Model Driven Implementation of Security Management Process

Bruno A. Mozzaquatro[1], Ricardo Jardim-Goncalves[1] and Carlos Agostinho[2]
[1]*DEE/FCT, Universidade Nova de Lisboa (UNL), 2829-516 Caparica, Portugal*
[2]*Centre of Technology and Systems, UNINOVA, 2829-516 Caparica, Portugal*

Keywords: Business Process, MDA, Security Management, Model Transformations, MDSEA.

Abstract: Services composition involves many time and effort to describe high-level requirements of the business process. To this purpose, the Model Driven Service Engineering Architecture (MDSEA) is a methodology to distinguish the business view and technical point of view in products and services and to agilize the software development. Such capabilities demand more effective process applied to specify, evaluate, communicate and design the system as well as system functionalities and security issues. Security aspects are critical when it involves privacy of data exchange of devices. In this context, the definition of security artifacts during the design of a business process consumes time of system funcionalities development. This paper proposes an implementation of a security management process using the methodology MDSEA as support to promote model transformations from business model to specific artifacts and configurations. This support enables to enrich a solid business model with technical details by specialists.

1 INTRODUCTION

Collaboration among enterprises has become a better solution towards market opportunities. Its products and related services result in cooperative networks to support service life cycle. Services composition requires a lot of time and effort to describe and design more accurately business details (Bazoun et al., 2013). However, at high-level requirements of business process involves stakeholders to specify, evaluate, communicate and design the system supporting the service and its lifecycle around enterprise processes.

Process modeling notations is an appropriately implementation to separate business details from technical details of a service system. Usually, business details are focused to domain experts to identify requirements around services and technical details consist of specific information about implementation of this service system (Münch et al., 2012). Model Driven Service Engineering Architecture (MDSEA) is a methodology responsible for supportting service model transformations from high-level concepts to specific artifacts in the process. This methodology promotes models reuse at different abstraction levels defining some ways to transform concepts in implementation (Bazoun et al., 2016).

According to a service system some concerns about secure issues are needed to take in consideration and to make the proper functioning of the system. Usually, these details are despised during a business process model elaboration because it requires security professionals within system development teams (Lambert et al., 2006). Model driven security and generation of security software artifacts has been a topic of research in recent years (Derdour et al., 2015) (Brucker et al., 2012) (Jürjens, 2002) (Lodderstedt et al., 2002).

Model driven security has focused on improving security quality with separation of functional issues from non-functional aspects of the whole system. Usually, well-known languages are used to achieve security process integration as well as Architecture Description Languages (ADLs) and Unified Modeling Language (UML) (Ren and Taylor, 2005). Security processes involve security analysis and verification of security properties (e.g. availability, confidentiality, integrity) while system requirements at the design time and runtime (Mozzaquatro et al., 2016). This paper demonstrates how model transformation between different abstraction levels of process models address the gap between business and technical security configurations. It is based on a modeling language to facilitate the development of business processes using a services composition to enterprises.

Mozzaquatro B., Jardim-GonÃğalves R. and Agostinho C.
Model Driven Implementation of Security Management Process.
DOI: 10.5220/0006329602290238
In *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017)*, pages 229-238
ISBN: 978-989-758-210-3

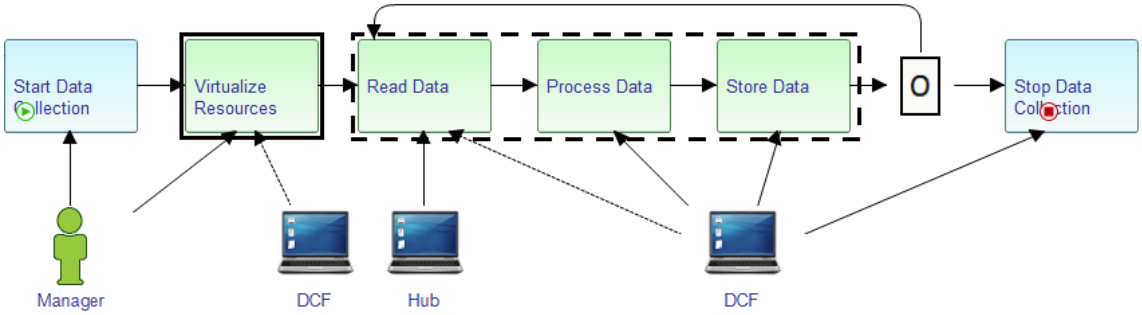

Figure 1: Baseline scenario model.

1.1 Baseline Scenario

This section describes a baseline scenario of a manufacturing scenario of the H2020 European Project: Cloud Collaborative Manufacturing Networks (C2NET) project, which reflects the data collection from sensors networks. Figure 1 presents a business process of baseline scenario addressed in this paper. It is a high-level abstraction model defined by a business managers team without technological details.

