

Capturing Security Requirements in Business Processes Through a UML 2.0 Activity Diagrams Profile

Alfonso Rodríguez¹, Eduardo Fernández-Medina², and Mario Piattini²

¹ Departamento de Auditoría e Informática, Universidad del Bío Bío,
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. Security has become a crucial aspect for the performance of present organizations since the protected object is the mission of them. Therefore, the management approach oriented to business processes has been a good answer for the current scenarios, changing and complex, where organizations develop their task. Both subjects form a basic requirement to reach not only the mission but also the organizational objectives in a strongly connected global economy. In this work, we will show a microprocess through which it is possible to specify and refine security requirements at a high level of abstraction, in a way that they can be incorporated into the development of a software system. In addition, an extension of UML 2.0 activity diagrams will be presented through which it is possible to identify such requirements.

1 Introduction

The new business scene, where there are many participants and an intensive use of communications and information technologies, implies that enterprises not only expand their businesses but also increase their vulnerability. As a consequence, with the increase of the number of attacks on systems, it is highly probable that sooner or later an intrusion can be successful [22].

Regardless of the importance of the security notion for companies, this is often neglected in business process models, which usually concentrate on modeling the process in a way that functional correctness can be shown [3] mainly due to the fact that the expert in the business process domain is not an expert in security [10]. Typically, security is considered after the definition of the system. This approach often leads to problems, which most of the times are translated into security vulnerabilities [19], which clearly justify the need of increasing the effort in the pre-development phases, where fixing the bugs is cheaper [16].

If we consider that empirical studies show that it is common at the business process level that customers and end users are able to express their security needs [16], then it is possible to capture at a high level, security requirements easily identifiable by those who models business processes. Besides, requirements specification usually results in

a specification of the software system which should be as exact as possible [2], since, effective business process models facilitate discussions among different stakeholders in the business, allowing them to agree on the key fundamentals and to work towards common goals [6].

In our proposal, we consider the definition of a microprocess that complements the requirements capture defined in the Unified Software Development Process [11] and we have defined a UML 2.0 activity diagrams profile to capture security requirements.

The structure of the rest of the paper is the following: in Section 2, we will summarize the main issues about security in business processes. In Section 3, we will present a brief overview of UML 2.0 activity diagrams and profiles. In Section 4, we will propose a microprocess for the security requirements specification and a UML 2.0 profile that allows the business analyst to carry out this task. Finally, in Section 5, we will present an example and in Section 6 our conclusion will be drawn.

2 Security in Business Process

In spite of the importance of security for business processes, we have found out two problems. The first one is that modeling has not been adequate since, generally, those who specify security requirements are requirements engineers that have accidentally tended to use architecture specific restrictions instead of security requirements [7]. And in the second place, security has been integrated into an application in an ad-hoc manner, often during the actual implementation process [3], during the system administration phase [15] or it has been considered like outsourcing [18].

Moreover, capturing the security requirements of a system is a hard task that must be established at the initial stages of system development, and business spruces offer a view of business structure that is very suitable as a basis for the elicitation and specification of security requirements. Business process representations may in this way present in all stages of system development different levels of abstraction appropriate for each stage [16]. Consequently, we believe that business analysts can integrate their view on business security into the business process perspective. In addition, security requirements since any application at the highest level of abstraction will tend to have the same basic kinds of valuable and potentially vulnerable assets [8].

In the review of related works, we have had the possibility to check that not only in those works directly referring to security regarding business processes [3, 10, 17, 23, 24, 27] but also in those that have to do with security and information systems [1, 2, 4, 12, 15, 19, 25, 28], security specifications made by the business analyst are absent. Moreover and in spite of the fact that in some of these works, UML is used for security specifications, none of them use the activity diagrams available in UML 2.0.

3 UML 2.0 Activity Diagrams and UML 2.0 Profiles

Activity diagrams are the UML 2.0 elements used to represent business processes and workflows [13]. In UML previous versions, expressivity was limited and this fact confused users that did not use the orientation to objects as an approach for modeling. Now, it is possible to support flow modeling across a wide variety of domains [5]. An

activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions [20]. The graphical notation of an activity is a combination of nodes and connectors that allow us to form a complete flow.

