

A Method for Security Governance, Risk, and Compliance (GRC) A Goal-Process Approach

Yudistira Asnar and Fabio Massacci

Department of Information Engineering and Computer Science
University of Trento, Italy
firstname.lastname@unitn.it

Abstract. The Governance, Risk, and Compliance (GRC) management process for Information Security is a necessity for any software systems where important information is collected, processed, and used. To this extent, many standards for security managements at operational level exists (eg ITIL, ISO27K family etc). What is often missing is a process to govern security at organizational level. In this tutorial, we present a method to analyze and design security controls that capture the organizational setting of the system and where business goals and processes are the main citizen. The SI*-GRC method is a comprehensive method that is composed of i) a *modeling framework* based on a requirement engineering framework, with some extensions related to security & GRC concerns, such as: trust, permission, risk, and treatment, 2) a *analysis process* defining systematical steps in analyzing and design security controls, 3) *analytical techniques* to verify that certain security properties are satisfied and the risk level is acceptable, and at last 4) a *CASE tool*, namely the SI* Tool to support analysts in using the method. To illustrate this method, we use a running example on e-Health adapted from a real-life process in an hospital partner.

1 Introduction

The last decade has seen two parallel and conflicting demands on businesses services: business processes are increasing in complexity and unpredictability, while demands for accountability, regulatory compliance and security are becoming mandatory.

Thus, a structured approach to Governance, Risk and Compliance (GRC) of Information Security has become a high priority goal [1]:

- *Governance* is the set of policies, laws, culture, and institutions that define how an organization should be managed (as opposed to the way it is actually managed);
- *Risk Management* is coordinate activities that direct and control an organization forecasting and managing events/risks that might have a negative impact on the business;
- *Compliance* is the act of adhering to regulations as well as corporate policies and procedures.

GRC products support the achievement of compliance by automatic monitoring of controls across different applications, offers audit facilities, the prioritization of corrective actions according risk based profiles [2]. Traditional audit methods involve the

review of transactions at a given time (i.e., sampling). However, this approach does not give sufficient assurance over the level of security and compliance of business processes over an entire period, and more advanced GRC products (e.g. SAP and Oracle Financials) offer Continuous Controls Monitoring techniques. Tools such as Audit Command Language [3] are becoming increasingly popular as cheap and flexible options for the implementation of continuous controls.

GRC solutions for technical systems are well understood (even if not thoroughly deployed) and a number of textbooks clearly identify the desired security properties. For example in [4] we find confidentiality, availability, and integrity of computer-related assets, and it also includes possession/ownership, utility, authenticity, non-repudiation, and privacy of information. In addition to those qualities, compliance with specific regulations might require some additional qualities, such as accountability (SoX, Basel II), assurance (SAS 70, CMMI, CommonCriteria), etc.

However, it is becoming increasingly evident that managing security and GRC of a system is not only a matter of technical solutions. Most of the state of practices (e.g., ISACA RiskIT[5], COBIT [6]) define risks as the obstruction originated from the availability of resources (e.g., documents, servers, production machines). However, the availability of resources is not sufficient to guarantee the security or GRC of a system, because sometime a disruption is originated from the process or even the objective level of the system. In other words, security violations might be originated from any layer of the organization (i.e., strategic, process, infrastructure, information).

The focus of this work is to consider as a “system”, a more complex construction and namely a *socio-technical system* [7] where both social and technical aspects, and its relations are considered as an integrated whole. The method presented in this tutorial distills a number of research articles and empirical studies of the University of Trento and a number of colleagues from industry and academia (See the acknowledgments). It analyzes security and GRC aspects of a socio-technical system and focuses on the organization’s goals and processes of the system. The method as a whole [8] assumes that a security initiative is a long-life program that an organization needs to carry on (i.e., not a project-based initiative) based on the Deming management cycle of plan-do-check-act [9]. In other words, one needs to consider how to monitor and continuously improve the implemented controls in the system.

Here we put the emphasis on the *Plan* phase where analysts model the target system and analyze the security GRC issues and design security mechanisms to manage the excess risk. Table1 shows the basic checklist for the method.

In the following sections, we demonstrate how the SI*-GRC method is used to analyze and design security and GRC concerns of the information system used in a rich scenario from e-Health (§2). We then we provide an overview of the method describing who the participating roles are, what the steps are, and what is contained in the models of the method (§3). We explain in details how the method support the Security and GRC of the system during the Plan phase. We start by describing how to capture a target of analysis the “SI*-GRC” way (§4), then how to analyze security and GRC concerns (§5), and finally how to define the required controls (§6). We discuss briefly the technique for in-vivo monitoring and unpatch of the controls (§7) and conclude.

Table 1. SI*-GRC Method Checklist.

<ul style="list-style-type: none">– The <i>Role Identification and Organizational Set-up</i> specifies the roles and responsibilities for the SI*-GRC team, the business process owners and the IT management.– The <i>Specification of the Target of Analysis</i> captures at high level the socio-technical system with the key relevant information. The analyst should quickly identify the following aspects:<ul style="list-style-type: none">• <i>Actor Models</i> capture relevant actors (i.e., social and technical ones) their organizational structure.• <i>Goal Models</i> capture the objectives of the organizations by spelling out the intentions and capability of identified actors and the functional interdependency between actors and goals.• <i>Process and Services Maps</i> identify the processes which achieve the organization’s goals. In this model, one can see the relation between how a goal is being carried out by a business process, and how the business process is supported by series of business services and/or resources. An organization typically has formalized its business process in some formalization (e.g., BPMN, Flow-chart, UML Activity Diagram). This formalization should be re-used as much as possible.– The <i>Security and Risk Analysis</i> captures which aspects of the system are not protected<ul style="list-style-type: none">• <i>Business Continuity Critical Points Identification</i> specifies the business processes and actors whose commitment is essential to achieve the high-level goals of the organization.• <i>Unauthorized Processing Identification</i> identifies existing permissions (if known) and how they are delegated to other actors. Provide as output potential unlawful processing and over-entitlement scenarios.• <i>Trusted Computing Base Identification</i> identify the trust relationships between actors in possibly making references to specific goals. Provided as output the boundaries of the TCB that can be sources of potential failures of reliability and misuses of permission in terms of scenarios for failures of the organizational goals.• <i>Unwanted Scenario Identification</i> supports the analysts in identifying threat, events and gives guidance how to structure them at resource, process or strategic level.• <i>Risk Assessment</i> estimates the risk level of identified risks and specifies which risks will be treated or accepted– <i>Control Analysis</i> identifies the control mechanisms put in place to address the risks.<ul style="list-style-type: none">• <i>Control Goal Model</i> specifies and elaborate the control goals in order to cover most (if not all) risks, with appropriate and precise measures.• The <i>Control Processes and Services Map</i> identifies the control processes that achieve the control goals.
--

2 The Running Example Scenario

The scenarios used in this article are adopted from the *drug reimbursement* process in a major hospital in Italy [10]. This process is only applied for a specific set of drugs (called File F drugs) to be delivered to outpatients, patients in “day hospital” regimen, and patients during the last day of hospitalization. This list includes innovative and expensive drugs, caring chronic diseases with a high social impact (tumors, multiple sclerosis, HIV, etc.).

The economic aspect of this business process is particularly relevant as the regional wide expenditure for these drugs is raising year by year (in +12% in 2006, +10% in 2008), bringing to an augmented attention of the Health Authority toward the existence of an effective system of controls over the dispensations of these drugs ¹.

As a consequence, this business process is subject to significant security and compliance requirements: all regulations ² of the File F Regulatory Framework have to be respected by the hospital in order to obtain the reimbursement from the regional health authority. Moreover, since the drug reimbursement process includes the processing of personal and sensitive data (the recipients of the drugs), it must also satisfy the regulations on privacy from the Legislative Decree no. 196 of 30.06.2003 called "personal data protection code", in addition to well-established security requirements (e.g., non-repudiation of a drug dispensation, availability of the prescription data, integrity of the drug dispensation process) In nutshell, the *Drug Reimbursement* process is a mechanism that allows the hospitals to refund the costs of the dispensed File F drugs. The process is composed of three macro-phases: Drugs Prescription, Drugs Dispensation, Generation & Sending File F reports to the Health Authority.

In the *Drugs Prescription phase*, a prescription is done by a doctor to a patient using an IT system. The doctor selects File F drugs to administer to the patient and of their posology and quantity. The selection of the drugs can be done by copying the past prescriptions (in case of a chronic diseases) or also choosing them from a complete list of File F drugs. Finally, a prescription sheet is produced containing the patient's name and other personal data (e.g. the Fiscal Code), and also sensitive data (e.g., the patient disease, the prescribed drugs and their quantities/posologies).

In the *Drug Dispensation phase*, dispensers (doctors or nurses) dispense File F drugs following a prescription given to a patient. At this phase the dispensers need to indicate which drugs, from the prescription, are being dispensed, since ones might dispense a part of the prescription for some reasons (e.g., the drug is currently unavailable). In some occasion, some drugs appear to be unavailable in the computer system though in reality they are available in the hospital ward drug stock. In this case, the dispensers must dispense the drugs because it is critical for the patient's safety. Finally, a dispensation sheet is printed and needs to be signed by the dispensers and the patients. This sheet contains similar information as the prescription except the patient disease.

The *Report Generation phase* is constituted two parts: 1) the generation of the File F reports to be sent to the Health Authority; and 2) the sending of the File F reports to the Health Authority. In the first part, the Accounting Office retrieves all the data about the dispensed drugs in a month, checks the data, produces the File F reports, and sends them to the Hospital Medical Director. In the second part, an operator from

¹ The Regional Directive n. 5743 - 31.10.2007 provides indications to optimize and improve the process design about prescription/dispensation/accounting of File F drugs, and the Regional Directive VIII/1375 - 14.12.2005 stresses the priority to implement actions towards the verification of the appropriateness of the use of File F drugs.

² To have an idea, without mentioning privacy requirements, the File F mechanism was instituted by the regional circular 17/SAN 3.4.1997, and successively has been emended by the Circular No.5/SAN 30.1.2004, by the Circular No.45/SAN 23.12.2004, by the Note 30.11.2007 H1.2007.0050480, by the Note 27.3.2008 H1.2008.0012810 and by the Note 04.12.2008 H1.2008.0044229.

the Medical Director verifies the File F reports and send them to the Health Authority supervised by the head of department. All the operations are assisted by the IT system and the File F reports have to be sent to the Health Authority following a special guideline, namely the “debito informativo” IT channel, that permits to send and receive information through common e-mail clients with additional security mechanisms that guarantee the confidentiality of the exchanged data (encryption of the message and of the attachments, digital signature, etc.). It is another security & compliance requirement posed in the information system.

