



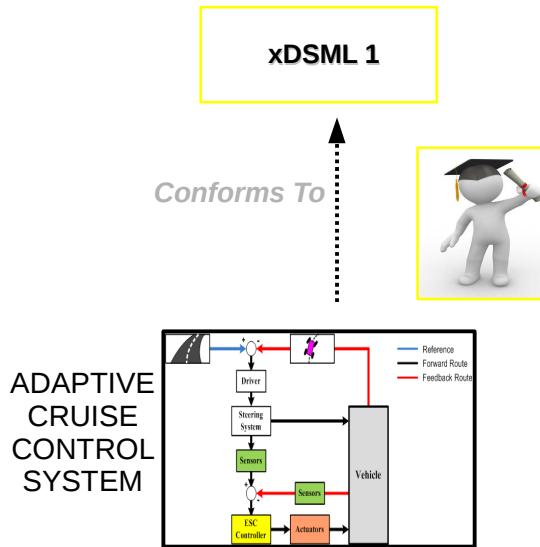
BCOOL

# The **B**ehavioral **C**oordination **O**perator Language

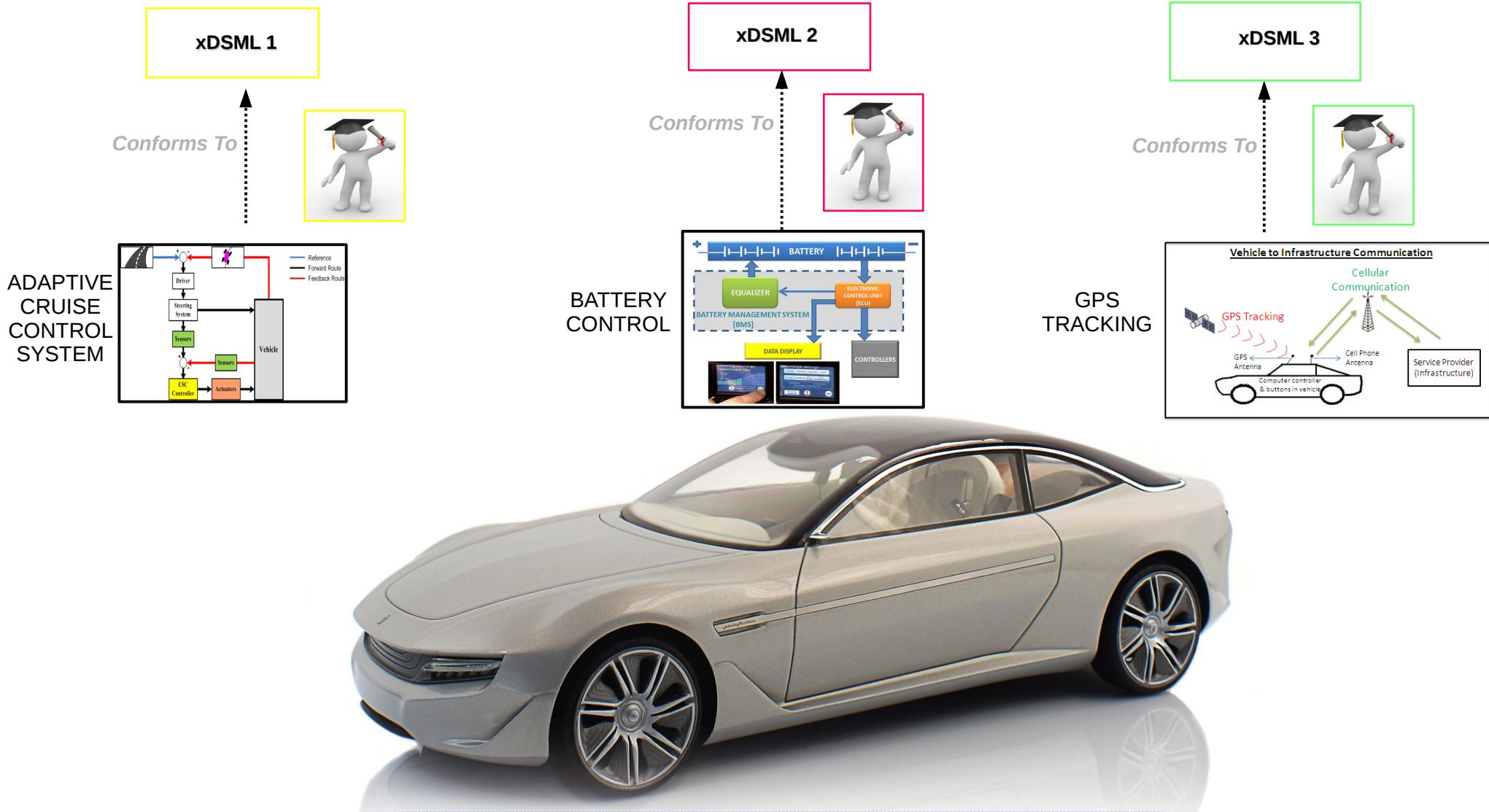
**Gemoc Final Workshop, March 17<sup>th</sup>,  
2016**

Julien Deantoni  
University of Nice, I3S CNRS, INRIA AOSTE  
[Julien.deantoni@polytech.unice.fr](mailto:Julien.deantoni@polytech.unice.fr)

## Discrete Time Behavior



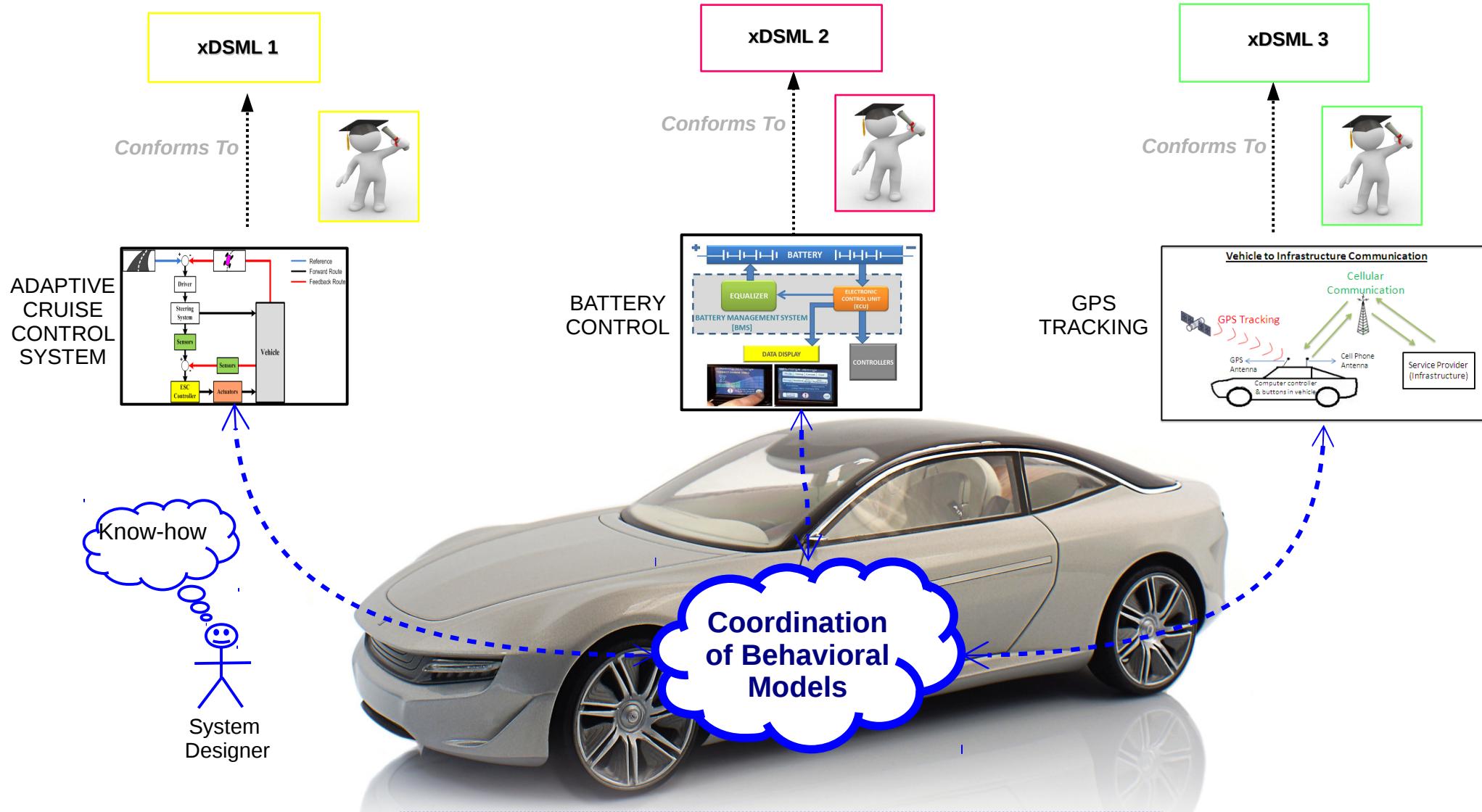
# Discrete Time Behavior



Heterogeneous models and languages

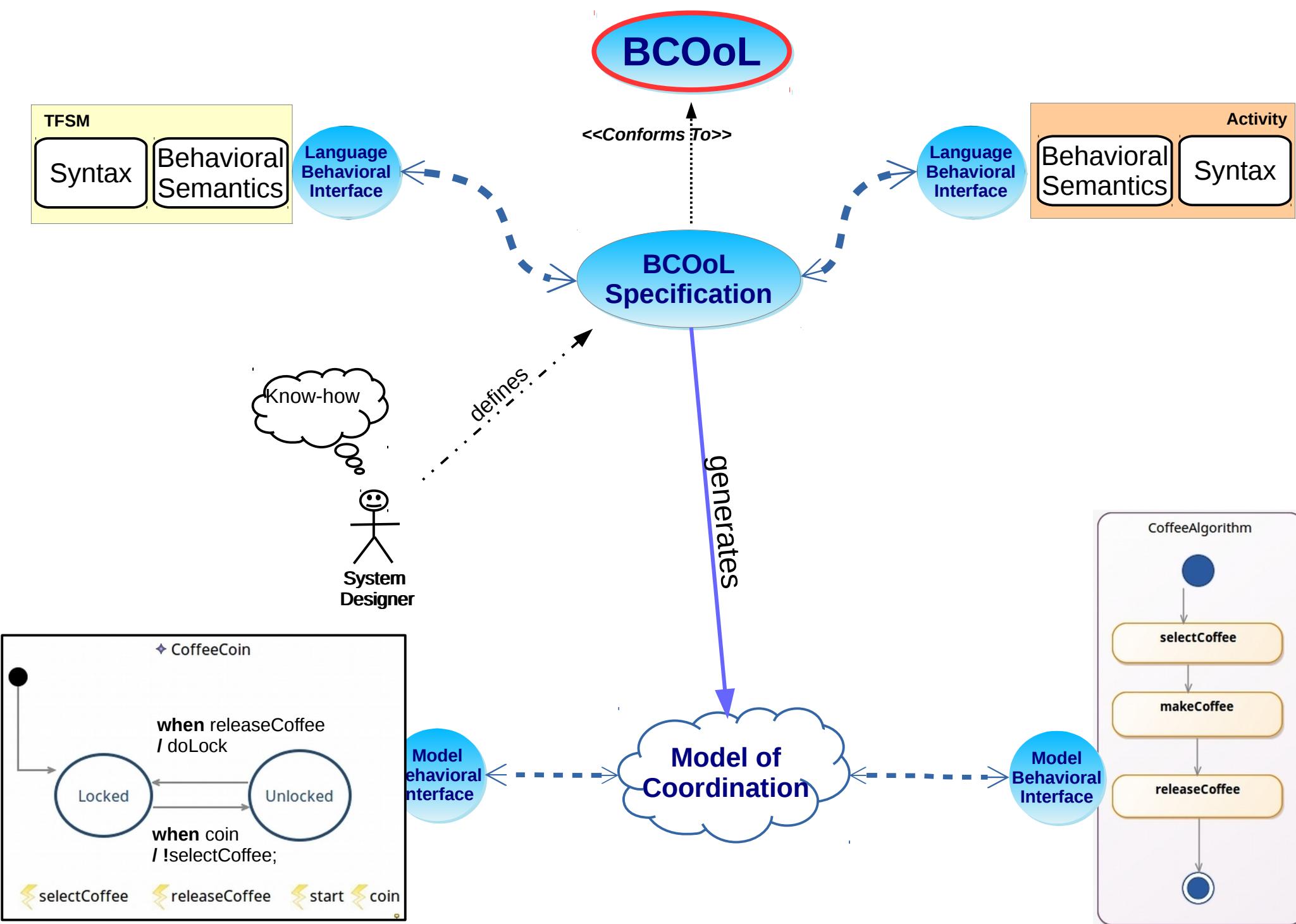
Emerging System Behavior difficult to apprehend

