Introduction to Model-Driven Simulation

Mathias Fritzsche[1], Jendrik Johannes[2], Steffen Zschaler[2], Anatoly Zherebtsov[3], Alexander Terekhov[3]

[1]SAP CEC Belfast, [2]TU Dresden, [3]XJ Technologies

Context of this work

- The present courseware has been elaborated in the context of the MODELPLEX European IST FP6 project (http://www.modelplex.org/).
- Co-funded by the European Commission, the MODELPLEX project involves 21 partners from 8 different countries.
- MODELPLEX aims at defining and developing a coherent infrastructure specifically for the application of MDE to the development and subsequent management of complex systems within a variety of industrial domains.
- To achieve the goal of large-scale adoption of MDE, MODELPLEX promotes the idea of a collaborative development of courseware dedicated to this domain.
- The MDE courseware provided here with the status of open-source software is produced under the EPL 1.0 license.

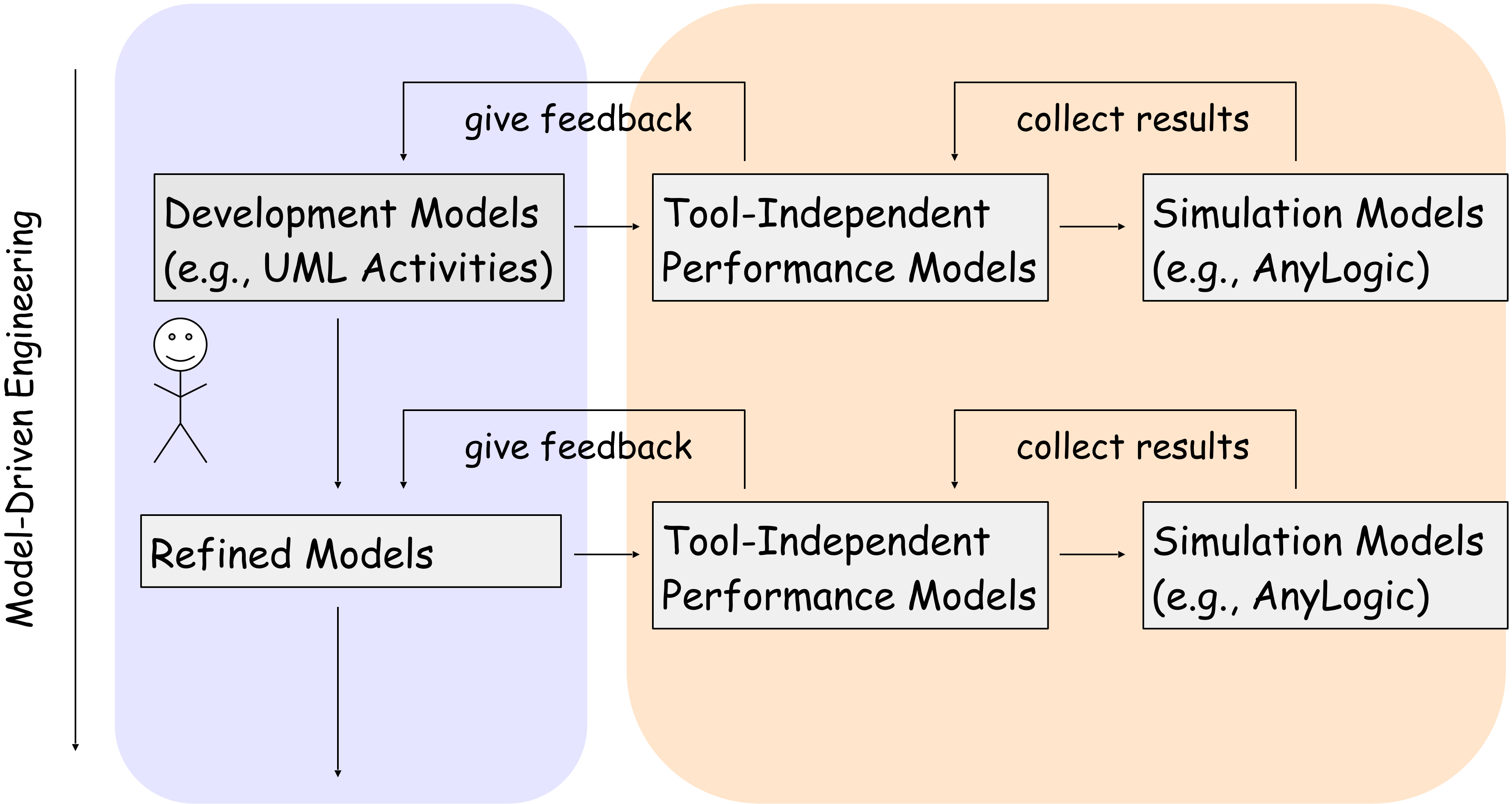
Outline

- Motivating Model-Driven Performance Engineering
- Definition of the MDPE process
- The TIPM metamodel
  - Concepts from CSM and TIPM Monitors
- Application and experiences: performance data annotation
  - Annotating UML Activities with MARTE
  - Transforming Activities to TIPM
- Application and experiences: simulation
  - Introduction to AnyLogic
  - Transforming TIPMs to AnyLogic simulations
  - collecting results and filling monitors
- Applications and experiences: result tracing
  - Tracing monitors to Activities

Motivating Model-Driven Performance Engineering

- Apply performance engineering at different abstraction levels
- Use MDE techniques to derive simulation models (e.g., AnyLogic) from development models (e.g., UML Activities with MARTE profile)
- Trace simulation results back to enable developers (who are not performance experts) to improve system design

