

# An MDA approach to develop secure business processes through a UML 2.0 extension

Alfonso Rodríguez<sup>1</sup> Eduardo Fernández-Medina<sup>2</sup> and Mario Piattini<sup>2</sup>

<sup>1</sup>Departamento de Auditoría e Informática, Universidad del Bío Bío, Chillán, Chile.  
Email: alfonso@ubiobio.cl

<sup>2</sup>ALARCOS Research Group, Information Systems and Technologies Department, UCLM-Soluziona Research and Development Institute, University of Castilla-La Mancha, Ciudad Real, Spain.  
Email: {Eduardo.FdezMedina,Mario.Piattini}@uclm.es

---

Business Processes have become essential to the performance of enterprises since they incorporate the differentiating aspects that generate a higher financial performance. Consequently, business process modeling is the centre through which to conduct and improve how a business is operated. Moreover, security is a crucial issue for business performance, but is usually considered after the business processes definition. Many security requirements can be expressed at the business process level. A business process model is important for software developers, since they can use it to obtain the necessary requirements for software design and creation. In this work, we shall show a microprocess, consistent with the MDA approach, through which it is possible to specify and refine security requirements at a high level of abstraction and obtain a subset of all analysis-level classes and use cases from the context problem, in such a way that they can be incorporated into the development of a software system. In addition, an extension of UML 2.0 activity diagrams through which it is possible to identify such requirements will be presented.

Keywords: Business Process, Security Requirement, UML, Activity Diagrams

---

## 1. INTRODUCTION

Business process identification and modeling are becoming more and more important for enterprises since their own resources may be essential in their performance in a market that is continually more complex. A business process can be considered as a source of comparative advantage that, once its efficiency and effectiveness have improved, can give place to positions of comparative advantage that will suppose a higher financial performance consistent with the resource advantage theory of competition proposed in [17]. Consequently, business processes, are a key factor in maintaining competitiveness, since they are the means by which an enterprise may describe, standardize, and adapt the way in which it reacts to certain types of business events, and how it interacts with suppliers, partners, competitors, and customers [38].

The new business scene, in which there are many participants and an intensive use of communication and information technologies, implies that enterprises not only expand their businesses but also increase their vulnerability. As a consequence, and with the increase of the number of attacks on systems, it is highly probable that sooner or later an intrusion may be successful [36]. This security violation causes losses. For this reason, it is necessary to protect computers and their systems in the best possible way. The best possible security does not necessarily mean absolute security, but a reasonably high security level in relation to the given limitations [43].

Regardless of the importance of the notion of security for enterprises, it is often neglected in business process models, which usually concentrate on modeling the process in order to show functional correctness [4]. This is mainly due to the fact that the expert in the business process domain is not an expert in security

[16]. Typically, security is considered after the definition of the system. This approach often leads to problems, which are usually translated into security vulnerabilities [31], which clearly justify the need to increase the effort at the pre-development stages, when fixing bugs is cheaper [26].

If we consider that 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 [26], then it is possible to obtain a high level of security requirements which are easily identifiable to those who model business processes. Moreover, requirements specification usually results in a specification of the software system which should be as exact as possible [3], since effective business process models facilitate discussion among the different stakeholders in the business, allowing them to agree on the key fundamentals as well as to work towards common goals [11].

There are several languages and notations for business process modeling [15]. However, Unified Modeling Language (UML) is a widely accepted standard notation. The most important change to the UML 2.0 version with respect to the previous ones has been that of activity diagrams which improve the business process representation. Our work considers a UML 2.0 extension that allows us to incorporate security requirements into activity diagrams from the perspective of the business analyst. We have considered the security requirements identified in the taxonomy proposed in [13].

Our proposal is based on the Model Driven Architecture (MDA) approach. We have defined a microprocess that allows us the early identification and representation of business requirements (including those of security) which are defined in Computation Independent Models (CIM). From these requirements and through the application of transformation rules, we can obtain the UML artifacts used to describe the problem in the context of Platform Independent Models (PIM). Such artifacts permit us to complement the acquisition requirements defined in the Unified Software Development Process [18]. We have defined a UML 2.0 activity diagram extension to obtain security requirements.

The structure of the remainder of the paper is as follows: in Section 2, we will summarize the main issues regarding security in business processes and information systems. In Section 3, we will present an overview on business process modeling techniques. In Section 4, we will present a brief overview of UML 2.0 activity diagrams and extensions. In Section 5, we will propose a microprocess for the security requirements specification and a UML 2.0 extension that will allow the business analyst to carry out this task. Finally, in Section 6, we will present an illustrative example, and in Section 7 our conclusions will be drawn.

## 2. SECURITY IN BUSINESS PROCESSES AND INFORMATION SYSTEMS

In this section we shall first present a review of the main studies related to security in business processes, and then, those related to information systems.

Security in business processes which takes an early specification of security requirements into consideration has been dealt

with in [16, 37]. In [16] an approach to model security by considering several perspectives is presented. The authors take the following perspectives into consideration: static, which deals with the security of processed information, functional, from the viewpoint of the system processes, dynamic, which deals with the security requirements from the life cycle of the objects involved in the business process, organizational, which is used to relate responsibilities to acting parties within the business process and the business process perspective. This provides us with an integrated view of all perspectives with a high degree of abstraction. On the other hand, [37] complements the previous approach by establishing that this perspective can be used at the different stages that make up a business transaction. The authors propose Commercial Protocols and Service (COPS) as an infrastructure to build adaptable electronic markets that emphasize security and equity and Modelling Security Semantics of Business Transactions (MOSS) as a methodology to analyze and model the semantics of security in business transactions. Both proposals are complementary and very clear in aspects related to the need of integrating security from early stages, considering for this purpose business processes as the starting point of the specification. However, the approaches within the business process paradigm do not address either the issue of how they can be integrated into business processes or Information Systems (IS) development approaches and no explicit connection with processes or notations of IS or "normal" business process development methods [39] is shown. In spite of this, this approach helps us solve a problem that has not previously been considered from this perspective. It puts forward the need to aggregate a new view of security without rejecting the traditional view of security implemented by experts. Moreover, it clearly establishes the need to use these specifications to achieve implementation.