For technical details, we assume the drug dispensation process is supported by an information system implemented in terms of services, in SOA [11] sense. Still, we believe this method is applicable for other technical architecture (e.g., object-oriented, web-based)

3 Overview of the Method

This section illustrate the SI*-GRC method at high-level, starting from the *organizational context* to be prepared before using the method, the conceptual *modeling* used in the SI*-GRC, namely the SI* modeling framework, and finally illustrate the SI*-GRC process that will be detailed in the following sections.

3.1 Roles and Organizational Set-up

In order to manage security and GRC, the preliminary organizational steps are essential to ensure that security & compliance is achieved effectively across the organization and in a timely manner. More importantly, they aim to obtain management buy-in, to clarify the ownership of a process & the responsibility of controls, to align the organization’s strategic objectives with regulatory requirements, and above all to maintain security across system evolution.

Often such aspect is usually neglected in research papers on information system engineering (e.g., [11,12]) which directly focuses on the functional design of the system. The preparation steps are the following ones:

1. The **Identification or set-up a GRC council** ensures the periodic review of the organization’s security policies and procedures; the level of compliance with these policies and procedures; the level of exposure of key information assets during implementation; the ongoing improvement of existing control implementations (i.e., control processes and indicators); and the availability of sufficient resources for SI*-GRC. In other words, these actors are essential staff members during the *Check* and *Act* phase.
2. the **Definition of roles and responsibilities for the SI*-GRC team**. Actors that make up the analysis-design Council: (i) management (e.g., members of the Board of Directors, the Chief Executive Officer (CEO), and Shareholders); (ii) business process owners (key personnel that are responsible for the daily operations of the concerned business process); and (iii) IT management (those responsible for the governance, daily operations and maintenance of the IT infrastructure).

The followings are actors involved in the design of SI*-GRC controls:

Business Analysts are those responsible for the analysis of the organization's business goals & processes;

Risk Analysts are responsible to assess the level of risk of some threats to the business, and decide whether they are acceptable or not;

Security Analysts are responsible for the design and implementation of actions aimed at providing a reasonable level of assurance concerning compliance with security policies. Security analysts generally make part of IT management;

SI*-GRC Analysts are proficient with the method that can coordinate the analysis and design process of control processes and indicators. Moreover, the SI*-GRC analysts are responsible for ensuring that the SI*-GRC controls and indicators are implemented according to the designs and provides an appropriate/improved level of assurance concerning the achievement of strategic objectives.³ Note the performance of these actors is critical for the success of security & GRC initiatives.

3. The **creation of a timeline for the SI*-GRC deployment**, indicating the key goals and milestones of the SI*-GRC application, as well as those responsible for each milestone. The timeline should list all tasks to be carried out during the SI*-GRC application including decision points.

3.2 Process Overview

The SI*-GRC method includes a description of systematic steps to support the *Plan-Do-Check-Act* (PDCA) of the Deming Cycle [9] to ensure the continuous monitoring and improvement of the secure system as depicted in Figure 1. The *Plan* phase aims at capturing and analyzing the problems in the target system and at the end results in design of necessary controls. The *Do* phase concerns on the implementation up to the execution of such controls. The *Check* phase aims at reviewing the implemented controls and the existing organization context. Finally, analysts needs to re-act upon the review results to improve the security and GRC of the system at the *Act* phase. These reactions must be planned before they are implemented in the system.

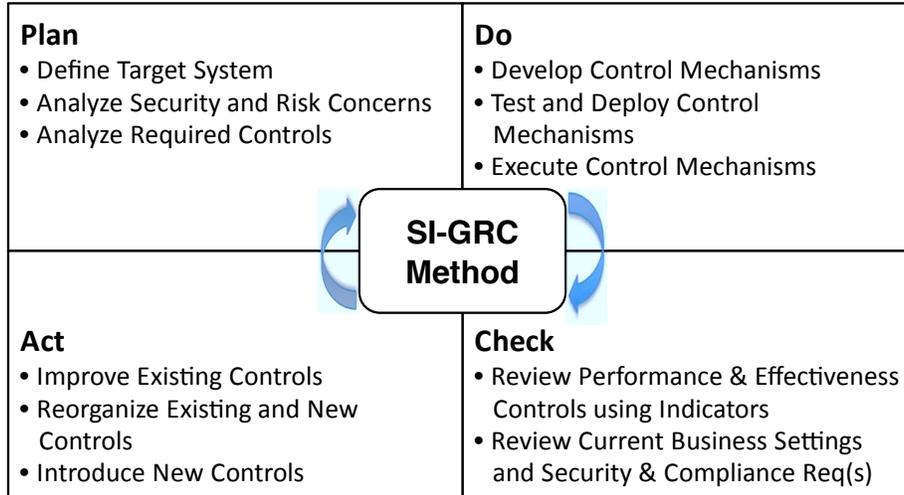
The general steps of the *Plan* phase are depicted in Figure 2, where includes

1. *Target System Definition* where an information system is formalized in the context of an organization;
2. *Security and Risk Analysis* where and it captures and analyzes security and GRC aspects (i.e., including trust and permission) and then assess the possible risks;
3. *Control Analysis* where some necessary controls are formalized including their quantitative indicators indicating their effectiveness and performance.

The *Target System Modeling* step covers three modeling activities. Firstly, the *Actor Modeling* captures actors (i.e., human or technical) involve in the system including their structure in the organization. Analysts then capture and analyze the strategic interests in

³ In this work, we assume an improvement at the information security, governance, compliance, and risk management will lead to the better assurance of the business

Fig. 1. The Plan-Do-Check-Act Cycle of the SI*-GRC Method

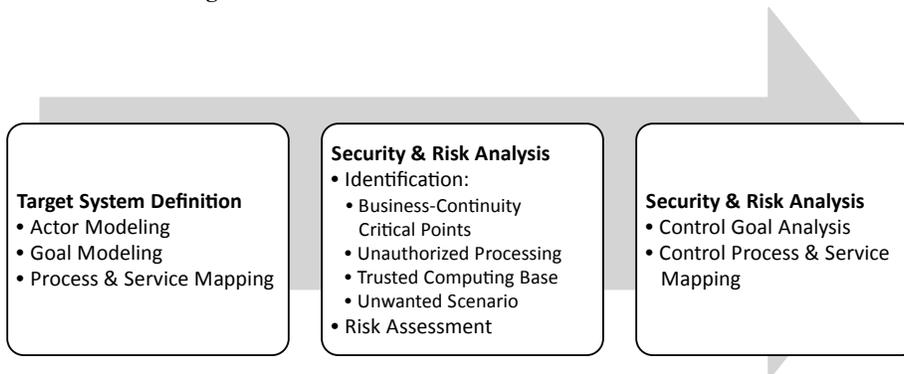


the *Goal Modeling*. Moreover, analysts need to capture strategic rationales and dependencies of an actor [13] and each capabilities. To achieve those interests, an organization defines systematic steps (i.e., a business process). In our method, we use BPMN [14] to capture the business process though we believe other process models (e.g., YAWL [15]) will also be applicable. The *Processes and Services Mapping* indicates which process satisfies which goal. Moreover, in the business process one must indicate which resources (i.e., data object in BPMN or underlying business services [11]) involve in the execution of business process.

At *Security Analysis* step, it composed of four activities:

1. *Unwanted Scenario Identification* which identifies potential events

Fig. 2. The SI*-GRC Process Overview for the Plan Phase



2. the *Unauthorized Processing Identification* models actors' entitlements and the delegation permission to another actor, the
3. *Trusted Computing Base Identification* where capture the trust relationship between actors involved in the system,
4. the *Risk Assessment* needs to measure the level of risk for the high-level goals of the target system after considering its trust and permission models, and finally decides which risks that are unacceptable therefore require some controls to mitigate them.

The *Control Analysis* aims at analyzing further the control mechanisms. It starts from the *Control Goal Establishment* describing the objectives of the controls. The *Control Goal Analysis* specifies and elaborate the established control goal so that they cover most (if not all) risks, accurate, and have a clear-cut definition of their fulfillment. At last, the analysts need *Control Process and Services Map* which details the means to achieve a control goal. This map is very similar to the one used for business process and business goals excepts that it aims at protecting the business process from the excess risks by achieving the control goals.

The final outcomes of the SI*-GRC for the plan phase are a list of detailed control mechanisms composed of control goals describing the state-of-affairs to be achieved/protected, a process description of how those goals are implemented in term of control processes, and a series of indicators indicating their effectiveness and performance, and the organization's trust and permission model.

Example 1. Based on the scenario given in Section 2.

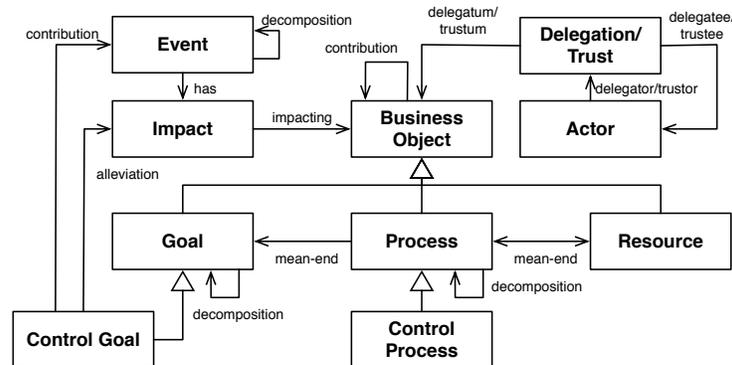
- Control Goal - File F reports must respect the privacy preference defined by a patient;
- Control Process - Anonymize the name of patient that requests to, after File F reports are generated;
- Indicators
 - The number of privacy litigation coming from patients – effectiveness;
 - The number of File F records not being anonymized where the patient requests to be anonymized – performance;
 - How often the anonymization process is executed – performance;
 - How many records have been anonymized – performance.
- Trust Model - Accounting Officers trust that dispensers will be honest in inputting dispensation data;
- Permission Model - Doctors have permissions to access a patient's medical info.

