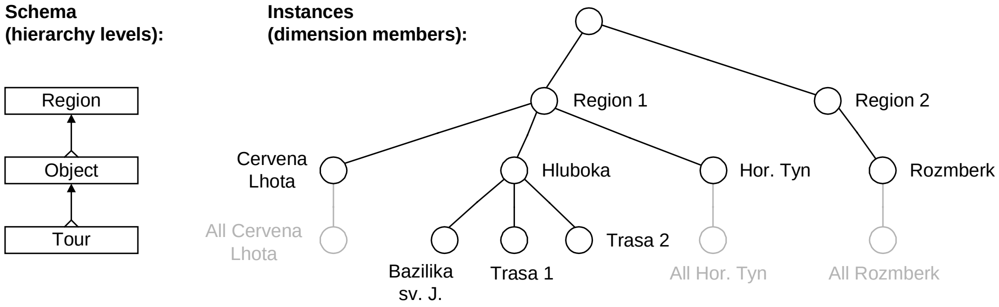

Figure 3. Frayed dimension due to member filtering

fied analysis results. No explicit denial of queries or "N/A" marking of cells can be accomplished.

4.2 Microsoft SQL Server (OLAP/Analysis Services)

The security model for the first release of Microsoft OLAP Services (bundled with SQL Server 7.0) was, for the most part, extremely simple [21]. Only two scopes for access control (server and cube level) were available. In the service pack 1 (SP1) release *cell-level security* was introduced, providing a finer degree of control [12]. Using cell-level security, users can be granted or denied permission to access data down to individual cells within a cube. Cell-level security is a rule based approach. The restrictions are defined by access rules using MDX Boolean expressions. Even complex constraints can be expressed with cell-level security rules. However, queries are explicitly denied or the returned facts are filtered (partial execution with blank or "N/A" cells).

SQL Server 2000 enhances the security model of OLAP Services (now called Analysis Services) to include, in addition to cell-level security, a *dimension security* feature that allows metadata and dimension member filtering to provide end-user transparency. With dimension security measures, hierarchy levels, and dimension members (i.e. slices) can be hidden.

Table 1. Security feature comparison of the evaluated products

	Product	ROLAP based products	Microsoft SQL Server 2000	MicroStrategy 7	Cognos PowerPlay	Oracle Express
Info	Evaluated release	N/A	8.0 BETA	7.0 BETA	6.0	6.2
	Supported security feature(s)	SQL views	Cell-level and dimension security	Access control list and security filters	User class and dimension views	Permission programs
	Security enforcing architecture component	DBMS	OLAP server or OLAP front-end	OLAP server or OLAP front-end	OLAP front-end	OLAP server
	General approach	View	Hybrid[3]	View	View	Rule
	Security policy	Closed world	Open world	Open world	Open world	Open world
	Security administration	Ownership	Administrator	Ownership	Administrator	Administrator
Requirements	Hide whole cubes	●	●	●	●	●
	Hide certain measures	●	●	●	●	●[3]
	Hide slices of a cube	●	●	●	●	●[3]
	Hide levels of detail	●	●	●	●	●[3]
	Hide levels of detail in certain slices of same dimension	●[4]	●	○[6]	●	●
	Hide certain measures in certain slices	●[5]	●	○[6]	–	●
	Hide complex slices (dices) of a cube	●[5]	●	●[5]	–	●
	Hide levels of detail in certain slices of a different dimension	○[5,7]	●	○	–	●
	Dynamic/data driven constraints	○[5,8]	○[8]	○[5,8]	–	○[8]
	Inference control	–	–	–	–	–

[2] Microsoft combines cell-level security as a rule based authorization mechanism with view limiting dimension security.

[3] As a purely rule based approach, permission programs just hide the data (facts) and not their existence, even for basic requirements.

[4] SQL views may become very complex and their maintenance may become an issue.

[5] No explicit denial of queries (or "N/A" marking of cells) can be accomplished. Thus, complex fact filtering is dangerous as it might falsify analysis results.

[6] Only by means of a SQL view layer.

[7] Only possible, if the system supports multiple fact table partitions at different granularities.

[8] If a direct implementation is not possible, some dynamic constraints can be converted to static ones, which, however, have to be maintained appropriately.

4.3 MicroStrategy 7

MicroStrategy 7 provides two means of access control [13]. First, an *access control list* is maintained for all metadata objects, including attributes (i.e. dimension hierarchy levels) and metrics (measures), but also filters, templates and predefined reports. The owner or an administrator decides who may access the object. For example, if a user is not granted read access to a certain dimension hierarchy level, he will not be able to build reports on that granularity level, or drill to that level from an existing report. However, if someone with access to that level creates such a report and grants another user access to it, he will be able to run it.