In [4], security is dealt with under the business processes engineering perspective. The authors propose that special attention should be paid to the incorporation of cryptography as a security requirement. To do so, they consider an approach based on refining by stages and they widen this by aggregating security requirement specifications as well as trust models. This will generate a security specification that will later be transformed into refined specifications already incorporating security. Although this approach clearly establishes the need to specify security requirements at an early stage, it does not mention either the way in which these security requirements will be specified (notation or technique) or the different roles that will be fulfilled by those involved in a business process.

In [27, 42] a framework managed by business processes for security engineering with UML is proposed. UML is used to represent security semantics in an environment of integrated development including business processes and systems models. This allows us to integrate security requirements in the same way as other requirements in the context of software development. Furthermore, a method for systems development managed by business processes in which technology decisions are managed by the business model is proposed. The need to express security requirements at the level of a business model is due to the fact that applications considering electronic commerce transactions are conceptually similar to non-automated traditional transactions. Notions such as non repudiation, confidentiality, integrity, access control and authentication were already present

in business transactions before the appearance of automated systems. This framework is based on UML and integrates security requirements into the system's business process model. Security notions are expressed through a UML extension. Both proposals have the advantage of recognizing the need to express security requirements at an early stage and use them to achieve the system implementation. They use a widely accepted modeling language that facilitates their use and understanding. Nevertheless, the authors recognize that, despite the formalisms which are necessary to build models that allow us to express security in business processes, it is also necessary to carry out additional work to define the semantics for business process models in an exact manner.

In [19, 20] a proposal for the development of secure systems using a language based on UML is presented. The author has based this proposal on the idea of using a standard in the modeling industry which is oriented towards an object with the aim of integrating the security requirements analysis and the systems development process to avoid the classical problems associated with a late consideration of these requirements. From a practical point of view, this approach allows developers (who are probably not specialists in security) to use their knowledge about security engineering through the proposed extension (UMLSec). However, this proposal is mainly oriented towards access control policies and the way in which they can be integrated into the software development process [31].

In [5, 24] a proposal for a language based on UML for security modeling oriented by models is presented. The authors present a modeling language based on UML for the development of secure distributed models. This approach is based on the Role-Based Access Control (RBAC) policy with an additional support for restrictions specifications. This extension, SecureUML, is well defined and satisfies the access control specification using RBAC. Nonetheless, other security requirements such as privacy or non-repudiation are excluded and furthermore, the way in which the behaviour of the system's components is represented is not indicated.

The proposal of [3] is oriented towards solving the problem of designing secure systems. With this purpose in mind, the authors present guidelines that allow us to perform a security engineering process that helps the application developer in the conceptual design of secure systems. These guidelines can easily be applied by those who are familiar with the software development process and who have a basic knowledge of security. The main contributions of the proposed approach are related to the need to obtain security requirements at an early stage, integrate them into different stages of the system's development and to the use of a standard (UML) for modeling. However, it does not refer to the modeling of the system's dynamic characteristics.

A model of secure information systems is proposed in [30, 31]. These authors present an approach which integrates both security engineering and information systems, using the same concepts and notations, throughout the whole process of the system's development. This proposal is based on the Tropos methodology which considers security requirements as an integral part of the whole development process.

A complete analysis of approaches oriented towards developing security in information systems is stated in [39]. The author identifies the disciplines as well as the research communities that form the basis of the approaches related to security in informa-

tion systems. The author also puts forward the assumptions that support these approaches and in conclusion he presents a classification of five generations concerning the approaches related to security in information systems.

Finally, the proposal of [44] points out that software engineering and security engineering should be unified. To do so, this approach complements the traditional stages of software development with security specifications. At the stage of systems engineering, the authors propose that not only the subjects concerning functionality but also security must be identified by users or interested parties. Such specifications must be referred to system elements such as hardware, software, people involved, databases, documentation and procedures. This proposal studies all the aspects of systems construction and relates them to security. In this work, the existing difficulty to represent security at the early stages of software development is clearly indicated. The use of scenarios to cover that stage is proposed. However, due to the absence of an example or case study, it is not possible to observe the way in which specifications are performed and how they behave until they achieve software specification and implementation.

In summary, we have discovered two problems related to security in business process and information systems. The first is that modeling has not been adequate since, generally, those who specify security requirements are requirements engineers who have accidentally tended to use specific restrictions architecture instead of security requirements [12]. The second is that security has been integrated into an application in an ad-hoc manner, often during the actual implementation process [4], during the system administration phase [24] or it has been considered as outsourcing [28].

### 3. BUSINESS PROCESS MODELING TECHNIQUES

The main objective of business process modeling is to produce a realistic description of, 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, it is important to have a notation that allows us to model the essence of the business as clearly as possible. This notation must allow us to incorporate different perspectives which give place to different diagrams in which the rules, goals, objectives of the business and not only relationships but also interactions are shown [10]. A high percentage of the success of modeling is based on the ability to express the different needs of the business as well as having a notation in which these needs can be described. This is why when choosing an approach and/or notation, the properties of the object to be modeled (in other words, the business process, the environment features and the underlying reasons for its use) must be taken into account [6].

Among the techniques that have been used for business process modeling, we can highlight the following: flow diagrams, data flow diagrams, entity-relationship diagrams, state-transition diagrams, Gantt charts, Role Activity Diagrams (RAD), the family of techniques known as Integration Definition for Function Modeling (IDEF), Petri Nets, simulation, techniques based on knowledge (artificial intelligence) and workflow techniques [1, 15].

At present, and according to the state of the business process modeling industry [25, 29], it is possible to identify the Unified Modeling Language (UML) [34] and the Business Process Modeling Notation (BPMN) [9], among the main standards; we shall thus focus our analysis on both of them.

The use of UML is highly spread out in relation to business process modeling [10, 21, 24, 27, 40, 42], since it is a consolidated language, which is easy to learn and which allows fluent communication between the different participants in the model.

In this paper, we have used UML 2.0 [33] because the business process representation has been improved through activity diagrams. Therefore, its capacity to represent many aspects of the systems from early stages in software development is also improved.

#### 4. UML 2.0 ACTIVITY DIAGRAMS AND UML 2.0 EXTENSIONS

UML 2.0 is divided into structural and behavioral specifications. Behavior models specify how the structural aspects of a system change over time. UML has three behavior models: activities, state machines, and interactions. Activities focus on the sequence, conditions, and inputs and outputs to invoke other behaviors, state machines show how events cause changes of object state and invoke other behaviors, and interactions describe message-passing between objects that causes invocation of other behaviors [8].

