Proceedings of the
1st International Workshop on
Security in Information Systems SIS 2002
ISBN: 972-98050-9-1
http://www.iceis.org

Roberto Moya Eduardo Fernández-Medina (Eds.) SIS 2002 Security in Information Systems

SIS 2002

Roberto Moya Eduardo Fernández-Medina (Eds.)

Security in Information Systems

Proceedings of the
1st International Workshop on
Security in Information Systems
SIS 2002
In conjunction with ICEIS 2002
Ciudad Real, Spain, April 2002

Roberto Moya and Eduardo Patón (Eds.)

Security in Information Systems

Proceedings of the
1st International Workshop on
Security in Information Systems,
SIS 2002
In conjunction with ICEIS 2002
Ciudad Real, Spain, April 2002

ICEIS PRESS
Setúbal, Portugal

Volume Editors

Roberto Moya
ATI, Spain
rmoya@dimasoft.es

and

Eduardo Fernández-Medina Patón
University of Castilla-La Mancha, Spain
efmedina@jur-to.uclm.es

Proceedings of the 1st International Workshop on Security in Information Systems Systems – (SIS-2002)
Ciudad Real, Spain, April 2002.
Roberto Moya and Eduardo Patón (Eds.)

Printed in Portugal
Escola Superior de Tecnologia de Setúbal
Campus do Instituto Politécnico de Setúbal
Rua do Vale de Chaves, Estefanilha
2914-508 Setúbal

Depósito Legal Nº 176842/02
ISBN 972-98050-9-1

Foreword

These proceedings contain the papers of the First International Workshop on Security in Information Systems, which was organized by the Alarcos Research Group of the School of Computer Science, Ciudad Real, Spain, April 2 – 6, 2002.

Security in Information Systems is one of the most pressing challenges facing organizations today. Global, national, and even local enterprises are driven by information. Many companies have discovered how critical this information is to the success of their businesses. And yet, few companies have effective ways of keeping their information safe, avoiding unauthorized access, intrusions, secret information disclosure, etc.

The purpose of this workshop is to serve as a forum to gather researchers, practitioners, students and anyone that works or studies or is interested in the field of Information Systems Security. The workshop shows new techniques and tendencies, and provides renovated ideas and discussion on the following specific areas related to Security in Information Systems: Security in Mobile Execution Environments, Multilevel Secure Database Systems, Authentication Mechanisms, Security in Web Databases, Certification of Security, and new directions of security, such us Immune Techniques, or Neuronal Networks.

We would like to thank all members of the Program Committee and the reviewers for their work in reviewing and selection the papers that appear in this book. We would also like to thank all the authors who have submitted their papers to this workshop. Special thanks go to the steering and organizing committee of the ICEIS, especially to Joaquim Filipe and Mario Piattini, who did a great job.

April 2002,

Roberto Moya

Eduardo Fernández-Medina Patón

Workshop Chairs

Roberto Moya
ATI, Spain

and

Eduardo Fernández-Medina Patón
University of Castilla-La Mancha, Spain.

Program Committee

Mario Piattini
University of Castilla-La Mancha, Spain.

Ambrosio Toval
University of Murcia, Spain.

Esperanza Marcos
Rey Juan Carlos University, Spain.

Verónica Canivell
University of Deusto, Spain.

Marta Oliva
University of Lleida, Spain.

Elena Ferrari
Università degli Studi dell'Insubria, Italy.

Francisco López Crespo
MAP, Spain.

Table of Contents

10. Dasgupta, D., 1999b. Immunity-Based Intrusion Detection System: A General Framework. *In the proceedings of the 22nd National Information Systems Security Conference (NISSC)*, October 18-21, 1999.
11. Dasgupta, D., 2001. Artificial Immune Systems: A Bibliography www.cs.memphis.edu/~dasgupta/AIS/AIS-bib.pdf, Dec 2001.

UML for the Design of Secure Databases

Eduardo Fernández-Medina, and Mario Piattini

Escuela Superior de Informática. University of Castilla-La Mancha.
Ronda de Calatrava 5, 13071, Ciudad Real. (SPAIN).
Tel: 34 926 29 53 00 (ext. 3715). Fax: 34 926 29 53 54
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Abstract. In this article we argue the importance of security in databases, and the need to consider security as a fundamental requirement in their development and one which is integrated at all stages of design, instead of being an isolated and marginal requirement. We propose an extension of the Use Case and Class models of UML using their standard extension mechanisms (stereotypes, tagged values and constraints) to allow us to design multilevel databases. We carry out the necessary modifications on the meta-model of UML so that security level is considered as an intrinsic aspect of the elements of the models. Finally we show an example.

