

The construction of Models Needs Idempotent Transformations

A way to restore models....

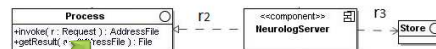
Clémentine NEMO - Mireille BLAY-FORNARINO
 {nemo,blay}@polytech.unice.fr

SAFA 2010
 ☺ Clémentine Nemo

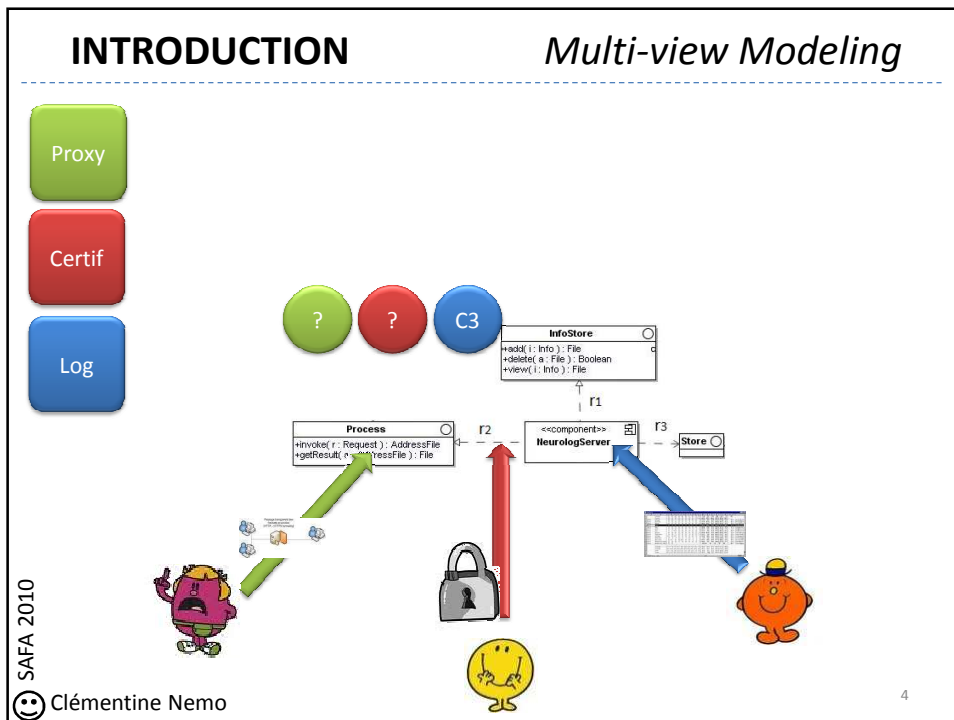
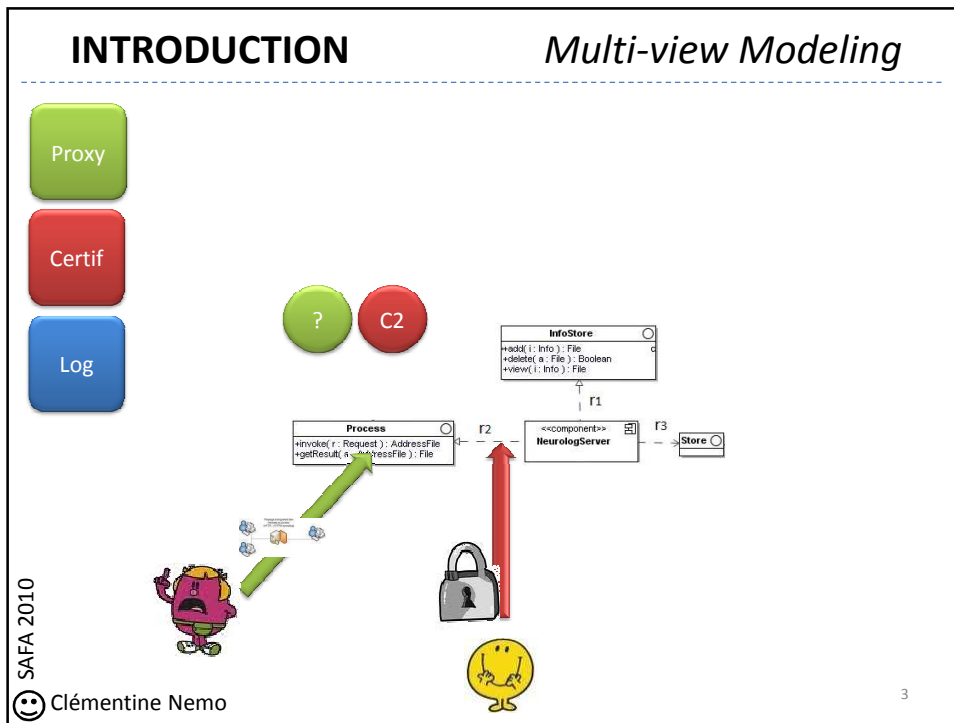


