# Simulating Stochastic Processes with OMNeT++

Jan Kriege, Peter Buchholz

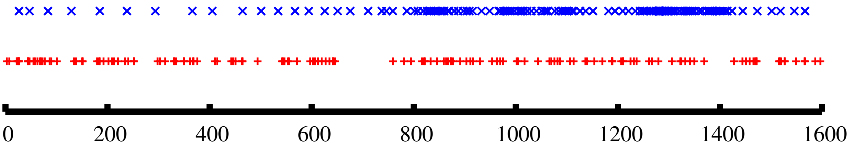
Department of Computer Science,
TU Dortmund

March 21, 2011

# Outline

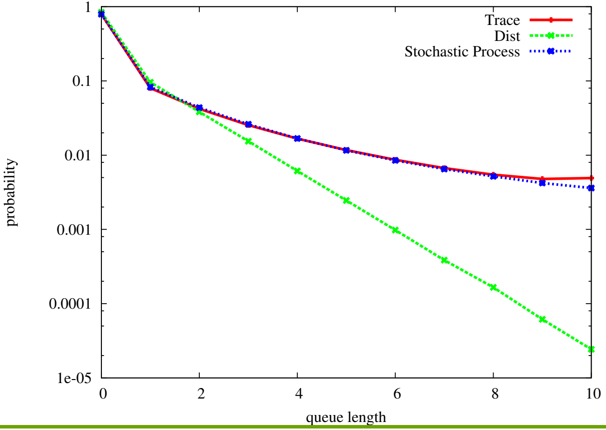
# Introduction

- Traffic processes in computer networks include dependencies and correlation
- Modeling with Poisson processes or even more complex interarrival time distributions is not sufficient
- Neglection of correlation may result in a dramatic underestimation of resource requirements



# Motivation

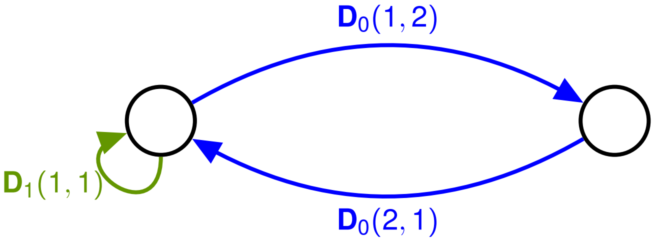
Performance of a single server queue with correlated and uncorrelated arrivals



# Markovian Arrival Processes (MAPs)

- two $n \times n$ matrices $(\mathbf{D}_0, \mathbf{D}_1)$
- $\mathbf{D}_0$: rates of transitions without arrival
- $\mathbf{D}_1$: rates of transitions generating an arrival

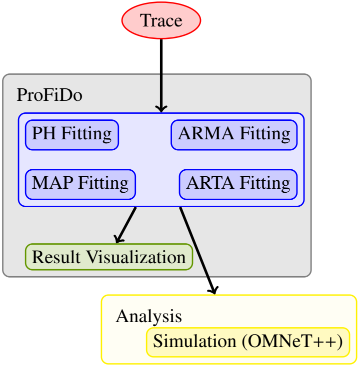
- $\mathbf{D}_0(i,j) \geq 0$ for $i \neq j$
- $\mathbf{D}_0(i,i) \leq -\sum_{j=1, j\neq i}^{n} \mathbf{D}_0(i,j)$
- $\mathbf{D}_1 \geq 0$



# Motivation

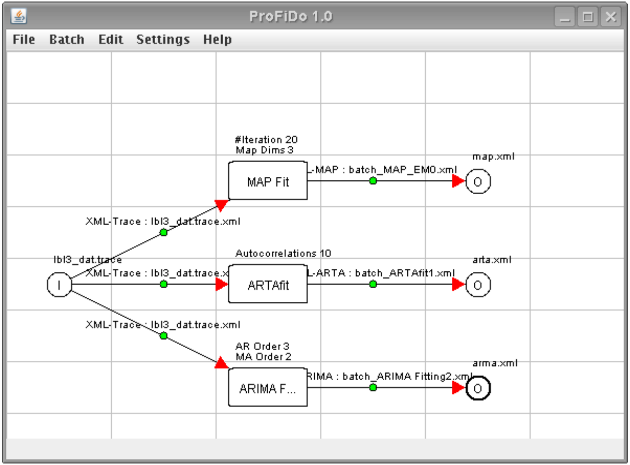
- Little support for stochastic processes in simulation literature
- Simulation software often limited to distributions
- Use of correlated arrival streams is prohibited by missing tool support to generate arrival process specifications from measured data and by missing support to represent arrival processes in simulation tools
- $\Rightarrow$ Framework to support stochastic processes in OMNeT++ simulation models