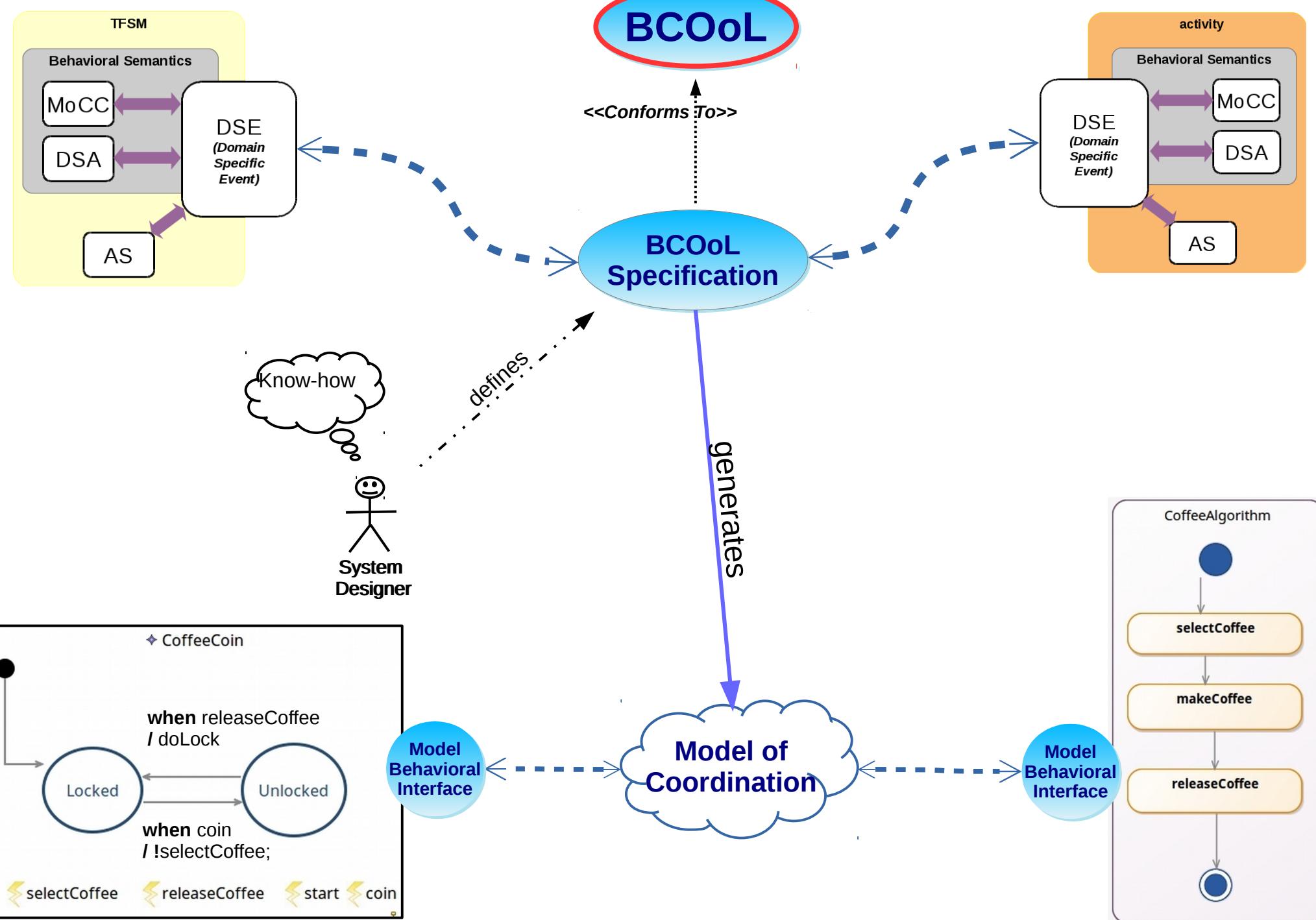
# Discrete Time Behavior

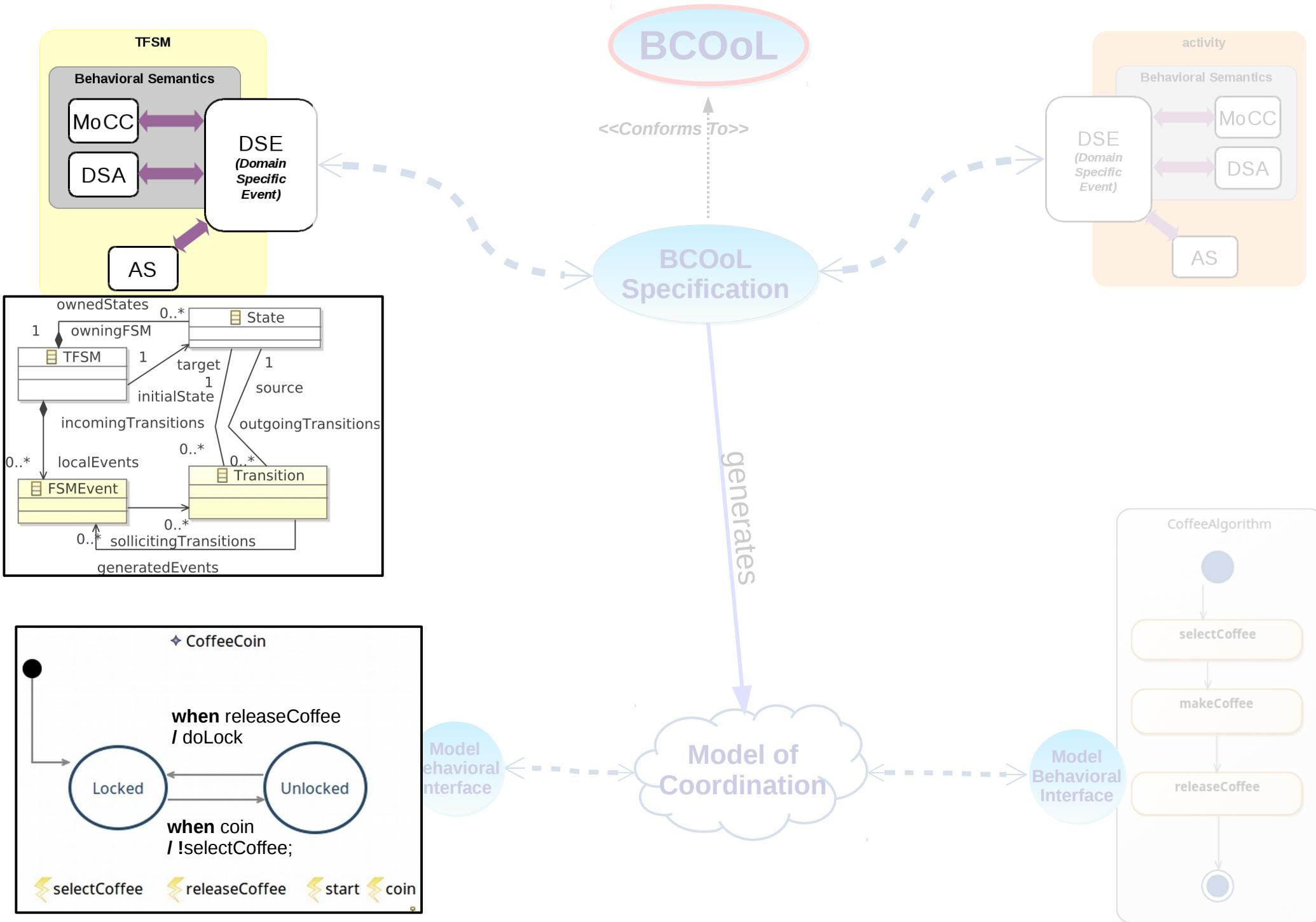


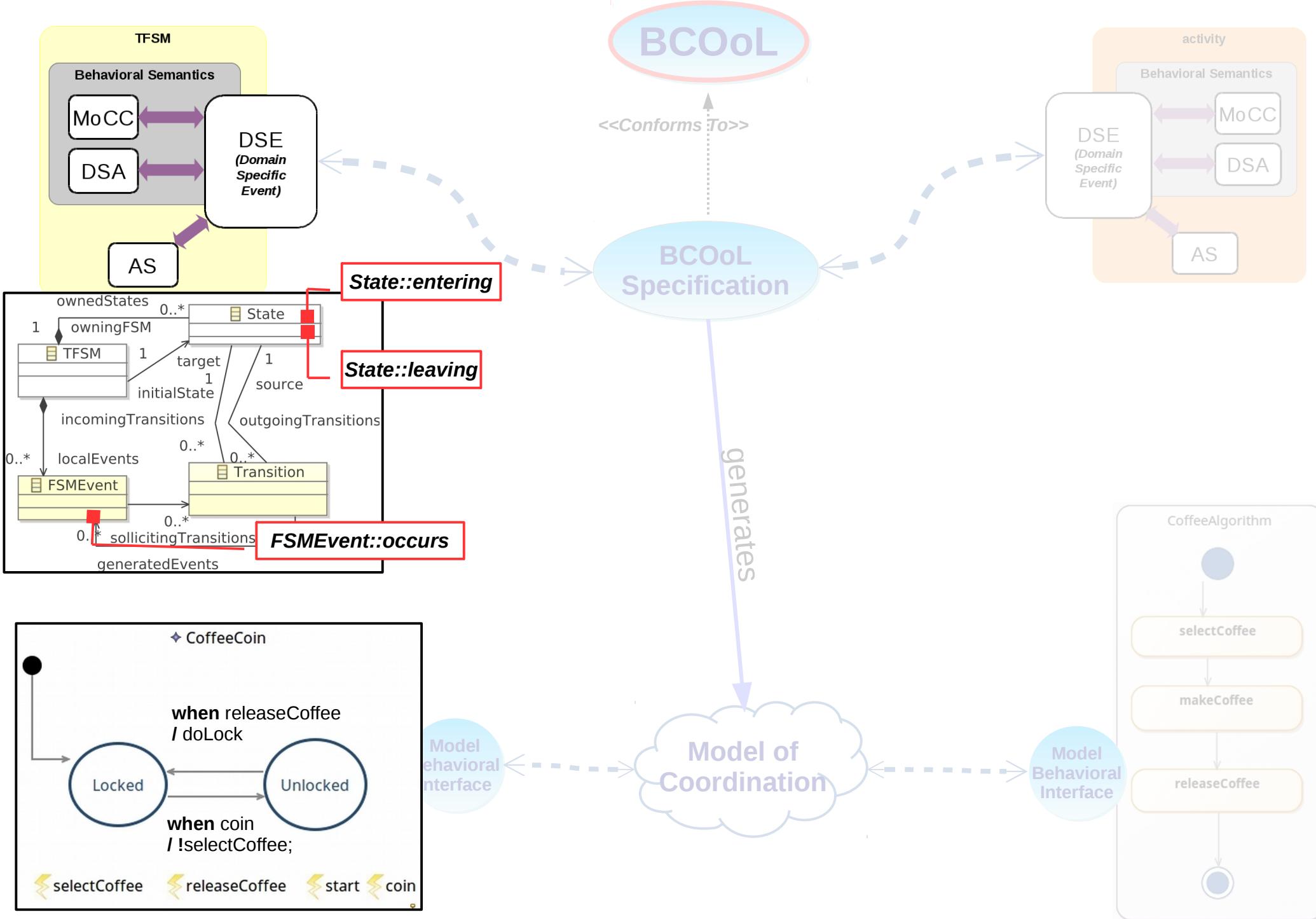
Heterogeneous models and languages

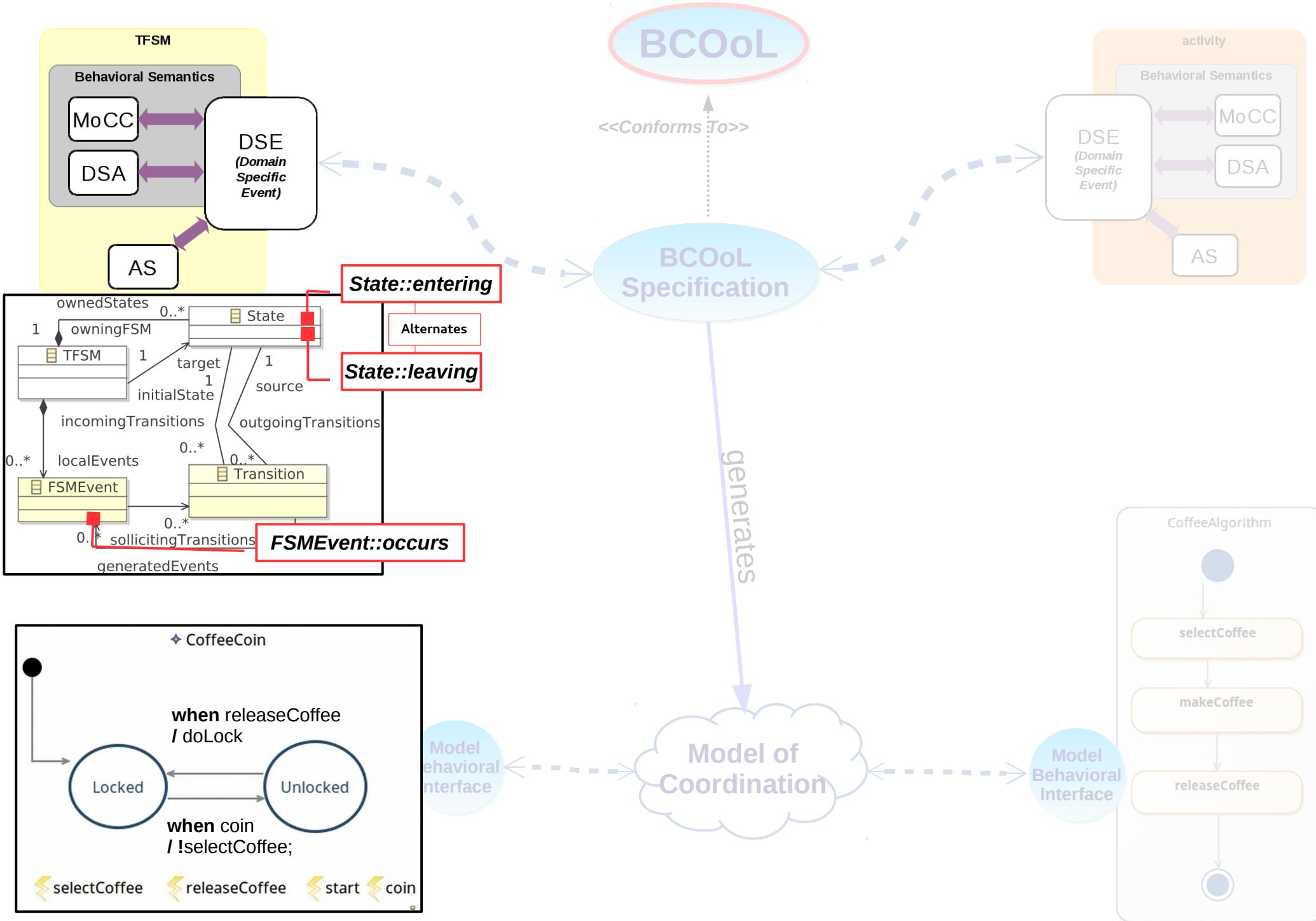
Emerging System Behavior difficult to apprehend

