Modeling Execution Architecture of Software System with Colored Petri Nets

Jianli Xu and Juha Kuusela

Nokia Research Center

Presentation at WOSP'98

© NOKIA

Architecture modeling

- **Architecture of a software system**
  - defines the system in terms of components and interactions among those components
  - provides a global view of the system
- **Architecture model**
  - is a formal or semi-formal specification of one (or a group of) system wide property of the architecture
  - has to be complete in respect to that property
  - is used to analyze, predict and control that property

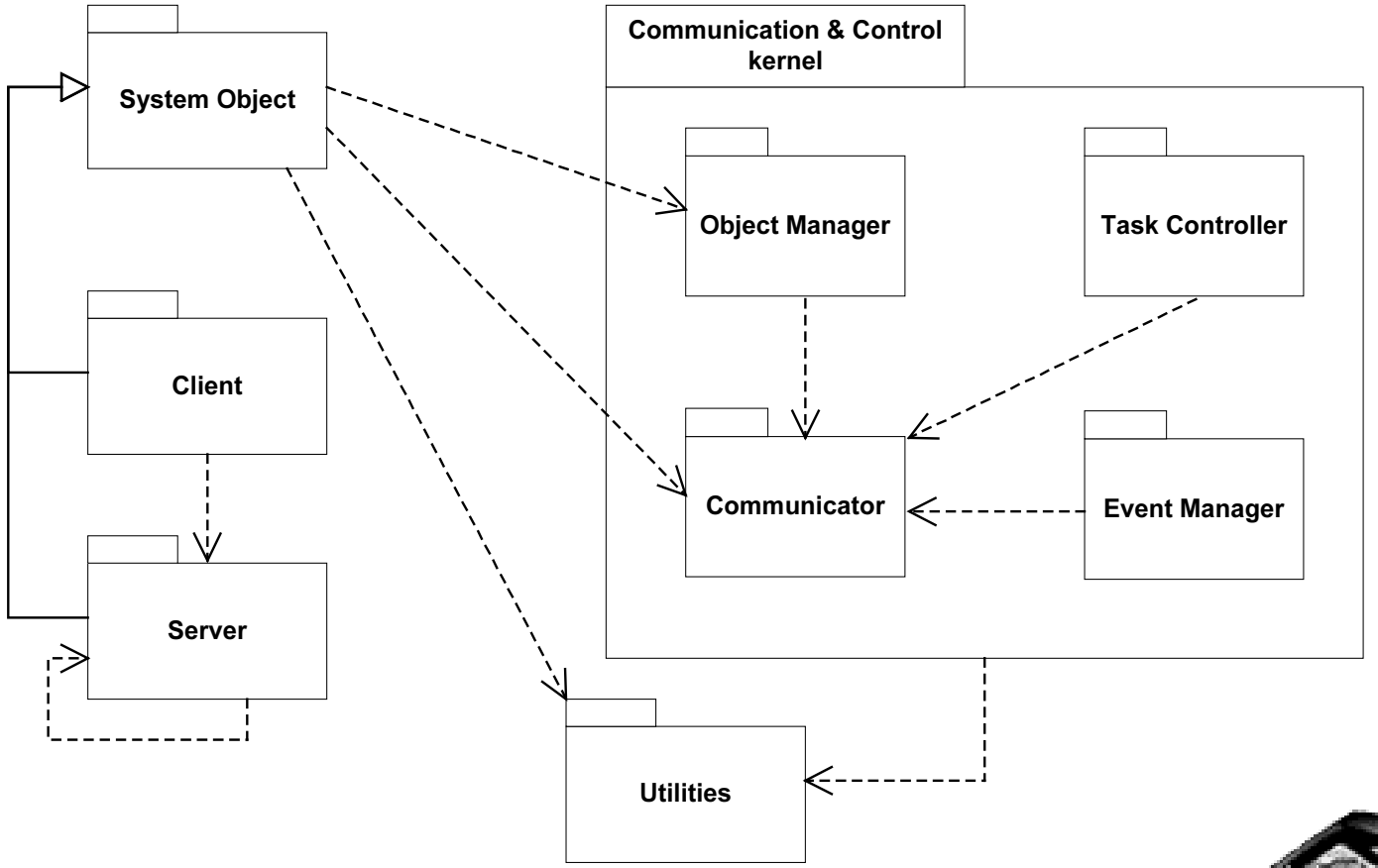
Architectural modeling from industrial perspective

- Modeling techniques should be classified according to the reasoning they can support
- Pre-made models should exist for different architectural constructs allowing analysis of different properties
- Models should be composable
- Model-checking techniques should be more practical
- Handbooks on how to model different architectural constructs should exists

Why Colored Petri Nets?