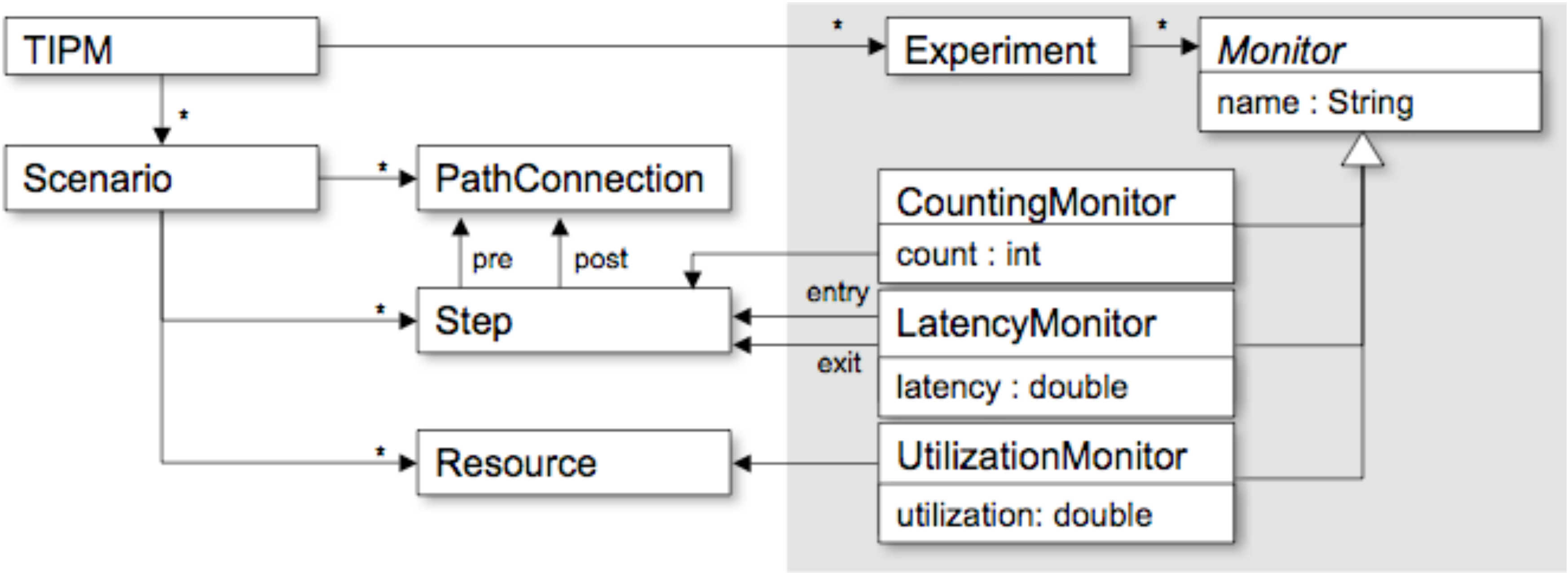
Definition of the MDPE process


Model-Driven Engineering
give feedback
collect results
Development Models (e.g., UML Activities)
Tool-Independent Performance Models
Simulation Models (e.g., AnyLogic)
give feedback
collect results
Refined Models
Tool-Independent Performance Models
Simulation Models (e.g., AnyLogic)

The Tool-Independent Performance Meta-Model

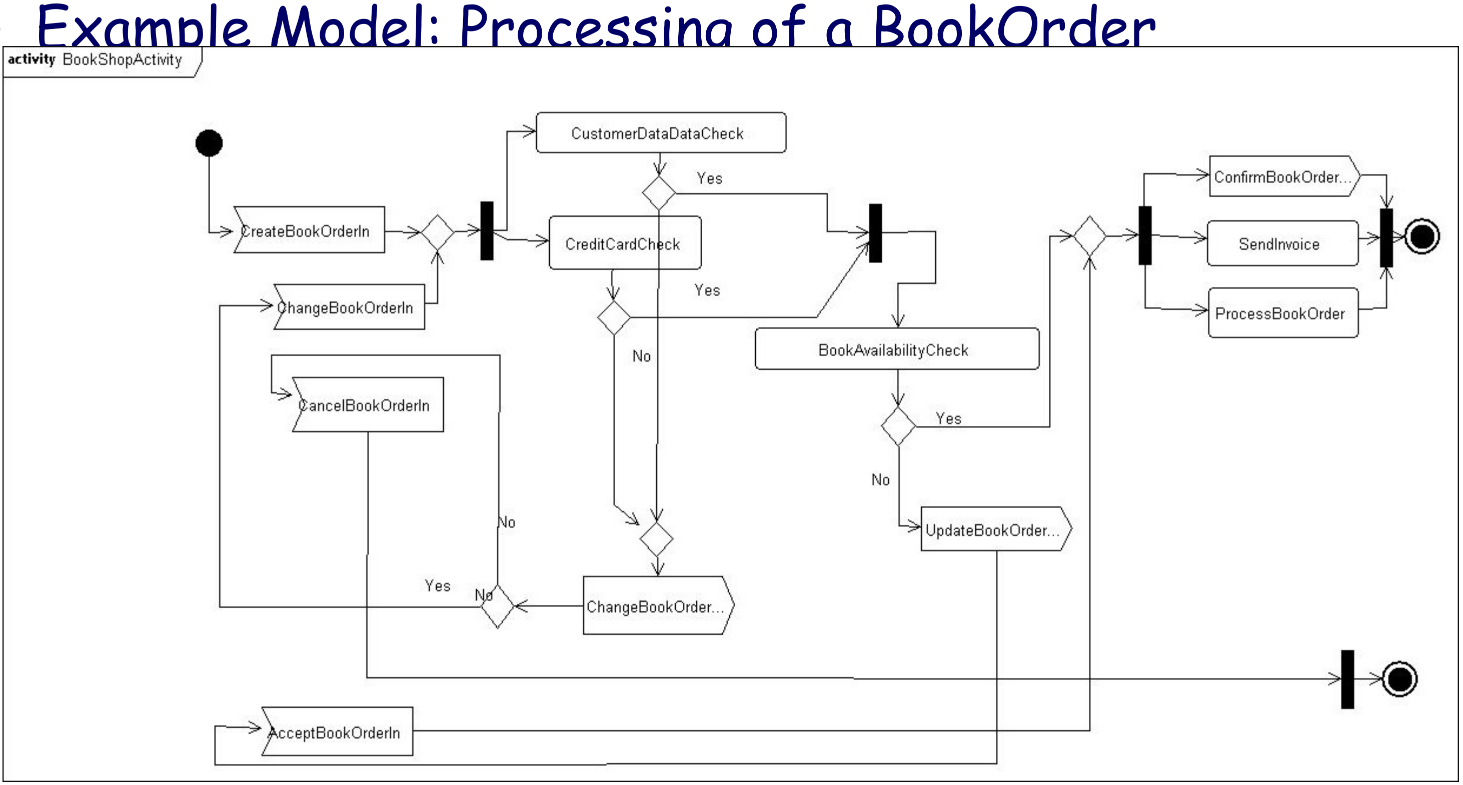
system definition (based on CSM)

