

The effect of heterogeneity and autonomy on federated database security

Russell Davie^a and Reinhardt Botha^b

^a Port Elizabeth Technikon, s20026335@student.petech.ac.za

^b Port Elizabeth Technikon, reinhard@petech.ac.za

Abstract: *Federated Database Systems have been developed in order to provide an effective means of integrating scattered database systems. These systems are designed to allow seamless access to shared organisational information without affecting the existing infrastructure. The enforcement of security at the federation level must take into account the security requirements and the protection of the component systems. Components in a federation remain autonomous and may be heterogeneous, thus complicating the task of ensuring federated security. This paper describes a Federated Database System and discusses the effect that heterogeneity and autonomy have on security in a Federated Database System.*

Keywords: *Federated Database System, Database Security, Data Integration*

1. Background

The rapid growth of database technologies as well as the growth of networking over the last two decades, has had a major impact on the information processing requirements of organisations. Much effort has been placed on interconnecting the increasing number of databases scattered across several sites in order to share information. To accomplish the integration of information sources, computing resources must be interconnected into enterprise-wide information systems. These enterprise-wide systems are expected to provide seamless access to information, while providing the protection specified by the security policies of the participating organisations.

In order to reconcile the contrasting requirements of the different Database Management Systems (DBMSs), tools that enable users of one system to use other systems, are being developed. One such development is a Federated Database System (FDBS). In complex information systems such as FDBSs, security requirements have to be considered right from the beginning. The enforcement of security at the federation level must take into account the protection requirements and the protection of the component systems (Vimercati & Samarati, 1997). This task can be further complicated by the autonomy and heterogeneity of the component systems. This increased complexity has brought about the need to understand the effects of heterogeneity and autonomy on the security requirements of a federated database.

2. Federated Database Systems

The term Federated Database System was conceived by Heimbigner and McLoed (1985). The term has been used for several different but related database system architectures. A Federated Database System (FDBS) is a collection of co-operating but autonomous component database systems (Heimbigner & McLoed, 1985; Sheth & Larson, 1990).

