

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

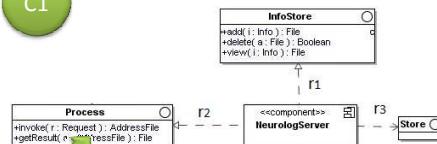
1

INTRODUCTION

Multi-view Modeling



C1



SAFA 2010

(;) Clémentine Nemo

2

INTRODUCTION

Multi-view Modeling

The diagram illustrates a multi-view modeling approach with three views:

- Proxy View:** Represented by a green rounded rectangle labeled "Proxy".
- Certif View:** Represented by a red rounded rectangle labeled "Certif".
- Log View:** Represented by a blue rounded rectangle labeled "Log".

Below the views, there is a UML-like diagram showing interactions between a client, a server, and a store. The client interacts with the server via a process. The server interacts with an InfoStore and a Store. The Log view is shown as a blue rounded rectangle at the bottom right.

SAFA 2010
Clémentine Nemo

3

INTRODUCTION

Multi-view Modeling

The diagram illustrates a multi-view modeling approach with four views:

- Proxy View:** Represented by a green rounded rectangle labeled "Proxy".
- Certif View:** Represented by a red rounded rectangle labeled "Certif".
- Log View:** Represented by a blue rounded rectangle labeled "Log".
- C3 View:** Represented by a blue rounded rectangle labeled "C3".

Below the views, there is a UML-like diagram showing interactions between a client, a server, and a store. The client interacts with the server via a process. The server interacts with an InfoStore and a Store. The Log view is shown as a blue rounded rectangle at the bottom right. A new view, C3, is shown as a blue rounded rectangle at the top right.

SAFA 2010
Clémentine Nemo

4

INTRODUCTION *Multi-view Modeling*

The diagram illustrates a multi-view modeling approach. On the left, three colored boxes represent views: a green box for 'Proxy', a red box for 'Certif', and a blue box for 'Log'. Above them, three circles labeled C1 (green), C2 (red), and C3 (blue) represent components. A large upward-pointing arrow indicates the flow from the views and components to a central 'NeurologServer' component. Below the server is a database icon. The server interacts with three external entities: a purple devil-like character (labeled 'SAFA 2010'), a yellow smiley face, and an orange character. A process diagram at the bottom shows interactions between the server and these entities.

```

    graph TD
        subgraph Top [ ]
            direction TB
            P[Proxy] --- C1
            C[Certif] --- C2
            L[Log] --- C3
        end
        subgraph Bottom [ ]
            direction TB
            NS[NeurologServer]
            subgraph P1 [Process]
                direction TB
                P1_1[+invoke<sub>r</sub>(<sub>Request</sub> : AddressFile<br/>+getResult<sub>a</sub> : AddressFile<sub>b</sub> : File)]
                P1_2[+add<sub>r</sub>(<sub>Info</sub> : Info<br/>+delete<sub>r</sub>(<sub>Info</sub> : Info<br/>+view<sub>r</sub>(<sub>Info</sub> : Info<br/>+File))
            end
            P1 -- r1 --> NS
            NS -- r2 --> P1
            NS -- r3 --> S[Store]
            subgraph P2 [Process]
                direction TB
                P2_1[+invoke<sub>r</sub>(<sub>Request</sub> : AddressFile<br/>+getResult<sub>a</sub> : AddressFile<sub>b</sub> : File)]
                P2_2[+add<sub>r</sub>(<sub>Info</sub> : Info<br/>+delete<sub>r</sub>(<sub>Info</sub> : Info<br/>+view<sub>r</sub>(<sub>Info</sub> : Info<br/>+File))
            end
            P2 -- r1 --> NS
            NS -- r2 --> P2
            NS -- r3 --> S
            subgraph P3 [Process]
                direction TB
                P3_1[+invoke<sub>r</sub>(<sub>Request</sub> : AddressFile<br/>+getResult<sub>a</sub> : AddressFile<sub>b</sub> : File)]
                P3_2[+add<sub>r</sub>(<sub>Info</sub> : Info<br/>+delete<sub>r</sub>(<sub>Info</sub> : Info<br/>+view<sub>r</sub>(<sub>Info</sub> : Info<br/>+File))
            end
            P3 -- r1 --> NS
            NS -- r2 --> P3
            NS -- r3 --> S
        end
        subgraph Bottom_Icons [ ]
            direction TB
            D Devil
            S Smiley
            O Officer
        end
    
```

