Performance Evaluation of ?? Executable Domain-Specific Languages

Projet RODIC - RAPID RECONFIGURATION OF MANUFACTURING SYSTEMS : A MODEL-BASED SOFTWARE ENGINEERING AND HUMAN INTERACTION COUPLED APPROACH

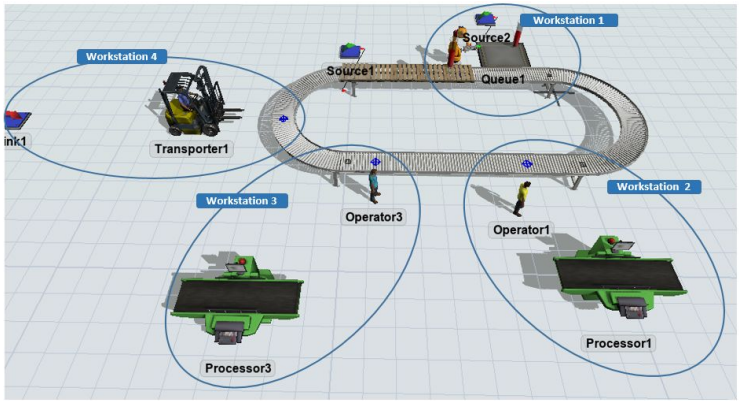
OutLine

General Context

Rapid Reconfiguration of Manufacturing Systems

The operator of the industry needs to :

- interact himself with the simulation,
- verify the correctness of the new configuration,
- estimate its performance before the deployment.