The process' workflow starts with an activity responsible to collect device's information to the data collection scenario. Start Data Collection activity interacts with the Pilot Manager to provide a set of devices that it will compose the sensor network. It is forwarded to the activity called Virtualize Resources to register these devices in the platform as well as Data Collection Framework (DCF). Each real device will have a virtual copy in the platform. Next step, the IT resource (Hub) provides support to data collection through the Read Data activity. It collects data from sensors devices and send to the DCF. The Process Data activity, localized in DCF, filters data collected and process according to the needs of the project to be stored (Data Store activity).

Security activities are not highlighted in this high-level perspective of the process, but it is intrisic and should be considered at lower levels following a security process by services, archifacts and implementations. For example, activities in the continuous line require security concerns about authentication and confidentiality. Dotted line's activities need the security requirement of integrity and confidentiality, which defines ways to protect data collection of sensors network over the public network. In the whole workflow, availability is another requirement essential to ensure reliability and to maintain the right functioning of the system.

1.2 Paper Outline

The rest of paper is organized as follow: Section 2 presents the background research of this paper following by related works (Section 2.1), Model Driven Service Engineering Architecture (Section 2.2), and a modeling tool called SLMToolBox in Section 2.3. Section 3 describe a detailed security process through a model tranformation from a business model. Section 4 describes a methodology of this paper and his contribution of models transformations based on MDSEA. Some tests are presented in Section 5. Finally, conclusions and future works is presented in Section 6.

2 BACKGROUND RESEARCH

This section describes main subjects involved in this paper to understand the security process based on the transformation different abstraction levels of models.

2.1 Literature Review

This section describes some related works regarding model transformations and model driven representation of security process. These two categories are outlined in this section.

2.1.1 Model Transformations

The authors (De Castro et al., 2011) propose a model driven approach that apply CIM-to-PIM model transformation from high-level business models to lower-level information system models. The key advantage is to use real high-level business models into the process using an MDA-based approach that offers models of different abstraction levels and model transformations. The results demonstrate how a model driven approach helps the alignment process between different views when adoption service oriented approach for software development.

The author (Chen, 2015) presents a methodological approach for the generation of manufacturing service involving steps of a service life cycle. It considers that there is no complete service engineering methodological approach in the market. However, the methodology proposed in this paper considers a bag of assets with i) service modeling based on MDSEA, adapted and extended from MDA/MDI approaches; ii) service engineering process to ensure the customer's needs within an entire life cycle (Bahill and Gissing, 1998); iii) service governance framework to control enterprises within a manufacturing service ecosystem and your respectives interactions with data and information; iv) service live life cycle framework to capture, categorize, and structure a set concepts and issues relating to manufacturing servitization; and v) SLMToolBox consists several graphical editors to model manufacturing services and service systems from business and functional models.

The fourteen steps defined in the methodology is considered as reference guidelines to select and customize according to needs. Though some limitations are highlighted by authors as well as the time to learn several models and tools contained in the methodology, not all models are supported by tools, and the application of this methodological approach would be more suitable to large companies even though it has been experimented with success in SME.

2.1.2 Model Driven Security Process

The adoption of model driven security in business processes is an emerging research area. Some works explore the model driven approach for process-oriented systems focusing on access control such as SecureUML (Basin et al., 2003), UMLSec extension for UML (Jürjens, 2002).

The authors (Menzel et al., 2009) propose an approach to describe security requirements at the business process layer and their translation to concrete security configuration for service-based systems. Such security requirements involve some intentions such as confidentiality or integrity on an abstract level. Information at the modeling layer is transformed to a domain-independent security model following the generation of security configurations based on the modeled requirements with a pattern and its relationship to the security model. This pattern is composed of problem, context, forces and solution. Each pattern refers to a particular security goal and identifies suitable security protocol/mechanism.

Wolter (Wolter et al., 2009) describes a generic security model that specifies security goals, policies, and constraints based on a set of basic entities, such as objects, attributes, interactions, and effects. This model has an abstraction from technical details and hence they provide a mapping from this abstract model to platform specific target languages (e.g. XACML or AXIS2) with authorisation and confidentiality constraints defined.

2.2 Model Driven Service Engineering Architecture

The Model Driven Service Engineering Architecture (MDSEA) is an approach to distinguish the business and technical point of view in product and service systems. This architecture follows Model Driven Architecture (MDA) paradigm to provide guidelines for structuring the specifications of engineering activities. On the basis of MDA paradigm, MDSEA proposes a framework for service system modeling using three levels: BSM, TIM and TSM (Bazoun et al., 2013).

- Business Service Model (BSM): this level specifies models at high-level and reflects the business perspective of the service system, without considerations between technologies that will be used. It is the link between domain experts and development experts.
- Technology Independent Model (TIM): this level delivers detailed specifications of the structure of the service system, which focuses on the functional and operational details used for implementation.
- Technology Specific Model (TSM): this level presents procedures to the implementation of the system to use a particular type of technology, including middleware, operating systems, and programming languages. According to these specifications, the next step consists of the implementation of the designed service system.