- A formal and executable model is needed to verify dynamic properties and estimate the performance accurately
- CPN has two characteristics which are of special interest for architecture modeling:
  - flexibility of token definition and manipulation
  - compositional way of model building
- Tool support
  - Design/CPN

New software for mobile phone product family



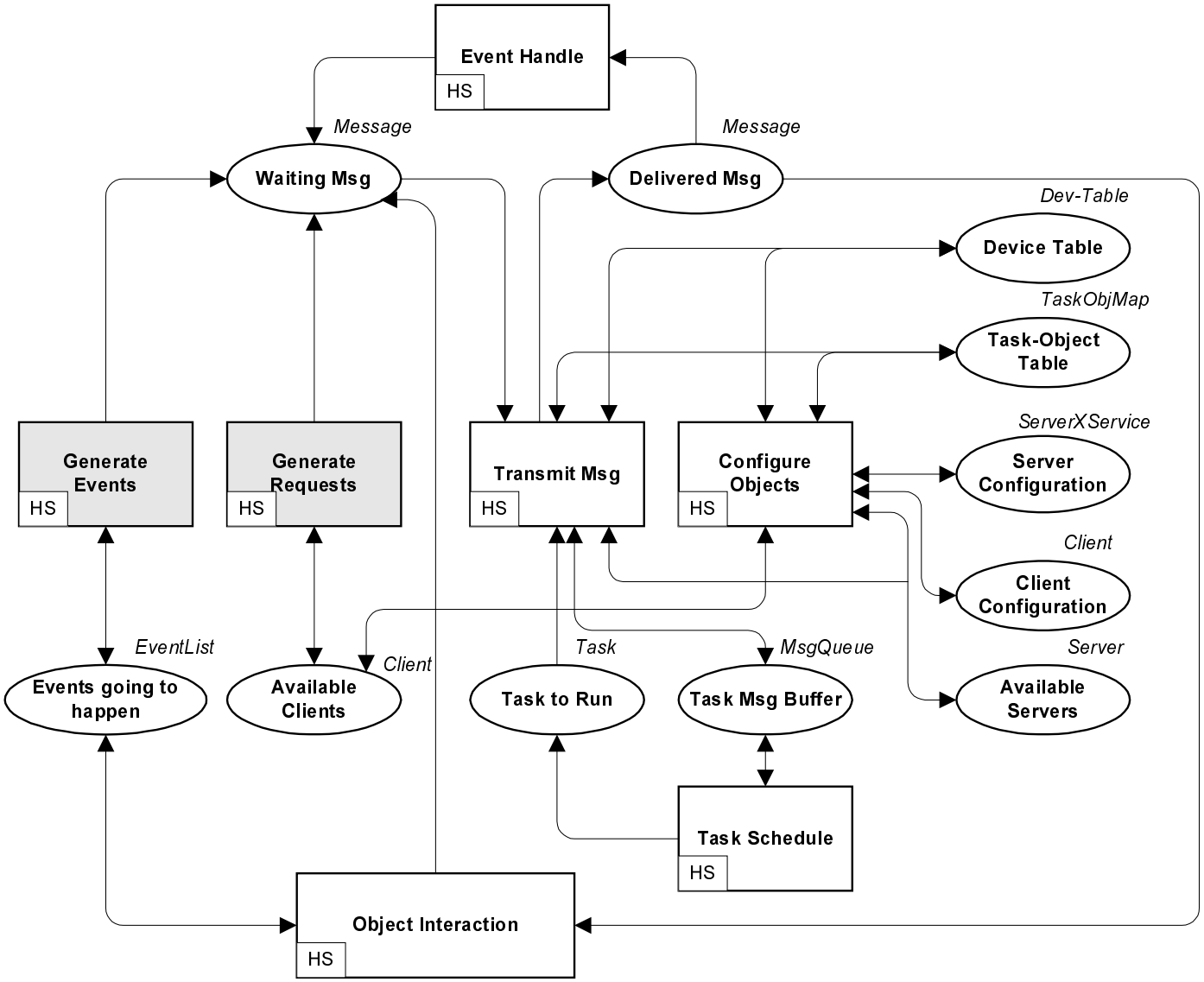
What we expect from the CPN model

- Specify and verify the task control mechanism