Introduction - RODIC Project

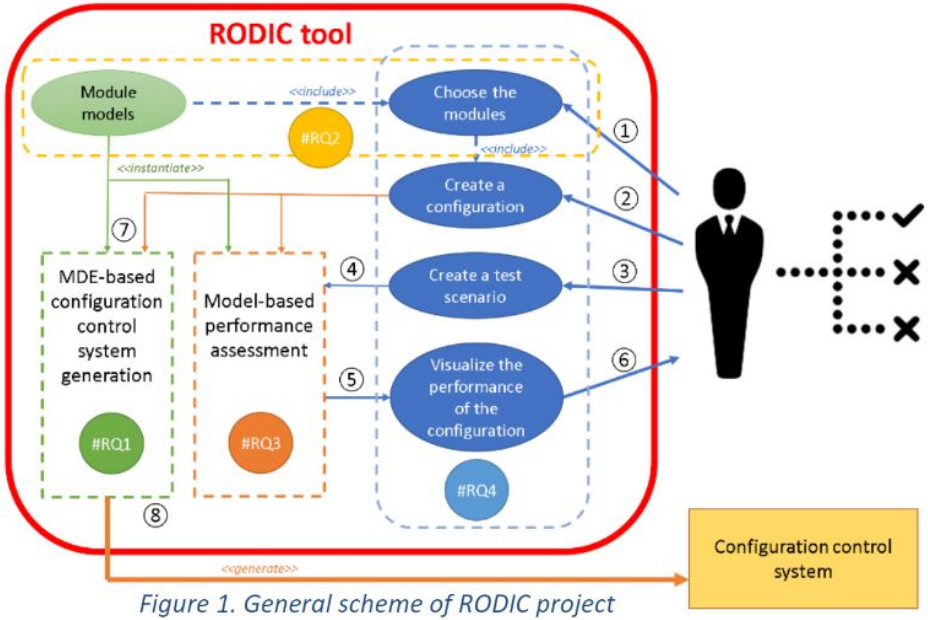


Figure 1. General scheme of RODIC project

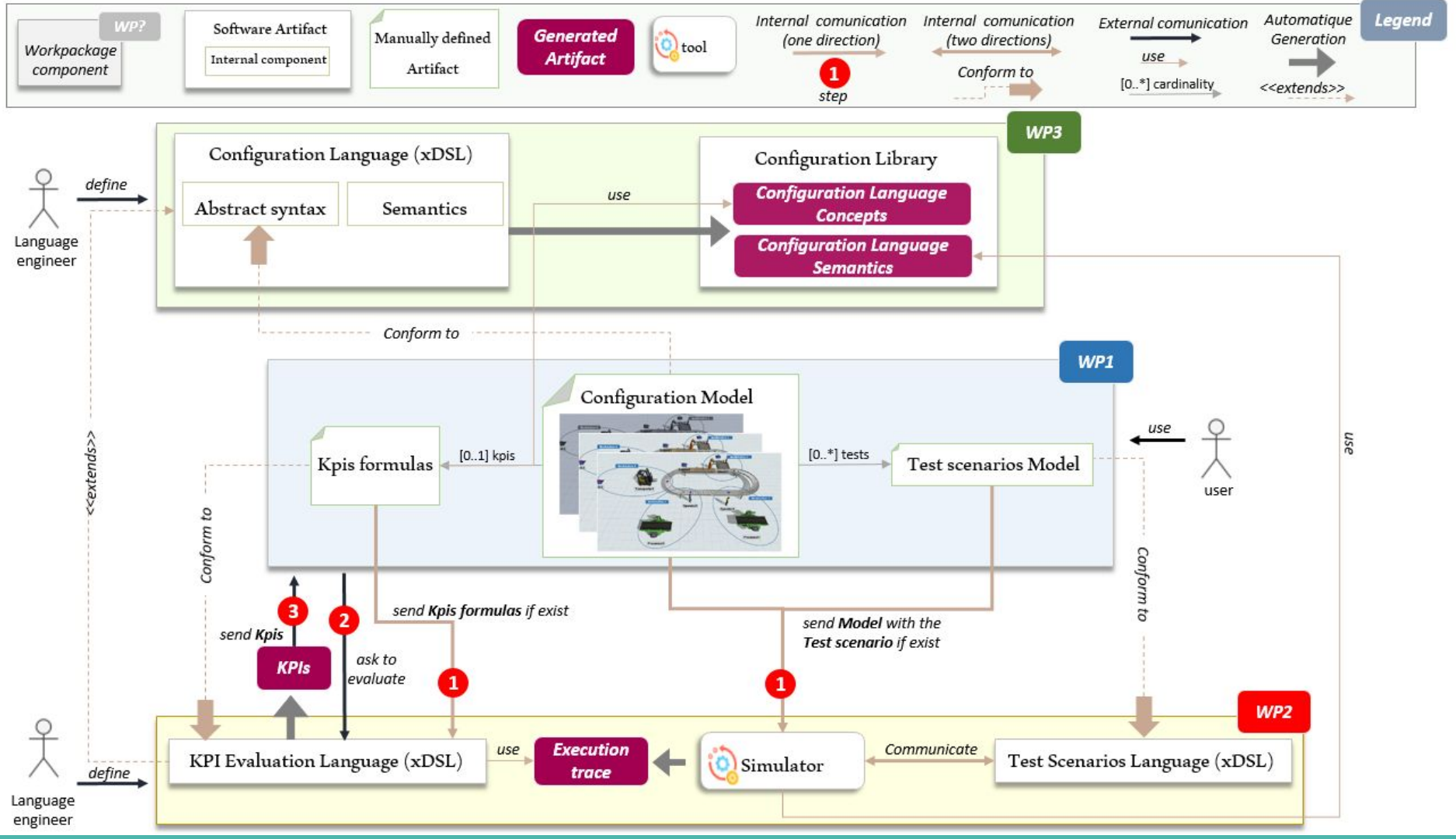
RODIC Approach

