Formal Model Driven Engineering (FM & MDE)

Habilitation à Diriger des Recherches

Akram Idani

26 Mai 2023

Overview

- MDE
  - Communication
  - Standard

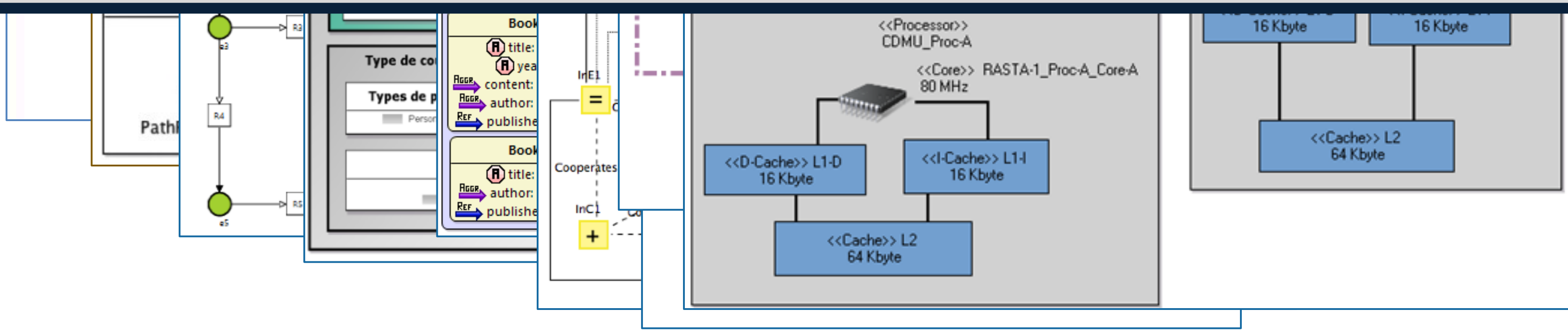
Proofs / Refinements

End User needs

- FM
  - Automated reasoning
  - Safety-critical systems

« *The picture changes dramatically for* ***safety-critical****, high-assurance software. Here, validation by testing reaches its limits and needs to be* ***complemented or even replaced*** *by the use of formal methods such as model checking, static analysis, and program proof.* »

Leroy, X.: Formal verification of a realistic compiler. Communications of the. ACM (2009)



Overview

- MDE
  - Communication
  - Standard

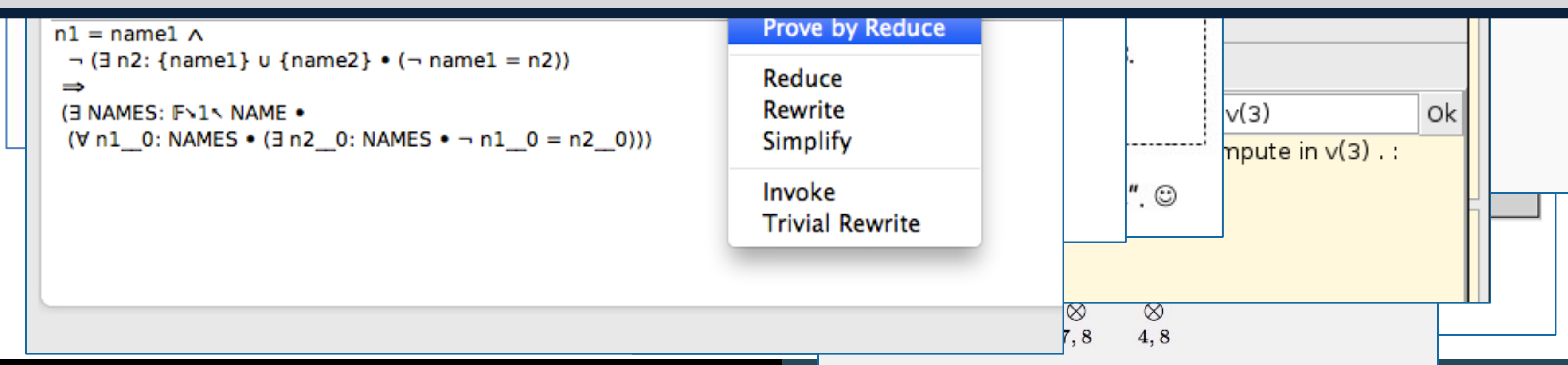
Proofs / Refinements

End User needs

- FM
  - Automated reasoning
  - Safety-critical systems

« [...] the learning curve of formal methods is steep, whereas the learning curve for drawing diagrams on the black board is very low. »

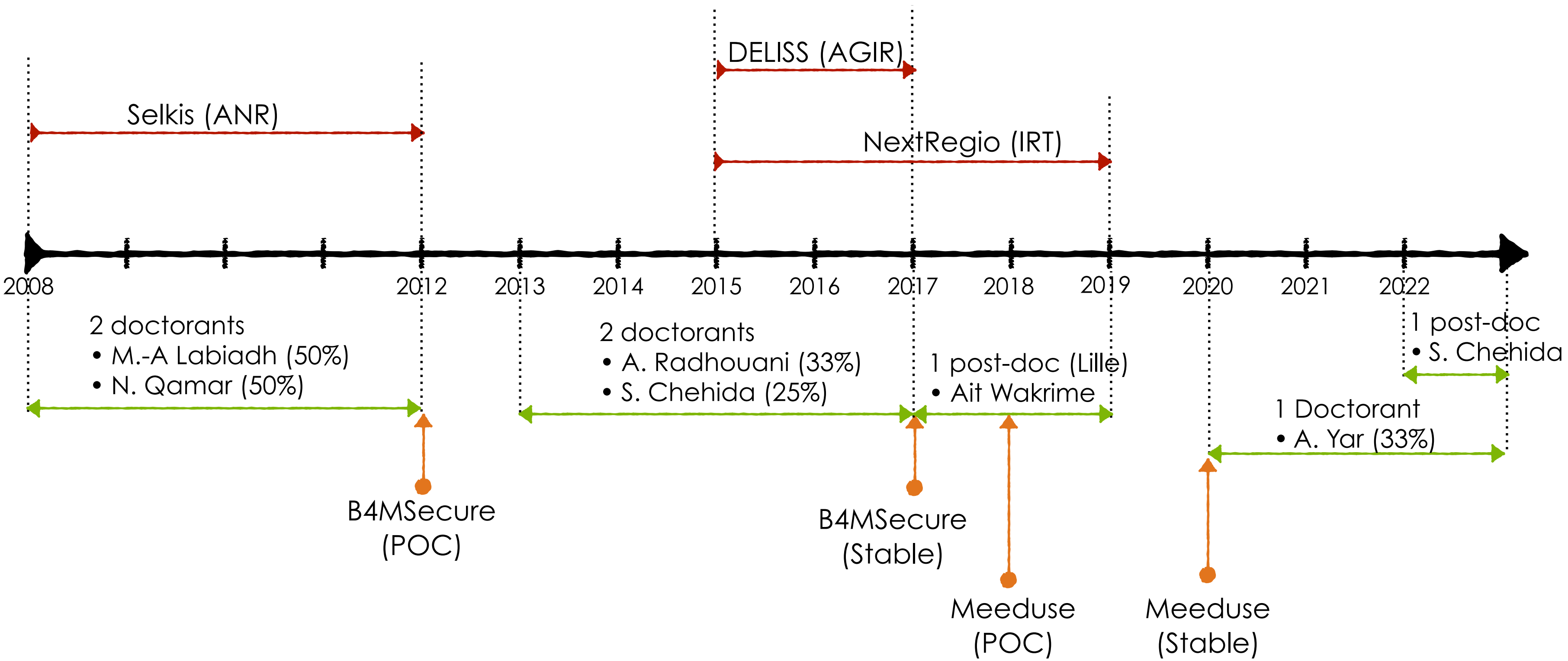
Andova, S. et al.,: MDE basics with a DSL focus. In: Formal Methods for Model-Driven Engineering. LNCS, vol. 7320 (2012).



Outline

- Introduction
- **Model Driven Security**
  - UML / SecureUML
  - Insider attacks
- **Executable DSLs**
  - Motivations and results
  - Applications
- Conclusion

Introduction



- **Selkis** (projet ANR, 2008/2012) : A development method of secure health care networks information systems : from requirements engineering to implementation
- **DELISS** (Alpes Grenoble Innovation Recherche, 2015/2017) : Déploiement validé de politiques de Sécurité en systèmes d'Information.
- **NextRegio** (IRT Railenium, 2015/2019) : solutions d’exploitation pour les lignes de desserte fine du territoire.