# ProFiDo - Processes Fitting Toolkit Dortmund

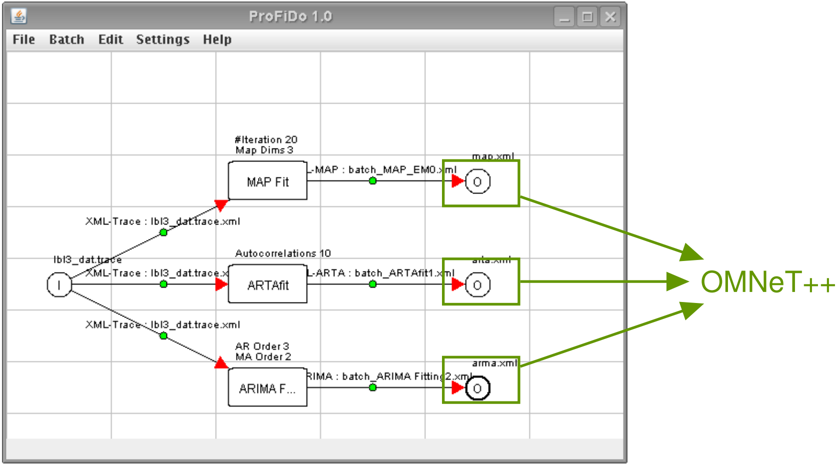



- flexible Java-based toolkit for consistent use of commandline-oriented fitting tools
- fitting of stochastic processes: choose parameters such that characteristics of trace are matched
- visualization of properties
- workflows to realize different steps of data preprocessing, parameter fitting and analysis of stochastic processes

# ProFiDo - Processes Fitting Toolkit Dortmund

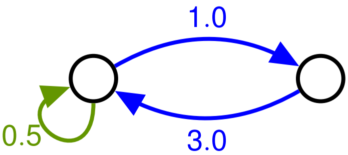


# ProFiDo - Processes Fitting Toolkit Dortmund



# ProFiDo - XML Interchange Format

- XML interchange format for description of stochastic processes
- ensures interoperability of different fitting tools in a workflow




**XML description**

```
<map>
  <states>2</states>
  <d0>
    -1.5  1.0
     3.0 -3.0
  </d0>
  <d1>
    0.5 0.0
    0.0 0.0
  </d1>
</map>
```

# OMNeT++ Arrival Process Module

- simple module that can generate random numbers from stochastic processes
- model description is parsed from file in XML interchange format

## NED description

```
simple ArrivalProcess
  parameters:
    xml model;
    string transform = default("");
    @display("i=block/source");
  gates:
    output out;
```

# OMNeT++ Arrival Process Module

- simple module that can generate random numbers from stochastic processes
- model description is parsed from file in XML interchange format

## NED description

```
simple ArrivalProcess
  parameters:
    xml model;
    string transform = default("");
    @display("i=block/source");
  gates:
    output out;
```

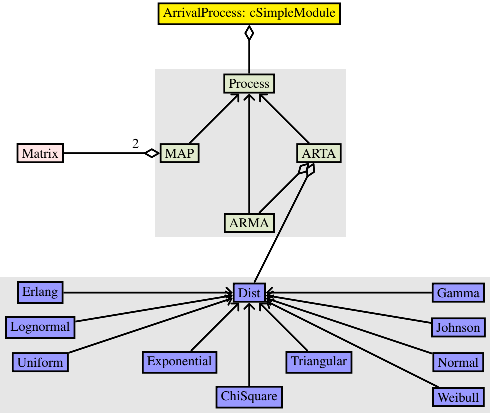
# OMNeT++ Arrival Process Module

- simple module that can generate random numbers from stochastic processes
- model description is parsed from file in XML interchange format

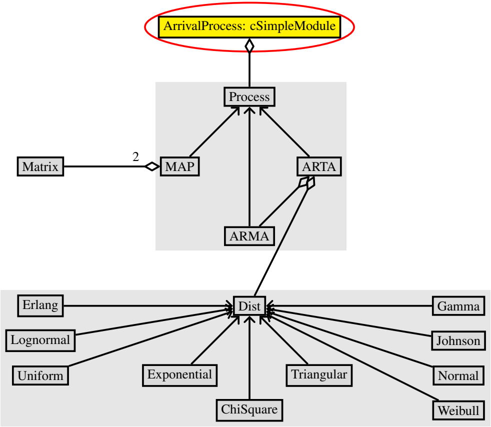
## NED description

```
simple ArrivalProcess
  parameters:
    xml model;
    string transform = default("");
    @display("i=block/source");
  gates:
    output out;
```