Activity diagrams are the UML 2.0 elements used to represent business processes and workflows [22]. In previous UML versions, expressivity was limited and this fact confused users who did not use orientation towards objects as an approach for modeling. It is now possible to support flow modeling throughout a wide variety of domains [7]. An activity specifies the coordination of executions of subordinate behaviors, using a control and a data flow model. Activities may form invocation hierarchies invoking other activities, ultimately resolving individual actions [33]. 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 meta-models to be extended in order to adapt them for different purposes. The profiles mechanism is consistent with the OMG Meta Object Facility (MOF) [33]. 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. pre or post conditions, invariants). They can be expressed in natural language, programming language or through OCL (Object Constraint Language). Tagged values are additional meta-attributes assigned to a stereotype, specified as name-value pairs.

Research works related to UML 2.0 extensions and business processes refer to aspects of the business such as the Customer, Type of Business Process, Goal, Deliverability and Measure [23], Data Warehouse and its relationship to the business process dynamic structures [41], or they add semantics to the activities by considering organizational aspects that allow us to express resource restrictions during the execution of an activity [22].

### 5. 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 [35]. These models consider a stage in which we should obtain the system requirements either from the client or from the interested parties 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 [18] paying special attention to the acquisition of security requirements. To do so, a UML 2.0 activity diagram profile is proposed.

Figure 1 shows a general overview of our proposal. BPSec is situated at the top of the figure. This is the extension that we propose in which to include security in business processes (see section 5.2). The specification of business processes including security requirements generates a Secure Business Process. Through the application of a set of transformation rules described with Query/View/Transformation (QVT) (C/P-1, C/P-2 y C/P-3) to the Secure Business Process, it is possible to obtain a subset of the classes of analysis and use cases that facilitate the understanding of the problem. These specifications in UML are used as an entrance to the first stages of the Unified Process (right side of the figure). The application of these transformations is within the scope of the MDA proposal [32] where we went from a Computation Independent Model (CIM) to Platform Independent Models (PIM) (left side of the figure). We have created a microprocess (SeReS4BP described in detail in section 5.1) that facilitates not only the understanding but also the systematic application of the elements considered in our proposal. In this work, we will only show the description of the microprocess and the extension of activity diagrams in detail. Nevertheless, in Section 6, we have included the results derived from the transformations application, with the sole purpose of illustrating the way in which UML artifacts complementing the first stages of the Unified Process are obtained.

#### 5.1 SeReS4BP Microprocess

We have considered the use of the Unified Software Development Process stated by Jacobson, Booch and Rumbaugh (2000) since it is a fairly consolidated and successful software construction method [14]. This process is composed of 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 performed during the inception and elaboration stages. The objective of this task is to describe the system's requirements (conditions and capabilities that must be fulfilled by the system) well enough to determine what the system must or must not do. To do so, 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 are considered.

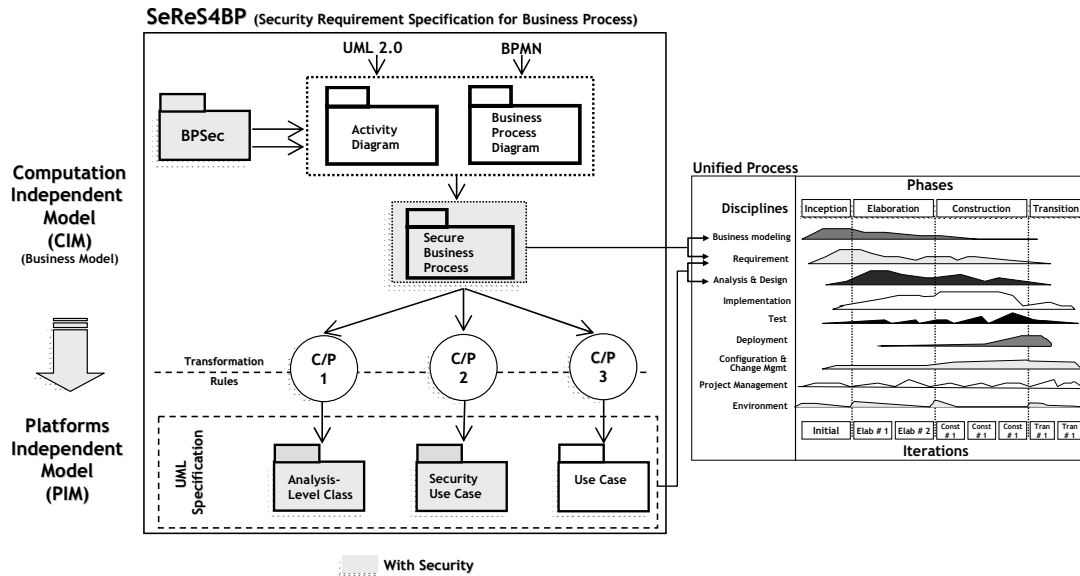


Figure 1 Our architectural proposal.

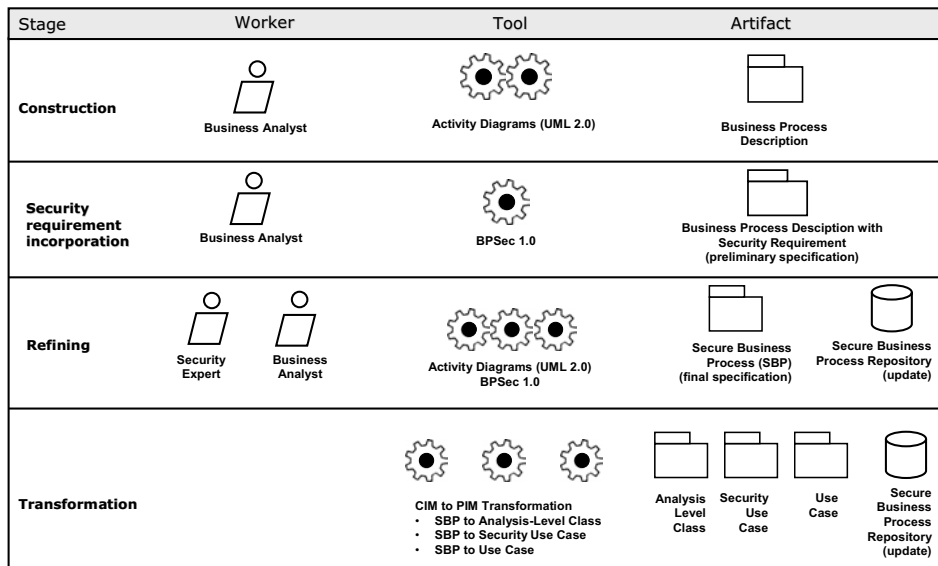


Figure 2 Complete view of the SeReS4BP microprocess.