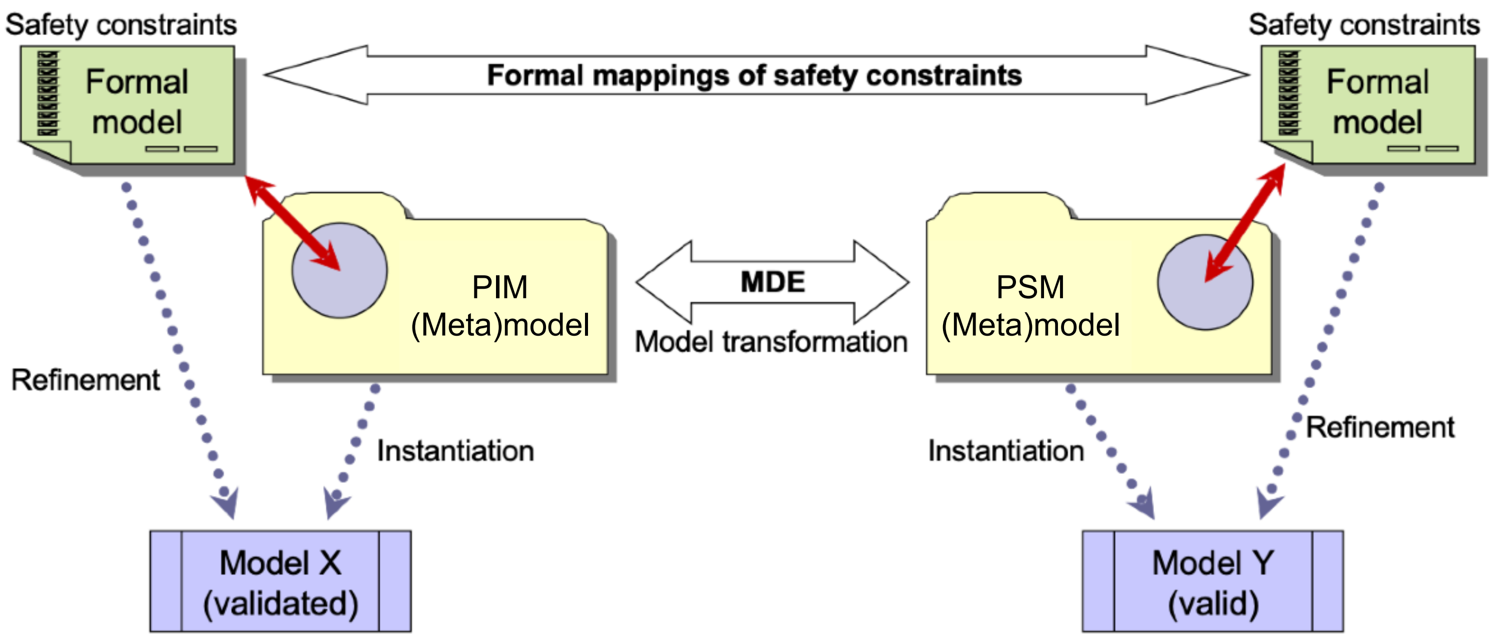
Introduction

Robert France, Bernhard Rumpe., Model-driven Development of Complex Software: A Research Roadmap. *International Conference on Software Engineering 2007.*

« *Members of the FST and the MDE communities need to collaborate.* »

« *Modeling languages must have formally defined semantics if they are to be used to create analyzable models.* »

My research project (2008)



Introduction

- Two schools
  - The Extensible General-Purpose Modeling Language School
    - e.g. SysML, OntoML, UML-RT
    - Provide a language with extension mechanisms
    - **B4MSecure: Model-Driven Security**
  - The Domain Specific Modeling Language School
    - e.g. Capella, OWL, Lustre
    - Tool support for engineering modeling languages
    - **Meeduse: grammars and Meta(-meta)-modeling**

Robert France, Bernhard Rumpe., Model-driven Development of Complex Software: A Research Roadmap. *International Conference on Software Engineering* 2007.

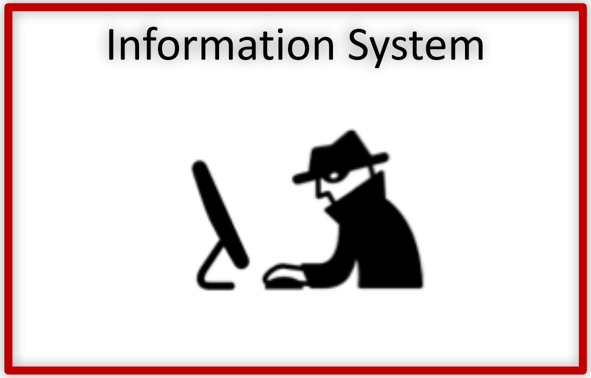
« […] *can both play vital roles in an MDE environment. We envisage that research in both schools will provide valuable insights and research results that will lead to a convergence of ideas.* »

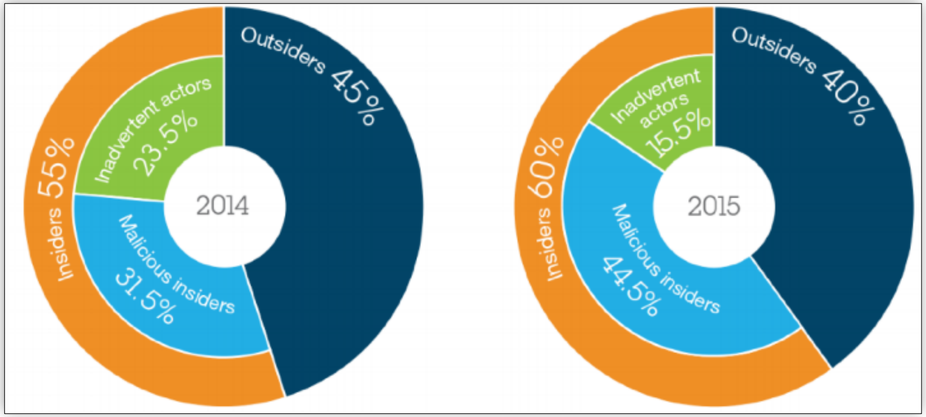
Outline

- Introduction
- **Model Driven Security**
  - UML / SecureUML
  - Insider attacks
- **Executable DSLs**
  - Motivations and results
  - Applications
- Conclusion

Access control

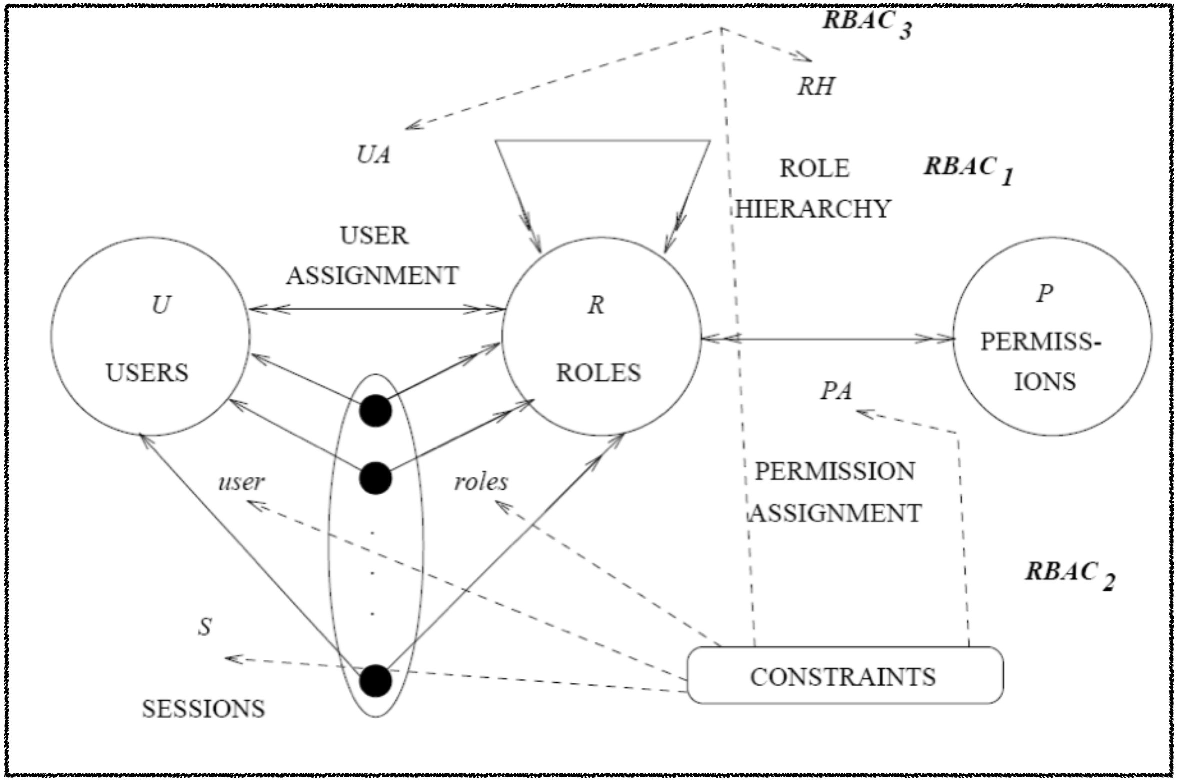
- Access control is horizontal :
  - Ad-hoc integration is error prone and costly
- High-level of abstraction :
  - modeling structure and behavior
- Correctness :
  - Testing / model-checking / Proofs
- Prevention against attacks