# OMNeT++ Arrival Process Module - Class Hierarchy



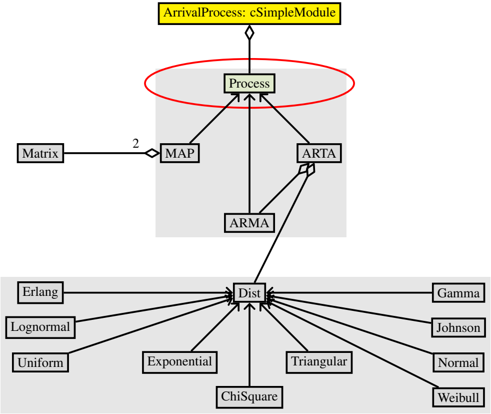
# OMNeT++ Arrival Process Module - Class Hierarchy




## ArrivalProcess

- load process description from XML file
- initialize Process
- deal with message events: handleMessage()

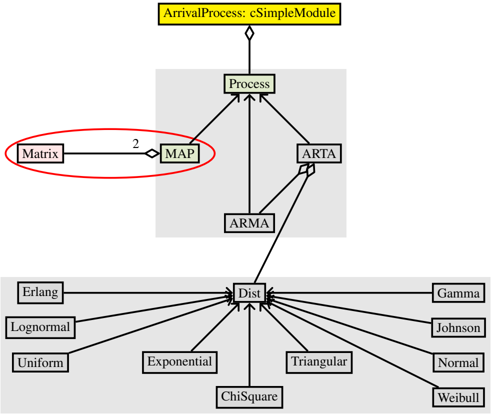
# OMNeT++ Arrival Process Module - Class Hierarchy




## Process

- abstract base class for stochastic processes
- getNextRandomVariate(): implemented in inheriting classes

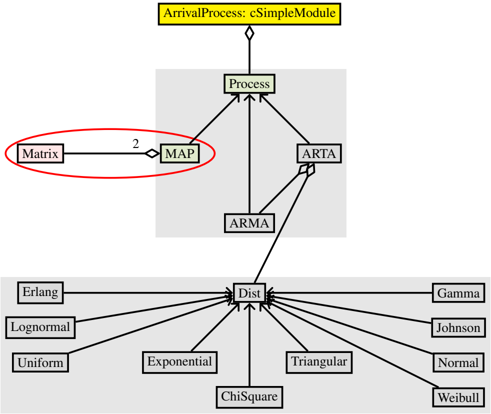
# OMNeT++ Arrival Process Module - Class Hierarchy




## MAP

- draw random numbers from Markovian Arrival Processes
- Simulation of the underlying Markov chain
- Utility class Matrix to store matrices $\boldsymbol{D}_0$ and $\boldsymbol{D}_1$

# OMNeT++ Arrival Process Module - Class Hierarchy

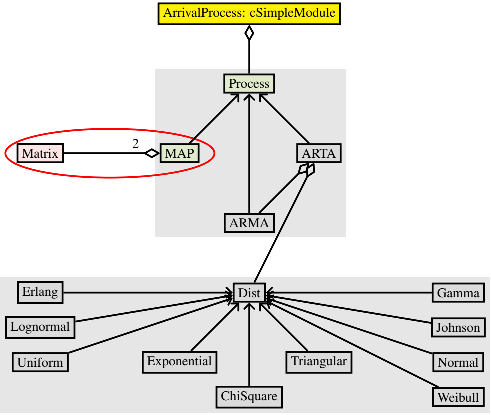



## MAP

### Initialization

- draw initial state from the distribution defined by $\pi$ (stationary distribution just after an arrival)
- $\pi$ is the unique solution of $\pi(-D_0^{-1}D_1) = \pi$ normalized to 1