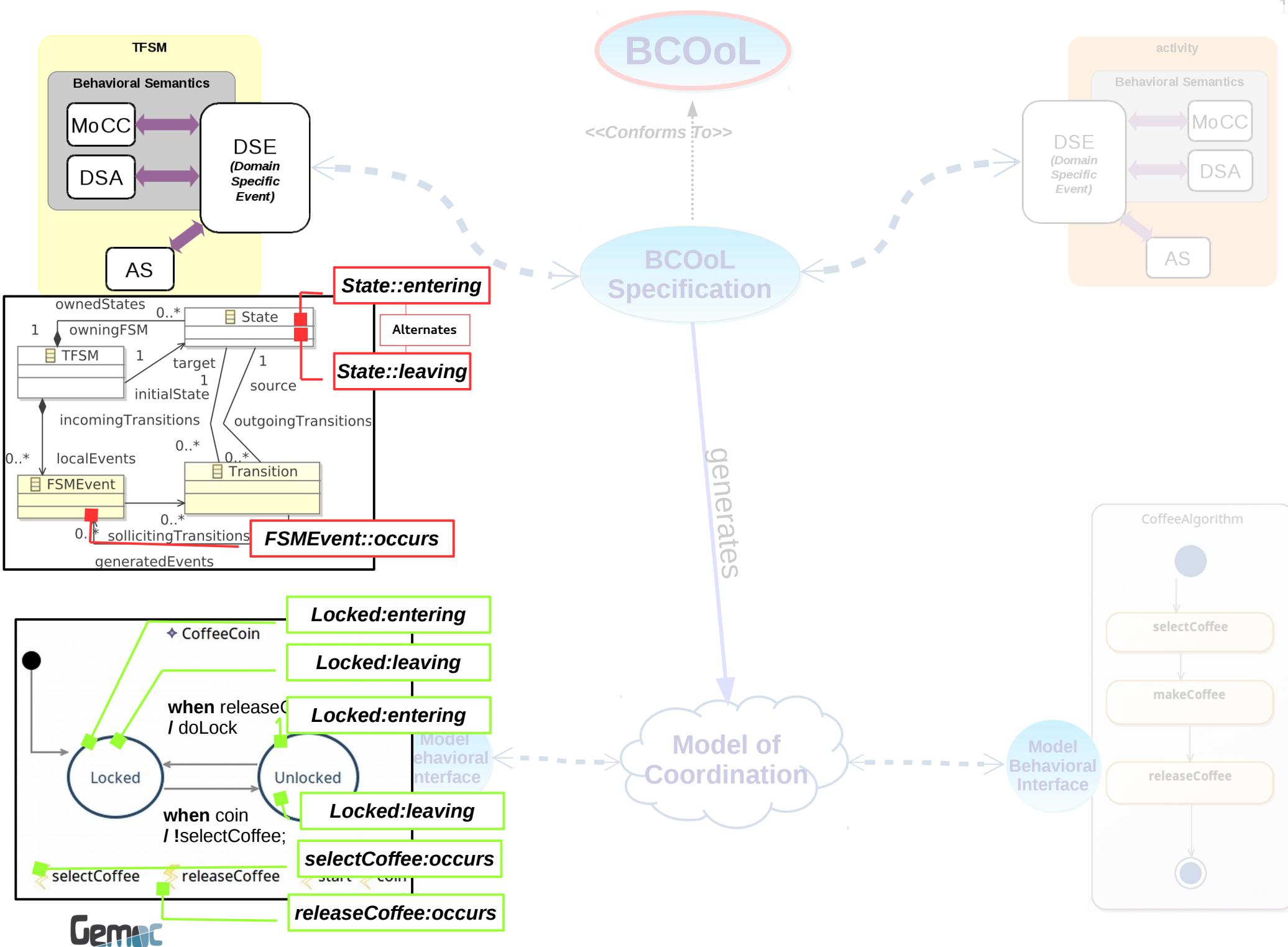


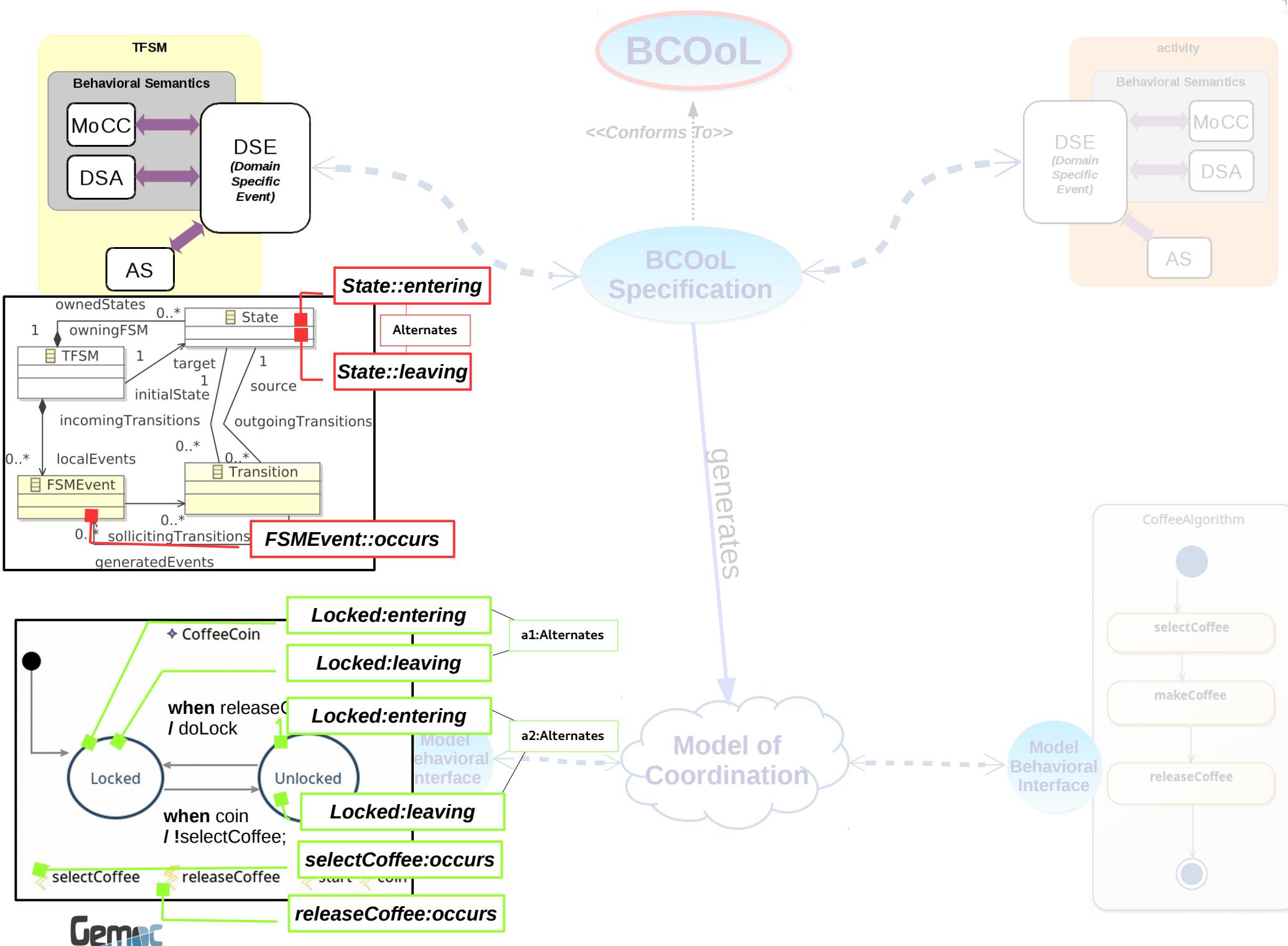


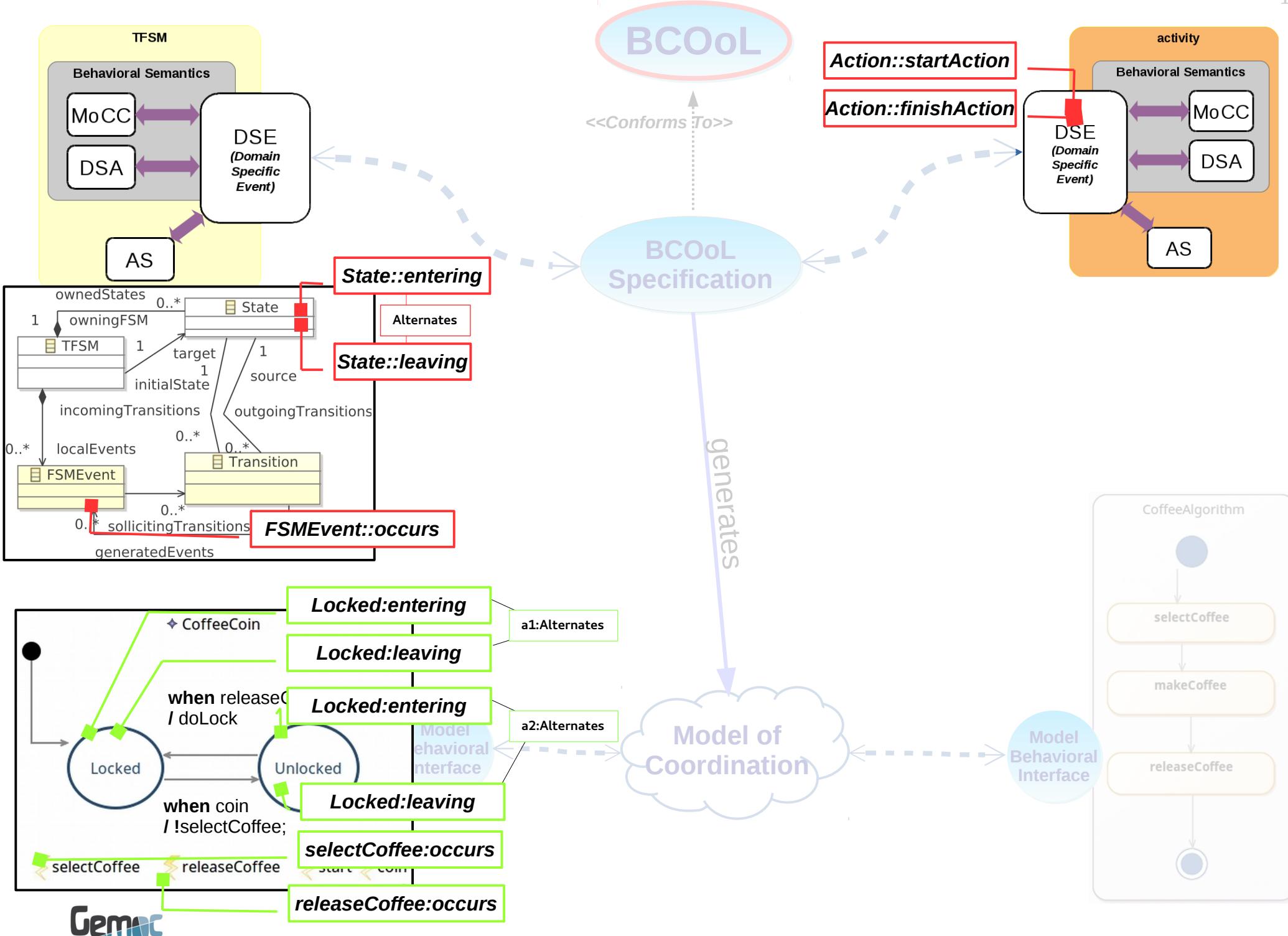


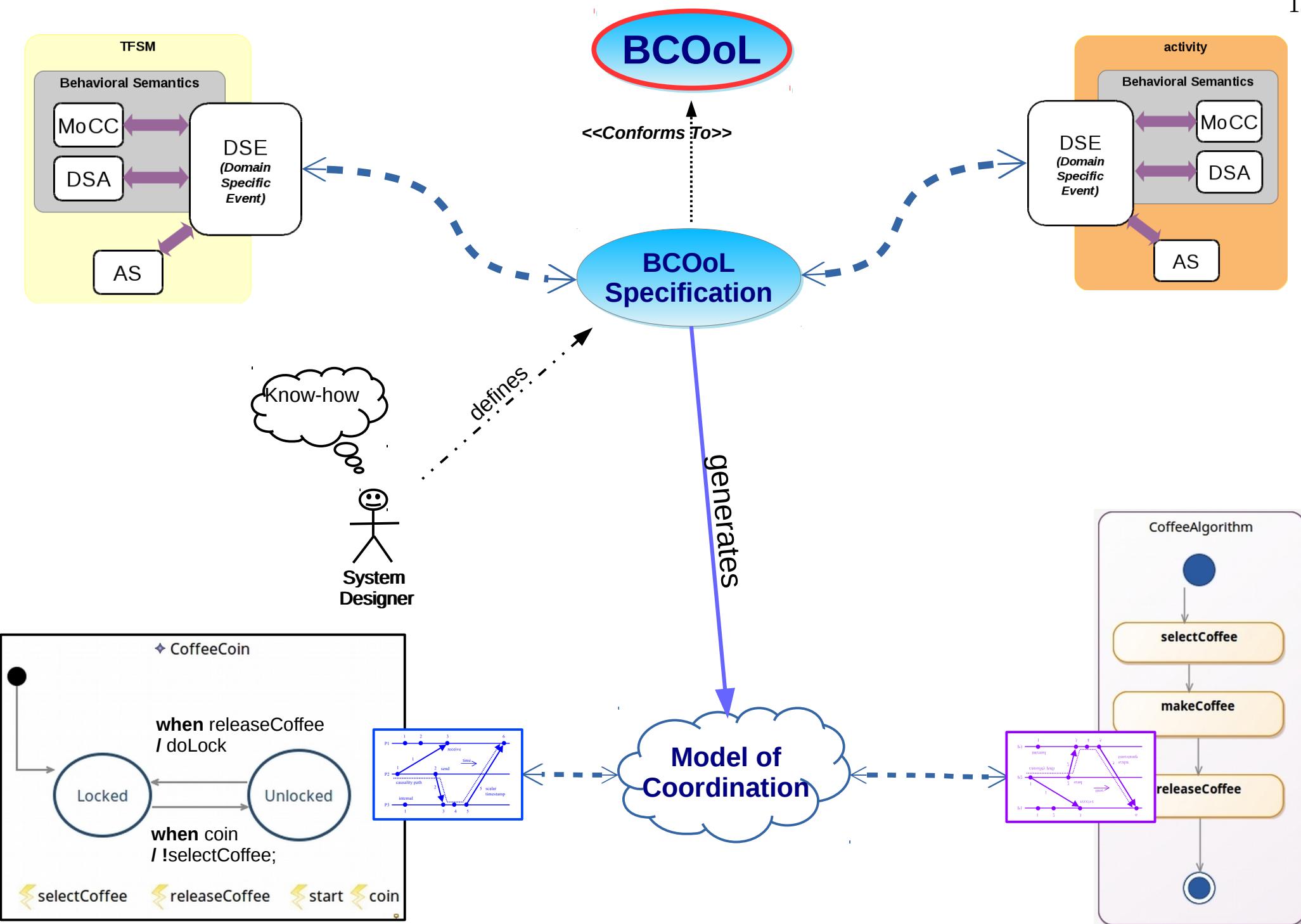


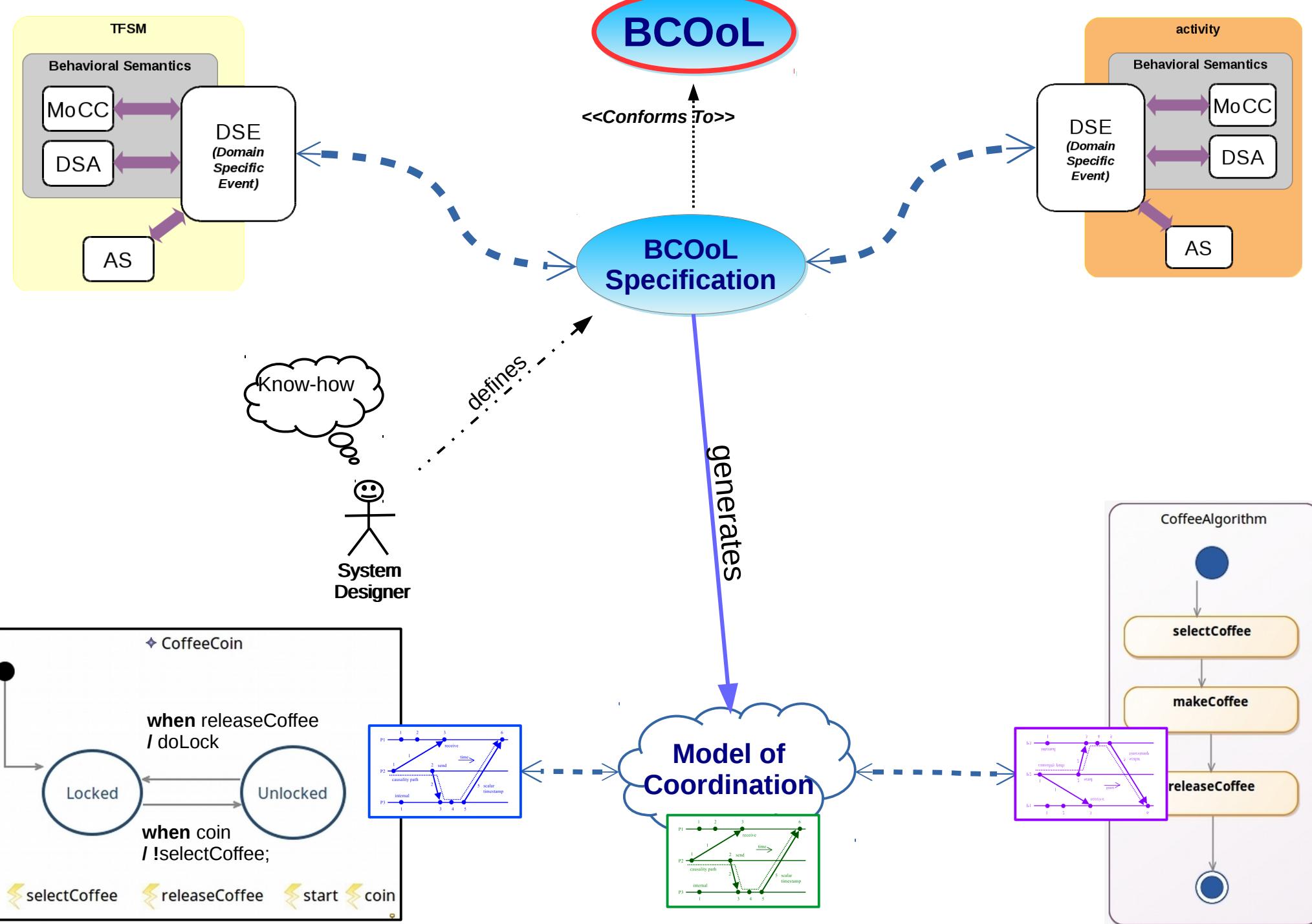


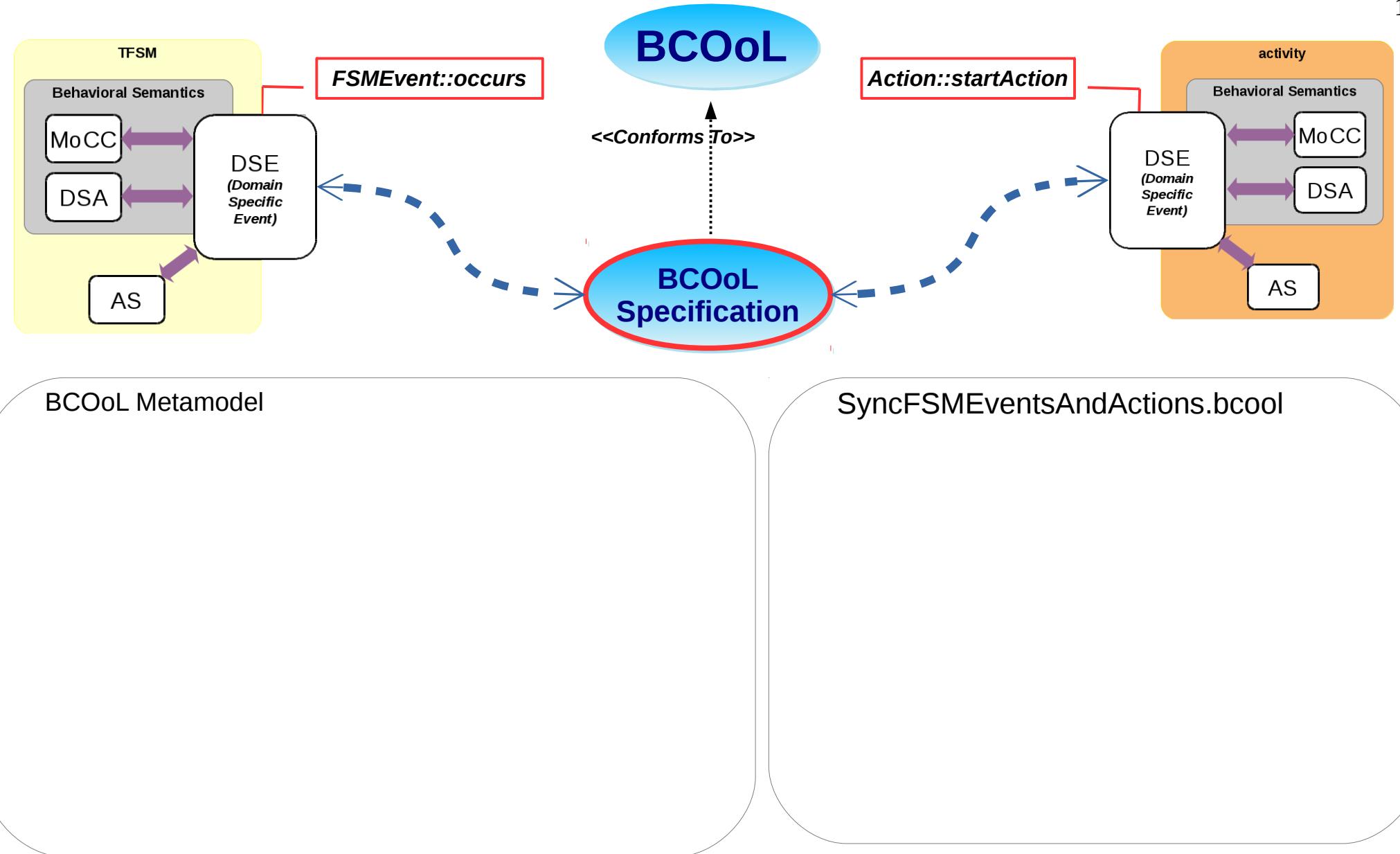


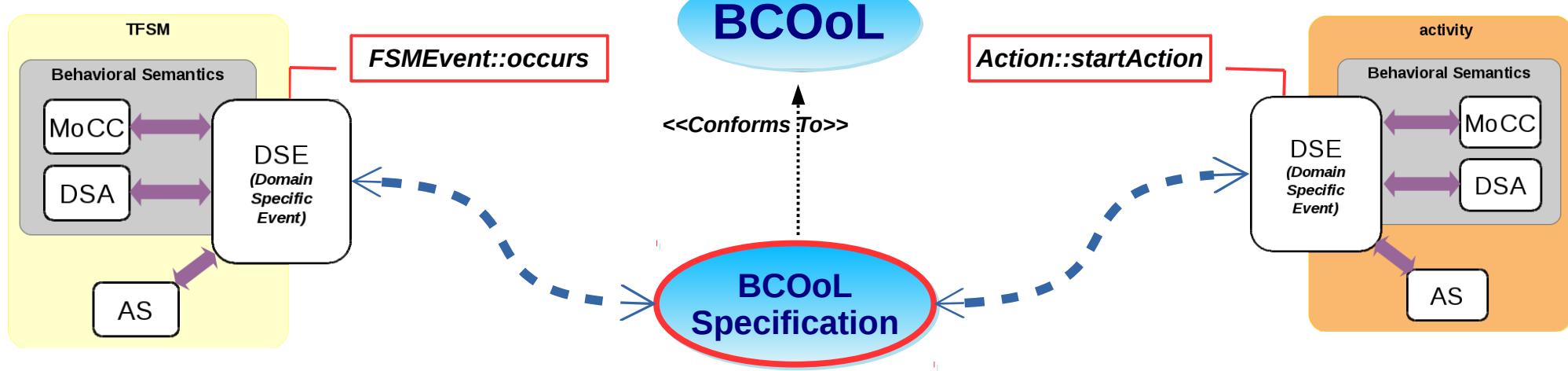