1 Introduction

The rapid technological advances achieved over recent years are leading to a greater use of information systems on the part of organisations (communications, transport, banks, education, manufacturing, design, medicine, etc.) as well as an increase in the complexity of information requirements. Often information systems manage a great quantity of information, which can be especially important for the organisation. Indeed the very survival of the organisation may depend on the correct management, security and confidentiality of this information. The information managed by information systems is stored in databases and data warehouses, and therefore these are key points to examine when we come to analyse the protection of information. For this reason the security of databases is a serious aspect which must be considered explicitly, not as an isolated aspect, but as an element present in all the stages of the life cycle of the database construction [11].

Information relating to banking data, judicial information, bills, insurance data, military information, and many other types of valuable information which has to be protected, is managed and stored in databases [6]. Sometimes databases store another type of information that can be considered sensitive and which must be specially protected. This type of sensitive information usually refers to intimate or personal aspects of individuals, like personal identification data, medical data or even religious beliefs, ideologies or sexual tendencies. Information systems that manage databases with this type of information should be equipped with mechanisms that prevent non-authorised access to the information, thus guaranteeing people's rights to privacy and

fulfilling the Data Protection Laws existing, like in Spain, the *Organic Law for the Protection of Personal Data* [21].

The problems of security of information are increased as a consequence of the technological changes which are taking place: access to databases via the web, development of electronic commerce, advances in data warehouses and even the use of datamining techniques [30]. As a result of these advances, for example, systems are continually being attacked by hackers [4].

The previous arguments, indicating the existence of a serious risk to the data stored in databases, together with the results of studies such as those of [26] show that great areas of vulnerability exist in databases. These studies demonstrate the disinterest of organisations concerning security in their information systems (more than 70% of directors of developing businesses do not know when their security policies are checked) and how little of the budget is dedicated to security (more than 50% of companies dedicate less than 5% of their budget to security, and less than 5% dedicate more than 15% of the budget to security). In consequence, we need to commit our efforts to designing databases that are more secure.

All the initiatives carried out over recent years are very important and indicate the great interest aroused by the database security problem. The problem is that all the solutions that have been offered are only partial, isolated and unconnected ones, which do not resolve the problem of database protection globally, and which also do not address the problem at design level. For this reason we propose a methodological focus which allows us to design databases taking into account aspects of security from the earliest stages until the end of development. This focus should also be an extension of the methodologies and standards of modelling existing at the moment, since if this is not the case, the organisations really interested in the security of databases would have to make a great effort to adapt to a new methodology.

At the moment the standard for modelling is UML [5]. According to [20], UML can be used (via an appropriate process) in the design of databases. For this reason the idea of equipping UML models with security characteristics, in order to design secure databases, is attractive. Thus, a methodology of database design based on UML language, with the addition of security aspects, would allow us to design databases with the syntax and power of UML and with the new security characteristics ready to be used whenever the application has security requirements that demand them. This measure would allow us to fulfil the conditions imposed by [8] to systematically design information security in systems by integrating security requirements in the design and by providing the designers with models specifying security aspects. We would also be meeting the challenges posed by [11], which concern the unifying of security with software engineering at the same time as we unify security with the models of the system.

2 Secure Databases

Over the course of the last decade various initiatives have arisen relating to security in general, some of which relate directly to databases, like analysis techniques and risk management [19], access control techniques ([12]; [17]; [24]; [31]; [13]), methodologies for the development of security techniques [1], methods for designing security requirements ([27]; [28]) and even the occasional methodology for the development of secure databases ([18]; [25]). The ISO/IEC (International Organisation for Standardisation /International Electrotechnical Comission) has also prepared a guide for the management of security of information technologies [16]. In the following pages we review some of the most important aspects relating to security in databases.

2.1 Access Control Techniques

Access control is a mechanism by means of which we will ensure or attempt to ensure that only authorised personnel access our information resources, and that this personnel can only carry out the actions or activities authorised in accordance with their level of accreditation within the system [15].

Normally access control is achieved using authorisation rules which are a tuple with the following format **<s,o,a>**, indicating that the subject **s** can access the object **o** by performing action **a**. These three concepts which play an important part in the authorisation rules are defined as follows:

- The *subjects* are the entities who can be given authorised access to the objects, and although normally they are individual users they may also be groups of users, roles or even processes that are executed in the name of the users.
- The *objects* are the elements to which we want to control access. In the case of relational database systems, the objects may have varying granularity, that is to say we can access complete relationship, views, individual attributes, etc.
- The *actions* are the possible operations that may be performed, which in relational databases are normally selection, insertion, delete and update.

