

Tulsa: A Tool for Transforming UML to Layered Queueing Networks for Performance Analysis of Data Intensive Applications

Chen Li

Department of Computing
Imperial College London
chen.li1@imperial.ac.uk

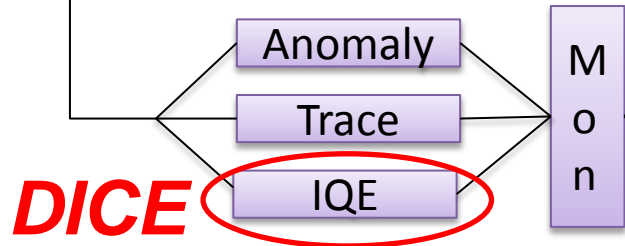
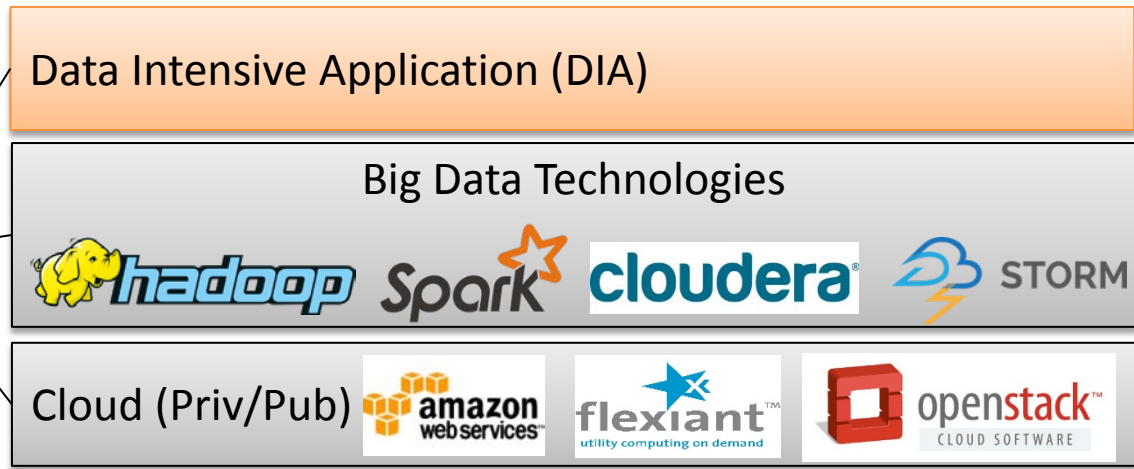
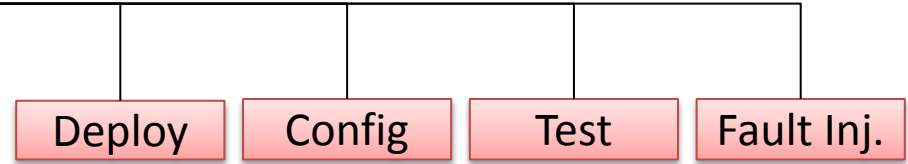
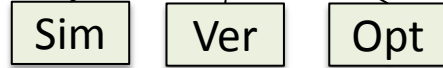
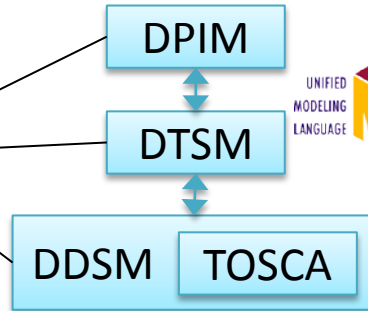
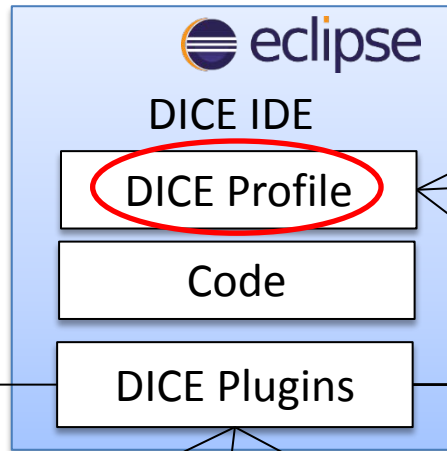
Joint work with:

