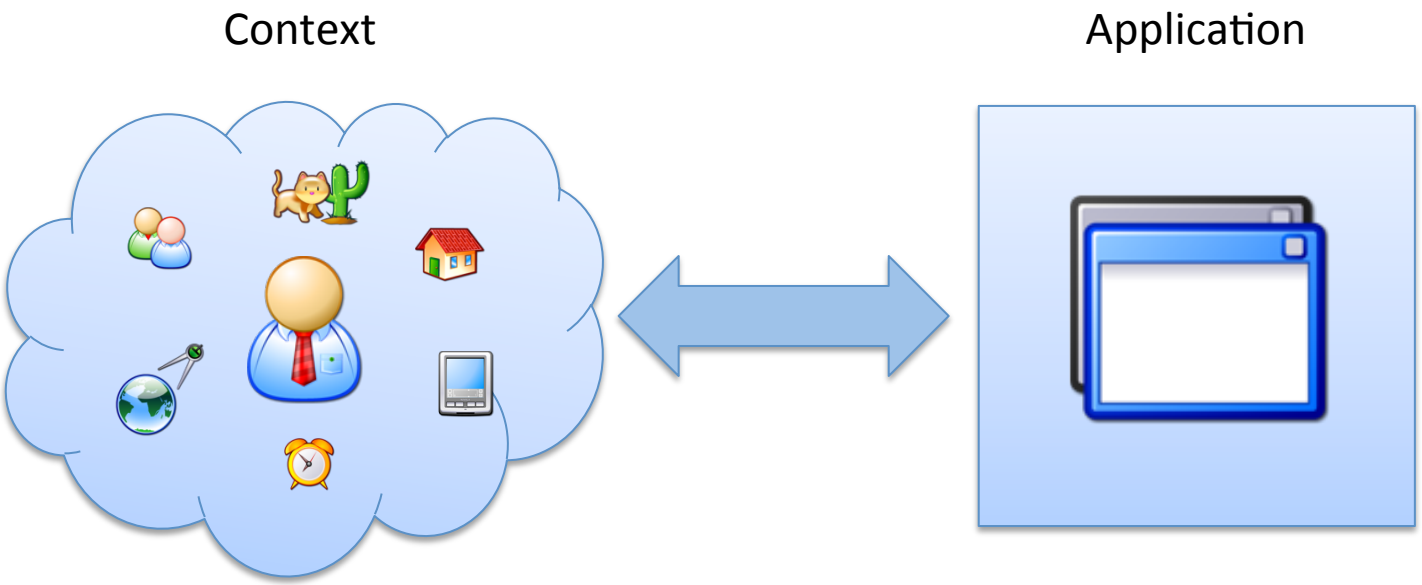
A Model-Driven Approach for Incorporating Reactive Rules in Declarative Interactive TV Applications

Patrícia Dockhorn Costa, João Paulo A. Almeida, Igor Magri Vale, Izon Thomaz Mielke

CS Department, Federal University of Espírito Santo, Brazil

pdcosta@inf.ufes.br, jpalmeida@ieee.org, igormagrivale@gmail.com, izontm@gmail.com

Context-Aware Application



Context-aware application is a **distributed** application whose behaviour is **affected by** its users' **context**.

Some Application Areas

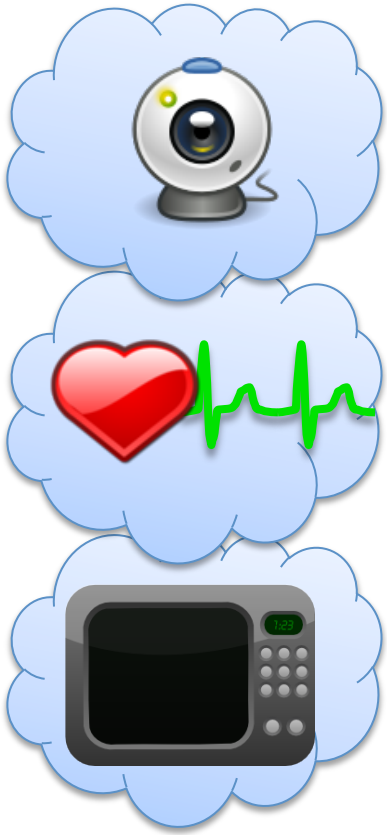
- Health Care;
- Smart Home;
- Business Processes;
- Social Network;
- Network Security;
- **Interactive Digital TV.**