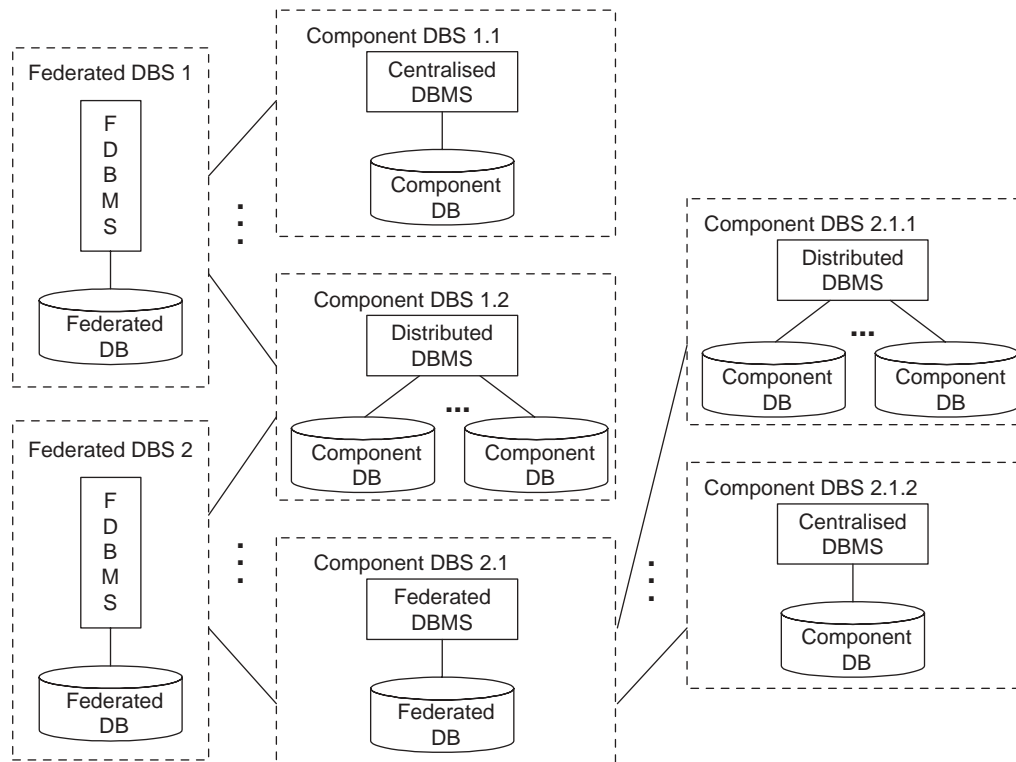


Figure 1: An example Federated Database System layout.

The component databases systems (CDBSs) are integrated to various degrees and are controlled and co-ordinated by a Federated Database Management System (FDBMS).

Figure 1 shows a layout of a FDBS. A component of a FDBS can participate in more than one FDBS (Component DBS 1.2) and a component can be a federation itself (Component DBS 2.1). A component DBMS (CDBMS) can be a centralised (Component DBS 1.1) or distributed DBMS (Component DBS 1.2).

A significant aspect of a FDBS is that a CDBS can continue its own operations and participate in a federation at the same time. The integration of the CDBSs can be controlled by the users or the administrator of the FDBS together with the administrators of the CDBSs. The amount of integration depends on the needs of the federation users and desires of the CDBSs to integrate and share their databases.

2.1 The road towards Federated Database Systems

A DBS may either be centralised or distributed. A centralised DBS consists of a single DBMS controlling a single database on the same computer system. A distributed DBS consists of a single DBMS controlling multiple databases. A Multidatabase system (MDBS) is a system that controls the operation of multiple component DBSs (CDBSs). Each CDBS is managed by a component DBMS (CDBMS) which may be different from each other. A MDBS is called a homogeneous MDBS if the DBMSs of each component are the same, otherwise the MDBS is called a heterogeneous MDBS.

Multidatabase systems can be classified into two types based on the autonomy of their

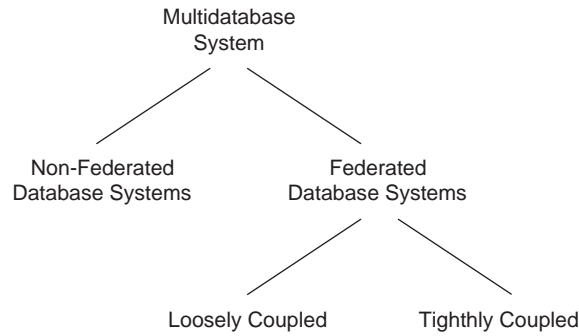


Figure 2: Taxonomy of multidatabase systems.

component database systems, as shown in Figure 2. A Non-Federated database system is the integration of component database systems that are not autonomous. A Non-Federated DBS does not distinguish between local and nonlocal users. A Federated Database System is the integration of component database systems that remain autonomous, yet allow the partial and controlled sharing of data within the component database systems. There is no centralised control in a federated system, because the CDBSs and their administrators control the access to their data.

A FDBS can be categorised according to who manages the federation. A Federated Database System is loosely coupled if it is the user's responsibility to create and maintain the federation. The federation and its administrators enforce no control in loosely coupled federated database systems. A Federated Database System is tightly coupled if the administrators of the federation are responsible for creating and maintaining the federation and actively control access to the component database systems.

2.2 Characteristics of Federated Database Systems

Multidatabase system, of which FDBSs are a specific type, can be characterised along three related dimensions, namely distribution, heterogeneity and autonomy (Sheth & Larson, 1990). Figure 3 shows the three dimensions and how DBS systems are derived from these characteristics.

Distribution refers to data being distributed among multiple databases. The distribution of data can be regarded among the following aspects: Data may be stored on single or multiple computer systems. CDBSs may be centralised or distributed but interconnected by a communication system. Data may be differently distributed vertically or horizontally (in relational terms) among multiple databases and identically structured, multiple copies of data may be maintained of some or all of the data. Distribution of data increases availability and reliability as well as improved access time due to the optimised location of data.