INTRODUCTION

Multi-view Modeling



SAFA 2010
 ☺ Clémentine Nemo



INTRODUCTION Multi-view Modeling

Proxy

Certif

Log

SAFA 2010
Clémentine Nemo 5

INTRODUCTION Multi-view Modeling

Proxy

Certif

Log

SAFA 2010
Clémentine Nemo 6

Outline

- **Motivation**
 - Need Idempotent for Property
 - Idempotent Existing Strategies
- **Idempotent Transformation by Construction (ITC)**
 - Formalisation
 - ITC Expression
 - ITC Contextualisation
 - ITC Application
- **Conclusion**

SAFA 2010

Clémentine Nemo

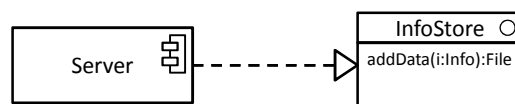
7

Motivation *Need Idempotent for Property*

Guideline Example :

From the initial model, constructing a model by :

- 1) introduction of one policy,
- 2) user action.



Initial model

SAFA 2010

Clémentine Nemo

8

Motivation Need Idempotent for Property

Guideline Example

```

classDiagram
    class CertificationAuthority {
        <><>
    }
    class Check {
        <><>
        checkCertificate(c:Certificate):Boolean
    }
    class Server {
        <><>
    }
    class InfoStore {
        <><>
        addData(i:Info, c:Certificate):File
    }
    CertificationAuthority ..> Check
    Server ..> Check
    Server ..> InfoStore
    
```

Certif

All operations of all provided interfaces of the secured component have a Certificate parameter as input. A CertificationAuthority component provides a Check interface. The secured component requires the Check interface.

SAFA 2010
Clémentine Nemo 9

Motivation Need Idempotent for Property

Guideline Example

```

classDiagram
    class CertificationAuthority {
        <><>
    }
    class Check {
        <><>
        checkCertificate(c:Certificate):Boolean
    }
    class Server {
        <><>
    }
    class InfoStore {
        <><>
        addData(i:Info, c:Certificate):File
        deleteData(f:File):Boolean
    }
    CertificationAuthority ..> Check
    Server ..> Check
    Server ..> InfoStore
    
```

Certif **User**

All operations of all provided interfaces of the secured component have a Certificate parameter as input. A CertificationAuthority component provides a Check interface. The secured component requires the Check interface.

SAFA 2010
Clémentine Nemo 10

Motivation Need Idempotent for Property

Guideline Example

Certif

User

Certif

All operations of all provided interfaces of the secured component have a Certificate parameter as input. A CertificationAuthority component provides a Check interface. The secured component requires the Check interface.

```

classDiagram
    class CertificationAuthority {
        <<interface>>
    }
    class Check {
        <<interface>>
        checkCertificate(c:Certificate):Boolean
    }
    class Server {
    }
    class InfoStore {
        <<interface>>
        addData(i:Info,c:Certificate):File
        deleteData(f:File,c:Certificate):Boolean
    }
    CertificationAuthority --|> Check
    Server ..|> InfoStore
    
```

