

# Introduction to Model-Driven Simulation

Mathias Fritzsche<sup>1</sup>, Jendrik Johannes<sup>2</sup>, Steffen Zschaler<sup>2</sup>,  
Anatoly Zhrebtssov<sup>3</sup>, Alexander Terekhov<sup>3</sup>

<sup>1</sup>SAP CEC Belfast, <sup>2</sup>TU Dresden, <sup>3</sup>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.

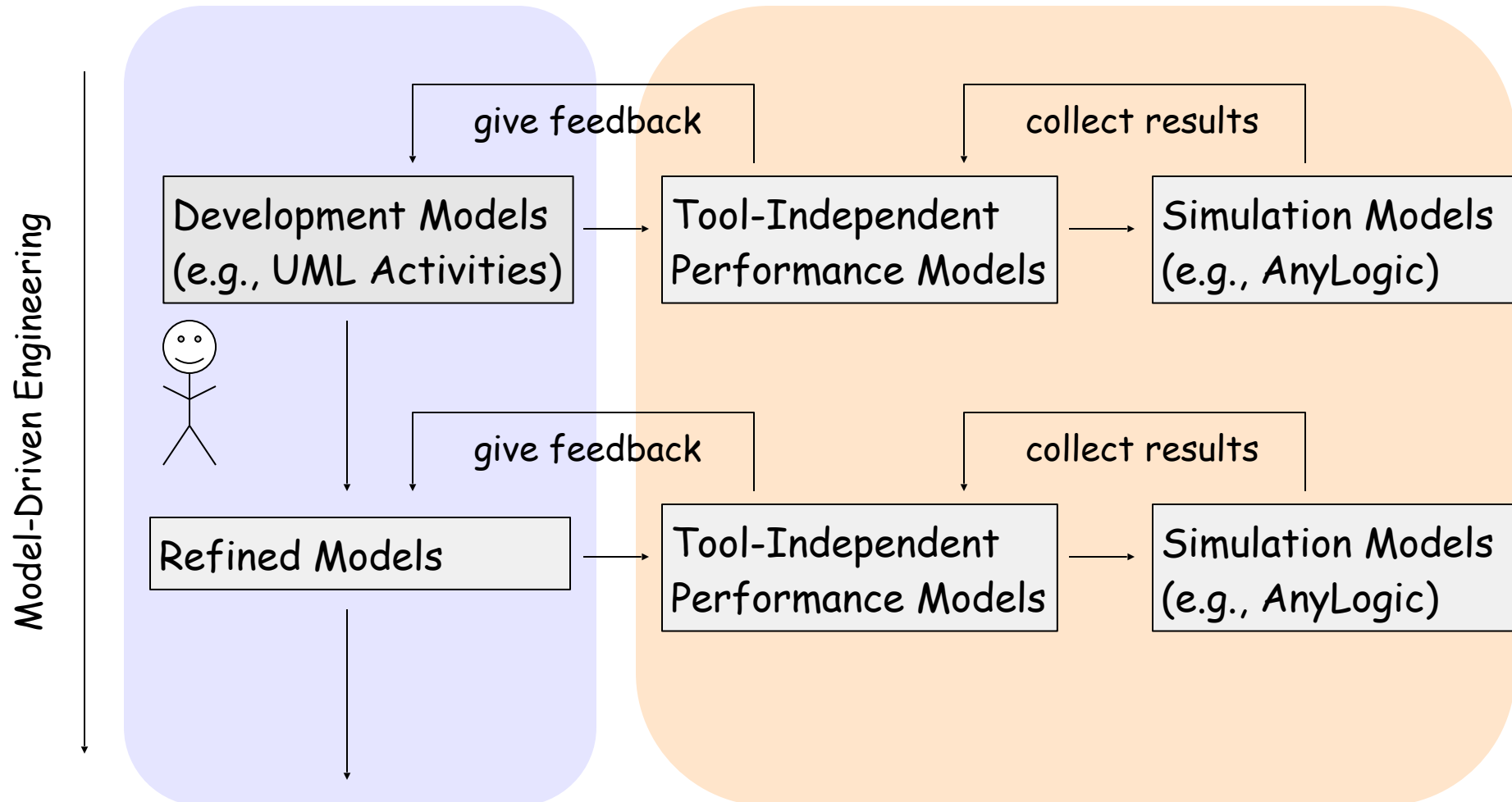
# Outline

- 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

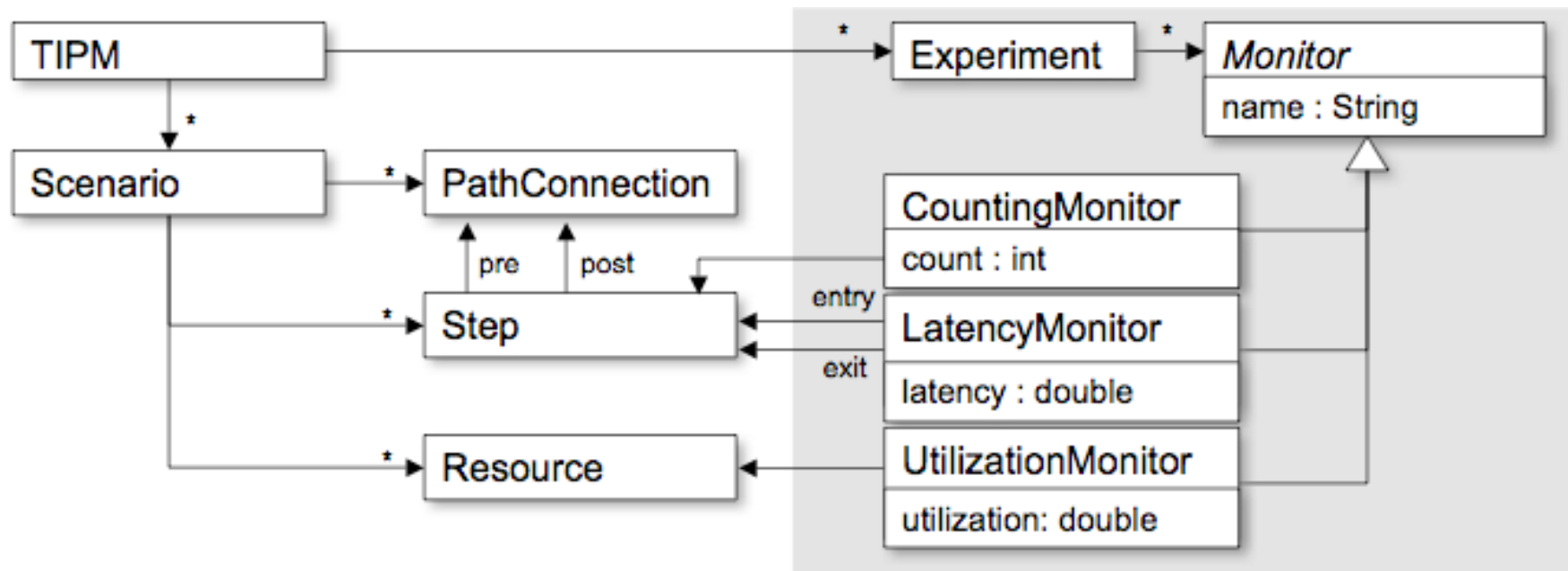
# Definition of the MDPE process



# The Tool-Independent Performance Meta-Model

system definition  
(based on CSM)