monitoring

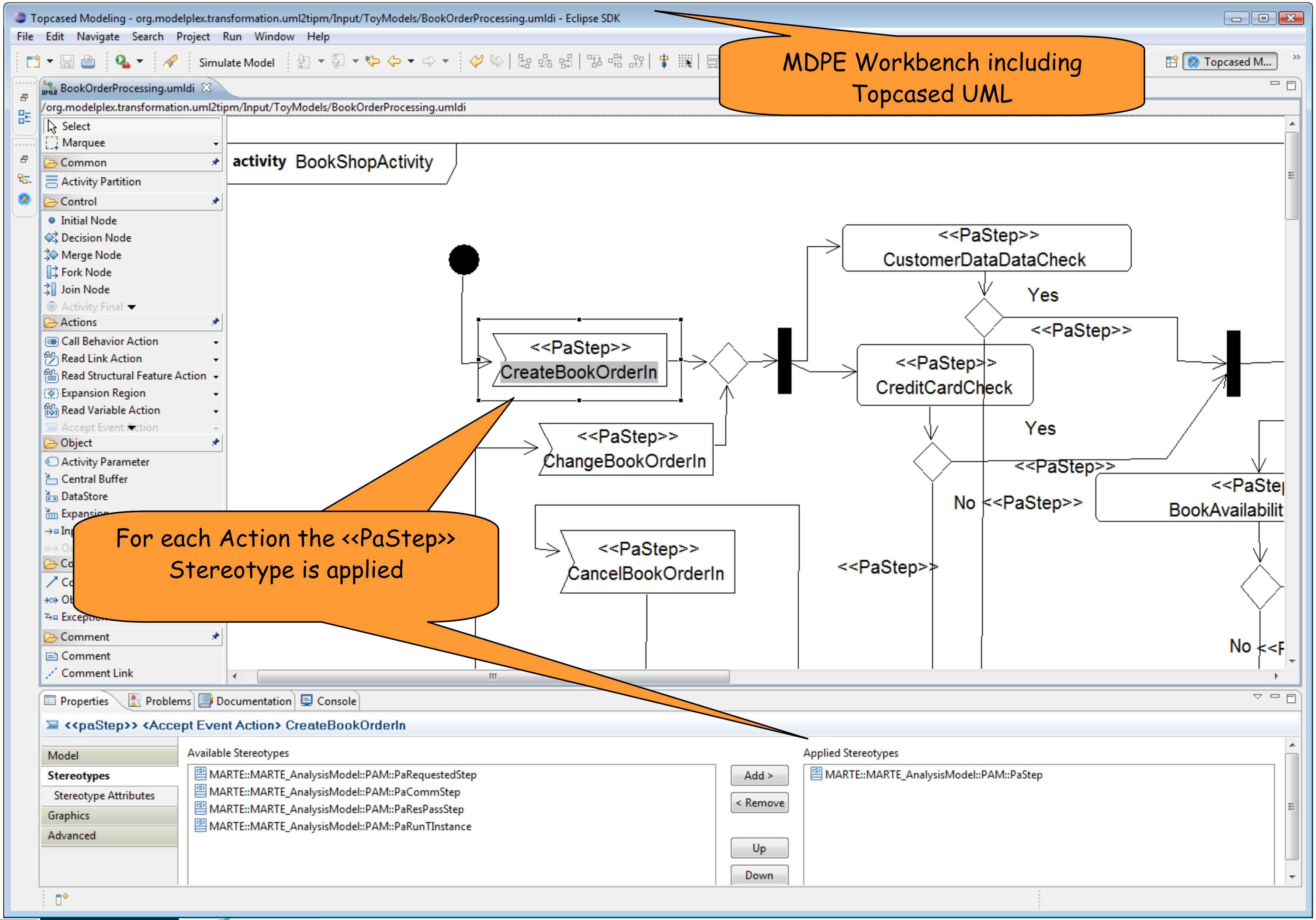


Applications and experiences: Annotating performance data (1)

- Overview
  - Simulation of UML Activities
  - Performance data are annotated by using the MARTE profile
    - Each UML Action has an expected execution time
    - To be simulated: execution time for the Activity
- Example Model: Processing of a BookOrder



Annotating performance data (2)



<<paStep>> <Accept Event Action> CreateBookOrderIn

Available Stereotypes

- MARTE::MARTE_AnalysisModel::PAM::PaRequestedStep
- MARTE::MARTE_AnalysisModel::PAM::PaCommStep
- MARTE::MARTE_AnalysisModel::PAM::PaResPassStep
- MARTE::MARTE_AnalysisModel::PAM::PaRunTInstance