Model-Driven Engineering (MDE) is a practice for developing model driven applications through the use of models, allowing concepts closer to the domain of problems. It involves a software engineering approach to address system complexity related with Model Drive Development (MDD) to describe and build software systems (Atkinson and Kuhne, 2003).

The main aspect to avoid some problems of the difficult task of developing effective and efficient software is automating developments in an information system life cycle. It suggests the description of a system in an abstract way performing transformations in several steps into real, executable systems (e.g. source code) (Selic, 2003).

The utilization of models has several meanings, among which can be cited: i) set of extracts of a system under study (Seidewitz, 2003), ii) simplification

of reality (Selic, 2003), and set of formal elements which describe something that is being developed for a specific purpose and can be analyzed by methods (Mellor et al., 2003). Historically, models have been used to ensure the efficacy of systems before the superior effort about all system directly. It could be said that MDE vision develops a principle that "everything is a model" (i.e., platforms, components, legacy software, services, etc.) (Agostinho et al., 2012) (Bézivin, 2006). These models represent the results a fast development of products and systems based on the communication among product managers, designers, and member of the development team.

Usually, MDE consists of the model transformations between layers as depicted in Figure 2. The following sections describe the modeling environment, called SMLToolbox, to define a high-level process and perform transformations according to the MDSEA.

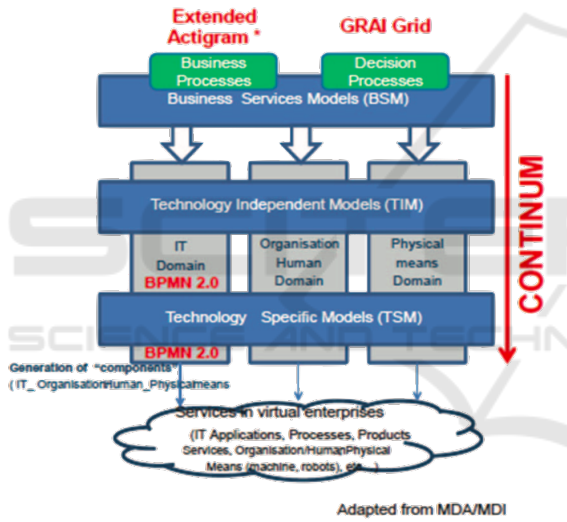

Figure 2: Overview of MDSEA (Ducq et al., 2014).

2.3 SLMToolbox Modeling Environment

This section describes an integrated modeling tool dedicated to services lifecycle management with activities as well as syntactic validation, model transformation, execution process, modeling process, engineering process, and workflow monitoring and control.

Service Life Management Tool Box (SLMToolBox) is a software developed in the frame of the EU FP7 MSEE project[1]. SLMToolBox consists in some facilities to develop a new service or improve an existing one, within a single enterprise or a virtual enterprise (Bazoun et al., 2014). According to Bazoun (Bazoun et al., 2016), it is an implementation of the BSM and TIM levels of MDSEA.

SLMToolBox composes several scientific concepts about service innovation into one tool such as service modeling, engineering, simulation, monitoring, and control (Bazoun et al., 2016). This tool takes benefits of a model based architecture (validation, transformation and execution), maintaining the coherence of the transformation from business requirements to IT implementation (modeling), simulating the result of the service (engineering), and designing the governance of the service (monitoring and control) as depicted in Figure 3.

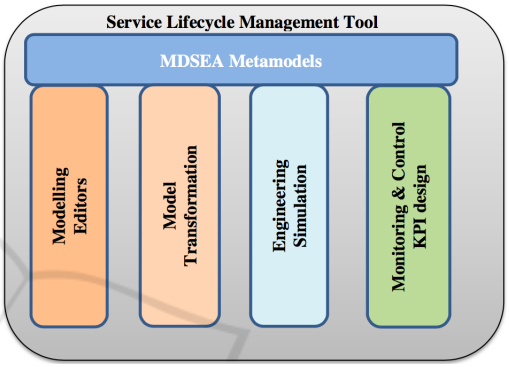

Figure 3: Service Life Management Tool Box (Bazoun et al., 2016).

These four pillars support several goals to enable elaboration of service descriptions, modeling activities based on a methodological support, simulation of business processes providing animation and reports, and implementation of service system's governance.

The modeling architecture adopted by the SLMToolBox composes three modeling levels of MDSEA: BSM, TIM, and TSM. Each level has a set of graphical modeling languages according to the level's viewpoint. It enables non-expertise humans to organize elements around domains of expertise by separating and decomposing the concerns. For example, BSM and TIM use graphical modeling languages to represent in more details certain aspects of the service model, namely as BSM Templates, Extended Actigram Star (Bazoun et al., 2016), GRAI Grid, and UML to BSM level; and TIM Templates, BPMN 2.0, UML, and DEVS (Atomic and Coupled Models) to TIM level.

