02- IS

ISAS
SCI 2002

The 6th World Multiconference on Systemics, Cybernetics and Informatics

July 14-18, 2002
Orlando, Florida, USA

PROCEEDINGS

Volume VII

Information Systems Development II

Organized by IIIS
International
Institute of
Informatics and
Systemics

Member of
International Federation of
Systems Research IFSR

EDITED BY
Nagib Callaos
John Porter
Naphtali Rishe

Implementation of Security in Semantic Binary Object Database[*]

Naphtali RISHE, Artyom SHAPOSHNIKOV, Scott GRAHAM, Gustavo PALACIOS

High Performance Database Research Center
School of Computer Science
Florida International University
Miami, FL 33199, USA

ABSTRACT

In this paper we introduce a semantic binary object database technology with userviews and show how the security and data protection can be implemented in a client-server semantic database. We describe how user accesses can be restricted to parts of semantic schemas (userviews) utilizing a schema management solution based on a semantic metaschema. We also describe how network and password security is implemented in our semantic binary database server.

Keywords: semantic, object oriented, database, security, userview, metaschema.

1. INTRODUCTION

Semantic Object Binary Database—SemODB [1] is a technology developed for storing information as a collection of binary facts about real world objects. The Semantic Binary Database uses more flexible data representation than the regular object oriented models. For example, an object can be dynamically assigned to arbitrary new classes called categories. One-to-many relationships are represented efficiently and naturally using the same binary facts used to represent the regular many-to-one relationships. Like object-oriented databases, semantic databases use database schemas that describe a set of possible object classes called *categories*, sets of object attributes, methods for each category and relationships between objects called *relations*. Only binary relationships are allowed in a semantic binary database, meaning that all possible relationships are limited to represent a relation between two objects. While this limitation allows simple and efficient data structures, it does not limit the generality of semantic databases, because N-ary type of relationships can be easily converted to N binary relations to an additional object that represents the relation itself. By limiting ourselves to binary relationships, we can store all possible types of object relationships and attributes as an ordered collection of binary facts. Without loss of generality, the object attributes can also be considered as a special case of binary relation: a relation between an object and a value representing the attribute value, such as a number or a string.

Userviews allow database administrators to create a number of virtual semantic databases that are derived from an existing database. A view defines virtual categories and relations that map to physically stored categories and relations. Such mapping can be a direct one-to-one mapping or a more complex mapping that is defined by database queries. By limiting user accesses to certain userviews, we can implement very flexible data protection schemes for semantic databases. Not only can we hide certain categories, attributes and relations from unauthorized users, we can also rename the categories and relations, as well as define completely new schemas that use database queries to filter out potentially security-sensitive information.

This paper will introduce how the security and userviews were implemented in our semantic database. We start by introducing semantic schemas in general and a semantic metaschema for userviews [1]. We also discuss the implementation of security features in our database server.

2. SEMANTIC SCHEMAS

Categories and relations of a particular database form a semantic schema. Categories are represented as boxes on the schema. Attributes of objects are represented as a collection of names and types inside the boxes. Relations between categories are represented as directed arrows pointing from *domain* of a relation to the *range* of the relation. Figure 1 shows a semantic schema of a simple semantic database:

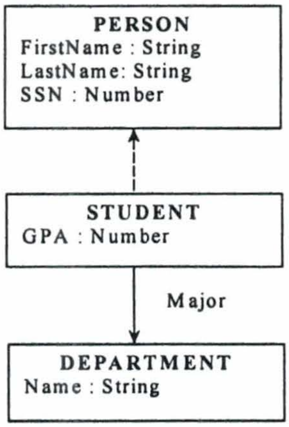

Figure 1. A simple schema.

Here the dashed line represents an inheritance relation between STUDENT and PERSON. A more detailed description of semantic binary schemas can be found in [1].

[*] This research was supported in part by NASA (under grants NAG5-9478, NAGW-4080, NAG5-5095, NAS5-97222, and NAG5-6830), NSF (CDA-9711582, IRI-9409661, HRD-9707076, and ANI-9876409), ONR (N00014-99-1-0952), and the FSGC.

3. METASCHEMAS

Our implementation of a semantic database stores the database schemas as a part of the database itself. The categories and relations of an arbitrary database schema are represented as objects conforming to a semantic *metaschema*. The semantic metaschema is a schema of a database for storing any semantic schema. The objects in any semantic schema such as the category Student and attribute GPA are metaobjects belonging to some categories of the metaschema. Figure 2 shows the metaschema for our semantic binary database.