- Specify and verify the task communication mechanism
- Evaluate different task divisions and allocations
- Simulate typical use cases
- Estimate message buffer usage and message delays
- and more

Creating the CPN model

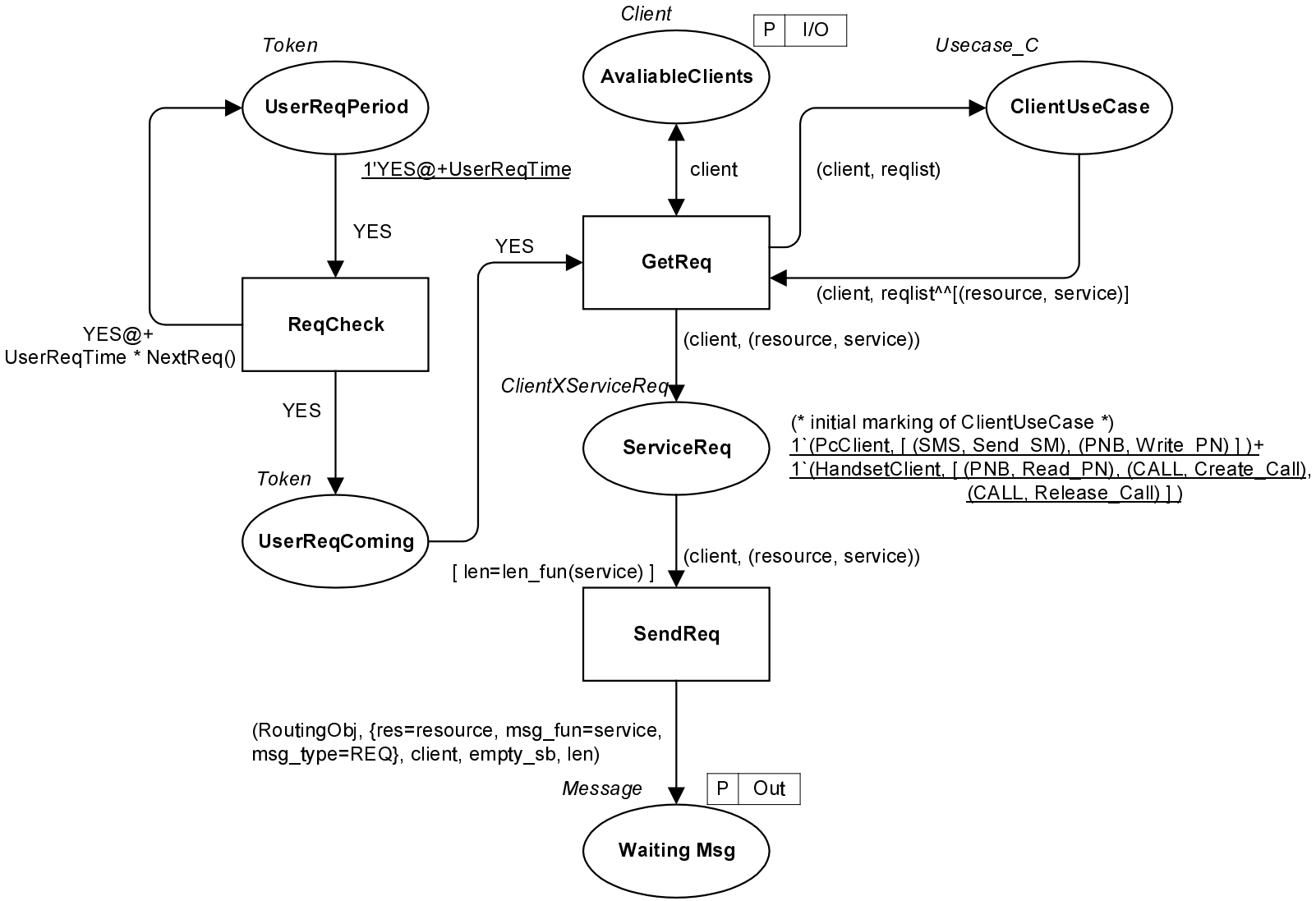
- Basic system elements are defined by types of tokens. Compositions of the basic types are used to describe higher level tokens
- Object interactions, task scheduling and message transmission mechanisms are described using CPN structure and inscriptions

