IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Continuous and Secure Evolution of Software Models

# Methodology protocol description

Chahrazed Boudjemila

IMT-Atlantique engineering school in Brest
P4S, IMT-Atlantique

# Contents

**1 Introduction** **1**

**2 Methodology** **2**

2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.2 Meta-models and Models . . . . . . . . . . . . . . . . . . . . . 2

2.3 Identify inter-model correspondences . . . . . . . . . . . . . . . 2

2.4 Security Rules . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2.5 Model evolution . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**3 Evaluation Protocol** **4**

3.1 Methodology Evaluation Guidelines . . . . . . . . . . . . . . . . 4

3.1.1 The evaluation purpose . . . . . . . . . . . . . . . . . . . . 4

3.1.2 Evaluation Plan . . . . . . . . . . . . . . . . . . . . . . . . 4

3.1.3 Case Study for the evaluation of the Methodology . . . . . 8

**4 Security Specification** **11**

**5 Matching and security annotations tables** **12**

# 1 Introduction

Model Driven Engineering (MDE) is an approach to systems engineering that focuses on using models as primary artifacts to represent system requirements, architecture, behavior, and other aspects as system constraints and validation. This approach is for designing, analyzing, and managing complex systems throughout their lifecycle. In the context of MDE, we propose a methodology based on the model federation paradigm which integrates within the design phase of the system's development process. The purpose of our methodology is to detect security inconsistency of a multi-model system under study.

This document presents a detailed protocol for evaluating our methodology based on model federation. It describes the methodology evaluation process, covering key aspects such as the methodology overview, the evaluation purpose, the federation tool, the evaluation criteria, and the evaluation plan. The evaluation criteria include methodology comprehension, consistency, usability and user experience, adaptability, and implementation facilities. Additionally, the evaluation plan depicts the users involved, the schedule, and resources, and outlines the steps to be followed throughout the evaluation process.

# 2 Methodology

## 2.1 Introduction

Our methodology outlines the essential elements and processes necessary to design and maintain security consistency across heterogeneous models. This methodology is integrated into the design phase of the system development process, incorporating new tasks and artifacts.

## 2.2 Meta-models and Models

The proposed methodology does not rely on any particular Modeling Language. This ensures that it can be effectively applied across a wide range of contexts and systems. The selection of models and meta-models depends on the specific system under study and the objectives of the designers. They choose the models according to the particular aspects of the system they aim to address, as well as their individual needs and requirements.

In Model Driven Engineering (MDE), various meta-models and models are used to represent different aspects of a system such as: UML, BPMN, SysML, OWL, EMF. In the context of our methodology, we choose some meta-models and models incorporating security information to ensure that the models effectively address security concerns within the system design. This approach enables us to analyze and validate security consistency across models.

## 2.3 Identify inter-model correspondences

Complex systems are represented using various models. Each model focuses on specific aspects such as requirement, behavior, implementation, data structuring. Additionally, these models may be created by different designers using various modeling languages and tools. However, to provide a comprehensive view of the system, the elements of these various models need to be linked. These established dependencies ensure that: 1) the aspects of system are represented accurately across each model; 2) verify the impact of changes on multi-models to detect security inconsistency.

Various approaches have been developed to establish correspondences between heterogeneous models such as unification and integration, as well as federation approaches. Our methodology is based on the model federation approach to establish correspondences between elements of models.

The model federation approach involves creating correspondences across multi-models to provide a unified view of a complex system while keeping models in their technology space. It allows users to establish correspondences between different model elements created using different modeling language and tools. The model federation approach plays a crucial role in enabling users to manage elements represented in separates models, ensuring coherence, evolution and security consistency.

Understanding the relationships between meta-model elements is essential for identifying correspondences between model elements. We based on the different research done in MDD (Model-Driven Development) to identify potential relationships between meta-model elements.This relation may mean equivalence, dependency, specialization, etc. The provided table 3 showcases correspondences between BPMN, UML, and EMF meta-model elements. Our focus is on establishing correspondence that is relevant to security. Identifying these correspondences requires examining additional information at the model level, including any integrated security information.

## 2.4 Security Rules

Model Driven Engineering (MDE) promotes the representation of systems through various models that capture different aspects, such as functional and architectural perspectives. These models are dependent as they represent different views of the system. In our case, we verify inter-model consistency w.r.t to security. Ensuring the consistency of security information across models, requires adding inter-model constraints, which we refer as the security rules.

Security rules represent inter-model security constraints to ensure that security requirements are consistently applied across the various models. These rules may address data integrity, access control, and confidentiality and other security aspects to protect sensitive information from unauthorized access. They ensure that all models comply with relevant security standards and regulations.