monitoring

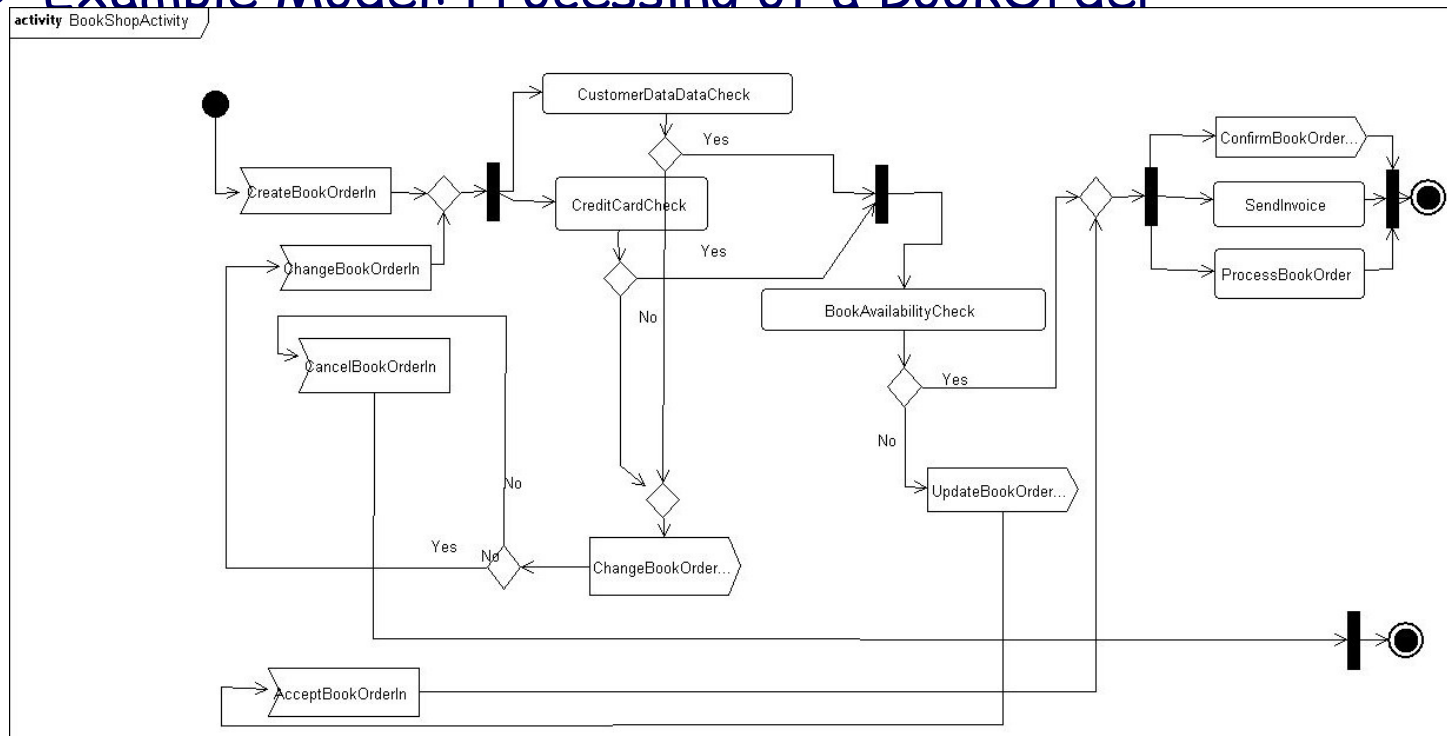


# Applications and experiences: Annotating performance data (1)

## ● 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)

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

File Edit Navigate Search Project Run Window Help

Simulate Model

BookOrderProcessing.uml-di

org.modelplex.transformation.uml2tipm/Input/ToyModels/BookOrderProcessing.uml-di

activity BookShopActivity

MDPE Workbench including Topcased UML

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

Properties Problems Documentation Console

<<paStep>> <<Accept Event Action>> CreateBookOrderIn

Model

Stereotypes

Stereotype Attributes

Graphics

Advanced

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

Diagram description: The diagram shows an activity flow for 'BookShopActivity'. It starts with an initial node leading to a 'CreateBookOrderIn' action, which is annotated with the '<<PaStep>>' stereotype. This is followed by a decision node, then a 'ChangeBookOrderIn' action (also annotated), and another decision node. The flow then branches into 'CustomerDataDataCheck' and 'CreditCardCheck' actions, both annotated with '<<PaStep>>'. There are further decision nodes and actions like 'BookAvailability' and 'CancelBookOrderIn' (annotated). The diagram illustrates how performance data is annotated onto UML activity actions.



# Applications and experiences: performance data annotation (2)

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

File Edit Navigate Search Project Run Window Help

Simulate Model

BookOrderProcessing.uml

activity BookShopActivity

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

CustomerDataDataCheck

Yes

<<PaStep>>

CreditCardCheck

Yes

<<PaStep>>

BookAvailabilit

No <<PaStep>>

CancelBookOrderIn

ChangeBookOrderIn

CreateBookOrderIn

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

Properties Problems Documentation Console

<<paStep>> <<Accept Event Action>> CreateBookOrderIn

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

Pa Step

# Annotating performance data (3)

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

File Edit Navigate Search Project Run Window Help

Simulate Model 150%

BookOrderProcessing.uml

For UML DecisionNodes we can annotate a probability to outgoing UML Control Flows.