Taghreed Altamimi, Mana Hassanzadeh Zargari, Giuliano Casale,
and Dorina Petriu

Motivation

- Model-Driven Development (MDD)
 - Changes the focus from code to models
 - Facilitate the non-functional properties analysis
- Problems
 - Reason on the quality of an application design based on runtime info.
 - Identify refactoring actions and provide timely feedback
- Motivation
 - Reflecting the runtime monitoring performance information back into design time models
 - Detecting software performance anti-patterns

Background

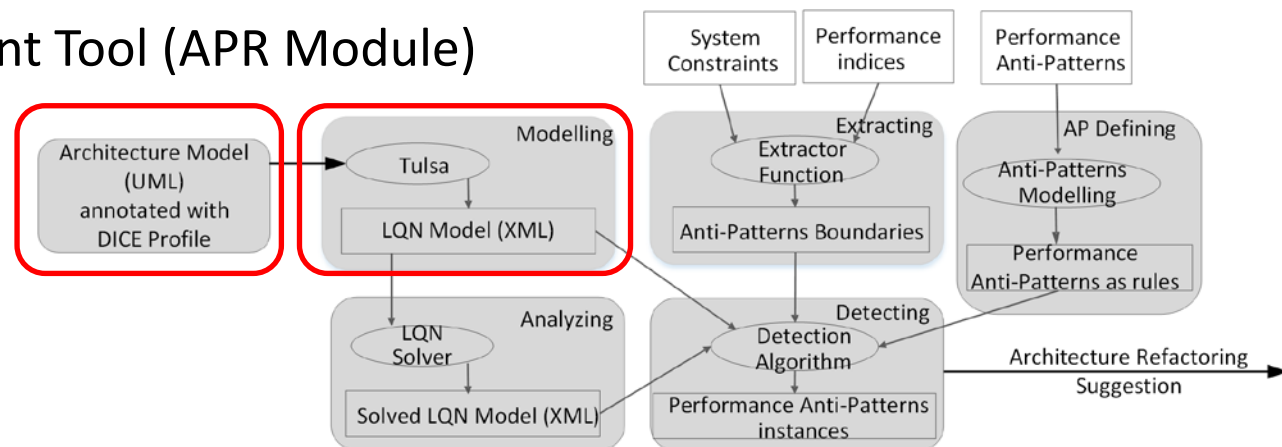


DICE Framework

Objectives

- Objectives

- Transforming UML diagrams annotated with UML profiles to Layered Queueing Networks (LQN)
- Using a family language, Epsilon, to implement model transformation
- Generating a LQN model understood by the existing LQN solvers
 - LQNS
 - LINE
- DICE Enhancement Tool (APR Module)