Heterogeneity occurs because of technical differences such as hardware, system software (operating systems) and communication systems. Heterogeneity in FDBSs is more concerned with differences related to the DBMSs of the CDBSs. Differences in DBMSs result from differences in the data models and differences

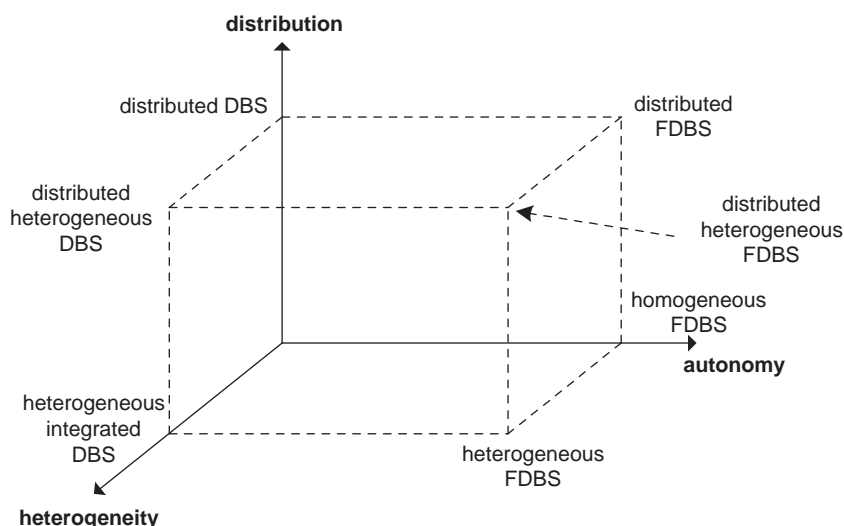


Figure 3: Taxonomy of DBS characteristics and types.

at system level. Data structure and constraints as well as differences in query languages aspects can lead to the heterogeneity in data models. Differences in semantics occurs when there is a disagreement about the meaning, interpretation or intended use of the same or related data.

Autonomy means that the organisational entities control their own component DBSs in a federation. This means that the CDBSs may be under separate, independent control. There seems to be a trade-off between the federation's functionality and the degree of autonomy held by each CDBS (Essmayr, Katner, Preishiber, Pernul, & Tjoa, 1995).

A CDBS participating in a FDBS may possess several types of autonomy.

- **Design autonomy** - implies that a CDBS has the ability to choose its own design with respect to management, representation and semantic integrity of data, as well as the constraints, functionality and implementation of the system (Veijalainen & Popescu-Zeletin, 1988).
- **Communication autonomy** - implies that a CDBMS has the ability to decide whether to communicate with other CDBMSs (Veijalainen & Popescu-Zeletin, 1988).
- **Execution autonomy** - implies that a CDBMS has the ability to execute local operations without interference of external operations from other CDBMSs or FDBMSs (Veijalainen & Popescu-Zeletin, 1988).
- **Association autonomy** - implies that a CDBS has the ability to decide whether and how much to share its functionality and resources with other CDBSs (Sheth & Larson, 1990).
- **Authorisation autonomy** - implies that a CDBS has the ability to specify the access allowed or denied on its objects by other CDBS (Jonscher & Dittrich, 1993; Vimercati & Samarati, 1997; Jajodia & Wijesekera, 2001).

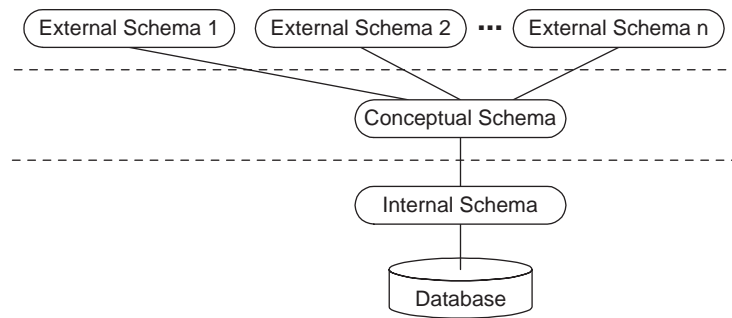


Figure 4: The three-level ANSI/SPARC DBMS schema architecture.