On the other hand, the Profiles package contains mechanisms that allow meta-classes from existing metamodels to be extended to adapt them for different purposes. The profiles mechanism is consistent with the Meta Object Facility (MOF) [20]. UML profiles consist of Stereotypes, Constraints and Tagged Values. A stereotype is a model element defined by its name and by the base class to which it is assigned. Constraints are applied to the stereotype with the purpose of indicating limitations (e.g. invariants). They can be expressed in natural language, programming language or through Object Constraint Language (OCL). Tagged values are additional meta-attributes assigned to a stereotype, specified as name-value pairs.

Research works related to UML 2.0 profiles and business processes refer to aspects of the business such as Customer, kind of Business Process, Goal, Deliverable and Measure [14], Data Warehouse and its relation to business process dynamic structures [26] or they add semantics to the activities considering organizational aspects that allow us to express resource restrictions during the execution of an activity [13].

4 Microprocess and UML 2.0 Profile for Security Requirements

Requirements specification is a stage that has been taken into account in the most important software construction models such as the traditional waterfall model, the prototype construction, the incremental model, and the spiral model, among others. [21]. In these models, it is considered a stage in which we should obtain the system requirements either from the client or from the interested people in order to start the software construction from that point.

Our proposal studies a microprocess that complements the specification of the system context defined in the Unified Process [11] paying special attention to security requirements capture. To do so, a UML 2.0 activity diagrams profile is proposed.

4.1 SeReS4BP Microprocess

We have considered the use of the Unified Software Development Process stated by Jacobson, Booch y Rumbaugh (2000) since it is a quite consolidated and successful software construction method [9]. This process is composed by a set of activities that allow us to transform a user's requirements into a software system.

In the Unified Process, requirements capture is mainly done during the inception and elaboration stages. The objective of this task is to make a good enough description of the system's requirements (conditions and capabilities that must be fulfilled by the system) to determine what the system must or must not do. To do so, it is considered the performance of an enumeration of the requirements of the candidates, the understanding of the system context, and the capture of both functional and non functional requirements.












Stage	Worker	Tool	Artifact
Construction	 Business Analyst	 UML 2.0 Activity Diagrams	 Business Process Model
Security requirements incorporation	 Business Analyst	 BPsec 1.0	 Business Process Model with Security Specifications (preliminary specifications)
Refining	 Security Expert	 UML 2.0 Activity Diagrams and BPsec 1.0	 Business Process Model with Security Specifications (final specifications)
	 Business Analyst		 Business Process Repository with Security Specifications (update)

Fig. 1. Complete view of the SeReS4BP microprocess

The security requirements specified in the business process can be perfectly linked to the Unified Process. To do so, we propose to complement the task “to understand the system context” with specifications of the domain built by the business analyst. Our proposal is a microprocess that considers the necessary activities that allow us to specify requirements (particularly, security requirements) taking into account the business analyst’s perspective. This microprocess is called SeReS4BP (Security Requirement Specification for Business Process). Figure 1 shows us a view of the main activities performed in this microprocess and Table 1 shows us a details description.

Table 1. SeReS4BP activities

<p>Stages: <i>Construction:</i> whose objective is the business process model construction. To reach this objective, the UML 2.0 activity diagram must be used. <i>Security requirements incorporation:</i> this stage consists of incorporating security requirements, from the business analyst viewpoint, into the business process model that was specified in the previous stage. <i>Refining:</i> This stage corresponds to the review and complementing of the security specifications that have been incorporated into the business process. At this stage, the business analyst and the security expert work together. The specifications that will be finally incorporated into the business process will be agreed at this stage.</p> <p>Workers: <i>Business Analyst:</i> he/she will be responsible for the specifications related to the business itself as well as for incorporating, from his/her point of view, security requirements into the specifications considering a high level of abstraction. <i>Security Expert:</i> he/she will be the responsible for refining the security specifications indicated by the business analyst. Such refining considers the verification of the specifications validity and complementation.</p> <p>Tools: <i>UML 2.0 Activity Diagrams</i> for the business process specification. <i>BPsec 1.0</i> for security requirements specifications</p> <p>Artifacts: <i>Business process model:</i> This artifact is the result of the construction stage. It contains the business process specifications and it can be built using UML. It does not contain security specifications. <i>Business Process Model with Security Specifications.</i> This artifact is the result of the stages of incorporation of security requirements and refining. The first stage contains security preliminary specifications that, after refining, will be converted into definitive security specifications. <i>Business Process Repository</i> that contains security specifications. This repository is composed of a set of business processes that have security requirements already incorporated. This repository must be updated with the business process resulting from the refining stage.</p>
--