At BSM level, Extended Actigram Star Editor models offer the elements of connectors which represent cooperation between entities within the same organization or different organizations. On the other hand, at TIM level, BPMN 2.0 provides an integrated editor with Eclipse platform to provide an intuitive modeling tool for the business rules, graphical edition with sup-

[1] http://www.msee-ip.eu/project-overview

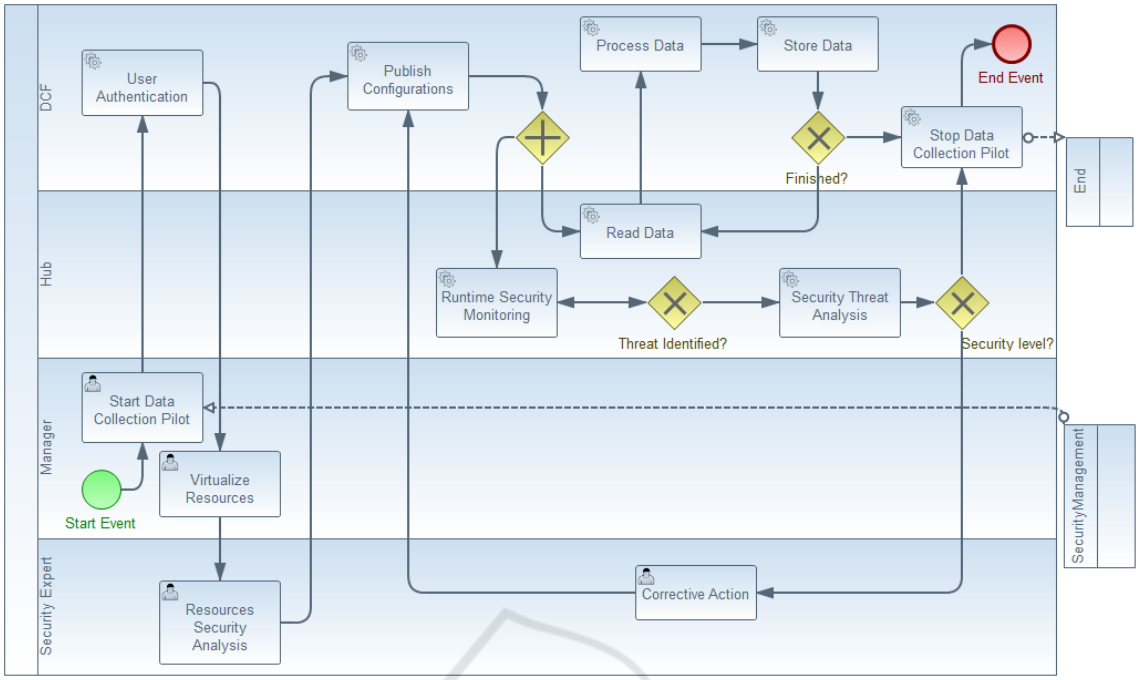

Figure 4: BPMN of the security process representated in SLMToolBox.

port for the BPMN domain.

SLMToolBox enables to create new diagrams using the combination of ATL (Atlas Transformation Language) and XSLT (eXtensible Stylesheet Language Transformations) in two ways: create a new BPMN diagram by the standard way, or to create a new diagram from an existing EA*. The transformation of EA* model to BMPN model is performed by ATL. XSLT is used to generate the graphical diagram view of the model, resulting in a BPMN diagram.

More specific information can be obtained in references (Bazoun et al., 2013) (Bazoun et al., 2014) (Bazoun et al., 2016).

3 SECURITY PROCESS

In this section we describe a security process composed by activities required in the business model of a scenario of H2020 European Project: Cloud Collaborative Manufacturing Networks (C2NET) project. This project designed a cloud-based system to data collection in manufacturing enterprises. The business model was presented to be followed in this paper in Section 1.1.

This security process has an important role to protect security properties namely: availability, confidentiality, integrity. The objective of this process is to ensure these security requirements providing activities in the workflow to protect data exchange of sensors devices.

Basically, the model presented in Figure 1 demonstrates only information about data collection activities using C2NET platform. Some security details and activities needs to be added to protect the system, which involves external users and data exchange over public network.

Following the MDSEA methodology, an advantage of the model transformation between BSM to TIM is the addition of technological details about the process. Figure 4 presents the BPMN diagram, which is the product of this model transformation using the modeling tool SLMToolBox. Also, this diagram contains more details of the data collection as well as Publish Configurations, which involves the registration of configurations of resources in the C2NET platform. In addition, some activities were added to ensure security properties such as Availability, Confidentiality, and Integrity. Availability consists of tasks to maintain the system operable and accessible to authorized users. Confidentiality means that information should be available only those person authorized. Main security mechanisms to protect unauthorized access are cryptography and access controls. Integrity means ensure the trustworthiness, origin, correctness of information. It is based on integrity of information but also to the origin integrity.