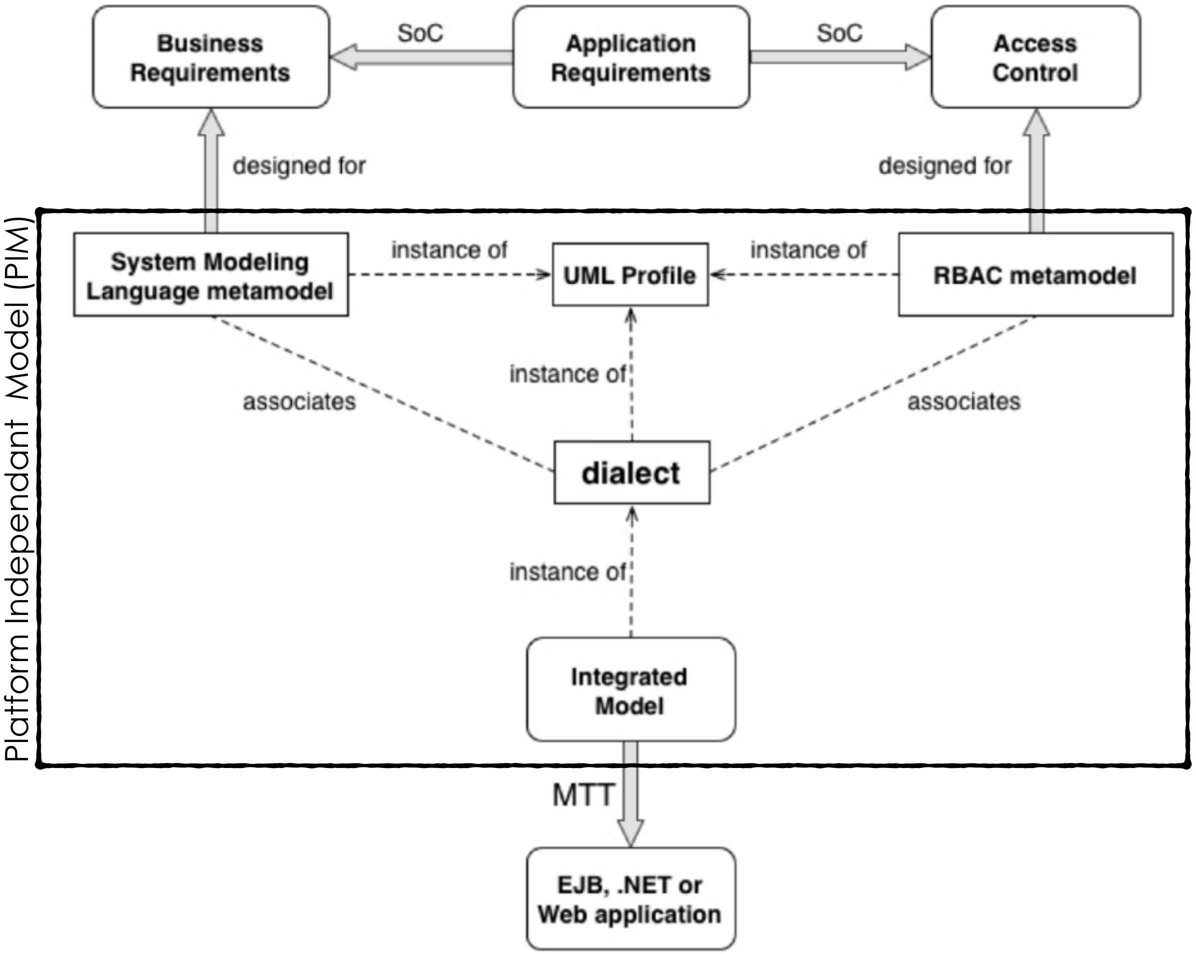
Role-Based Access Control

- RBAC : 1992 by D. Ferraiolo and R. Kuhn ☞ 2000
- Constraints
  - Static Separation of duties
  - Dynamic Separation of duties
  - Contextual
  - **Data-centric too**!
- ABAC : 2011



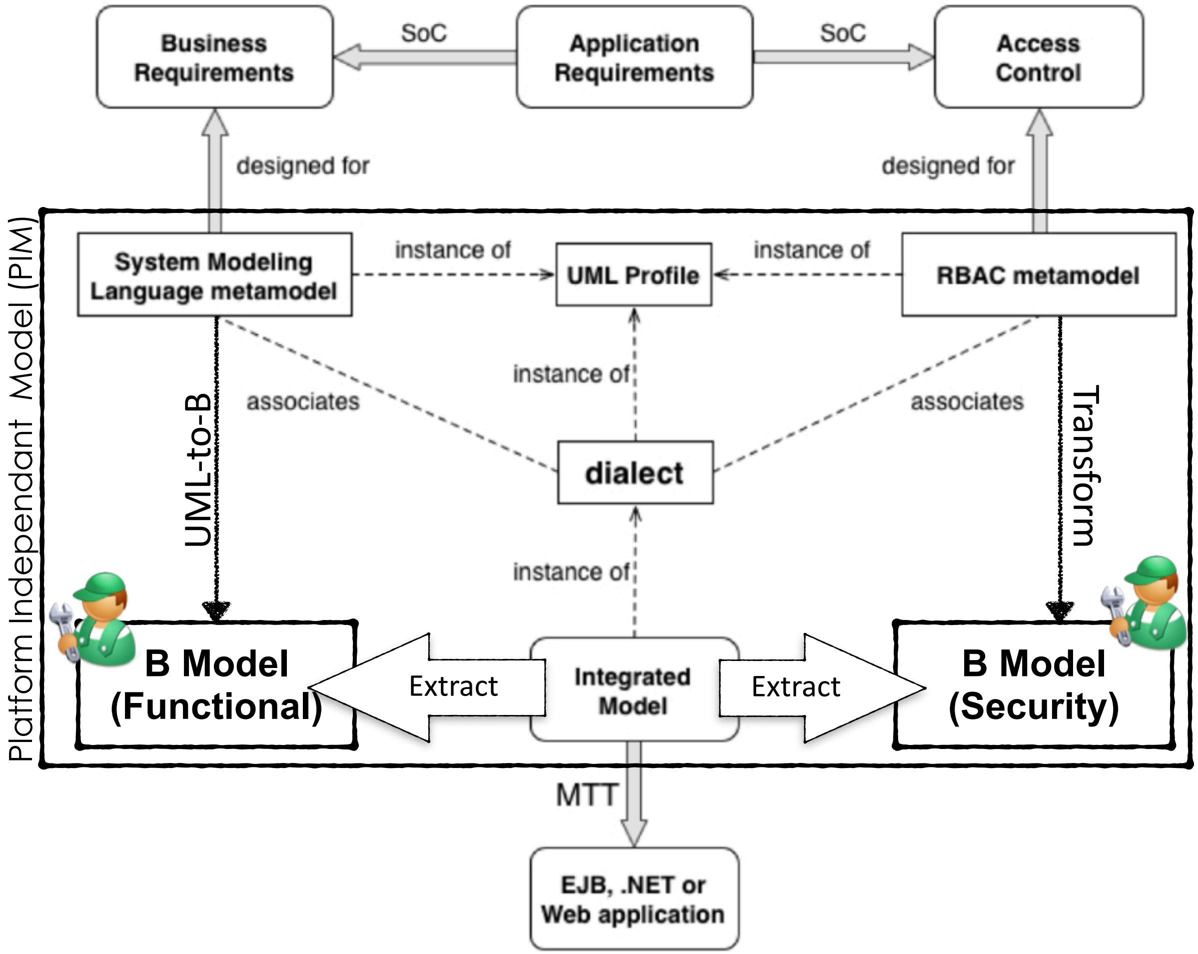
MDE for Security

- Separation of concerns
- *e.g.* SecureUML



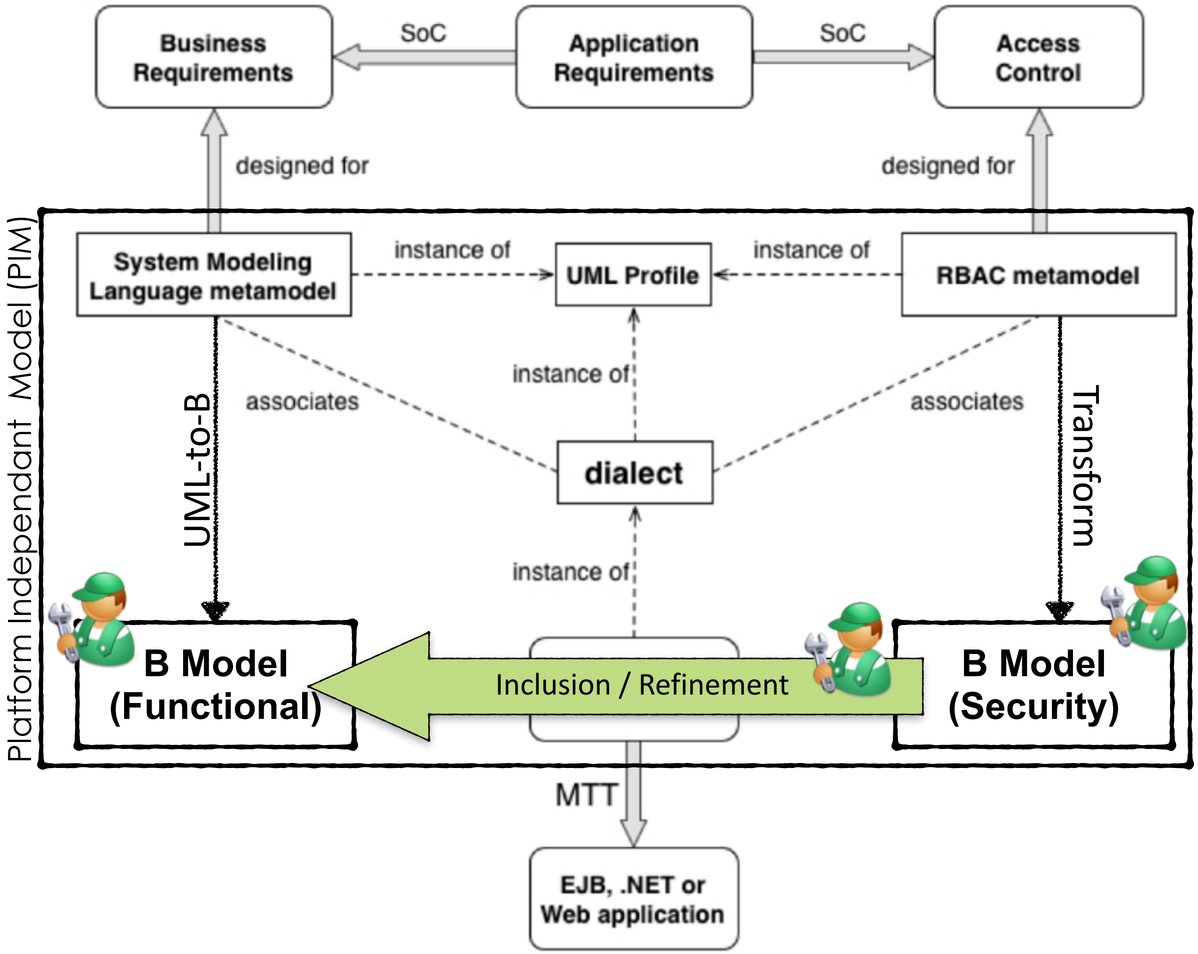
MDE for Security

- Separation of concerns
- e.g. SecureUML
- Contribution:
  - Formal MDS
    - Proof
    - Test/Animation
    - Model-checking