More details will follow in the incoming sections (i.e., Section 4-6); and for other phases (i.e., *Do*, *Check*, and *Act*, we only illustrate the usages of the method in-vivo of the information system (Section 7).

3.3 SI* Modeling Framework

The SI*-GRC method uses the SI* modeling framework [16] for most of its modelings except the modeling of business process where uses BPMN. The SI* framework is a modeling framework extending the *i** framework [13] to support security requirement analysis. In Figure 3, The conceptual model of this modeling framework is based on basic concepts representing requirement analysis

Fig. 3. The SI* Conceptual Modeling



- Actor** is an autonomous entity that has its own intentions (human and software), capabilities, and entitlements. This concept is then realized as ;
- Goal** is a state-of-affair that an actor intends to achieve;
- Process** is a means to fulfill a goal, to furnish a resource;
- Resource** is an artifact that is consumed/produced by a process;
- Event** is an uncertain circumstance that affects a goal satisfaction (in/directly);
- Trust** captures a believe of the capability/honesty of one actor (trustor) to another actor (trustee) in fulfilling/using a business object (trustum);
- Delegation** depicts a transfer of the responsibility/right from one actor (delegator) to another actor (delegatee) in fulfilling/using a business object (delegatum).

● Shall we talk about the control framework???

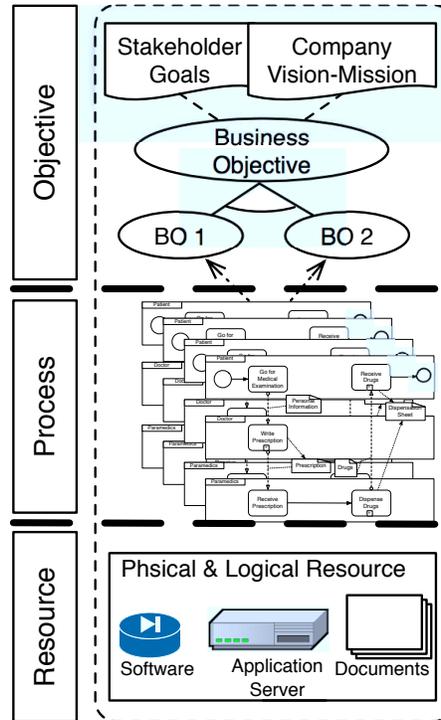
The Appendix A lists all constructs (i.e., concepts and relation) in the SI* Framework including the graphical diagrams and DLV predicates. Details relations between constructs will be explained and illustrated along the analysis and design process. To avoid confusion, we use an *italic* text to indicate a SI*-GRC basic concepts and a **sans-serif** text for the one related to the scenario.

4 Target System Definition

The first step of the SI*-GRC process is modeling the target system where one needs to analysts its security and GRC concerns. In some literature in the organization behavior [17], management information system [18], and enterprise architecture [19], one can organize an information system in an enterprise into three levels of abstraction as depicted in Figure 4. Essentially, an information systems always serves the *objectives* of the stakeholders (i.e., including the organization’s shareholders). These objectives is implemented by series of business *processes* that are inter-related one to another. To execute such processes, participating actor might need some *resources* either physical or logical ones.

To capture such settings, analysts need to define the *Target System* in terms of the structure of actors in the organization, their interests and capabilities, and inter-dependencies between them. Finally, we need to map which process is associated to

Fig. 4. Three Layer Model of an Information System in an Enterprise



which stakeholders' goal. The overall process of this phase is described in Figure 5. Note that in the SI*-GRC method analysts might choose any approach for a *Process Modeling* in the organization, such as Event-Driven [20], Workflow (e.g., BPMN [14], YAWL [15]), UML-based [21], Declarative BPs [22], or Case-base [23].

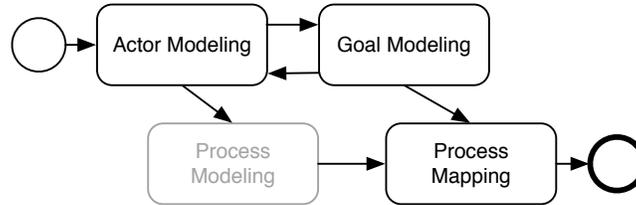
4.1 Actor Modeling

First task towards modeling the target system is the identification of the key social actors whose intentions need to be fulfilled by the system or are mostly involved in the system.

Depending on the level of analysis one might consider and analyze strategic interests of social actors [24] or technical actors (e.g., email service, data management agent) and analyze the social actors' goals that are delegated to the technical actors. For a comprehensive analysis we need to consider most actors involved in all phases of the system existence (if relevant) and in particular

- *subject actors*, whose information will be stored/processed by the information system - e.g., Patient;
- *usage actors* will use/consume the information produced/managed by the information system - e.g., Pharmacy, the Health Authority, the Auditor, the Accreditation Office (e.g., Joint Commission);

Fig. 5. The Process Model of the Target System Definition Step



- *system actors* who are part of the information system to use and maintain it - e.g., the IT departments, the Risk & Compliance Office, business analysts, and all actors in Figure 6;

Analysts need to be not confusing between role or agent. A *role* is an abstract characterization of intentional entity's behaviors, and an *agent* is an intentional entity with a concrete manifestation. In other words, one can distinguish an *agent* as an object level entity that has one-to-one correspondence with an entity in the system world; while *role* as the class actor needs to be distinguished into *role*

Example 2. In our scenario, the Hospital "H", the healthcare authority, Dr. Dave, Alice, and the Head of Hospital are considered as *agent*; while doctor, nurse, dispenser, and accounting officer as *role*.

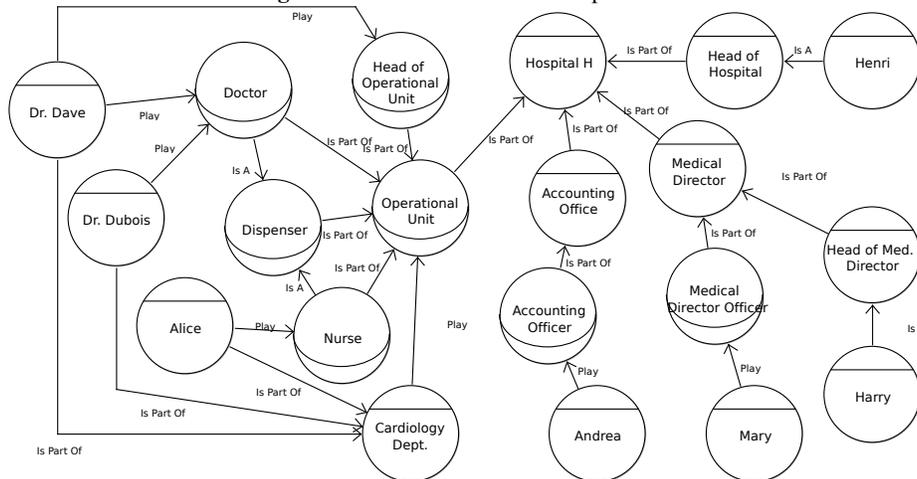
Given a series of actors, analysts needs to structure them using the following relations:

1. *play* indicates an agent that plays a particular role;
2. *is_part_of* indicates an actor that is part/member/composed of another actor;
3. *is_a* captures a specialization relation between two actors (i.e., role-to-role or agent-to-agent);
4. *supervise* captures a supervision relation between two actors (i.e., role-role or agent-agent).

Example 3. In Figure 6, we model the Hospital "H" that is composed of a series of operational unit (e.g., Cardiology, Gynecology, etc.), an Accounting Office and a Medical Director; leads by a head of hospital Mr. Henri. Dr Dave and Dr Dubois *play* a role of doctor, while Alice *plays* a role as nurse and dispenser. All these agents are member of (*is_part_of*) the Cardiology Department. Since Doctor and Nurse are (*is_a*) dispenser, all these can act as a dispenser in the Cardiology Department.

These actors are essential to set up the scope of the following modeling activities. Note in Example 3, we only consider actors that are part of the drug reimbursement system.

Fig. 6. The Actor Model in the Hospital “H”



4.2 Goal Modeling

For a given actor model, analysts need to identify the strategic interests (goals or objectives) that the actors intend to achieve and model using a **Request** relation depicted in Figure 7. These cover responsibilities, motivations, and intentions of actors in the organization and business objectives of the organization.

Example 4. The patient needs to get medical service and the hospital interest to obtain the reimbursement of dispensed drugs.

In particular to the context of security and GRC, one considers also the high level norms imposed by the organization information policy or by the standardization body or by the regulator.

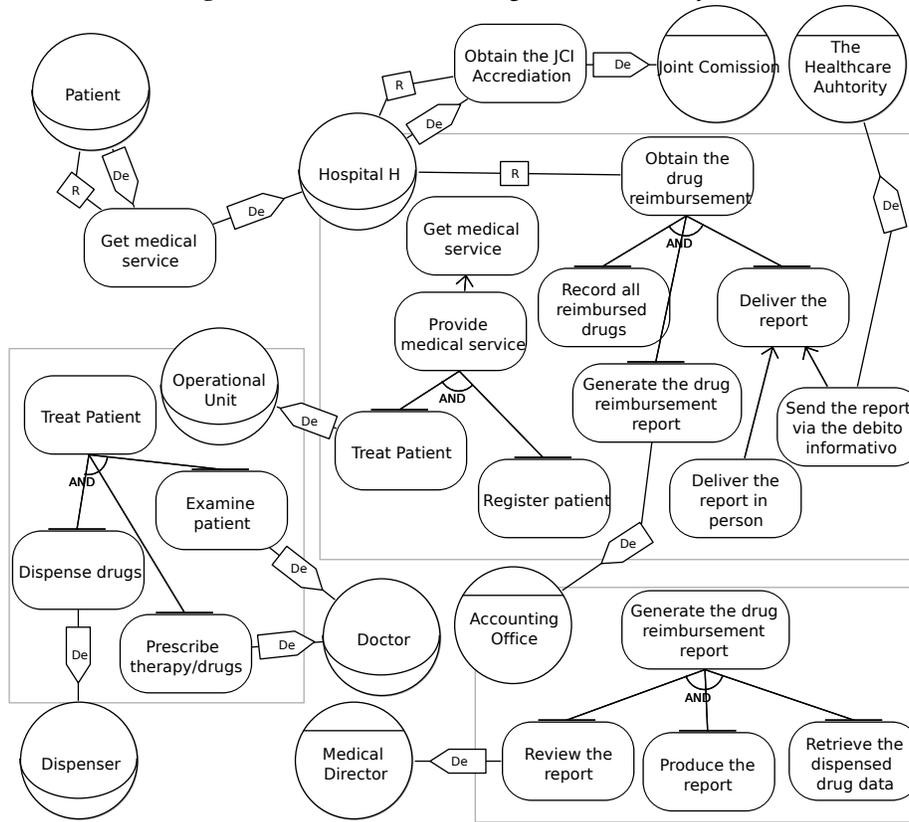
Example 5. The hospital “H” management intends to keep the all medical records in two modalities (e.g., paper and digital) for the next 3 years;

The Joint Commission (a standardization bodies) requires an hospital to initiate and maintain all clinical records of every patient assessed or treated (MCI.19) - completeness, sufficiency, integrity, and accountability; while MCI.10 & MCI.11 require information privacy & security are maintained;

- The health authority, in Regional Circular No.5/SAN 30-1-2004, obliges an hospital to follow a strict guideline to send a drug reimbursement report in to a defined format (e.g., three files - personal data, drug info, drug cost) delivered via a special channel, namely the “debito informativo” channel.