WP1: Construction of an interactive human-machine interface. (blue)

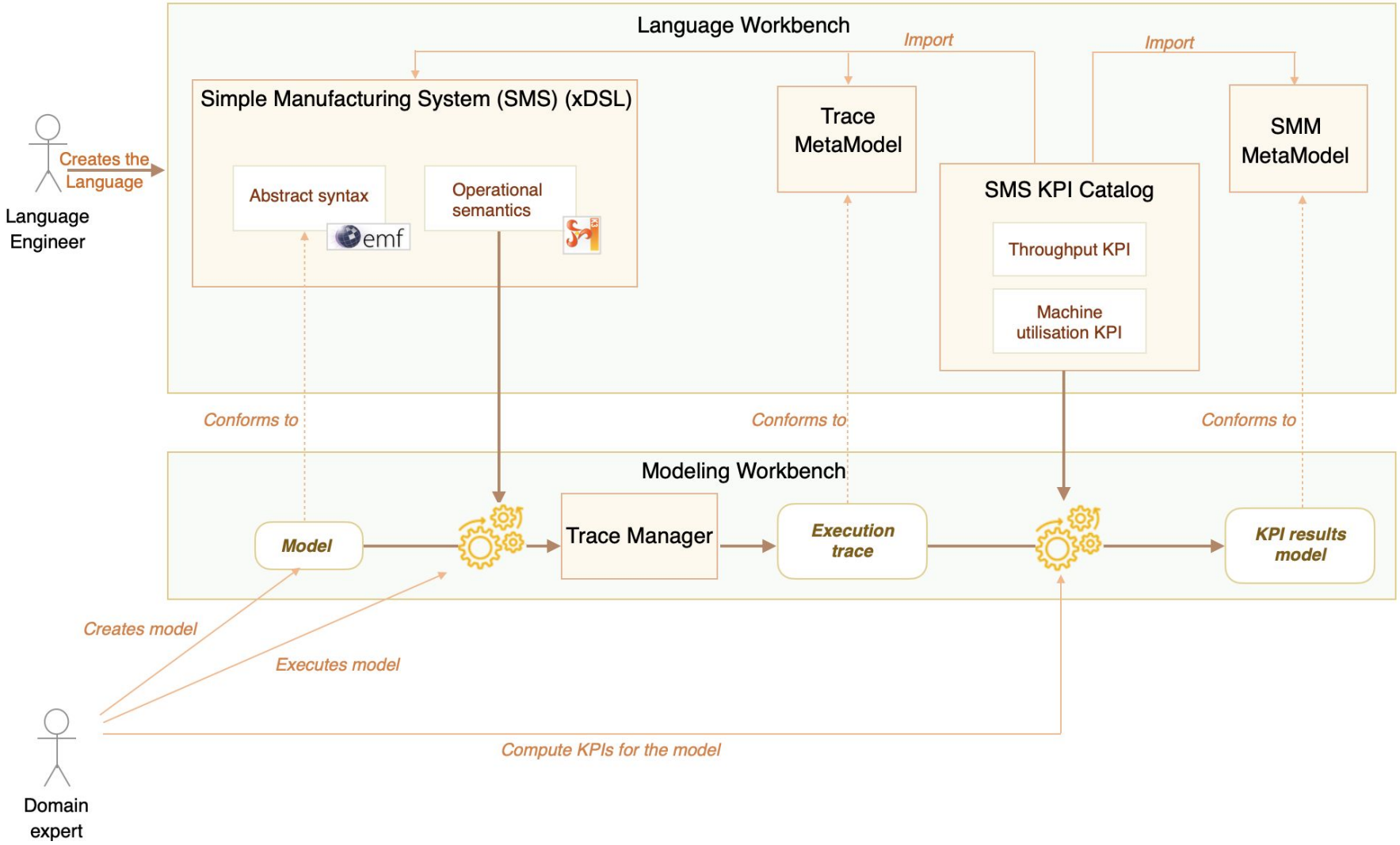
WP2: Construction of an interactive human-machine interface. (orange)