There exist several policies of access control, some of the most important being the following:

- Discretionary Access Control: This is based on the idea that the subjects access the objects on the basis of their identity and of authorisation rules that indicate what actions each subject may perform with the objects of the system. As indicated in [15], this access mechanism has some variants that need not be mutually exclusive. The most important are *positive and negative, strong and weak, explicit and implicit*, and *content-based*. This policy of access control has been widely used, but it has various characteristics, which make it less than suitable for current database systems, where data have their own confidentiality demands, independently of the subjects who want to access them. These

characteristics include the possibility of granting permits among subjects (which could give rise to violations of confidentiality) or the non-consideration of minimum requirements to access data depending on its characteristics.

- Mandatory Access Control. This is based on the model designed by Bell and LaPadula in [3] for operating systems, and consists of the classification of subjects and data at different levels of security such as: "unclassified", "confidential", "secret", and "top secret". Thus an item of data classified "top secret" could only be accessed by subjects who are classified as "top secret". The two basic rules which define the way of access to data by means of this policy, adapted to the paradigm of databases, indicate that a subject has reading access to an object if the accreditation level of the subject is greater than the confidentiality level of the object and that a subject has writing access to an object if the accreditation level of the subject is equal to the security level of the object, that is to say, subjects can only modify objects at their own level [15].
- Task-Based Access Control. This is suitable for distributed computation and for information processing activities with multiple access control points. This model deals with activities or tasks for representing authorisations. It uses time periods during which an authorisation remains valid. The main idea is to grant the correct quantity of permits, at the right moment, and only those that are strictly necessary, withdrawing permits once they become unnecessary [31].
- Role-Based Access Control. Permits are associated with roles and users are made members of these roles. The roles represent each functional group of the organisations, grouping together users with similar functions and responsibilities. By means of this mechanism it is very easy to carry out certain actions like changing users from one role to another or adding or eliminating specific permits as necessary. There are theories that affirm that mandatory and discretionary access control can be simulated via role-based access control [22]. We can find a general study and new investigation about this access control technique in [24] and [12].

2.2 Multilevel Databases

Multilevel databases support mandatory access control by considering different security levels to data and users. Each security level either forms part of a hierarchic category as discussed above (top secret, secret, confidential, unclassified) or may belong to a non-hierarchic category such as finance, sales, investigation, etc.). There are several relational prototypes like SeaView and Lock Data Views, some object oriented ones like SODA, SORION and Jajodia-Kogan models, and different commercial products, like Trudata, Secure Sybase, Trusted Oracle and Trusted Informix [15].

The objects which can be classified by security levels in a relational database according to their size or granularity could be the whole database, tables, tuplas or even attributes. If we can classify the tuplas by different levels of security, the problem of "polyinstantiation" arises, as we may have various tuples with the same

primary key and a different security level. The problem can be solved by taking as the primary key the outcome result of linking together the attributes which form the primary key with the security level of the tupla. An in-depth analysis of polyinstantiation can be found in [17] and [7].

Over the last few years a series of architectures has appeared to support access control in multilevel databases [15]. These are characterised among other things by the greater or lesser confidence in the security measures offered by the operating system, so as to provide the database management system with more or less security aspects.

3 Methodologies for Security Design and Database Development

There exist various methodologies for the development of databases, which generally consider the same stages: conceptual modelling, logical design and physical design ([9]; [2]). None of these methodologies contemplates security aspects in their stages, thus isolating security and relegating it to a back seat.

On the other hand various methods and methodologies have existed for security design. In [1] are classified these methodologies using the three following generations, none of which is for databases: checklist, engineering and logic transformation methods. One of the few methodologies for designing security in databases is the one appearing in [7], which considers phases very similar to those considered in the development of databases: preliminary analysis, security requirements and policy, conceptual design, logical design and physical design.

Some not very rigorous attempts have been made to unify the development of databases with security such as in the case of MOMT (Multilevel Object Modelling Technique) [18]. MOMT considers modifications in OMT (Object Modelling Technique) [23] to introduce security levels to the elements of class, dynamic and functional diagrams.

To tackle security in databases we need to develop a methodology (with a capital "M") [10], which includes as many aspects as possible, such as: persons, roles, techniques, tools, processes, activities, milestones, partial products, standards, quality measures, etc., and which considers security in all of them.