Motivation Need Idempotent for Property

Guideline Example

Certif

User

Certif

All operations of all provided interfaces of the secured component have a Certificate parameter as input. A CertificationAuthority component provides a Check interface. The secured component requires the Check interface.

```

classDiagram
    class CertificationAuthority {
        <<interface>>
    }
    class Check {
        <<interface>>
        checkCertificate(c:Certificate):Boolean
    }
    class Server {
    }
    class InfoStore {
        <<interface>>
        addData(i:Info,c:Certificate):File
        deleteData(f:File,c:Certificate):Boolean
    }
    CertificationAuthority --|> Check
    Server ..|> InfoStore
    
```


Motivation *Need Idempotent for Property*

Existing Idempotent Strategies

- # Specifying language mode :
 - # *Check-Before-Enforce* [QVT]
 - # Number of applications [LAM]

- # Factorizing same actions [SAN]

- # Avoiding duplicated elements by negative rules [MEN]

SAFA 2010
 Clémentine Nemo 13

Motivation *Need Idempotent for Property*

Existing Idempotent Strategies


- # Specifying language mode :
 - # *Check-Before-Enforce* [QVT]

- # Fa

Proposition

Transformation with
automatic idempotent application

- # Avoiding duplicated elements by negative rules [MEN]

SAFA 2010
 Clémentine Nemo 14

Outline

- **Motivation**
 - Need Idempotent for Property
 - Idempotent Existing Strategies
- **Idempotent Transformation by Construction (ITC)**
 - Formalisation
 - ITC Expression
 - ITC Contextualisation
 - ITC Application
- **Conclusion**

SAFA 2010

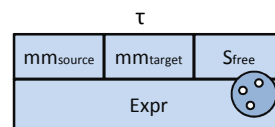
Clémentine Nemo

15

Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (mm_{source}, mm_{target}, Expr, S_{free})$



SAFA 2010

Clémentine Nemo

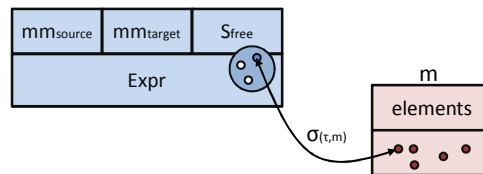
16

Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (mm_{source}, mm_{target}, Expr, S_{free})$

Binding $\sigma_{(\tau, m)} = \{(V \rightarrow e) \mid V \in S_{free}, e \in m\}$



SAFA 2010

Clémentine Nemo

17

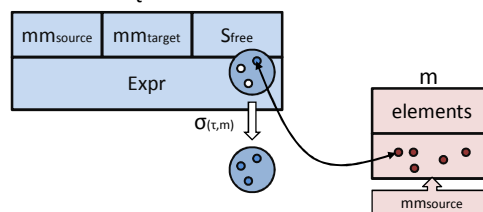
Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (mm_{source}, mm_{target}, Expr, S_{free})$

Binding $\sigma_{(\tau, m)} = \{(V \rightarrow e) \mid V \in S_{free}, e \in m\}$

Application



SAFA 2010

Clémentine Nemo

18

Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (mm_{source}, mm_{target}, Expr, S_{free})$

Binding $\sigma_{(\tau, m)} = \{(V \rightarrow e) \mid V \in S_{free}, e \in m\}$

Application $\tau \sigma(m) = m'$

SAFA 2010

Clémentine Nemo

19

Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (mm_{source}, mm_{target}, Expr, S_{free})$

Binding $\sigma_{(\tau, m)} = \{(V \rightarrow e) \mid V \in S_{free}, e \in m\}$

Application $\tau \sigma(m) = m'$

... with idempotent property

SAFA 2010

Clémentine Nemo

20

Idempotent Transformation by Construction

Expr = (Selection, Identification, Modification)

Selection=[elmtaryAction_{Selection}1...]