2.3 A Five-level Schema Architecture for Federated Databases

A DBMS provides three data description levels and mappings between them. The data seen by the users and applications is mapped to the physical data and vice versa, so that they are shielded from the low-level details of the database. To perform these mappings, the DBMS makes use of data descriptions, called schemas. Figure 4 shows the three schemas proposed by the ANSI/SPARC study group on database management systems (Tsichritzis & Klug, 1978):

Internal schema - describes the storage structure of the data within the files of physical memory. The data is physically stored as fixed or variable length records and record pointers.

Conceptual schema - describes the data of the database using the logical model of the selected DBMS. All the data and its relationships are described through the DDL of the selected DBMS and various operations on the conceptual schema are specified by the DML of the DBMS.

External schema - provides views on the data, according to the features of the selected logical model and needs of specific users and applications. Logical views are generally the description of a portion of the database's logical schema. DDL, QL and DML operations may be executed on the logical views.

The three-level schema architecture is, however, inadequate for describing the architecture of a FDBS. The three-level schema architecture needs to be extended to support the heterogeneity and autonomy of the FDBS. Sheth and Larson (1990) have adapted various schema architectures to describe a five-level schema architecture for federated systems shown in Figure 5. The five-level schema architecture includes the following:

Local schema - is the conceptual schema of a CDBS.

Component schema - is formed by translating the local schemas into a data model called the canonical or common data model (CDM) of the FDBS. Component schemas facilitate integration and negotiation tasks performed when developing a FDBS. Mappings required by the transforming processors are created during the process of schema translation from local schema to component schemas.

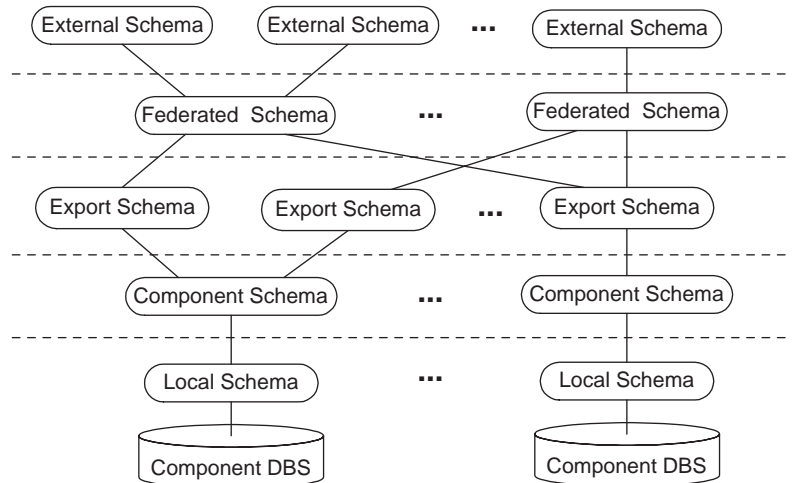


Figure 5: Five-level Schema Architecture for a FDBS.

Export schema - represent the subset of a component schema available to the FDBS.

Export schemas may include access control information regarding use by federated users. Export schemas facilitate control and management of association autonomy. Filtering processors provide the access control by limiting the set of operations that can be submitted on the component schema.

Federated schema - is an integration of multiple export schemas. A constructing processor transforms commands on a federated schema into the commands on one or more export schema. Constructing processors and export schemas support the distribution feature of a FDBS.

External schema - defines a schema for a user and/or application or class of user and/or applications in a FDBS. A filtering processor analyses the commands on the external schema to ensure their conformance with access control and constraints of the federated schema.

The extended five-level schema architecture supports the heterogeneity and autonomy of the FDBS. Each CDBS controls the information it exposes through the Export schema, thus maintaining autonomy. The heterogeneities of the CDBS are resolved and translated to the CDM of the FDBS. The architecture of a FDBS is primarily determined by the presence and arrangement of schemas and how they are constructed. Alternative architectures can be derived from the five-level schema architecture by inserting additional components, removing all basic components of a specific type and rearranging the components.

