

# Evaluation and First Implementation of ETCS Model

Institut Telecom

# Outline

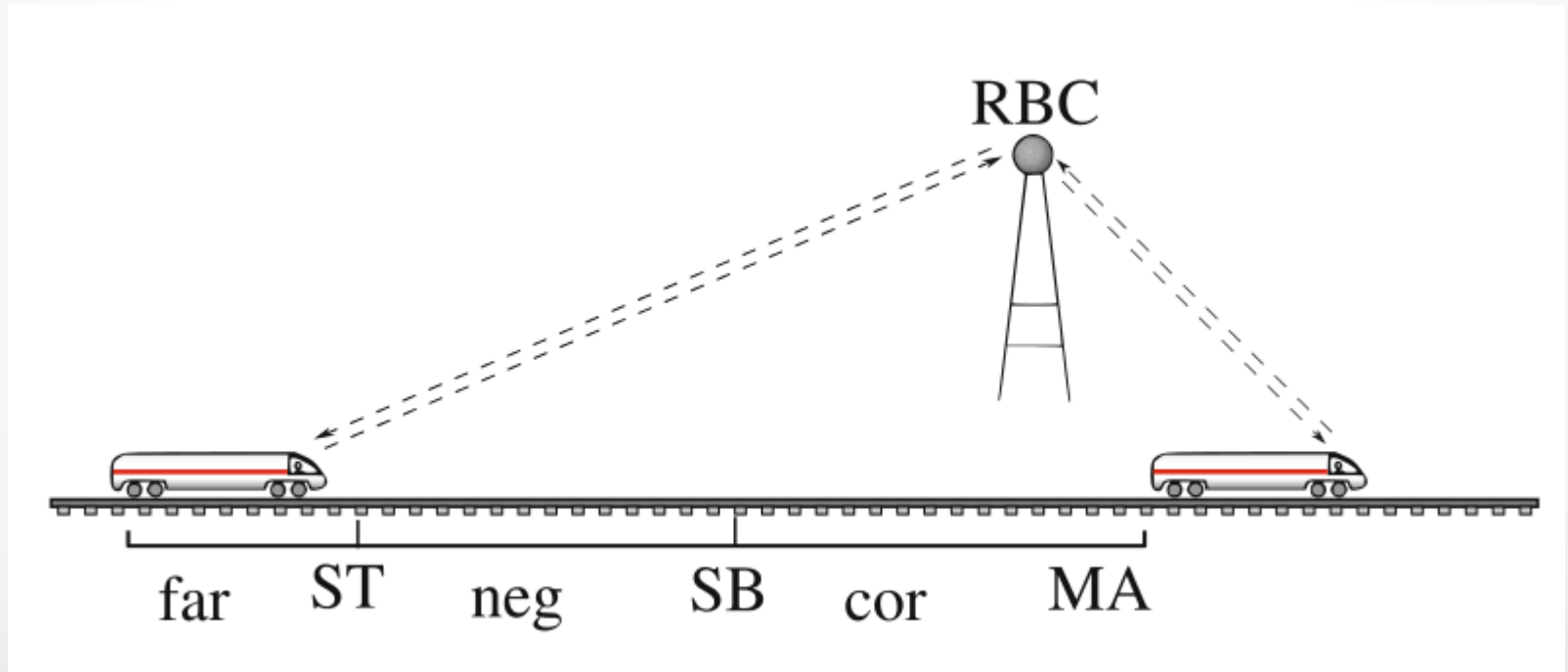
1. Formal ETCS model
2. Simulation

# Formal Model

- Translated from ETCS requirements
- Formalized using Finite State Machines
- Use of timeouts for temporal constraints

