Lecture Notes in Computer Science

The LNCS series reports state-of-the-art results in computer science research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNCS has grown into the most comprehensive computer science research forum available.

The scope of LNCS, including its subseries LNAI and LNBI, spans the whole range of computer science and information technology including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post-proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on outstanding PhD work, research projects, technical reports, etc.)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS Online.

Detailed information on LNCS can be found at
www.springer.com/lncs

Proposals for publication should be sent to
LNCS Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany
E-mail: lncs@springer.com

ISSN 0302-9743

ISBN 978-3-540-74408-5

9 783540 744085

Lecture Notes in Computer Science
LNCS LNAI LNBI

springer.com

Lambrinoudakis • Pernul
Tjoa (Eds.)

LNCS 4657

LNCS
4657

Trust, Privacy and Security in Digital Business

TrustBus

Costas Lambrinoudakis
Günther Pernul
A Min Tjoa (Eds.)

Trust, Privacy and Security in Digital Business

4th International Conference, TrustBus 2007
Regensburg, Germany, September 2007
Proceedings

Springer

Lecture Notes in Computer Science 4657

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Costas Lambrinoudakis Günther Pernul
A Min Tjoa (Eds.)

Trust, Privacy and Security in Digital Business

4th International Conference, TrustBus 2007
Regensburg, Germany, September 4-6, 2007
Proceedings

Springer

Volume Editors

Costas Lambrinoudakis
Department of Information and Communication Systems Engineering
University of the Aegean
Karlovasi, 83200 Samos, Greece
E-mail: clam@aegean.gr

Günther Pernul
Department of Information Systems
University of Regensburg
Universitätsstrasse 31
D-93053 Regensburg, Germany
E-mail: guenther.pernul@wiwi.uni-regensburg.de

A Min Tjoa
Institute for Software Technology and Interactive Systems
Favoritenstrasse 9-11/188
Vienna University of Technology
A-1040 Vienna, Austria
E-mail: amin@ifs.tuwien.ac.at

Library of Congress Control Number: 2007933177

CR Subject Classification (1998): K.4.4, K.4, K.6, E.3, C.2, D.4.6, J.1

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-74408-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-74408-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12112197 06/3180 5 4 3 2 1 0

Table of Contents

Session 4: Authentication and Access Control

Session 5: Compliance and User Privacy

Session 6: Policy Management

Session 7: Security System Management

Session 8: Security and Trust

Analysis-Level Classes from Secure Business Processes Through Model Transformations

Alfonso Rodríguez[1], Eduardo Fernández-Medina[2], and Mario Piattini[2]

[1] Departamento de Auditoría e Informática, Universidad del Bio Bio Chillán, Chile
alfonso@ubiobio.cl
[2] ALARCOS Research Group, Information Systems and Technologies Department, UCLM-Soluziona Research and Development Institute, University of Castilla-La Mancha, Ciudad Real, Spain
{Eduardo.FdezMedina,Mario.Piattini}@uclm.es

Abstract. Nowadays, business processes (BP) are important in the maintenance of competitiveness within enterprises. Moreover, security is a crucial issue in business performance. In the last few years, the languages used for BP representation have been improved and new notations have appeared. Proposals for security requirement specifications at this high level of abstraction have also appeared. Nevertheless, these models have not been transformed into concrete models that can be used in a software development process. In our proposal, we will obtain analysis-level classes from a business process specification in which security requirements are included. Model transformations are within the scope of MDA and they are specified by using the QVT standard. Finally, we shall apply this approach to a typical health-care business process.

1 Introduction

In recent years, enterprise performance has been linked to the capability that they have to adapt themselves to the changes that arise in the market. In this context, business processes have become valuable resources that have been used to maintain competitiveness since they are the means through which an enterprise describes, standardizes, and adapts the way it reacts to certain types of business events, and how it interacts with suppliers, partners, competitors, and customers [19].

On the other hand, economic globalization, along with the intensive use of communications and information technologies, have caused enterprises to not only expand their businesses but also to increase their vulnerability. As a consequence, and with the increase in the number of attacks on systems, it is highly probable that sooner or later an intrusion may be successful [14].