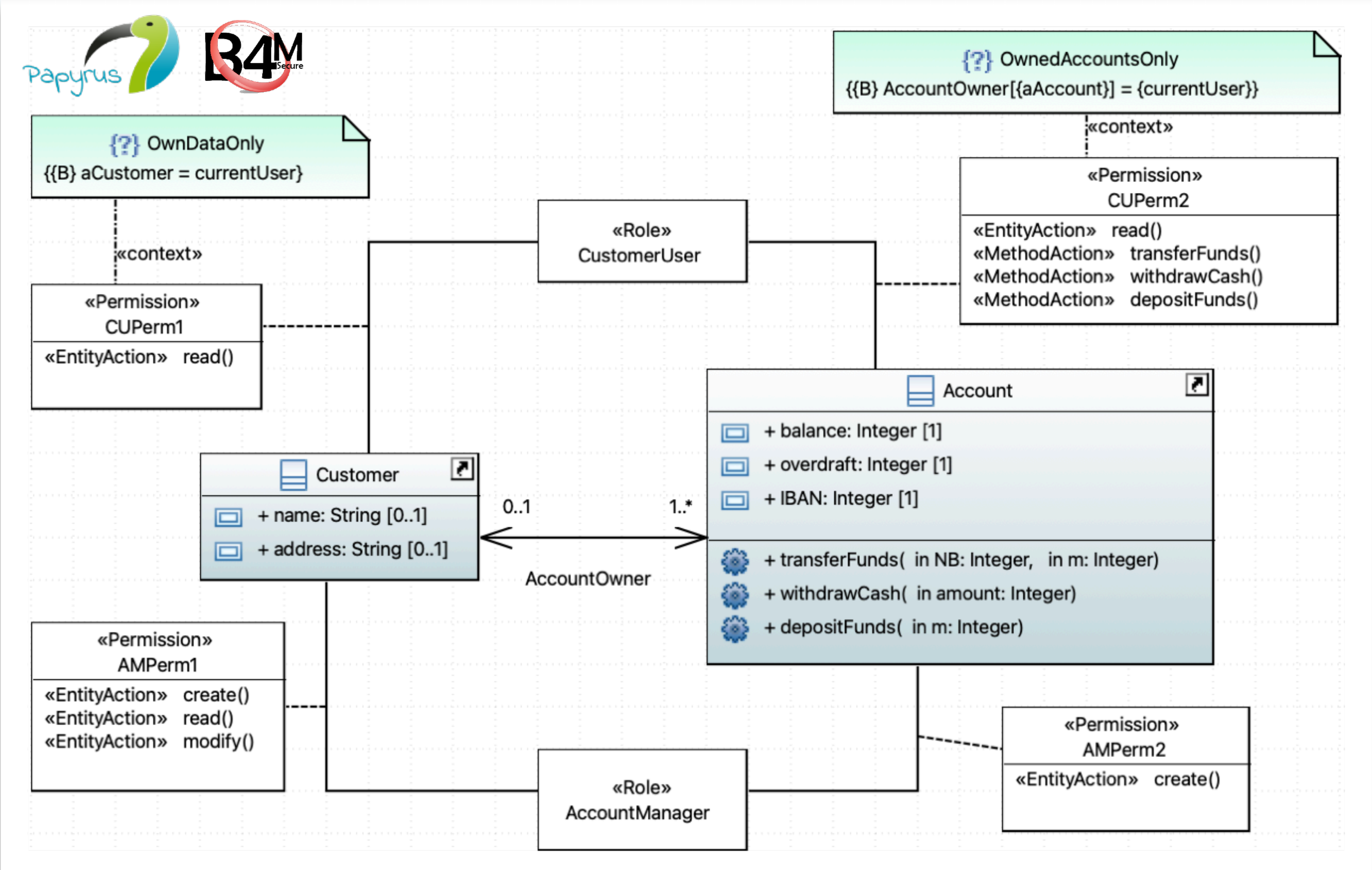


MDE for Security

- Separation of concerns
- e.g. SecureUML
- Contribution:
  - Formal MDS
    - Proof
    - Test/Animation
    - Model-checking



A Simple Example



Verification & Validation (1/2)

- Static V&V
  - USE [Kuhlmann et al. 2013] , SecureMOVA [Basin et al. 2009], Alloy-Analyser [Zao et al. 2003, Ahn et Hu 2007], etc.
  - Evaluation is done in a given state
    - Given an atomic action, which roles can perform this action?
    - Given a role and an atomic action, under which circumstances can a user in this role perform this action?
    - Do two permissions overlap?

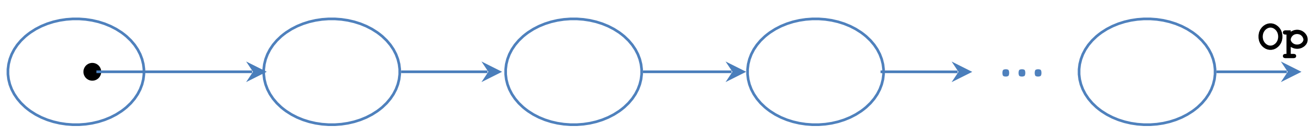
Is the manager able to transfer funds from a customer account?

A static query: **NO**

Verification & Validation (2/2)

- Dynamic V&V
  - **Jaza/animation [PhD of N. Qamar (50%)]: static queries + animation**
  - USE/OCLE [Yu et al. 2009], generation of execution schemas.
  - RW [Zhang et al. 2008], model-checking
  - SMP (Logic for State Modifying Policies) [Becker et Nana 2010]
  - Theorem proving [A. Mammar et al., FAC'15]

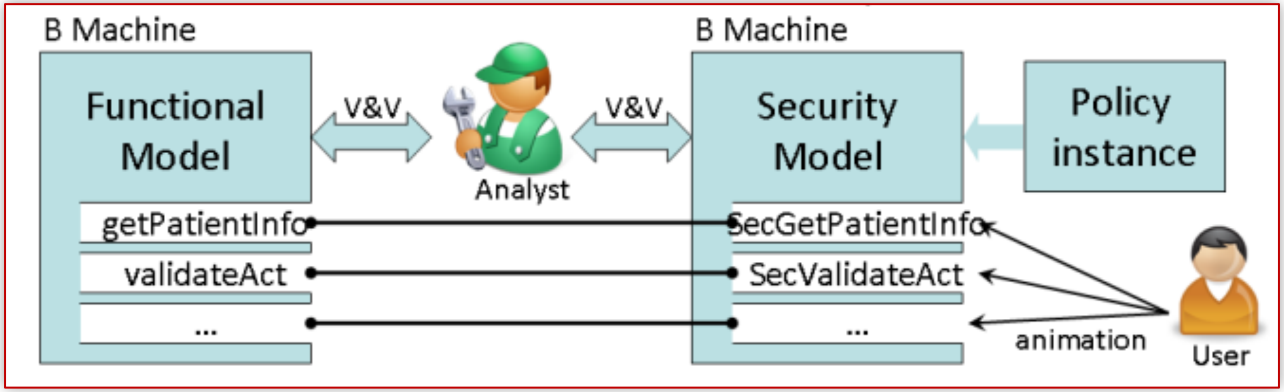
Is there a sequence of operations that can be executed by a manager in order to transfer funds from a customer's account?



Research directions

- Modeling/Testing
  - **Configurable transformations**
  - Abstraction levels: M1 ; M2 and M1/M2
  - PhD M.-A. Labiadh (50%) + 1 M2 + 1 PFE
- Attacks
  - Forward search
    - **Guided model-checking** (2 M2)
    - Ant-colony optimisation (2 IRL-ENSIMAG)
  - Backward search
    - **Proof** and constraint solving
    - PhD A. Radhouani (33%) + 1 M2
- Business processes
  - Task-based access control: PhD S. Chehida (25%)
  - BPMN: 2 M2, collaboration with SIGMA/LIG

Modeling/Testing



Modeling/Testing (1/3)

- Various tools and approaches, but
  - Most of them are unavailable
  - Should be revisited/updated

Class
attribut : TypeAttribut

```
MACHINE
  Class_CLASS
SETS
  Class_SET
VARIABLES
  Class, attribut
INVARIANT
  Class ∈ ℙ(Class_SET)
  attribut ∈ Class → TypeAttribut
```

(Snook *et al.*, 2004)

```
MACHINE
  BasicClass
SETS
  CLASS
VARIABLES
  Class, attribut
INVARIANT
  Class ⊆ CLASS
  attribut ∈ Class → TypeAttribut
```

(Laleau, 2002)