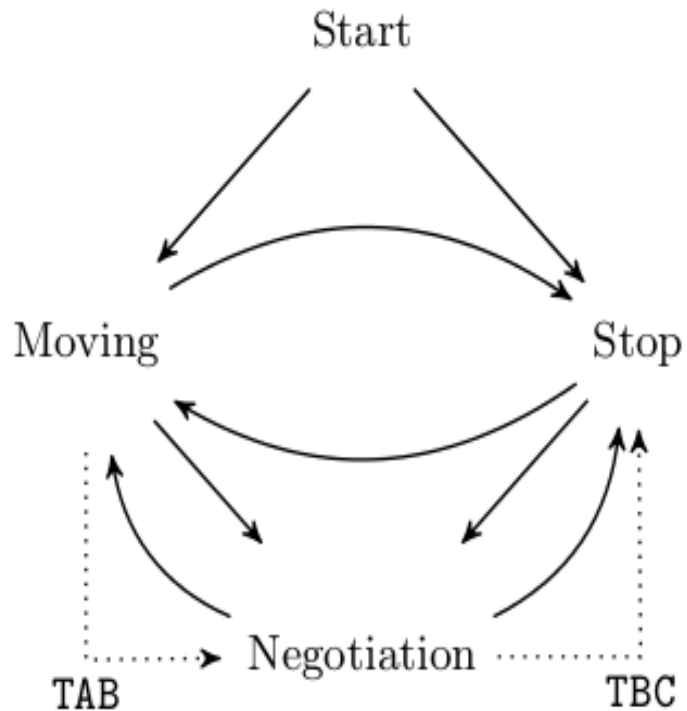
# Roles

- Train
- RBC: Radio Block Center

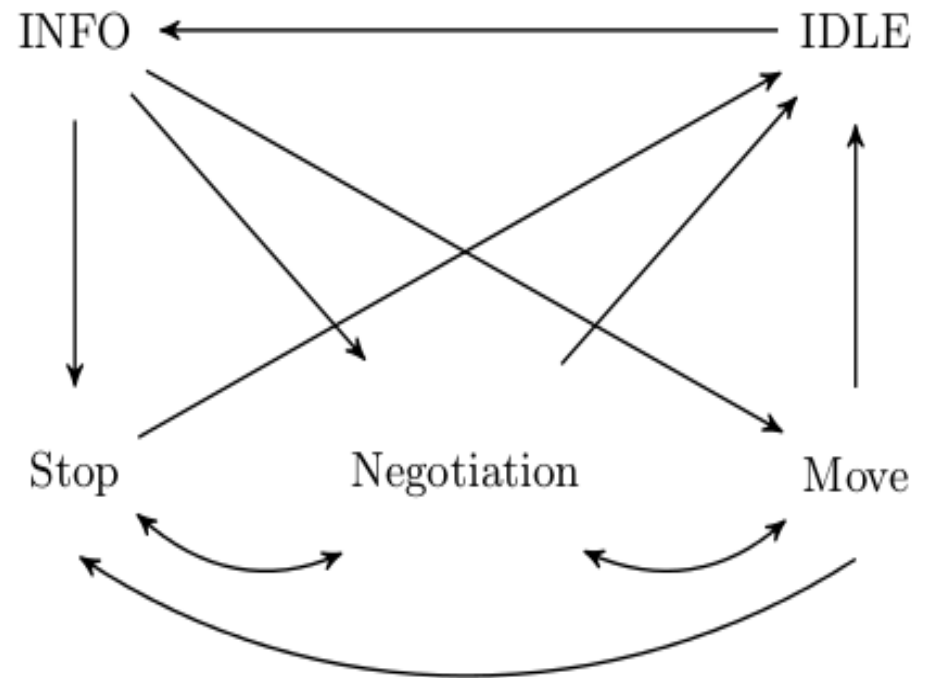


# Behaviour of a train / RBC

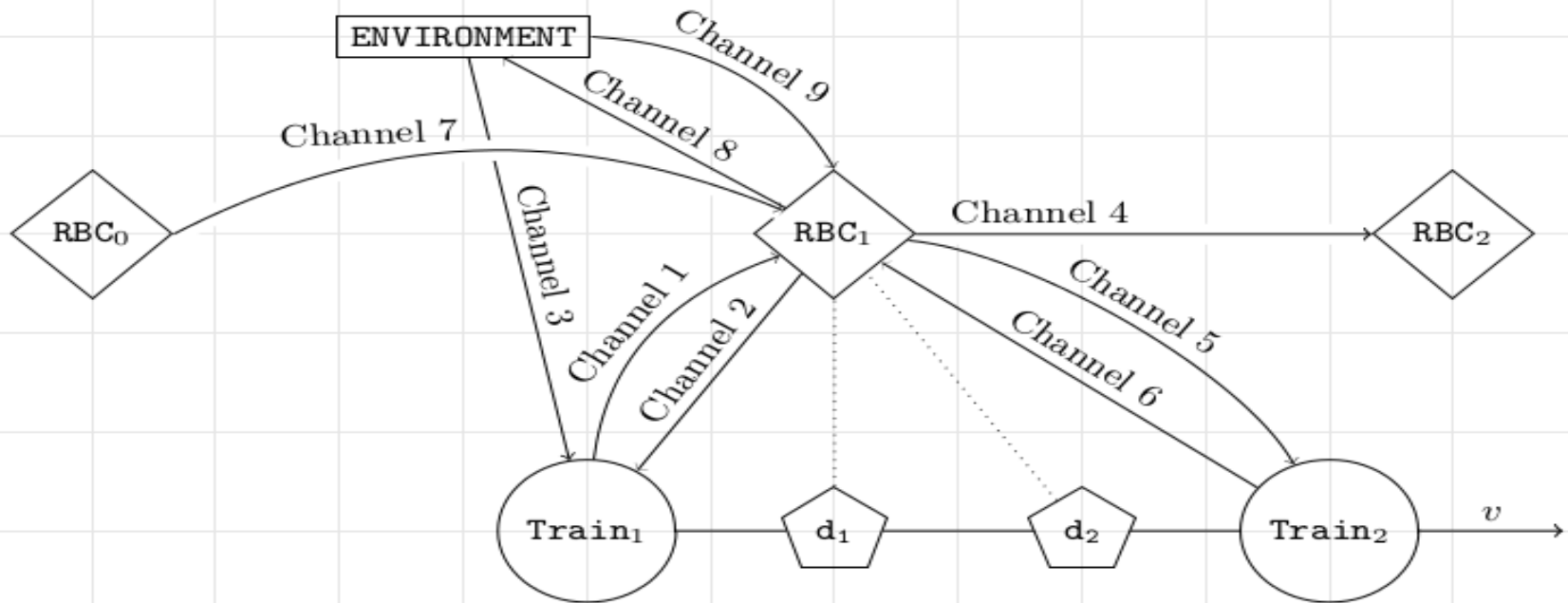
Simple scheme train system



Simple scheme RBC system



# Communication



# Simulation

Three steps:

- Represent the FSM in an XML file
- Parse the XML file and create a FSM automatically
- Execute lists of commands (inputs or pauses)

# XML File

<model>

<state id="s0">

<transition dest="s1" input="a" output="z" time="1"></transition>

</state>

<state id="s1">

<transition dest="s2" input="a" output="y" time="5"></transition>

<timeout dest="s3" time="5">

</state>

<state id="s2">

<transition dest="s0" input="b" output="m" time="3"></transition>

<transition dest="s2" input="a" output="z" time="3"></transition>

</state>

<state id="s3">