Basic Requirements

Capture Context and Model Behaviour

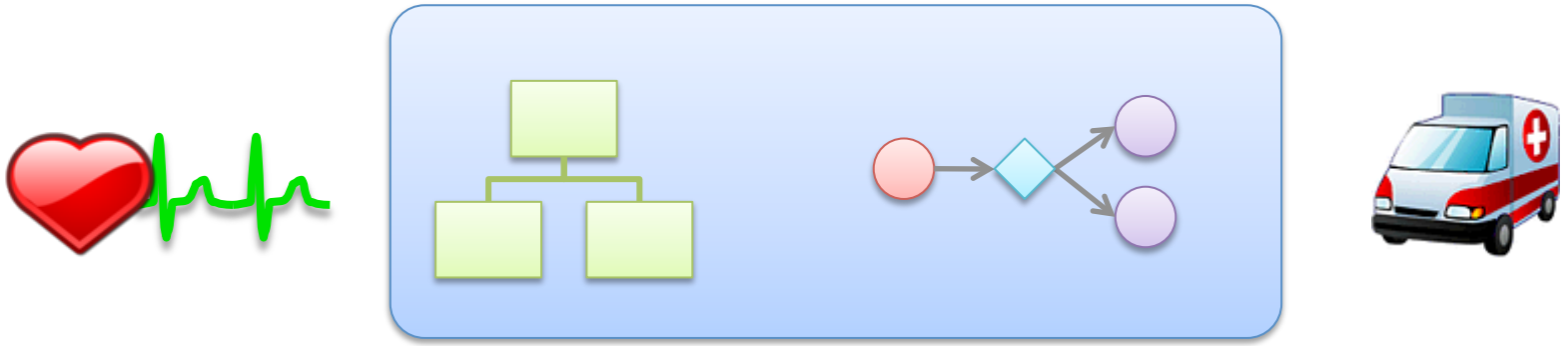
```
require 'rubygems'
require 'atchoum'
class SneezyWeb < Atchoum::Website
  def layout
    html do
      head do
        title 'I sneeze'
      end
      body do
        self << yield
      end
    end
  end
  def index_page
    text 'Checkout my'
    a 'poem', :href => :poem
  end
  def poem_page
    h1 'The allergies season'
    p 'Flowers are red'
    p 'Buds open'
    p 'I sneeze a lot!'
  end
end
```

Various Sensors in a Distributed environment



```
require 'rubygems'
require 'atchoum'
class SneezyWeb < Atchoum::Website
  def layout
    html do
      head do
        title 'I sneeze'
      end
      body do
        self << yield
      end
    end
  end
  def index_page
    text 'Checkout my'
    a 'poem', :href => :poem
  end
  def poem_page
    h1 'The allergies season'
    p 'Flowers are red'
    p 'Buds open'
    p 'I sneeze a lot!'
  end
end
```

Supporting Context-Aware Application Development



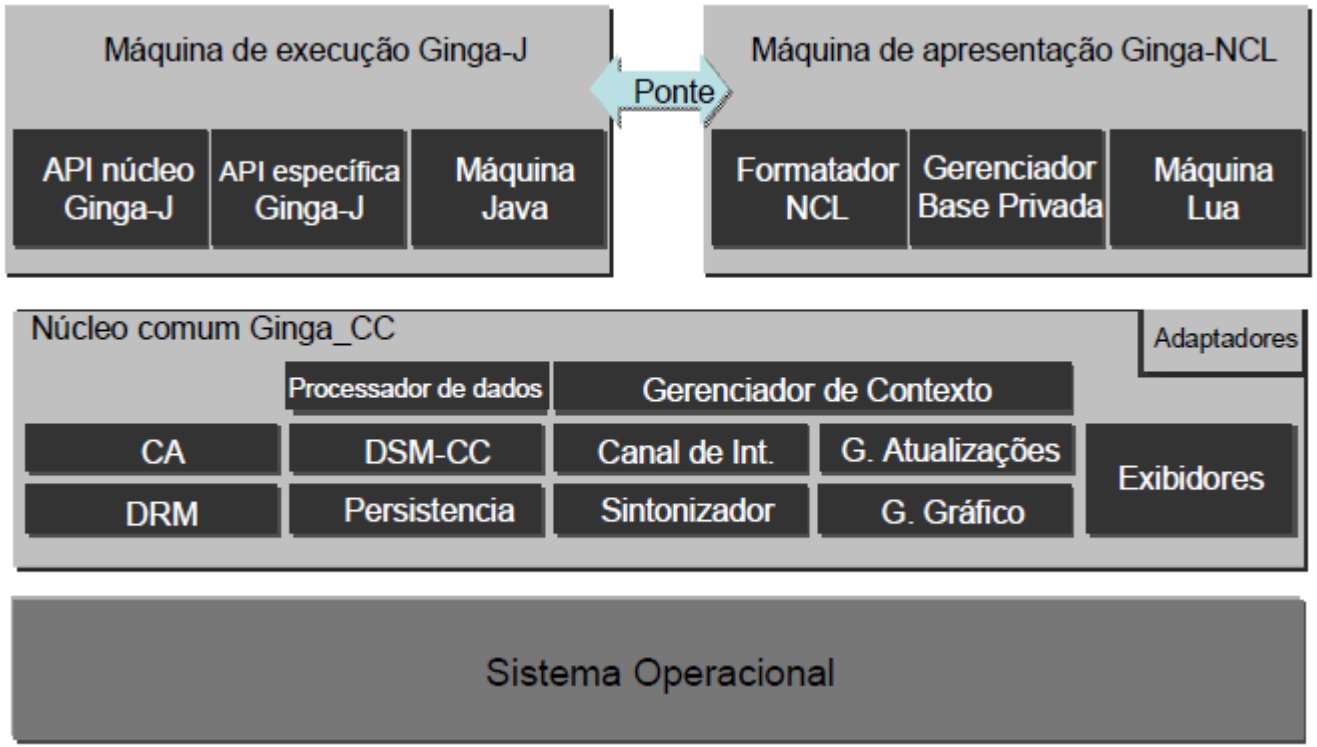
Supporting Context-Aware Application Development (cont.)