SAFA 2010

Clémentine Nemo

5

INTRODUCTION *Multi-view Modeling*

The diagram is similar to the one above, showing three views (Proxy, Certif, Log) and three components (C1, C2, C3). A large upward-pointing arrow indicates the flow to a central 'NeurologServer' component. Below the server is a database icon. The server interacts with three external entities: a purple devil-like character (labeled 'SAFA 2010'), a yellow smiley face, and an orange character. A purple box labeled 'Challenge' contains the text: 'Validation and Restoration of models constructed by introduction of policies.' A process diagram at the bottom shows interactions between the server and these entities.

```

    graph TD
        subgraph Top [ ]
            direction TB
            P[Proxy] --- C1
            C[Certif] --- C2
            L[Log] --- C3
        end
        subgraph Bottom [ ]
            direction TB
            NS[NeurologServer]
            subgraph P1 [Process]
                direction TB
                P1_1[+invoke<sub>r</sub>(<sub>Request</sub> : AddressFile<br/>+getResult<sub>a</sub> : AddressFile<sub>b</sub> : File)]
                P1_2[+add<sub>r</sub>(<sub>Info</sub> : Info<br/>+delete<sub>r</sub>(<sub>Info</sub> : Info<br/>+view<sub>r</sub>(<sub>Info</sub> : Info<br/>+File))
            end
            P1 -- r1 --> NS
            NS -- r2 --> P1
            NS -- r3 --> S[Store]
            subgraph P2 [Process]
                direction TB
                P2_1[+invoke<sub>r</sub>(<sub>Request</sub> : AddressFile<br/>+getResult<sub>a</sub> : AddressFile<sub>b</sub> : File)]
                P2_2[+add<sub>r</sub>(<sub>Info</sub> : Info<br/>+delete<sub>r</sub>(<sub>Info</sub> : Info<br/>+view<sub>r</sub>(<sub>Info</sub> : Info<br/>+File))
            end
            P2 -- r1 --> NS
            NS -- r2 --> P2
            NS -- r3 --> S
            subgraph P3 [Process]
                direction TB
                P3_1[+invoke<sub>r</sub>(<sub>Request</sub> : AddressFile<br/>+getResult<sub>a</sub> : AddressFile<sub>b</sub> : File)]
                P3_2[+add<sub>r</sub>(<sub>Info</sub> : Info<br/>+delete<sub>r</sub>(<sub>Info</sub> : Info<br/>+view<sub>r</sub>(<sub>Info</sub> : Info<br/>+File))
            end
            P3 -- r1 --> NS
            NS -- r2 --> P3
            NS -- r3 --> S
        end
        subgraph Bottom_Icons [ ]
            direction TB
            D Devil
            S Smiley
            O Officer
        end
    
```

SAFA 2010

Clémentine Nemo

Challenge

Validation and Restoration of models
constructed by introduction of policies.

6

Outline

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

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

Motivation

Need Idempotent for Property

Guideline Example

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.

Certif

SAFA 2010

Clémentine Nemo

9

Motivation

Need Idempotent for Property

Guideline Example

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.

X Certif

SAFA 2010

Clémentine Nemo

10

Motivation	Need Idempotent for Property
Guideline Example	<p>Certif User Certif</p> <p>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.</p> <pre> graph TD CA[CertificationAuthority] --> C[Check] C --> S[Server] S -.-> IS[InfoStore] </pre> <p>SAFA 2010</p> <p>Clémentine Nemo</p>

Motivation	Need Idempotent for Property
Guideline Example	<p>Certif User Certif</p> <p>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.</p> <p>SAFA 2010</p> <p>Clémentine Nemo</p>

Motivation	<i>Need Idempotent for Property</i>
<i>Existing Idempotent Strategies</i>	
<ul style="list-style-type: none"> # Specifying language mode : <ul style="list-style-type: none"> # <i>Check-Before-Enforce</i> [QVT] # Number of applications [LAM] # Factorizing same actions [SAN] # Avoiding duplicated elements by negative rules_[MEN] 	

SAFA 2010

(;) Clémentine Nemo 13

Motivation	<i>Need Idempotent for Property</i>
<i>Existing Idempotent Strategies</i>	
<ul style="list-style-type: none"> # Specifying language mode : <ul style="list-style-type: none"> # <i>Check-Before-Enforce</i> [QVT] # Fa <div style="border: 1px solid black; padding: 10px; width: fit-content; margin-left: 20px;"> Proposition Transformation with automatic idempotent application </div> # Avoiding duplicated elements by negative rules_[MEN] 	

SAFA 2010

(;) Clémentine Nemo 14

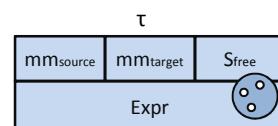
Outline

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

Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (\text{mm}_{\text{source}}, \text{mm}_{\text{target}}, \text{Expr}, \text{Sfree})$