Top page of the CPN model



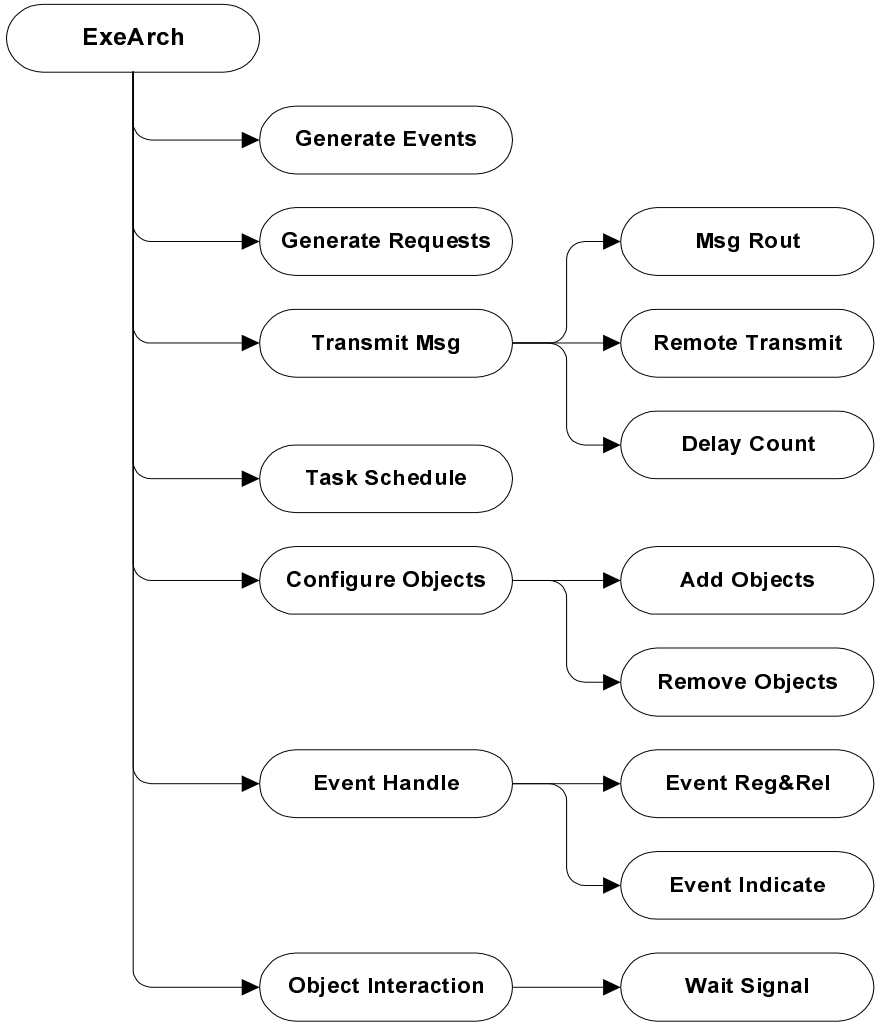
© NOKIA

Use cases are important to the model



© NOKIA

Model hierarchy