- Quite some work on that:
  - context-aware patterns;
  - services platform (context sources, managers, controllers, action resolvers, etc.);
  - context and situation models;
  - situation reasoning and (distributed) detection;
  - ECA language for modelling reactive behaviours, etc.

The SBTVD

- Better quality of images/videos;
- Provides interactivity;
- Includes standards for data modulation, transmission, data encoding, and a **middleware (coined Ginga)**.

Middleware Ginga



(SOARES e CASTRO, 2008)

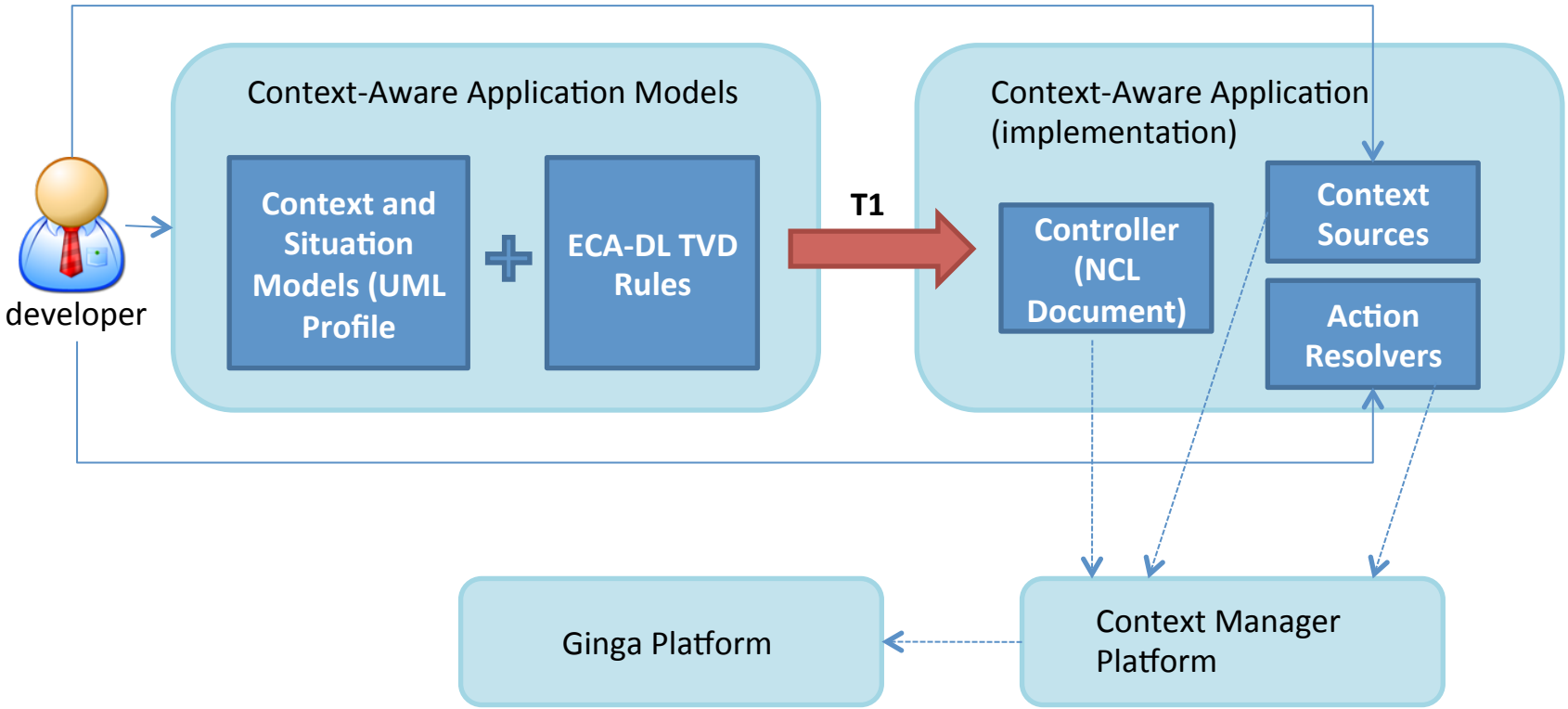
Nested Context Language

- Hypermedia authoring language used to describe multimedia applications with space-time synchronization between media objects (e.g., video, audio, images, etc.);
- As such, it does not have built-in concepts for context-handling and reactivity;
- NCLua
  - Scripting language;
  - Allows execution of imperative code in the declarative environment.