The five-level schema architecture allows development that meets the functional requirements of a FDBS. In the development of a complex information system, however, it is also important to consider security of the system and the resources it integrates.

Table 1: Security mechanisms

Security Mechanism	Description
Identification	Identification of users and applications before access is allowed.
Authentication	Verification of user or application identity.
Authorisation	Comprises and enforces the policies that determine how access is granted and delegated to users and applications.
Access Control	Mechanisms required to check whether requests or operations issued by users and applications are allowed or not and enforce the corresponding decisions.
Integrity	Enforce a set of constraints that define the correct states of a system during operations.
Confidentiality	Mechanisms required to maintain secrecy and non-disclosure of protected data.
Auditing	Record and review all security-relevant operations to maintain and improve security controls.

3. Information security in Database systems

Although the increasingly widespread use of distributed processing has become critical to support business functions, it has also posed serious problems of data security. Damage in a database environment does not only affect a single user or application but rather the entire information system. Therefore, in computer-based information systems, technology, tools and procedures concerning security are essential both to ensure system continuity and reliability and to protect data and programs from intrusions, modifications, theft and unauthorised disclosure (Saltzer & Schroeder, 1975).

Information security in a database system is concerned with preserving the three elements of confidentiality, integrity and availability of the information stored in the database. These three main aspects are each described in the following subsections.

- **Confidentiality** - means preventing, detecting and/or deterring the improper disclosure of information.
- **Integrity** - means preventing, detecting and/or deterring the improper or unauthorised modification of information.
- **Availability** - means preventing, detecting and/or deterring the improper denial of access to services provided by the system.

The security requirements of a system are specified by means of a security policy. A security policy is a written statement describing which assets to protect and why they are being protected, and which behaviours are accepted and which are not. The security policy is enforced by means of security mechanisms.

The security mechanisms that enforce the requirements of the security policy can be classified into various categories, as described in Table 1.

Table 2: Database Security Requirements

Security Requirement	Description
Protection from improper access	Consists of granting access to a database only to authorised users or applications. The DBMS has to check the requests by a user or application to their respective authorisations.
Protection from inference	Prevent the possibility of using available, aggregated information to deduce or create previously unavailable information.
Integrity of the database	Concerned with maintaining information that is correct, complete and timely.
Accountability and auditing	Consists of recording all accesses to data by all users or applications in the issue of any operation on the data.
User authentication	Users or applications are only allowed to access data once they have been identified as an “authorised” user of the system.
Confinement	Intended as the necessity to avoid undesired information transfer between authorised and unauthorised system programs.

3.1 Security problems in databases

Achieving security in database systems means identifying the threats and choosing the proper policies, “what” the security system is expected to do, and mechanisms, “how” the security system should achieve these security goals (Castano, Fugini, Martella, & Samarati, 1994). It also involves the provision of security system assurance, “how well” the security system meets the protection requirements and executes the expected functions.

The first step in achieving security is to identify the threats to the database system. A threat can be defined as any act or object that poses a danger to any computer asset (Schneider & Perry, 2001). Threats to database systems consist of improper reading, modification or deleting of data and can be classified as either accidental or malicious. Accidental threats are casual accidents independent of a determined will to cause damage. Malicious threats are intentional acts that denote a determined and fraudulent will to cause damage (Castano et al., 1994).

Secondly, the security requirements of the database system need to be identified in order to ensure proper protection. The protection requirements of a database system are described and summarised, as shown in Table 2.

Finally, security controls need to be implemented to enforce the security requirements of the database system. Castano et al. (1994) suggest that flow control, inference control and access control mechanisms should be implemented to ensure database protection and enforce the requirements of a database security system. The additional control of cryptographic techniques can be added. Cryptography hides stored data under a secret enciphering key, making data visible to anyone, but understandable to only authorised users and application, thus ensuring the confidentiality of the data (Denning, 1982).