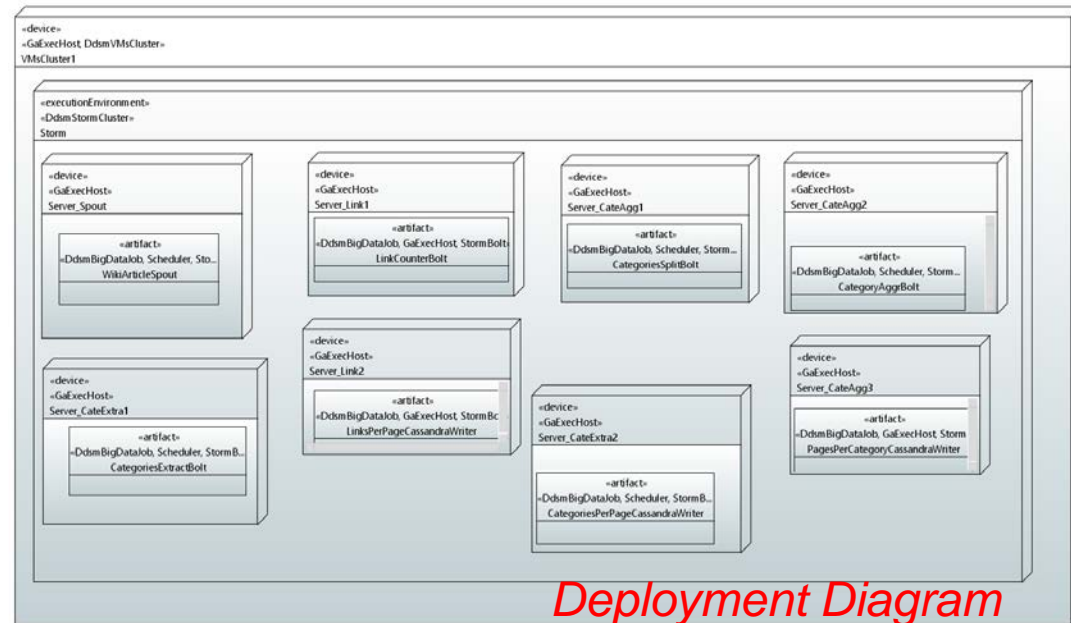
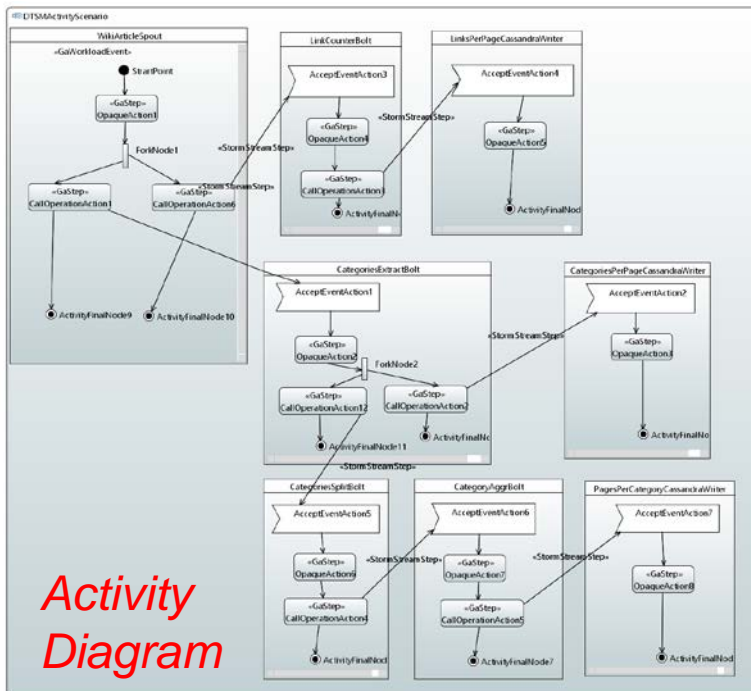


UML Profiles

- MARTE Profile
 - Stereotypes, e.g., <<GaExecHost>>, <<Scheduler>>, <<GaStep>>
 - Tags, e.g., resMult, schedPolicy, prob
- DICE Profile
 - DICE Platform Independent Model (DPIM)
 - Define a high-level topology
 - DICE Technology Specific Model (DTSM)
 - Refinement of the DPIM layer
 - Encompasses technological decisions
 - DICE Deployment Specific Model (DDSM)
 - Specify deployment decisions on cloud infrastructures
- *E.g. Storm Scenario
 - Stereotypes, e.g., <<StormSpout>>, <<StormBolt>>
 - Tags, e.g., parallelism, hostDemand

Source Model

- UML Model
 - Two Diagrams
 - Activity Diagram - Define the behavior of the system
 - Deployment Diagram - Specify the system configuration