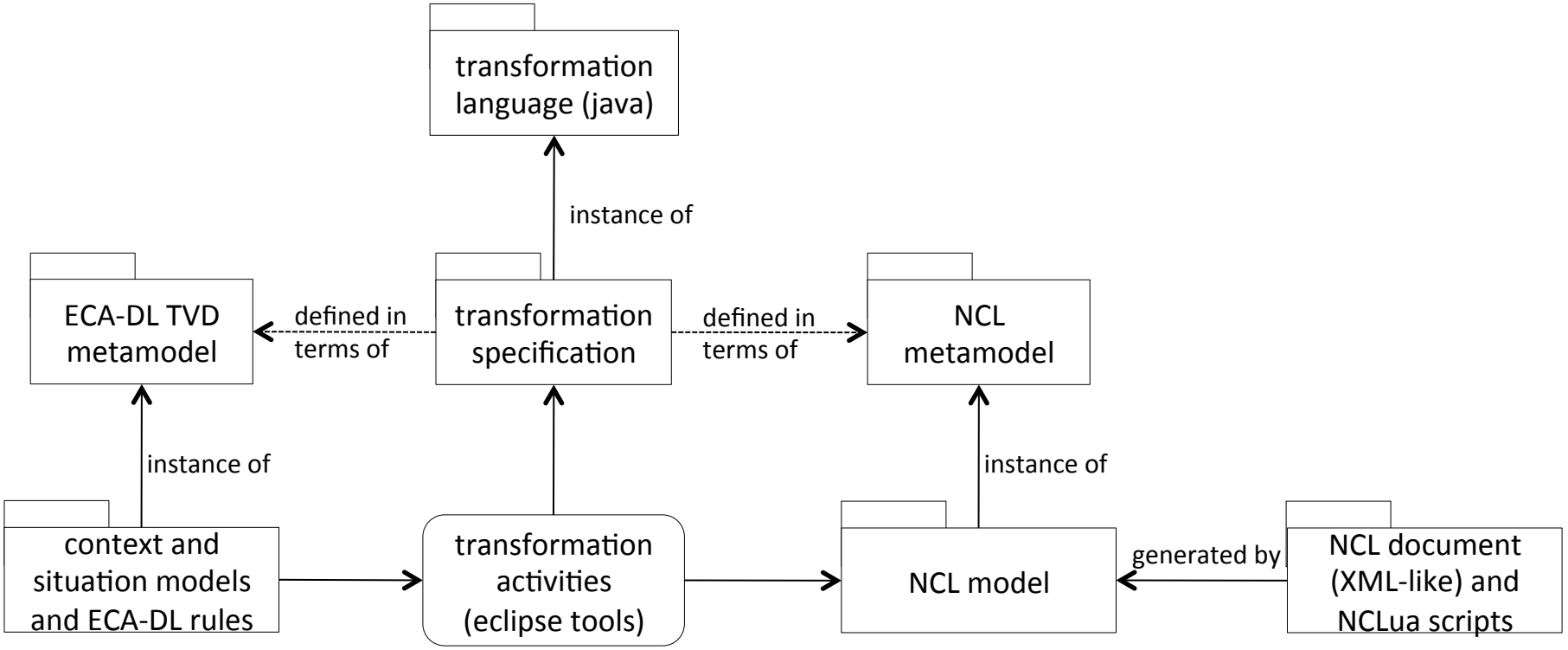
ECA-DL TVD

- Domain-specific language;
- Rule-based;
- Describes the reactive behaviour of context-aware applications in the Interactive TV Domain;
- In line with the **event-condition-action** pattern;
- Rules are defined based on both context and situation models (vocabulary for rule definition).