ITC_{certif}

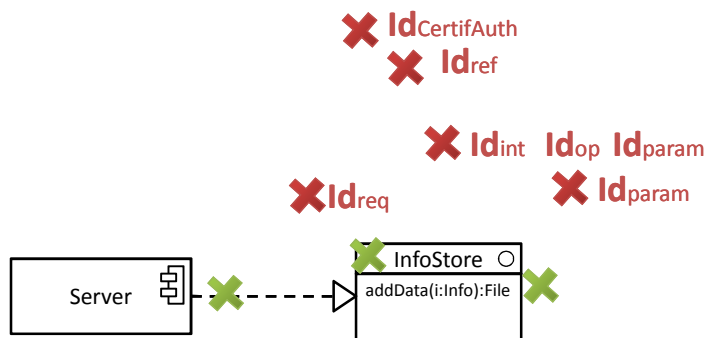
All operations of all provided interfaces of the secured component have a Certificate parameter as input.
 A CertificateAuthority component provides a Check interface.
 The secured component requires the Check interface.

SAFA 2010
 Clémentine Nemo

Idempotent Transformation by Construction

Expr = (Selection, Identification, Modification)

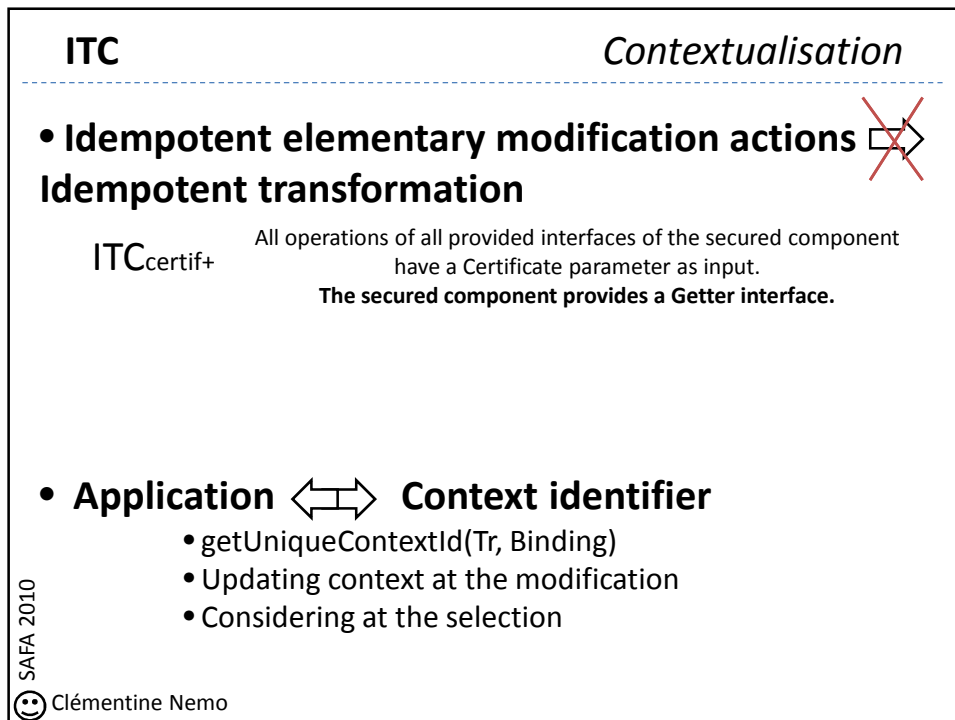
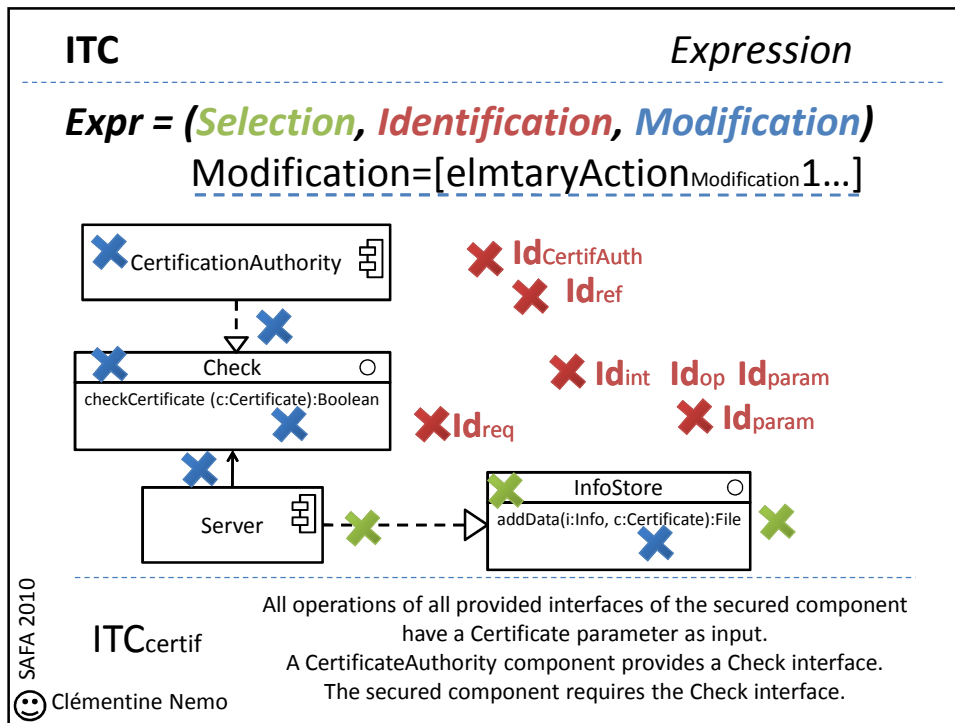
Identification=[elmtaryAction_{Identification}1...]