4.2 BPsec Version 1.0 for Modeling Security Requirements in Business Processes

In this section, we will present the main aspects of our profile for representing security requirement in business process. Our proposal allows business analysts to specify security requirements in the business process by using activity diagrams. We have considered the security requirements identified in the taxonomy proposed in [8]. Later on, these requirements will be transformed, by the security experts, into technical specifications including all necessary details for their implementation.

Our Profile will be called BPsec (Secure Business Process) and will be represent as a UML Package. This profile will incorporate new data types, stereotypes, tagged value and constrains. In Figure 2, a high level view is provided.

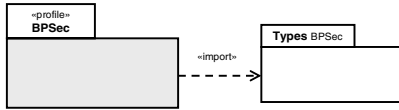


Fig. 2. High level view of BPsec Profile

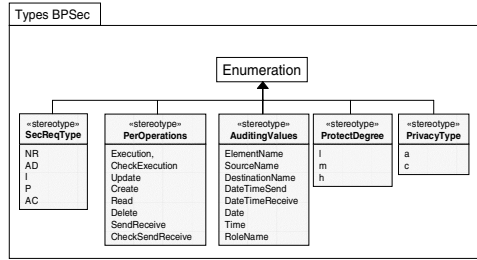


Fig. 3. Value associated to new data type

In addition we need the definitions of some new data types to be used in tagged value definitions. In Table 2, we will show the new data type stereotypes definitions.

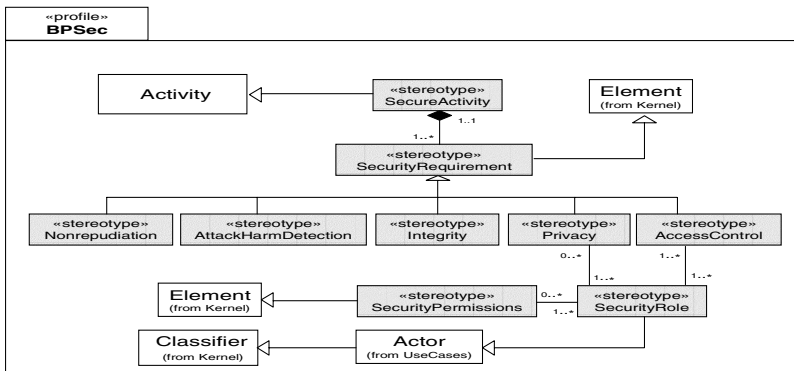


Fig. 4. New Stereotypes

In Figure 3, we can observe the values associated to each one of the necessary type. All the new type must be considered when the business analysts to specify security requirements in business process. We have defined a package that includes all stereotypes that will be necessary in our profile. In Figure 4 we show the stereotypes (in dark) for Secure Activity specifications.

Table 2. New data types

Name	Description	Values associated
SecReqType	It represents a type of security requirement. It must be specified for Non Repudiation, Attack/Harm Detection, Integrity, Privacy or Access Control.	NR, AD, I, P, AC
PerOperations	It is an enumeration for possible operations over objects in activity diagrams. These operations are related to permissions granted over the object	Execution, CheckExecution, Update, Create, Read, Delete, SendReceive, CheckSendReceive
ProtectDegree	It is an abstract level that represents criticality. This degree can be low (l), medium (m) or high (h).	l, m, h
PrivacyType	It consists of anonymity (a) or confidentiality (c).	a, c
AuditingValues	It represents different security events related to the security requirement specification in business processes. They will be used in later auditing	ElementName, SourceName, DestinationName, DateTimeSend, DateTimeReceive, Date, Time, RoleName

A Secure Activity is a stereotype derived from Activity. «SecureActivity» is strongly associated with security requirements stereotypes. «SecurityRequirement» has a composition relationship with «SecureActivity». The proposed notation for

«SecurityRequirement» must be complemented by adding it letters that will allow us to identify the type of requirement that is specified.