# OMNeT++ Arrival Process Module - Class Hierarchy

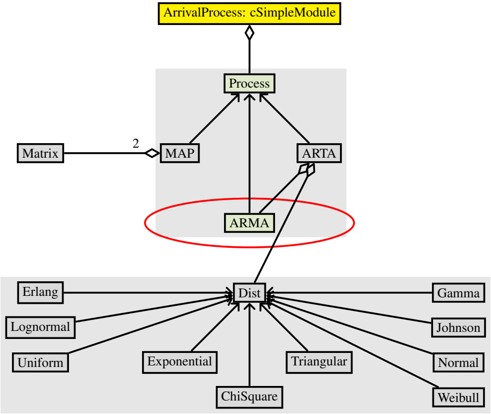



## MAP

### Simulation

- next transition time: exponentially distributed with rate $|D_0(i,i)|$
- next state: uniformly distributed according to $D_0(i,j)/|D_0(i,i)|$ and $D_1(i,j)/|D_0(i,i)|$
- Transition from $D_1$: Generate arrival $\Rightarrow$ return sum of transition times

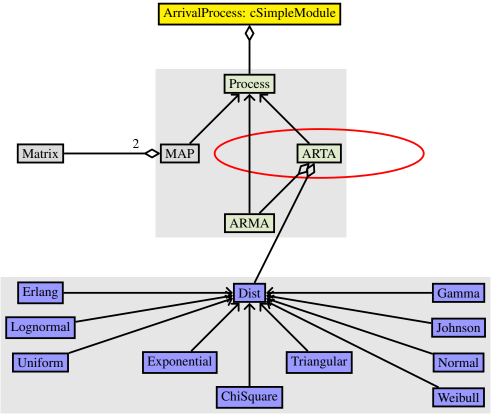
# OMNeT++ Arrival Process Module - Class Hierarchy




## ARMA

- simulation of Autoregressive Moving Average Processes
- initialization step to start in a stationary state
- simulation step to draw random numbers

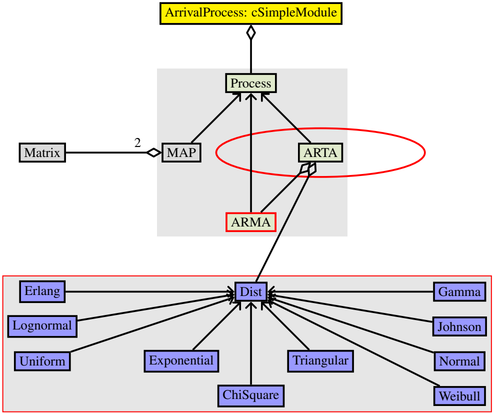
# OMNeT++ Arrival Process Module - Class Hierarchy




## ARTA

- simulation of Autoregressive To Anything Processes
- initialization step to start in a stationary state
- simulation step to draw random numbers

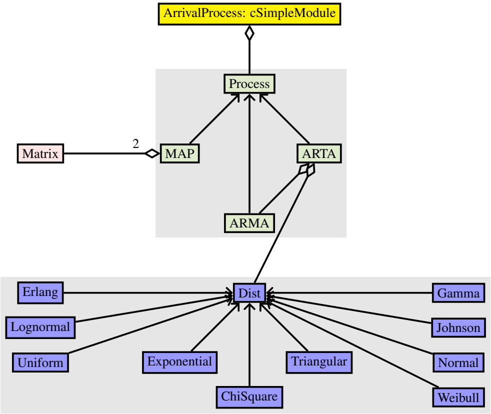
# OMNeT++ Arrival Process Module - Class Hierarchy




## ARTA

- simulation of Autoregressive To Anything Processes
- combination of ARMA process with arbitrary marginal distribution
- support for various different distributions

# OMNeT++ Arrival Process Module - Class Hierarchy



# Post-Processing of the Time Series

## Transformation of generated interarrival times

- Fitted input process uses a different time scale than the rest of the model
- Stochastic process (e.g. ARMA) might output invalid values

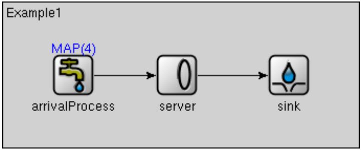
$\Rightarrow$ linear and non-linear transformations of the time series

- Specification using OMNeT++'s NED language expressions
- Transformation function is passed as parameter `transform` to Arrival Process module

# Application Examples

- Two application examples to show how the ArrivalProcess module can be incorporated into OMNeT++ models
- First example: simple queueing model
- Second example: modified NClients model from the INET Framework
- Simulation results support the observation that negligence of autocorrelation may have serious impact on simulation results.