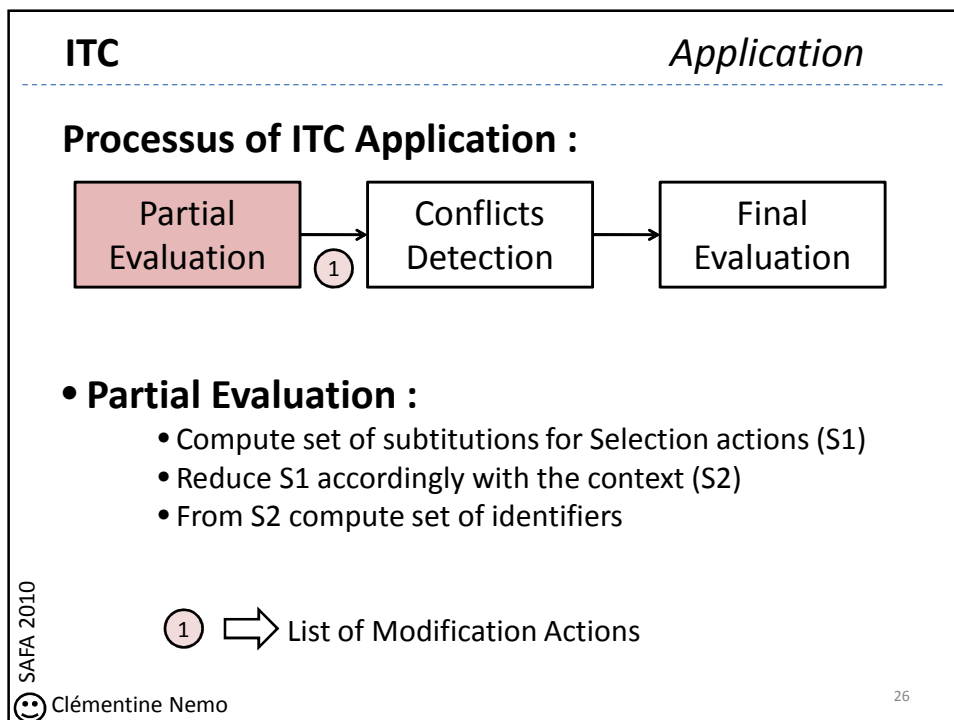
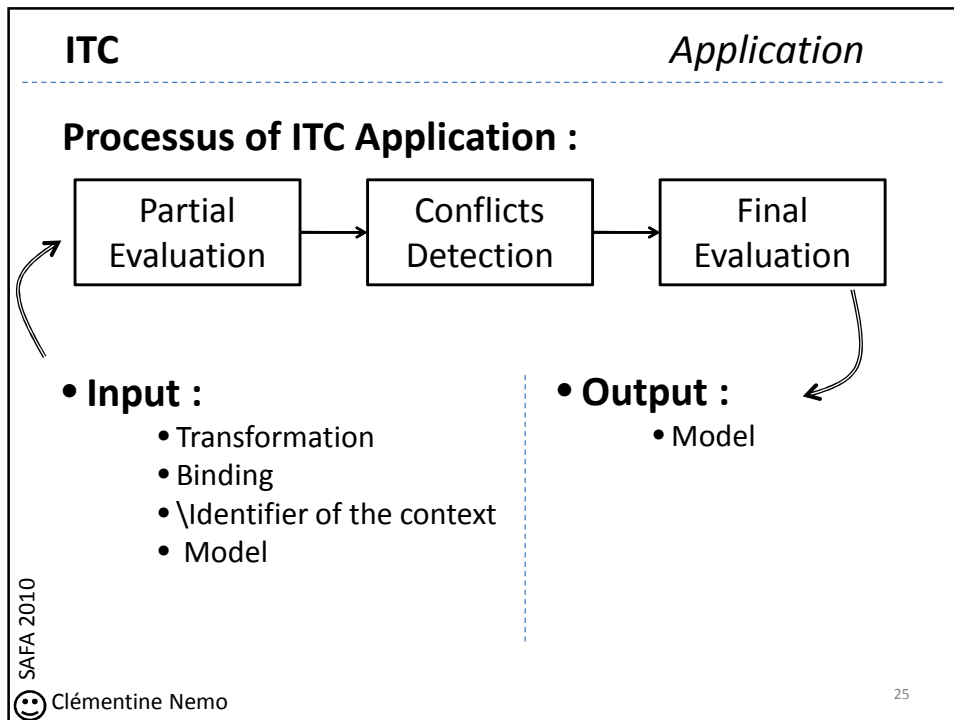