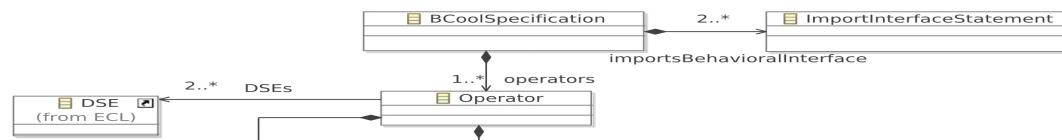








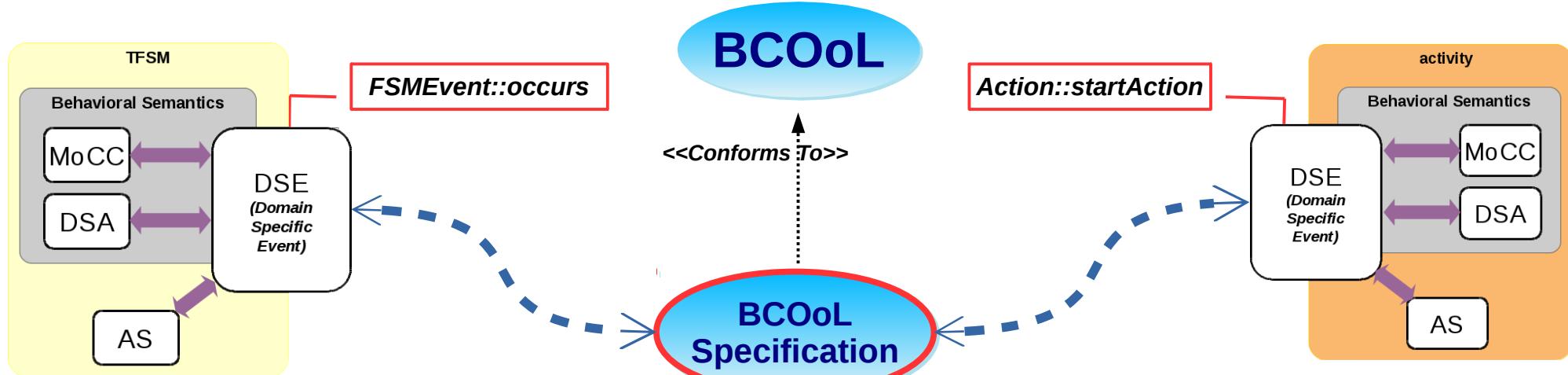
### BCOoL Metamodel



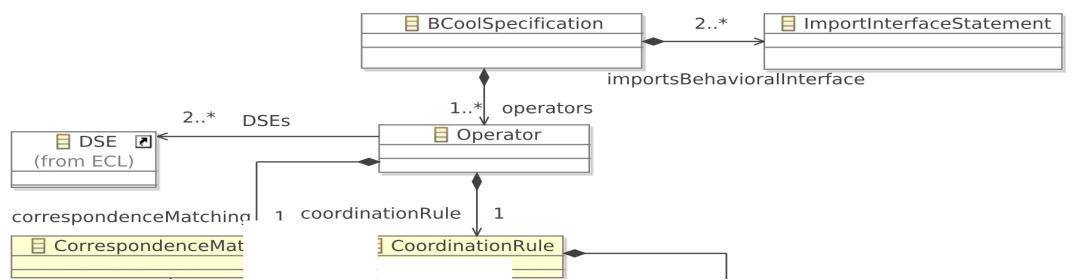
### SyncFSMEventsAndActions.bcool

```

ImportInterface fsm;
ImportInterface Activity;
  
```