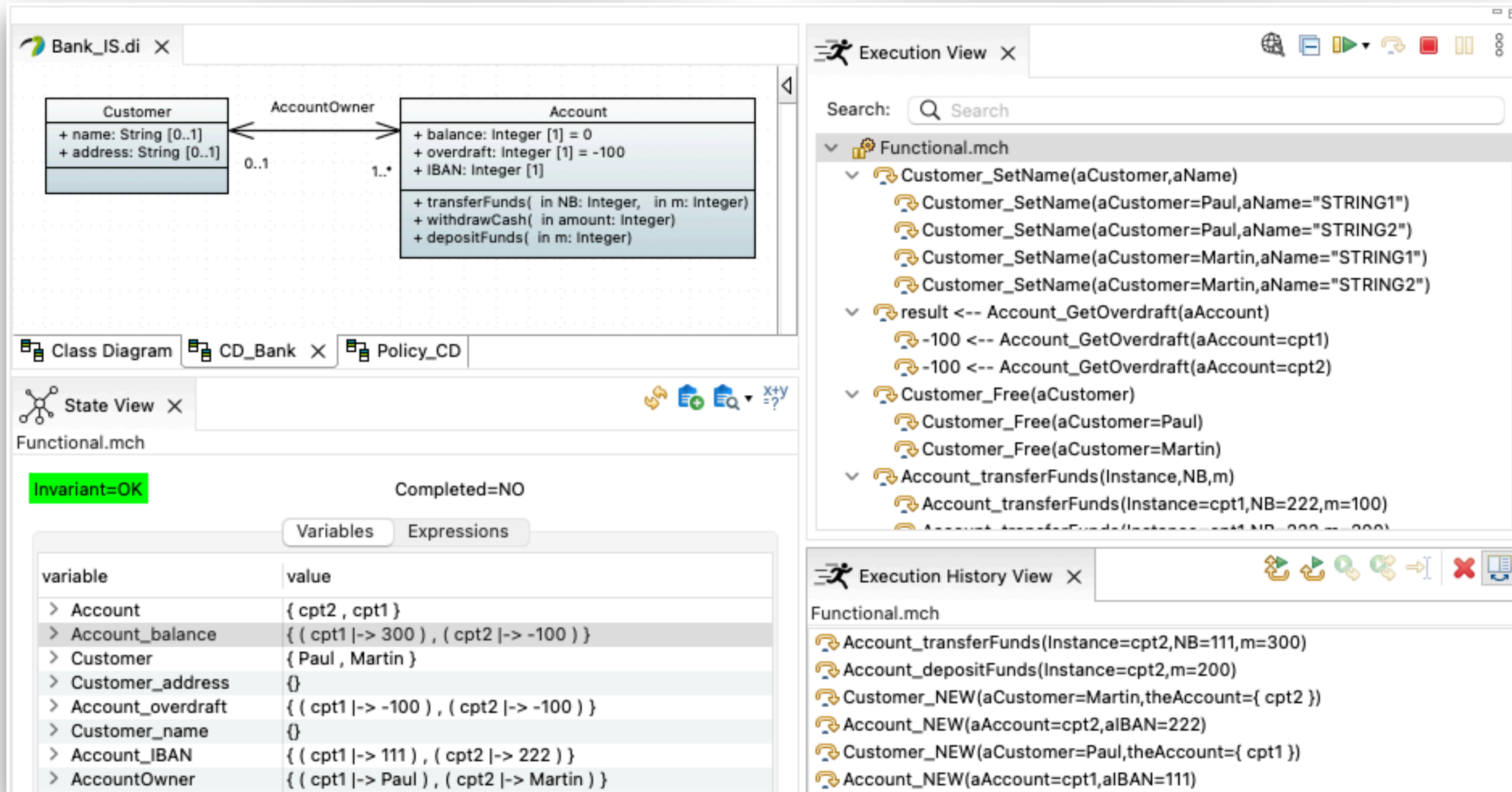
```
MACHINE
  Class
SETS
  OBJECTS
CONSTANTS
  CLASS
PROPERTIES
  CLASS ⊆ OBJECTS
VARIABLES
  class, attribut
INVARIANT
  class ⊆ CLASS ∧
  attrib ∈ class ↔ TypeAttribut
```

(Meyer, 2001)

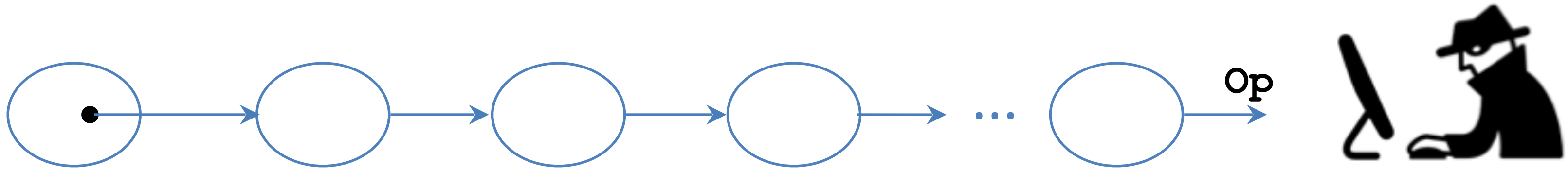
Modeling/Testing (2/3)

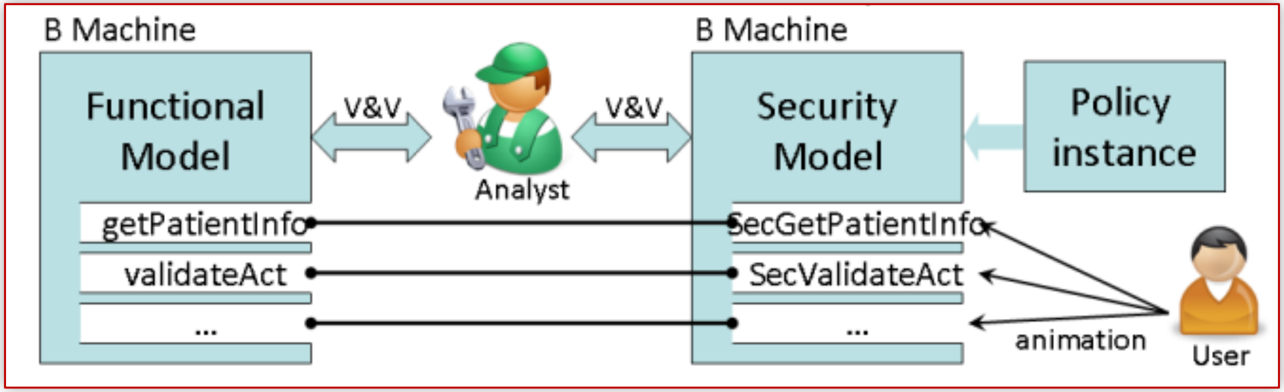
	R. Laleau A. Mammar	E. Meyer H. Ledang	C. Snook M. Butler
Classes	X	X	X
Classes (fixed instances)	-	-	X
Attributes	X	X	X
(Single/Multi)-valued attributes	X	-	-
Inheritance	X	X	X
Multiplicities	X	X	X
Navigation	-	X	X
Roles	X	-	X
Association constraints	X	X	-
Fixed/Non-fixed associations	X	-	-
Association classes	X	X	-
Association classes with inheritance or other relationships	X	-	
Parameterized classes	-	-	X

Modeling/Testing (3/3)



Attacks


...
Op


B Machine
Functional Model
getPatientInfo
validateAct
...
V&V
Analyst
V&V
B Machine
Security Model
SecGetPatientInfo
SecValidateAct
...
Policy instance
animation
User

Insider attacks: forward search (1/3)

- Guided model checking (CSP||B)

Is there a sequence of operations that can be executed by a manager in order to transfer funds from a customer's account?

```
MAIN = UI

UI = (Connect?user!{AccountManager} -> setCurrentUser(user) -> MANAGER_FUNC
     [] Connect?user!{CustomerUser} -> setCurrentUser(user) -> CLIENT_FUNC)
     ;  disconnectUser -> UI

MANAGER_FUNC =
        CREATE_ACCOUNT [] CREATE_CUSTOMER [] UPDATE_CUSTOMER [] SKIP

CREATE_ACCOUNT =
        secure_Account_NEW -> (CREATE_ACCOUNT [] MANAGER_FUNC)

CREATE_CUSTOMER =
        secure_Customer_NEW?customer -> secure_Customer__SetName!customer
        -> (ADD_CUSTOMER_ACCOUNT(customer) [] CREATE_CUSTOMER [] MANAGER_FUNC)
```

Insider attacks: forward search (2/3)

- Guided model checking (CSP||B)

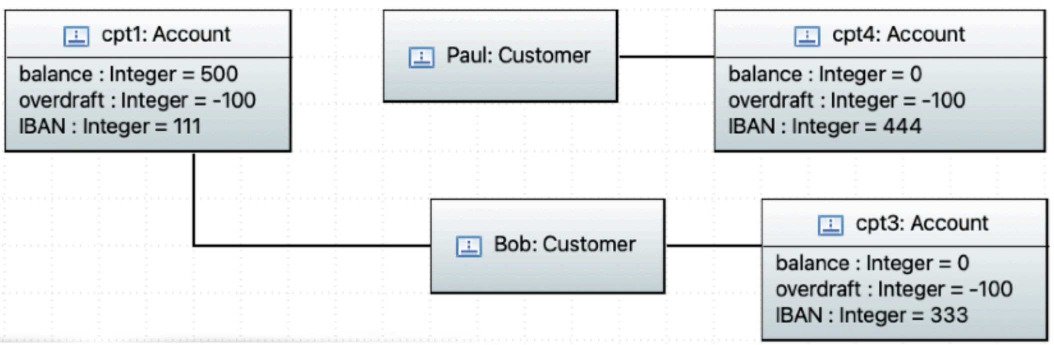
Is there a sequence of operations that can be executed by a manager in order to transfer funds from a customer's account?


```
MAIN = UI [|{| Connect, secure_Account_transferFunds |}|] ATTACK

ATTACK = ATTACKER ||| secure_Account_transferFunds!cpt1 -> goal -> SKIP

ATTACKER = []role:Set(ROLES) @ Connect!Bob!role -> ATTACKER
```

Insider attacks: forward search (3/3)

- Guided model checking (CSP||B)

Is there a sequence of operations that can be executed by a manager in order to transfer funds from a customer's account?

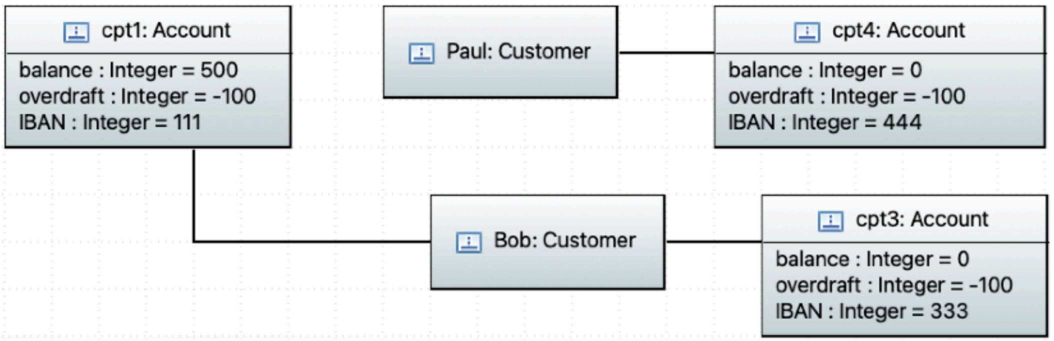


Connect(Bob, {AccountManager}) ;
setCurrentUser(Bob) ;
secure_Account_NEW(cpt_3, 333) ;
secure_Customer_NEW(Bob,{cpt_3}) ;
secure_Customer_SetName(Bob, "...") ;

Insider attacks: forward search (3/3)

- Guided model checking (CSP||B)

Is there a sequence of operations that can be executed by a manager in order to transfer funds from a customer's account?