Although the importance of business process security is widely accepted, the business analyst perspective in relation to security has hardly been dealt with to date. In [17] we introduced security representation into business processes. To do so, we extended the UML 2.0 Activity Diagram [13] by creating the BPSec profile, which allows us to capture security requirements expressed by the business analyst. Such a specification gives origin to a Secure Business Process.

Nowadays, model transformation has come under the scrutiny of the community of researchers and practitioners since it focuses upon solving the problems of time, cost

C. Lambrinoudakis, G. Pernul, A M. Tjoa (Eds.): TrustBus 2007, LNCS 4657, pp. 104–114, 2007.

and quality associated with software creation. The OMG (Object Management Group) proposal in relation to this fact is called MDA (Model-Driven Architecture) [12]. MDA is a framework for software development that allows the creation of models which are independent of technological implementation and QVT (Query/View/Transformation) [15], a standard for model transformation.

The MDA approach is composed of the following perspectives: (i) the Computation Independent viewpoint which focuses on the environment of the system, (ii) the Platform Independent viewpoint which focuses on the operation of a system whilst concealing the details necessary for a particular platform, and (iii) the Platform Specific viewpoint which combines the platform independent viewpoint with an additional focus on the detail of the use of a specific platform by a system [12].

In our proposal, we consider that an SBP (Secure Business Process) is a CIM (Computation Independent Model) that can be transformed into a PIM (Platform Independent Model). This transformation, carried out with QVT, leads to the generation of UML artifacts that can be used in a systematic and ordered process in software development. We have chosen the UP (Unified Process) [8, 16], which is composed of a set of activities necessary for transforming user requirements into a software system, due to the fact that it is a consolidated and successful software construction method [5].

The structure of the remainder of the paper is as follows: in Section 2, we will summarize the main issues concerning security in business processes together with our profile of a security requirement specification in business processes. In Section 3, we will present our proposal. Finally, in Section 4, we will put forward an example and in Section 5 our conclusions will be drawn.

2 Security in Business Process

In business process modeling, the main objective is to produce a description of reality, for example, the way in which a commercial transaction is carried out, in order to understand and eventually modify it with the aim of incorporating improvements into it. As a consequence, a notation must allow us to incorporate different perspectives which give place to various diagrams in which the rules, goals, objectives of the business and not only relationships but also interactions are shown [3].

In spite of the importance of security within business processes, the research works related to the security specifications carried out by business domain experts are; (i) scarce [1, 6, 7, 10], (ii) orientated towards transaction security [18], (iii) directly orientated towards information systems in general [21] or (iv) intended for security and software engineers [11].

However, at the present it is possible to capture security requirements at a high level, which are easily identifiable by those who model business processes, because: (i) the business process representation has improved in the UML 2.0 version, (ii) the security requirement will tend to have the same basic kinds of valuable and potentially vulnerable assets [4], and (iii) empirical studies show that it is common at the business process level for customers and end users to be able to express their security needs [9].

Therefore, we have approached the problem of including security in business processes [17] by extending the UML 2.0 Activity Diagram (UML 2.0-AD) which allows business analysts to specify security requirements. The proposed extension, which we have called BPSec, basically considers the graphical representation of security requirements, a non-limited list (see Figure 1) taken from the taxonomy proposed in [4].

Fig. 1. Icons to represent security requirements in BPSec

In our proposal we have used a padlock (Figure 1a) to represent security requirements in a standard way. The same symbol, the padlock, but with a twisted edge (Figure 1b) is used to represent a Security Requirement with Audit Register.

The relation between security requirement (dark-coloured) and the UML 2.0-AD is shown in Figure 2. «SecurityRole», «SecurityPermission», «G-AuditRegister», «NR-AuditRegister» and «SP-AuditRegister» stereotypes have been added with the purpose of complementing the security requirements specification.