The second way to control the data access in MicroStrategy 7 is by so called *security filters*. These prevent users from seeing certain data in the database. A filter is a construct that basically represents the slicing of an OLAP query. Filters can be based on attribute (dimension) members or metrics (measures). Security filters result in implicit WHERE clauses in the generated SQL code (MicroStrategy 7 is a relational OLAP tool). If certain complex filtering rules cannot be accomplished by security filters, a SQL view layer can be additionally put in place (compare section 4.1).

4.4 Cognos PowerPlay

In Cognos PowerPlay [3] security is enforced by the front-end tool using secure (possibly encrypted) authorization files. Access control is defined on category (Cognos terminology for dimension member), measure, or dimension level, using a multidimensional view approach. It is implemented either by creating multiple cubes using *dimension views* and distributing the cubes to the different user groups, or by defining *user class views* on shared cubes.

Cognos PowerPlay is very flexible in hiding categories of a single dimension. However, it is not possible to define complex constraints involving multiple dimensions. Such restrictions must be implemented by populating multiple cubes with different data subsets (like data marts).

4.5 Oracle Express

In Oracle Express access to a database is controlled by using database *permission programs* [14], providing a rule based approach. In each database, permission programs can be created as user-defined Boolean functions. When a database is opened, Oracle Express runs the corresponding permission program. Within the database permission programs, PERMIT commands are used to establish access conditions for objects in the database. The conditions for granting permission on a database object consist of one or more Boolean expressions.

Dimension members on different granularity levels are treated independently (data is held redundantly, precalculating all aggregates). Thus, Oracle Express is very flexible in specifying even complex constraints. On the other hand, as permission programs provide only (fact) data filtering, leaving the dimension members and metadata unchanged, it is not possible to hide the existence of sensitive data (end-user transparency).

5. CONCLUSIONS AND FUTURE WORK

In this paper we have first given an overview of data warehouse and OLAP security. Due to the fact that the range of possible users accessing a data warehouse via OLAP applications is steadily growing, the necessity of proper access control mechanisms is crucial in order to ensure the confidentiality of the sensitive data.

We have introduced an OLAP security design methodology and identified different access control requirements. Today’s commercial systems provide some mechanisms to cope with these requirements. However, the approaches are highly proprietary. There is no proper foundation for access control in the multidimensional world.

In our future work, within and beyond the GOAL project, we will concentrate on modeling the security semantics. As mentioned before, there is no conceptual layer for OLAP security design. In order to approach the issue from the application side we will follow a methodology that was already proposed by [15]:

- Carefully define the security semantics by refining the presented requirements
- Outline a security constraint language (SCL) for expressing corresponding rules in the (in this case multidimensional) conceptual model
- Provide a graphical notation for the constraints, in this case based on a multidimensional notation, such as ME/R [17] or an UML-based approach

The final step will be a tool support to automatically implement the modeled constraints in a target system.

6. ACKNOWLEDGMENTS

We would like to thank our project partners for helpful comments and stimulating discussions. The partners involved in the GOAL project are Vienna University of Technology (A), Czech Technical University in Prague (CZ), University of Essen (D), Technical University Košice (SK), as well as CertiCon and LUMARE GIS companies (both CZ). Additionally, we would like to thank the vendors of the evaluated products for their support.

7. REFERENCES

[1] Castano, S., Fugini, M., Martella, G., Samarati, P.: Database Security. ACM Press, 1994.

[2] Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. 1996.

[3] Cognos Incorporated: Schrittweise Anleitungen für Transformer. Cognos Power-Play Version 6.0. 1998.

[4] Denning, D.E., Schlörer, J.: Inference Controls in Statistical Databases. In IEEE Computer Vol. 16(7); July 1983.

[5] Doshi, V., Jajoda, S., Rosenthal, A.: A Pragmatic Approach to Access Control in Data Warehouses. Via private communication, 1999.

[6] Essmayer, W., Wagner, R., Kapsammer, E., Tjoa, A.M.: Meta-Data for Enterprise-Wide Security Administration. In Proceedings of the Third IEEE Computer Society Metadata Conference; NIH Campus, Bethesda, Maryland, April 6-7, 1999.

[7] Katic, N., Quirchmayr, G., Schiefer, J., Stolba, M., Tjoa, A.M.: A Prototype Model for Data Warehouse Security Based on Metadata. In Proc. DEXA 98, Ninth International Workshop on DEXA; Vienna, Austria, August 26-28, 1998.

[8] Kimball, R.: Hackers, Crackers, and Spooks; Ensuring that your data warehouse is secure. In DBMS Magazine; April 1997.

[9] Kimball, R. et al.: The Data Warehouse Lifecycle Toolkit. John Wiley & Sons, Inc.; New York et al., 1998.

[10] Kouba, Z., Mikšovský, P., Matoušek, K., Zach, P.: Application A1 Specification. GOAL Technical Report TR8; INCO-Copernicus project no. 977091; March 1999.