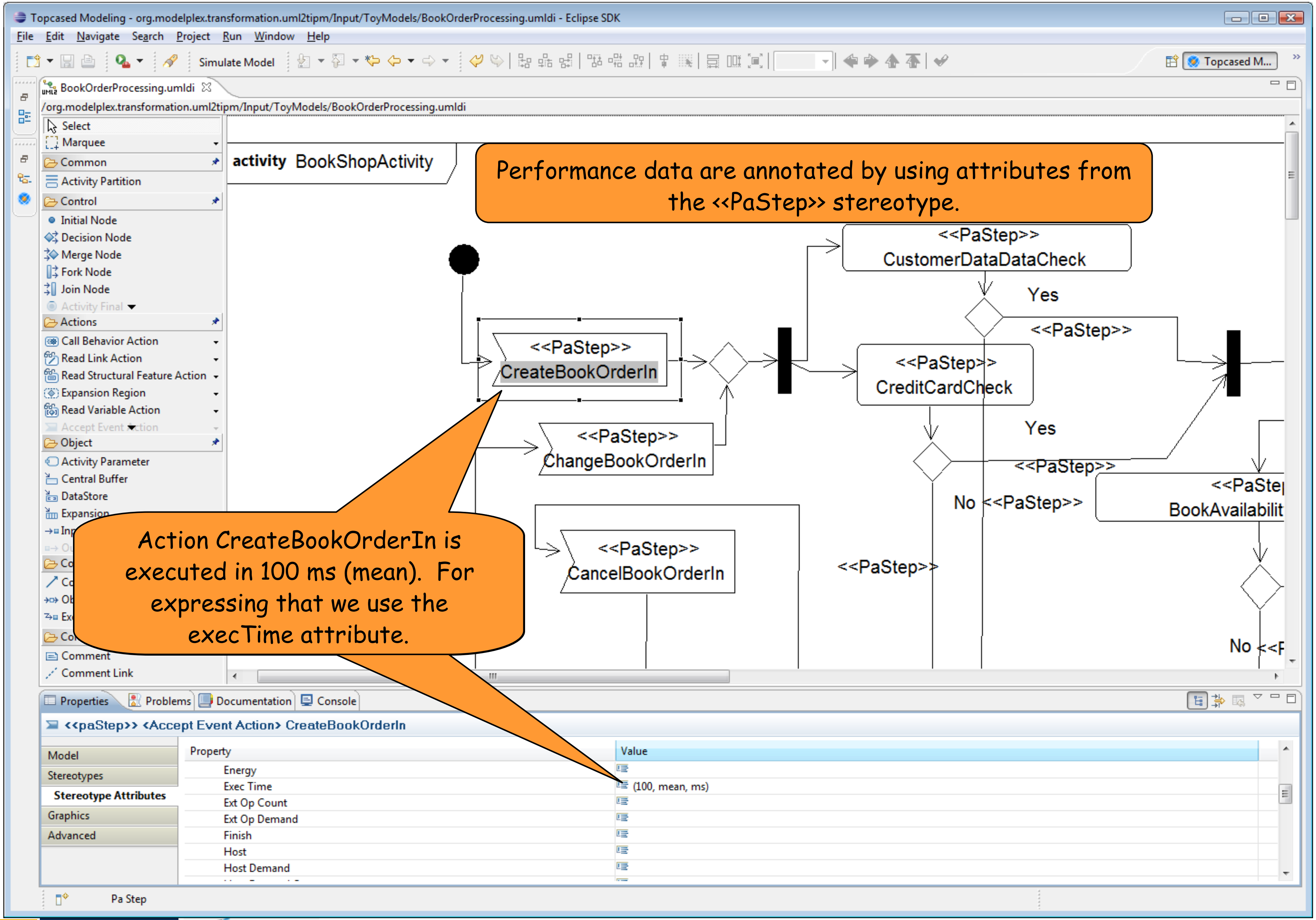
Applied Stereotypes

- MARTE::MARTE_AnalysisModel::PAM::PaStep

MDPE Workbench including Topcased UML

For each Action the <<PaStep>> Stereotype is applied

Applications and experiences: performance data annotation (2)



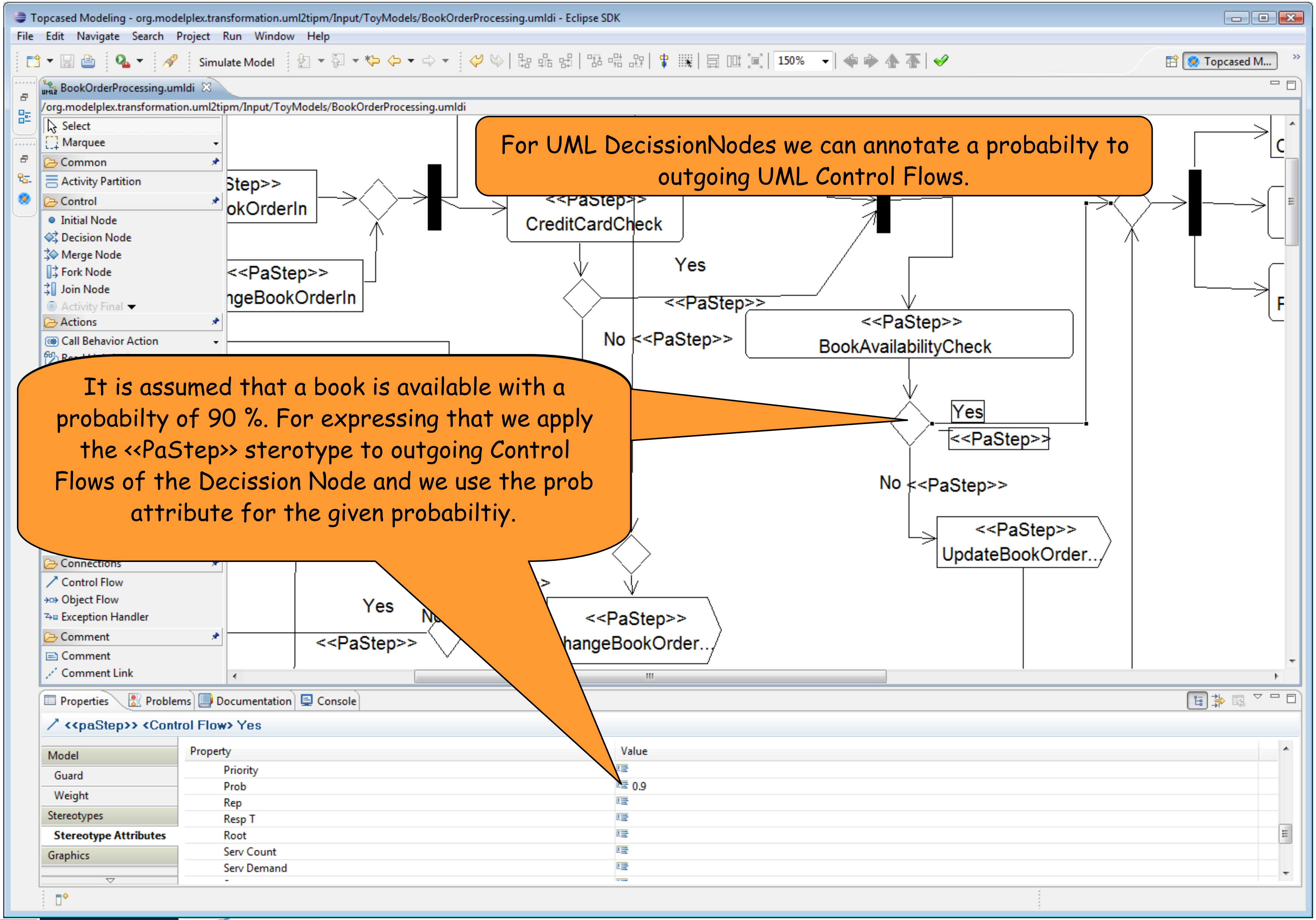
Performance data are annotated by using attributes from the <<PaStep>> stereotype.

Action CreateBookOrderIn is executed in 100 ms (mean). For expressing that we use the execTime attribute.