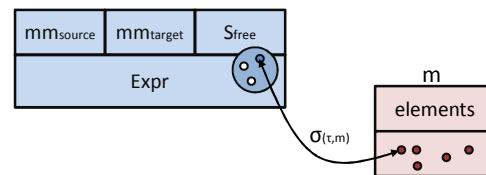


Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (\text{mm}_{\text{source}}, \text{mm}_{\text{target}}, \text{Expr}, S_{\text{free}})$

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



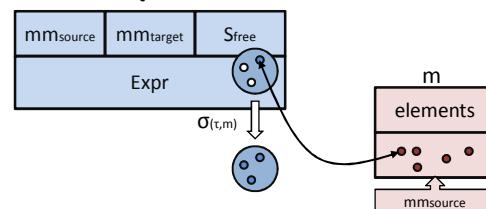
Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (\text{mm}_{\text{source}}, \text{mm}_{\text{target}}, \text{Expr}, S_{\text{free}})$

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

Application



Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (\text{mm}_{\text{source}}, \text{mm}_{\text{target}}, \text{Expr}, S_{\text{free}})$

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

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

SAFA 2010

Clémentine Nemo

19

Idempotent Transformation by Construction

Formalisation for a normal transformation

Transformation $\tau = (\text{mm}_{\text{source}}, \text{mm}_{\text{target}}, \text{Expr}, S_{\text{free}})$

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

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

... with idempotent property

Idempotent Transformation by Construction (ITC)

SAFA 2010

Clémentine Nemo

20

Idempotent Transformation by Construction

Expr = (Selection, Identification, Modification)

Selection=[elmtaryAction_{Selection1...}]



All operations of all provided interfaces of the secured component
have a Certificate parameter as input.

ITCcertif

A CertificateAuthority component provides a Check interface.

(;) Clémentine Nemo

The secured component requires the Check interface.

SAFA 2010

Idempotent Transformation by Construction

Expr = (Selection, Identification, Modification)

Identification=[elmtaryAction_{Identification1...}]

✗ IdCertifAuth
✗ Idref

✗ Idint Idop Idparam
✗ Idreq ✗ Idparam



All operations of all provided interfaces of the secured component
have a Certificate parameter as input.

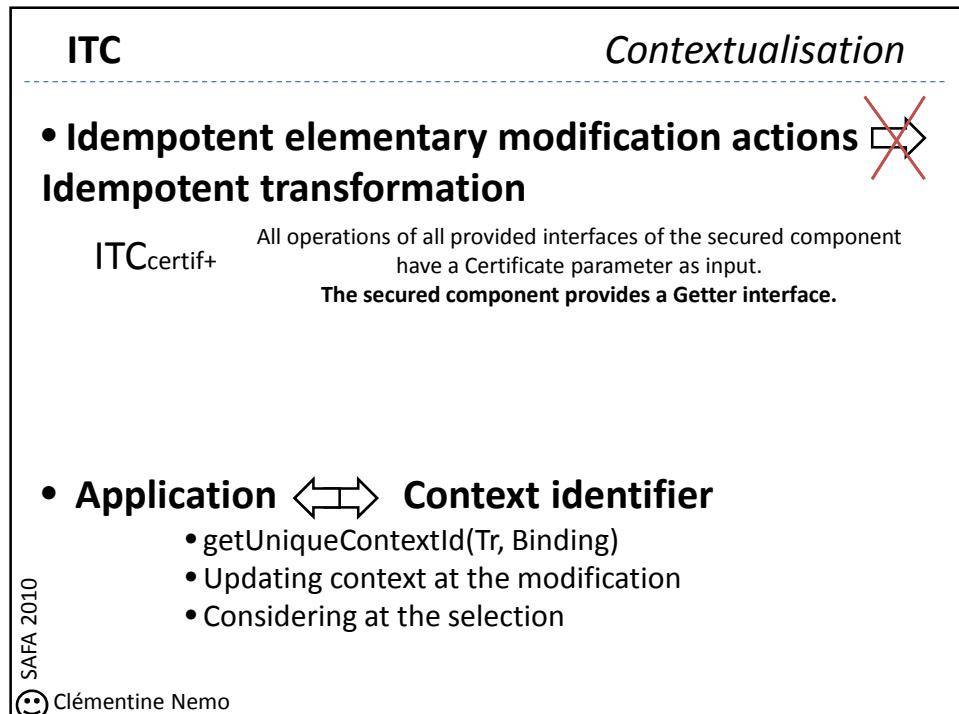
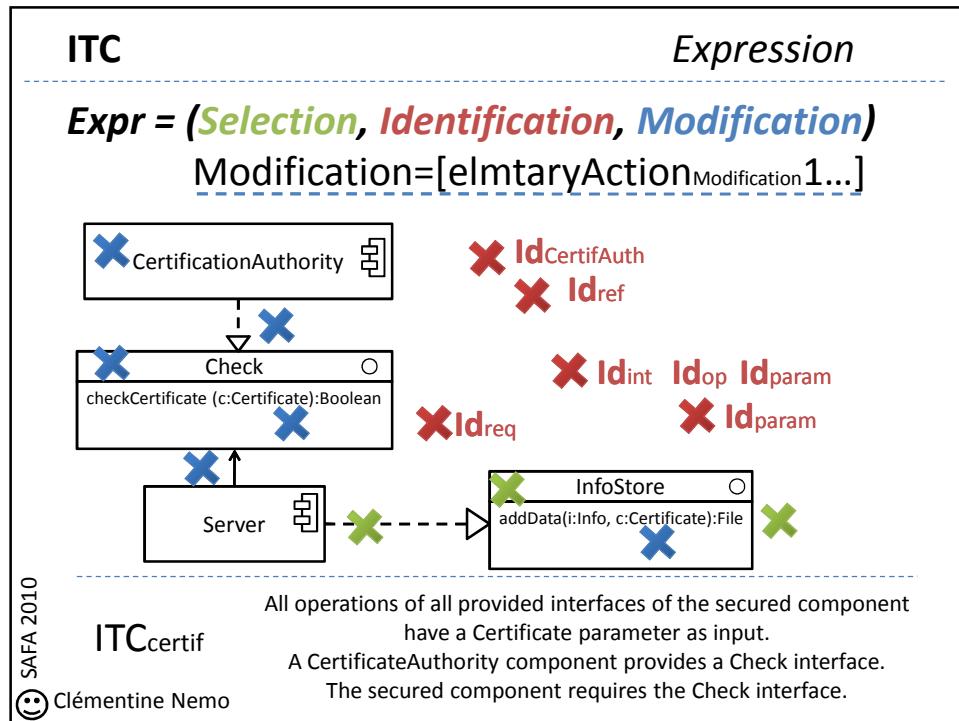
ITCcertif

