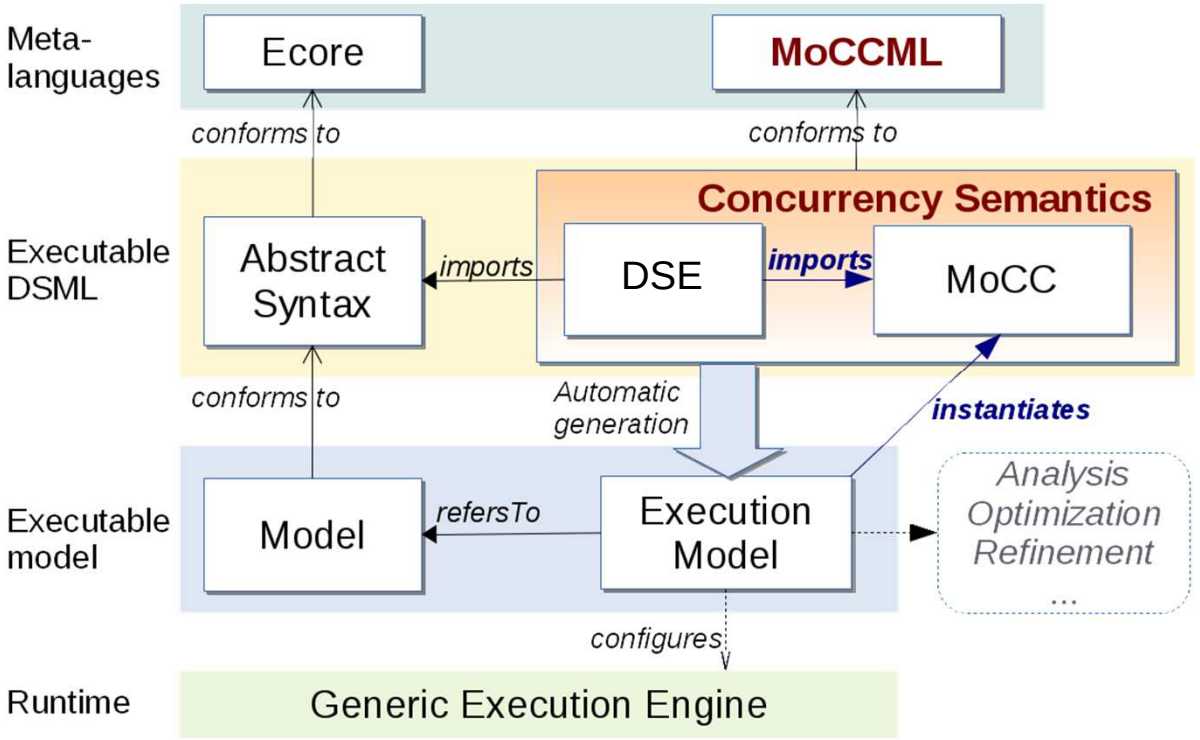
Concurrency reification in the xDSML with MoCCML