As a technique of this methodology, a specification language for security constraints has been developed, called OSCL (Object Security Constraint Language) [14]. This is based on OCL (Object Constraint Language) [32], which is the standard language of constraints of UML.

4 Extension of UML for Designing Secure Multilevel Databases

The aim of this section is to carry out modifications in the Use Cases and Class diagram of UML in order to be able to design secure multilevel databases. Currently it

is advisable to use object oriented modelling languages, like UML, to design databases [20]. This is due to the fact that, using class diagrams, it is possible to collect at a conceptual level all the semantics associated with a database, including other object oriented aspects not contemplated in the entity-relationship model, and which can be important for later transforming the conceptual model to the logical diagram.

4.1 Extension of the Use Cases Model

In use case models it will be necessary to represent those situations where a use case requires special attention concerning security. In these cases it will also be necessary to indicate which actors require specific accreditation to participate in these use case.

The extension of the use case model is carried out using stereotypes, creating *«safe UC»* which will be added below the use case symbol. For actors who participate in a 'secure' use case and who need specific accreditation the stereotype *«accredited-actor»* is created. The reason for using this representation is so that the diagrams will be easily transportable, an important condition for any extension of the UML language. See Figure 1.

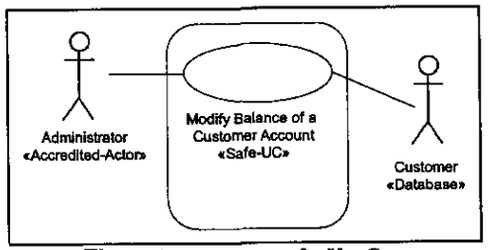

Fig. 1. New stereotypes for Use Cases

4.2. Extension of the Class Model

Several kinds of constraints exist: those inherent in a class model for secure databases, which are not represented explicitly, and those defined by the user which do need to be defined in the model. The principal inherent constraints in a model are the following:

- All the elements of a model will have a security level by default: the least restrictive.
- All the objects will have a security level included in the range of security levels of the class to which they belong.
- Associations will have a security level that must be equal to or superior to the level of classes it relates. In the same way, an instance by an association will have a security level which is higher than or equal to that of the association and that of the related objects.
- In a generalisation relationship, if the security level of the subclasses is omitted, they will inherit the security level of the superclass.
- As a general rule in the generalisation relationship, the security level of the subclasses must be greater than, or more restrictive than, the security level of the superclass, due to the inheritance mechanism.

In accordance with the classification carried out by [29], we could have the following types of security constraints defined by the user:

- Single: Those which classify a single element in a security level.
- Content based: Classifying part of the model depending on the value of a certain attribute.
- Event based: Those which represent the fact that the security level of some element of the model depends on some external event.
- Association based: Assign a level of security to an association.
- Aggregation: When an aggregation of elements is classified at a particular security level.
- Level based: Indicate that the security level of a specific element depends on the security level of another.
- Logical: When they specify implications.
- Meta-constraints: Constraints that classify constraints and meta-data.

The following UML extension mechanisms are used to represent the above:

1. Tagged values: This kind of extension is used to assign security levels to the elements of a class diagram, that is, to represent single constraints and constraints of associations. Some examples are shown in figure 2 where security levels are assigned to two classes, to their attributes and to the association between the classes.
2. Constraints: Constraints are specified using the language OSCL [14] which is specially designed to specify security constraints. Using this constraint language it is possible to specify both the inherent constraints in the model and all other constraints (except those based on events, which would be managed from an extended dynamic model). An example of this is seen in figure 3, which specifies the constraint that all the accounts in a bank with a balance of more than 1 million dollars will have the security level "top secret" and the rest "secret".