MDA Approach: Overview



Transformation Overview



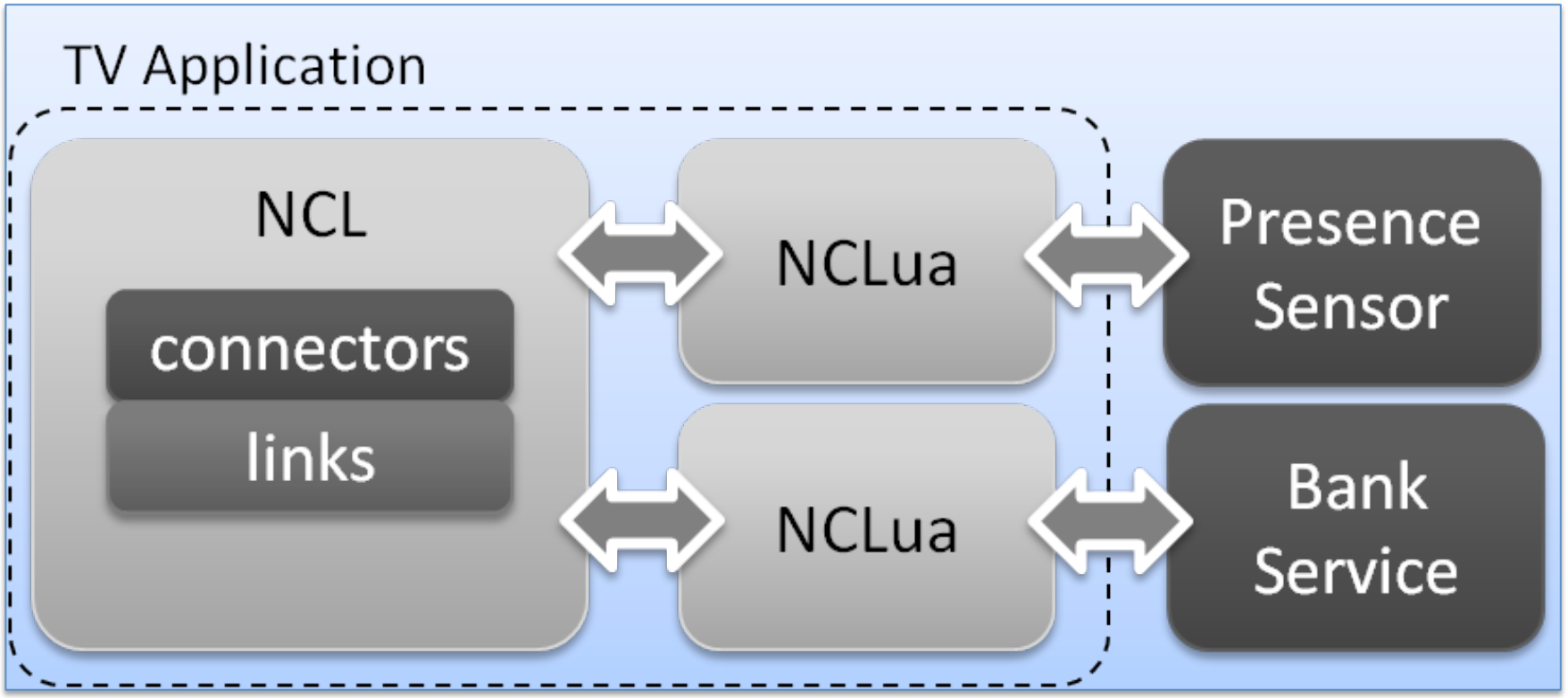
Case Study

Context-aware bank application that hides private information autonomously

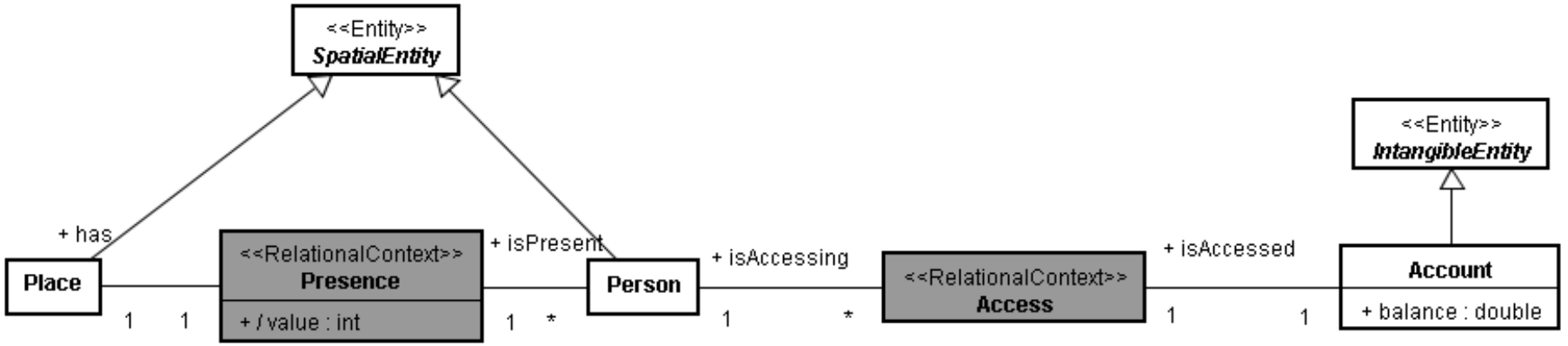
John uses a web-based bank application through his TV