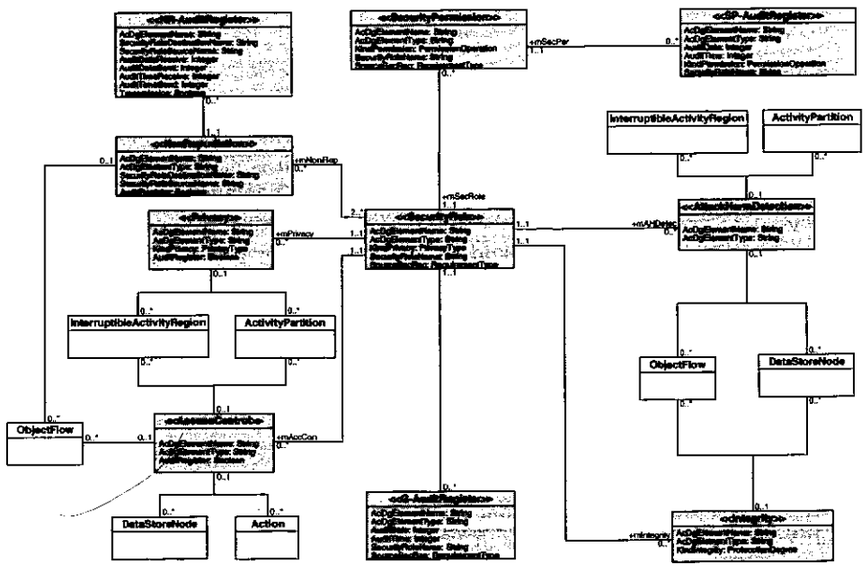

Fig. 2. BPSec and UML 2.0-AD Elements Model

Table 1. Security Requirements and UML 2.0-AD elements

Access Control: This corresponds to the limitation of access to resources by authorized users only. It implies the limitation of access to a set of resources that are considered important enough to be protected in a special way
– Action: This implies the definition of a secure role and security permissions associated with the action. The possible permissions are: Execution (default value) and CheckExecution.
– ActivityPartition: This implies the creation of a secure role and security permissions associated with actions, data store and data flows contained in the partition
– DataStoreNode: This implies the definition of a secure role and security permissions associated with the data store: The possible permissions are: Create, Delete, Read, and Update (default value)
– InterruptibleActivityRegion: This implies the creation of a secure role and security permissions associated with actions, data stores, and data flows contained in the region
– ObjectFlow: This implies the definition of a secure role and of security permissions associated with the object flow: The possible permissions are: SendReceive (default value) and CheckSendRecieve
Attack Harm Detection: This is defined as the detection, register and notification of an attempted attack or threat, whether it is successful or not. This requirement represents an attention signal covering the elements which are indicated.
– ActivityPartition: This implies the identification of a security role associated with the partition and the registration of the date and time of the produced accesses to the partition
– DataStoreNode: This implies the identification of a security role and the registration of the date and time of the accesses produced upon the data store
– InterruptibleActivityRegion: This implies the identification of a security role and the registration of the date and time when the accesses are produced in the region
– ObjectFlow: This implies the identification of the security roles (sender and receiver) related to the object flow and the registration of the date and time of the sending and reception of the flow
Integrity: This is related to the protection of components from intentional and non-authorized corruption. The integrity specification is valued as low, medium, and high. An integrity specification (at/any degree) is related to the importance of the information contained in the data store or data flow
– DataStoreNode: This implies the protection of the data store content. Together with this, the security role, date and time of all accesses to the data store are registered
– ObjectFlow: This implies the protection of the data contained in the object flow. Additionally, security roles involved in the flow, date and time of sending and reception are registered
Non Repudiation: This establishes the need to avoid the denial of any aspect of the interaction (e.g. message, transaction, transmission of data) so that any future problems (e.g. legal and liability) can be avoided.
– ObjectFlow: This implies flow protection. Additionally, the date and time of the sending and reception of the flow involved in the interaction are registered.
Privacy: This is related to conditions of information protection concerning a determined individual or entity, thus limiting access to sensitive information by non-authorized parties. From the point of view of the business analyst, the privacy specification implies the non-revelation (confidentiality) and non-storage (anonymity) of the information regarding a determined role.
– ActivityPartition: This implies the creation of a secure role associated with the partition
– InterruptibleActivityRegion: This implies the creation of a secure role associated with the region

The set of security requirements, which is not exclusive, is described in Table 1. The meaning of the relationship between each security requirement and UML 2.0-AD element is also described.