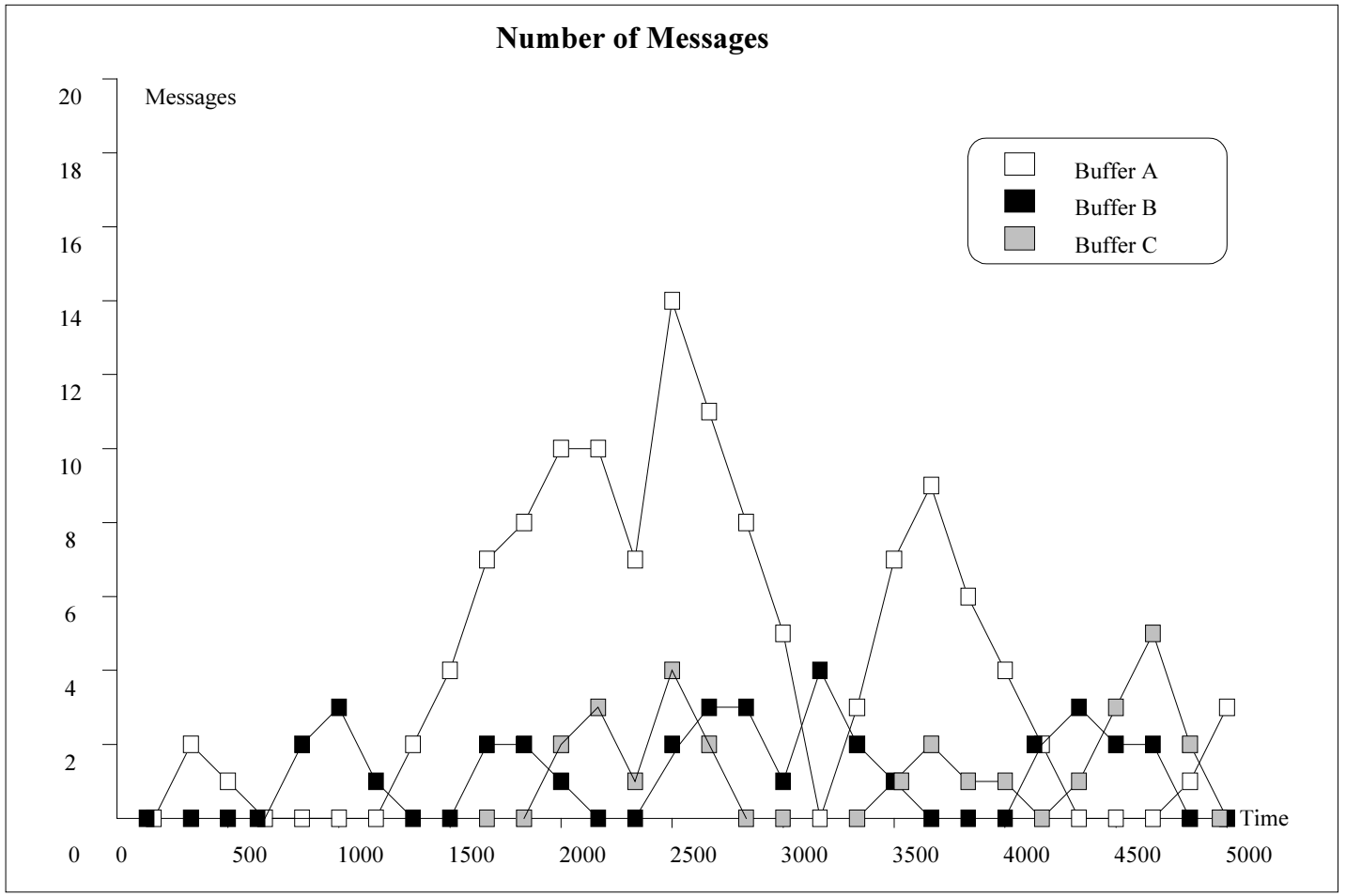


- The model contains 16 pages about 100 transitions and 160 places and more than 60 color sets