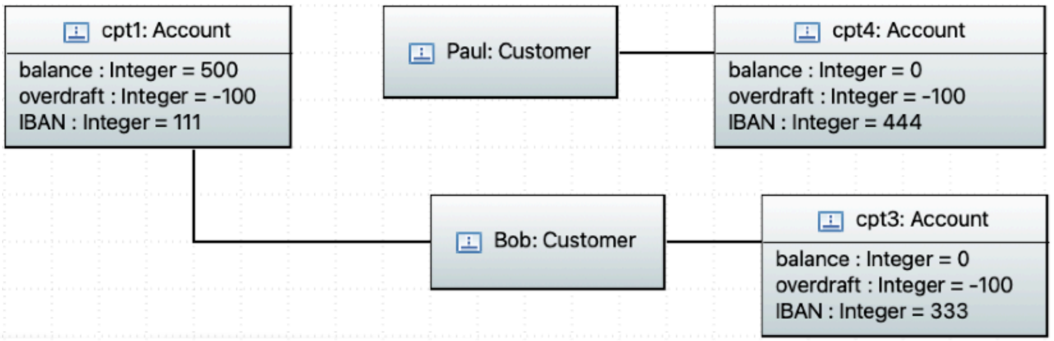
Connect(Bob, {AccountManager}) ;
setCurrentUser(Bob) ;
secure_Account_NEW(cpt_3, 333) ;
secure_Customer_N
secure_Customer_S

secure_Account_NEW(cpt_4, 444) ;
secure_Customer_AddAccount(Paul,{cpt_4}) ;
secure_Customer_RemoveAccount(Paul,{cpt_1}) ;
secure_Customer_AddAccount(Bob,{cpt_1}) ;

Insider attacks: forward search (3/3)

- Guided model checking (CSP | | B)

Is there a sequence of operations that can be executed by a manager in order to transfer funds from a customer's account?



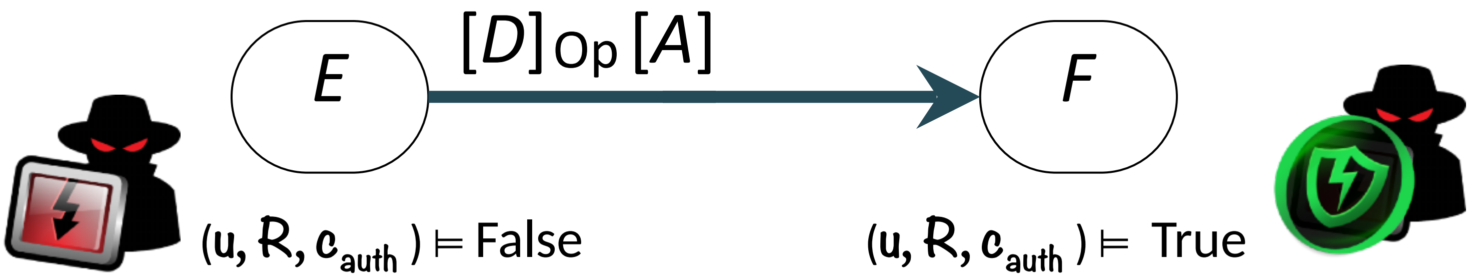
Connect(Bob, {AccountManager}) ;
setCurrentUser(Bob) ;
secure_Account_NEW(cpt_3, 333) ;
secure_Customer_NEW(Bob,{cpt_3}) ;
secure_Customer_SetName(Bob, "...") ;

secure_Account_NEW(cpt_4, 444) ;
secure_Customer_AddAccount(Paul,{cpt_4}) ;
secure_Customer_Rem...
secure_Customer_Add...

disConnect(Bob) ;
Connect(Bob, {CustomerUser}) ;
secure_Account_transferFunds(cpt_1, 333, 500) ;

Insider attacks: backward search

States = Predicates



(1) always enabled: $\forall x.E \Rightarrow Pre(op)$

(2) never enabled: $\forall x.E \Rightarrow \neg Pre(op)$

(3) possibly enabled (¬ (1) ∧ ¬ (2)): $\exists x.E \wedge Pre(op)$

(4) always reached: $\forall x.E \wedge Pre(op) \Rightarrow [Action(op)]F$

(5) never reachable: $\forall x.E \wedge Pre(op) \Rightarrow [Action(op)]\neg F$

(6) possibly reached (¬ (4) ∧ ¬ (5)): $\exists x.E \wedge Pre(op) \wedge \neg[Action(op)]\neg F$

Insider attacks: backward search

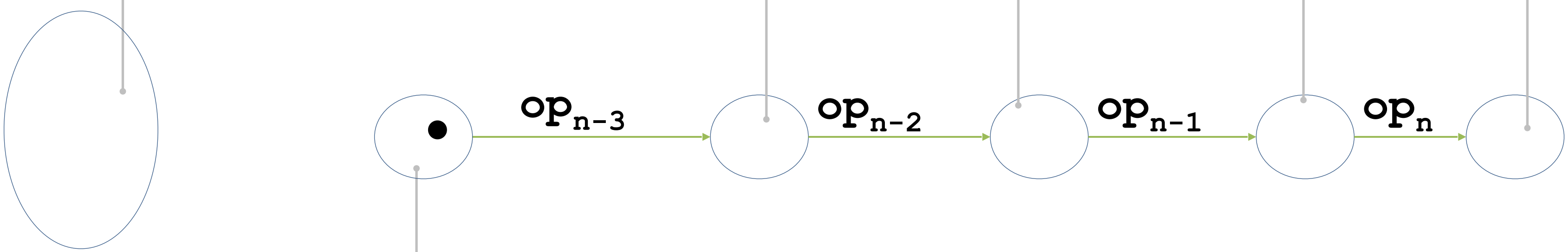
$\neg P_F \wedge \neg Pre(op_n) \wedge Pre(op_{n-1})$

$\neg P_F \wedge \neg Pre(op_n) \wedge \neg Pre(op_{n-1}) \wedge Pre(op_{n-2})$

$\neg P_F \wedge \neg Pre(op_n) \wedge \ldots \wedge \neg Pre(op_{n-3})$

$\neg P_F \wedge Pre(op_n)$

P_F



$\neg P_F \wedge \neg Pre(op_n) \wedge \ldots \wedge Pre(op_{n-3})$

Insider attacks: backward search

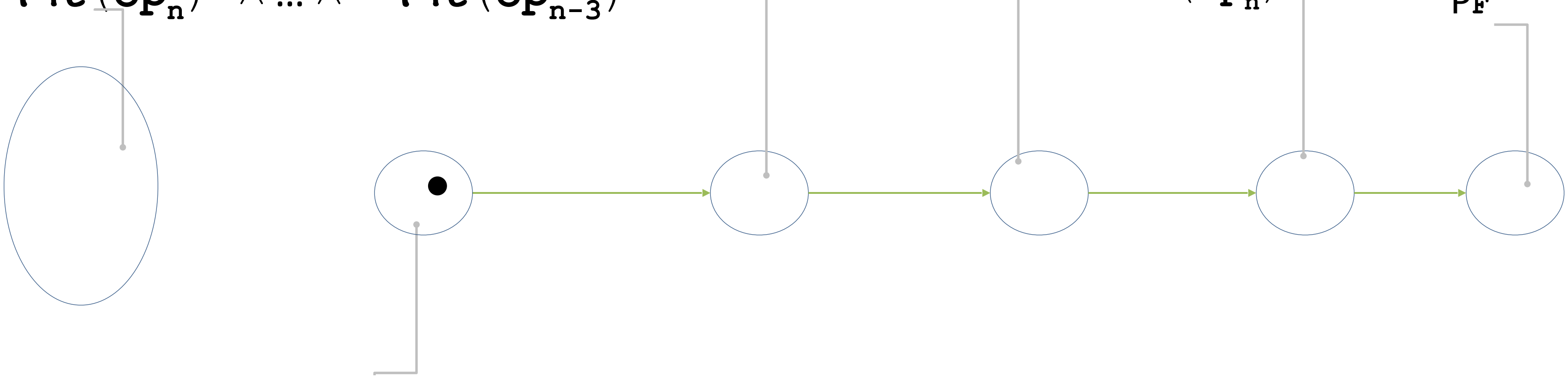
$\neg P_F \wedge \neg Pre(op_n) \wedge Pre(op_{n-1})$