The category STUDENT is an object of the metacategory CATEGORY. The relation Major is an object that belongs to the metacategory RELATION. The categories can be abstract (represented by the ABSTRACT CATEGORY) and concrete (CONCRETE CATEGORY). Concrete categories include attributes such as numbers, strings, enumerated type, and binary attributes. The concrete categories are represented by the categories NUMBERS RANGE, STRINGS RANGE, ENUMERATED TYPE, and BINARY DATA on the metaschema. Abstract categories such as STUDENT, are stored in the category ABSTRACT CATEGORY on the metaschema. A supercategory METAOBJECT stores both categories and relations. Basic integrity constraints are represented by the categories CO-IDENTIFICATION KEY, MANY TO ONE, ONE TO MANY, TOTAL, COVERING GROUP, and DISJOINT GROUP at the metaschema.

Our implementation of SemODB allows the users to perform the same database operations with respect to schemas as to the regular data. The metaschema is an integral part of each database. This permits the semantic database to easily implement on-the-fly schema changes. Database applications using SemODB can dynamically change the database schemas as well as data stored in the database in the same transaction.

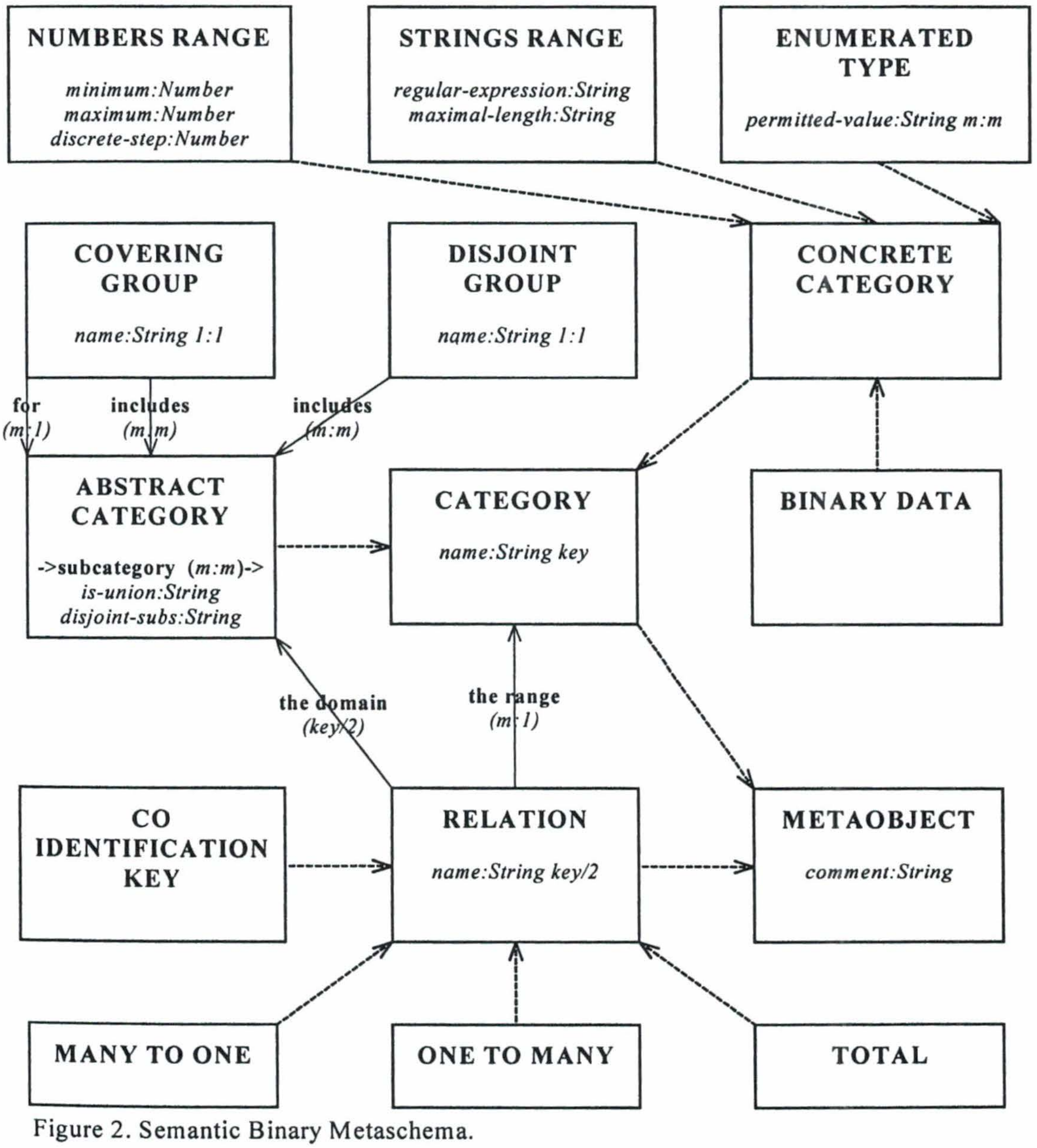

Figure 2. Semantic Binary Metaschema.