The security requirements specified in the business process can be perfectly linked to the Unified Process. To do this, we propose complementing the task “to understand the system context” with the 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 the business analyst’s perspective into account. This microprocess is called SeReS4BP (Security Requirement Specification for Business Process). Figure 2 shows us a view of the main activities performed in this microprocess.

We will describe SeReS4BP through:

(i) The stages forming it:

- **CONSTRUCTION:** The objective of this stage is to construct the business process model. To attain this objective, the UML 2.0 activity diagram must be used.

- **SECURITY REQUIREMENTS INCORPORATION:** This stage consists of incorporating security requirements, from the business analyst viewpoint, into the business process model specified in the previous stage.
- **REFINING:** 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 and the specifications that will finally be incorporated into the business process are agreed. Together with the requirements validation, the attribute *priority* must be added to each requirement. This attribute may have the following values: *must have*, *should have*, *could have*, or *want to have* according to the established in [2].
- **TRANSFORMATION:** The objective of this stage is to use the activity diagram specification to obtain the analysis-level

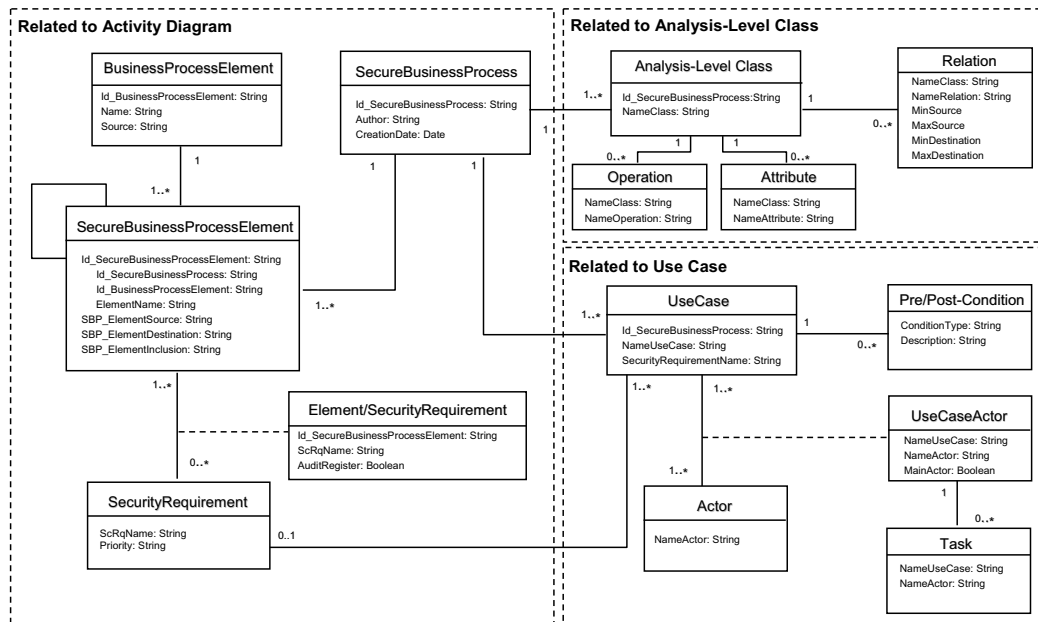


Figure 3 Secure Business Process Repository Model.

classes, the use cases related to security and the use cases related to another aspect concerning the business process. This stage does not require workers because the artifacts are automatically generated.

(ii) The type of workers involved:

- **BUSINESS ANALYST:** 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.
- **SECURITY EXPERT:** he/she will be the person responsible for refining the security specifications indicated by the business analyst. Such refining considers the verification of the validity and completeness of the specifications.

(iii) The tools used:

- **UML 2.0 ACTIVITY DIAGRAMS:** for business process specification.
- **UML 2.0 USE CASES DIAGRAMS:** automatically generated from Secure Business Process.
- **UML 2.0 CLASS DIAGRAMS:** automatically generated from Secure Business Process.
- **BPSEC 1.0:** for security requirements specifications.

(iv) The artifacts generated from its application:

- **BUSINESS PROCESS DESCRIPTION:** This artifact is the result of the construction stage. It contains the business process specifications and can be built using UML. It does not contain security specifications.
- **SECURE BUSINESS PROCESS:** This artifact is the result of the stages of incorporation of security requirements and refining. The first stage contains preliminary security specifications which, after refining, will be converted into definitive security specifications.

- **SECURE BUSINESS PROCESS REPOSITORY** (see Figure 3): This contains information about the secure business process specification. Information concerning the activity diagram, security requirements specification, analysis-level classes and use cases are stored in it.
- **ANALYSIS-LEVEL CLASS:** This contains a subset of the analysis-level classes which describes the problem of modeling a secure business process. This model of classes is automatically obtained and includes the classes related to security that are derived from the specifications performed in the secure business process.
- **USE CASE DESCRIPTION:** This artifact contains a subset of the use cases of the described problem of the secure business process. It is automatically obtained and the main elements describing it are stored in the secure business process repository.
- **SECURITY USE CASE DESCRIPTION:** This contains the specifications of the security use cases that are automatically obtained from the security specifications carried out in the secure business process. Information concerning security use cases is stored in the repository.

## 5.2 BPsec Version 1.0 for modeling security requirements in Business Processes

In this section, we shall present the main aspects of our profile for representing security requirements in business processes. 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 [13]. Later these requirements will be transformed, by the security experts, into technical specifications including all the details necessary for their implementation.

Our Profile will be called BPSec (Secure Business Process) and will be represented as a UML Package. This profile will incorporate new data types, stereotypes, tagged value and constraints. In Figure 4, a high level view is provided.