A brief overview of security services and activities of the EU FP7 C2NET project is described in Table 1 and 2. These activities follow the workflow to regis-

Table 1: Data collection activities of the workflow.

Activity	Description
Virtualize Resources	Sends information about resources to the Hub.
Publish Configurations	Registers resource's configurations in the cloud platform (DCF).
Read Data	Collects data about resources.
Process Data	Analyzes filtered data about resources.
Store Data	Stores all information of resources in database.

Table 2: Security activities of the workflow.

Activity	Description
User Authentication	Identifies the pilot manager in the data collection platform. e.g. Single Sign On.
Resources Security Analysis	Verifies resource's characteristics and its security risks to adopt suitable security mechanisms. e.g. Security professional analysis
Runtime Security Monitoring	Uses security tools to monitor resources and services. e.g. Intrusion Detection Systems.
Security Threat Analysis	Analyzes security alerts generated by the monitoring tools. e.g. Correlation between threats and security solutions.
Corrective Action	Reacts a threat and change security policies to restablish right functioning of the sensors network. e.g. Security professional

ter resources (e.g. sensors devices) and send theirs configurations between DCF and Hub of the C2NET platform. The model transformation of business model into the BPMN diagram is depicted in Figure 4. BPMN model contains more details about data collection and security activities, which are described in Section 3.1 and 3.2.

3.1 Design Time Approach

Design time is an approach to establish some configurations before the start of the system, at least pre-configured definitions. Activities of Data Collection (Virtualize Resources and Publish Configurations) and Security (User Authentication, Resource Security Analysis) are pre-configured tasks essential to manage resources at the run time approach.

Virtualize Resources activity consists of receive device's information to be registered in the cloud-based platform. Publish Configurations activity is responsible to the process of the dissemination of device's information.

User authentication is an activity responsible to register and identify users of the manufacturing pilot. Once registered, authorized users are identified and allowed to send resources information to the platform. Also, it prevents unauthorized users from making improper or unauthorized modifications to resources, which maintain the consistency of the system. This activity uses authentication services with different protocols to autenticate users such as Single-Sign On (SSO), username/password verification or hibrid (both).

Resource Security Analysis is an activity responsible to verify potential solutions to detected threats in the system. It uses a knowledge database to analyse this information (Mozzaquatro et al., 2015).

3.2 Run Time Approach

Run time approach is responsible to perform some tasks during the execution of the system, maintaining correct functioning of the system, but also to recover of an anomaly behavior. It involves tasks of resources analysis and management within data collection scope and security issues such as Data Collection (Read Data, Process Data and Store Data) and Security activities (Runtime Security Monitoring, Security Threat Analysis and Corrective Action).

In the security context, there is an activity to configure security tools responsible to the resources monitoring to identify potential threats such as Runtime Security Monitoring. Several security tools are suitable to monitor in realtime resource's behavior such as firewall (Hossain and Raghunathan, 2010), intrusion detection systems (Patel et al., 2016) (Butun et al.,), proxies, etc. Usually, these tools generate security alerts of an anomaly based on detection rules. Each tool send security alerts to an activity responsible to identify the reason of the detection.

This activity is Security Threat Analysis, which consists in the identification of reasons of each anomaly detection considering previous information (e.g. knowledge base). Depending on threat's security level, an activity of the workflow is responsible to realize revise and recover configurations of resources (Corrective Action). So, if there exists solutions for detected threats, it will be corrected without stop the system. In contrast, the system will be stopped and a security expert will perform manual configurations to block this threat.

4 MODEL TRANSFORMATION BASED ON INFORMATION SECURITY PROCESS

This section describes a methodology used to support model transformation from a business model (high-level) to security specifications (low-level). Figure 5 presents the methodology and its models transformations until achieve the detailed process with execution of security services.

Initially, the business model is represented in Extended Actigram Star (EA*) in high level of abstraction. The diagram depicted in Figure 1 is modeled within the modeling tool SLMToolBox.

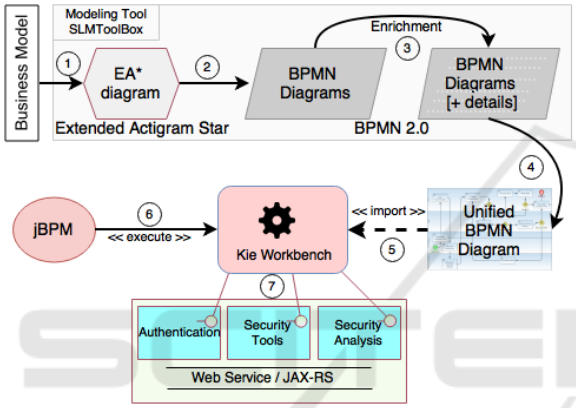

Figure 5: Adopted methodology to models transformations and execution of security services.