As a result of BPSec application, a Secure Business Process is obtained. The SBP description is used to obtain the analysis-level classes.

3 Analysis-Level Classes from Secure Business Processes

A business process built by a business analyst is not only useful in the specific business field, but is also very useful in a process of software construction, and can be used to obtain numerous kinds of system requirements. In our proposal, CIM2PIM transformations are aimed at obtaining useful artifacts in software development in such a way that automatically obtained analysis-level classes become part of an ordered and systematic process of software development.

In Figure 3, the basic aspects of our proposal are shown. At the top, we can see UML 2.0-AD and BPSec. In an MDA approach, an SBP description corresponds to a

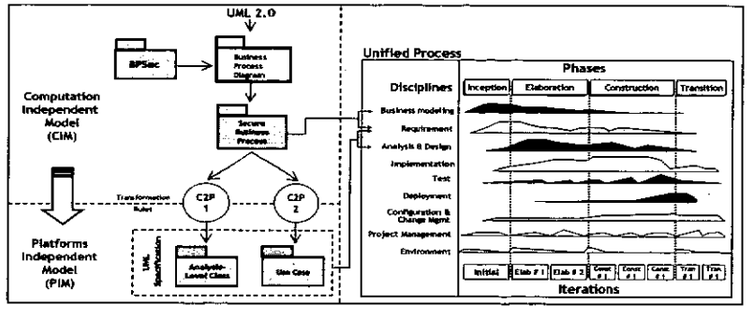

Fig. 3. An overview of our proposal

computation independent model (left-hand side). Through the application of a set of transformation rules, C2P_1, applied to SBP, it is possible to obtain a subset of the security analysis-level classes that facilitate the understanding of the problem. UP (right-hand side) is considered because the SBP description will be useful in the "Business Modeling" and "Requirement" disciplines, and the analysis-level classes complement the "Requirement" and "Analysis & Design" disciplines.

In our review of related literature, ranging from business processes to analysis-level class transformations, we have found two works that deal directly with this type of transformations. In the first [2], activity diagrams are transformed into analysis classes. This transformation is not performed automatically, and a previous version of UML 2.0 is used. In the second work [20], the software designer studies the business process model described with BPMN by extracting the UML classes which are later refined. The differences between these proposals and ours are that, firstly we use QVT for transformation specifications, secondly we pay special attention to security requirements, and finally we connect the result of transformations with a software development process.

In order to obtain a clearer view of the transformation rules, we shall present them in the following order: (i) QVT rules mapping general aspects of SBP which are not related to security specifications, (ii) QVT rules directly related to security specifications and finally, (iii) refinement rules that must be applied once analysis-level classes have been obtained as a consequence of QVT rule application.

QVT rules that are not related to security specifications can be used to obtain analysis-level classes derived from partitions, regions and the operations associated with the classes obtained. The QVT rules are described in Table 2.

If the QVT rules are applied to the security requirements described with BPSec, we can directly obtain analysis-level classes that have the requirement name. Indirectly, a class called SecurityRole is created and eventually SecurityPermission. The specification of audit register gives place to classes of the AuditRegister type associated with SecurityRole, SecurityPermission or a particular security requirement. The QVT specifications for these rules are described in Table 3.

Table 2. Mapping between Activity Diagrams and Class Diagrams elements

```
transformation ActivityDiagram2ClassDiagram
 top relation R1  // from Activity Partition to Analysis-Level Class
 {
   checkonly domain uml_ActivityDiagram ap:ActivityPartition {name = n}
   enforce domain uml_ClassDiagram c:Class {name = n}
   where { ap.containedNode → forAll(cn:Action|R4(cn))}
 }
 top relation R2  // from Interruptible Activity Region to Analysis-Level Class
 {
   checkonly domain uml_ActivityDiagram iar:InterruptibleActivityRegion {name = n}
   enforce domain uml_ClassDiagram c:Class {name = n}
   where { ap.containedNode → forAll(cn:Action|R4(cn))}
 }
 top relation R3  // from Data Store Node to Analysis-Level Class
 {
   checkonly domain uml_ActivityDiagram dsn:DataStoreNode {name = n}
   enforce domain uml_ClassDiagram c:Class {name = n}
 }
 relation R4 // from Action to Operation in Analysis-Level Class
 {
   checkonly domain uml_ActivityDiagram ac:Action {name = n, inPartition=ap}
   enforce domain uml_ClassDiagram op:Operation {name = n, ownerClass=c:Class{name=ap.name}}
 }
```