[11] Kurz, A.: Data Warehousing Enabling Technology. MITP-Verlag; Bonn, 1999.

[12] Microsoft Corporation: Microsoft SQL Server OLAP Services Cell-level Security White-paper. 1999.

[13] MicroStrategy Incorporated: MicroStrategy 7 Administrator Guide. 2000.

[14] Oracle Corporation: Oracle Express Database Administration Guide. Release 6.2; Part No. A59962-01; 1998.

[15] Pernul, G.: Database Security. In Advances in Computers, Vol. 38, Yovits, M.C. (Ed.); Academic Press, 1994.

[16] Rosenthal, A., Sciore, E., Doshi, V.: Security Administration for Federations, Warehouses, and other Derived Data. In Database Security, IFIP 11.3; 1999.

[17] Sapia, C., Blaschka, M., Höfling, G., Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. In Kambayashi, Y. et al. (Eds.): Advances in Data-base Technologies; LNCS Vol. 1552; Springer, 1999.

[18] Shoshani, A.: OLAP and Statistical Databases: Similarities and Differences. In Proc. PODS 97; Tucson, AZ, 1997.

[19] Samtani, S., Mohania, M., Kumar, V., Kambayashi, Y.: Recent Advances and Research Problems in Data Warehousing. In Proceedings of the International Workshop on Data Warehousing & Data Mining, Mobile Data Access, and New Database Technologies for Collaborative Work Support & Spatio-Temporal Data Management; Singapore, November 19-20, 1998.

[20] Steger, J., Günzel, H.: Identifying Security Holes in OLAP Applications. To appear in Proc. Fourteenth Annual IFIP WG 11.3 Working Conference on Database Security Schoorl (near Amsterdam); The Netherlands August 21-23, 2000

[21] Thomsen, E., Spofford, G., Chase, D.: Microsoft OLAP Solutions. John Wiley & Sons, Inc.; New York et al., 1999.

[22] Thuraisingham, B., Schlipper, L., Samarati. P., Lin, T.Y., Jajodia, S., Clifton, C.: Security issues in data warehousing and data mining: panel discussion. In Data-base Security XI; IFIP, 1998.

APPENDIX. SCENARIO DATA MODEL

This (slightly simplified) version of the data model of GOAL application A1 is used as a scenario. It turns out to be quite suitable to describe the different access control approaches. There are two separate cubes (*People* and *Services*) with different sets of measures. *People* facts store the data containing the visitor admissions to the historical monuments. *Services* contains the data concerning the services consumed by the visitors within a monument visit. Four dimensions exist in total, two of them (*Time* and *Geography*) are shared by both cubes. Facts of the *People* cube are on a *Tour* granularity level, while facts of the Services cube are of less detail (*Object* granularity level).

Figure 4 shows the conceptual multidimensional data model in ME/R notation [17] which extends traditional E/R techniques in order to be able to express dimension hierarchies, etc.

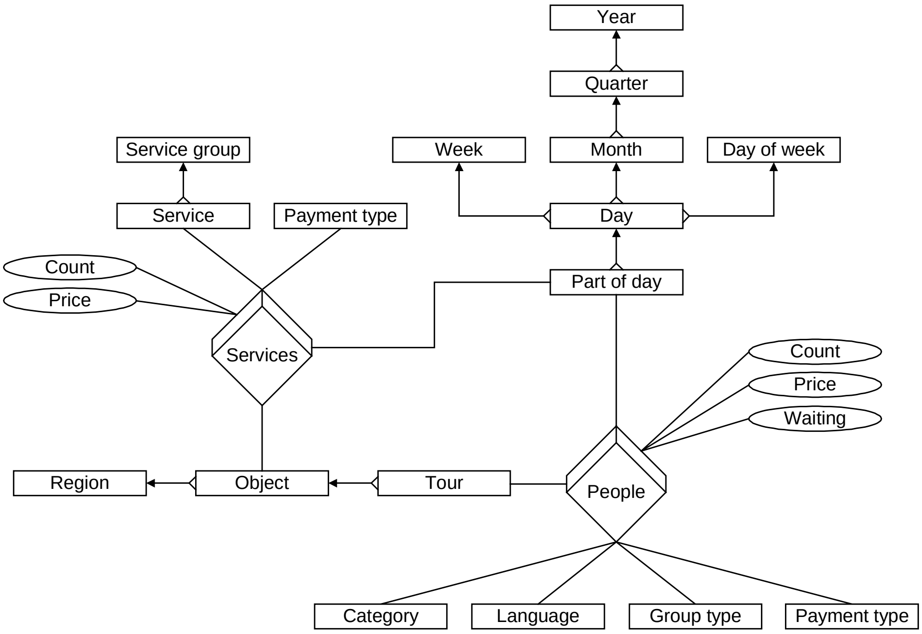

Figure 4. Multidimensional model of the scenario