ITC_{certif}

All operations of all provided interfaces of the secured component have a Certificate parameter as input.
 A CertificateAuthority component provides a Check interface.
 The secured component requires the Check interface.

SAFA 2010
 Clémentine Nemo





ITC
Application

Processus of ITC Application :

Partial
Evaluation

① →

Conflicts
Detection

② →

Final
Evaluation

List of Modification Actions
Valid List of Modification Actions

• Conflict Detection :

Create(elmt1) Create(elmt2) Delete(elmt3)	Create(elmt3) Create(elmt2) SetValue(elmt2,[elmt3])
---	---

SAFA 2010
Clémentine Nemo
27

ITC
Application

Processus of ITC Application :

Partial
Evaluation

① →

Conflicts
Detection

② →

Final
Evaluation

List of Modification Actions
Valid List of Modification Actions

• Final Evaluation

- Element modification : creation, removal, value setting, ...
- Context updating

SAFA 2010
Clémentine Nemo
28

Conclusion

- **Guaranteeing that a model conforms to an ITC for each application**

- Idempotent Transformation by Construction
- Processus of Application

- **Prolog Implementation**

- Atomic actions
 - Applications engine
-

- **Perspectives**

- Guaranteeing that a model conforms to a set of transformations
- Automatic re-applications process

Conclusion

Thank you



- [QVT] OMG, MOF QVT Final Adopted Specification, OMG Document ptc/2005-11-01, Object Modeling Group, Jun. 2005.
- [LAM] A. Lajmi, S. Cauvin, M. Ziane, and T. Ziadi, "A Multi-View Model-Driven Approach for Packaging Software Components," in 25th Annual Symposium on Applied Computing(SAC 2010). ACM, Mar. 2010.
- [MEN] T. Mens, G. Kiesel, and O. Runge, "Transformation dependency analysis- A comparison of two approaches," in Langages et Mod`eles `a Objets(LMO2006), Mar. 2006.
- [RAT] I. Rath, G. Bergmann, A. Okros, and D. Varro, "Incremental Pattern Matching in the VIATRA Model Transformation System," in International Conference on ModelTransformation(ICMT'08). Springer, Jul.2008.
- [SAN] J. Sanchez and J. Garcia, Approaches for Model Transformation Reuse:Factorization and Composition. Springer, 2008.