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

UML Activity Diagram: BookOrderProcessing.uml

```

    graph TD
      Start(( )) --> BookOrderIn[BookOrderIn]
      BookOrderIn --> Merge(( ))
      Merge --> CreditCardCheck[CreditCardCheck]
      CreditCardCheck --> Decision1{ }
      Decision1 -- Yes --> BookAvailabilityCheck[BookAvailabilityCheck]
      Decision1 -- No --> ChangeBookOrderIn[ChangeBookOrderIn]
      BookAvailabilityCheck --> Decision2{ }
      Decision2 -- Yes --> UpdateBookOrder[UpdateBookOrder...]
      Decision2 -- No --> ChangeBookOrderOut[ChangeBookOrder...]
      ChangeBookOrderIn --> Merge2(( ))
      ChangeBookOrderOut --> Merge2
      Merge2 --> Fork(( ))
      Fork --> C[ ]
      Fork --> F[ ]
  
```

Properties: <<paStep>> <Control Flow> Yes

Property	Value
Priority	
Guard	
Weight	
Rep	
Resp T	
Root	
Serv Count	
Serv Demand	
prob	0.9

# Annotating performance data (4)

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.

The screenshot shows the Eclipse IDE with the following elements:

- Activity Diagram:** An activity named `BookShopActivity` is shown with several performance steps:
  - `<<PaStep>> OrderIn`
  - `<<PaStep>> BookOrderIn`
  - `<<PaStep>> CancelBookOrderIn`
  - `<<PaStep>> CustomerDataDataCheck`
  - `<<PaStep>> CreditCardCheck`
  - `<<PaStep>> BookAvailable`
- Properties View:** The `<<paStep>> <Activity> BookShopActivity` is selected, showing the `execTime` property with the value `($result, min, ms), ($result, mean, ms), ($result, max, ms)`.