prototipo.ogv

Context-Aware Application in NCL

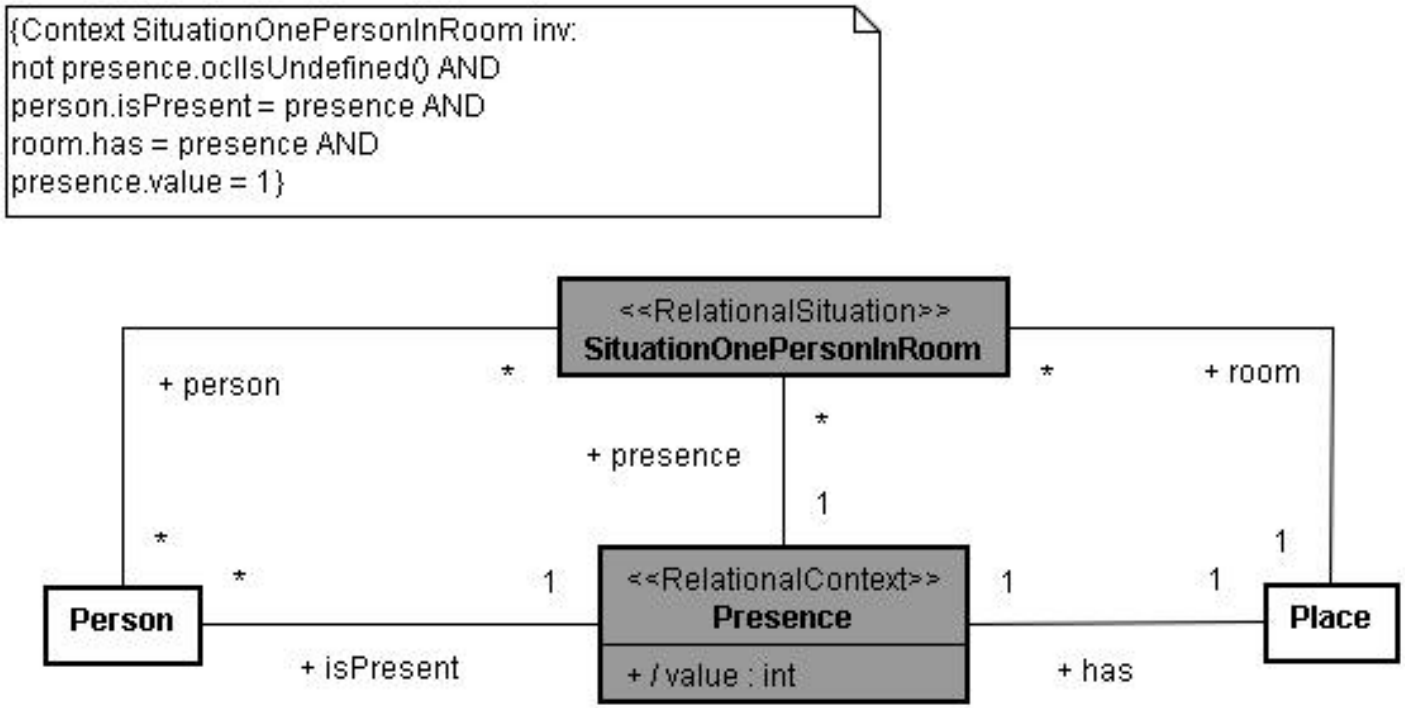


Case Study: Context Model



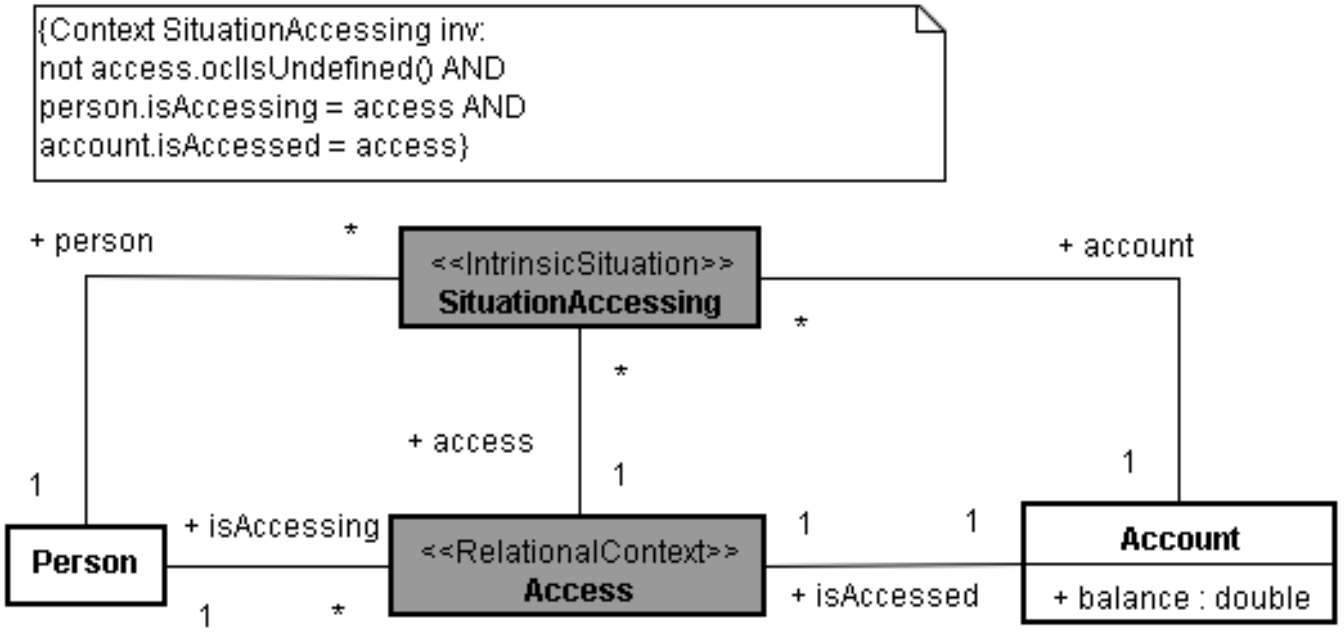
Case Study: Situation Model

- Situation “One person in the room”:



Case Study: Situation Model

- Situation “Accessing”:



ECA-DL TVD rules

Upon EnterTrue SituationAccessing (person1)
When SituationOnePersonInRoom (room1)
Do startWithBalance(account1)

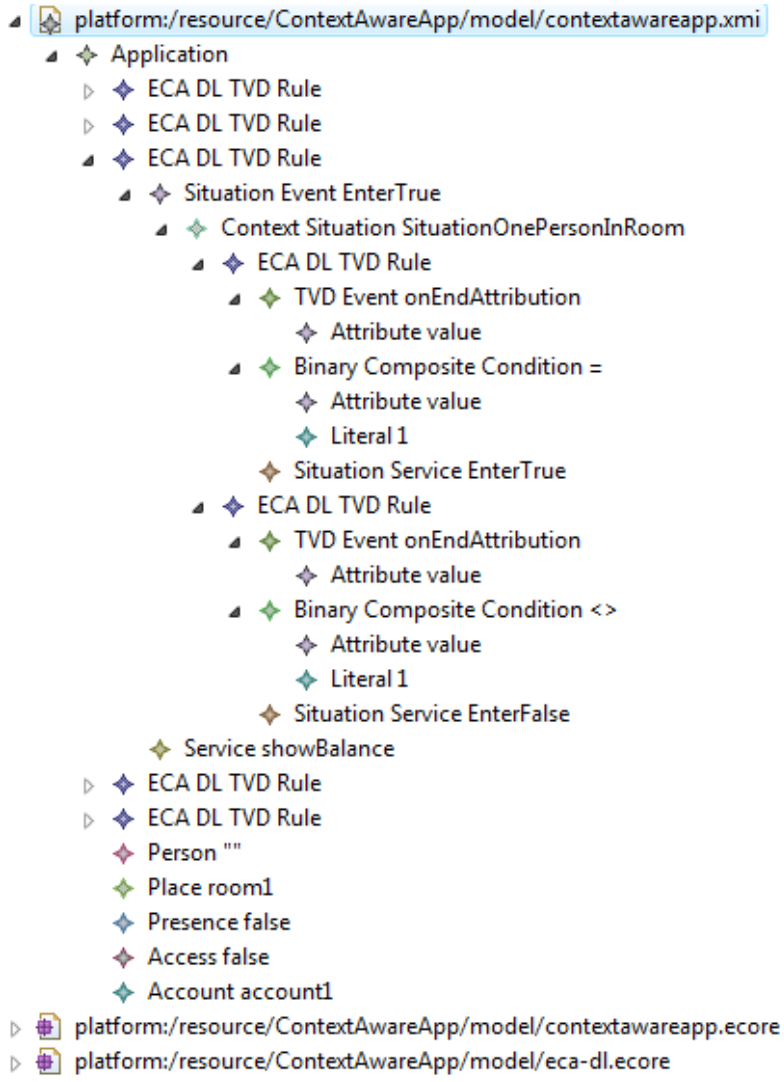
Upon EnterTrue SituationAccessing (person1)
When SituationMoreThanOnePersonInRoom (room1)
Do startWithoutBalance(account1)