<transition dest="s2" input="c" output="n" time="3"></transition>

</state>

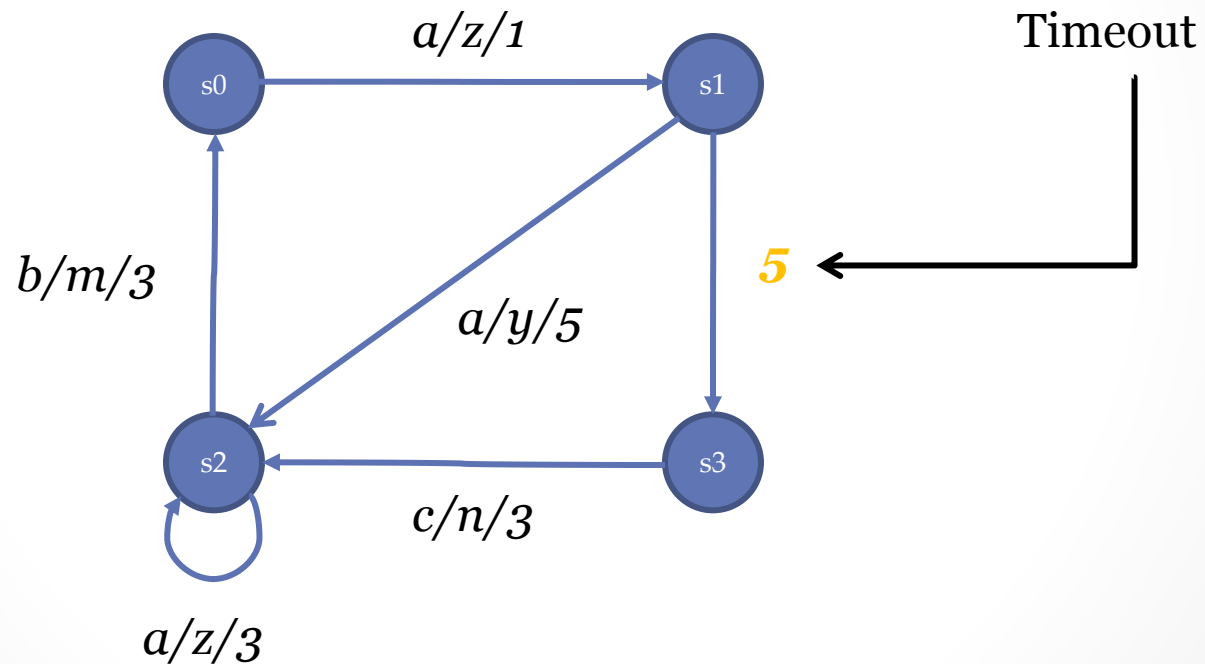
</model>



# Generation of the FSM

- XML file as the input for parsing
- During the parsing, the different components (states, transitions, etc) are created
- Once the parsing is complete, the FSM is ready to run lists of commands (tests)

# Generation of the FSM



# Execution

Using the generated FSM and a list of commands, it is possible to execute the FSM and view the expected outputs of the model.

Commands can be of two types:

- Input
- Pause

# Execution

Information obtained:

- States visited
- Outputs
- Information about timeouts

# Contribution

- Verify that the model has the expected behaviour
- Help in the creation of test cases for the implementation
- Aid in debugging of the implementation (by using techniques like Model Based Diagnosis)

Thank you for your attention