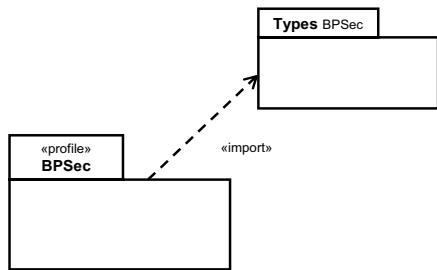


Figure 4 High level view of BPSec Profile.

In addition, the definitions of some new data types to be used in tagged value definitions are necessary. In Table 1, shows the new data type stereotypes definitions. In Figure 5, we have shown the stereotypes (dark-coloured) for Secure Activity specifications. In Figure 6, we can observe the values associated with each of the necessary types. All the new types must be considered when business analysts specify security requirements in the business process. We have defined a package that includes all the stereotypes that will be necessary in our profile.

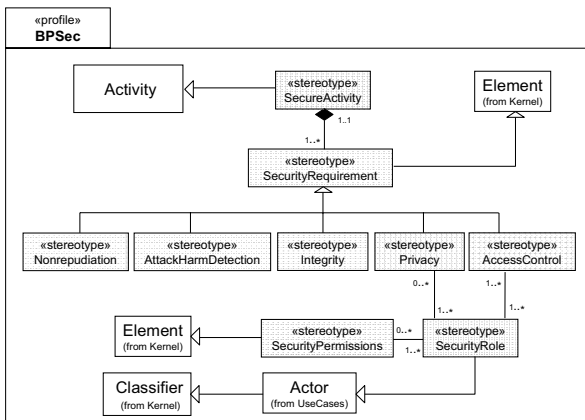


Figure 5 New Stereotypes.

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 letters to it that will allow us to identify the type of requirement that is specified.

The existing relationship between the new stereotypes and the activity diagram elements is shown in Table 2 and is indicated with a ✓ symbol. In this way, the stereotypes derived from “SecurityRequirement” (Nonrepudiation, AttackHarmDetection, Integrity, Privacy or AccessControl) can be added to activity diagram elements. Any security requirement can be added to activity diagram elements (see Table 2 and Figure 7). For example, an “Integrity” requirement can be specified to the data store and/or the object flow.

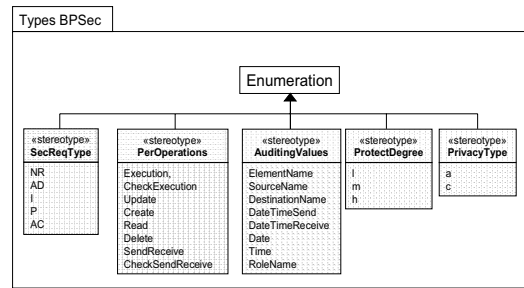


Figure 6 Values associated with new data types.

“SecurityRole” and “SecurityPermissions” are related in different ways as both can be obtained from the UML 2.0 element of activity diagrams (see Table 2 and Figure 7). For example, “SecurityRole” can be obtained from partitions or regions specifications, but is not specified in an explicit way in these activity diagram elements. “SecurityPermission” is a special case because permissions depend on each activity diagram element which they are related to. For example, for Actions object, Execution or CheckExecution, “SecurityPermission” operations must be specified (see Table 5).

Figure 7 shows us the model of classes that describes the existing relationships between the elements of the UML activity diagram and the new stereotypes. This model is complemented with the relationships shown in Table 2 and is used to validate security specifications in a business process.

In Tables 3 and 4 the stereotypes for secure activity specifications will be shown in much greater detail. Each stereotype specification contains: name, base class, description, notation (optional), constraints and tagged values (optional).

## 6. EXAMPLE

Our illustrative example (see Figure 8) 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 (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 the Activity Partition “Patient”, with the aim of preventing the disclosure of sensitive information about Patients. “Nonrepudiation” has been defined over the control flow which 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. 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 a role name, a date and the time of all events related to the interruptible region. Integrity (high) requirement has been specified for Data Store “Clinical Information”. Finally, the business analyst has

Table 1 New data types.

Name	Description	Values associated
SecReqType	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	An enumeration for possible operations on objects in activity diagrams. These operations are related to permissions granted to the object	Execution, CheckExecution, Update, Create, Read, Delete, SendReceive, CheckSendReceive
ProtectDegree	An abstract level that represents criticality. This degree may be low (l), medium (m) or high (h).	l, m, h
PrivacyType	Consists of anonymity (a) or confidentiality (c).	a, c
AuditingValues	Different security events related to the security requirement specification in business processes. These values will be used in later auditing	ElementName, SourceName, DestinationName, DateTimeSend, DateTimeReceive, Date, Time, RoleName

Table 2 Security Requirements and Activity Diagram Elements.

Stereotypes for secure activity specification	UML 2.0 element for containment in activity diagrams				
	Activity Partition	Interruptible Activity Region	Action	Data StoreNode	ObjectFlow (data)
Nonrepudiation					✓
AttackHarmDetection	✓	✓		✓	✓
Integrity				✓	✓
Privacy	✓	✓			
AccessControl	✓	✓	✓	✓	✓
Security Role	✓	✓			
SecurityPermissions			✓	✓	✓

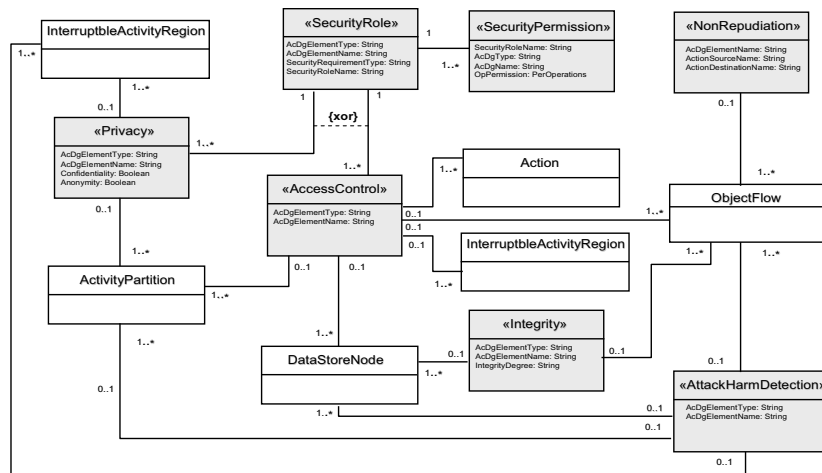


Figure 7 BPSec stereotypes and activity diagram elements.

specified Attack Harm Detection with auditing requirement. All events related to attempt or success of attacks or damages are registered (the names in this case are: clinical information, the date and the time).