Often an actor cannot achieve its goal by itself therefore one might 1) appoint another actor to full the goal using a *delegation of execution*, or 2) decompose the goal further into more detail/precise subgoals using an *AND decomposition* relation and assign parts of them to other actors. AND-decomposition indicates that all subgoals must

Fig. 7. The Goal Model of the Drug Reimbursement System



be fulfilled by the system in order to achieve the main goal. Sometimes it is useful to specify alternatives to fulfill a goal using several *means-end* relations indicating alternatives to fulfill the goal. This decomposition is rarely used in practice during an audit or assessment step of existing security mechanisms. However, it is mostly used during the plan phase of a new security mechanism.

Example 6. To obtain a drug reimbursement, the hospital “H” needs to do all these (AND-decomposition): record all reimbursed drugs, produce the drug reimbursement report, and deliver the report. To achieve the goal of deliver the report, the hospital can *send the report via the “debito informativo” channel* managed by the Healthcare Authority or *deliver the report in person*

This analysis is an iterative process, where sub goals need to be refined further with AND-decomposition/means-end relations or being delegated its fulfillment to another actor. This decomposition stops for an actor when all leaf goals (i.e., the lowest sub-goals) are have being delegated to other actors via *delegation execution* or because the actor can fulfill the subgoal by itself.

Later in the analysis of the target system we might refine further this capability by specifying a business process that achieves that. Otherwise, we do not care how the goal is actually achieved (but we know it will be achieved) we can simply mark this assumption in the model. Notice that this is an important assumption that would need to be validated at some point as it enters in the trust model of the system.

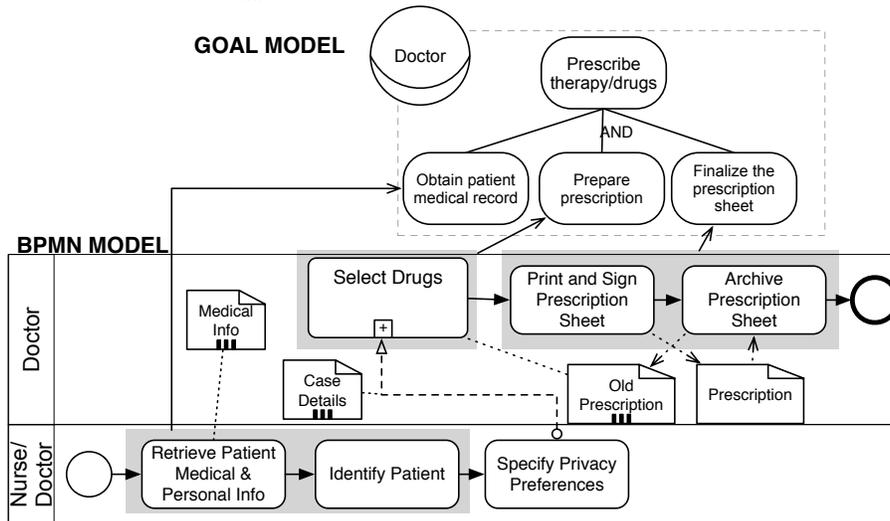
Example 7. For achieving the goal of generate the drug reimbursement report, the hospital delegates it to the accounting office. However, the accounting office is only capable to retrieve the dispensed drug data and produce the report, but it still needs to delegate further a part of the goal, namely the subgoal review the report to the medical director.

The final outcome of this modeling is a goal model, as in Figure 7, which captures a network of actors' dependencies on fulfilling their goals and the rationales of such goals.

4.3 Process and Services Mapping

As mentioned earlier, SI*-GRC allows analysts to use an existing formalization for the process modeling. The aim of this modeling to indicate which process, a part of the business process, is a means to fulfill actors' goals. In the goal modeling, analysts have identified the leaf-goals that actors are able to fulfill; and in the process mapping, analysts map those leaf-goals to the processes using *means-end* relations in which their executions will fulfill the leaf-goals as depicted in Figure 8.

Fig. 8. The Process Mapping in the Prescription Phase of the Drug Reimbursement Process



Example 8. To achieve the goal prescribe therapy/drugs, a doctor needs to fulfill three subgoals as depicted in Figure 8. To fulfill obtain patient medical record, a doctor/nurse needs to perform two activities retrieve patient medical & personal info and identify patient consecutively. Moreover, those two activities can also be performed by a nurse.

By means in this mapping, analysts indirectly define which actors have capabilities to perform particular processes/activities (including sub-process or task in BPMN) to fulfill a goal. Moreover, it can also infer which actors can provide/require particular resources (i.e., data object in BPMN) from which activity produces/consumes some resources. In a SI* model, these capabilities are captured with *Provide* relations.

Example 9. Based on Figure 8, we can infer the capabilities of a doctor and a nurse that can perform the activity of retrieve patient medical & personal info and *provide* the resource of medical info.

As mentioned before, sometime analysts might prefer to indicate actors' capabilities at the high-level (i.e., goal) without caring how it is actually achieved. As we shall see later this is an important trust assumption.

Example 10. A doctor can *provide* the fulfillment of the goal examine patient.

From Figure 8, the complete capabilities of each actor from are listed in Table 2.

In SI*-GRC *resource* is not solely data-related artifacts, but it expands into other physical and logical resource, such as: prescription & dispensation sheet, underlying business web services that supports the execution of some activity, softwares used to process some data while performing some activity, and infrastructures used to collect, store, and transmit the data during the process execution. At the end of this mapping, analysts have completed a three-layer model of the enterprise information system as depicted Figure 4.

Table 2. Actors' Capabilities on the Drug Reimbursement System

Capability - <i>Provide</i> relation	Doctor	Nurse
<i>Performing an Activity/Process</i>		
• Patient Medical & Personal Info	X	X
• Identify Patient	X	X
• Specify Privacy Preferences	X	X
• Select Drugs	X	-
• Print & Sign Prescription Sheet	X	-
• Archive Prescription Sheet	X	-
<i>Providing a Resource</i>		
• Medical Info	-	-
• Case Details	X	X
• Old Prescription	X	-
• Prescription	X	-

5 Security and Risk Analysis

After defining the target system, analysts need to analyze the security and GRC concerns of the system. In SI*-GRC we consider a whole system as an asset, something valuable, therefore we need to protect it from unacceptable events that might harm the security and compliance requirements of the system.

This step aims to analyze security and GRC concerns of the system; it starts by identifying critical points that can disrupt the business continuity. Analysts then identify the ownership of processes and resources in the system, and delegation its permission to other actors, and defining the trust relations between actors exist within the organization. Given such settings, we might identify several unwanted events (i.e., threats, failures, errors) that can compromise the security & GRC of the system. To decide which unwanted events need to be controlled, analysts need to estimate the level of risk of such events, and finally take into account the organization policies on controlling them (e.g., risk tolerance, ethics, IT architecture policy).

5.1 Identification of Business Continuity Critical Points

The outcome of the target analysis phase is a complex web of relations between actors and business objects. The preliminary analysis of this model is to ensure that all critical points for business continuity have been identified.

Intuitively, a business continuity disruption happens when an intention to fulfill a goal (or a part of goal) is passed across actors using delegations of execution relations and it ends to an actor who does not have explicitly the capability to fulfill the goal (or part of it) in term of performing a process or providing a resource.

This property can be easily visualized on the model as a path that does not end with either a *means-end* relation into a business process or with an atomic *provide* relation between the last actor in-charge and the goal.

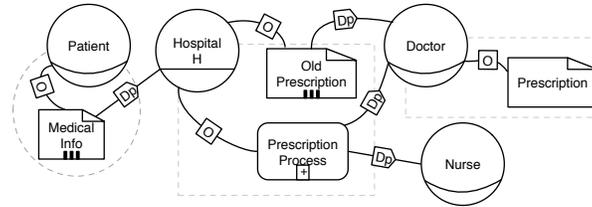
Example 11. In Figure 7, the Hospital “H” need to delegates the goal **register patient** to the Operation Unit. Unfortunately, that actor, including any actor part of it, does not have necessary capabilities to fulfill the goal, and not even pass the goal to another actor.

This potential disruption can be resolved by reorganizing the delegation of execution to end with a specific business process or by allocating to actor that are explicitly declared to be capable of achieving the desired goal (i.e., outsourcing might be the least thing that an organization can do, when none in the system is capable).

In the former case we have specified how the goal will be actually achieved in the system and the business process itself is a critical point for business continuity. In the latter case we have only specified that an actor is in charge of this goal which therefore makes the behavior of this actor critical for the overall achievement. This case can be later analyzed by considering possible unwanted scenarios that lead to the disruption.

Analytical techniques are available with the SI* tool. The papers [16,25] describe the formal properties in more details.

Fig. 9. An Example of the Permission Model in the Prescription Phase of BP



5.2 Unauthorized Processing Identification

So far we have modeled the transfer of responsibility via *delegation execution* due to actors' capabilities. However, this concept does not capture the notion of *ownership* of a particular process or resource. Sometime to execute a process or to access a resource, an actor needs to have a permission from the owner of process/resource.