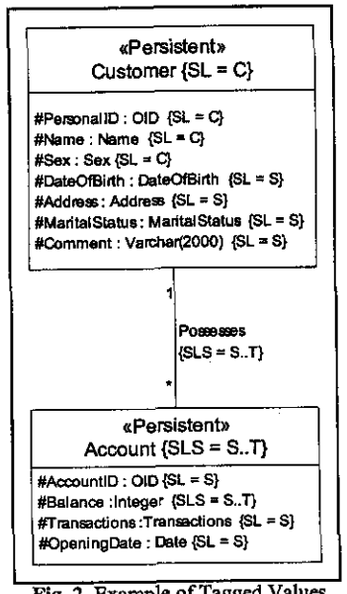

Fig. 2. Example of Tagged Values

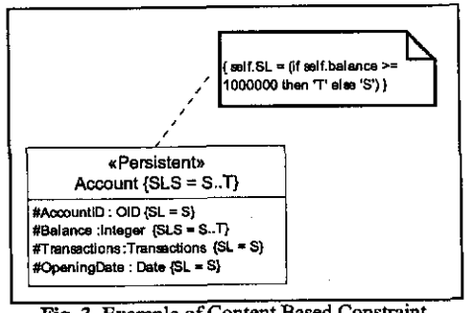

Fig. 3. Example of Content Based Constraint

The existence of tagged values which assign a security level or range of security levels to the attributes, classes, associations etc, can lead us to think about the necessity of modifying the metamodel, and to consider the properties "security level" and "range of security levels" as something inherent in the model elements, as are for example attributes for objects or the type of attribute for attributes. Thus all the elements of the model of classes will effectively have security levels although in many cases they will by defect have the least restrictive value (unclassified). The modifications of the metamodel are as follows:

- Two new types of data, named LEVEL and LEVELS are added (see figure 4). These will be the types of attributes that will be added to the elements of the

model to allow us to express security characteristics. The type LEVEL indicates the security level of an element of the model. The type LEVELS is considered since sometimes the elements of the model may have a level of security within a certain range, which will depend on determined circumstances, such as for example the value of certain attributes (content based constraint).

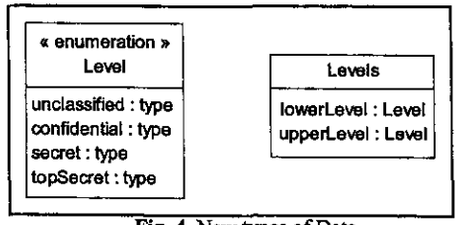

Fig. 4. New types of Data

- In the UML metamodel there is a class called 'ModelElement' which specialises in all the elements of the model. As we want to equip all the elements with security levels (classes, attributes, applications, operations, methods, associations and class associations, principally), the modification carried out, as we can see in Figure 5, involves adding security attributes to the class 'ModelElement' which will inherit all the elements of the model.

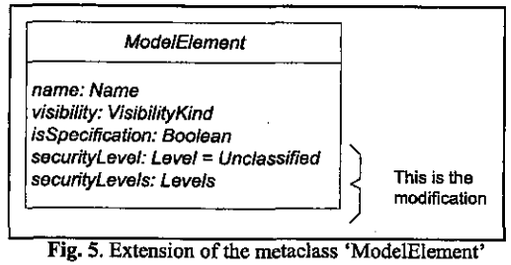

Fig. 5. Extension of the metaclass 'ModelElement'

In the following section we show an example of modelling in which all the extension mechanisms mentioned above are applied to express security characteristics and constraints.

5 Example of Application of the UML Extension

Let's suppose that we want to design a database to manage a hospital. To simplify the problem and to able to focus on security characteristics, we have considered only a few of the important aspects that would have to be modelled. A use case diagram for this problem would be as shown in figure 6.

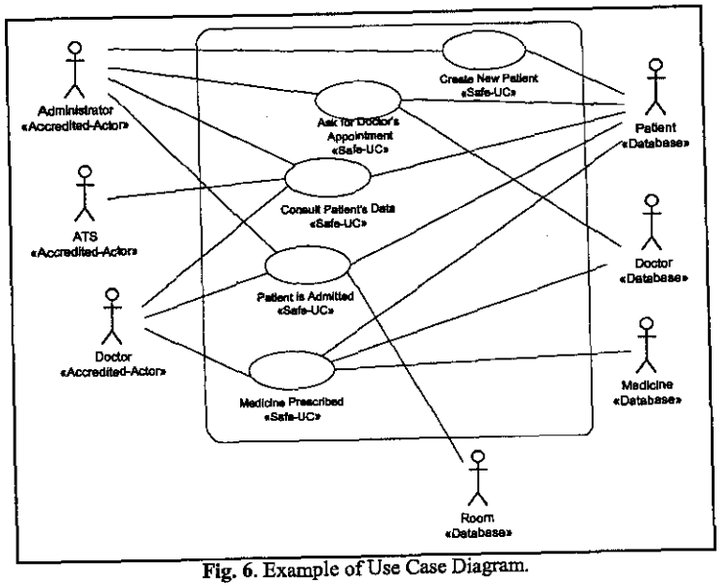

Fig. 6. Example of Use Case Diagram.