This is one of the advantages of SemODB as compared to regular object oriented databases that usually do not permit on the fly schema changes.

4. USERVIEWS

Userviews are also an integral part of each database. A userview defines a sub-schema of a semantic schema. A user has access only to a limited number of userviews and always opens a database through some userview. A database administrator may access the complete schema that includes all categories and relations without restriction.

Figure 3 shows the metaschema for userviews. The USERVIEW category is a subcategory of the category METAOBJECT. It defines a user's view of the schema. Userview is a collection of metaobjects with restricted VIEWABILITY. The VIEWABILITY defines which metaobjects can be read, inserted or deleted in a particular userview. The category USER stores the information about the users who can access the USERVIEW.

The database server uses the meta-data stored according to the metaschema for userviews to verify if a user has permissions to read or modify objects that belong to certain categories and relations. If a user does not have access to a category, relation, or attribute, such metaobjects and the data will not be visible to the user application.

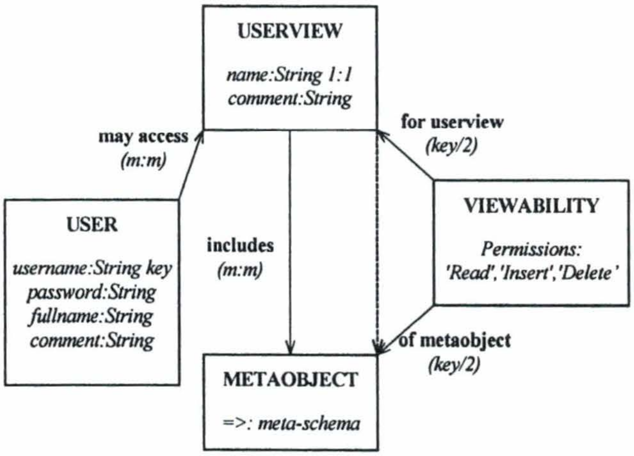

Figure 3: Metaschema of userviews.

5. SECURITY IN CLIENT-SERVER MODEL

The SemODB server is started by running a SemODB kernel application that we call Control Server (see Figure 4). After the SemODB Control Server is started, the remote clients can connect to the server via the internet and the TCP-IP protocol stack. There are two types of remote SemODB clients: SemODB API remote clients and semantic SQL/ODBC remote clients. SemODB API remote clients include Java and C++ applications connecting to the database directly and utilizing the SemODB application programming interface. SQL/ODBC clients connect to the database via TCP-IP to the ODBC driver.

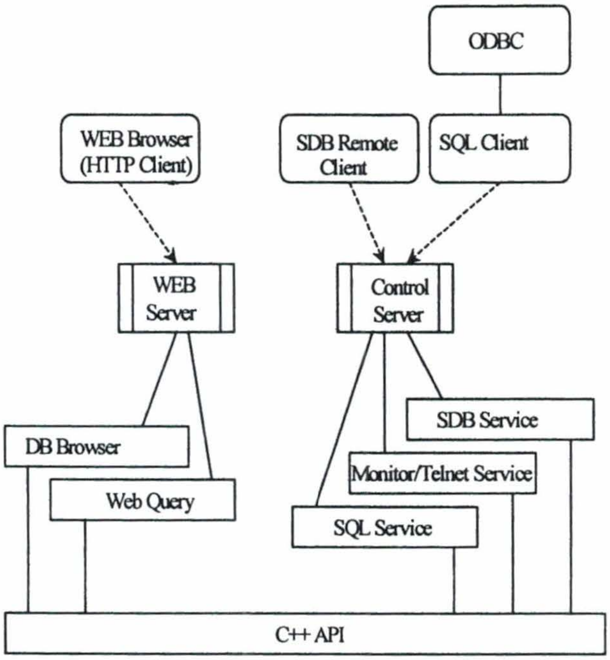

Figure 4. Client-Server SemoDB.

Figure 5 shows the architecture design of our security implementation for a client-server semantic database. Database clients connect to the database control server using an SSL encrypted connection. Network password security is protected by SSL as well as by hash digests. The passwords are never transmitted over the network, instead we transmit their hash digests obtained using the Haval [3] secure hash algorithm. It is computationally infeasible to find the password given the secure hash digest.

The database Control Server authenticates the clients using the Encrypted Client Authentication Database. The database contains access rights for each user. The access rights are grouped into access groups represented by the category ACCESS. An access group may include several userviews to one or more databases. There is one default userview for each access group that is used when a userview is not specified by the client. When a database client connects to the database, the Database Control Server queries the Client Authentication Database and assigns a userview to the client based on the username, password and the requested name of userview. Once a connection is established, the user can only retrieve the data in the corresponding userview.