Gemoc Final Workshop, March 17th, 2016

Joël Champeau (Lab-STICC – ENSTA Bretagne)
Joel.champeau@ensta-bretagne.fr

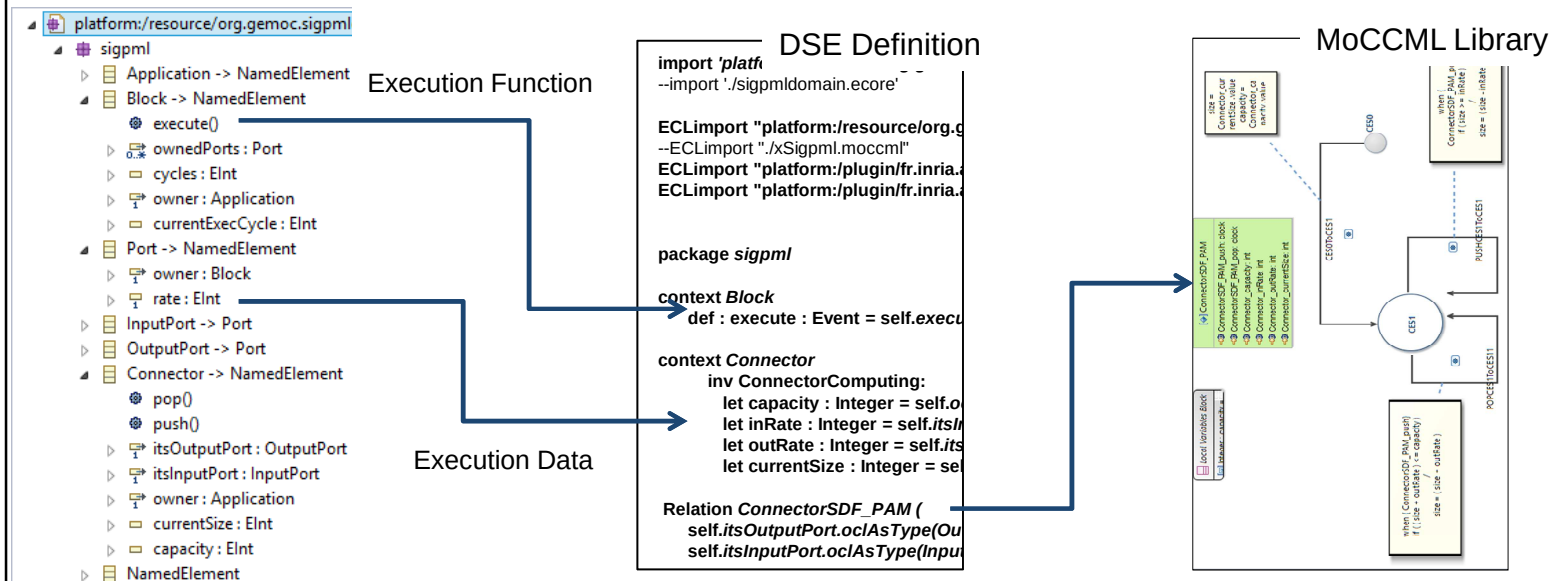
The MoCCML approach



DSE definition and MoCCML Libraries

Focus on DSE and MoCCML specifications

- Domain Model with Execution Function and Data
- DSE = Event definition on EF
- MoCCML definition = Declarative constraints on events



DSE declaration

```
import 'platform:/resource/org.gemoc.sigpmldomain/model/sigpmldomain.ecore'

ECLimport "platform:/resource/org.gemoc.sigpmltuto.mocc/mocc/SigpmlTuto.moccml"
ECLimport "platform:/plugin/fr.inria.aoste.timesquare.ccslkernel.model/ccsllibrary/kernel.ccslLib"
ECLimport "platform:/plugin/fr.inria.aoste.timesquare.ccslkernel.model/ccsllibrary/CCSL.ccslLib"
```

```
package sigpml

    context Block

        def : execute : Event = self.execute()

    context Connector

        inv ConnectorSynchronization:

        Relation Coincides( self.itsOutputPort.oclAsType(OutputPort).owner.oclAsType(Block).execute,

                            self.itsInputPort.oclAsType(InputPort).owner.oclAsType(Block).execute )

endpackage
```

DSE declaration

```
context Block
        def : execute : Event = self.execute()
context Connector
    inv ConnectorComputing:
    let capacity : Integer = self.oclAsType(Connector).capacity in
    let inRate : Integer = self.itsInputPort.oclAsType(Port).rate in
    let outRate : Integer = self.itsOutputPort.oclAsType(Port).rate in
    let currentSize : Integer = self.oclAsType(Connector).currentSize in

     Relation MoccSDFLike (
                    self.itsOutputPort.oclAsType(OutputPort).owner.oclAsType(Block).execute,
                    self.itsInputPort.oclAsType(InputPort).owner.oclAsType(Block).execute,
                    capacity,
                    inRate,
                    outRate,
                    currentSize  )
endpackage
```

MoCCML meta-language

MoCCML language

- Declarative language dedicated to concurrency definition
- Clock-based approach
- Formally defined
- MoCCML = Declarative operators and state machine constraints