<<paStep>> <Accept Event Action> CreateBookOrderIn

Property	Value
Energy	
Exec Time	(100, mean, ms)
Ext Op Count	
Ext Op Demand	
Finish	
Host	
Host Demand	

Annotating performance data (3)



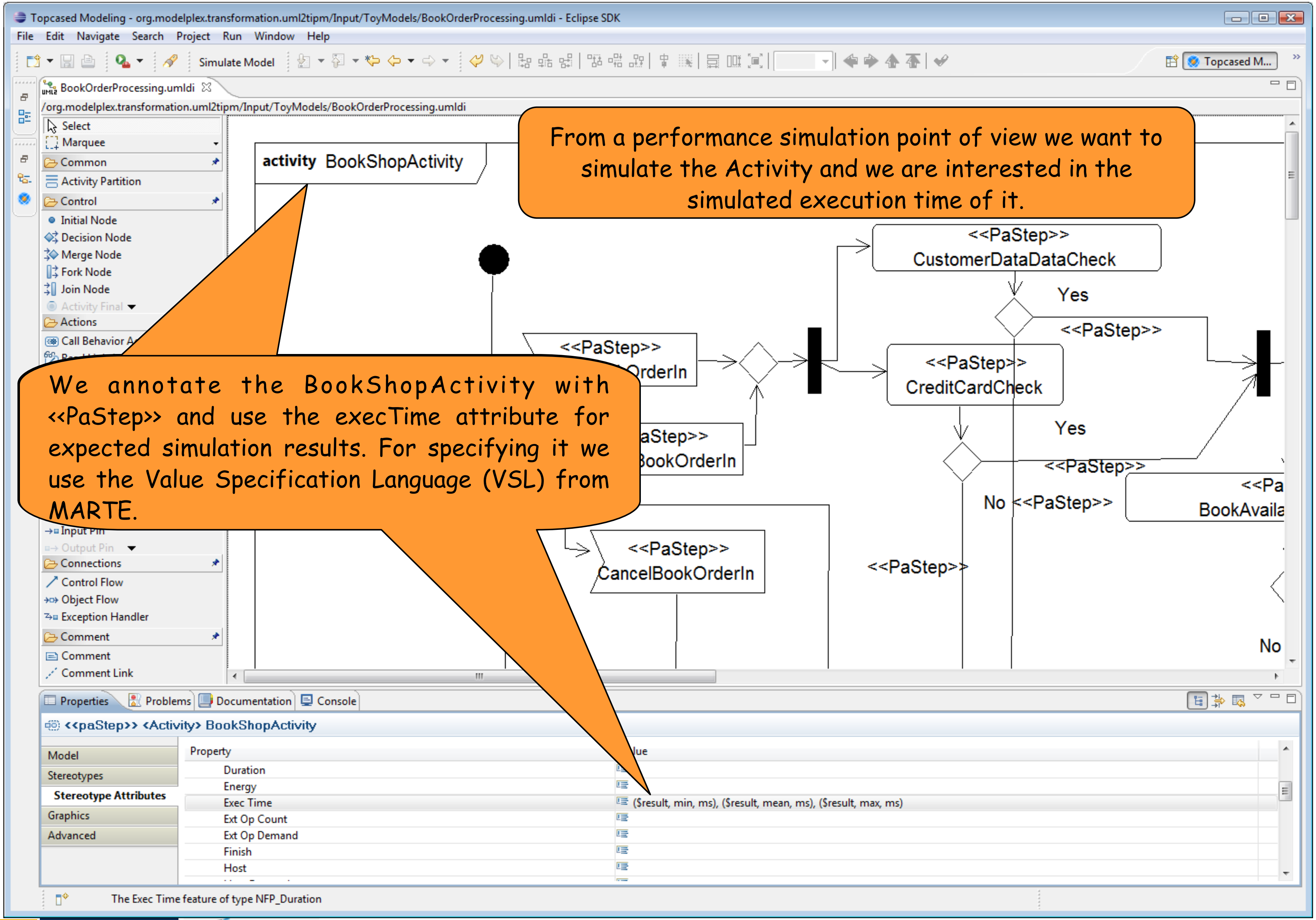
For UML DecissionNodes we can annotate a probabilty to outgoing UML Control Flows.

It is assumed that a book is available with a probabilty of 90 %. For expressing that we apply the <<PaStep>> sterotype to outgoing Control Flows of the Decission Node and we use the prob attribute for the given probabiltiy.

<<paStep>> <Control Flow> Yes

Property	Value
Priority	
Prob	0.9
Rep	
Resp T	
Root	
Serv Count	
Serv Demand	

Annotating performance data (4)



Topcased Modeling - org.modelplex.transformation.uml2tipm/Input/ToyModels/BookOrderProcessing.umldi - Eclipse SDK

/org.modelplex.transformation.uml2tipm/Input/ToyModels/BookOrderProcessing.umldi

From a performance simulation point of view we want to simulate the Activity and we are interested in the simulated execution time of it.

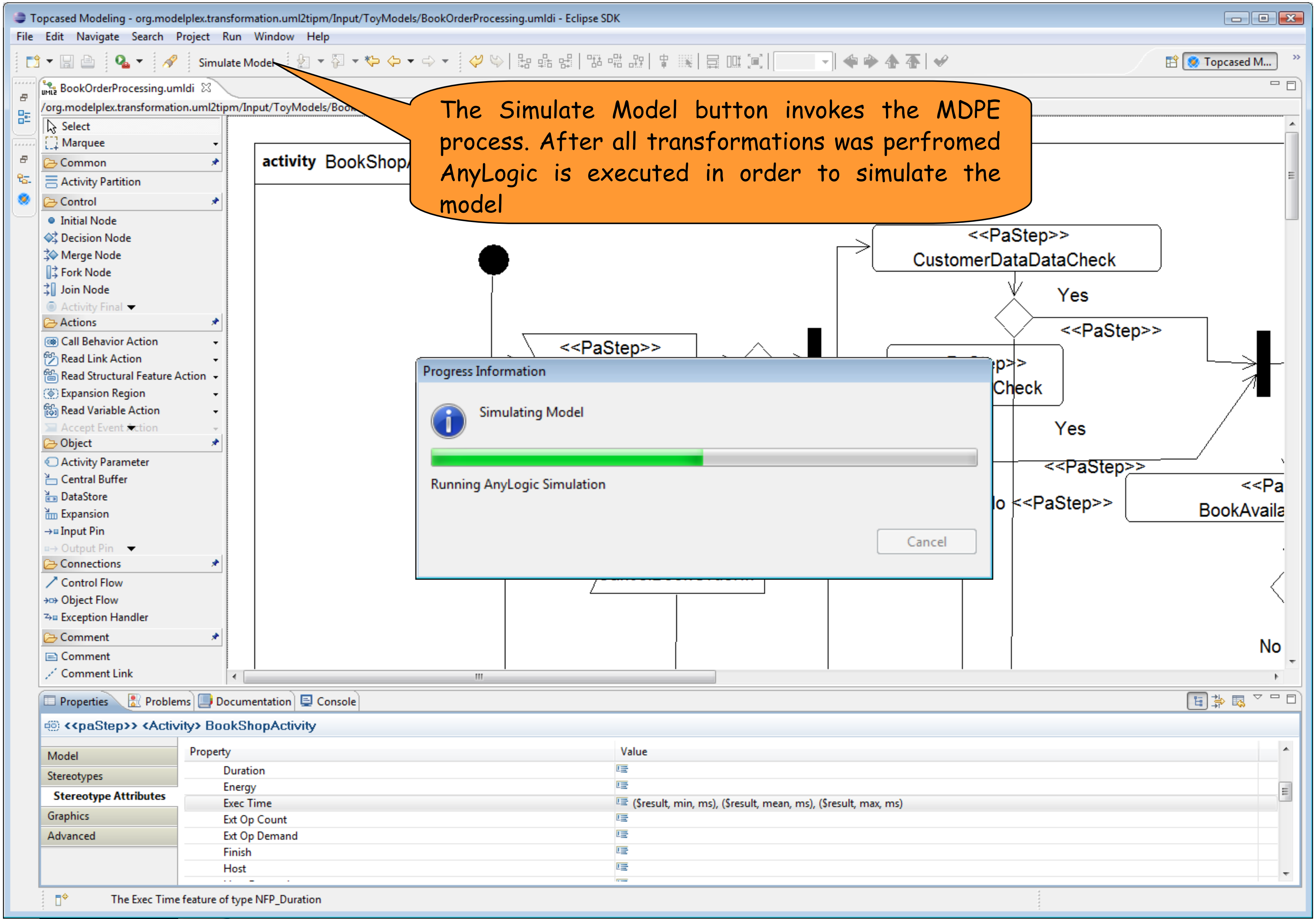
We annotate the BookShopActivity with <<PaStep>> and use the execTime attribute for expected simulation results. For specifying it we use the Value Specification Language (VSL) from MARTE.