### BCOoL Metamodel



### SyncFSMEventsAndActions.bcool

```

ImportInterface fsm;
ImportInterface Activity;

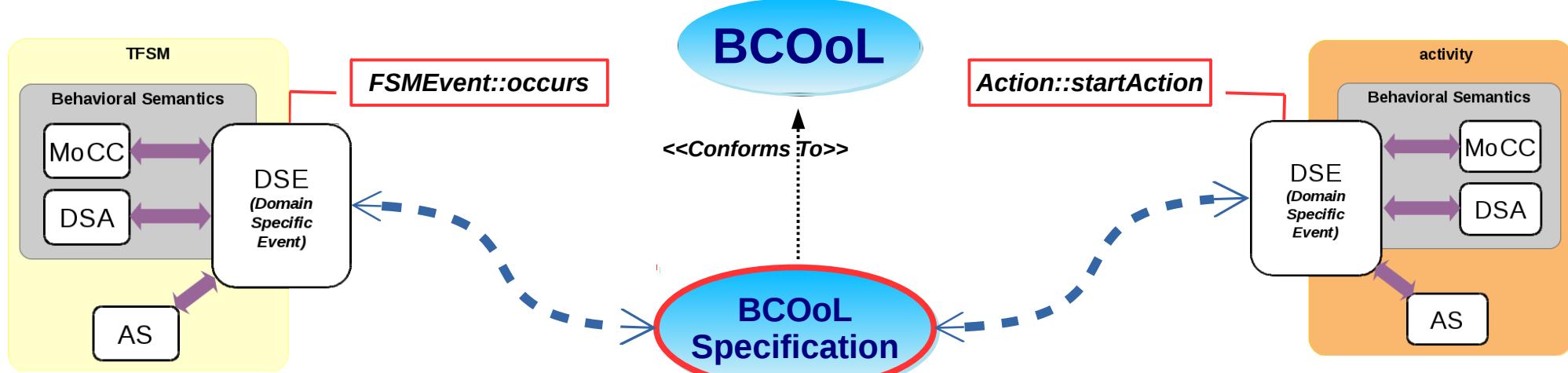
Operator RendezVousWhenSameName
(FSMEvent::occurs, Action::startAction)


```

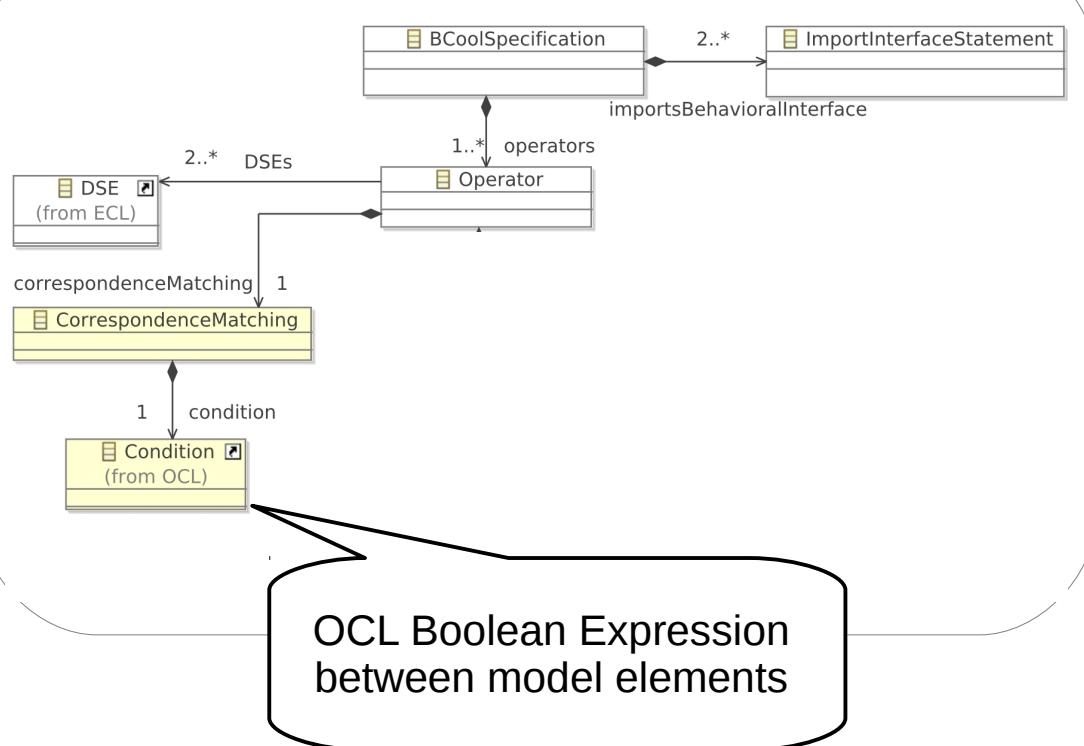
```

End Operator;