## MDPE Process Execution

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

Progress Information

Simulating Model

Running AnyLogic Simulation

Cancel

activity BookShop

CustomerDataDataCheck

Yes

Check

Yes

BookAvailable

No

Properties Problems Documentation Console

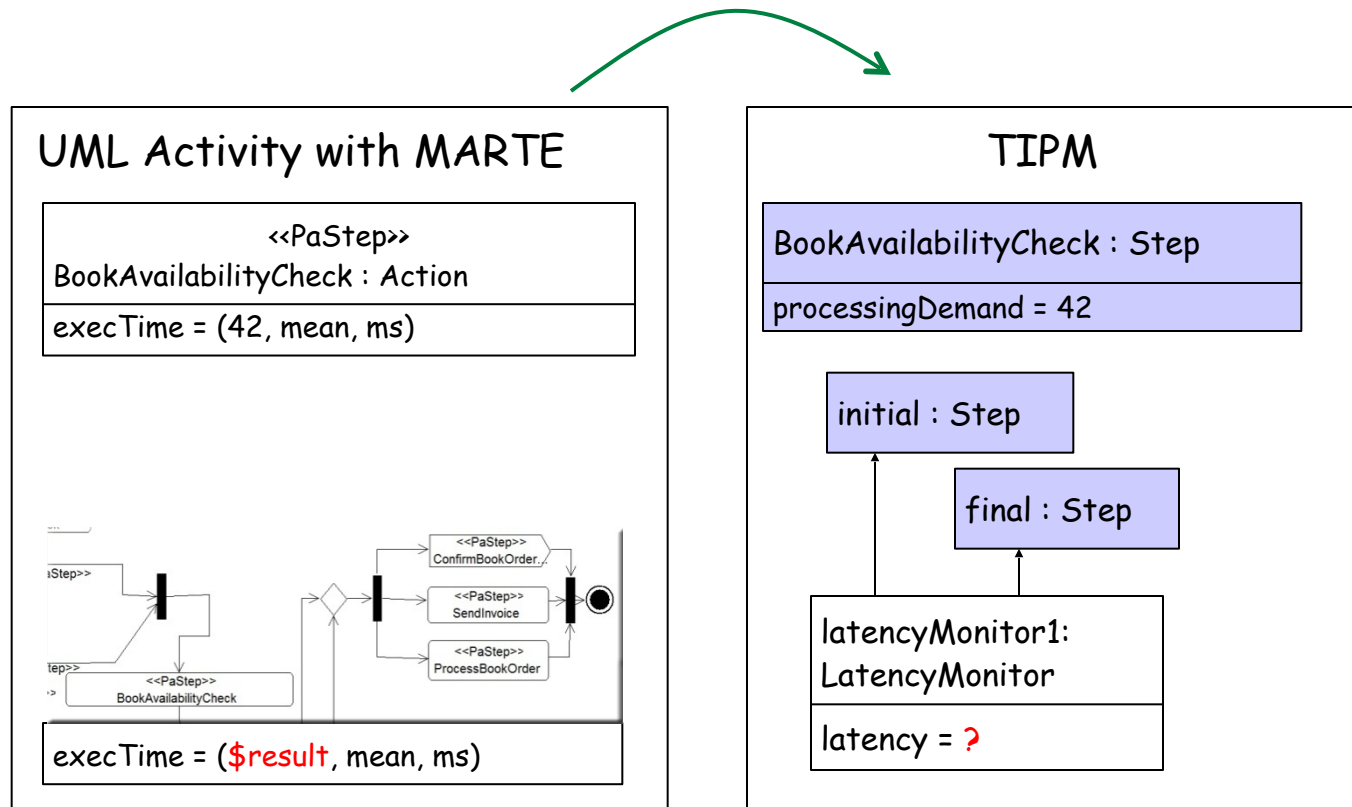
<<paStep>> <Activity> BookShopActivity