4. Security in Federated Database Systems

While much research has been carried out in the development of FDBSs, relatively little progress has been made on development of appropriate security mechanisms for FDBSs (Jonscher & Dittrich, 1994). A FDBS provides interoperability between existing heterogeneous databases, thus providing two main advantages: it provides a user the capability to retrieve data located at different heterogeneous databases by using a single database interface and it provides organisations the means to integrate existing data from different sources within a global view. Interoperability is a significant advantage but it also dramatically increases the need for protecting the security of the CDBSs and their local users.

The existing security mechanisms have to provide new functionality when applied to a FDBS. When CDBSs join a FDBS it is important that the FDBMS afford the components the protection that they require on their vital information. Security of the federation is an important global concern that needs to be specified and translated to and collectively supported by the underlying security enforcement mechanisms of the CDBSs. A FDBS must have a security policy that co-exists with those of its components. Jonscher and Dittrich (1994) state that the security of the federation is not the property of an isolated component, but of the system in its entirety. A major problem in this context is the establishment of administrative policies that determine the authority of the different federation participants for the specification of access authorisations (Vimercati & Samarati, 1997).

To evaluate the security issues present in a FDBS, a list comparing the security requirements of a centralised DBS to those of a FDBS and the issues raised by these differences is presented in Table 3. These security issues are raised as a direct result of the three defining characteristics of a FDBS, namely heterogeneity, autonomy and distribution.

Identification and authentication mechanisms need to provide components the ability to authenticate the federation site which has requested access, as well as the option to authenticate the global user at their local site. In turn, the federation may wish to authenticate the local site to which a connection for a global user should be established. The degree to which the local site controls how subjects access its data is called authorisation autonomy (Jonscher & Dittrich, 1993). Authorisation autonomy is classified as Full, Medium and Low authorisation autonomy. Full authorisation autonomy requires that each federation subject be known to each component and must be authenticated in order to access data. Medium authorisation autonomy requires that each subject authenticates itself to the federation and the federation authenticates itself to the component. Low authorisation autonomy allows the federation to authenticate itself as well as the federation subject using the federation's identity. It may be allowed that components choose different schemes within the federation. Therefore, the FDBMS must keep track which scheme was chosen by each component and apply the proper protocol.

A powerful federated authorisation model needs to be established to account for the heterogeneous models of the component systems. Each component models may be based on relational or object-orientated DBMS, therefore heterogeneities with respect to the granularity and semantics of security subjects, security objects and access types need to be mapped and integrated. Authorisation policies may be open, a request is allowed

Table 3: Security Requirements Comparison Matrix.

	Centralised DBS	FDBS	FDBS Security Issues
Identification & Authentication	- Single user profile table.	- Multiple heterogeneous authentication policies.	- Authentication of users & sites at local & global level.
	- Single centralised site.	- Multiple distributed & autonomous sites.	- Integration & allowance for Full, Medium and Low Authorisation autonomy.
Authorisation & Access Control	- Single AC policy.	- Multiple heterogeneous AC policies.	- Mapping between MAC, DAC & RBAC policies. - Granularity of access type.
	- Single set of authorisation rules.	- Multiple sets of heterogeneous authorisation rules.	- Combination of heterogeneous closure assumptions.
	- Single security model.	- Multiple security models.	- Different data models (Relational vs. OO) - Protection defined at different levels of security model.
Integrity	- Single set of integrity constraints.	- Multiple DBMS with heterogeneous sets of constraints.	- Semantic heterogeneity of data. - Heterogeneous integrity & consistency constraints.
Confidentiality	- Data stored & accessed centrally.	- Data distributed & accessed remotely.	- Network & communication channels security. - Consider local transactions as part of global transaction, to avoid inference.
Auditing	- Centralised administration & local log files.	- CDBS maintain autonomous control & administration.	- Co-operation between CDBS administration. - Difficult to trace transactions.

unless a prohibition is inferred, or closed, a request is denied unless a permission is inferred. The combination of closure assumptions forces the choice between maintaining autonomy of components or creating a powerful federated security model. Furthermore, access control policies can be based on either discretionary, mandatory or role-based rules. Each component site maintains a set of access rules which may be heterogeneous with respect to type of policy as well as representation of similar rules. Therefore, the federation must translate and integrate the access rules specified by the CDBMSs into a common form.