```



### BCool Metamodel



### SyncFSMEventsAndActions.bcool

```

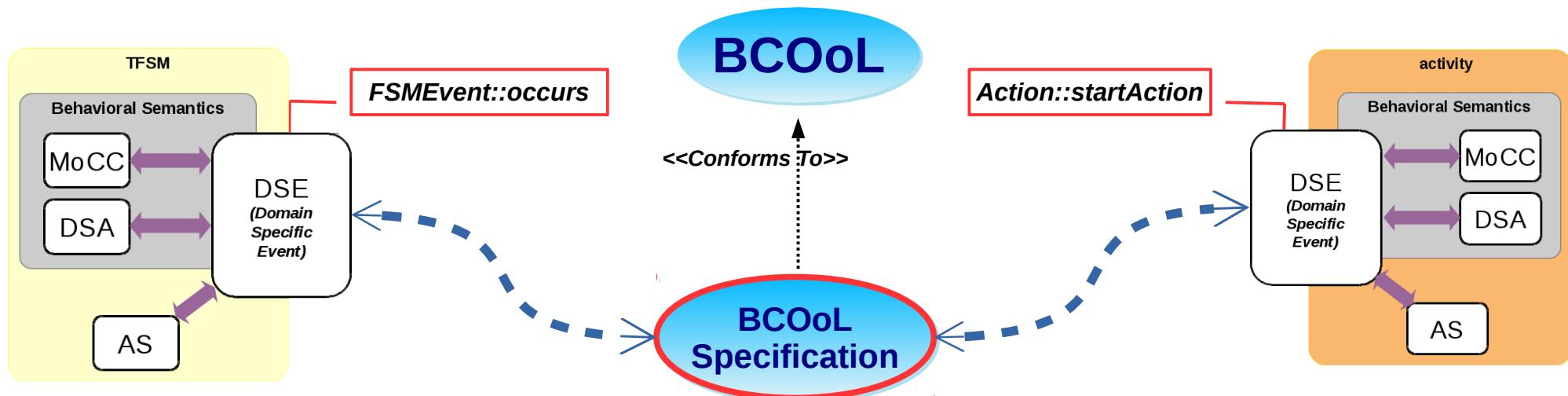
ImportInterface fsm;
ImportInterface Activity;

Operator RendezVousWhenSameName
(FSMEvent::occurs, Action::startAction)

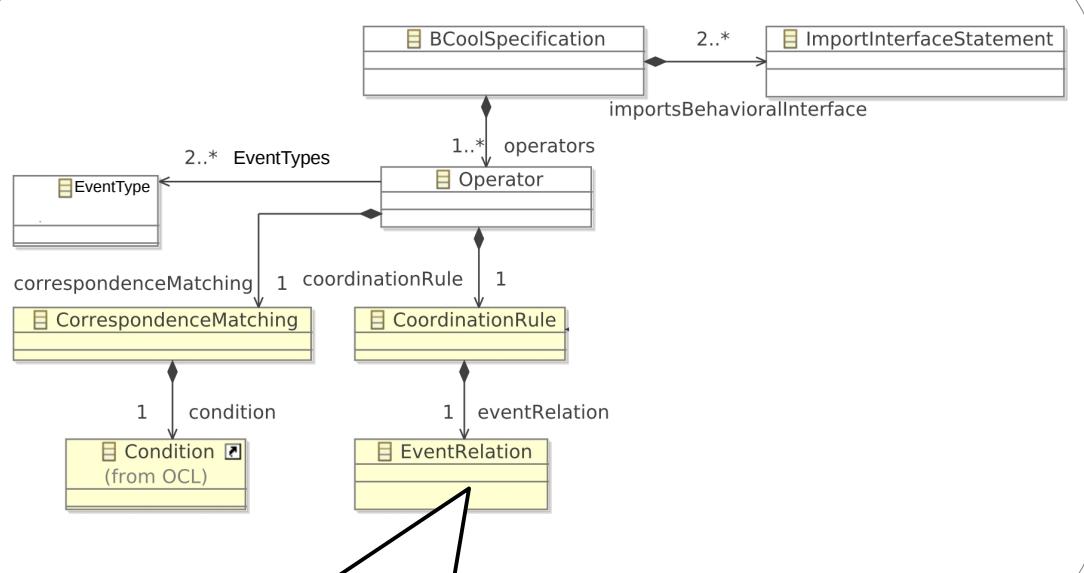
When(occurs.name = startAction.name);

End Operator;

```



### BCOoL Metamodel



Causal and Temporal  
relations between Events  
e.g., Rendezvous,  
Precedes, etc.

### SyncFSMEventsAndActions.bcool

```

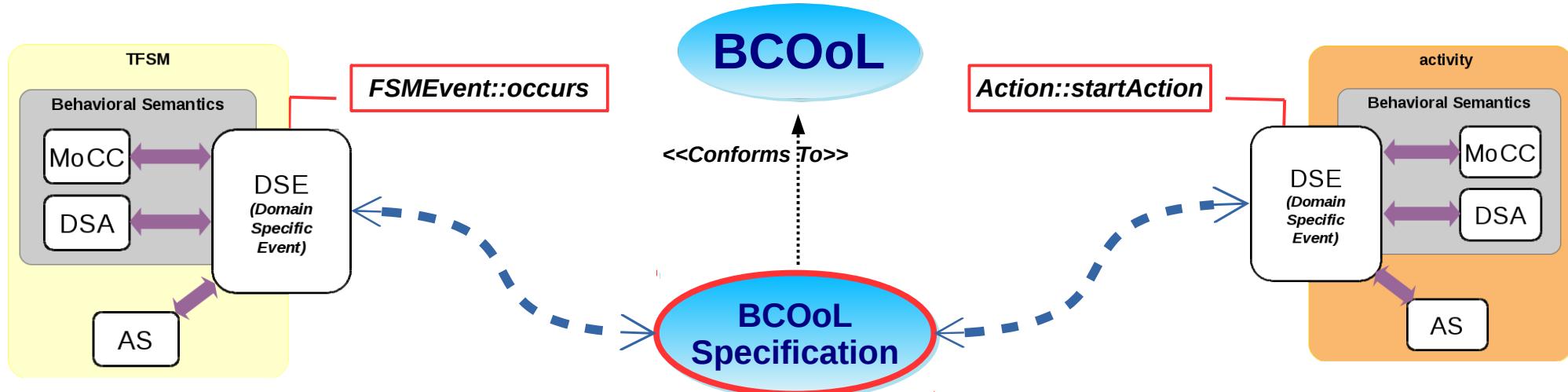
ImportInterface fsm;
ImportInterface Activity;

Operator RendezVousWhenSameName
(FSMEvent::occurs, Action::startAction)

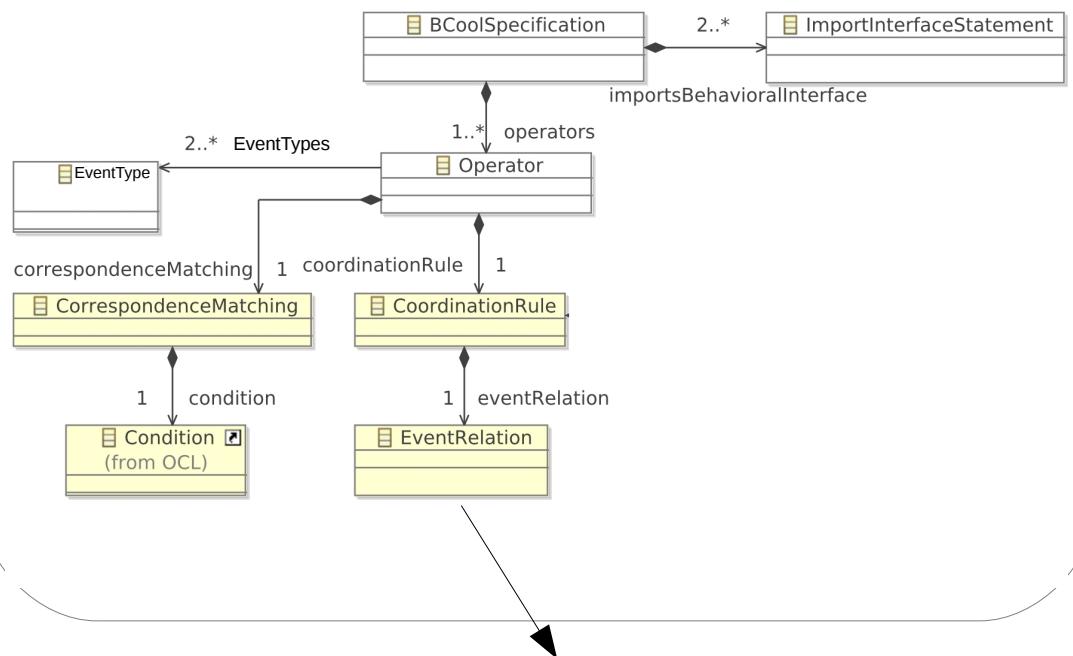
When(occurs.name = startAction.name);

CoordinationRule:
RendezVous (occurs, startAction)
End Operator;

```



### BCOoL Metamodel



Defined in MoCML  
(Model of Concurrency and Communication Modeling Language)