This part is important to specify the model at the global level with few details of services running between differents business perspectives and independent from technologies. The first model transformation in Figure 5, depicted as (1), consists in the collection of information and requirements at BSM level to the TIM level. This translation aims to save effort and reduce errors by automating models development when possible.

The model transformation (2) between EA* and BPMN diagrams enables to add more details about the process' workflow. It follows the Ecore metamodel approach to defines the structure of the Extended Actigram Star and BPMN models. For that, XML Metadata Interchange (XMI) is used to save source and target models and also mapping rules. This rules is implemented using Atlas Transformation Language (ATL). It relies on XSLT transformation conforms to the BPMN modeler requirements.

The mapping rules creates correspondences and links between concepts and their relations from EA* and BPMN language. The modeling tool SLMToolBox supports the transformation and the mapping of activities with human interaction (e.g. Pilot Manager).

The second model transformation (3) is an enrichment of the BPMN model, in which specialized humans resources add more technological details in the process. In this paper, we intend to improve security aspects around the business process with specific details about security activities as depicted in Figure 4.

The model transformation (4) is the unification of differents BPMN of the process in a unique BPMN diagram. The business process management suite (jBPM) imports an BPMN diagram. (5) and (6) represent the integration and execution of the workflow in the jBPM suite. In the execution, some security services implemented in Web Services (JAX-RS technology) are instantiated and used by the workflow (7).

5 TESTS

This section describes the execution of the process based on a Business Process Management (BPM) Suite, which is responsible to control the execution of the process and enable perform services calls. This approach is essential to produces the services orchestration based on the model driven service engineering architecture.

The unified BPMN is imported to the Kie Workbench to execute following services calls. This environment requires some configurations of the workflow as well as variables to control the process' workflow, and also, request calls to Web Services are defined.

Figure 6 presents a workflow within the KIE Workbench. This solution has an integration for rule orchestration/flow offering process management of a combination of processes, rules and events.

To simplify the workflow, the unified BPMN imported centralized different pools/lanes in a unique pool. In this case, there is a abstraction of these details in the execution of the process. Two types of activities are used in this process to perform services calls and users tasks to interacting with human resources to receive inputs and show the service's results.

The process is composed by some user tasks at design time to define configurations about data collection and runtime security monitoring. For example, the manufacturing pilot manager is responsible to virtualize his resources in the C2NET platform. So, a set of resource's configurations need to be uploaded in this tasks to be published in the platform. Considering security issues, a security service is responsible to collect information about resources to be implemented some security mechanisms to protect them.

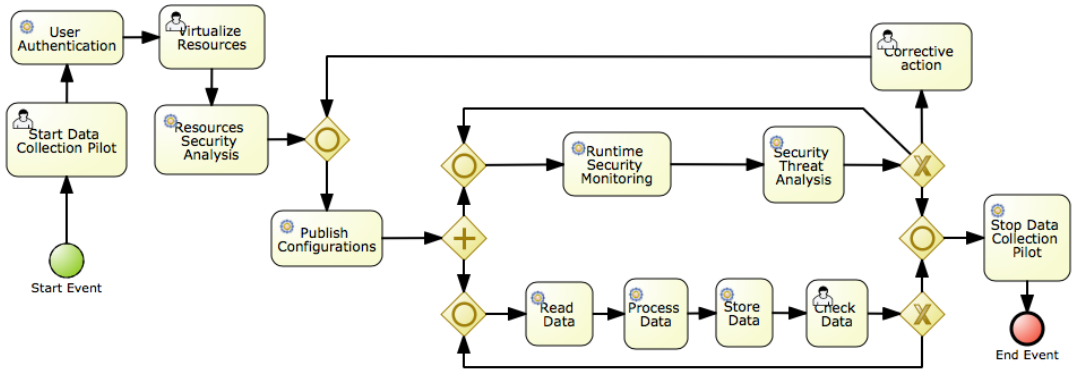

Figure 6: BPMN of security management process.