Model	Property	Value
Stereotypes	Duration	
	Energy	
Stereotype Attributes	Exec Time	
	Ext Op Count	(\$result, min, ms), (\$result, mean, ms), (\$result, max, ms)
Graphics	Ext Op Demand	
	Finish	
Advanced	Host	

The Exec Time feature of type NFP\_Duration

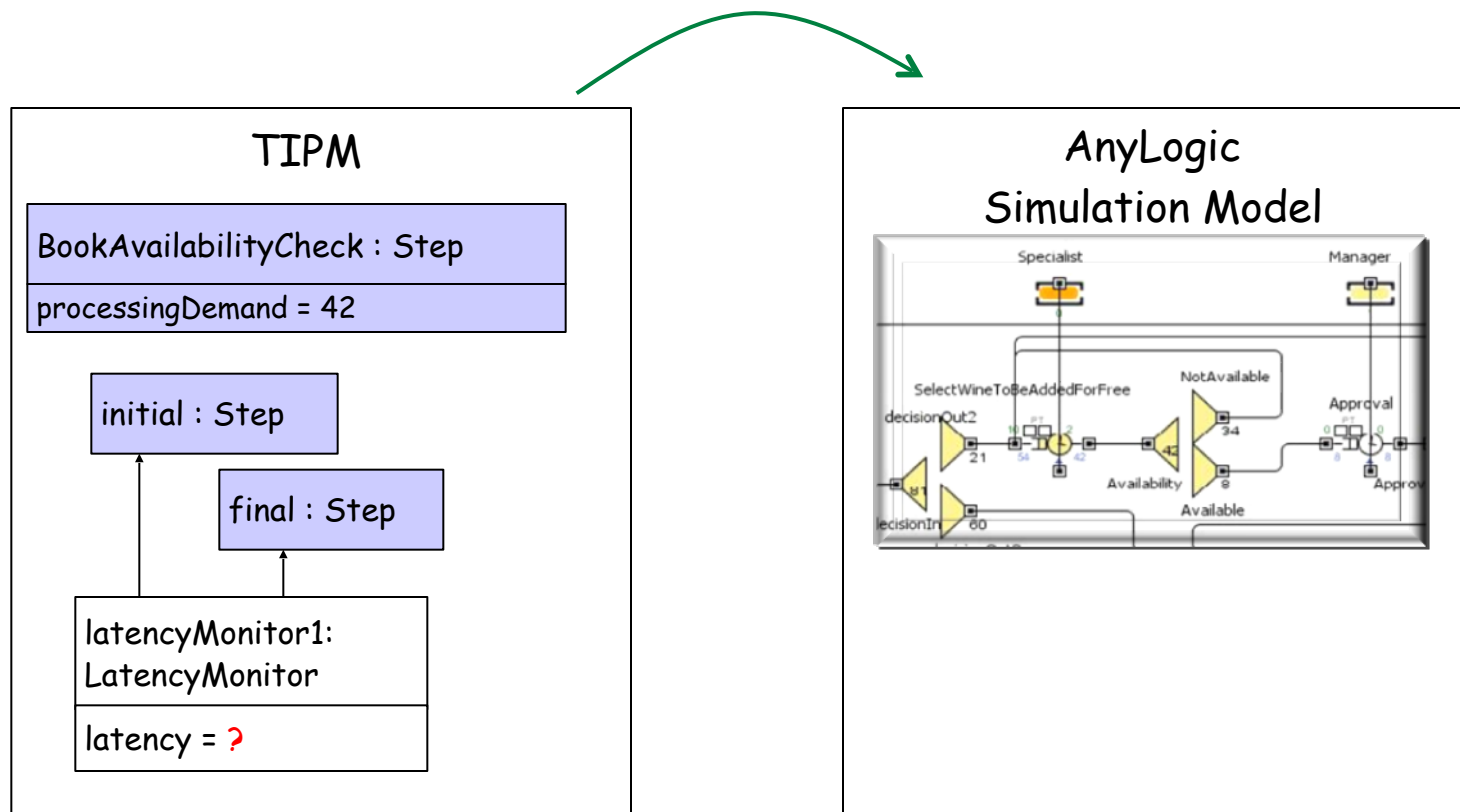
# MDPE process background: Transforming Activities to TIPM