WP3: Design of configuration models. (green)

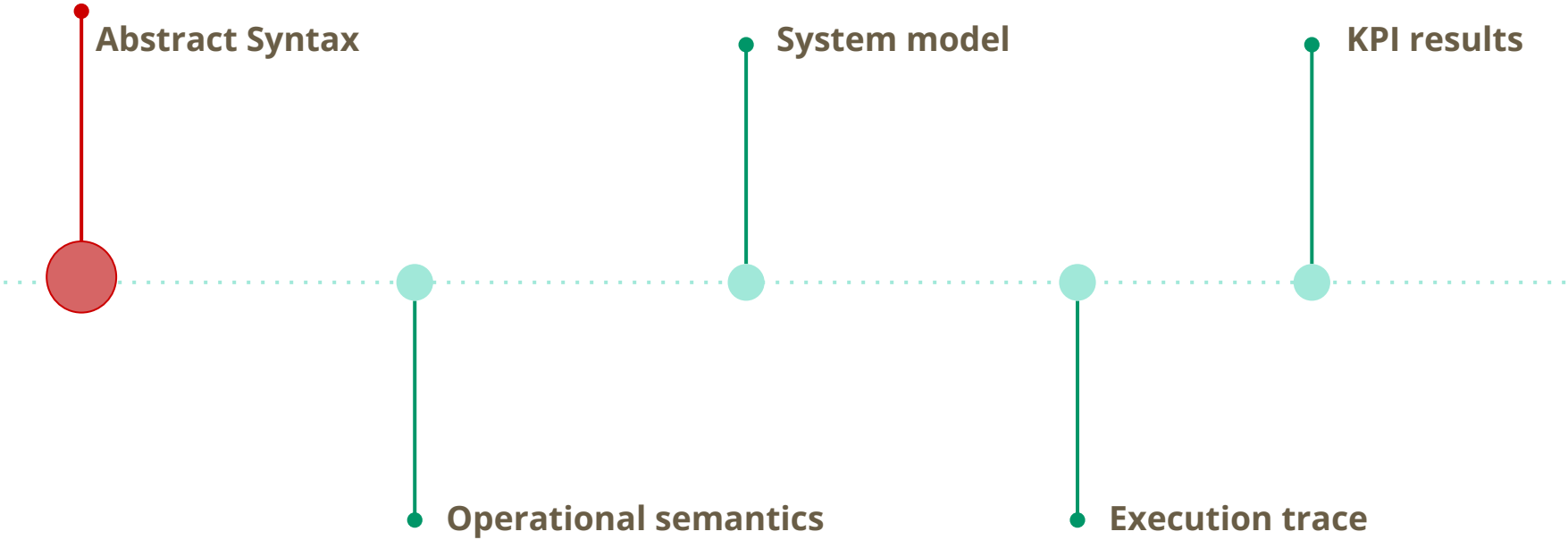
Overall architecture diagram of RODIC


WP?
Workpackage component
Software Artifact
Internal component
Manually defined Artifact
Generated Artifact
tool
Internal comunication (one direction)
1
step
Internal comunication (two directions)
Conform to
External comunication
use
[0..*] cardinality
Automatique Generation
<<extends>>
Legend
WP3
Configuration Language (xDSL)
Abstract syntax
Semantics
Configuration Library
Configuration Language Concepts
Configuration Language Semantics
define
Language engineer
use
Conform to
WP1
Configuration Model
Kpis formulas
[0..1] kpis
[0..*] tests
Test scenarios Model
use
user
<<extends>>
Conform to
Conform to
use
send Kpis
3
KPIs
2
ask to evaluate
send Kpis formulas if exist
1
send Model with the Test scenario if exist
1
WP2
define
Language engineer
KPI Evaluation Language (xDSL)
use
Execution trace
Simulator
Communicate
Test Scenarios Language (xDSL)