The client authentication database is encrypted by using a strong 128 bit twofish encryption algorithm [2] to prevent unauthorized access to passwords even if the access to the authentication database file is compromised.

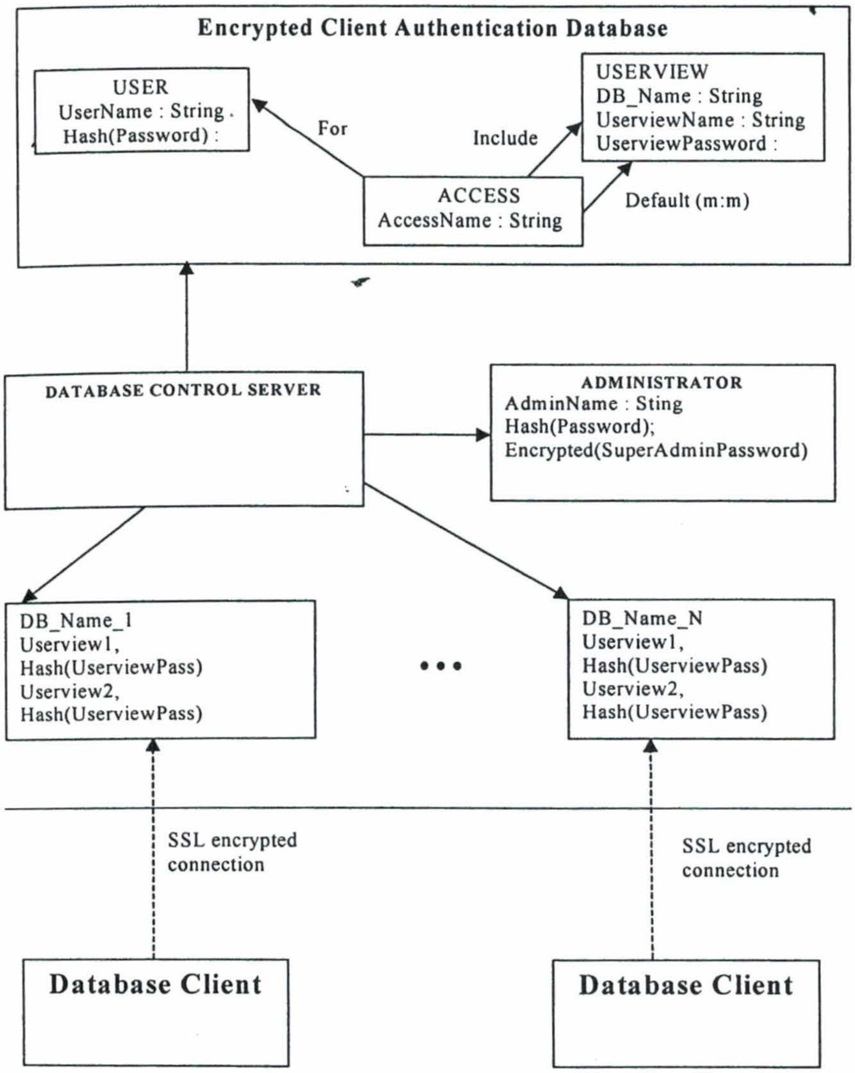

Figure 5. Security implementation.

Only an administrator can start the database control server. When the Control Server starts, the administrator's password is verified by using a small database shown in the ADMINISTRATOR category box that stores secure hash digests derived from the administrator passwords using the Haval algorithm [3]. Once the administrator credentials have been verified by the server, the SuperAdminPassword is decrypted using the administrator's password. The SuperAdminPassword is then used to decrypt the encrypted Client Authentication Database that stores the permissions for the clients. This setup allows us to have more than one database administrator and password. A GUI administrative tool was developed to manage the userviews, user access permissions, users, and administrators.

6. CONCLUSION

We have described the security model of our semantic database based on a semantic metaschema and userviews. The Semantic database offers convenient and flexible tools to restrict user access to parts of databases called userviews. Userviews are implemented as a part of semantic binary Metaschema. We also described how our implementation of semantic database server implements network and password security.

7. REFERENCES

[1] Naphtali Rishe. Database Design: the semantic modeling approach. McGraw-Hill, 1992, 528 pp.

[2] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, "Twofish: A 128-Bit Block Cipher," http://www.counterpane.com/twofish-paper.html

[3] Y. Zheng, J. Pieprzyk and J. Sebery, "*HAVAL - a one way hashing algorithm with variable length of output*", Advances in cryptology - AUSCRYPT 92, Lecture Notes in Computer Science, vol. 718, pp. 83-104, 1993.