In this modeling phase, we capture the notion of *ownership* of process/resource and the notion transfer of rights on a business object from one actor to another using a *delegation permission* relation. The **Owner** relation depicts a relation between an actor to a process/resource that indicates the actor has full authority concerning the execution of a process or the access of a resource.⁴

Example 12. The patient owns the resource of its medical info; and the Hospital "H" has a full authority for all activities in the prescription process depicted in Figure 8.

Typically, defining the owner of a resource is simple because a resource is tangible and often some policies/regulations already define the notion of ownership (e.g., EU Data Protection Directive-Directive 95/46/EC or the company property policy). However, it is less trivial for the process ownership.

However, in many situation the owner of such process/resource needs to delegate the permission to another actor because the owner does not have capability to execute the process or the other actor needs to access the resource/to execute the process to fulfill its responsibilities. For such situation, the SI* models the action of giving a permission using a *delegation permission* relation.

Example 13. The Hospital "H" delegates the permission to doctor on executing activities in the prescription process (as in Figure 8) because the hospital "H" needs the doctor to fulfill the goal *prescribe therapy/drugs* that at the end fulfills the top level goal *provide medical service*.

The final result of this modeling activity is represented in term of a SI* model as depicted in Figure 9. This model can then be used to identify more precisely two possible threats which are very important for compliance:

⁴ Process Owner is defined as an actor that is held accountable and responsible on the performance and improvement of a defined process (including its subprocesses) in <http://www.gao.gov/special.pubs/bprag/bprgloss.htm>

- *lack of Authorization* (or unlawful processing) is present when an actor has been assigned a process or managing a resource without a proper authorization path stemming from the owner. This aspect is particularly critical when demonstrating compliance with privacy legislation;

Example 14. In Figure 9, the doctor does not have authorization to access the **medical info** that is required to perform the **prescription process**

- *Over-entitlement* when an actor has been delegated the permission to access a resource or to execute a process but the latter is not required to achieve the goals assigned to the actor. At minimum this might be a violation of the minimal disclosure principle for the compliance with privacy legislation or might be the source of more serious troubles if the actor can potentially misbehave (e.g., fraud, internal-trading).

Example 15. In Figure 9, the nurse have permissions to execute all activities in the **prescription process**, though in practice the nurse only responsible to perform the **retrieve patient medical & personal info activity** and the **identify patient activity**.

These properties can be easily visualized on the model as paths. Intuitively, a unlawful processing for a key business objectives is present when there is a path across decompositions and delegations of execution that arrives to an actor. Yet this processing actor, in order to achieve its delegated subgoal make use of some process or some resource whose owner has not delegated the authorization directly or indirectly to the processing actor. This latter properties is simply the absence of some path from the owner to the processor.

Also in this case analytical techniques are available with the SI* tool vulnerabilities and threats of a SI* model formalized in Answer-Set Programming (ASP) [16,25].

These potential threats should then be considered in the final mitigation reports because both threats introduce various type risks to the system's security, governance, and compliance.

5.3 Trusted Computing Base Identification

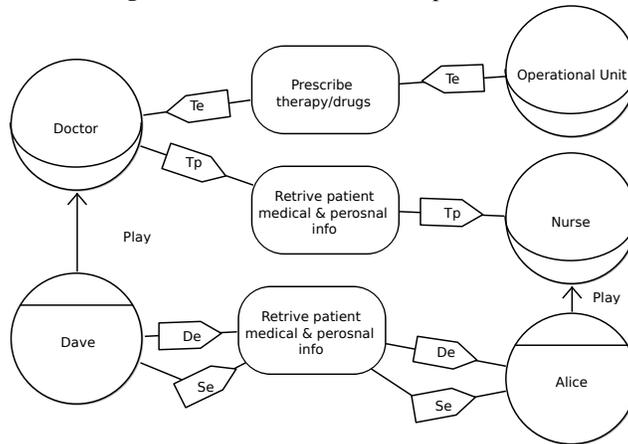
Trust, in SI*-GRC method, captures a social relationship that indicates the belief of one actor (*trustor*) on another actor (*trustee*) capability or honesty. The SI*-GRC method distinguishes two types of trust relation:

- *Trust of execution* represents the trustor's expectations concerning the ability of the trustee accomplishes the trustum (i.e., achieving a goal, executing a process, delivering a resource);

Example 16. The operational unit trusts the doctor on fulfilling the goal of **prescribe therapy/drugs**

- *Trust of permission* models the trustor's expectations that the trustee is honest/does not misuse the permission on the trustum (i.e., fulfillment of a goal, execution of a process, access to a resource). In other words, by trusting in permission, the trustor believes that the trustee will not use the given permission for accomplishing a purpose different from the one for which the permission has been granted.

Fig. 10. Trust Model in the Prescription Phase



Example 17. The doctor believes that the nurse will not misuse the permission to retrieve patient medical and personal info

A frequent error in this analysis phase is to assume that each delegation of execution step has to be “completed” by adding the corresponding trust relationship. However, such situation is not always be the appropriate because we might need to delegate actions to people we don’t trust.

Example 18. Dr. Dave might not trust Alice because of some personal experience. However, he needs to delegate the goal of *retrieve patient medical record* because he needs to do other urgent responsibilities and only Alice is available to help.

A more frequent example at organizational level is that

Example 19. company “X” is known to be unreliable from the point of view of Henri (Head of the Hospital “H”), the hospital “H” needs to archive their prescriptions with the company “X” because they have won a county-wide procurement contracts for all hospitals on the region.

Once the model has been specified it can be used for a precise analysis of boundary of the (un)trusted computing base at organizational level:

- *Potential Unreliability* might happen when an actor has been assigned a goal but is not trusted to achieve it. This might generate a potential cascading failure for some of the key goals of the actor who delegated this responsibility.

Example 20. Since in Figure 7 the patient might perceived at risk because it does not trust the Hospital “H” to fulfill its goal *get medical service*

- *Potential Misuse* when an actor has been delegated an authorization but is not trusted not to misuse it.

Example 21. From Figure 9 the Hospital “H” believes a potential misuse on the permission on executing the prescription process given to the Nurse

Also in this case analytical techniques are available with the SI* tool vulnerabilities and threats of a SI* model formalized in Answer-Set Programming (ASP) [16,25].

These potential failures indicate some critical points in the trusted computing base. The trusting computing base might be too narrow with respect to our expectations in terms of successfully achieving the goals of the target systems and preventing misuses. At this point we have two alternatives:

- identify possible scenarios on a breach of trust for the critical actor and their critical goals (following the steps detailed in the next section and identify possible countermeasures);
- or extend our trust models to incorporate the critical actors in our trusted computing base.

This last step should not be interpreted as “we just add the trust link in the model” (albeit this is the most common errors that students do). Rather this means that we put in place alternatives techniques for trust building such as those based on legal measures as identified in [26].

Example 22. To extend the trusted computing base in Example 19, Henri includes in the service contract a clause about compensation for punitive damages due to disclosure of the prescription information by the Company “X”.

5.4 Unwanted Scenarios Identification

The next step is to identify unwanted events that might disrupt the system in term of its security & GRC relevant properties (e.g., confidentiality, integrity, availability, authenticity, privacy, etc.). In this tutorial, we emphasize on events with negative impact, and a more comprehensive classification can be found in [27].

Looking at the architecture of an enterprise information system in Figure 4 one might consider events that might risk the system into three classes:

- *Resource level* - events that occur because some disruptions at the resource level;

Example 23. The Archive service is not functioning so that it cannot archive the prescription sheet that surely disrupts the continuity of the prescription process.

- *Process level* - events that occur because the business process has been performed different from its design;

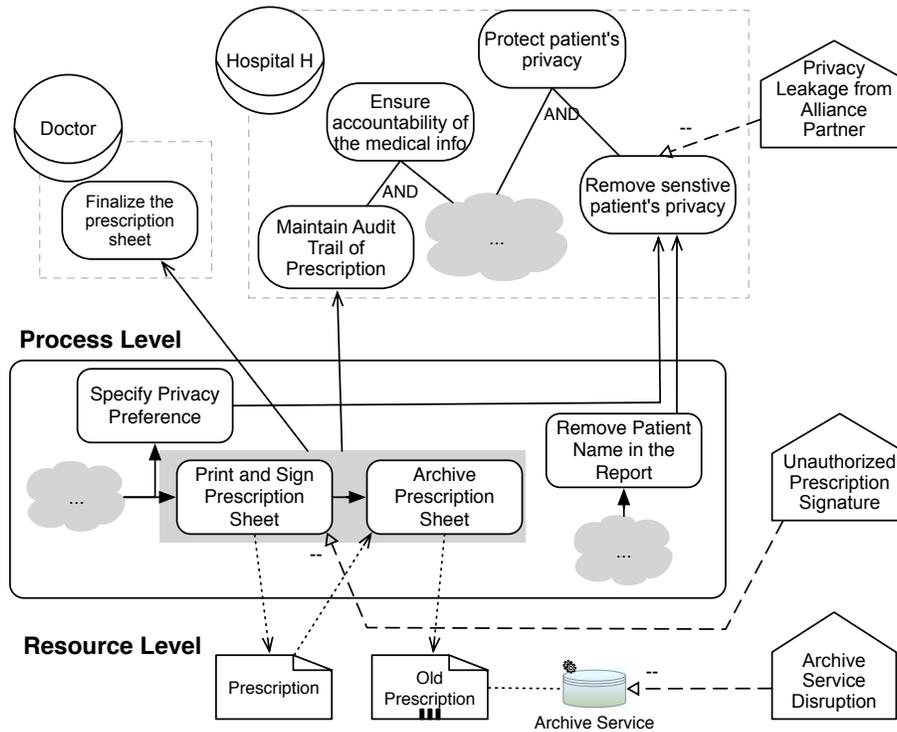
Example 24. The Dr. Dave is too busy to print & sign the prescription sheet therefore he asks Alice to do so. This scenario might compromise the goal maintain audit trail of prescription that is critical to get an accreditation from the Joint Commission.

- *Objective level* - events that prevent the state-of-affairs that the system intends to fulfill.

Example 25. A new business policy of making a Hospital Alliance with competitor hospitals require the hospital “H” to give an access to its medical info that might cause a privacy leakage.