From the specification of a secure business process (see Figure 8) and by applying a set of transformation rules (C/P-1, C/P-2 and C/P-3 in Figure 1), it is possible to obtain a subset of the analysis-level classes and use cases that allow us to describe the



Table 3 Stereotypes specifications for security requirement.

<b>Name</b>	<b>SecurityPermission</b>
Base Class	Element (from Kernel)
Description	Contains permission specifications. A permissions specification must contain details about the objects and the operations involved Must be associated with security role specification <b>context</b> SecurityPermission <b>inv:</b> self.SecurityRole ->size()>= 1 Must be associated with Actions, DataStoreNode or ObjectFlow <b>context</b> SecurityPermissions <b>inv:</b> self.Actions.size+self.DataStoreNode.size+self.ObjectFlow.size=1 Must be specified such as Objects and Operations pairs. <b>context</b> SecurityPermissions <b>inv:</b> 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
Constraints	
Tagged Values	SecurityPermissionOperation: SecPerOperations
Name	SecurityRole
Base Class	Actor (from UseCases)
Description	Contains a role specification. This role must be obtained from access control and/or privacy specifications The role in the security role stereotype can be derived from: Activity, ActivityPartition and/or InterruptibleActivityRegion Must be associated with an access control specification and can be associated with privacy and security permissions
Constraints	<b>context</b> SecurityRole <b>inv:</b> self.AccessControl -> size() >= 1 <b>context</b> SecurityRole <b>inv:</b> self.Privacy -> size()>= 0 <b>context</b> SecurityRole <b>inv:</b> self.SecurityPermission -> size()>= 0

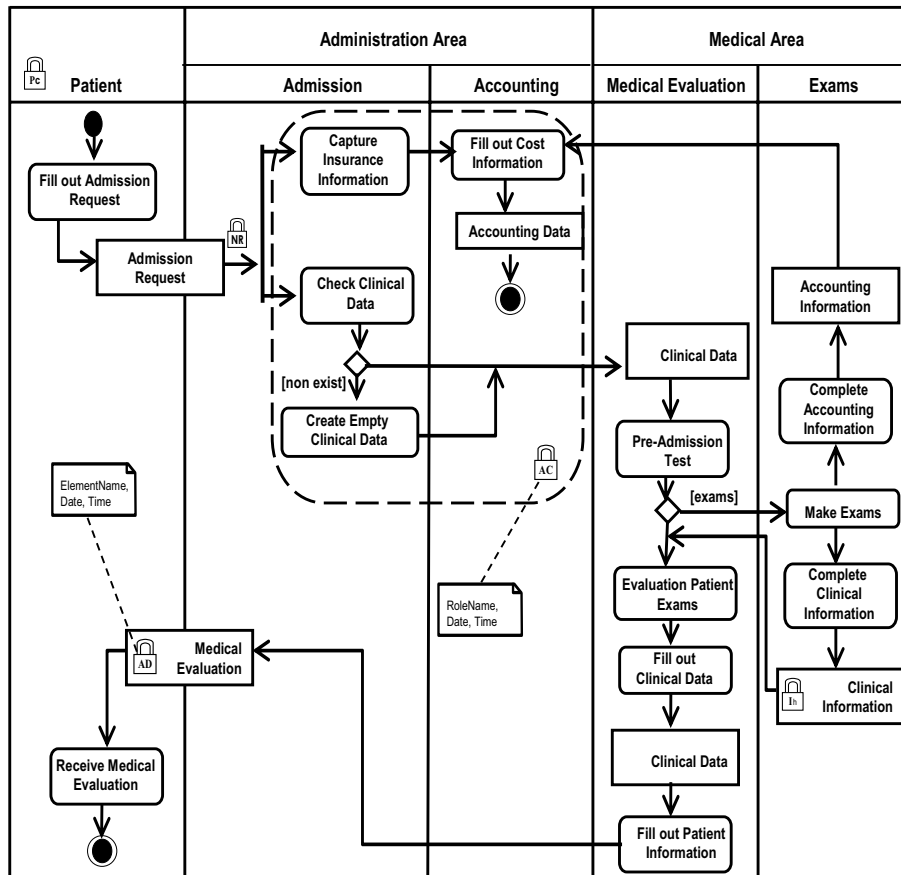








Figure 8 Admission of Patients to a Medical Institution.