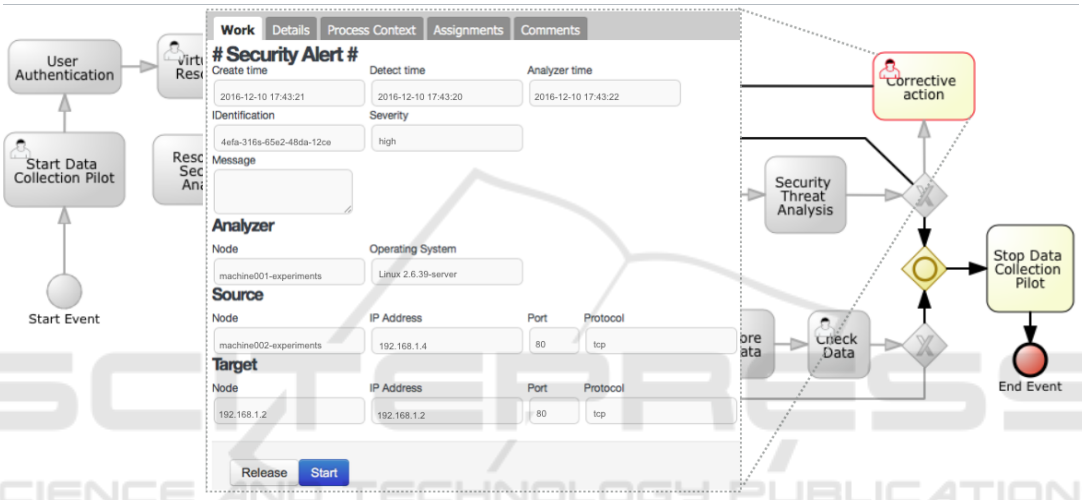

Figure 7: Form of a security alert generated by a security tool.

This information will be used by the runtime security monitoring service to start specific tools according information collected.

After to publish configurations in the C2NET platform, the workflow starts two parallels tasks such as Data Collection and Security Monitoring, respectively. Data collection tasks are responsible to read data of resources in the sensor network that it will be processed based on data filtering according to data standards of the C2NET project.

Meanwhile, security mechanisms are used to monitor resources based on previous information about devices. Each mechanisms report security alerts when an anomaly is detected. These alerts are important to the security process to identify potential solutions based on the threat detected. In order to facilitate the interoperability across a wide range of security mechanisms, IETF INtrusion Detection Platforms Working Group (Debar et al., 2007), established since 2007, has defined a standard interoperable message-type formats to carry alerts or notifications corresponding to events as well as parameterization commands. Many research contributions and projects have adopted the XML-IDMEF that are available in different languagues such as C, C++ Java and Python.

Security alerts are formatted based on this standard format, which each alert IDMEF-Message is composed with such information: CreateTime, DetectTime, AnalyzerTime, Source (Node, User, Process or Service), Target (Node, User, Process or Service), Classification, Assessment, AdditionalData (Debar et al., 2007).

Some information are used in this paper to describe a security alert to identify mechanisms that can be used to recover by a human resource. In this paper, a form is used to show detailed information about an anomaly detection of a security tool. Follow section will describe an execution of the workflow based on detected threat and forwarded to corrective action.

The execution of workflow allows to perform several services calls and to provide an interaction with human resources based on some model transformation from business models to BPMN diagram. In this simple example, the data collection process emcom-

pass a security monitoring service to identify potential threat in the sensor network.

Figure 7 demonstrated a human task (Corrective Action) responsible to the perform some configurations to recover a detected threat. This interaction with a security expert is enriched with the result of Security Threat Analysis, which proposes security mechanims or only changes in security polices to recover an anomaly to the network.

This security analysis enables to identify false positives that can occur and it must not interrup the workflow. In contrast, some problems can result to the end of the workflow. For instance, if an threat results in a critical problem for the infrastructure.

In this context, the human receive detailed information about detected threats and security solutions to be implemented.

6 CONCLUSIONS AND FUTURE WORKS

Cooperative networks include organizations and businesses that are owned and managed by the people providing products and supplying of services. The creation of new services requires time and effort to design effective services. The adoption of a methodology of the Model Driven Service Engineering Architecture (MDSEA) implements the process of model transformation from business details to specific artifacts.

The paper presented a implementation of a security management process in a business model of the manufacturing scenario of C2NET project as well as the execution of model transformations following a methodology MDSEA to enrich specific details during the process.

Due to the facility to construct services based on model driven engineering behind manufacturing enterprises, the methodology MDSEA could be implemented to demonstrate the transition between business models to security artifacts and implementations. This implementation of the security management process provided a view to understand how to enrich a simple business model based on a methodology as well as transforming high-level abstraction models to add specific details of security activities.

As future works, we intend to use model driven engineering to support services development that should to be executed in this security management process. Also, the integration of ontology-based solution need to be studied for the adoption of an ontology-based framework to support making decision in realtime.

ACKNOWLEDGEMENTS

The research leading to this work has received funding from CAPES Proc. N^{o}: BEX 0966/15-0 and European Commission's Horizon 2020 Programme (H2020/2014-2020) under grant agreement: C2NET N^{o}: 636909.

REFERENCES

Agostinho, C., Černỳ, J., and Jardim-Goncalves, R. (2012). Mda-based interoperability establishment using language independent information models. In *International IFIP Working Conference on Enterprise Interoperability*, pages 146–160. Springer.

Atkinson, C. and Kuhne, T. (2003). Model-driven development: a metamodeling foundation. *IEEE software*, 20(5):36–41.

Bahill, A. T. and Gissing, B. (1998). Re-evaluating systems engineering concepts using systems thinking. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(4):516–527.