To define these security rules, we need to first select the added security information attached to model elements which are linked (exist a correspondence between them) and then analyze its semantics. Finally, we must specify the security rules regarding the semantics of the security information. As an example of such inter-model security constraints, if a model contains data annotated as ”Privacy”, it indicates the presence of sensitive information within that model. In response, a security rule may dictate the encryption of this data before transmission in the deployment model.

## 2.5 Model evolution

Systems evolve over time due to changes in requirements, technologies, or deployment. The models representing these systems must also evolve accordingly by modifying existing models (add, modify and remove model elements). The management of the model evolution is essential to ensure that the information remains consistent across models that depict the system. In our methodology, we focus in analyzing changes that have been identified that could impact the security concerns of the models. Specifically, we select the correspondences related to the changed elements and then the security rules attached to the selected correspondences are evaluated to verify the security consistency of models system.

# 3 Evaluation Protocol

## 3.1 Methodology Evaluation Guidelines

### 3.1.1 The evaluation purpose

The purpose of the evaluation is to identify both strengths and weaknesses of the methodology while also validating its applicability across various contexts. Additionally, it seeks to validate its usability, effectiveness, and user experience, ensuring that it offers improved facilities for users. Furthermore, this evaluation will enable us to improve and refine various process steps within our methodology based on the feedback of users.

### 3.1.2 Evaluation Plan

Throughout the evaluation process, we will assist users in implementing our methodology to ensure security consistency across their various models in the selected case study. To achieve this, we divide the evaluation into three phases:

1. *Resource Supply Phase:* In this phase, we outline the methodology's purpose, its various phases to users and what we expect from users via a presentation. Additionally, we present a different case study in detail and equip users with essential resource files to evaluate the methodology. The list of resources is prepared and printed before the evaluation session (the duration of the presentation will be 15 min and 5 minutes for user question).

   (a) We offer users a variety of resource files, as listed below:

      - A list of models involved in the case study.
      - Tables detailing the security annotations used in each model, along with their corresponding security concerns.
      - Tables containing matching rules for elements of the models involved in the case study at meta-model level, facilitating the identification of correspondences.

- The questionnaire 1 to assess the user level of knowledge in the domain of modeling system engineering
- A File containing an empty table where they can identify corresponding elements of the models for each case study and specify the security rules to attach to the identified correspondences.

(b) During this stage, the user must fill out the questionnaire to help us assist them during the application process of our methodology.

(c) *The criteria for evaluating this phase rely on user answers to questionnaire 1, and their reaction* :

- If user has understood the methodology and its purpose
- If user has understood the basics of the case study
- If the time reserved for this step is respected.

2. *Examination phase:* This phase aims to guide users to understand the models and their associated security annotations to identify correspondences between the provided models files (we reserve 1 hour for this phase).

   (a) In this phase, all the necessary files are already provided to the user in the preceding step. However, our role in this phase consists of :

   - Providing users with detailed explanations of the various models involved in the case study, including their purpose, structure, key elements, and the security annotations applied.
   - Answering users' questions.
   - Clarifying the significance of each security annotation by explaining the table that detailing the security annotations with the corresponding security concerns.

   (b) At the end of this phase, the user is required to complete the table by identifying the corresponding elements in the models without filling the security rules column.

   (c) *The criteria to evaluate this phase:*

   - *Capacity to establish correspondences between different model elements:* upon receiving the completed match file from the user, we assess the matches identified by the user by comparing them with our identified correspondences.
   - User's ability to identify security information within models
   - If the time reserved for this step is respected.

3. *Implementation Phase:* this phase consists of creating the security federation utilizing tables of matching rules to establish correspondences between elements of different models in the case study and the implementation of the provided security specifications (we reserve 1 hour to this phase). The environment of the implementation (install Openflexo, creating a project, importing the models, declare elements types of models etc. ) is prepared before the evaluation session :

   (a) In this phase, our role consist on:

      - Give instructions for using the Openflexo environment.
      - In this step, we have two scenarios. In the first scenario, we provide the user with both the security rules specification file and the implementation of the security rules. In the second scenario, we only provide the user with the security rules specification (we reserve 1 hour to this phase) .

   (b) At the end of this phase, the user is required to do the following :

      - The user is required to complete the security rules column, adding the security rule specifications for each identified correspondence.
      - The user needs to establish correspondences and attach security rules (implementing them if they only has the specification) using the provided implementation environment (Openflexo).
      - The user must fill the rest of questionnaire 2 about the experience.