### SyncFSMEventsAndActions.bcool

```

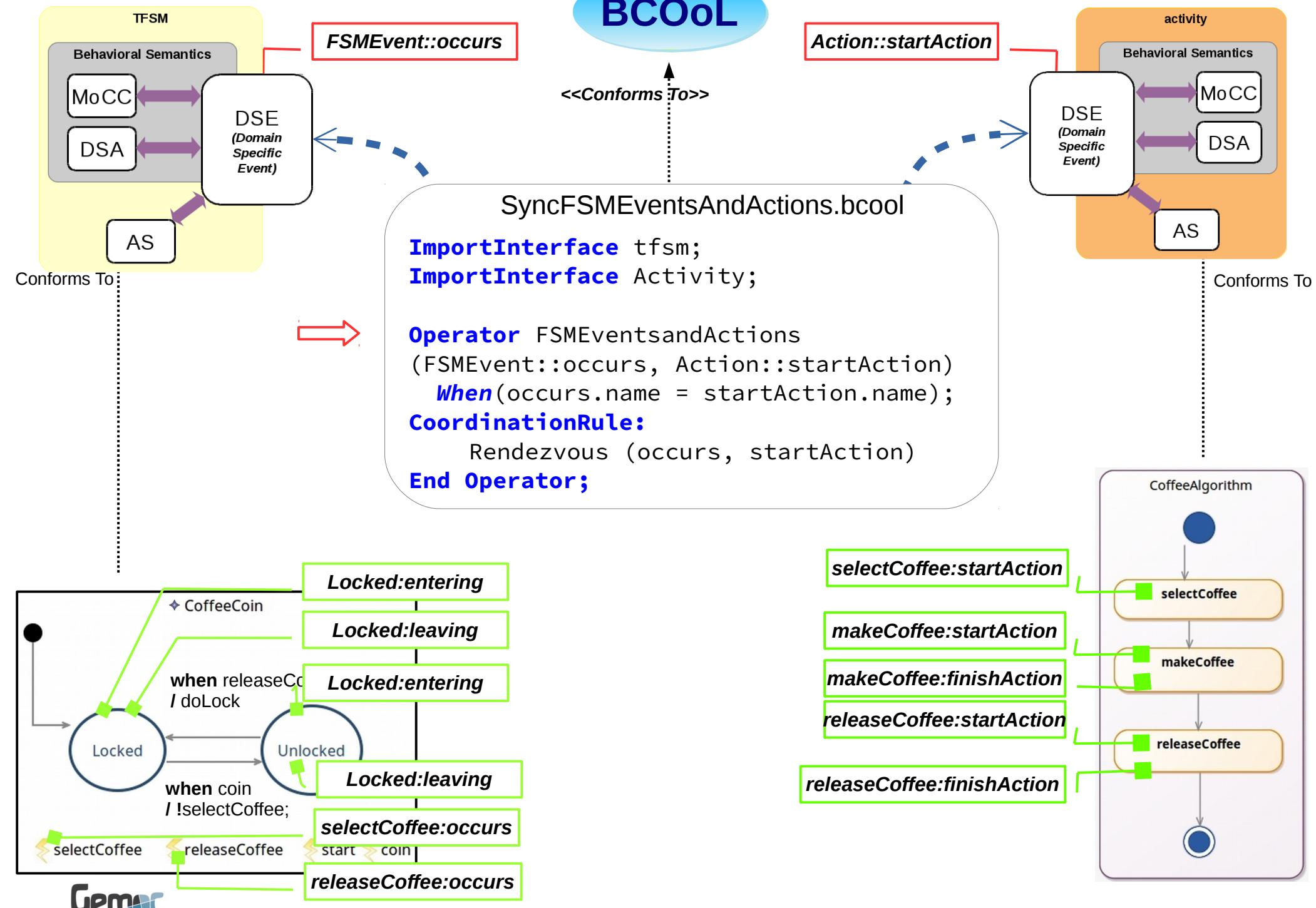
ImportInterface fsm;
ImportInterface Activity;

Operator RendezVousWhenSameName
(FSMEvent::occurs, Action::startAction)

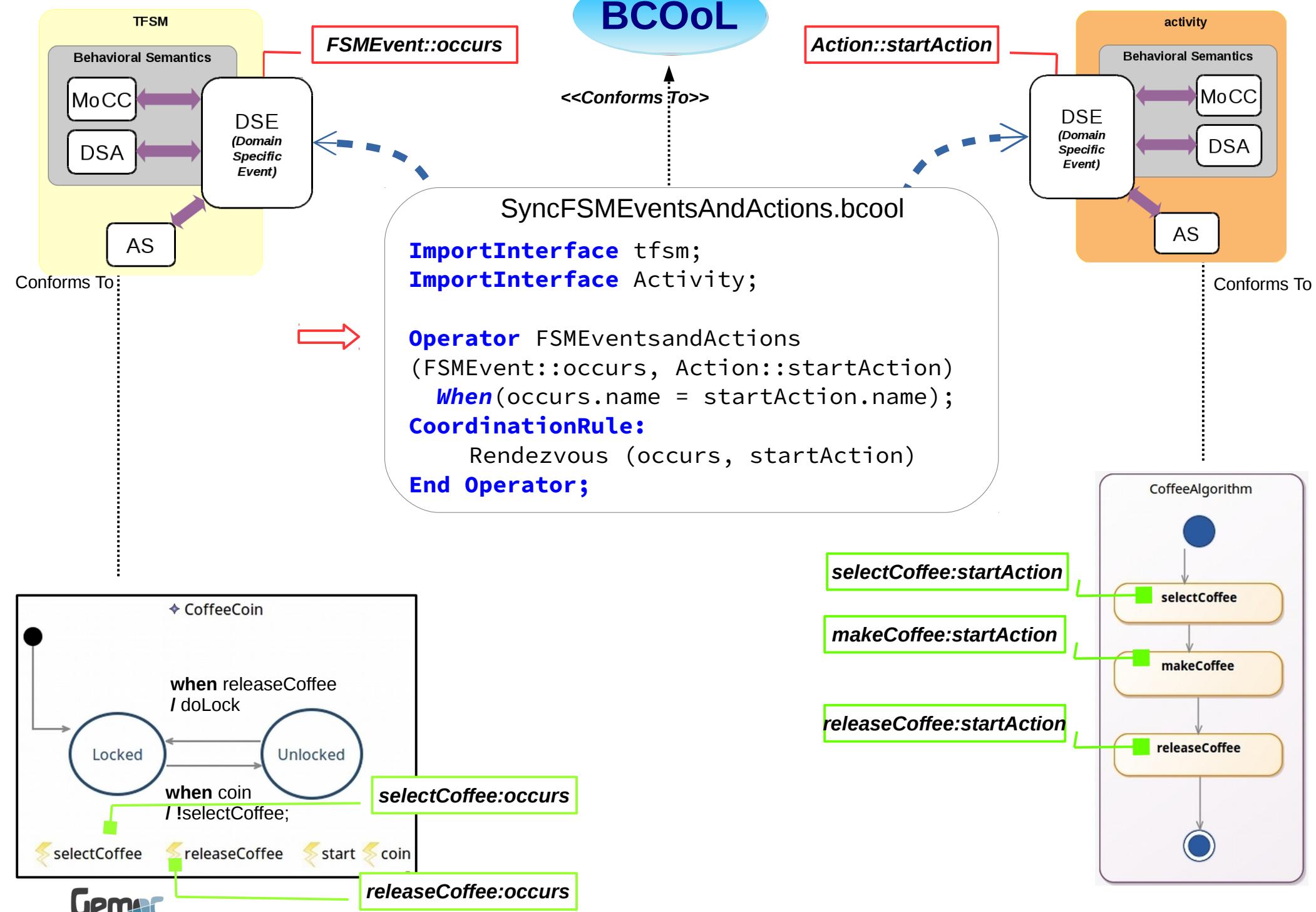
When(occurs.name = startAction.name);

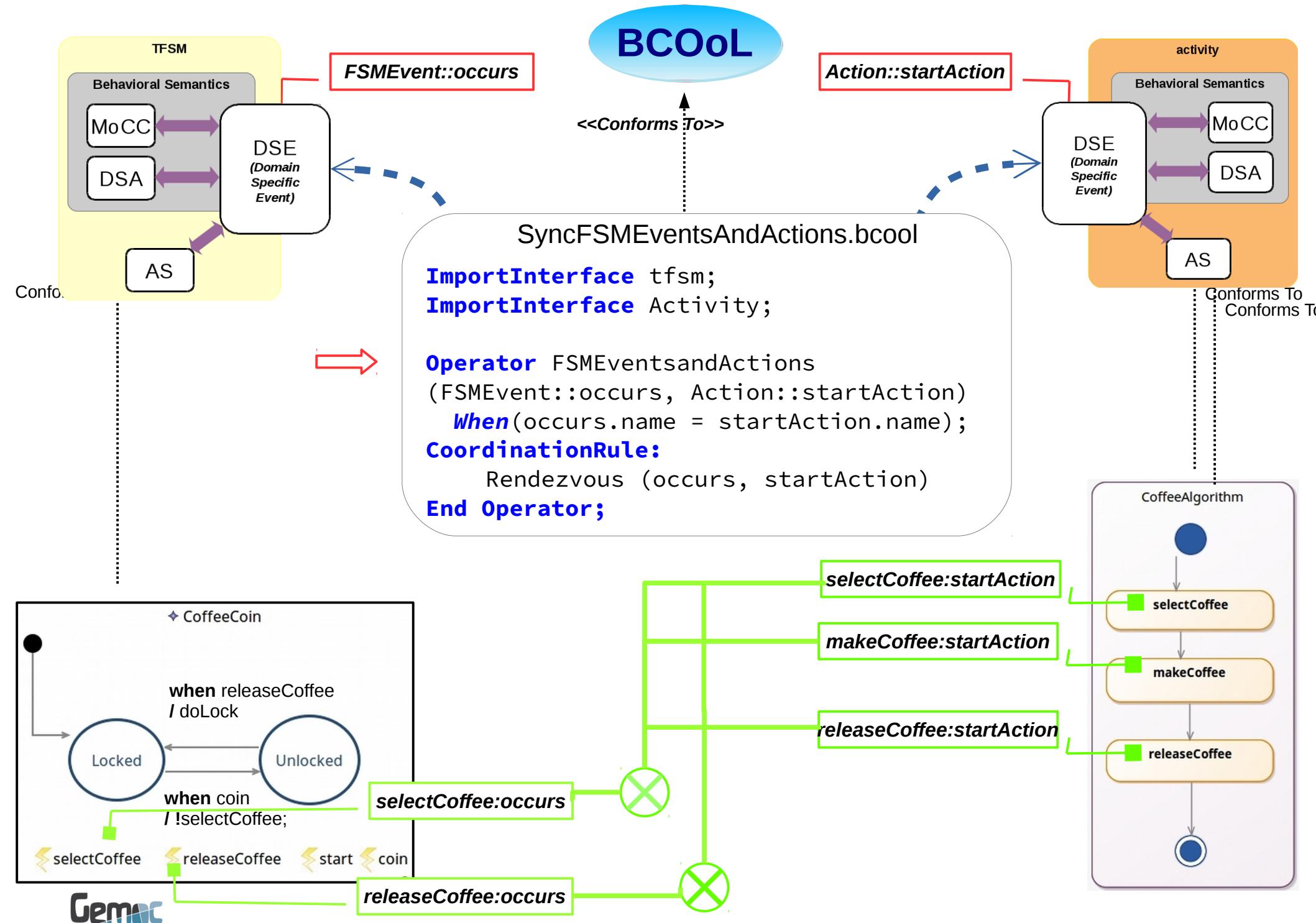
CoordinationRule:
  RendezVous (occurs, startAction)
End Operator;
  