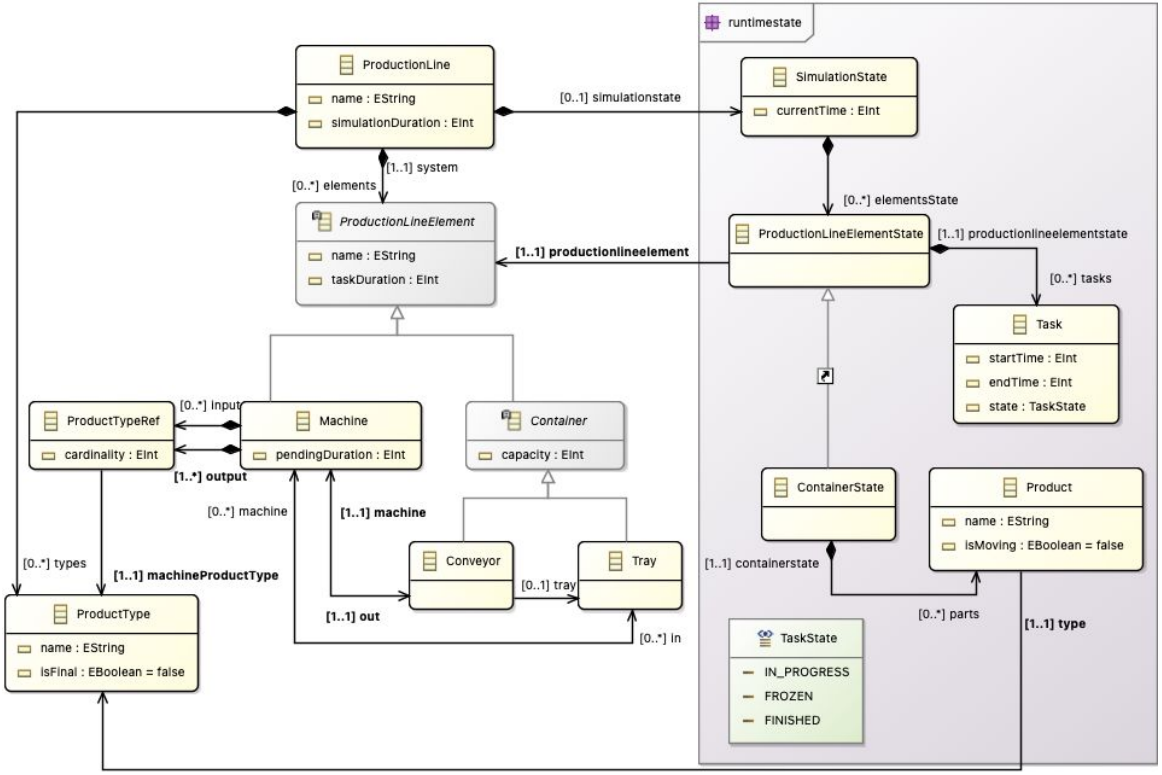
Article core subject
KPI definition and computation process


Language Workbench
Import
Import
Simple Manufacturing System (SMS) (xDSL)
Abstract syntax
emf
Operational semantics
Trace MetaModel
SMM MetaModel
SMS KPI Catalog
Throughput KPI
Machine utilisation KPI
Creates the Language
Language Engineer
Conforms to
Conforms to
Conforms to
Modeling Workbench
Model
Trace Manager
Execution trace
KPI results model
Creates model
Executes model
Compute KPIs for the model
Domain expert