Table 3. Mapping between BPSec and Class Diagrams elements

```
transformation BPSec2ClassDiagram
top relation R5   // from Security Requirement to Analysis-Level Class
 {
  checkonly domain bpmn_BPSec sr:SecurityRequirement {requirementtype = n}
  enforce domain uml_ClassDiagram c:Class {name = n}
 }
top relation R6   // from Security Requirement to specific Analysis-Level Class
 {
  checkonly domain bpmn_BPSec sr:SecurityRequirement {requirementtype = n}
  enforce domain uml_ClassDiagram c:Class {name ="SecurityRole"}
 }
top relation R7   // Access Control to specific Analysis-Level Class
 {
  checkonly domain bpmn_BPSec ac:AccessControl {name = n}
  enforce domain uml_ClassDiagram c:Class {name ="SecurityPermission"}
 }
top relation R8   // from AccessControl to audit register Class
 {
  checkonly domain bpmn_BPSec ar:AuditRegister {requirementtype = n}
  enforce domain uml_ClassDiagram c:Class {name =nc}
      where { nc= if (n="AC") then "SP_AuditRegister" endif;}
 }
top relation R9   // from Integrity to generic audit register Class
 {
  checkonly domain bpmn_BPSec In:Integrity {name = n}
  enforce domain uml_ClassDiagram c:Class {name ="G_AuditRegister"}
 }
top relation R10   // from AttackHarmDetection to generic audit register Class
 {
  checkonly domain bpmn_BPSec Ad:AttackHarmDetection {name=n}
  enforce domain uml_ClassDiagram c:Class {name ="G_AuditRegister"}
 }
top relation R11   // from Privacy to generic audit register Class
 {
  checkonly domain bpmn_BPSec ar:AuditRegister {requirementtype = n}
  enforce domain uml_ClassDiagram c:Class {name =nc}
      where {nc= if (n="P") then "G_AuditRegister" endif;}
 }
top relation R12   // from NonRepudiation to audit register Class
 {
  checkonly domain bpmn_BPSec ar:AuditRegister {requirementtype = n}
  enforce domain uml_ClassDiagram c:Class {name =nc}
      where {nc= if (n="NR") then "NR_AuditRegister" endif;}
 }
```

Table 4. Refinement Rules for Analysis-Level Classes

RR1: InterruptibleActivityRegion Name is obtained by linking the ActivityPartition names in which the Region is contained
RR2: Composition relationships are obtained from top and middle ActivityPartitions
RR3: Relationships between classes derived from security requirements and the activity diagram element are obtained from the "BPSec and AD-UML 2.0-AD Elements Model" (Figure 2)
RR4: Relationships between classes derived from security requirements are obtained from "BPSec and AD-UML 2.0-AD Elements Model" (Figure 2)
RR5: Redundant specifications must be eliminated

4 Example

Our illustrative example (see Figure 4) describes a typical business process for the admission of patients to a health-care institution. In this case, the business analyst identified the following Activity Partitions: Patient, Administration Area (a top partition which is divided into the Admission and Accounting central partitions), and the Medical Area (divided into Medical Evaluation and Examination).