The data model of the FDBS has to provide the means to specify and integrate the integrity and consistency constraints of the CDBSs since the data offered by the FDBS may be scattered over various local sites as well as copied from one site to another. There may be semantic heterogeneities of data between CDBSs, which must be checked and resolved in order to maintain data integrity.

A FDBS by its nature requires that access to its distributed data be via some type of network. It is necessary to maintain confidentiality of all information that is transferred over the communication network and standardise communication methods. In addition,

local transactions must be analysed as part of global transactions, as to the combination of data from component sites and the inferences obtained therein (Jonscher & Dittrich, 1993; Jajodia & Wijesekera, 2001).

Security auditing must be performed at the federation site as well as at the local sites. Additional audit records must be provided if control passes from the global layer of the federation to a particular CDBS as the local layer. Identifiers used within the FDBS may be managed by different administrators and stored at different locations. Complications may arise and require co-operation between federation and local administrators in auditing and accounting functions.

5. Conclusion

The development of a Federated Database System is very complex, especially when it comes to integration of the security models of the component DBSs. These security complexities are a direct result of the heterogeneity and autonomy of the FDBS. The security issues presented in this paper must be considered and resolved in order to develop a secure FDBS. The proposed, existing federated security models offer solutions to some of the security issues in a FDBS. However, there is still no model that has been able to address all security requirements and still offer true autonomy while integrating heterogeneous component DBSs.

Federated Database Systems certainly do hold great potential in enabling integration of scattered database systems. As so much current research in the field of Information Technology revolves around the integration and standardisation of information, this continued research and development of new technologies are certain to have a significant and positive effect on the development of secure Federated Database Systems.

References

- Castano, S., Fugini, M. G., Martella, G., & Samarati, P. (1994). *Database Security*. Reading, MA: Addison-Wesley.
- Denning, D. E. (1982). *Cryptography and Data Security*. Addison-Wesley.
- Essmayr, W., Katner, F., Preishiber, S., Pernul, G., & Tjoa, A. M. (1995). Access Controls for Federated Database Environments - taxonomy of Design Choices. In *Proceedings of the Joint IFIP TC 6 and TC 11 Working Conference on Communications and Multimedia Security*. Graz, Austria: Chapman and Hall.
- Heimbigner, D., & McLoed, D. (1985). A Federated Architecture for Information Management. *ACM Trans. Off. Inf. Syst.*, 3(3), 253–278.
- Jajodia, S., & Wijesekera, D. (2001). Security In Federated Database Systems. *Information Security Technical Report*, 6(2), 69–79.
- Jonscher, D., & Dittrich, K. R. (1993). Access Control for Database Federations a discussion of the state-of-the-art. In *DBTA Workshop on Interoperability of DBSs and DB App.* (pp. 1–23). Fribourg.

- Jonscher, D., & Dittrich, K. R. (1994). An Approach For Building Secure Database Federations. In *Proceedings of the 20th VLDB Conference* (pp. 1–12). Santiago, Chile.
- Saltzer, J. D., & Schroeder, M. D. (1975). The Protection of Information in Computer Systems. *Proc. IEEE*, 63(9).
- Schneider, G. P., & Perry, J. T. (2001). *Electronic Commerce* (2nd ed.). Canada: Course Technology.
- Sheth, A. P., & Larson, J. A. (1990). Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3), 183–236.
- Tsichritzis, D., & Klug, A. (Eds.). (1978). *The ANSI/X3/SPARC DBMS Framework* (No. 4).
- Veijalainen, J., & Popescu-Zeletin, R. (1988). Multidatabase Systems in ISO/OSI Environment. In N. Malagardis & T. Williams (Eds.), *Standards in Information Technology and Industrial Control* (pp. 83–97). North-Holland, Netherlands.
- Vimercati, S. De Capitani di, & Samarati, P. (1997). Access Control in Federated Systems. In *ACM New Security Paradigm Workshop* (pp. 87–99). Lake Arrowhead, USA.