MoCCML in Gemoc

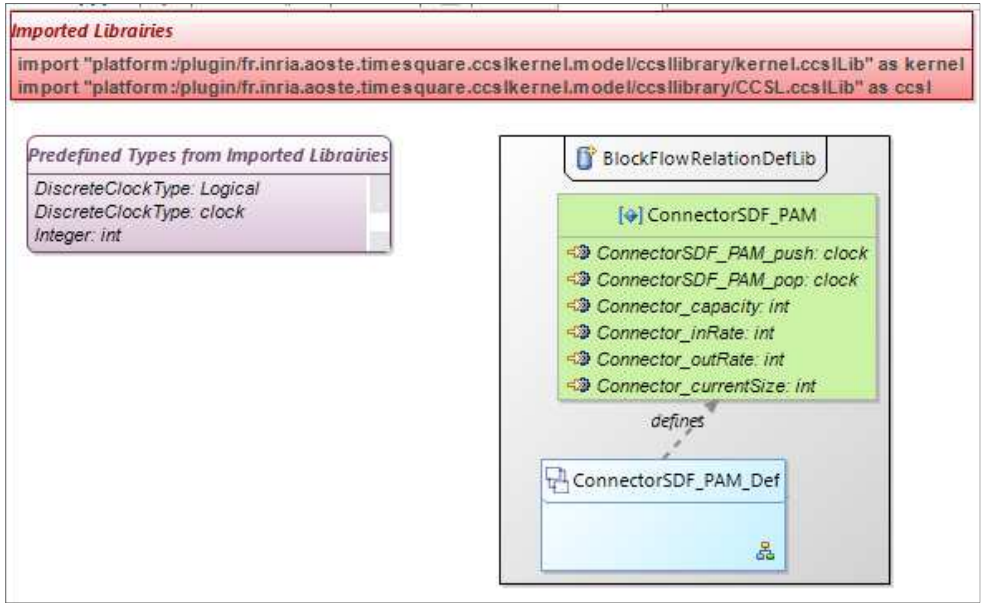
- Concurrency definition at meta-level
- Constraints the model events
- Enables simulation and state-space analysis

Towards a Meta-Language for Concurrency Concern in DSLs
Julien Deantoni, Papa Issa Diallo, Ciprian Teodorov, Joël Champeau, Benoit Combemale,
In Design, Automation and Test in Europe Conference and Exhibition (DATE 2015), 2015

MoCCML language

MoCCML usage

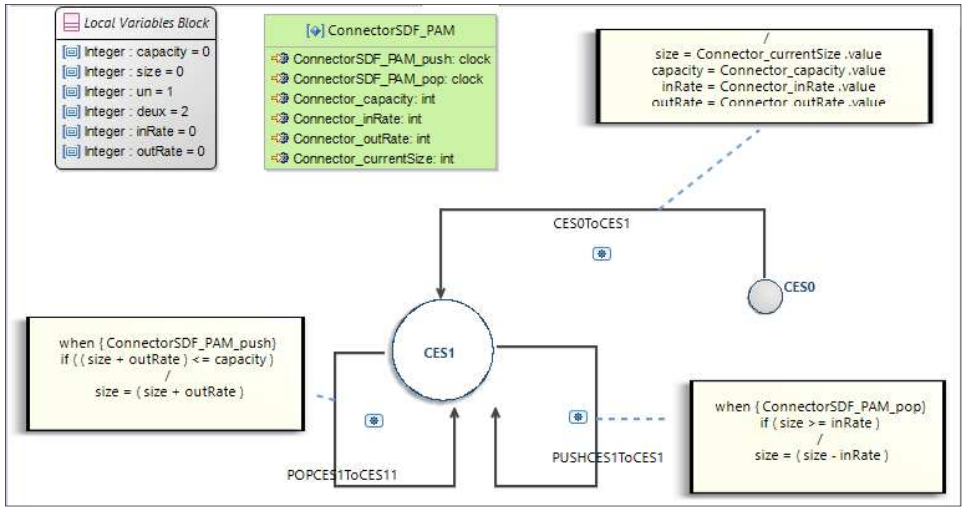
- Declarative predefined operators in DSE definition
- Reusable and parametric library definitions



MoCCML language

MoCCML state machine intuitive semantics

- Triggers as Clocks
- Guards as Condition expressions on integer variables
- Action as assignment and arithmetic operators on variables