Fig. 11. Unwanted Scenarios in the Drug Reimbursement System

Objective Level



Analysts start from eliciting events that might compromise a security & GRC property of the system. We recommend analysts to start the event identification process from the resource level then move up to the process, and finally the objective level. In this manner, we prevent the spurious identification of an event's impact. If an event disrupts a resource, then certainly it will also produce a disruption effect to the process that uses the resource, and consequently this will affect the goals that the process aims to satisfy. Conversely, in the case of the negative event at the process level (e.g., employee strike), the event cannot be mis-captured as an event originated from a resource disruption. All these are depicted in a SI* model as in Figure 11. Finally, analysts can identify what other security & GRC properties that might also be compromised.

5.5 Risk Assessment

This assessment includes risk analysis (identification and estimation) and risk evaluation [28]. Based on the unwanted scenarios identified in the previous section the analysts estimate their likelihood and severity [28]. Note that risk estimation can be either qualitative or quantitative depending on the availability of evidence for the occurrence

of unwanted scenarios. The SI*-GRC method allows analysts using any risk models to estimate the risk level of the system, such as Probabilistic Risk Analysis [29], Bayes Network [30], CORAS [31], Fault-Tree [32], GR-Framework [27]. In the paper [33] we explain how to interplay the goal methodology with other industrial methodologies. For practical reasons our SI* Tool, so far, only implements the automated reasoning from the GR framework.

Risk evaluation compares the result of risk the estimation with the risk criteria (e.g., costs, benefits, priorities, acceptable loss) defined by the GRC council (see Section 3.1). In case the risk level is unacceptable (i.e., it is beyond the risk tolerance that an actor can sustain), the analyst needs to identify appropriate treatments to mitigate the risk either by reducing their likelihood or their severity. Mitigation by control is explained in the next section.

6 Control Analysis

Once the risk level of an event is unacceptable, analysts need to formalize a statement of the desired result/state-of-affair or purpose to be achieved by implementing controls. In SI*-GRC we call such statement as *control goal*. The establishment of control goals and their corresponding implementation by control processes is what ensures the security of business objectives and processes.

6.1 Control Goal Analysis

During this step, the analysts must describe the control goals together with the actors responsible to manage them (i.e., the owner). These goals can be refined in a process that mirrors the construction of the target goal model in Section 4. For the refinement of business goals the process can be stopped as soon as we can identify a business process that can fulfill the goal. For control goals we should stop the process of refinement until we can fulfill the following qualities:

Complete - all unwanted scenarios leading to unacceptable risks are addressed by at least one control goals ;

Appropriate - there must be sufficient evidence that the achievement of a control goal will actually avoid the unwanted scenario or mitigate its effects;

Precise - control goals and actors in charge of their achievement must be clearly specified, enabling unambiguous interpretation of the level of compliance or failure of a business process with regard to the control goal.

These three qualities are complementary; that is, a set of control goals might be complete, but not appropriate. For example, it covers all relevant business needs, but wrong security assumptions might lead to an unacceptable risk level. The analysis might be appropriate (and determine the right effect in terms of impacts and likelihood of harmful events), but the description of the control is not precise enough to allow for the correct implementation or the automation of the solutions, for example because it does not specify who is in charge of achieving the control goals.

The control goals specified in Table 3 might be clear and easy to understand by the stakeholders. These control goals are still not precise enough to be machine implemented and monitored in terms of their effectiveness and performance.

Table 3. Establishing Control Goals

Business Goal: Prescribe therapy/drugs	
Compliance Requirement: • Ensure accountability of the medical info	Quality-Attribute: Q_1 Integrity Q_2 Accountability
Risk: • Compromised Prescription (Risk of Q_1) • Unauthorized modification of the medical info (Risk of Q_1) • Unidentifiable clinical case entry in the medical info (Risk of Q_1)	Control Goal: • Ensure all prescriptions are correct • Detect unauthorized modification of the medical info • Trace any un-accountable entry to the medical info

By taking advantage of repetitive patterns in control design, we can reduce the modeling effort and provide compliance experts with reusable process knowledge through a set of *control patterns*. Each pattern acts as a generalized description of actions that are frequently used in mitigating similar risks. See [34] for further details and examples of control patterns.

At the level of control goals the following patterns are the ones most frequently found in the literature:

- *avoidance* controls try to select an alternative path where the unwanted scenario does not materialize;

Example 26. To avoid such unwanted incident in Example 11, the hospital “H” can decide to delegate the goal `register patient` to another actor that has necessary capabilities (e.g., the Patient Admin Division)

- a *prevention* mechanism aims at preventing unwanted events to occur in the system (or at least reducing their likelihood);

Example 27. In Example 25, such initiative might cause a breach on patients’ privacy. However, the sharing is necessary because it brings numerous business opportunities. Therefore, every sharing of medical information to another hospital must be anonymized except the data subject is the patient at the other hospital.

- an *attenuation* mechanism tackle an unwanted events that cannot be prevented and tries to attenuate its severity.

Example 28. In Example 25, analysts might decide to obtain an insurance in case the hospital “H” get a privacy-related law-suit.

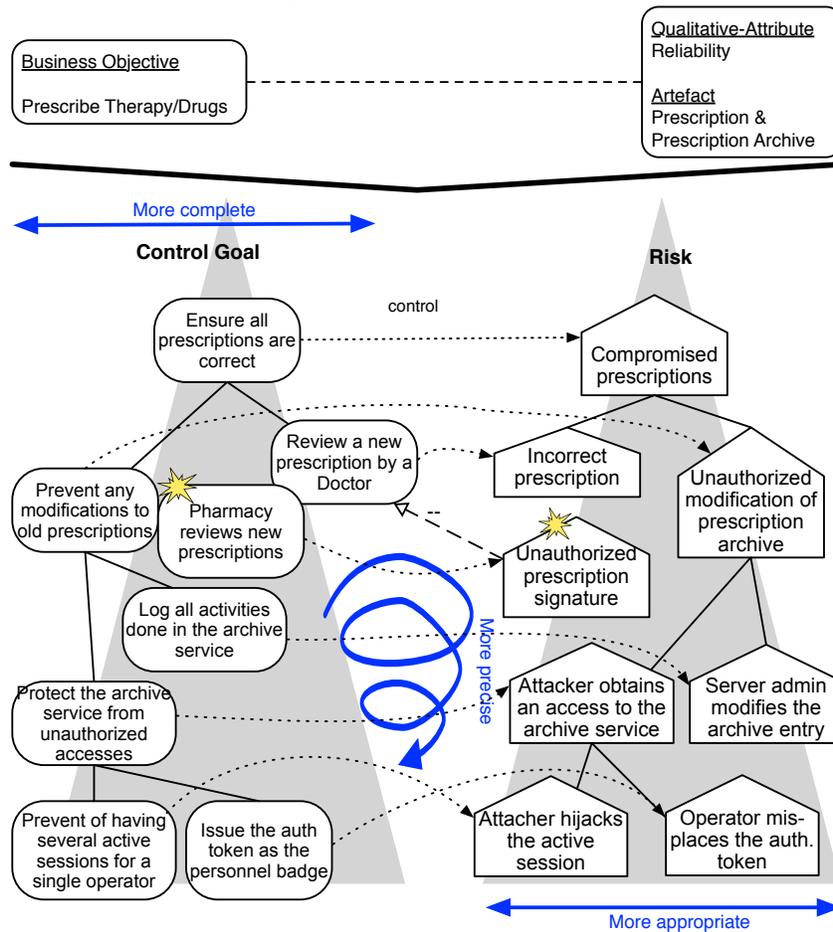
Moreover, one might follow the control goal templates (Table 4) in describing control goals in the context of the drug reimbursement system at the hospital “H”.

A control goal describing the activity above is at a high-level of abstraction. Therefore, it needs to be further refined into specific definitions. The refinement is performed iteratively, resulting in a “tree” of control goals, similar with the tree in the goal modeling (Section 4.2). Further down the tree, the “leaf” control goals become more precise and specific, and they are designed to provide specific risk mitigation function.

Table 4. Control Goal Template.

<actor>	<modality + verb>	<attribute>	<artefact>
- Hospital	- should preserve - should ensure - must prevent - need to protect	- privacy - integrity - confidentiality - continuity - reliability	- patient personal information - drug reimbursement records - the register patient process - the medical service to patients - all prescriptions

Fig. 12. Control Goal Refinement.



This refinement of control goals can be continued in parallel to a refinement of the unwanted scenarios. Each refinement and review iteration of the models leads to an increase in precision, while the broadening of controls increase completeness as shown in Figure 12; where the starred constructs are newly identified during the refinement.

Table 5. KAIs of Control Goals

Control Goal	KAI
• Issue the authorization token as the personnel badge	The number of system access done by reported missing personnel badges
• Prevent of having several active sessions for a single operator	The number of two activities done from two different locations with in less than 30 minutes
• Protect the archive service from unauthorized access	The number of suspected modification
• Log all activities done in the archive	The number of identified suspicious access
• Pharmacy reviews new prescriptions	The number of possible adverse drug reactions The drug values from excessive prescriptions
• Review a new prescription by a doctor	The number of near-miss event ⁵
• Ensure all prescriptions are correct	The number of reported medical error

Example 29. In Figure 12, we analyze the control goal ensure all prescriptions are correct so that complete, appropriate, and precise. Along the refinement of the risk model (i.e., unwanted event), analysts refine the control goal. into finer (sub) control goals. The event of unauthorized prescription signature, as in Example 24, is newly identified after the 1st iteration because it might affect one of the (sub)control goals (i.e., review a new prescription by a doctor). This situation requires analysts to enrich the control goal model by introducing a new control goal pharmacy reviews new prescriptions.

More detailed risk analysis improves the appropriateness of the control and the corresponding mitigation effects. This refinement process is “sufficient” when further refinement of risks do not give new control goals or make the existing control goals more precise. In other words the analyst stops the refinement process because there is no advantage in producing a richer or more detailed risk model.