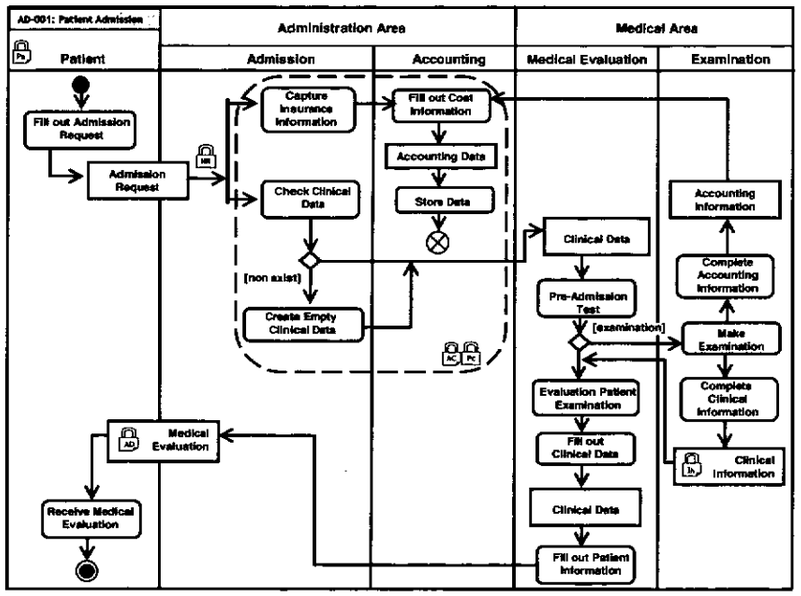

Fig. 4. Admission of Patients in a Medical Institution

The business analyst has considered several aspects of security. He/she has specified «Privacy» (anonymity) for the "Patient" Activity Partition, with the aim of preventing the disclosure and storage of sensitive information about Patients. «Nonrepudiation» has been defined for the control flow which goes from the action "Fill out Admission Request" to the actions "Capture Insurance Information" and "Check Clinical Data" with the aim of avoiding the denial of the "Admission

Table 5. QVT and refinement rules applied to Patient Admission Business Process

R1: Patient, Administration Area, Admission, Accounting, Medical Area, Medical Evaluation, and Examinations
R2: Region 01 (from InterruptibleActivityRegion)
R3: Admission Request, Accounting Data, Clinical Data, Accounting Information, Medical Evaluation and Clinical Information
R4: Patient [Fill out Admission Request and Receive Medical Evaluation]; Admission [Capture Insurance Information, Check Clinical Data, and Create Empty Clinical Data]; Accounting [Fill out Cost Information, and Store Data]; Administration Area [Capture Insurance Information, Check Clinical Data, Create Empty Clinical Data, Fill out Cost Information, and Store Data]; Medical Evaluation [Pre-Admission Test, Evaluation Patient Examinations, Fill out Clinical Data, and Fill out Patient Information]; Examinations [Complete Accounting Information, Carry out Examinations, and Complete Clinical information]; Medical Area [Pre-Admission Test, Evaluation Patient Examinations, Fill out Clinical Data, Fill out Patient Information, Complete Accounting Information, Carry out Examinations, and Complete Clinical information]; Region 01 [Capture Insurance Information, Check Clinical Data, Create Empty Clinical Data, Fill out Cost Information, and Store Data]
R5: Privacy (anonymity), NonRepudiation, Access Control and Privacy (confidentiality), Integrity (high), and AttackHarmDetection
R6: SecurityRole
R7: G-AuditRegister
R8: SP-AuditRegister
R9: G-AuditRegister
R10: G-AuditRegister
R11: NR-AuditRegister
RR1: AdmissionAccounting (name assigned to Region 01)
RR2: Administration Area composed of Admission and Accounting; Medical Area composed of Medical Evaluation and Examinations
RR3: Privacy → Patient ; Privacy → AdmissionAccounting; NonRepudiation → Admission Request; AccessControl → AdmissionAccounting; Integrity → Clinical Information; AttackHarmDetection → Medical Evaluation
RR4: Privacy → SecurityRole; AccessControl → SecurityRole → SecurityPermission → SP-AuditRegister; Integrity → SecurityRole → G-AuditRegister; AttackHarmDetection → SecurityRole → G-AuditRegister
RR5: SecurityRole → G-AuditRegister redundancies must be eliminated