The stereotypes derived from «SecurityRequirement» can be *added* to activity diagrams elements. Any security requirement can be added to activity diagram elements (see Table 3). For example, an «Integrity» requirement can be specified over data store, control flow or object flow.

«SecurityRole» and «SecurityPermissions» are related in different ways; because both can be *obtained* from the UML 2.0 element of activity diagrams (see Table 3). For example, «SecurityRole» can be obtained from activities, partitions or regions specifications, but it is not specified in an explicit way over these activity diagrams elements. «SecurityPermission» is a special case, because, permissions depending on each activity diagram element which they are related to. For example, for Actions object, Execution or CheckExecution operations must be specified (see Table 5).

Table 3. Security Requirements and Activity Diagram Elements







Stereotypes for secure activity specification	UML 2.0 element for containment in activity diagrams					
	Activity	Activity Partition	Interruptible Activity Region	Action	Data Store Node	Object Flow
Nonrepudiation						✓
AttackHarmDetection	✓	✓	✓	✓	✓	✓
Integrity					✓	✓
Privacy		✓				
AccessControl	✓	✓	✓			
Security Role	✓	✓	✓			
SecurityPermissions				✓	✓	✓

In Table 4 we show the stereotypes for secure activity specifications extensively. Each stereotype specification contains: name, base class, description, notation (optional), constrains and tagged values (optional).

Table 4. Stereotypes specifications for security requirement

Name	SecureActivity
Base Class	Activity
Description	A secure activity contains security specification related to requirements, role identifications and permissions
Constrains	It must be associated at least with one SecurityRequirement context SecureActivity inv: self.SecurityRequirement->size()>=1
Name	SecurityPermission
Base Class	Element (from Kernel)
Description	It contains permission specifications. A permissions specification must contain details about the objects and operations involved
Constrains	It must be associated with security role specification context SecurityPermission inv: self.SecurityRole ->size()>= 1 It must be associated with Actions, DataStoreNode or ObjectFlow context SecurityPermissions inv: self.Actions.size+self.DataStoreNode.size+self.ObjectFlow.size=1 It must be specified such as Objects and Operations pairs. context SecurityPermissions inv: if self.Actions->size()=1 then self.SecPerOperations="Execution" or self.SecPerOperations="Checkexecution" endif if self.Datastorenode->size()=1 then self.SecPerOperations="Update" or self.SecPerOperations="Create" or self.SecPerOperations="Read" or self.SecPerOperations="Delete" endif if self.Objectflow->size()=1 then self.SecPerOperations="Sendreceive" or self.SecPerOperations="Checksendreceive" endif
Tagged Values	SecurityPermissionOperation: SecPerOperations
Name	SecurityRole
Base Class	Actor (from UseCases)
Description	It contains a role specifications. This roles must be obtained from access control and/or privacy specifications

Table 4. (continued)