<<paStep>> <Activity> BookShopActivity

Property	Value
Duration	
Energy	
Exec Time	($result, min, ms), ($result, mean, ms), ($result, max, ms)
Ext Op Count	
Ext Op Demand	
Finish	
Host	

The Exec Time feature of type NFP_Duration

MDPE Process Execution



Topcased Modeling - org.modelplex.transformation.uml2tipm/Input/ToyModels/BookOrderProcessing.umldi - Eclipse SDK

The Simulate Model button invokes the MDPE process. After all transformations was perfromed AnyLogic is executed in order to simulate the model

Progress Information

Simulating Model

Running AnyLogic Simulation

Cancel

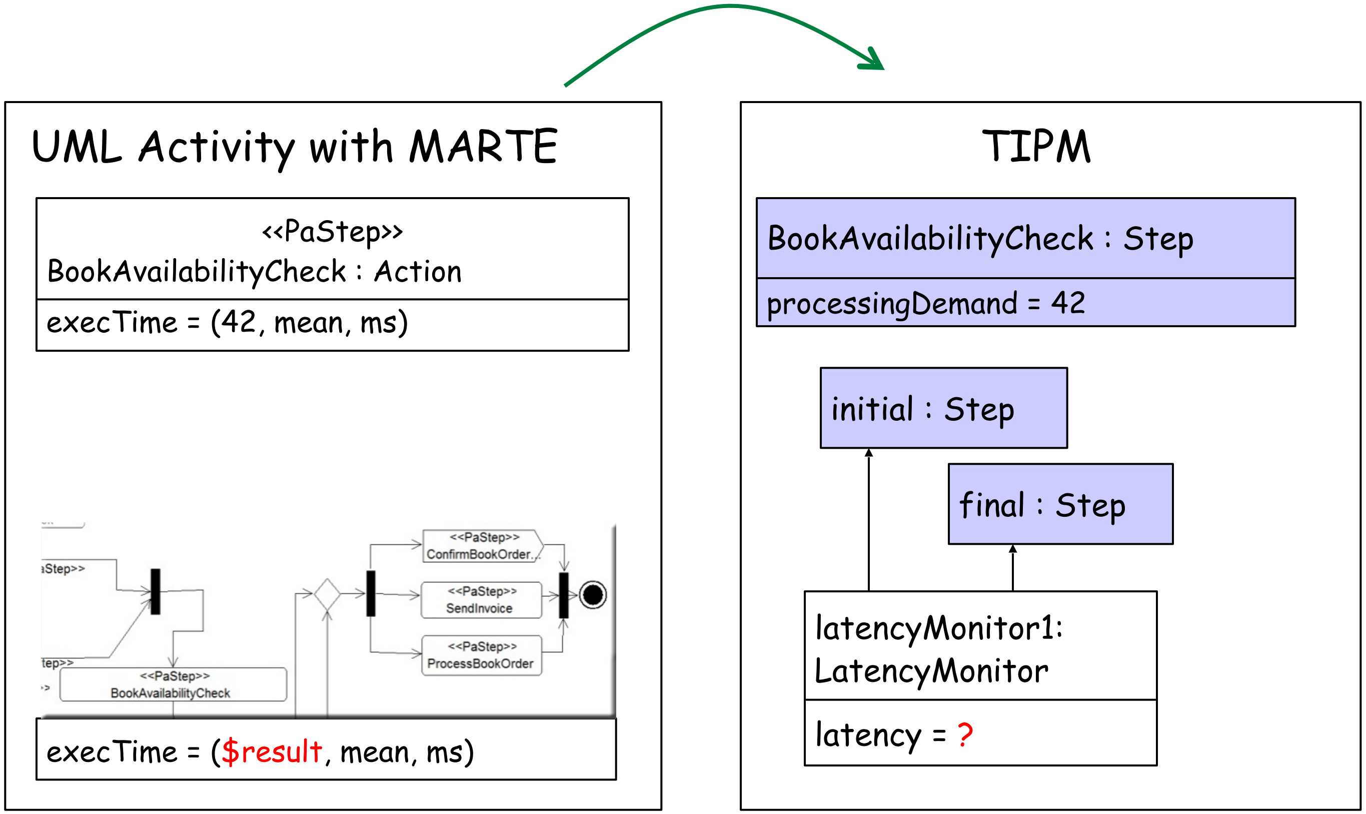
<<paStep>> <Activity> BookShopActivity

Property	Value
Duration	
Energy	
Exec Time	($result, min, ms), ($result, mean, ms), ($result, max, ms)
Ext Op Count	
Ext Op Demand	
Finish	
Host	