- 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



# 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

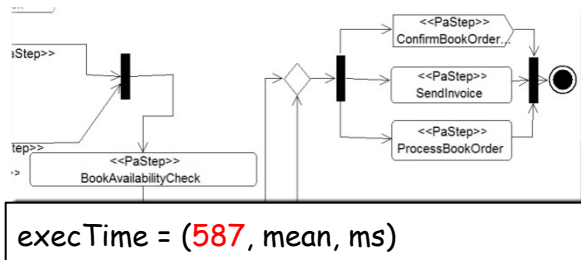
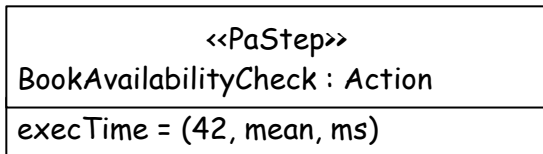


TSPM ... Tool Specific Simulation Model

# 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



## TIPM

<pre> BookAvailabilityCheck : Step processingDemand = 42 </pre>
---

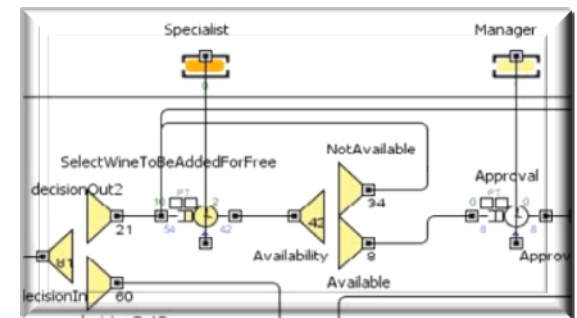
<pre> initial : Step </pre>
-----------------------------

<pre> final : Step </pre>
---------------------------

<pre> latencyMonitor1: LatencyMonitor </pre>
--

<pre> latency = 587 </pre>
----------------------------

## AnyLogic Simulation Model



<pre> Result propagation service </pre>
---

# Visualizing simulation results in the development model

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

File Edit Navigate Search Project Run Window Help

Simulate Model

150%

Topcased M... ATL

BookOrderProcessing.umlidi

activity BookShopActivity

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

```

    graph TD
      Start(( )) --> CreateBookOrderIn[<<PaStep>> createBookOrderIn]
      CreateBookOrderIn --> Merge1{ }
      ChangeBookOrderIn[<<PaStep>> changeBookOrderIn] --> Merge1
      Merge1 --> CancelBookOrderIn[<<PaStep>> cancelBookOrderIn]
      CancelBookOrderIn --> Merge2{ }
      Merge2 --> CustomerDataDataCheck[<<PaStep>> CustomerDataDataCheck]
      CustomerDataDataCheck -- Yes --> Merge3{ }
      Merge3 --> CreditCardCheck[<<PaStep>> CreditCardCheck]
      CreditCardCheck -- Yes --> Merge4{ }
      Merge4 --> BookAvailable[<<PaStep>> BookAvailable]
      CreditCardCheck -- No --> Merge5{ }
      Merge5 --> End(( ))
  
```

Properties Problems Documentation Console

<<paStep>> <Activity> BookShopActivity

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