# Example 1 - Queueing Model



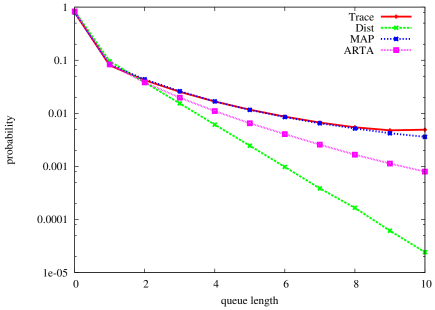

- different configurations of the model: MAP, ARTA, trace driven simulation, iid arrivals (Poisson process)
- different utilization levels for the server
- queue length distribution as result measure

## Configuration

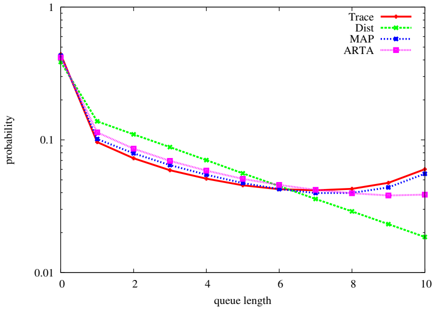
```
[General]
network = Example1
**.server.serviceTime = exponential(0.5s)
**.server.buffer = 10

[Config MAP]
description = "Arrivals from MAP"
**.arrivalProcess.model = xmldoc("map.xml")

[Config ARTA]
description = "Arrivals from ARTA process"
**.arrivalProcess.model = xmldoc("arta.xml")
```

# Example 1 - Queue Length Distribution

$\rho = 0.4$:

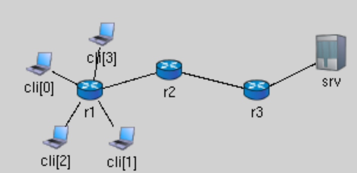



$\rho = 0.8$:

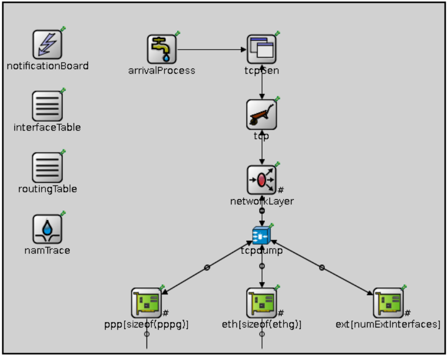


# Example 2 - NClients Model from INET Framework

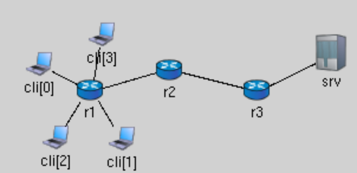
- Four client hosts connected to a server via different routers.



# Example 2 - NClients Model from INET Framework



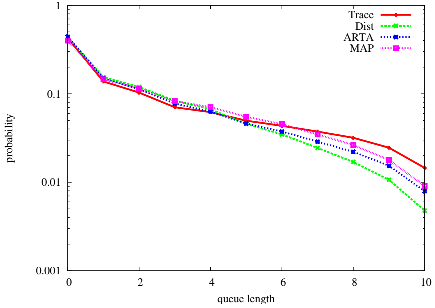
# Example 2 - NClients Model from INET Framework



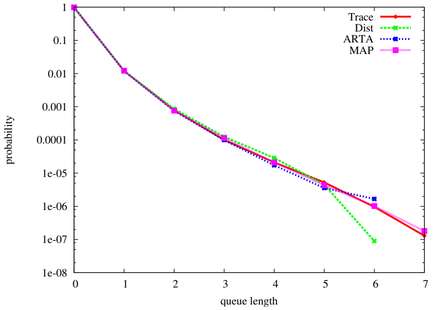

- Four different configurations:
  - Arrivals according to MAPs
  - Arrivals according to ARTA process
  - Trace driven simulation
  - Poisson process (iid arrivals)
- Queue length distribution of server's network interface and router interfaces as result measures.

# Example 2 - Queue Length Distribution

Server:




Router:



# Conclusions

- OMNeT++ module that can be used in simulation models as a traffic source.
- Support for stochastic processes with wide variety of marginal distributions.
- Random number generation according to ARMA processes, ARTA processes and MAPs.

# Conclusions

- Process description in XML format
- Module is linked to the toolkit ProFiDo for fitting stochastic processes.
- Application examples demonstrate the importance of incorporating autocorrelation into input models and how the new module can be used with existing models.

- ProFiDo and OMNeT++ Arrival Process Module freely available (GPL):
  $\Rightarrow$ `http://ls4-www.cs.tu-dortmund.de/profido`