**Table 4** Security activity and security requirement stereotypes.

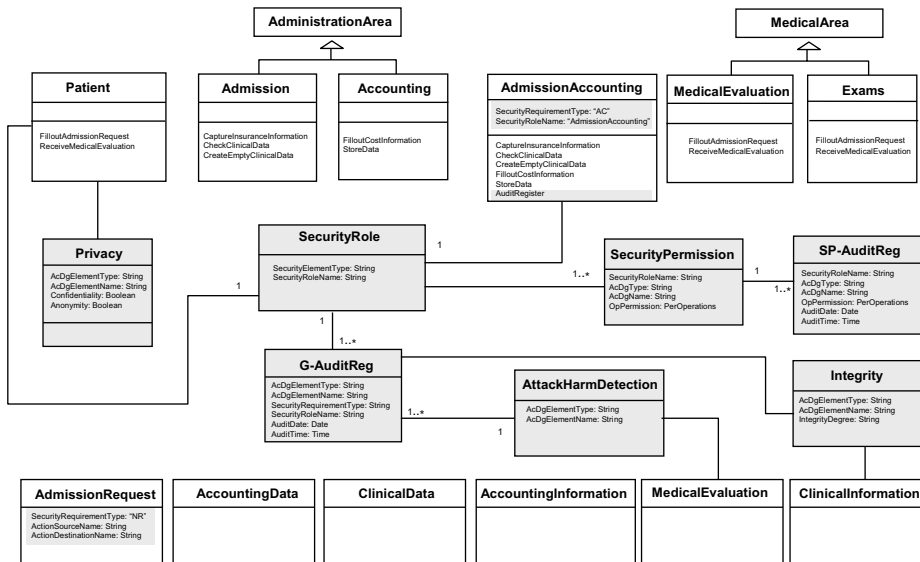
Name	SecureActivity	
Base Class	Activity	
Description	A secure activity contains security specifications related to requirements, role identifications and permissions	
Constraints	Must be associated with at least one SecurityRequirement <b>context</b> SecureActivity <b>inv:</b> self.SecurityRequirement->size()>=1	
Name	SecurityRequirement	<b>Notation</b>
Base Class	Element	
Description	Abstract class containing security requirements specifications. Each security requirement type must be indicated in some of its subclasses	
Constraints	A security requirement must be associated with a secure activity <b>context</b> SecurityRequirement <b>inv:</b> self.SecureActivity ->size()=1 The notation must be completed in the subclass specification for each security requirement. One security requirement type must be used.	
Tagged Values	SecurityRequirementType: SecReqType	
Name	Nonrepudiation	<b>Notation</b>
Base Class	SecurityRequirement	
Description	Establishes the need to avoid the denial of any aspect of the interaction. An auditing requirement can be indicated in Comment	
Constraints	Can only be specified in the diagram elements indicated in Table 2.	
Tagged Values	AvNr: AuditingValues <b>context</b> Nonrepudiation <b>inv:</b> self.AvNr="ElementName" or self.AvNr="SourceName" or self.AvNr="DestinationName" or self.AvNr="DateTimeSend" or self.AvNr="DateTimeReceive"	
Name	AttackHarmDetection	<b>Notation</b>
Base Class	SecurityRequirement	
Description	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	
Tagged Values	AvAD: AuditingValues <b>context</b> AttackHaarmDetection <b>inv:</b> self.AvAD="ElementName" or self.AvAD="Date" or self.AvAD="Time"	
Name	Integrity	<b>Notation</b>
Base Class	SecurityRequirement	
Description	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.	
Constraints	Can only be specified in the diagram elements indicated in Table 2. The Protection Degree must be specified by adding a lower case letter according to the PDI tagged value.	
Tagged Values	PDI : ProtectDegree AvI: AuditingValues <b>context</b> Integrity <b>inv:</b> self.AvI="ElementName" or self.AvI="Date" or self.AvI="Time"	
Name	Privacy	<b>Notation</b>
Base Class	SecurityRequirement	
Description	Indicates the degree to which non authorized parts are avoided in order to obtain sensitive information. An auditing requirement can be indicated in Comment.	
Constraints	Can only be specified in the diagram elements indicated in Table 2. A privacy requirement has one security role specification <b>context</b> Privacy <b>inv:</b> self.SecurityRole -> size() = 1 The Privacy Type must be specified by adding a lower case letter according to the Pv tagged value. If privacy type is not specified then anonymity and confidentiality are considered.	
Tagged Values	Pv: PrivacyType AvPv: AuditingValues <b>context</b> Privacy <b>inv:</b> self.AvPv="RoleName" or self.AvPv="Date" or self.AvPv="Time"	
Name	AccessControl	<b>Notation</b>
Base Class	SecurityRequirement	
Description	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.	
Constraints	Can only be specified in the diagram elements indicated in Table 2. Valid only if at least one security role is specified. <b>context</b> AccessControl <b>inv:</b> self.SecurityRole -> size() >= 1	
Tagged Values	AvAC: AuditingValues <b>context</b> AccessControl <b>inv:</b> self.AvAC="RoleName" or self.AvAC="Date" or self.AvAC="Time"	

context of the problem. Such UML artifacts can be used as a means by which to enter the Unified Process as is shown on the right-hand side of Figure 1.

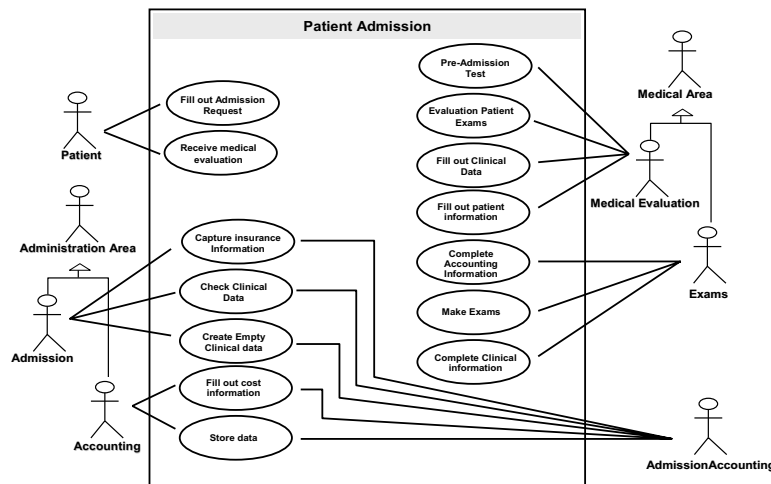
Figure 9 shows the analysis-level classes that can be obtained from the specification of a secure business process are shown. In some cases, the classes are derived directly from certain

**Table 5** “SecurityRole” and “SecurityPermission” specifications.

Role	Permissions		
	Objects	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



**Figure 9** Analysis-Level Classes.



**Figure 10** Use Cases.

elements in the activity diagram. For example, the class “Patient” is obtained from the specification of the patient partition and the operations of that class correspond to the actions carrying out this partition. We have distinguished (dark-coloured) the classes, attributes or operations derived from security specifications. For instance, the specification of “AccessControl” using an Audit Register gives place to attributes that allow us to identify the type of security requirement that has been specified as well as the name of the security role related to the specification. In the same way, the audit register operation has been added. The

classes “SecurityRole” and G-AuditReg are aggregated from the same specification.

In Figure 10, Use Cases derived from the specification of a secure business process are shown. These use cases are obtained from the Actions specifications in the activity diagram and the Actors are obtained from the specification “Activity-Partitions”. Finally, Figure 11 shows a Security Use Case obtained from the security requirement specification “Access-Control” that was indicated in the “AdmissionAccounting” region.