Simulation

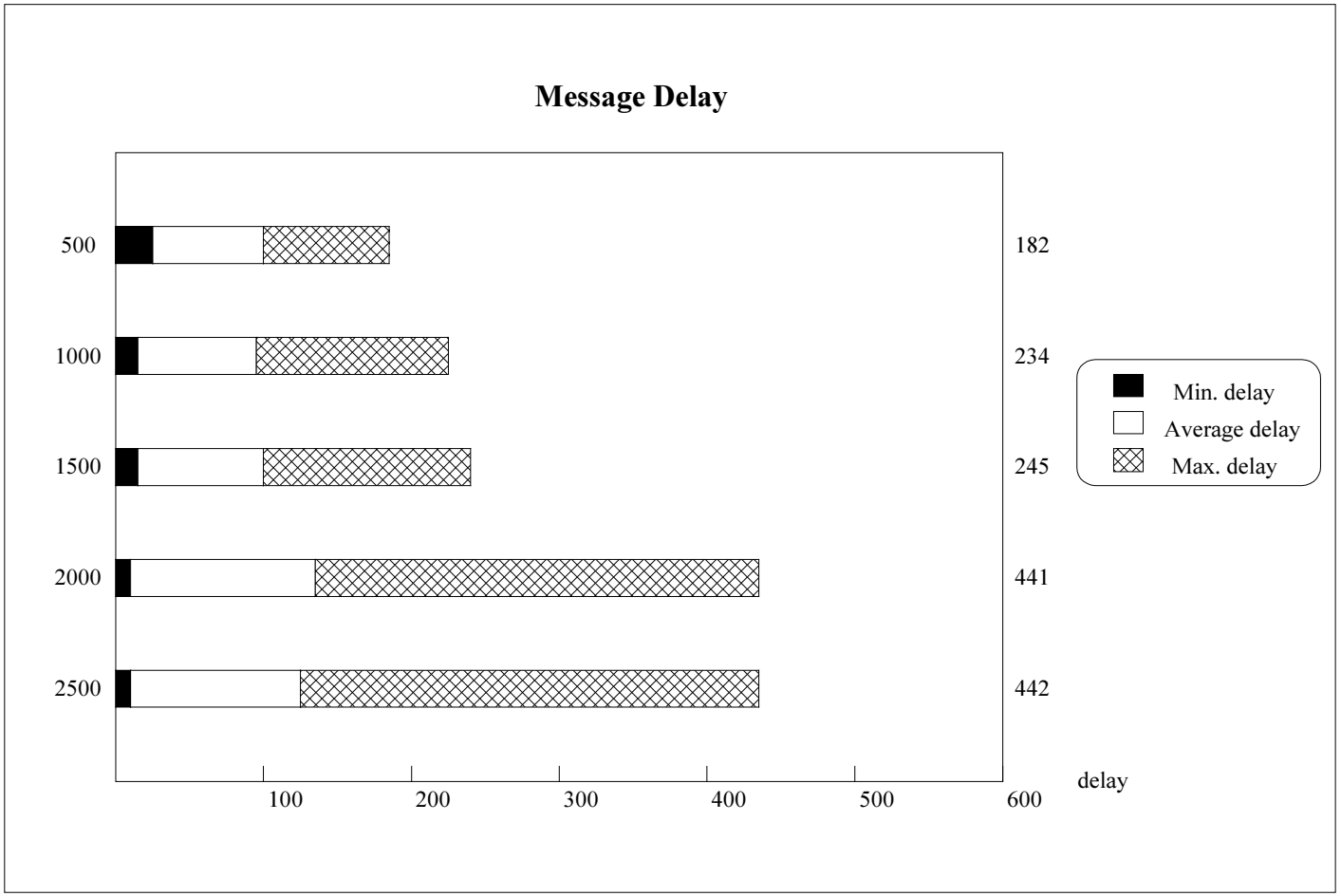
- Simulation without time
  - debug and revise the model
- Simulation with time
  - message delays on different message links
  - task-switching time of OS
  - event processing time
- Results
  - message buffer usage on the worst case
  - minimum, average and maximum message delays
  - number of task switches needed for each transaction
  - potential problems of the new system have been found

Example of statistic charts (1)



Line chart of message buffer usage

Example of statistic charts (2)



History chart of message delays

Formal analysis

- Only exercises on part of the system

Conclusion

- *Achievements*
  - Requirements placed on the execution architecture have been checked
  - Alternative mechanisms and policies have been evaluated
  - Modeling provided valuable feedback to system development
  - CPN and Design/CPN tool are fit for architectural modeling of industrial systems
- *Problems*
  - lack of documentation on how to use the modeling techniques
  - lack of experience on modeling of different architectural structure
  - how to use formal analysis techniques on large system is unknown to us