Target Model

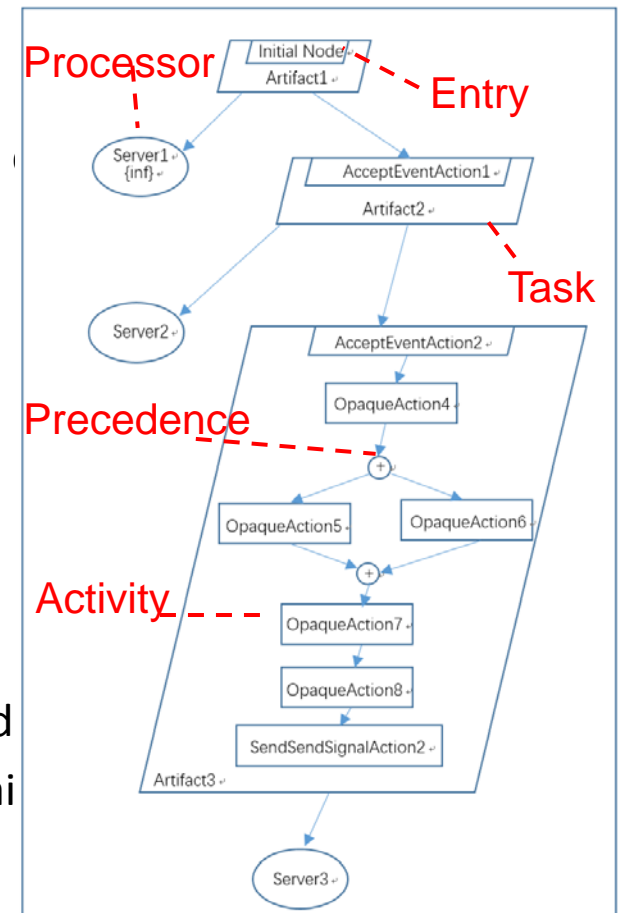
- Layered Queueing Networks

- What?

- Analytical performance models used to capture software layers
- Core elements
 - Processor
 - Task
 - Entry
 - Activity
 - precedence

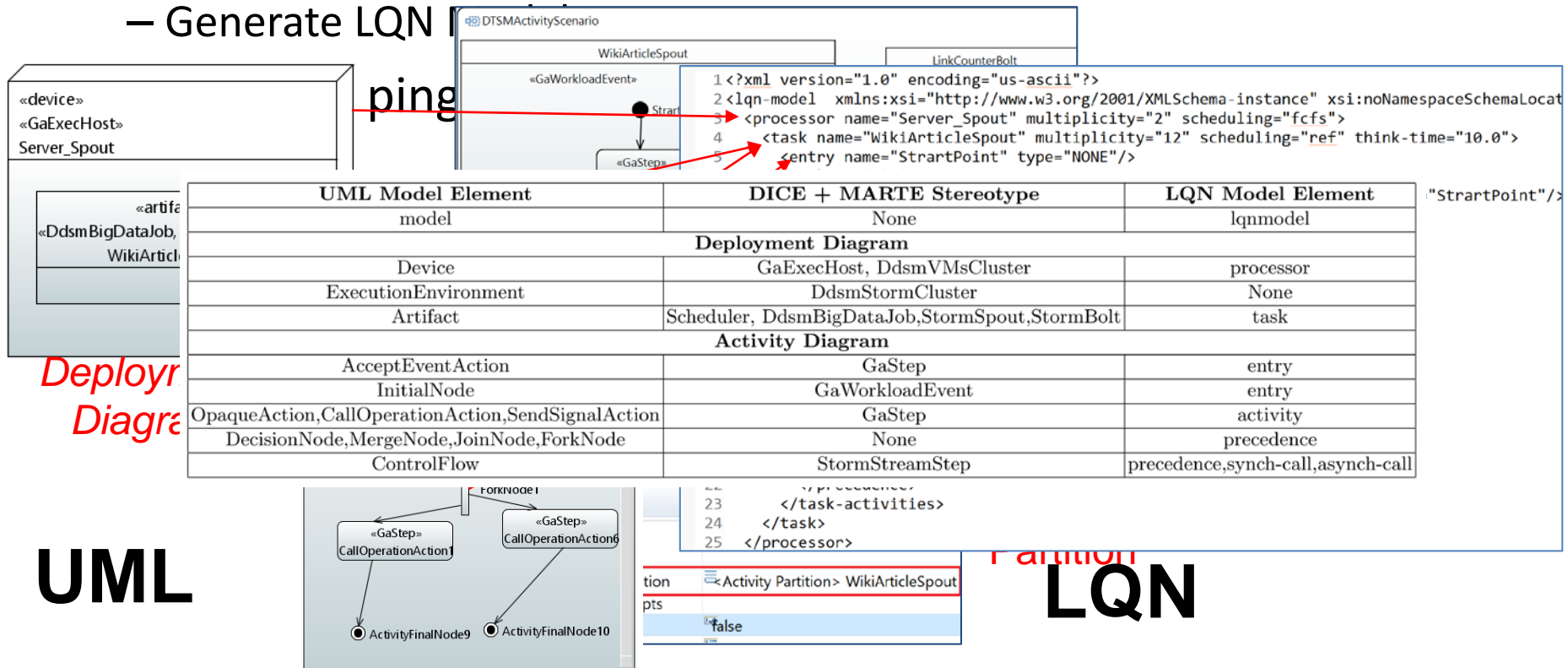
- Why?