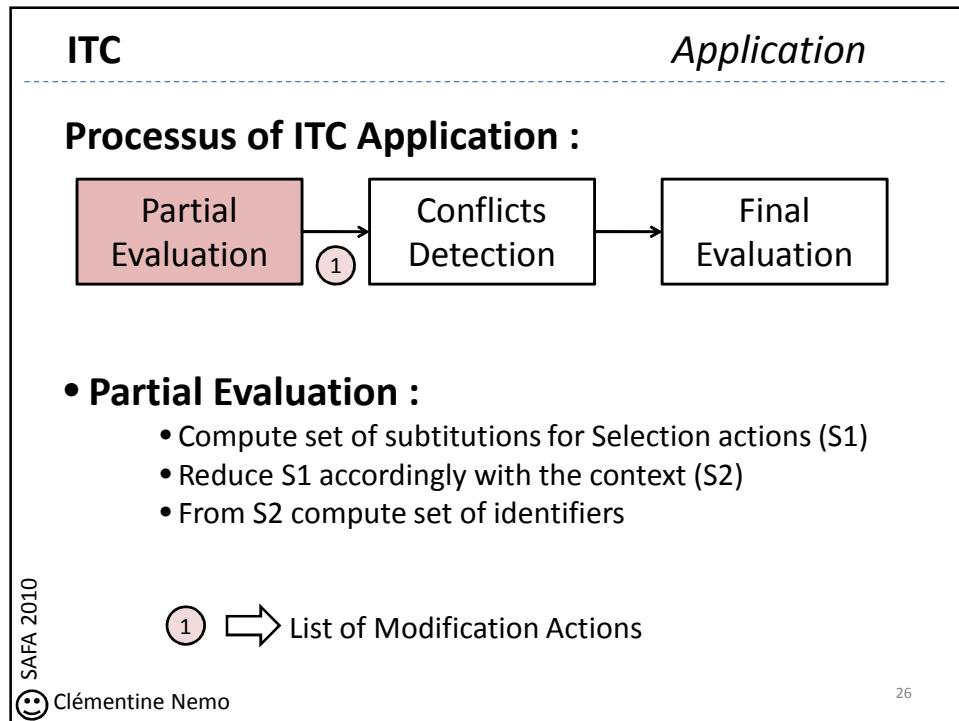
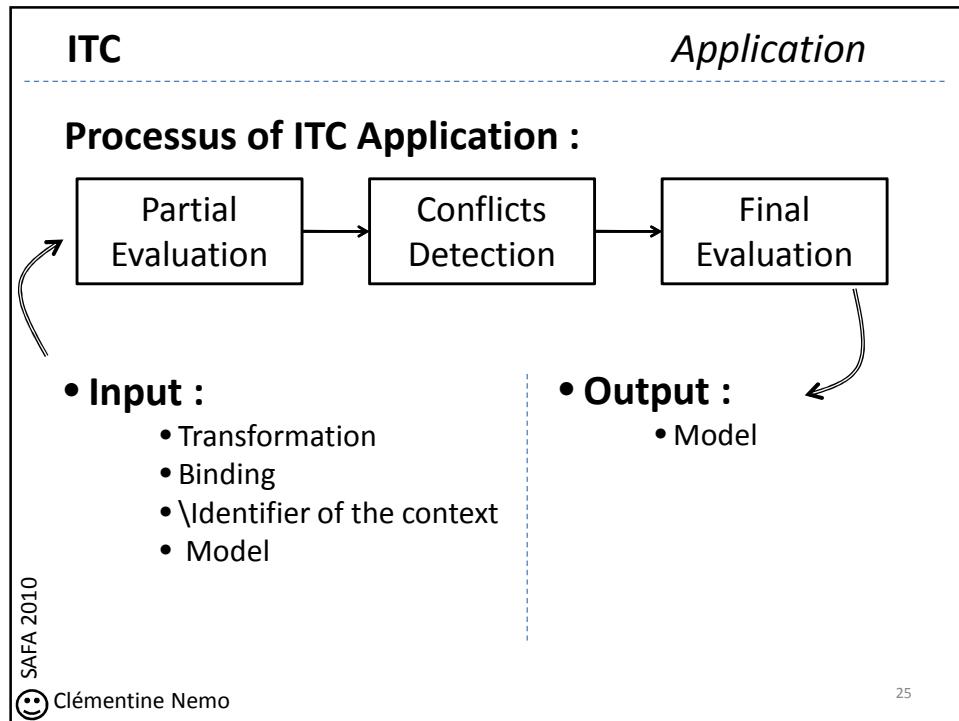
A CertificateAuthority component provides a Check interface.