$\neg P_F \wedge \neg Pre(op_n) \wedge \neg Pre(op_{n-1}) \wedge Pre(op_{n-2})$

$\neg P_F \wedge \neg Pre(op_n) \wedge \ldots \wedge \neg Pre(op_{n-3})$

$\neg P_F \wedge Pre(op_n)$

P_F



$\neg P_F \wedge \neg Pre(op_n) \wedge \ldots \wedge Pre(op_{n-3})$

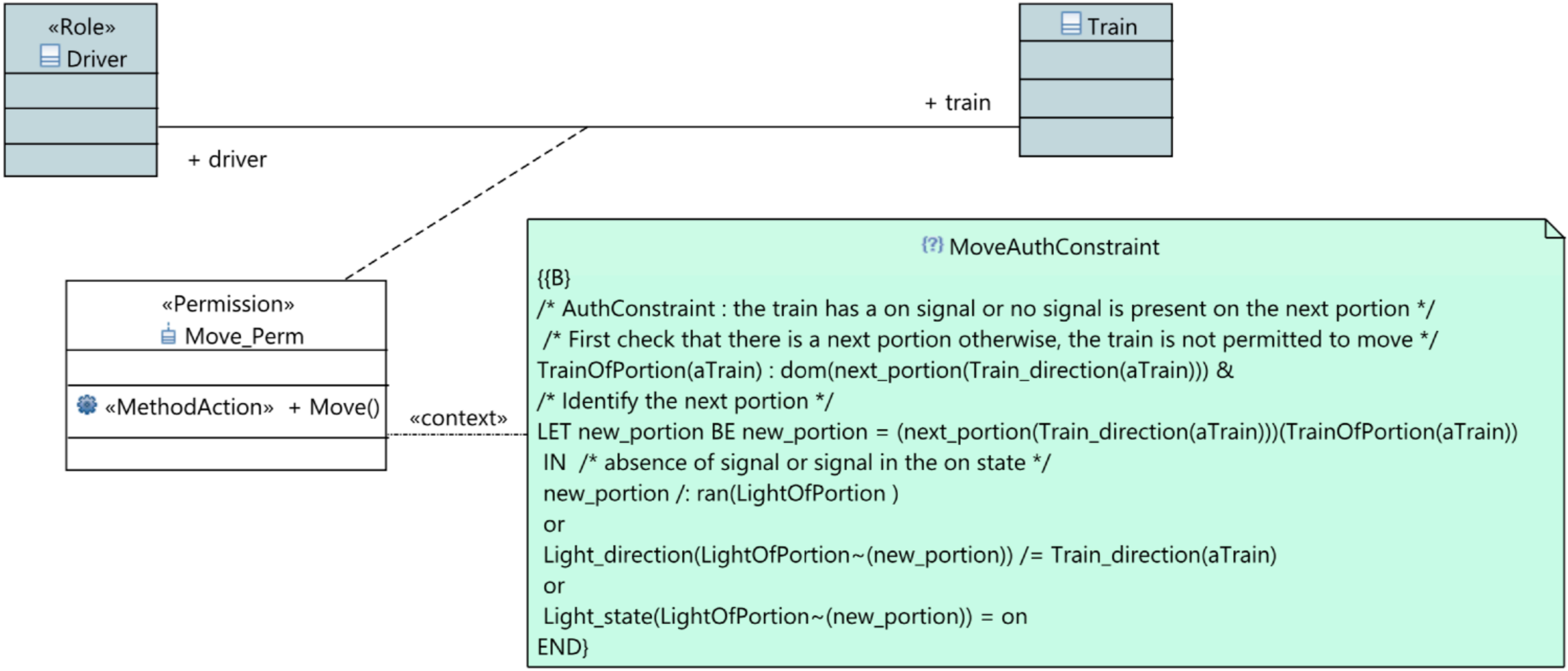
$$Q_{symb} = < \mathtt{init}, op_{n-3}, op_{n-2}, op_{n-1}, op_n >$$

Outline

- Introduction
- **Model Driven Security**
  - UML / SecureUML
  - Insider attacks
- **Executable DSLs**
  - Motivations and results
  - Applications
- Conclusion

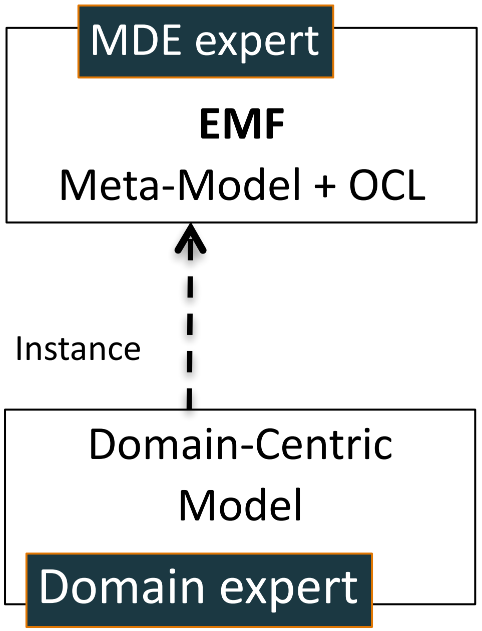
UML vs DSLs

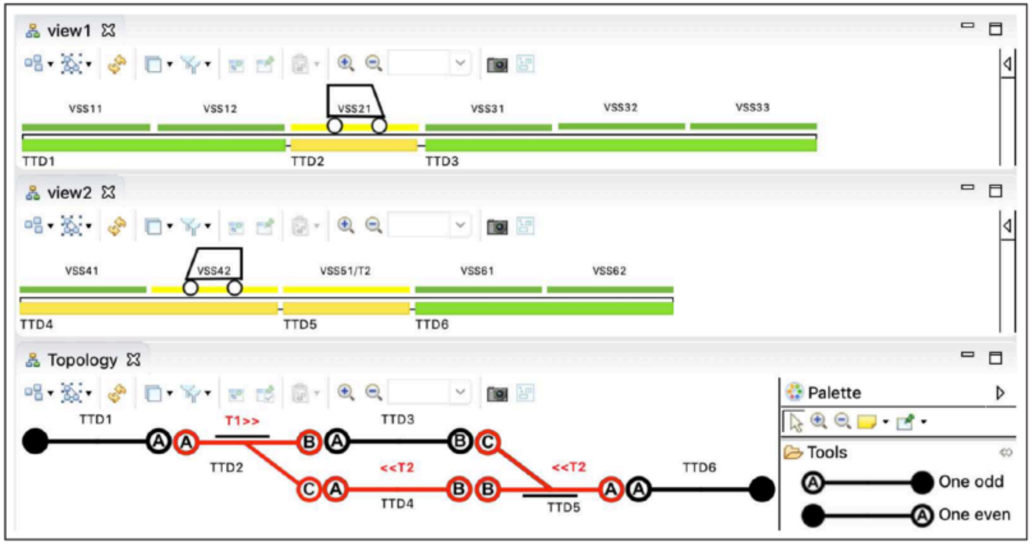
- NextRegio (IRT Railenium, SNCF réseau)
- Undesirable scenarios + responsibilities



UML vs DSLs

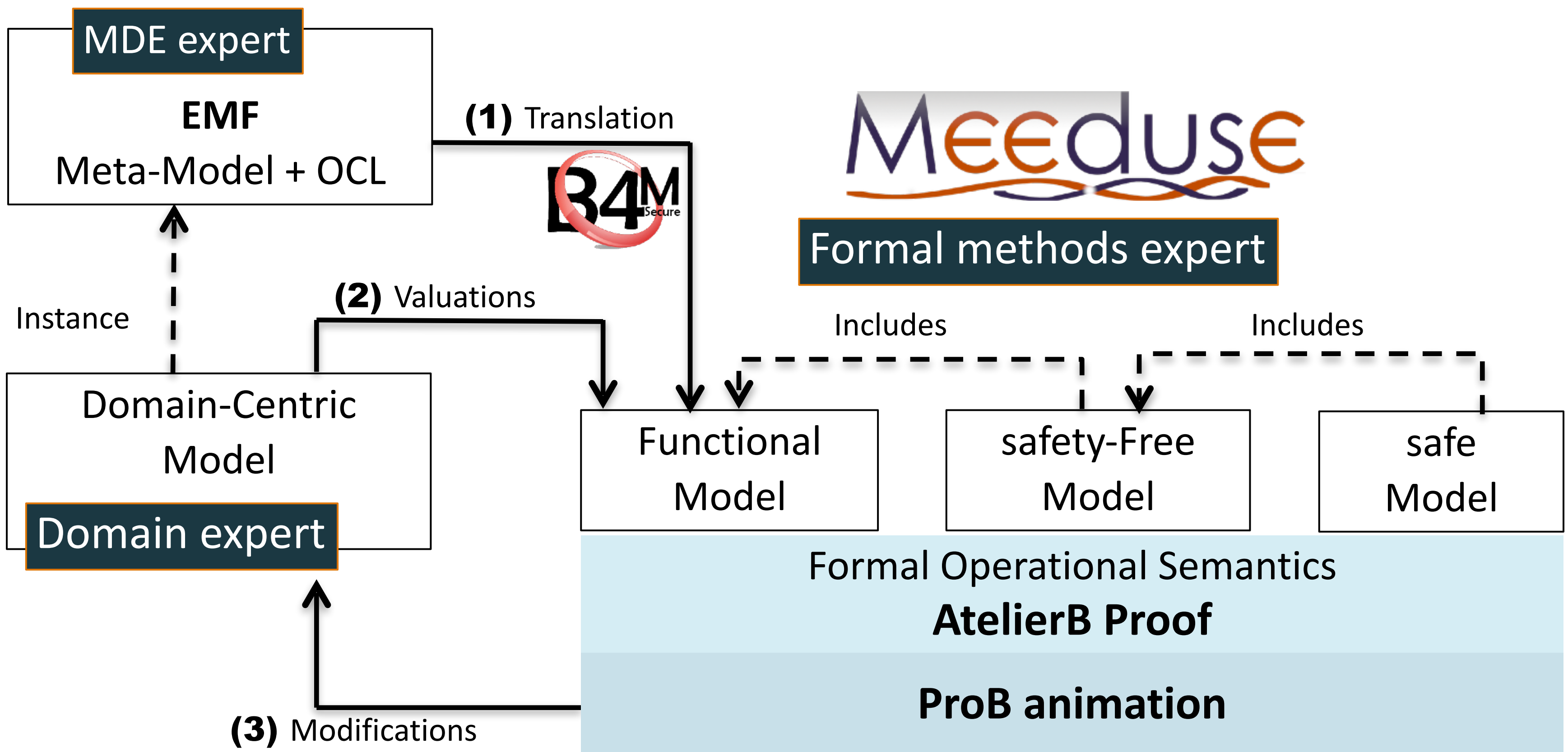
- NextRegio (IRT Railenium, SNCF réseau)
- Undesirable scenarios + responsibilities





UML vs DSLs

- NextRegio (IRT Railenium, SNCF réseau)
- Undesirable scenarios + responsibilities



Positioning

- DSL → FM
  - Survey 1 (2006/2012):
    - T. Kosar, S. Bohra, and M. Mernik, "Domain-specific languages: A systematic mapping study," Information and Software Technology

> *"there is an urgent need in DSL research for identifying the reasons for lack of using formal methods within domain analysis and possible solutions for improvement"*

  - Survey 2 (2012/2019):
    - A. Iung, J. Carbonell, L. Marchezan, E. M. Rodrigues, M. Bernardino, F. P. Basso, and B. Medeiros, "Systematic mapping study on domain-specific language development tools," *Empirical Software Engineering*.