- Semantic similarity – core elements of LQN model
- Available solvers – such as LINE or LQNS are available for analysis



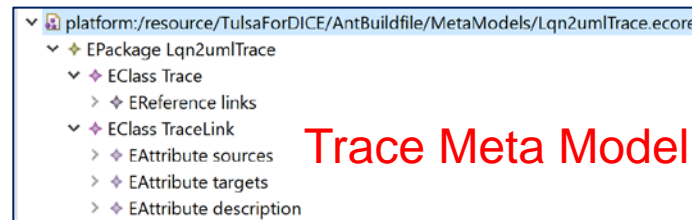
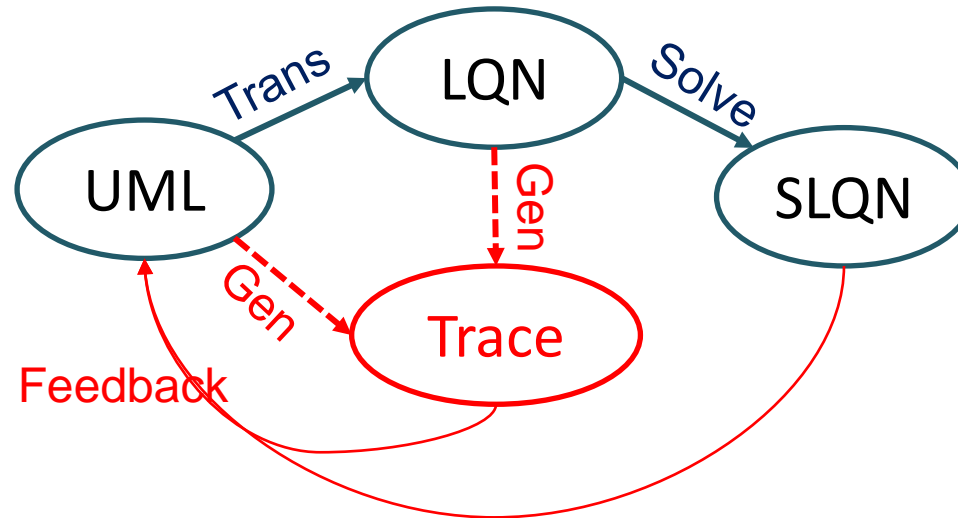
Transformation Process

- Transformation Design
 - Refine UML Model
 - Generate LQN



Transformation Process

- Trace Model



```

1 <?xml version="1.0" encoding="ASCII"?>
2 <Lqn2umlTrace:Trace xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:Lqn2umlTrace="Lqn2umlTrace">
3   <links sources="RootElement" targets="RootElement To lqn" description="Transformed by Model2Lqnmodel"/>
4   <links sources="Server_Spout" targets="Server_Spout" description="Transformed by Device2processor"/>
5   <links sources="WikiArticleSpout" targets="WikiArticleSpout" description="Transformed by Artifact2task"/>
  
```

Conclusion & Future Work

- What was achieved
 - Introduced DICE profile to annotate UML model for DIAs
 - Developed a transformation from UML+DICE+MARTE to LQN
 - Generated XML format LQN model understood by the existing solvers
- Limitations
 - Sensitive to the input model assumptions
 - Some features are not implemented
- Future Work
 - Develop pre-transformation verifications of the input model
 - Use cross-model traceability for the incremental propagation of changes
 - Extend the transformation to supports more DIAs

Thank You for your attention!