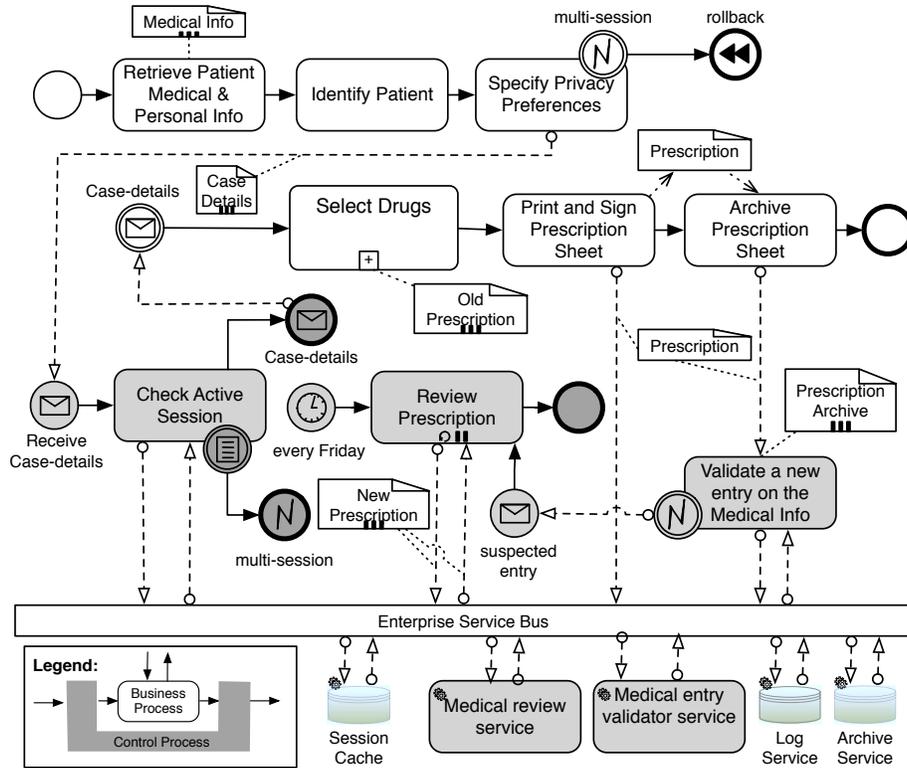
At the end of the refinement process it should be possible to identify Key Assurance Indicators (KAIs for short) which assess the achievement of these objectives. These indicators should not depend on the way in which the control goals are actually implemented by a control process but only on the final outcome in terms of desired and achieved security properties. Intuitively the KAIs play the same role of Key Goal Indicators (KGIs) in COBIT [6]. Some examples from identified control goals, in Figure 12, are illustrated in Table 5.

6.2 Control Process and Services Map

A *control process* is a realization of a detailed control goal (the leaf nodes of Figure 12) and in a SOA environment can be implemented as a wrapper to the business components (as depicted in Figure 13 ⁶) to preserve their quality attributes.

⁶ Grayed constructs are the parts that are introduced for implementing a control goal. Moreover, a construct with “mechanical-gear” indicates an underlying technical services supporting the execution of a process (i.e., business or control process)

Fig. 13. Interwoven Control Process and Business Process.



Example 30. To implement the control goal prevent of having several active sessions for a single operator, the control process intercepts the message of case-details and consequently blocks the prescription process proceeding to the select drugs activity. The control process triggers an error event of multi-session where there has already an active session (via SessionChace service) for that particular operator, otherwise will release the prescription process. The error event will lead to the current existing process to be aborted and all operations are being rollback.

To implement control processes one needs to explicitly specify which service events need to be controlled. So a human mapping is necessary, eventually aggregating conditions on services for achieving a control goal on the specific services on which it is required. The particular forms in which a control process is implemented might vary a lot depending on the architecture that is currently used. However, we can use also in this case a number of security patterns [35].

For a SOA scenario, a possible architecture is based on the Enterprise Service Bus (ESB) [36]. The ESB has the capability to detect when a message (e.g., request, response, notification) arrives and to perform some actions (e.g., block, delete, delay, re-

Table 6. ESB-related policies.

<ul style="list-style-type: none">- Block every result from the underlying service of Archive Prescription Sheet before storing the prescription at ArchiveService, and forward to MedicalEntryValidatorServices to be validated. MedicalEntryValidatorServices emits a release event to the ESB when the prescription is not suspected entry- Release the results of ArchivePrescriptionSheetService after receiving the release event from MedicalEntryValidatorServices; and store the prescription at the ArchiveService.- Trigger the log event each when there is any request to ArchiveService and forward the event log to be saved at LogService- Trigger a suspected entry event when the validate a new entry on the medical info activity, using MedicalEntryValidatorService, suspects the prescription- Trigger a start event, every Friday, for Pharmacy to Review Prescription all prescriptions indicated as suspected entries. This activity is also checked a sample of new prescriptions.

lease modify, forward, trigger). The basic principles for interweaving control and business services are the following:

- If a control service is executed before the business service is invoked – (i.e., filter in/out), the ESB will block the request message to the business service and forward the request to the control service. The control service will notify the messaging service whether to remove the blocked request if it is considered to be an inappropriate request, or to release it; as in Example 30.
- If a control service is executed after the business service invoked (i.e., verify), the ESB will block the result of the business service invocation before dispatching it to the subsequent service in the business process, and release the results after performing some operations (e.g., modify/add/remove some data items, attach signature) or even remove the result if it violates some policy (e.g., not sending confidential data).

Table 6 shows some examples of security control policies for the ESB scenario in implementing the control goal of log all activities done in the archive and the control goal of pharmacy reviews new prescriptions

At the end, each control process must be defined its Key Security Indicators (KSIs for short) which assess the level of correctness and coverage of such controls. By KSI-correctness, these indicators must indicate whether the result of the control is correct according to its design specification; while KSI coverage indicates the level coverage of the control in protecting the business process execution. Note KSI is closely dependent on the technical details how the control is implemented in the system. Intuitively the KSIs play the same role of Key Performance Indicators (KPIs) in COBIT [6]. Some examples from identified control presses, in Figure 13, are illustrated in Table 7. Note that often it is hard to define KSIs of the correctness of (semi-)manual control, because the correctness criteria are in the human knowledge and hardly encode-able for the knowledge of technical system.

Table 7. KSIs of Control Process

Control Process Name	KSI-correctness/coverage
<ul style="list-style-type: none"> Abort the prescription process when there is already an active session 	<p>Cor: The ration between the occurrences of multi-session event and the rollback events</p> <p>Cov: Percentage of the number of check active session execution over the specify privacy preference execution</p>
<ul style="list-style-type: none"> Log all requests to ArchiveService 	<p>Cor&Cov: The difference between the number log entries of archiving a prescription and the number of prescription sheet</p>
<ul style="list-style-type: none"> Friday review done by Pharmacy 	<p>Cov: The ratio between the number of reviews done by Pharmacy on the suspected prescription entry and the occurrences of suspected entry event</p> <p>Cov: The ratio between the number of prescription reviews done by Pharmacy and the number of prescription sheet</p>
<ul style="list-style-type: none"> Validate new prescriptions by MedicalEntryValidatorService 	<p>Cor: The number of prescription which triggers suspected entry event but at the end it is benign</p> <p>Cor: The number of archived prescription in ArchiveService that is suspected</p> <p>Cov: The ratio between the number of execution of MedicalEntryValidatorService in supporting the Validate a new entry in the medical info and the number of prescription sheet</p>

Besides implementing control processes, designers need to define the events (e.g., a service start/ finish/suspend or messages exchanged among services) that will be used to compute the KAIs and KSIs. To process these events, the business activity monitoring (BAM) [37] can be used, since it allows one to analyze real-time events from the business transaction and, furthermore, to compute KAIs/KSIs following the mathematical formula defined by the designers. BAM-related policies for computing KAI/KSIs related to the control goal of prevent of havign several active session for a single operator are illustrated in Table 8.

7 SI*-GRC In-Vivo

Once these controls are in place, we must still assess and monitor them at runtime for ensuring the level of security & compliance of the system; and improving the implemented controls.

7.1 The Check Phase: Monitor and Review Indicators

For each control goal and process, analysts need to identify indicators that measure its correctness and effectiveness. For these purposes, we use key assurance indicators

Table 8. KAIs/KAIs Policies in BAM

<ul style="list-style-type: none"> <p><i>Control Goal</i> Prevent of having several active sessions for a single operator</p> <p><i>KAI</i> The number of two activities done by the same operator from two different locations within less than 30 minutes</p> <p><i>BAM stores the last location and access/activity's timestamp from each operator taken. This information is obtained from the message request to SessionCache. It increments the value of KAI when the most recent activity is within 30 minutes from the previous one and done from different location</i></p>
<ul style="list-style-type: none"> <p><i>Control Process</i> Abort the prescription process when there is already an active session</p> <p><i>KSI_{cor}</i> The ration between the occurrences of multi-session event and the rollback events</p> <p><i>$N_{multi-session}/N_{rollback}$; where multi-session denotes that Check Active Session founds an active session in SessionCache for the same operator, and rollback is triggered when the prescription process is aborted and some rollback procedures is executed</i></p> <p><i>KSI_{cor} Cov:</i> Percentage of the number of check active session execution over the specify privacy preference execution</p> <p><i>$N_{check-session}/N_{specify-privacy} \times 100$; where check-session is an event triggered when Check Active Session is executed, and specify-privacy is triggered when the operator has specified the patient's privacy preference</i></p>

(KAIs) and key security indicators (KSIs) as illustrated in Table 5 and Table 7 respectively.

Typically, KAIs are the focus of the business analysts, because business analysts are more concerned with the level of compliance rather than how the control is implemented. KSIs, on the other hand, are of interest to risk/security analysts as they measure how well controls are implemented. Both KAIs and KSIs are critical for monitoring, evaluating and improving the GRC implementation. The indicators are computed independently to distinguish between cases in which the KAI of a control goal is “low” but the KSI’s associated control processes are “high”. In the former case, analysts might conclude that there are some risks that have not been mitigated. In the latter, it might be that the compliance of a business process is achieved through external factors (from luck to organizational procedures), rather than deployed controls.

The technological infrastructure implementing the controls and indicators identified by the method should then support Continuous Controls Monitoring techniques. For example, the tools developed in the MASTER project ⁷ makes this possible: monitoring and assessment infrastructures provide one main source of indication and feedback for the review. The monitoring infrastructure gathers and correlates events emitted from monitored services based on monitoring policies defined by control processes. The assessment infrastructure evaluates the relevant KAI/KSI values based on the events triggered by the monitoring infrastructure. The events may be stored in an audit trail or event log database for further processing and assessment as necessary. The KAI/KSI

⁷ <http://www.master-fp7.eu>

values are then presented to a (human) control process supervisor through a dashboard or a reporting tool.