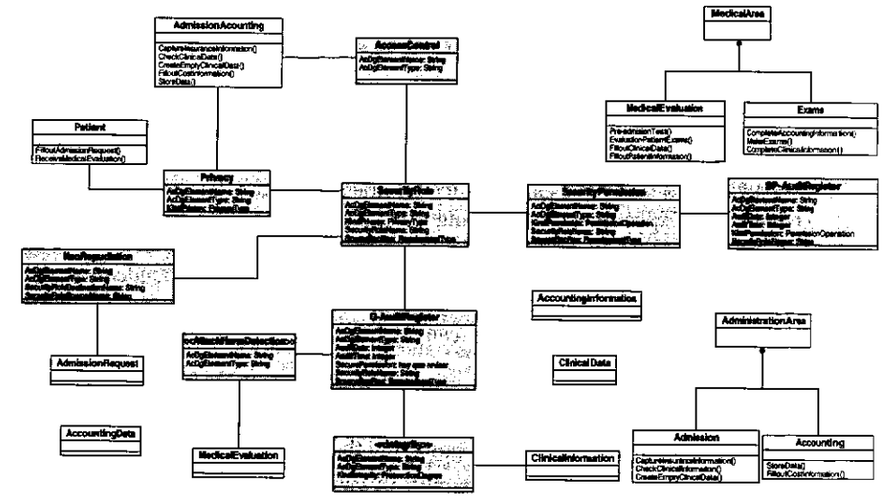

Fig. 5. Analysis-Level Class from Patient Admission

Request" reception. «AccessControl» and «Privacy» (confidentiality) has been defined for the Interruptible Activity Region. A «SecurityRole» can be derived from this specification. Admission/Accounting will be one role. All objects in an interruptible region must be considered for permission specification. The Access Control specification has been complemented with an audit requirement. This implies that it must register information about the security role and security permissions.

Integrity requirement (high) has been specified for the "Clinical Information" Data Store and finally, the business analyst has specified Attack Harm Detection for the "Medical Evaluation" Data Store, so that all events related to the attempt or success of attacks or damages are registered.

The attainment of analysis-level classes through the application of the transformations defined with the QVT rules (R) and the Refinement Rules (RR) are described in Table 5.

Figure 5 shows a graphical representation of the analysis-level classes which are presented in Table 5. This figure is enriched since, after the application of the QVT rules, we have named the region, we have incorporated the relationship between the elements in the class diagrams and we have eliminated the redundancies. In addition, the analysis-level class derived from the security requirement specification is shown in the dark-coloured areas.

5 Conclusion and Ongoing Work

One way in which to confront the problem of security consists of incorporating it into the business process specifications at an early stage. At that level, it is possible to capture security requirements which take into account the business analyst's viewpoint. In previous works, we have proposed an extension of the UML 2.0 Activity Diagram through which it is possible to specify security requirements at a high level of abstraction.

In addition, models transformation has come to the attention of the community of researchers and practitioners owing to the fact that it has the aim of solving the problems of time, cost and quality associated with software creation.

In this paper, we have presented a model transformation by using the MDA approach with QVT specification. By using a Secure Business Process specification, which is considered to be a Computation Independent Model, we have obtained a set of analysis-level classes, which are considered to be a Platform Independent Model. The analysis-level class obtains a subset of all classes which are necessary for describing a problem, and the SBP can be used in a well-known software development process.

Ongoing work is orientated towards enriching transformations in order to make it possible to obtain more complete models of analysis-level classes. Together with this, our future work also has the purpose of optimizing the prototype that we have created to carry out the transformations with the aim of improving specification reuse and documentation.

Acknowledgments. This research is part of the following projects: DIMENSIONS (PBC-05-012-1), and MISTICO (PBC06-0082) both partiality supported by the FEDER and the "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha", Spain, COMPETISOFT (506PI287), granted by CYTED and ESFINGE (TIN2006-15175-C05-05/) granted by the "Dirección General de Investigación del Ministerio de Ciencia y Tecnología", Spain.