Basin, D., Doser, J., and Lodderstedt, T. (2003). Model driven security for process-oriented systems. In *Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 100–109. ACM.

Bazoun, H., Ribault, J., Zacharewicz, G., Ducq, Y., and Boyé, H. (2016). Slmtoolbox: Enterprise service process modeling and simulation by coupling devs and services workflow. *International Journal of Simulation and Process Modelling*.

Bazoun, H., Zacharewicz, G., Ducq, Y., and Boye, H. (2013). Transformation of extended actigram star to bpmn2.0 and simulation model in the frame of model driven service engineering architecture. In *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative M&S Symposium*, page 20. Society for Computer Simulation International.

Bazoun, H., Zacharewicz, G., Ducq, Y., and Boyé, H. (2014). Slmtoolbox: An implementation of mdsea for servitisation and enterprise interoperability. In *Enterprise Interoperability VI*, pages 101–111. Springer.

Bézivin, J. (2006). Model driven engineering: An emerging technical space. In *Generative and transformational techniques in software engineering*, pages 36–64. Springer.

Brucker, A. D., Hang, I., Lückemeyer, G., and Ruparel, R. (2012). Securebpmn: Modeling and enforcing access control requirements in business processes. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, pages 123–126. ACM.

Butun, I., Morgera, S. D., and Sankar, R. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 16(1):266–282.

Chen, D. (2015). A methodology for developing service in virtual manufacturing environment. *Annual Reviews in Control*, 39:102–117.

De Castro, V., Marcos, E., and Vara, J. M. (2011). Applying cim-to-pim model transformations for the service-oriented development of information systems. *Information and Software Technology*, 53(1):87–105.

Debar, H., Curry, D. A., and Feinstein, B. S. (2007). The intrusion detection message exchange format (idmef). *IETF, NEtwork Working Group*.

Derdour, M., Alti, A., Gasmi, M., and Roose, P. (2015). Security architecture metamodel for model driven security. *Journal of Innovation in Digital Ecosystems*, 2(1):55–70.

Ducq, Y., Agostinho, C., Chen, D., Zacharewicz, G., and Goncalves, R. (2014). Generic methodology for service engineering based on service modelling and model transformation. *Manufacturing Service Ecosystem. Achievements of the European 7th FP FoF-ICT Project MSEE: Manufacturing SErvice Ecosystem (Grant No. 284860). Eds. Weisner S, Guglielmina C, Gusmeroli S, Doumeingts G*, pages 41–49.

Hossain, M. S. and Raghunathan, V. (2010). Aegis: A lightweight firewall for wireless sensor networks. In *International Conference on Distributed Computing in Sensor Systems*, pages 258–272. Springer.

Jürjens, J. (2002). Umlsec: Extending uml for secure systems development. In *International Conference on The Unified Modeling Language*, pages 412–425. Springer.

Lambert, J. H., Jennings, R. K., and Joshi, N. N. (2006). Integration of risk identification with business process models. *Systems engineering*, 9(3):187–198.

Lodderstedt, T., Basin, D., and Doser, J. (2002). Secureuml: A uml-based modeling language for model-driven security. In *International Conference on the Unified Modeling Language*, pages 426–441. Springer.

Mellor, S. J., Clark, T., and Futagami, T. (2003). Model-driven development: guest editors' introduction. *IEEE software*, 20(5):14–18.

Menzel, M., Thomas, I., and Meinel, C. (2009). Security requirements specification in service-oriented business process management. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, pages 41–48. IEEE.

Mozzaquatro, B. A., Jardim-goncalves, R., and Agostinho, C. (2015). Towards a reference ontology for security in the internet of things. In *IEEE International Workshop on Measurement and Networking*, pages 1–6.

Mozzaquatro, B. A., Melo, R., Agostinho, C., and Jardim-Goncalves, R. (2016). An ontology-based security framework for decision-making in industrial systems. In *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development*, pages 779–788.

Münch, J., Armbrust, O., Kowalczyk, M., and Soto, M. (2012). Process modeling notations and tools. In *Software Process Definition and Management*, pages 111–138. Springer.

Patel, H. B., Jinwala, D. C., and Patel, D. R. (2016). Baseline intrusion detection framework for 6lowpan devices. In *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, pages 72–76. ACM.

Ren, J. and Taylor, R. (2005). A secure software architecture description language. In *Workshop on Software Security Assurance Tools, Techniques, and Metrics*, pages 82–89.

Seidewitz, E. (2003). What models mean. *IEEE software*, 20(5):26.

Selic, B. (2003). The pragmatics of model-driven development. *IEEE software*, 20(5):19.

Wolter, C., Menzel, M., Schaad, A., Miseldine, P., and Meinel, C. (2009). Model-driven business process security requirement specification. *Journal of Systems Architecture*, 55(4):211–223.