7.2 Act Phase: React and Improve

In a SOA-based business environment, orchestration of services associated to business processes could change regularly. Each change may violate existing control goals or impact the effectiveness of the relevant controls. It is, therefore, crucial to have a continuous review of the compliance level of existing business processes and the effectiveness of implemented controls.

Following the result of the review different types of actions can be performed: *augmentative*, *corrective* and *preventive* actions, can be taken based on the review findings discussed earlier to ensure that actual control processes always reflect the latest business compliance requirements.

The methodological steps are detailed below:

1. *Identification of changes*. Generally, augmentative actions are required when there is an update to existing: (i) legal requirements; (ii) business requirements; or (iii) control activities as the result of a review performed on the associated control processes. Based on these updates, we first identify what potential changes to existing control objectives/activities that need to be made.
2. *Planning and implementation of changes*. This step repeats the tasks involved in the PLAN and DO phases so that new controls can be put in place or modification to existing controls can be implemented.
3. *Documentation and review*. All information gathered from performing the above steps should be documented for future references.

8 Final Remarks

What we have seen in this tutorial is a summary overview of SI*-GRC method to analyze and design security controls. The gist of the method is to capture the organizational setting of the system making sure that business goals and processes are the main citizen.

The basic building blocks of the methods are

1. a *modeling framework* based on a requirement engineering framework, with some extensions related to security & GRC concerns, such as: trust, permission, risk, and treatment,
2. a *analysis process* defining systematical steps in analyzing and design security controls,
3. a number of *analytical techniques* to verify that certain security properties are satisfied and the risk level is acceptable, and at last
4. a *CASE tool*, namely the SI* tool to support analysts in using the method.

Many details of the research methods that constitute the building blocks of this methods can be found in our research papers and in the research s of the our co-authors that we list in the acknowledgement section.

Acknowledgments

We would like to thank Paolo Giorgini, John Mylopoulos, Nicola Zannone from Trento for many useful discussion on the the large and unruly family of security requirements engineering. Vohla Bryl, Fabiano Dalpiaz, Federica Paci, Ayda Saidane, Angelo Susi, Minh Sang Tran and many others contributed to the discussion on what really is a good security requirements methodology. Daniela Marino and Andrea Micheletti from Hospital San Raffaele, Valentino Meduri and Alessandra Tedeschi from DeepBlue and Carlo Ricucci and Domenico Presenza from Engineering proved to be an essential soundboard for the applicability of our ideas. Claire Woolridge and Bernadette Louat were instrumental to link our methodology to COBIT and other ISACA methodologies. Any mistake is of course ours.

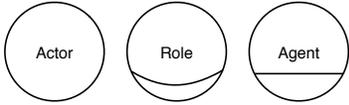
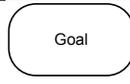
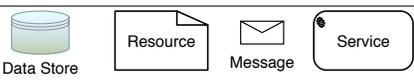
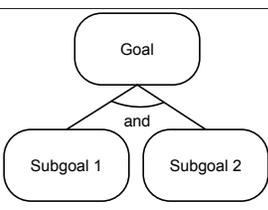
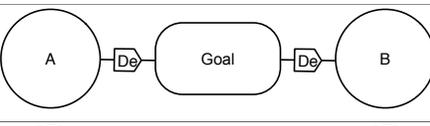
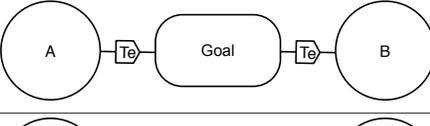
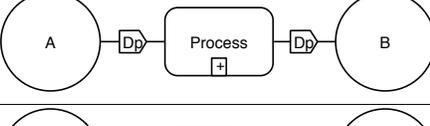
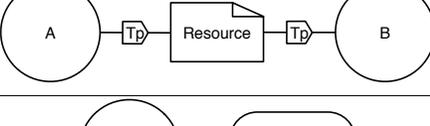
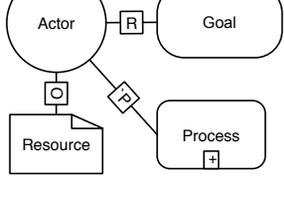
This work has been partially funded by EU-FP7-ICT-IP-ANIKETOS, EU-FP7-ICT-IP-SecureChange, and EU-FP7-ICT-NoE-NESSoS project.

References

1. Rasmussen, M., Kark, K., Penn, J., McClean, C., Bernhardt, S.: Trends 2007: Governance, risk and compliance: Organizations are motivated to formalize a federated GRC process. Technical report, Forrester Research (April 2007)
2. McClean, C., Whiteley, R., Kark, K., Dill, A.: The Forrester Wave: Enterprise governance, risk, and compliance platforms, Q3 2009. Technical report, Forrester Research (July 2009)
3. ACL: Audit command language. <http://www.acl.com/> - last check 15.07.2010 (3020)
4. Pfleeger, C.P., Pfleeger, S.L.: Security in Computing, 4th edn. Prentice-Hall (2006)
5. ISACA: The Risk IT Framework. ISACA (November 2009)
6. Institute, T.I.G.: CoBIT - Framework Control Objectives Management Guidelines Maturity Models. 4.1 edn. ISACA.org (2007)
7. Trist, E.: The evolution of Socio-Technical systems. Occasional Paper 2 (1981)
8. Asnar, Y., Lim, H.W., Massacci, F., Worledge, C.: Realizing trustworthy business services through a new GRC approach. ISACA Journal - JOnline 2 (2010)
9. Deming, W.E.: Out of the Crisis. MIT Press (2000)
10. Marino, D., Potral, J.J., Hall, M., Rodriguez, C.B., Rodriguez, P.S., Sobota, J., Jiri, M., Asnar, Y.: Master scenarios. Project Deliverable D1.2.1, MASTER Consortium (2009) This case study has been provided by Hospital San Raffaele Foundation. Its complete description is available at http://www.masterfp7.eu/index.php?option=com_docman&task=doc_details&gid=53&Itemid=60.
11. Erl, T.: SOA Principles of Service Design. Prentice Hall (2007)
12. Casteleyn, S., Daniel, F., Dolog, P., Matera, M.: Engineering Web Applications. Springer-Verlag New York Inc (2009)
13. Yu, E.: Modelling Strategic Relationships for Process Engineering. PhD thesis, University of Toronto, Department of Computer Science (1995)
14. OMG: Business process modeling notation (January 2009)
15. Hofstede, A.H.M., Aalst, W.M.P., Adams, M., Russell, N., eds.: Modern Business Process Automation-YAWL and its Support Environment. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
16. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Requirements engineering for trust management: model, methodology, and reasoning. International Journal of Information Security 5(4) (October 2006) 257–274

17. Robbins, S.P.: *Organizational Behavior, Concepts, Controversies, Applications - Seventh Edition*. 7th edn. Prentice-Hall (1996)
18. OCC: *Management information systems. the comptroller's handbook*, Office of the Comptroller of the Currency (May 1995)
19. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* **26**(3) (1987) 276–292
20. van der Aalst, W.M.P.: Formalization and verification of event-driven process chains. *Information and Software Technology* **41**(10) (July 1999) 639–650
21. Eriksson, H.E., Penker, M.: *Business modeling with UML: Business Patterns at Work*. John Wiley & Sons (2000)
22. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development* (March 2009)
23. van der Aalst, W.M., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. *Data & Knowledge Engineering* **53**(2) (May 2005) 129–162
24. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An Agent-Oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* **8**(3) (May 2004) 203–236
25. Massacci, F., Mylopoulos, J., Zannone, N.: Computer-aided support for secure tropos. *Automated Software Engineering* **14**(3) (August 2007) 341–364
26. Compagna, L., Houry, P.E., Krausová, A., Massacci, F., Zannone, N.: How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. *Artificial Intelligence and Law* **17**(1) (November 2008) 1–30
27. Asnar, Y., Giorgini, P., Mylopoulos, J.: Goal-driven risk assessment in requirements engineering. *Requirements Engineering* **16**(2) (2011) 101–116 10.1007/s00766-010-0112-x.
28. ISO/IEC: *Risk Management-Vocabulary-Guidelines for Use in Standards*. (2002) Published: ISO/IEC Guide 73.
29. Vose, D.: *Risk Analysis: A Quantitative Guide*. Wiley (2000)
30. Mosleh, A., Hilton, E.R., Browne, P.S.: Bayesian probabilistic risk analysis. *SIGMETRICS Perform. Eval. Rev.* **13**(1) (1985) 5–12
31. Lund, M.S., Solhaug, B., Stølen, K.: *Model-Driven Risk Analysis - The CORAS Approach*. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
32. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: *Fault Tree Handbook*. U.S Nuclear Regulatory Commission (1981)
33. Delande, O., Felix, E., Massacci, F., Paci, F.: Managing changes with legacy security engineering processes. In Zeng, D., Yang, C.C., Collberg, C., eds.: *Proc. of IEEE Internat. Conf. on Intelligence and Security Informatics (ISI 2011)*, IEEE Press (2011)
34. Namiri, K.: *Model-Driven Management of Internal Controls for Business Process Compliance*. PhD thesis, Universität Fridericiana zu Karlsruhe (2008)
35. Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., Sommerlad, P.: *Security Patterns: Integrating Security and Systems Engineering*. 1 edn. Wiley (March 2006)
36. Chappell, D.: *Enterprise Service Bus*. O'Reilly Media, USA (2004)
37. Kochar, H.: *Business Activity Monitoring and Business Intelligence*. <http://www.ebizq.net/topics/bam/features/6596.html> - last access at 2010.07.03 (December 2005)

Appendix A - SI* Syntax

SI* Diagram	DLV Predicate
	actor(Actor) role(Role) agent(Agent)
	goal(Goal)
	process(Process) task(Task)
	resource (Resource)
	event(UnwatedEvent)
	and_decomposition(Goal, SubGoal1, SubGoal2)
	del_exec(A,B,Goal)
	trust_exec(A,B,Goal)
	del_perm(A,B,Process)
	trust_perm(A,B,Resource)
	request(Actor,Goal) provide(Actor,Process) owner(A,Resource)

Note: See BPMN [14] for complete constructs for the Process Modeling.

hi from DeepBlue and Carlo Ricucci and Domenico Presenza from Engineering proved to be an essential soundboard for how our ideas could work in practice.