Constrains	The role in the security role stereotype can be derived from: Activity, ActivityPartition and/or InterruptibleActivityRegion It must be associated with an access control specification and can be associated with privacy and security permissions context SecurityRole inv: self.AccessControl -> size() >= 1 context SecurityRole inv: self.Privacy -> size() >= 0 context SecurityRole inv: self.SecurityPermission -> size() >= 0	
Name	SecurityRequirement	Notation 
Base Class	Element	
Description	Abstract class containing security requirements specifications. Each security requirement type must be indicated in some of its subclasses	
Constrains	A security requirement must be associated with a secure activity context SecurityRequirement inv: self.SecureActivity -> size()=1 The notation must be completed in the subclass specification for each security requirement. It must be used one security requirement type.	
Tagged Values	SecurityRequirementType: SecReqType	
Name	Nonrepudiation	Notation 
Base Class	SecurityRequirement	
Description	It establishes the need to avoid the denial of any aspect of the interaction. An auditing requirement can be indicated in Comment	
Constrains	It can be only specified in the diagram elements indicated in Table 3.	
Tagged Values	AvNr: AuditingValues context Nonrepudiation inv: self.AvNr="ElementName" or self.AvNr="SourceName" or self.AvNr="DestinationName" or self.AvNr="DateTimeSend" or self.AvNr="DateTimeReceive"	
Name	AttackHarmDetection	Notation 
Base Class	SecurityRequirement	
Description	It indicates the degree to which the attempt or success of attacks or damages is detected, registered and notified. An auditing requirement can be indicated in Comment	
Constrains	It can be only specified in the diagram elements indicated in Table 3.	
Tagged Values	AvAD: AuditingValues context AttackHarmDetection inv: self.AvAD="ElementName" or self.AvAD="Date" or self.AvAD="Time"	
Name	Integrity	Notation 
Base Class	SecurityRequirement	
Description	It establishes the degree of protection of intentional and non authorized corruption. The elements are protected from intentional corruption. An auditing requirement can be indicated in Comment.	
Constrains	It can be only specified in the diagram elements indicated in Table 3. The Protection Degree must be specified by adding a lower case letter according to PDI tagged value.	
Tagged Values	PDI : ProtectDegree AvI: AuditingValues context Integrity inv: self.AvI="ElementName" or self.AvI="Date" or self.AvI="Time"	
Name	Privacy	Notation 
Base Class	SecurityRequirement	
Description	It indicates the degree to which non authorized parts are avoided to obtain sensitive information. An auditing requirement can be indicated in Comment.	
Constrains	It can be only specified in the diagram elements indicated in Table 3. A privacy requirement has one security role specification context Privacy inv: self.SecurityRole -> size() = 1 The Privacy Type must be specified adding a lower case letter according to Pv tagged value. If privacy type is not specified then anonymity and confidentiality are considered.	
Tagged Values	Pv: PrivacyType AvPv: AuditingValues context Privacy inv: self.AvPv="RoleName" or self.AvPv="Date" or self.AvPv="Time"	
Name	AccessControl	Notation 
Base Class	SecurityRequirement	
Description	It establishes the need to define and/or intensify the access control mechanisms (identification, authentication and authorization) to restrict access to certain components in an activity diagram. An auditing requirement can be indicated in Comment.	
Constrains	It can be only specified in the diagram elements indicated in Table 3. It is valid only if it is specified at least one security role. context AccessControl inv: self.SecurityRole -> size() >= 1	
Tagged Values	AvAC: AuditingValues context AccessControl inv: self.AvAC="RoleName" or self.AvAC="Date" or self.AvAC="Time"	

5 Example

Our illustrative example (see Figure 5) describes a typical business process for the admission of patients in a health-care institution. In this case, the business analyst identified the following Activity Partitions: Patient, Administration Area (which is a

top partition that is divided into Admission and Accounting middle partitions), and the Medical Area (divided into Medical Evaluation and Exams).