```

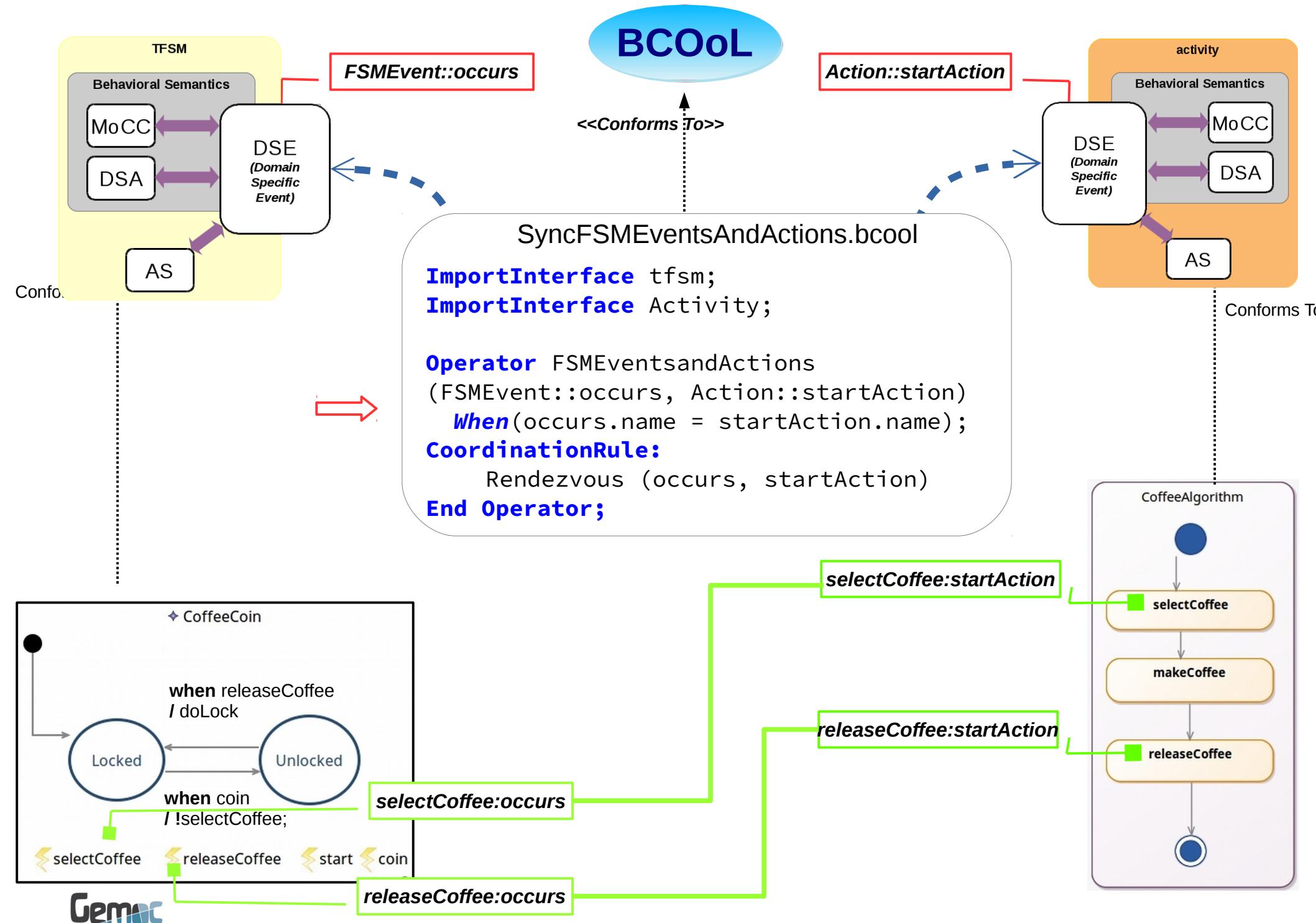
# BCOoL

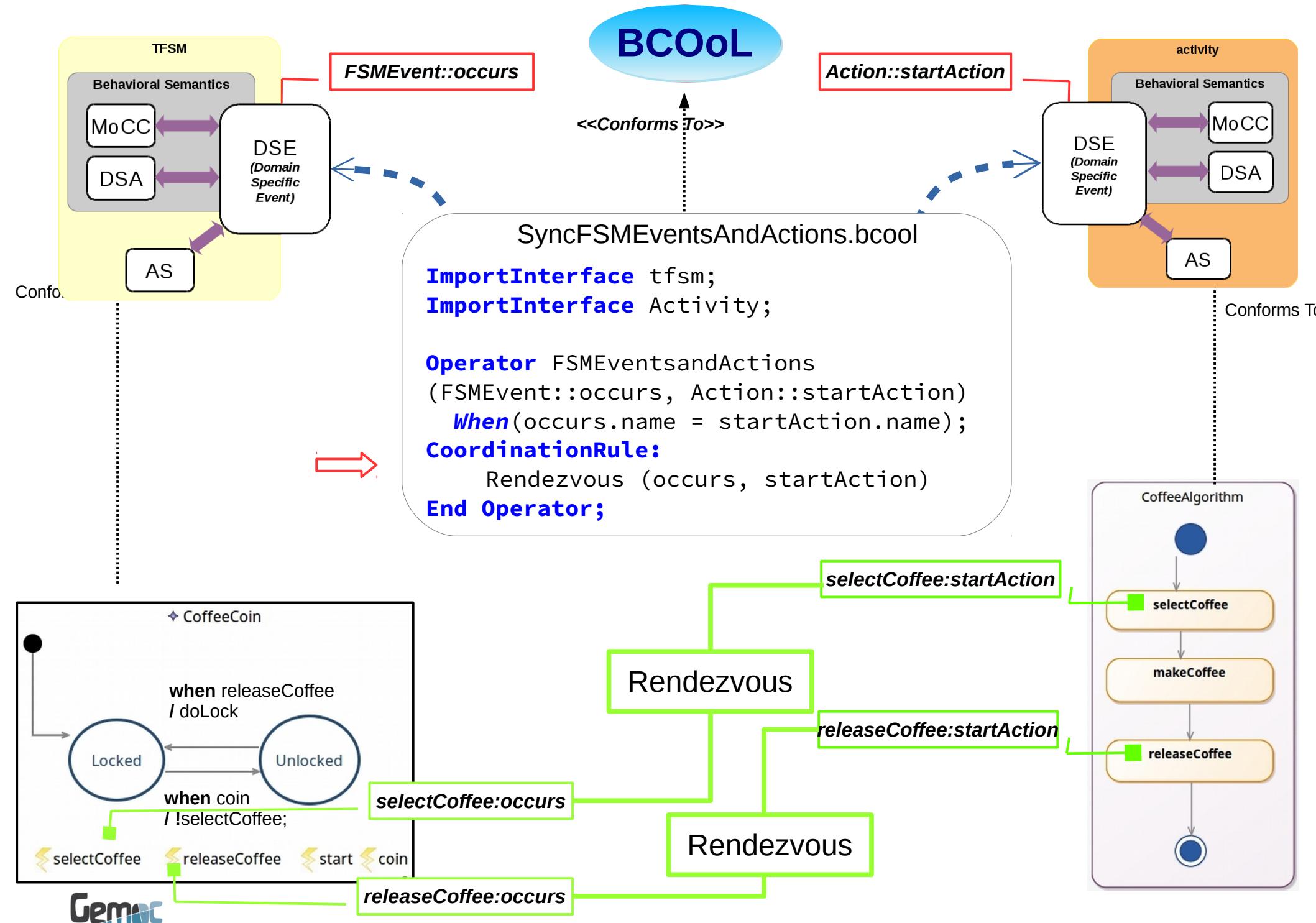


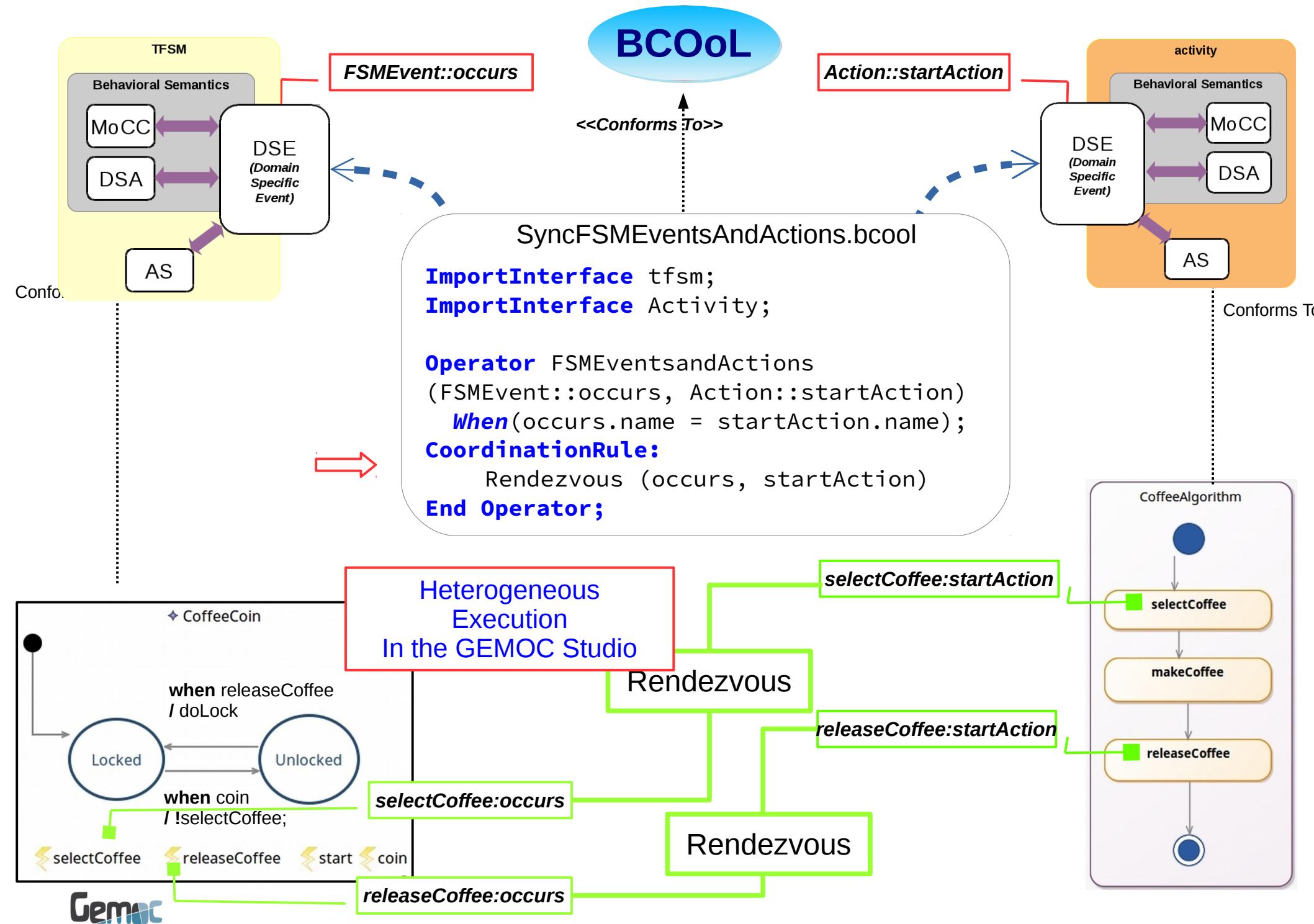
# BCOoL











# Implemented into the GEMOC studio

syncproductfsmwithfuml.bcool

```
SyncProductTfsmwithfUML

ImportLib "platform:/resource/org.gemoc.bcool.example.productfumlandtfsm/operator/facilities.bcoollib"
ImportLib "platform:/resource/org.gemoc.bcool.example.productfumlandtfsm/operator/bcoollib.ccslLib"

ImportInterface "platform:/plugin/org.modelexecution.operationalsemantics.gemoc.ecl/model/ActivityDiagramV2.ecl" as ad
ImportInterface "platform:/plugin/org.gemoc.sample.tfsm.eclmoc2as/ecl/TFSM.ecl" as fsm

@Spec test

@Operator MatchingandCoordinationSharedEvents (dsel : tfsm::occurs, dse2 : ad::executeIt)
@MatchingCorrespondance: when dsel.name = dse2.name ;
    CoordinationRule:
        facilities.RendezVous(dsel, dse2)
end operator;
```

Concurrent Logical Steps Decider

- LogicalStep [903192193]
  - MSE\_Unlocked\_entering
  - MSE\_makeCoffee\_executeIt
- LogicalStep [837141265]
  - MSE\_Unlocked\_entering
- LogicalStep [77004704]
  - MSE\_localclk\_ticks
  - MSE\_makeCoffee\_executeIt
- LogicalStep [482013118]
  - MSE\_localclk\_ticks
- LogicalStep [92429118]
  - MSE\_makeCoffee\_executeIt
- LogicalStep [442559956]
  - MSE\_Unlocked\_entering
  - MSE\_localclk\_ticks
  - MSE\_makeCoffee\_executeIt
- LogicalStep [691567303]
  - MSE\_Unlocked\_entering
  - MSE\_localclk\_ticks

\*CoffeeCoin

```
graph LR; init(( )) --> Locked((Locked)); Locked --> Unlocked((Unlocked)); Unlocked --> Locked; Locked -- "when start / doInit" --> Locked; Locked -- "when releaseCoffee / doLock" --> Unlocked; Unlocked -- "when coin / !selectCoffee;" --> Locked;
```

CoffeeAlgorithm

```
graph TD; start(( )) --> selectCoffee[selectCoffee  
heldTokens = 1]; selectCoffee --> makeCoffee[makeCoffee  
heldTokens = 0]; makeCoffee --> releaseCoffee[releaseCoffee  
heldTokens = 0]; releaseCoffee --> end(( ));
```

Editing facilities by using Xtext

# Implemented into the GEMOC studio

Schedule space exploration

Coordinated Heterogeneous Execution

The screenshot displays several windows from the GEMOC studio:

- SyncProductTfsmwithfuml.bcool**: A code editor window showing UML-like code for synchronization products. It includes imports for "platform:/resource/or" and "platform:/resource/or", and defines interfaces for "platform:/plug" and "platform:/plug". A block of code handles shared events between two states (`dse1` and `dse2`).
- \*computeValueBis Activity Diagram**: An activity diagram titled "CoffeeAlgorithm" with three steps:
  - selectCoffee**: heldTokens = 1
  - makeCoffee**: heldTokens = 0
  - releaseCoffee**: heldTokens = 0
- \*CoffeeCoin**: An activity diagram showing a state transition from "Locked" to "Unlocked". Transitions are triggered by "when start / doInit", "when releaseCoffee / doLock", and "when coin / !selectCoffee;".
- Logical Steps Decider**: A tree view of logical steps, many of which involve "MSE\_Unlocked\_entering" or "MSE\_makeCoffee\_executeIt".
- Schedule space exploration**: A large, complex state transition graph with many states and numerous transitions, enclosed in a red border.
- Coordinated Heterogeneous Execution**: A state transition graph with multiple parallel regions, also enclosed in a red border.

# Conclusion

- BCOoL is a dedicated metalanguage to capture coordination patterns.
  - It automates the coordination of models by relying on a formal language.
  - It is associated to the GEMOC language/modeling workbench to execute and analyze the coordinated system.
- Future work:
- Using the explicit coordination to generate master on co-simulation bus
  - Understanding the interconnection with physical model (continuous time)

<sup>1</sup><http://timesquare.inria.fr/BCOoL>

# Thanks

<http://timesquare.inria.fr/BCOoL>

<http://gemoc.org/ins>



**BCOOL**