   (c) *The criteria to evaluate this phase by comparing their work to ours:*

      - If the security rules are attached to the adequate correspondences.
      - If the security rule corresponds to the implementation of the security specification.
      - If the time reserved for this step is respected.

   Below is a table summarizing the individual responsible, task, and duration for each phase.

| | **Assessment coordinator (me)** | **User** | **Time reserved** |
|---|---|---|---|
| ***Phase 1:***<br>***Resource Supply Phase*** | Present the methodology and its purpose<br>Provide matching and security annotation tables<br>Provide case study description and description<br>Provide Questionnaires and empty table to fill | Fill the questionnaire 1 | 20min (15 min for the presentation + 5 min for user questions) |
| ***Phase 2:***<br>**Examination phase** | Explain models to user if he ask and answer to user's questions<br>Clarify the significance of each security annotation | Identify the correspondences between elements of models | 1 hour |
| ***Phase 3:***<br>**Implementation Phase** | Give instructions for using the Openflexo environment<br>Give user the security specification file | Establish the correspondences between elements of models<br>Attach the security specification to the adequate correspondence and implement them<br>Fill the questionnaire 2 | 1 hour |

Table 1: table summarizing the individual responsible, task, and duration for each phase

### 3.1.3 Case Study for the evaluation of the Methodology

1. *Case Study 1 : Instant messaging service* PICO (Presence and Instant Communication) a very simplified representation of ICQ or America Online Instant Messenger. Initially, users need to register with a PICO server. Once registered, they can indicate their availability and exchange brief instant messages with each other.

    - *Security Requirements :*

        (a) Messages sent must be transmitted using secure channels

        (b) Messages must be Confidential

        (c) Sender user must be authenticated

        (d) Users information must be private

    - *Models and Meta-Models :* The BPMN model is constructed using the secBPMN language, while the Deployment and Data models are developed using UMLsec.

        - Data Model
        - BPMN model
        - Deployment model

2. *Case Study 2 : iTrust is an open-source medical system developed in 2005 at the University of North Carolina with the aim of providing software engineering students with a complex real-world system for experimentation. In this sense, it has already been used in many different research works as a realistic use case. Regarding security, iTrust has requirements related to privacy, confidentiality, and integrity*

    - *Security Requirements :*

        (a) User's ID are private

        (b) The patient information are not updated by unauthorised person

        (c) Support an actor with multiple role

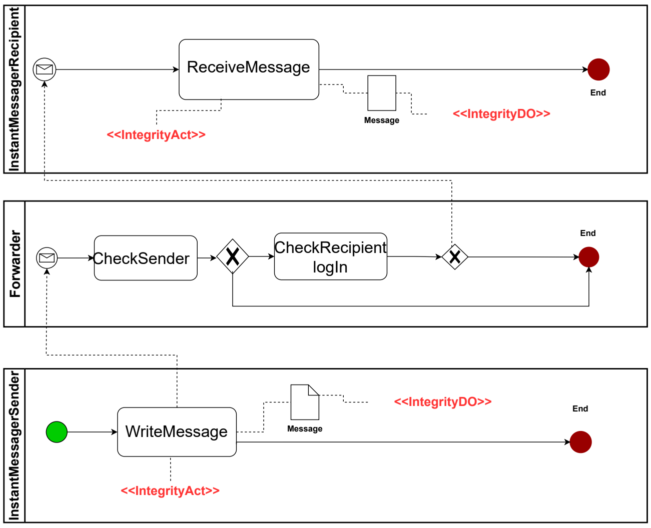
        (d) Users information must be private

    - *Models and Meta-Models :* The BPMN model is constructed using the secBPMN language, while the Deployment and Data models are developed using UMLsec.

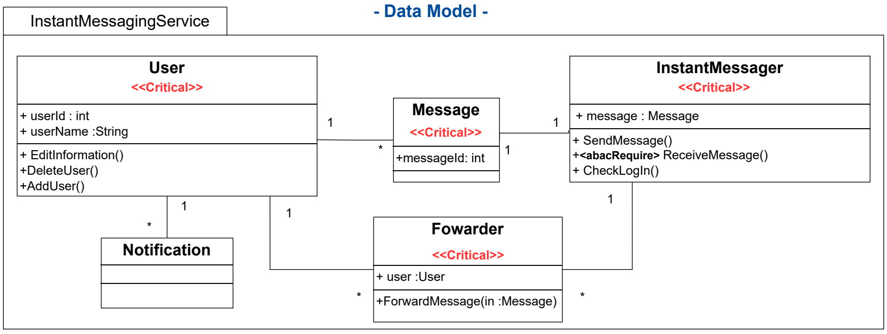
        - Data Model
        - BPMN model
        - Access Control Model