We can observe that in this example the stereotype «*database*» appears, indicating that these actors will store information that will be consulted or modified in these use case. The stereotype «*Safe-UC*» also appears which identifies 'safe' use cases. Finally, the stereotype that identifies actors who will need a certain level of accreditation to carry out the use case also appears.

In figures 7 and 8 various partial diagrams are shown, which belong to the class model which has been designed for this example.

In the diagram in figure 7 we can see the hierarchy of persons who will make up the hospital. We can observe that various classes of application of the extension mechanisms appear, such as the tagged values to indicate the level or range of security levels of the classes and attributes, and the constraints used to specify the security level of certain objects which depends on whether a condition on the value of one of its attributes is, or is not fulfilled. We can also see how inheritance does not only act for attributes and methods but also for security levels. For example, the class Doctor has a range of security levels from 'confidential' to 'Secret' that is inherited from the class 'Worker'. In this example we can observe some of the inherent constraints in the model discussed in section 4.2 relating to security levels between integrated classes in a generalisation relationship (subclasses must have a security level which is more restrictive than that of the superclass).

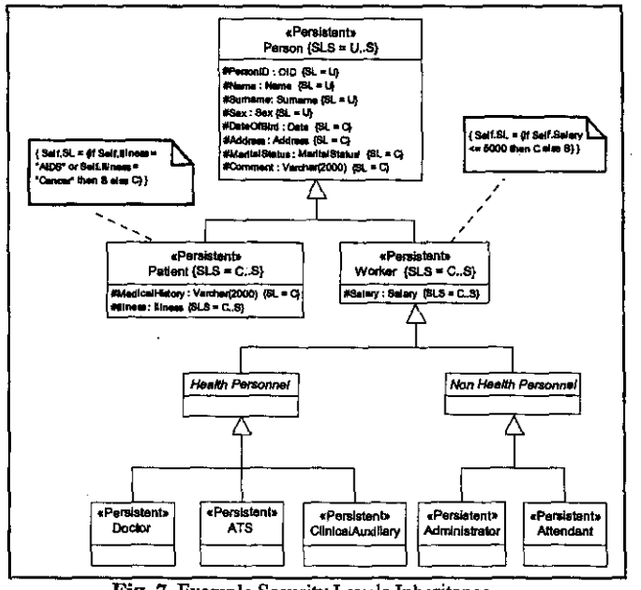

Fig. 7. Example Security Levels Inheritance.

In the diagram in figure 8 the security level of the associations has not been explicitly included since this is not relevant and because it would implicitly take on the range of levels of the related classes. We can see that some classes appear which do not have a specified security level. In these cases it is understood that the classes have no security requirements and, as such, their security level is decided by defect, that is to say, 'Unclassified'. Classes which have a range of security levels instead of just one security level, indicate that their security level depends on the level of one of their attributes (for example the case of the working personnel who depend on the salary level), or that they are association classes, where one of the classes which forms the association has a range of security levels instead of one definite level. Sometimes in the object diagram the security level of the attributes does not appear explicitly. In these cases the attributes inherit the security level specified for the class to which they belong.

6 Conclusions and Future Trends

Security in databases is a serious problem from various points of view, but above all from the point of view of confidentiality, due to the ever greater importance of the data stored in them. Many solutions have been put forward for the problem of security in databases, but they are all only partial solutions and do not solve the problem globally. In this article we have shown that the idea of integrating security design in

the design of databases is perfectly feasible and we have done so by extending the characteristics of the current modelling standard, UML.

Future work will be directed towards creating a complete methodology for designing secure databases which is based on these extended UML models and in the language designed for carrying out security specifications (OSCL). It will address all the factors related to the design process like tools, techniques, processes, activities, milestones, etc.