(;) Clémentine Nemo

The secured component requires the Check interface.

SAFA 2010





ITC *Application*

Processus of ITC Application :

```

graph LR
    A[Partial Evaluation] -->|1| B[Conflicts Detection]
    B -->|2| C[Final Evaluation]
    
```

The flowchart shows three main stages: Partial Evaluation, Conflicts Detection, and Final Evaluation. Stage 1 (Partial Evaluation) leads to stage 2 (Conflicts Detection). Stage 2 (Conflicts Detection) leads to stage 3 (Final Evaluation).

Below the flowchart:

- Partial Evaluation: List of Modification Actions
- Conflicts Detection: Valid List of Modification Actions
- Final Evaluation: (No explicit output listed)

• **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 :

```

graph LR
    A[Partial Evaluation] -->|1| B[Conflicts Detection]
    B -->|2| C[Final Evaluation]
    
```

The flowchart shows three main stages: Partial Evaluation, Conflicts Detection, and Final Evaluation. Stage 1 (Partial Evaluation) leads to stage 2 (Conflicts Detection). Stage 2 (Conflicts Detection) leads to stage 3 (Final Evaluation).

Below the flowchart:

- Partial Evaluation: List of Modification Actions
- Conflicts Detection: Valid List of Modification Actions
- Final Evaluation: (No explicit output listed)

• **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. Kriesel, and O. Runge, "Transformation dependency analysis- A comparison of two approaches," in Langages et Mod'èles à 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 Model Transformation(ICMT'08). Springer, Jul.2008.
- [SAN] J. Sanchez and J. Garcia, Approaches for Model Transformation Reuse:Factorization and Composition. Springer, 2008.

ITC *Contextualisation*

Contextualisation : $\text{getUniqueContextId}(\text{ITCcertif+}, \{\text{Secured} \rightarrow \text{Server}\}) = \text{○}$

ITCcertif+ All operations of all provided interfaces of the secured component have a Certificate parameter as input.
The secured component provides a Getter interface.

SAFA 2010

Clémentine Nemo

ITC *Contextualisation*

Contextualisation : $\text{getUniqueContextId}(\text{ITCcertif+}, \{\text{Secured} \rightarrow \text{Server}\}) = \text{○}$

ITCcertif+ All operations of all provided interfaces of the secured component have a Certificate parameter as input.
The secured component provides a Getter interface.

SAFA 2010

Clémentine Nemo