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

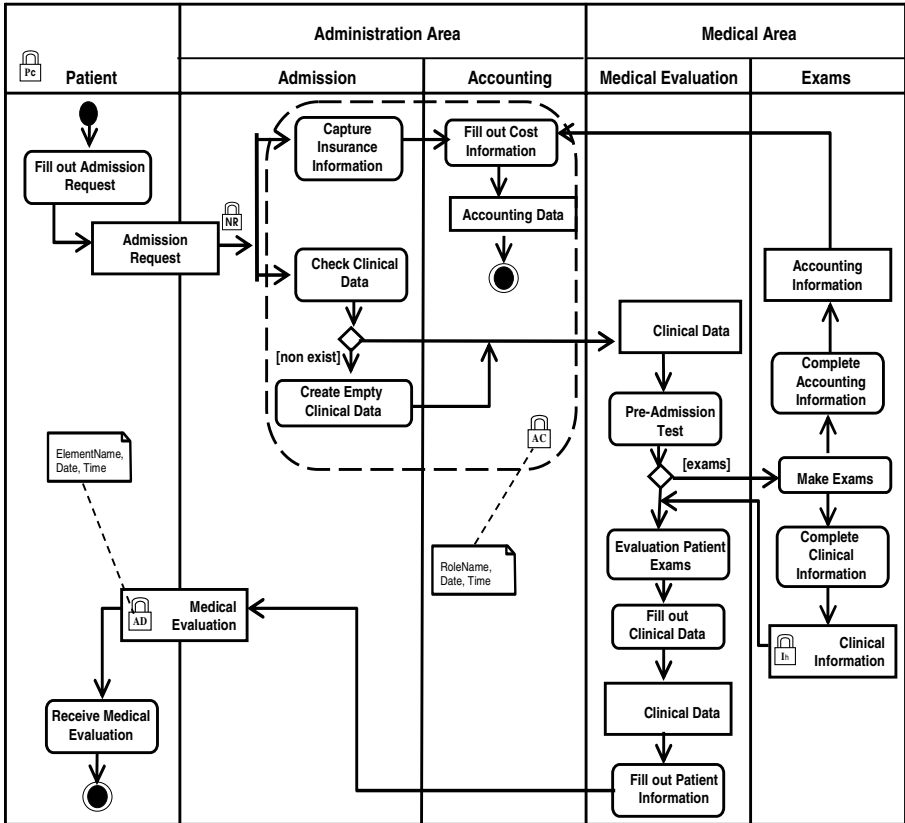


Fig. 2. Admission of Patients in a Medical Institution

Table 5. «SecurityRole» and «SecurityPermission» specifications

Role	Permissions → Objects		Permissions → Operations
Admission/Accounting	Action	Capture Insurance Information Fill out Cost information Check Clinical Data Create Empty Clinical Data	Execution CheckExecution Execution Execution
	DataStoreNode	Accounting Data	Update

A «SecurityRole» can be derived from this specification. Admission/Accounting will be a role. All objects in an interruptible region must be considered for permissions specification (see Table 5). Access control specification has been complemented with audit requirement. This implies that it must register role name, date and time of all events related to the region interruptible. Integrity (high) requirement has specified for Data Store “Clinical Information”. Finally, the business analyst has specified Attack Harm Detection with auditing requirement. All events related to attempt or success of attacks or damages are registered (names in this case are clinical information, date and time).

6 Conclusions and Ongoing Work

The advantage of early representing requirements, in this case, security requirements, favours the quality of the business process since it provides it with more expressivity and improves the software quality since it considers characteristics that, in other way, would have to be incorporated late. So, we can save on maintenance costs as well as on the total cost of the project. We have defined a microprocess that complements the requirements stage defined in the Unified Process and we have used UML 2.0 to represent security requirements.

The next step should be that of applying an MDA approach to transform the model (including the security requirements) into most concrete models (i.e. execution models). Therefore, future work must be oriented to enrich the security requirements specifications, improving the UML extension specification to complement it with Well-Formedness Rules and OCL.

Acknowledgements

This research is part of the following projects: DIMENSIONS (PBC-05-012-1) and MISTICO, both supported by FEDER and the “Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha”, and COMPETISOFT granted by CYTED.

References

1. Abie, H., Aredo, D. B., Kristoffersen, T., Mazaher, S. and Raguin, T.; *Integrating a Security Requirement Language with UML*, 7th International Conference, The UML: Modelling Languages and Applications. Vol. 3273. Lisbon, Portugal. (2004). pp.350-364.
2. Artelsmair, C. and Wagner, R.; *Towards a Security Engineering Process*, The 7th World Multiconference on Systemics, Cybernetics and Informatics. Vol. VI. Orlando, Florida, USA. (2003). pp.22-27.
3. Backes, M., Pfitzmann, B. and Waider, M.; *Security in Business Process Engineering*, International Conference on Business Process Management (BPM). Vol. 2678, LNCS. Eindhoven, The Netherlands. (2003). pp.168-183.