The Exec Time feature of type NFP_Duration

MDPE process background: Transforming Activities to TIPM

- TIPM is generated based on MARTE annotated UML Activities
- Example:
  - PaStep annotated Actios are transformed to a TIPM Step
  - For measuring the latency for an Activity a TIPM LatencyMonitor between the InitialNode and the FinalNode is generated



UML Activity with MARTE

<<PaStep>>
BookAvailabilityCheck : Action

execTime = (42, mean, ms)

execTime = ($result, mean, ms)

TIPM

BookAvailabilityCheck : Step

processingDemand = 42

initial : Step

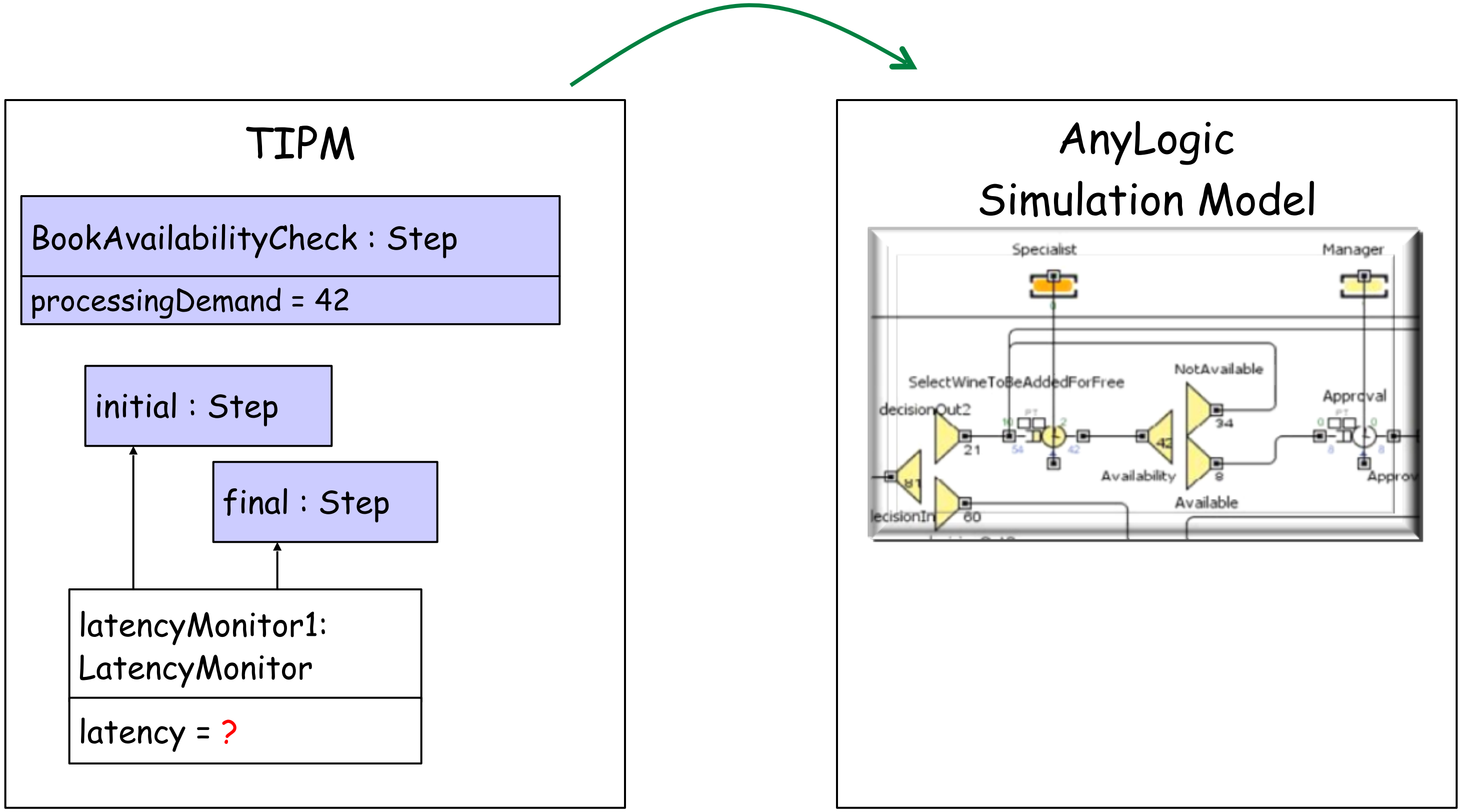
final : Step

latencyMonitor1:
LatencyMonitor

latency = ?

MDPE process background: Transforming TIPMs into simulation models

- TIPMs are transformed into AnyLogic Simulation models (TSPM)
- In AnyLogic TSPMs are based on a specific library for MDPE
- Simulation models can be simulated directly in AnyLogic



TIPM

BookAvailabilityCheck : Step

processingDemand = 42

initial : Step

final : Step

latencyMonitor1: LatencyMonitor

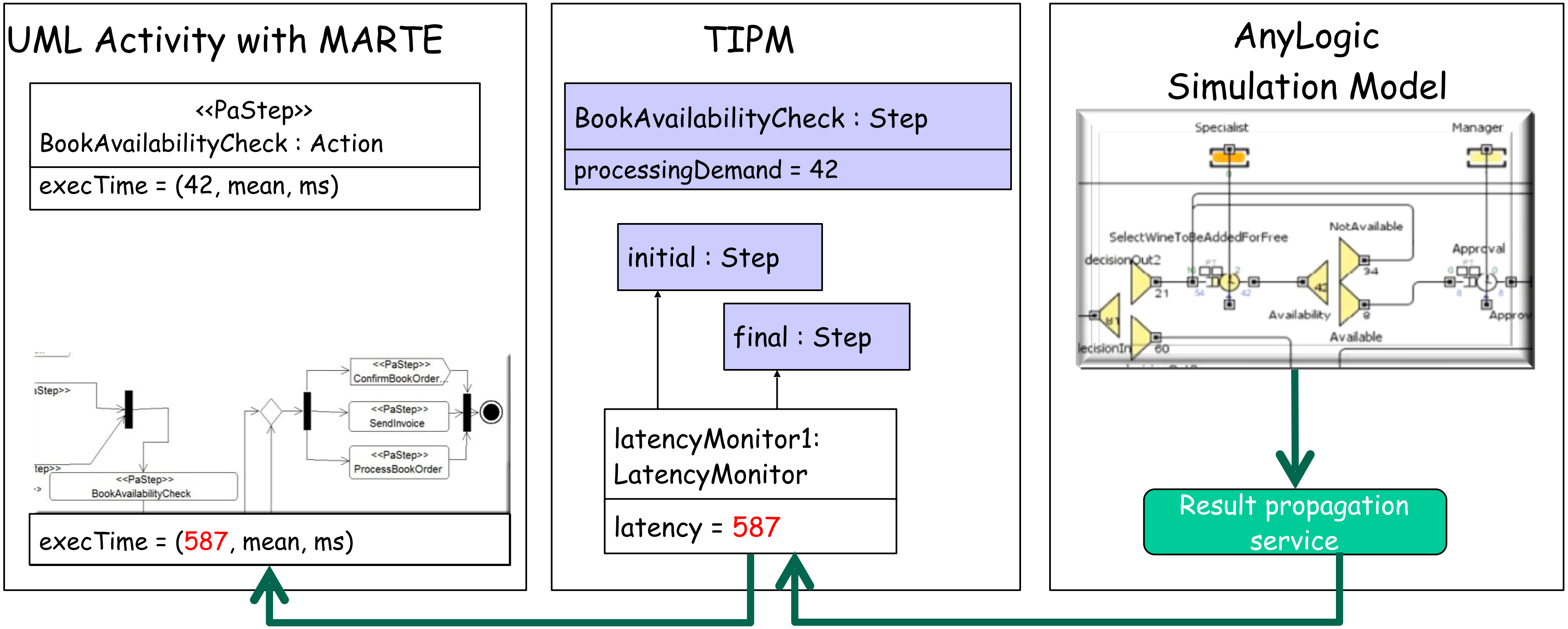
latency = ?