MoCCML language

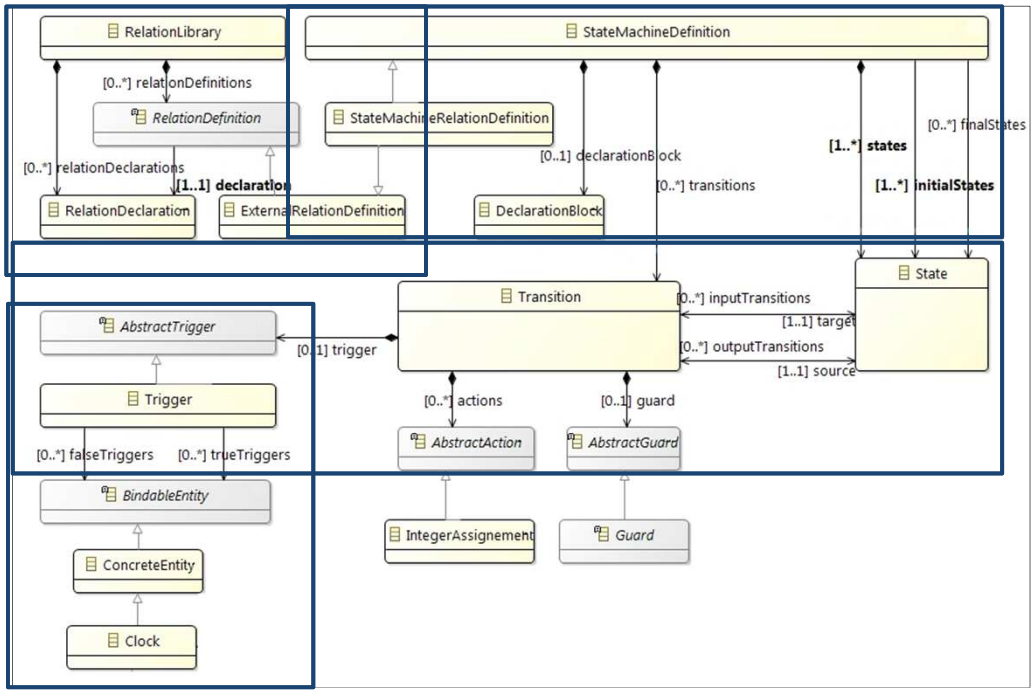
MoCCML models define

- MoCCML relations as a set of constraints on the model events
- The semantics defines a state space with the associated value of the events
- The acceptable schedules are a set of steps

  step = set of possible occurring events
- Adding MoCCML relation reduces the execution possibilities