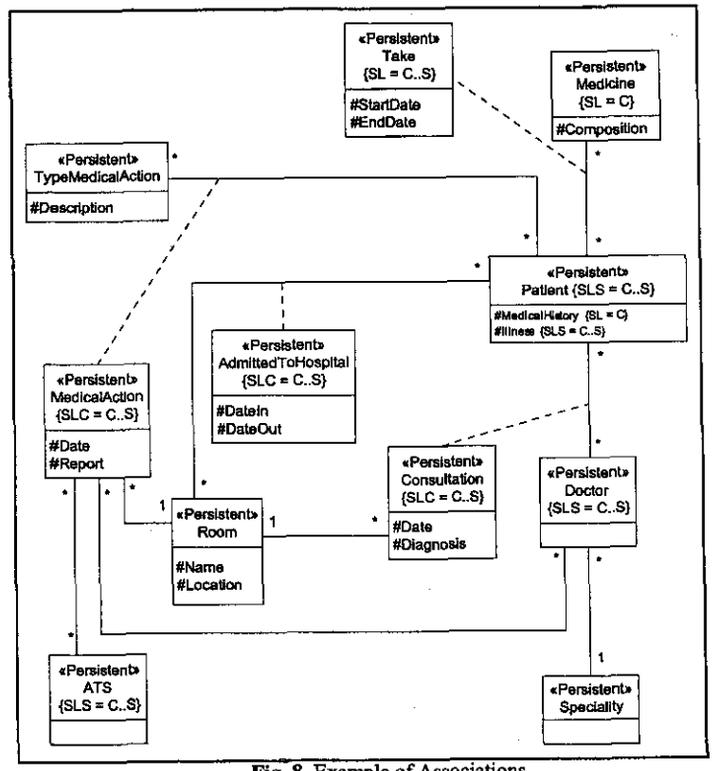

Fig. 8. Example of Associations

Acknowledgments

This research is part of the DOLMEN project supported by CICYT (TIC2000-1673-C06-06) and of the RETISBD project, supported by the Ministry of Culture and Education (TIC2000-1873-E), which is an special action into the National Program of the Information Technologies and Communications.

References

1. Baskerville, R. (1993). Information Systems Security Design Methods: Implications for Information Systems Development. *ACM Computing Surveys*. Vol. 25. Nº 4. December, pp. 375-415.
2. Batini, C., Ceri, S. and Navathe, S. (1992). *Diseño conceptual de bases de datos*. Addison-Wesley / Diaz de Santos.
3. Bell, D. and La Padula, L. (1973). Secure Computer Systems: Mathematical Foundation and Model. *Mitre Corp.*, Beldford, Mass.
4. Bellovin, S. (2001). Computer Security. An end State?. *Communications of the ACM*, Marzo. Vol. 44, Nº 3, pp. 131-132.
5. Booch, G., Rumbaugh, J. and Jacobson, I. (1999). *The Unified Modeling Language*, User Guide. Addison-Wesley, Reading, Mass.
6. Brinkley, D. and Schell, R. (1995*)*. What Is There to Worry About? An Introduction to the Computer Security Problem. *Information Security, An integrated collection of essays*. Eds.: Abrams, M., Jajodia, S. and Podell, H. IEEE Computer Society. California.
7. Castano, S., Fugini, M., Martella, G. and Samarati, P. (1994). *Database Security*. Addison-Wesley.
8. Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers. Boston/Dordrecht/London.
9. Connolly, T., Begg, C. and Strachan, A. (1998). *Database Systems*. Addison-Wesley.
10. Cockburn, A. (2000). Selection a Project's Methodology. *IEEE Software*. July-August. Pp. 64-71.
11. Devanbu, P. and Stubblebine, S. (2000). Software Engineering for Security: a Roadmap. *The Future of Software Engineering*. Ed: Finkelstein, A. Pp. 227-239.
12. Ferraiolo, D., Barkley, J. and Kuhn, R. (1999) A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and Systems Security*. Vol. 2, Nº 1, February 1999, pp. 34-64.
13. Fernández-Medina, E. and Piattini, M. (2001). Security in Database Systems: State of the Art. *Developing Quality Complex Database Systems: Practices, Techniques and Technologies*. Ed. Shirley Becker. Idea Group Publishing.
14. Fernández-Medina, E., Piattini, M. and Serrano, M. A. (2001). Specification of Security Constraints in UML. Proceedings of the *35th Annual 2001 International Carnahan Conference on Security Technology*. London.
15. Ferrari, E. and Thuraisingham, B. (2000). Secure Database Systems. *Advanced Databases: Technology Design*. Eds.: Piattini, M. and Díaz, O. Artech House. London.
16. ISO/IEC TR 13335 (1997*)*. *Information technology- Guidelines for the management of IT Security*.
17. Jajodia, S, Sandhu, R. and Blaustein, B. (1995*)*. Solutions to the Polyinstantiation Problem. *Information Security, An integrated collection of essays*. Eds.: Abrams, M., Jajodia, S. and Podell, H. IEEE Computer Society. California.