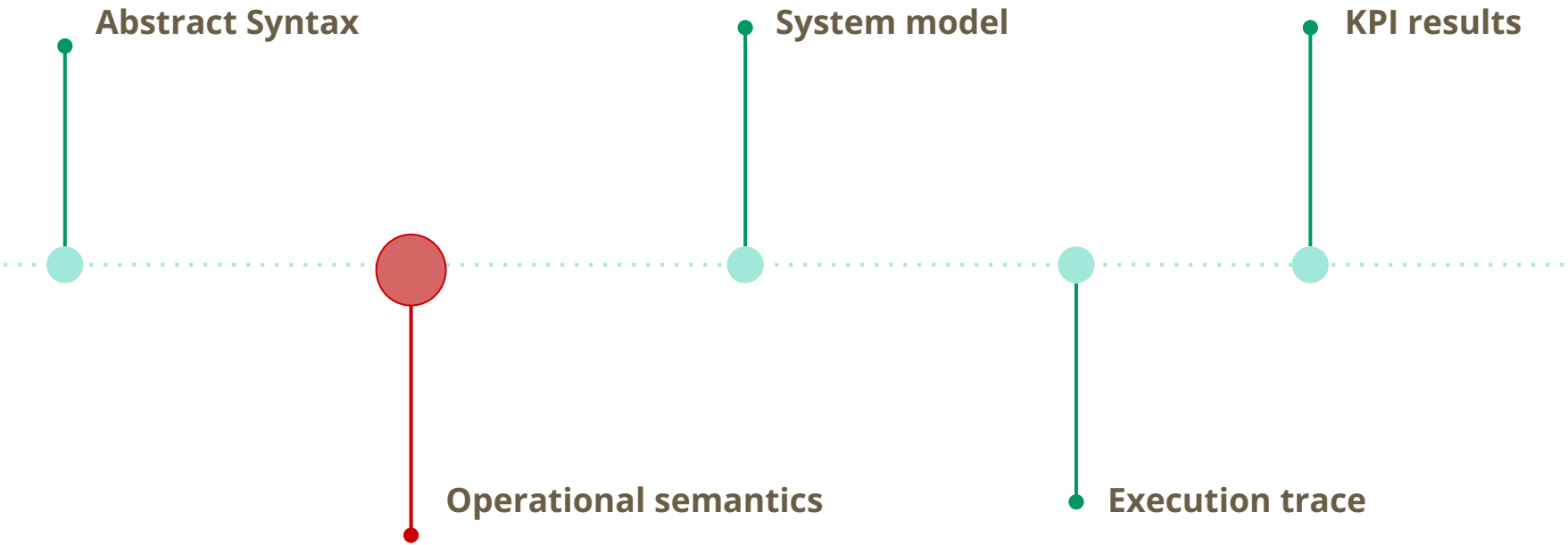
Implementation



Abstract syntax



Implementation

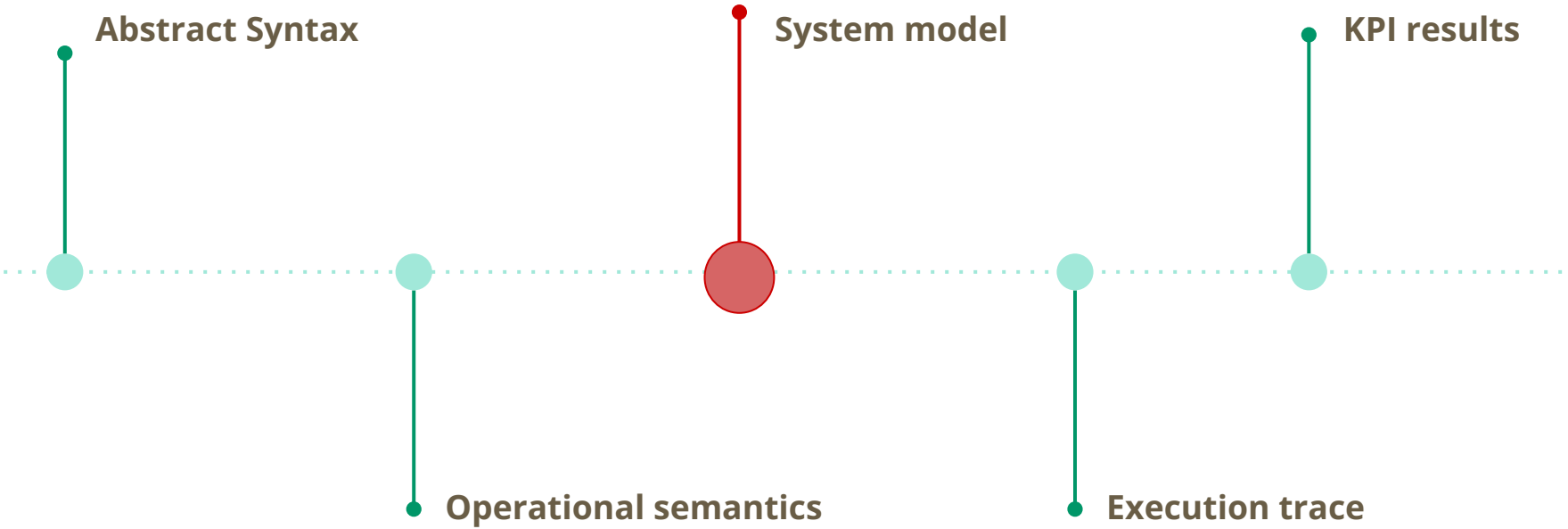


Operational semantics

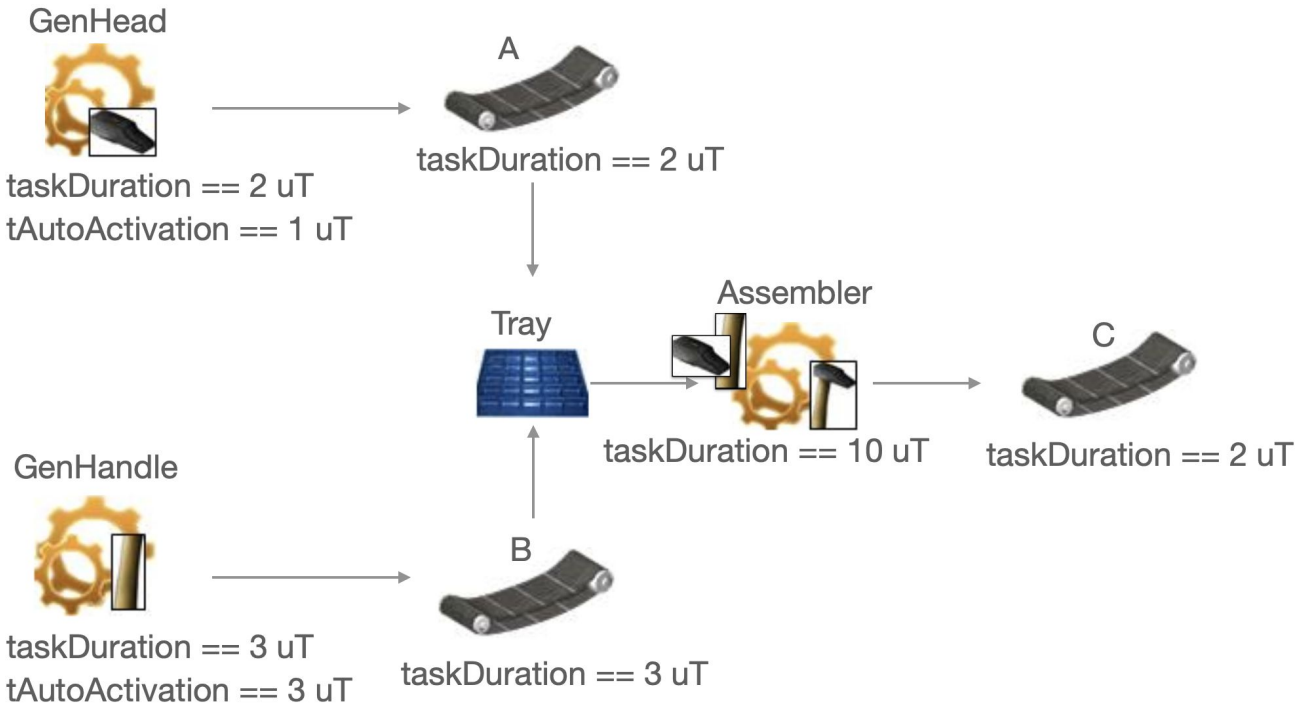
Algorithm 1: The main loop of system operation

```
Inputs:
ProductionLine: the model of the system
begin
    // Initialisation
    foreach machine ∈ ProductionLine.elements do
        if machine.input.isEmpty then
            machine.start();
        end
    end
    // The main loop
    while exists(task / task.state == IN_PROGRESS) && currentTime < ProductionLine.simulationDuration do
        // Update the current time
        currentTime ← minBy (task in progress/task.endTime);
        ProductionLine.simulationstate.currentTime = currentTime;

        // Finishing specific tasks
        foreach (task / task.state == IN_PROGRESS && task.endTime == currentTime) do
            task.finishTask();
        end

        // Asking elements to start working
        foreach element ∈ ProductionLine.elements do
            element.start();
        end
    end
end
```