MoCCML language

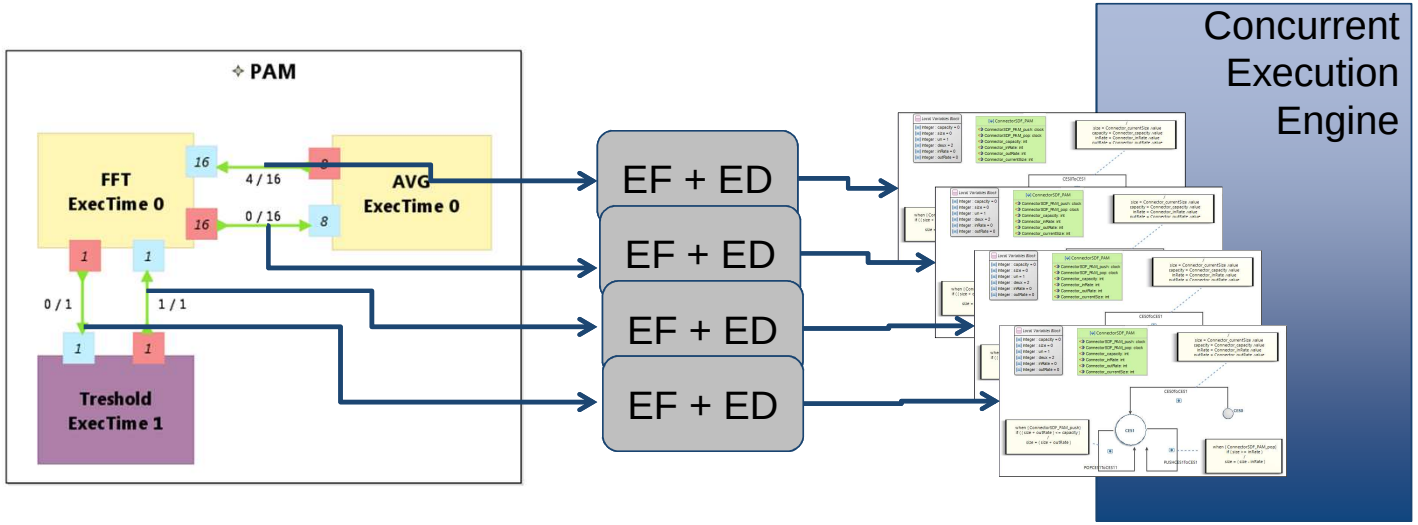
MoCCML metamodel



MoCCML instantiation process

MoCCML model instantiation

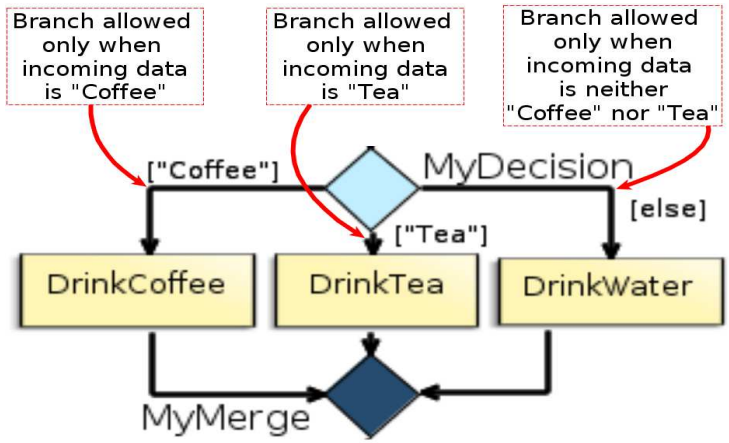
- A MoCCML relation is instantiated for each model element
- The execution model is interpreted by the solver



Backward DSE Mapping

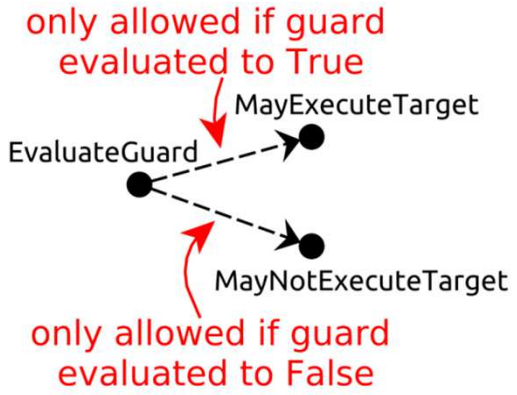
Feedback Protocol

- How a value returned by an Execution Function influences the decision of the execution engine?
- For language constructs whose control flow depends on runtime data available only at runtime (e.g. DecisionNode).



Feedback DSE extension

```
DSE EvaluateGuard:
 upon evaluateGuard
 triggers ActivityEdge.EvaluateGuard
 returning result
 feedback:
  [result] => allow MayExecuteTarget
  default => allow MayNotExecuteTarget
 end
end
```



Weaving Concurrency in eXecutable Domain-Specific Modeling Languages (Florent Latombe, Xavier Crégut, Benoît Combemale, Julien Deantoni, Marc Pantel), In 8th ACM SIGPLAN International Conference on Software Language Engineering (SLE 2015), ACM, 2015.

Concurrency reification in Gemoc

Concurrency specification

- DSE specification including feedback protocol
- MoCCML language
  - Textual and graphical syntax
  - Declarative operators and state machine relations
- MoCCML relations for architecture model to add constraints related to the mapping
- Relation instantiation at model level to simulate the xDSML in the Gemoc Studio (Demo ArduinoDesigner)
- Coupled with a state space analysis tooling as an extension of the Gemoc Studio (Demo ArduinoDesigner)