Upon EnterTrue SituationOnePersonInRoom (room1)
Do showBalance(account1)

Upon EnterTrue SituationMoreThanOnePersonInRoom (room1)
Do hideBalance(account1)

Upon EnterTrue SituationNoPersonInRoom (room1)
Do logout(account1)

Ecore Model


platform:/resource/ContextAwareApp/model/contextawareapp.xmi
Application
ECA DL TVD Rule
ECA DL TVD Rule
ECA DL TVD Rule
Situation Event EnterTrue
Context Situation SituationOnePersonInRoom
ECA DL TVD Rule
TVD Event onEndAttribution
Attribute value
Binary Composite Condition =
Attribute value
Literal 1
Situation Service EnterTrue
ECA DL TVD Rule
TVD Event onEndAttribution
Attribute value
Binary Composite Condition <>
Attribute value
Literal 1
Situation Service EnterFalse
Service showBalance
ECA DL TVD Rule
ECA DL TVD Rule
Person ""
Place room1
Presence false
Access false
Account account1
platform:/resource/ContextAwareApp/model/contextawareapp.ecore
platform:/resource/ContextAwareApp/model/eca-dl.ecore

Entities, Context and situations in NCL

- Entities: mapped to Media elements;
- Intrinsic context types: mapped to Property Anchor elements;
- Relational Context types and Situation types: mapped to Media elements, whose properties are the entities and contexts involved in that situation.

Generated NCL Document

- Entities and Intrinsic Context types:

```
…
<media id="person1" src="scripts/person1.lua" descriptor="person1Desc">
          <property name="name" value=" " />
</media>
…
```

- Relational Context types:

```
…
<media id="Presence_person1_room1" src="scripts/Presence_person1_room1.lua"
descriptor="Presence_person1_room1_Desc">
          <property name="person" value="person1" />
          <property name="place" value="room1" />
          <property name="value" value=" " />
          <property name="exists" value="false" />
</media>
…
```

Rules in NCL: connectors and links

- Used to implement synchronism between NCL elements
  - E.g.: at the end of a video, starts another video
- Allows (with some adaptation) implementation of context-aware reaction rules (ECA-DL TVD rules)

Generated Context Situations

Upon onEndAttribution (room1.has.value)
When room1.has.value <> 1
Do EnterFalse SituationOnePersonInRoom (room1)

```
…
<causalConnector id="SituationMoreThanOnePersonInRoom_room1_EnterFalse">
          <compoundCondition operator="and">
                    <simpleCondition role="upon10" transition="stops" eventType="attribution" />
                    <assessmentStatement comparator="lte">
                              <attributeAssessment role="when8" attributeType="nodeProperty"
eventType="attribution" />
                                        <valueAssessment value="1"/>
                              </assessmentStatement>
          </compoundCondition>
          <simpleAction role="action11" actionType="set" eventType="attribution" value="true" /
>
</causalConnector>
…
<link xconnector="SituationMoreThanOnePersonInRoom_room1_EnterFalse">
          <bind component="Presence_person1_room1" interface="value" role="upon10" />
          <bind component="Presence_person1_room1" interface="value" role="when8" />
          <bind component="SituationMoreThanOnePersonInRoom_room1" interface="end"
role="action11" />
</link>
…
```

Generated ECA-DL TVD

Upon EnterTrue
SituationOnePersonInRoom (room1)
Do showBalance(account1)

```
…
<causalConnector id="eca_dl_tvd_rule3">
            <compoundCondition operator="and">
                        <simpleCondition role="upon11" />
                        <assessmentStatement comparator="eq">
                                    <attributeAssessment role="upon12" attributeType="nodeProperty"
eventType="attribution" />
                                    <valueAssessment value="true"/>
                        </assessmentStatement>
            </compoundCondition>
            <simpleAction role="action14" actionType="set" eventType="attribution" value="true" />
</causalConnector>
…
<link xconnector="eca_dl_tvd_rule3">
            <bind component="SituationOnePersonInRoom_room1" interface="start" role="upon11" />
            <bind component="SituationOnePersonInRoom_room1" interface="start" role="upon12" />
            <bind component="account1" interface="showBalance" role="action14"/>
</link>
…
```

Final Considerations

- Our approach provides **high-level abstractions** to express application behavior conveniently in terms of **context events** and corresponding **application reactions;**
- Main challenge: transforming a rule-based specification at the platform-independent level into an implementation platform that is not rule-oriented, while preserving conformance to the standard.

On-going and Future Work

- Improving the ECA-DL TVD editor (which is currently the standard tree editor generated by Eclipse);
- Looking into mobile interactive TV context-aware applications:
  - NCL at the set-top box and rule-based implementations on back-end servers (Drools?);
  - Cope with scalability challenges that arise from managing a large number of mobile devices and context sources.

Thank you!