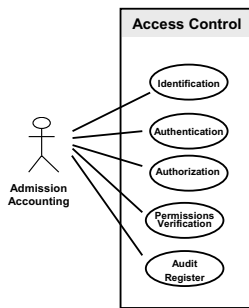


Figure 11 Security Use Case.

## 7. CONCLUSIONS AND ONGOING WORK

The advantage of representing requirements (in this case, security requirements) at an early stage favours the quality of the business process since it provides it with more expressivity and improves the software quality as it considers characteristics that would otherwise have to be incorporated at a later date. It is therefore possible to save on maintenance costs as well as on the total cost of the project. We have defined a microprocess, which under the MDA paradigm transforms the secure business process specifications (CIM) into a subset of analysis classes and use cases (PIM) related to the problem that we have the intention of solving. Our microprocess complements the requirements stage defined in the Unified Process and we have used UML 2.0 to represent security requirements.

The next step in our investigations must be directed towards developing transformations to obtain artifacts that will allow us continue with the software development. Moreover, future work must be oriented towards enriching the security requirement specifications by improving the UML extension specification so as 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 (PBC-06-0082), both supported by the FEDER and the “Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha”, COMPETISOFT (granted by CYTED) and RETISTRUST (TIN2006-26885-E) granted by the “Ministerio de Educación y Ciencia” (Spain).

## REFERENCES

- Aguilar-Savén, R. S.; *Business process modelling: Review and framework*, International Journal of Production Economics. Vol. 90 (2). (2004). pp.129–149.
- Arlow, J. and Neustadt, I., *UML 2*, Ediciones Anaya Multimedia, (2006). 608 p.
- 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.
- 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.
- 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).
- Bider, I.; *Choosing Approach to Business Process Modeling - Practical Perspective*. In <http://www.ibissoft.se/english/howto.pdf>. (2003).
- Bock, C.; *UML 2 Activity and Action Models*, Journal of Object Technology. Vol. 2 (4), July-August. (2003). pp.43–53.
- Bock, C.; *UML 2 Activity and Action Models, Part 2: Actions*, Journal of Object Technology. Vol. 2 (5), September-October. (2003). pp.41–56.
- BPMN; *Business Process Modeling Notation Specification*, OMG Final Adopted Specification, dtc/06-02-01. In <http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>. (2006).
- Castela, N., Tribolet, J., Silva, A. and Guerra, A.; *Business Process Modeling with UML*, Proceedings of the 3st. International Conference on Enterprise Information Systems. Vol. 2. Setubal, Portugal. (2001). pp.679–685.
- Eriksson, H.-E. and Penker, M., *Business Modeling with UML*, OMG Press. (2001).
- Firesmith, D.; *Engineering Security Requirements*, Journal of Object Technology. Vol. 2 (1), January-February. (2003). pp.53–68.
- Firesmith, D.; *Specifying Reusable Security Requirements*, Journal of Object Technology. Vol. 3 (1), January-February. (2004). pp.61–75.
- Fuggetta, A.; *Software process: a roadmap*, ICSE 2000, 22nd International Conference on Software Engineering, Future of Software Engineering. Limerick Ireland. (2000). pp.25–34.
- Giagliis, G. M.; *A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques*, International Journal of Flexible Manufacturing Systems. Vol. 13 (2). (2001). pp.209–228.
- Herrmann, G. and Pernul, G.; *Viewing Business Process Security from Different Perspectives*, 11th International Bled Electronic Commerce Conference. Slovenia. (1998). pp.89–103.
- Hunt, S., *A General Theory of Competition: Resources, Competences, Productivity, Economic Growth*, Sage Publication Inc., First Edition, (2000). 320 p.
- Jacobson, I., Booch, G. and Rumbaugh, J., *El proceso unificado de desarrollo de software*, . (2000). 464 p.
- Jürjens, J.; *Towards Development of Secure Systems Using UMLsec*, Fundamental Approaches to Software Engineering, 4th International Conference, FASE 2001 at ETAPS-2001. Vol. 2029. Genova, Italy. (2001). pp.187–200.
- Jürjens, J.; *Using UMLsec and goal trees for secure systems development*, Proceedings of the 2002 ACM Symposium on Applied Computing (SAC). Madrid, Spain. (2002). pp.1026–1030.
- Jürjens, J., *Secure Systems Development with UML*, Springer Verlag, (2004). 309 p.
- 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.
- 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).
- 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.
25. Lonjon, A.; *Business Process Modeling and Standardization*, BP-Trends. In <http://www.bptrends.com/>. (2004).
  26. 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.
  27. 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.
  28. 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.
  29. Mega; *Business Process Modeling and Standardization*. In <http://www.bpmg.org/downloads/Articles/Article-MEGA-BusinessProcessModeling&StandardizationEN.pdf>. (2004).
  30. Mouratidis, H., Giorgini, P. and Manson, G. A.; *Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems*, Advanced Information Systems Engineering, 15th International Conference, CAiSE 2003, Klagenfurt, Austria, June 16-18, 2003, Proceedings. Vol. 2681. (2003). pp.63–78.
  31. 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.
  32. Object Management Group; *MDA Guide Version 1.0.1*. In <http://www.omg.org/docs/omg/03-06-01.pdf>. (2003).
  33. 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).
  34. OMG; *Object Management Group*. In <http://www.omg.org/>. (2004).
  35. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 6th Edition, (2006). 880 p.
  36. Quirchmayr, G.; *Survivability and Business Continuity Management*, ACSW Frontiers 2004 Workshops. Dunedin, New Zealand. (2004). pp.3–6.
  37. 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.
  38. Roser, S. and Bauer, B.; *A Categorization of Collaborative Business Process Modeling Techniques*, 7th IEEE International Conference on E-Commerce Technology Workshops (CEC 2005). Munchen, Germany. (2005). pp.43–54.
  39. 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.
  40. Sparks, G.; *An Introduction to UML, The Business Process Model*. In [http://www.sparxsystems.com.au/WhitePapers/The\\_Business\\_Process\\_Model.pdf](http://www.sparxsystems.com.au/WhitePapers/The_Business_Process_Model.pdf). (2000).
  41. 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).
  42. 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.
  43. Zuccato, A.; *Holistic security requirement engineering for electronic commerce*, Computers & Security. Vol. 23 (1). (2004). pp.63–76.
  44. 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.