> *Refers to testing as "the" verification feature of Language Work-Benches*

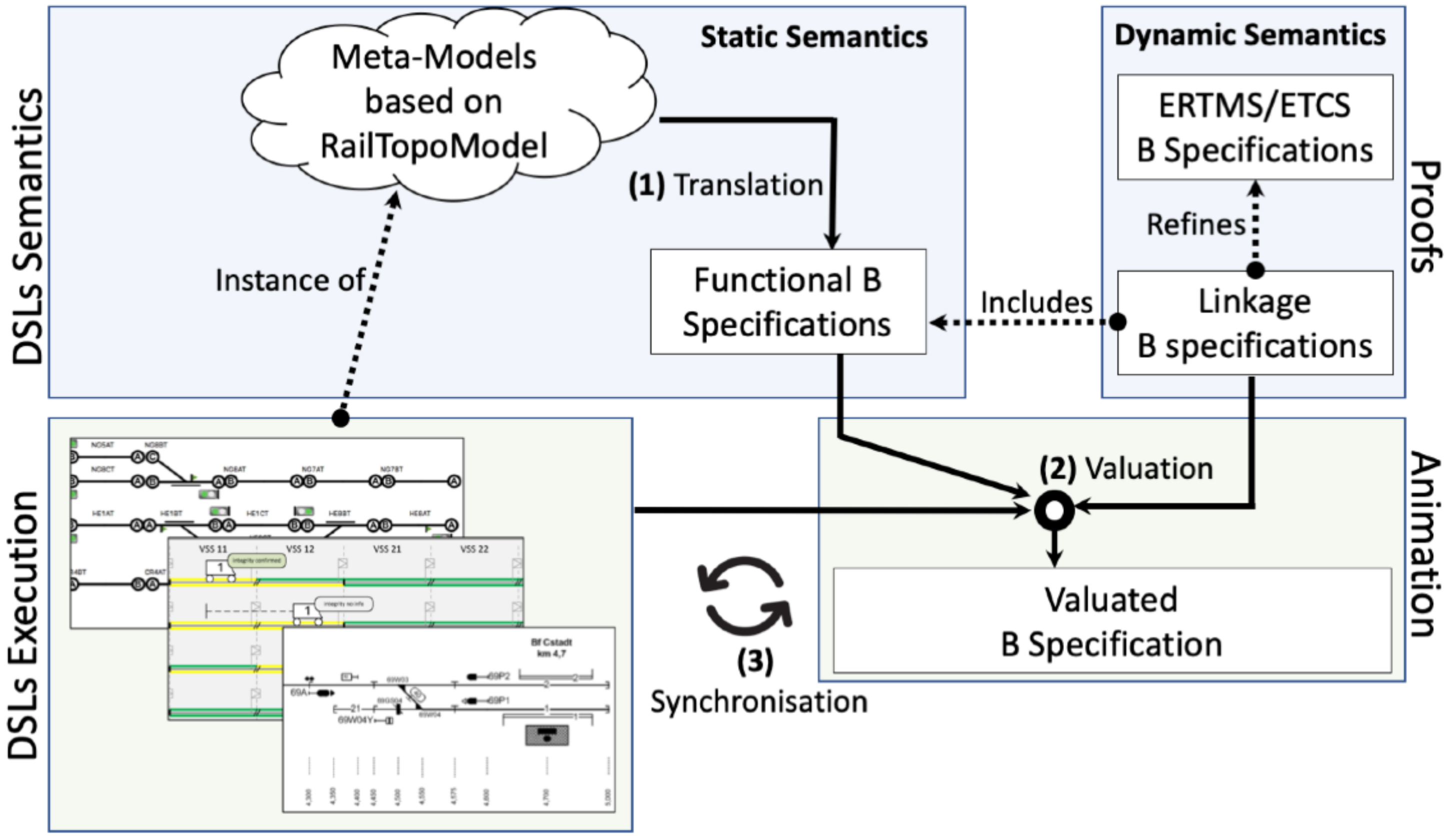
Akram IDANI

Personal opinion

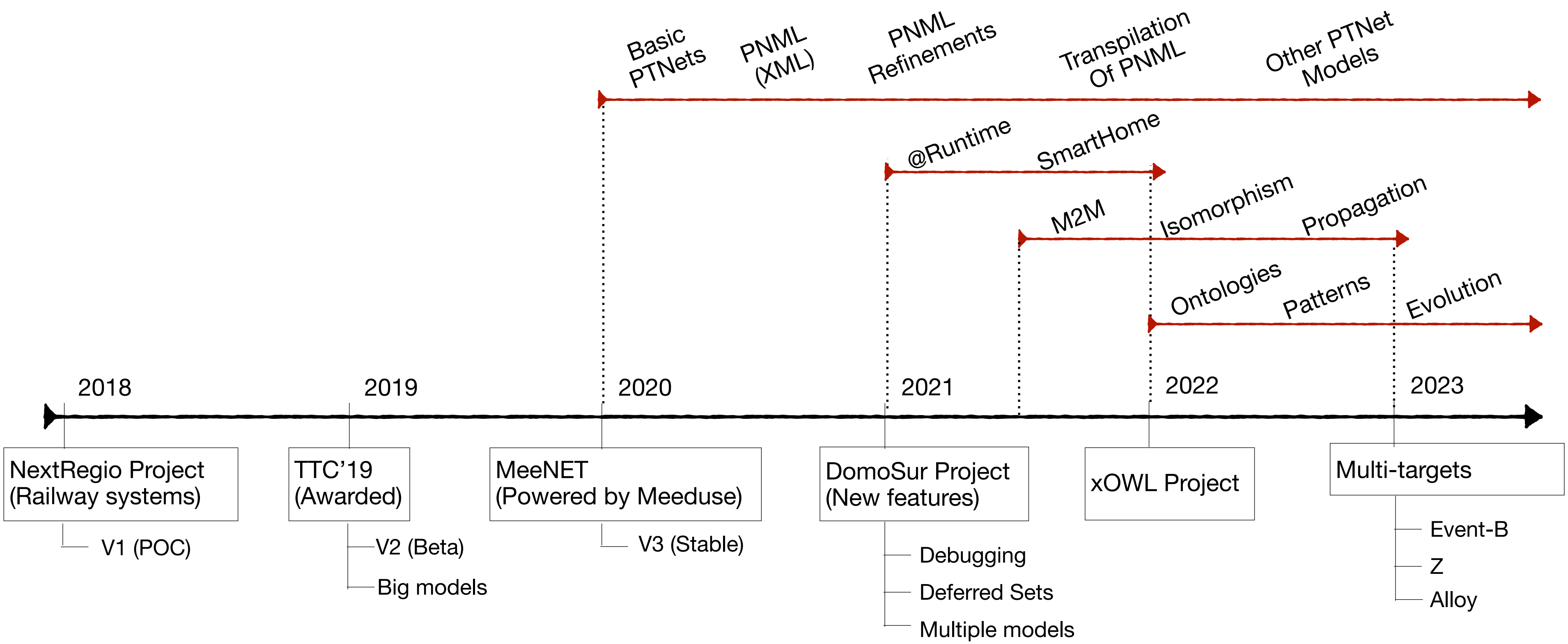
- DSL → FM
  - B. R. Bryant, J. Gray, M. Mernik, P. J. Clarke, R. B.France, and G. Karsai, "Challenges and directions in formalizing the semantics of modeling languages," Comput. Sci. Inf. Syst (2011).
  - Tanslational approaches:
    - **Advantages:**
      1. provide a well-defined and understood semantics, and
      2. the DSL can convey existing tools of the language into which it is translated.
    - **Limitations:**
      3. it is very challenging to correctly map the constructs of the DSL into the constructs of the target language, and
      4. the mapping of the verification results back into the DSL appears as a major issue of existing approaches.

Applications

- Railway systems - PhD A. Yar (33%)



Applications



Outline

- Introduction
- **Model Driven Security**
  - UML / SecureUML
  - Insider attacks
- **Executable DSLs**
  - Motivations and results
  - Applications
- Conclusion & Perspectives

Conclusion & perspectives

- FM & MDE
  - MDS: Healthcare IS (Security)
  - xDSLs: Railway sytsems (Safety)

- B4MSecure / Meeduse

- FM and MDS: is (still) an active topic
  - Conformance PIM/PSM/Application
  - Monitor/Controller synthesis
  - Align (Data, Security, business)-Models

- FM and xDSLs: not a new topic but requires further works
  - Material already exists (iFM, MoDeVVa, UML&FM, VOLT, etc)
  - Investigate other languages (Maude, ASM, Z, Alloy, etc)
  - Provide a unifying framework

Facts

- B4MSecure: Teaching (master degree, MoSIG 2)
  - Topic: Software/Hardware: quality engineering, models of computation
  - Lecture: Information Security
- Meeduse: Recent papers
  - A. Idani., The B Method meets MDE.
    Research Challenges in Information Science (RCIS-2022)
  - A. Idani., Formal Model Driven Executable DSLs: Application to Petri-nets.
    NASA Journal on Innovations in Systems and Soft. Engineering (ISSE-2022).
  - A. Idani et al., Alliance of model-driven engineering with a proof-based formal approach.
    NASA Journal on Innovations in Systems and Soft. Engineering (ISSE-2020).
  - A. Idani., Meeduse: A Tool to Build and Run Proved DSLs.
    Integrated Formal Methods (iFM-2020).
  - A. Idani et al., Incremental Development of a Safety Critical System Combining formal Methods and DSMLs - Application to a Railway System.
    Formal Methods for Industrial Critical Systems (FMICS-2019).
  - A. Idani et al., Towards a Tool-Based Domain Specific Approach for Railway Systems Modeling and Validation.
    Reliability, Safety and Security of Railway Systems (RSSRail-2019).


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
BİYAN
SHUKRIA
TINGKI
GRAZIE
MEHRBANI
MAAKE
KOMAPSUMNIDA
PALDIES
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
BOLZÏN
MERCI