4. Basin, D., Doser, J. and Lodderstedt, T.; *Model driven security for process-oriented systems*, SACMAT 2003, 8th ACM Symposium on Access Control Models and Technologies. Villa Gallia, Como, Italy. (2003).
5. Bock, C.; *UML 2 Activity and Action Models*, Journal of Object Technology. Vol. 2 (4), July-August. (2003). pp.43-53.
6. Eriksson, H.-E. and Penker, M., *Business Modeling with UML*, OMG Press. (2001).
7. Firesmith, D.; *Engineering Security Requirements*, Journal of Object Technology. Vol. 2 (1), January-February. (2003). pp.53-68.
8. Firesmith, D.; *Specifying Reusable Security Requirements*, Journal of Object Technology. Vol. 3 (1), January-February. (2004). pp.61-75.
9. Fuggetta, A.; *Software process: a roadmap*, ICSE 2000, 22nd International Conference on Software Engineering, Future of Software Engineering. Limerick Ireland. (2000). pp.25-34.
10. Herrmann, G. and Pernul, G.; *Viewing Business Process Security from Different Perspectives*, 11th International Bled Electronic Commerce Conference. Slovenia. (1998). pp.89-103.
11. Jacobson, I., Booch, G. and Rumbaugh, J., *El proceso unificado de desarrollo de software*, . (2000). 464 p.
12. Jürjens, J., *Secure Systems Development with UML*, Springer Verlag, (2004). 309 p.
13. Kalnins, A., Barzdins, J. and Celms, E.; *UML Business Modeling Profile*, Thirteenth International Conference on Information Systems Development, Advances in Theory, Practice and Education. Vilnius, Lithuania. (2004). pp.182-194.
14. List, B. and Korherr, B.; *A UML 2 Profile for Business Process Modelling*, 1st International Workshop on Best Practices of UML (BP-UML 2005) at ER-2005. Klagenfurt, Austria. (2005).
15. Lodderstedt, T., Basin, D. and Doser, J.; *SecureUML: A UML-Based Modeling Language for Model-Driven Security*, The Unified Modeling Language, 5th International Conference. Vol. 2460. Dresden, Germany. (2002). pp.426-441.
16. Lopez, J., Montenegro, J. A., Vivas, J. L., Okamoto, E. and Dawson, E.; *Specification and design of advanced authentication and authorization services*, Computer Standards & Interfaces. Vol. 27 (5). (2005). pp.467-478.
17. Maña, A., Montenegro, J. A., Rudolph, C. and Vivas, J. L.; *A business process-driven approach to security engineering*, 14th. International Workshop on Database and Expert Systems Applications (DEXA). Prague, Czech Republic. (2003). pp.477-481.
18. Maña, A., Ray, D., Sánchez, F. and 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. Leganés, Madrid. España. (2004). pp.383-392.
19. Mouratidis, H., Giorgini, P. and Manson, G. A.; *When security meets software engineering: a case of modelling secure information systems*, Information Systems. Vol. 30 (8). (2005). pp.609-629.
20. Object Management Group; *Unified Modeling Language: Superstructure*, version 2.0, formal/05-07-04. In <http://www.omg.org/docs/formal/05-07-04.pdf>. (2005).
21. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 6th Edition, (2006). 880 p.
22. Quirchmayr, G.; *Survivability and Business Continuity Management*, ACSW Frontiers 2004 Workshops. Dunedin, New Zealand. (2004). pp.3-6.
23. Röhm, A. W., Herrmann, G. and Pernul, G.; *A Language for Modelling Secure Business Transactions*, 15th. Annual Computer Security Applications Conference. Phoenix, Arizona. (1999). pp.22-31.

24. Röhm, A. W., Pernul, G. and Herrmann, G.; *Modelling Secure and Fair Electronic Commerce*, 14th. Annual Computer Security Applications Conference. Scottsdale, Arizona. (1998). pp.155-164.
25. Siponen, M. T.; *Analysis of modern IS security development approaches: towards the next generation of social and adaptable ISS methods*, Information and Organization. Vol. 15. (2005). pp.339-375.
26. Stefanov, V., List, B. and Korherr, B.; *Extending UML 2 Activity Diagrams with Business Intelligence Objects*, 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK2005). Copenhagen, Denmark. (2005).
27. Vivas, J. L., Montenegro, J. A. and Lopez, J.; *Towards a Business Process-Driven Framework for security Engineering with the UML*, Information Security: 6th International Conference, ISC. Bristol, U.K. (2003). pp.381-395.
28. Zulkernine, M. and Ahamed, S. I., *Software Security Engineering: Toward Unifying Software Engineering and Security Engineering*, in: Idea Group (Ed.), Enterprise Information Systems Assurance and Systems Security: Managerial and Technical Issues, M. Warkentin & R. Vaughn, 2006, p.215-232.