## Models for case study 1 :  Instant messaging service

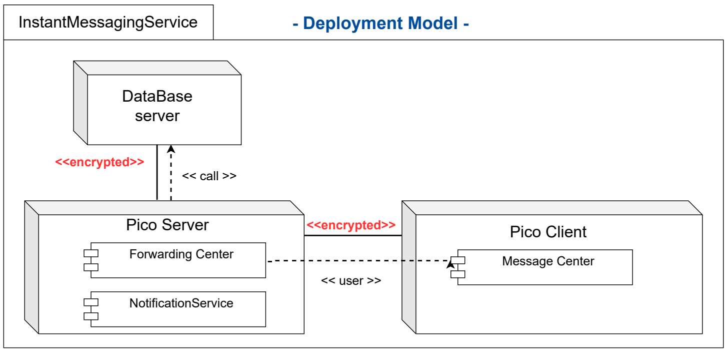
- BPMN Model -



- Data Model -

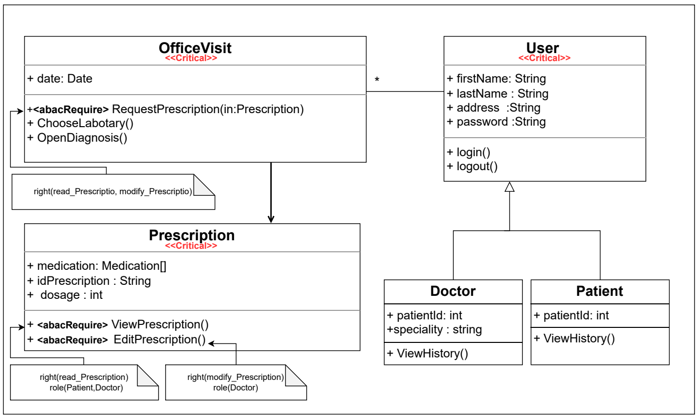


- Deployment Model -

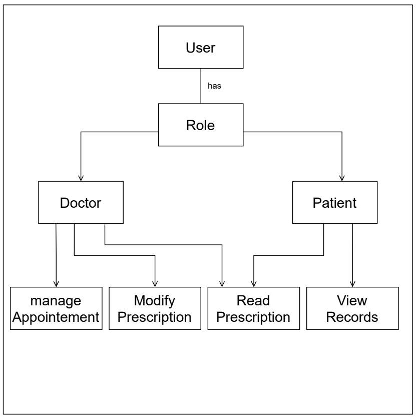


## Models for case study 2 : ITrust system medical

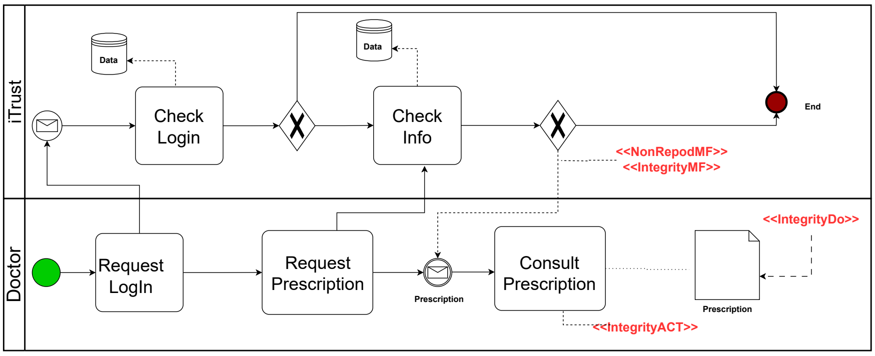
- Data Model -



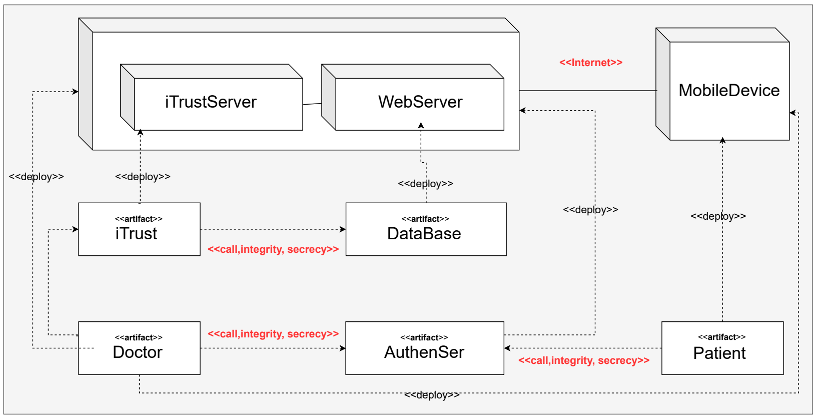
- Access Control Model -



- BPMN Model -



- Deployment Model -



# 4 Security Specification

1. *security Rules for case Study 1: Instant messaging service*

   **Security Rule-1** : When two associated classes in the data model (which corresponds to component in the model deployment), contains critical security annotation, the communication Path between the corresponding component must contains encrypted annotation. This rule attached to the correspondences between class and component.

   **Security Rule-2** : When the Operation in the data model (which corresponds to a Task in the BPMN model) annotated ”abacRequire” then the Task must be annotated ”integrityAct” or ”PrivacyAct”. This rule is attached to the correspondences between Operation and Task.

   **Security Rule-3** : When a Class in data model(which corresponds to Pool in the BPMN model), annotated as ”Critical” contains an Operation related to a Task of the corresponding Pool, the Task must be preceed by ”LogIn” Task. This rule is attached to the correspondences between class and pool

2. *security Rules for case Study 2: ITrust system medical*

   **Security Rule-1** : each user in the defined data model must have an associated role in the Access Control model. This rule is added to the correspondences between User and Role.

   **Security Rule-2**: when the Pool in the BPMN model (that corresponds to Class in the data model), contains a task which is the source or target for a MessageFlow that is linked to security annotations. The operation representing the task must have security annotation “abacRequire” with tag “right=Message/Attribute”. This Security Rule is attached to the correspondences between Pool and Class.

   **Security Rule-3**: when the Operation in the data model (that corresponds to Task in the BPMN model), contains ”AbacRequire” annotation . The Task in the BPMN Pool must be annotated IntegrityACT . This Security Rule is attached to the correspondences between operation and task.

   **Security Rule-4**: when two pools in the BPMN model (that correspond to artifacts in the deployment model), communicate security-critical data and are deployed in different devices, communication between the two devices must occur through an encrypted channel. This rule is attached to correspondences between Pool and artifact.

# 5 Matching and security annotations tables

Below, we present three types of tables representing various mappings: matching elements and security annotations.

### 1- Table for comparing security annotations between secBPMN and UMLsec

Table 2: Mapping security annotations from secBPMN to UMLsec

| Security Annotation | BPMN | UML | |
|---|---|---|---|
| | | data model | deployment model |
| Integrity | IntegrityAct<br>IntegrityMF<br>IntegrityDO | dataSecurity, Critical<br>«integrity» «secrecy»<br>abac «right», abacRequire | Integrity<br>Secrecy |
| Confidentiality | ConfidentialityDO<br>ConfidentialityMF | Critical<br>«integrity» «secrecy»<br>abac «right», abacRequire | Integrity<br>Secrecy |
| Privacy | PrivacyDO<br>PrivacyAct | Critical | encrypted<br>Secrecy |
| Non-repudiation | NonRepudDO<br>NonRepudAct | secure Links<br>«secrecy» | secure dependecy<br>encypted<br>Integrity |
| Availability | AvailabilityDO<br>AvailabilityAct<br>AvailabilityMF | - | - |

Table 3: Matching elements between BPMN, UML and Access control model

| BPMN | AccessControl | UML | |
|---|---|---|---|
| | | data model | deployment model |
| Pool-Lane | User | Class | Artifact, Node |
| DataObject | - | Class | Artifact, Node |
| Task | Authorization | Operation | - |
| MessageFlow | - | Association | CommunicationPath, Association |

**Identify and complete the elements constituting each correspondence, and associate the appropriate security specifications with the correspondences for case study 1.**

Table 4: Identify correspondences and security specification

| Correspondence | N°Security Rule | Elements of correspondences and models | |
|---|---|---|---|
| | | Element1 + Model | Element2 + Model |
| Correspondence1 | | | |
| Correspondence2 | | | |
| Correspondence3 | | | |

**Identify and complete the elements constituting each correspondence, and associate the appropriate security specifications with the correspondences for case study 2.**

Table 5: Identify correspondences and security specification

| Correspondence | N°Security rule | Elements of correspondences and models | |
|---|---|---|---|
| | | Element1 + Model | Element2 + Model |
| Correspondence1 | | | |
| Correspondence2 | | | |
| Correspondence3 | | | |
| Correspondence4 | | | |