References

1. Backes, M., Pfitzmann, B., Waider, M.: Security in Business Process Engineering. International Conference on Business Process Management (BPM). In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 168–183. Springer, Heidelberg (2003)
2. Barros, J.P., Gomes, L.: From Activity Diagrams to Class Diagrams. In: Workshop Dynamic Behaviour in UML Models: Semantic Questions In conjunction with Third International Conference on UML, York, UK (2000)
3. Castela, N., Tribolet, J., Silva, A., Guerra, A.: Business Process Modeling with UML. In: 3st. International Conference on Enterprise Information Systems, Setubal, Portugal, vol. 2, pp. 679–685 (2001)
4. Firesmith, D.: Specifying Reusable Security Requirements. Journal of Object Technology 3(1), 61–75 (2004)
5. Fuggetta, A.: Software process: a roadmap. In: ICSE 2000, 22nd International Conference on Software Engineering, Future of Software Engineering, Limerick Ireland pp. 25–34 (2000)
6. Herrmann, G., Pernul, G.: Viewing Business Process Security from Different Perspectives. In: 11th International Bled Electronic Commerce Conference, 1998, Slovenia, pp. 89–103 (1998)
7. Herrmann, P., Herrmann, G.: Security requirement analysis of business processes. Electronic Commerce Research 6(3-4), 305–335 (2006)
8. Jacobson, I., Booch, G., Rumbaugh, J.: El proceso unificado de desarrollo de software, p. 464 (2000)
9. Lopez, J., Montenegro, J.A., Vivas, J.L., Okamoto, E., Dawson, E.: Specification and design of advanced authentication and authorization services. Computer Standards & Interfaces 27(5), 467–478 (2005)
10. Maña, A., Montenegro, J.A., Rudolph, C., Vivas, J.L.: A business process-driven approach to security engineering. In: 14th. International Workshop on Database and Expert Systems Applications (DEXA), Prague, Czech Republic, pp. 477–481 (2003)
11. Maña, A., Ray, D., Sánchez, F., Yagüe, M. I.: Integrando la Ingeniería de Seguridad en un Proceso de Ingeniería Software, VIII Reunión Española de Criptología y Seguridad de la Información, RECSI, Madrid. Españ, pp. 383–392 (2004)
12. Object Management Group; MDA Guide Version 1.0.1. (2003), http://www.omg.org/docs/omg/03-06-01.pdf
13. Object Management Group; Unified Modeling Language: Superstructure, version 2.0, formal/05-07-04 (2005), http://www.omg.org/docs/formal/05-07-04.pdf
14. Quirchmayr, G.: Survivability and Business Continuity Management. In: ACSW Frontiers 2004 Workshops, Dunedin, New Zealand, pp. 3–6 (2004)
15. QVT, Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, OMG Adopted Specification ptc/05-11-01, p. 204 (2005)
16. Rational Software, Rational Unified Process, Best Practices for Software Development Teams, p. 21 (2001)
17. Rodríguez, A., Fernández-Medina, E., Piattini, M.: Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes. In: Fischer-Hübner, S., Furnell, S., Lambrinoudakis, C. (eds.) TrustBus 2006. LNCS, vol. 4083, pp. 51–61. Springer, Heidelberg (2006)

18. Röhm, A.W., Herrmann, G., Pernul, G.: A Language for Modelling Secure Business Transactions. In: 15th. Annual Computer Security Applications Conference, Phoenix, Arizona, pp. 22–31 (1999)
19. Roser, S., Bauer, B.: A Categorization of Collaborative Business Process Modeling Techniques. In: 7th IEEE International Conference on E-Commerce Technology Workshops (CEC 2005), Munchen, Germany, pp. 43–54 (2005)
20. Rungworawut, W., Senivongse, T.: Using Ontology Search in the Design of Class Diagram from Business Process Model, Enformatika, Transactions on Engineering. Computing and Technology 12, 165–170 (2006)
21. Tryfonas, T., Kiountouzis, E.A.: Perceptions of Security Contributing to the Implementation of Secure IS, Security and Privacy in the Age of Uncertainty, IFIP TC11 18th International Conference on Information Security (SEC2003), Athens, Greece, vol. 250, pp. 313–324 (2003)