18. Marks, D., Sell, P. and Thuraisingham, B. (1996). MOMT: A multilevel object modeling technique for designing secure database applications. *Journal of Object-Oriented Programming*. Vol. 9. Nº 4, pp. 22-29.
19. MAP (1996). *Risk analysis and management methodology for information systems*. MAGERIT v.1.0
20. Muller, R. (1999). *Database Design for Smarties. Using UML for Data Modeling*. Morgan Kaufmann Publishers, inc. San Francisco, California.
21. *Organic Law 15/1999, of December, 13 of Personal Data Protection*. BOE núm 298, de 14 de December 1999.
22. Osborn, S., Sandhu, R. and Munawer, Q. (2000). Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. *ACM Transactions on Information and System Security*, Vol. 3, Nº 2, May, Pp: 85-106.
23. Rumbaugh, J., Blaha, M, Premerlani, W., Eddy, F. and Lorensen, W. (1991). *Object-Oriented Modeling and Design*. Prentice Hall, Englewood Cliffs.
24. Sandhu, R. and Bhamidipati, V. (1997). The URA97 model for role-based user-role assignment, in *Database Security XI: Status and Prospects*. Eds.: T.Y. Lin and S. Qian. Chapman and Hall, London, pp. 262-275.
25. Sell, P. and Thuraisingham, M.B. (1993). Applying OMT for Designing Multilevel Database Applications. Proceedings of the *Seventh IFIP Working Conference on Database Security*. Huntville, September.
26. SIC (2001). *Seguridad en Informática y Comunicaciones*. April. Nº 44. P. 6.
27. Smith, G.W. (1990). The Semantic Data Model for Security: Representing the Security Semantics of an Application. Proceedings of *the Sixth International Conference Data Engineering*, IEEE, pp. 322-329.
28. Smith, G.W. (1991). Modeling Security-Relevant Data Semantics. Proceedings of de *IEEE Transactions on Software Engineering*, Vol. 17. Nº 11, November, pp. 1195-1203.
29. Thuraisingham, B. and Ford, W. (1995). Security Constraint Processing in a Multilevel Secure Distributed Database Management System. *IEEE transactions on knowledge and data engineering*, Vol 7. Nº 2. April. Pp. 274-293.
30. Thuraisingham, B., Schlipper, L., Samarati, P., Lin, Jajodia, S. and Clifton, C. (1997). Security issues in data warehousing and data mining: panel discussion, *in Database Security XI: Status and Prospects*. (eds. T.Y. Lin and S. Qian), Chapman and Hall, London, pp. 3-16.
31. Tomas, R. and Sandhu, R. (1997). Task-based authorization controls (TBAC): a family of models for active and enterprise-oriented authorization management, *in Database Security XI: Status and Prospects*. (eds. T.Y. Lin and S. Qian), Chapman and Hall, London, pp. 166-181.
32. Warmer, J. and Kleppe, A. (1998). *The object constraint language*. Massachusetts. Addison-Wesley.

Java Application Signature, Integrity Verification and Server Authentication Mechanism in MExE Terminal

Jarkko Holappa

VTT Electronics, PL 1100, 90570 Oulu, Finland
Email: Jarkko.Holappa@vtt.fi

Abstract. The great success of the Internet is reaching the wireless world as third generation networks evolve and see the daylight in the near future. Many existing services and new ones are accessible via a mobile phone equipped with a web browser –style application. While providing the user familiar interface to a service, the developer and distributor of the service must also take security issues into serious consideration. If the service in question is a banking application, or another application that transfers or creates sensitive data, there must be a mechanism to provide adequate confidentiality and integrity for the service. After describing the problem area, Mobile Execution Environment (MExE) is introduced in this paper as an example of a standardized execution environment for third generation networks. Java as a platform for networked applications is presented along with the X.509 authentication framework and the secure socket layer (SSL) to provide an example of today's existing security technologies as used in MExE environment. Finally, a proposal for a Java application signature and integrity verification mechanism with service source authentication is defined. This mechanism is an addition to MExE's current security framework to overcome the original research problem: how to avoid an application downloading from untrustworthy or even hostile servers, and on the other hand, how to preserve unchangeability and confidentiality in executables to avoid eavesdropping and many kinds of losses caused by insecure mobile applications.

Keywords. security, standardization, mobile networks, middleware.

1 Introduction

Mobile networks offer the end-user many new ways of accessing services regardless of time and location. Third generation (3G) networks (UMTS, Universal Mobile Telecommunications System) and other wireless access technologies like Wireless LAN (WLAN) and Bluetooth provide more bandwidth and flexible use. To make the best use of these network technologies, a common service platform, i.e. middleware, is needed. The platform enables service developers to build services without any need to think of the underlying transportation level. A platform provides basic services and access to system resources to the application above it, and secondly, it also offers standardised application programming interfaces to service developers.