AnyLogic Simulation Model

TSPM … Tool Specific Simulation Model

MDPE process background: Result tracing

- Model developers get simulation results directly into their development models
- For realizing this
  - AnyLogic provides a service for collecting results in the TIPM
  - By using model tracing techniques simulation results can be visualized directly the original development model



UML Activity with MARTE

<<PaStep>>
BookAvailabilityCheck : Action

execTime = (42, mean, ms)

execTime = (587, mean, ms)

TIPM

BookAvailabilityCheck : Step

processingDemand = 42

initial : Step

final : Step

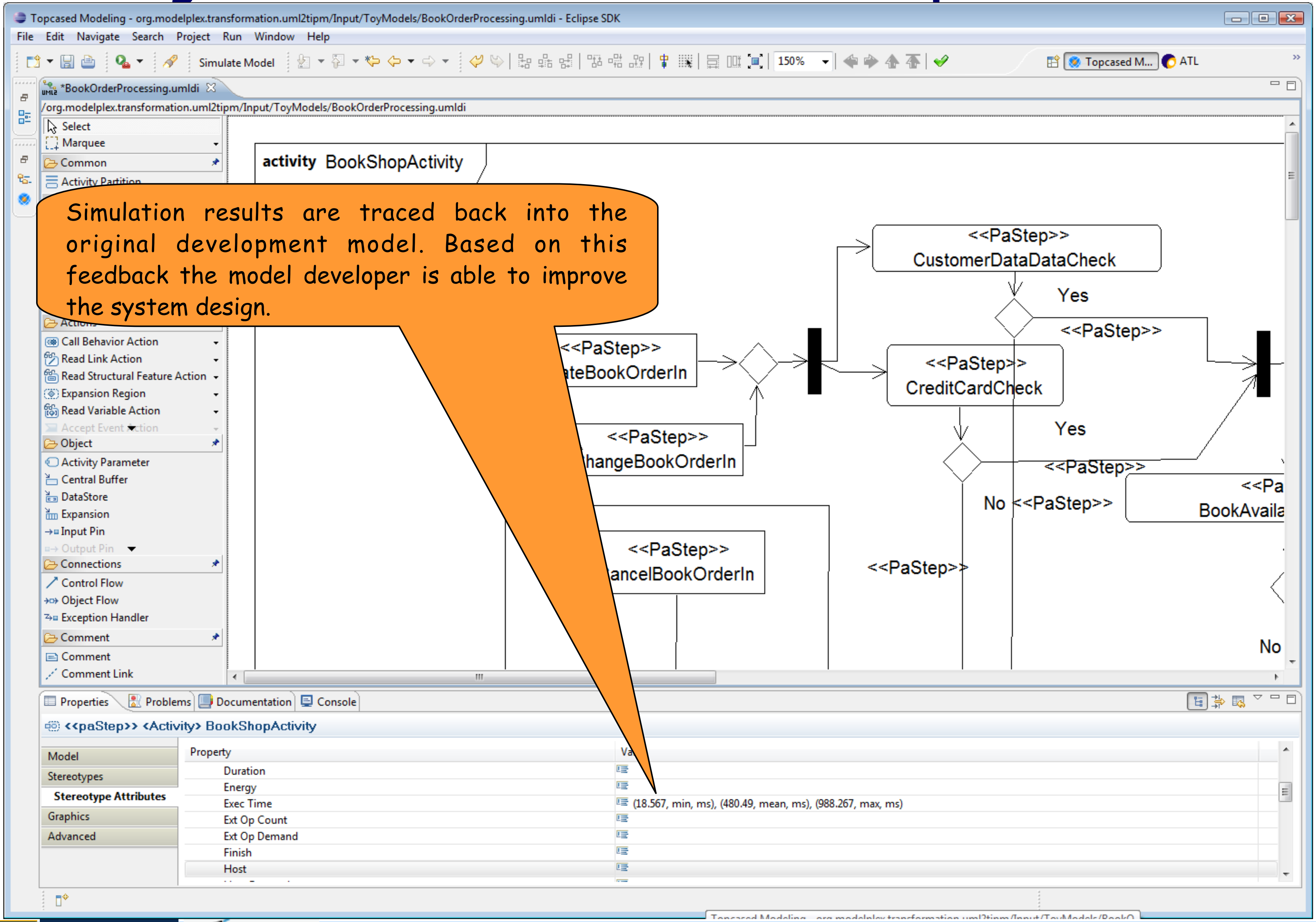
latencyMonitor1: LatencyMonitor

latency = 587

AnyLogic Simulation Model

Result propagation service

Visualizing simulation results in the development model



Simulation results are traced back into the original development model. Based on this feedback the model developer is able to improve the system design.

<<paStep>> <Activity> BookShopActivity

Property	Va
Duration	
Energy	
Exec Time	(18.567, min, ms), (480.49, mean, ms), (988.267, max, ms)
Ext Op Count	
Ext Op Demand	
Finish	
Host	