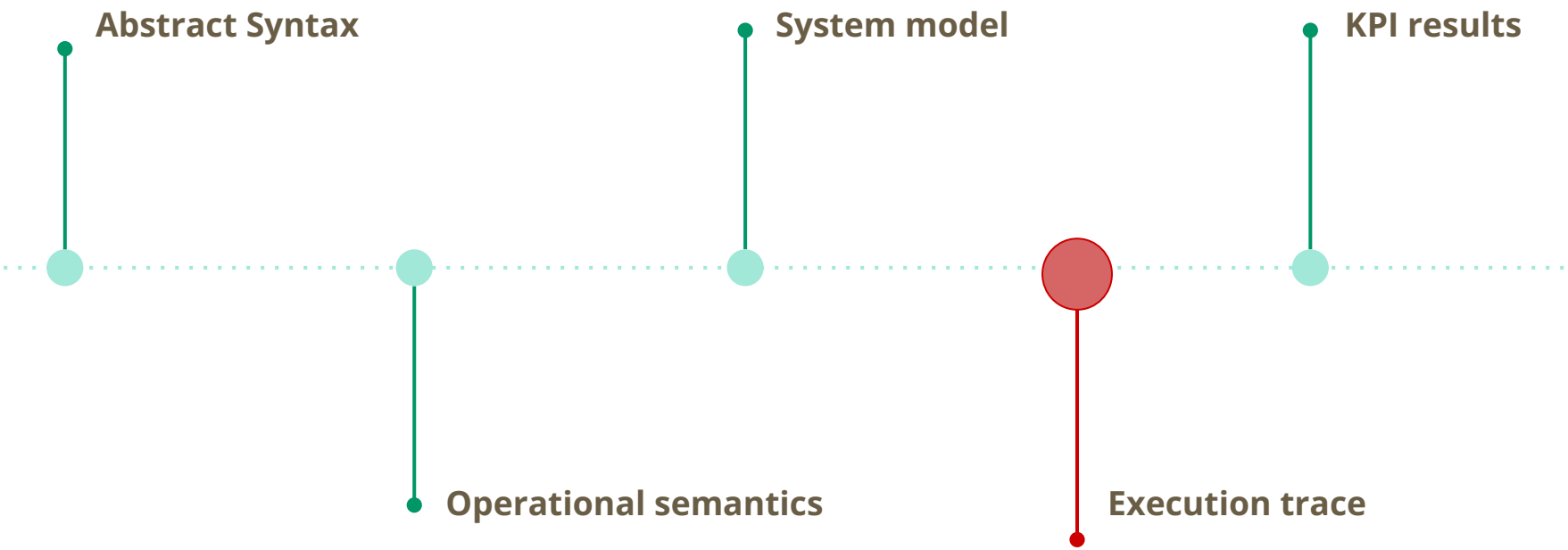
Implementation



System model

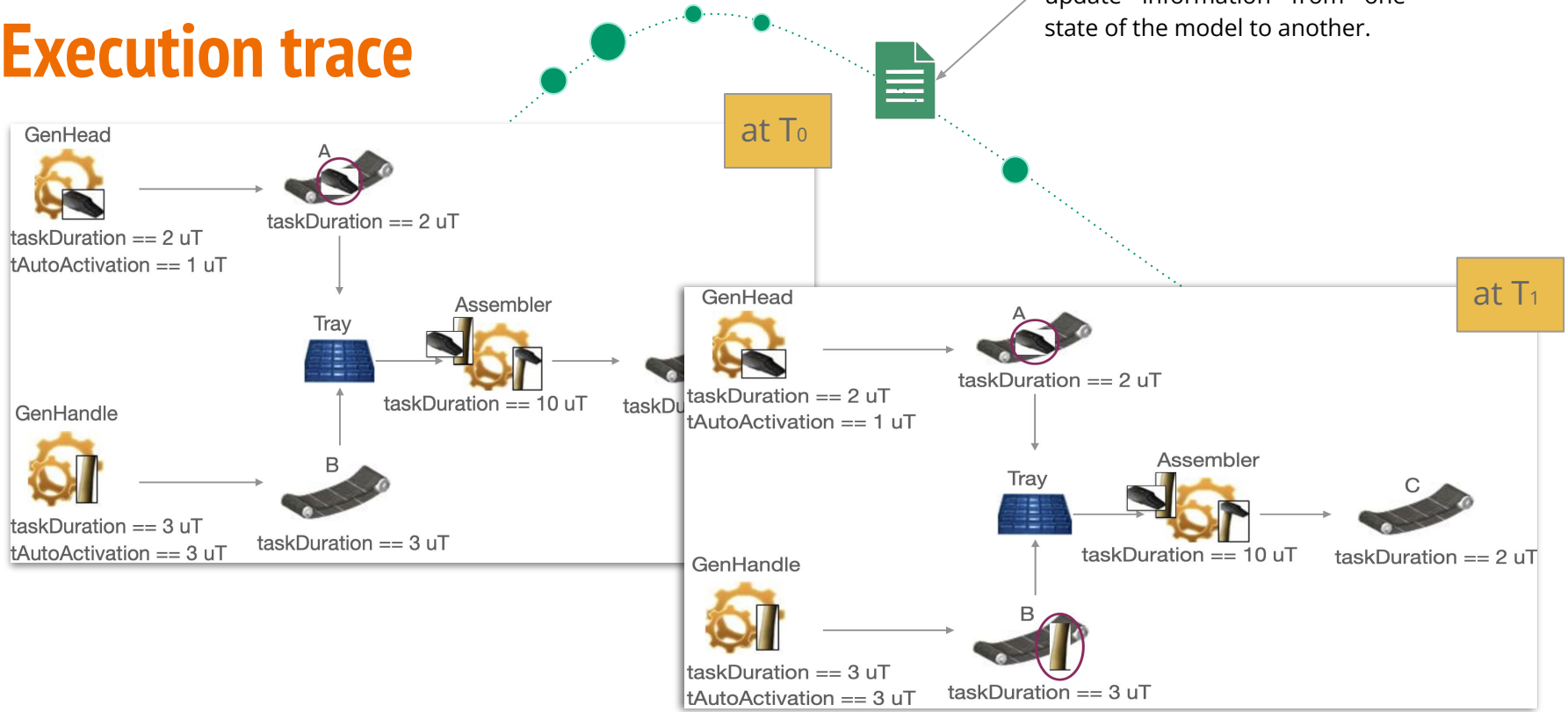


Implementation

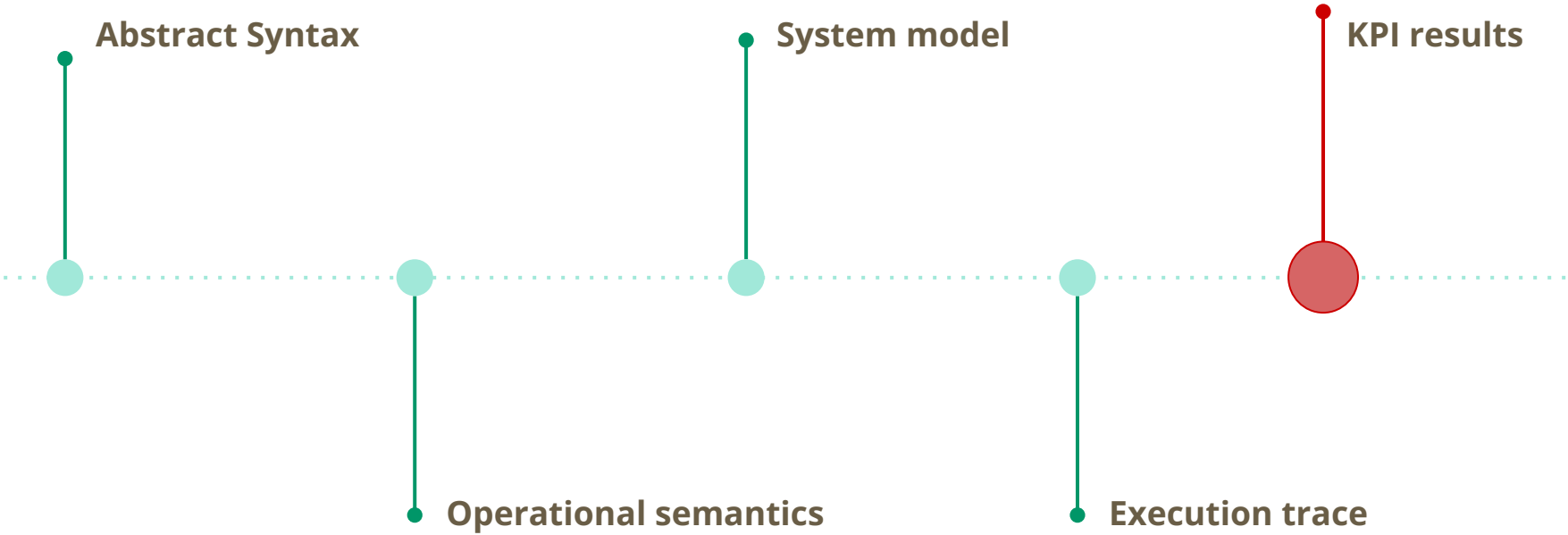


Execution trace

The execution trace captures the update information from one state of the model to another.

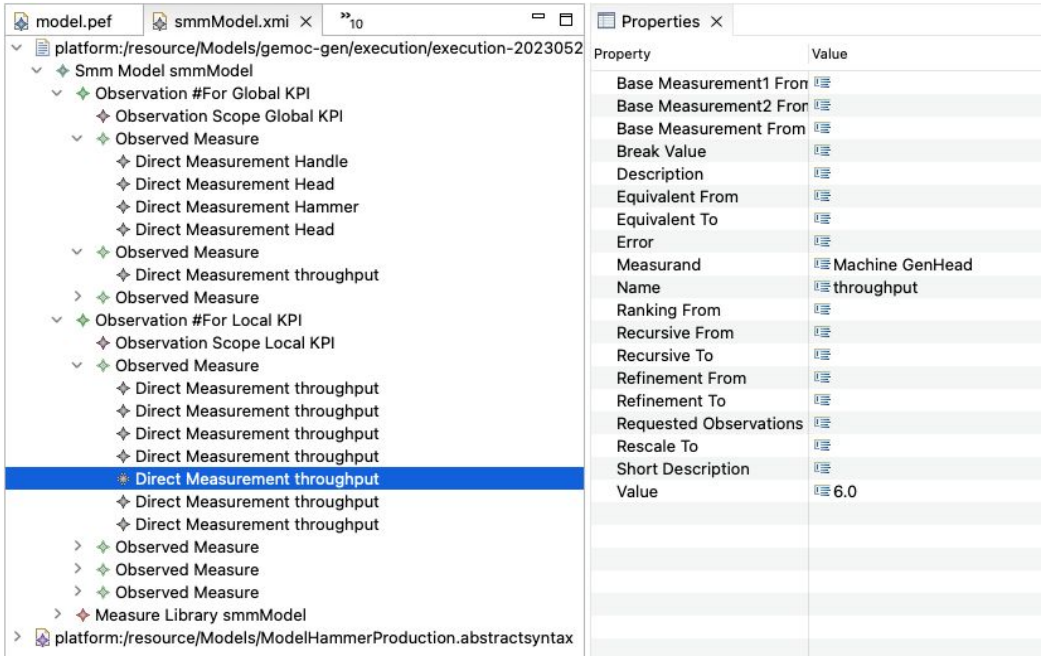


Implementation



KPI results

The execution trace is analysed to compute Different KPIs, then the results are stored in an model of Structure Metrics Meta-Model (SMM).
https://uncloud.univ-nantes.fr/index.php/s/HmtNwgnmDbsAcyY



Conclusion

- We localized WP2 at the level of the RODIC project.
- We outline the approach followed to evaluate the performance of an industrial system.
- We present the implementation suggested and the computation KPI results.

Future work

- Annotate the xDSLs with concepts needed to evaluate the performance.
- Personalize the performance definition.