RESTART Simulation of Colored Stochastic Petri Nets

TECHNISCHE UNIVERSITÄT ILMENAU

Armin Zimmermann
System and Software Engineering Group
TU Ilmenau, Germany

7th RESIM Workshop
Rennes, October 2008

Outline

- Introduction
- Colored stochastic Petri nets
- RESTART simulation of CSPNs
- TimeNET prototype implementation
- An example

Introduction

- Model-Based System Design
  - Complex, heterogeneous systems
  - Non-functional properties
  - Stochastic discrete event system model
  - Performability evaluation

  - Non-Markovian delays → simulation
  - Safety-critical applications → rare events

Introduction

- Rare Event Simulation
  - Speeds up generation of significant events
  - Many methods in the literature

- Necessary in many industrial applications
  - Fault tolerant, highly reliable, safety critical, ...
  - Avionics, automotive, train control, telecommunication, computer systems, ...

Introduction

- Obstacles to be solved
  - Background knowledge necessary for rare-event simulation techniques
    - e.g., importance function selection
  - Performance measure type limitations
    - e.g., only time until or probability of hitting a rare state set [RESIM2006]
  - Model class restrictions
    - e.g., Markov chains or queuing networks
  - Lack of software tools that are easy to use

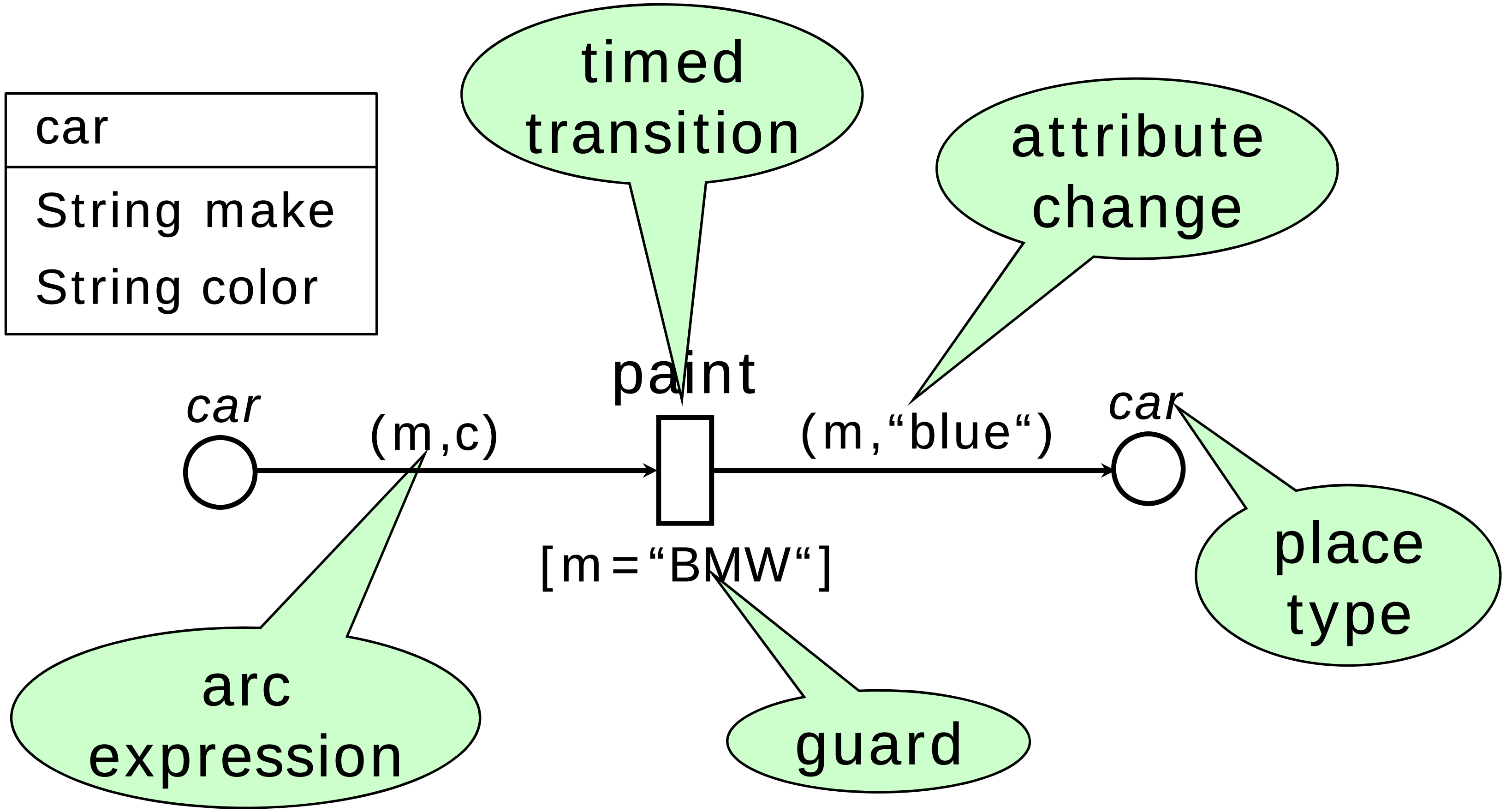
Outline

- Introduction
- Colored stochastic Petri nets
- RESTART simulation of CSPNs
- TimeNET prototype implementation
- An example

Colored Stochastic Petri Nets

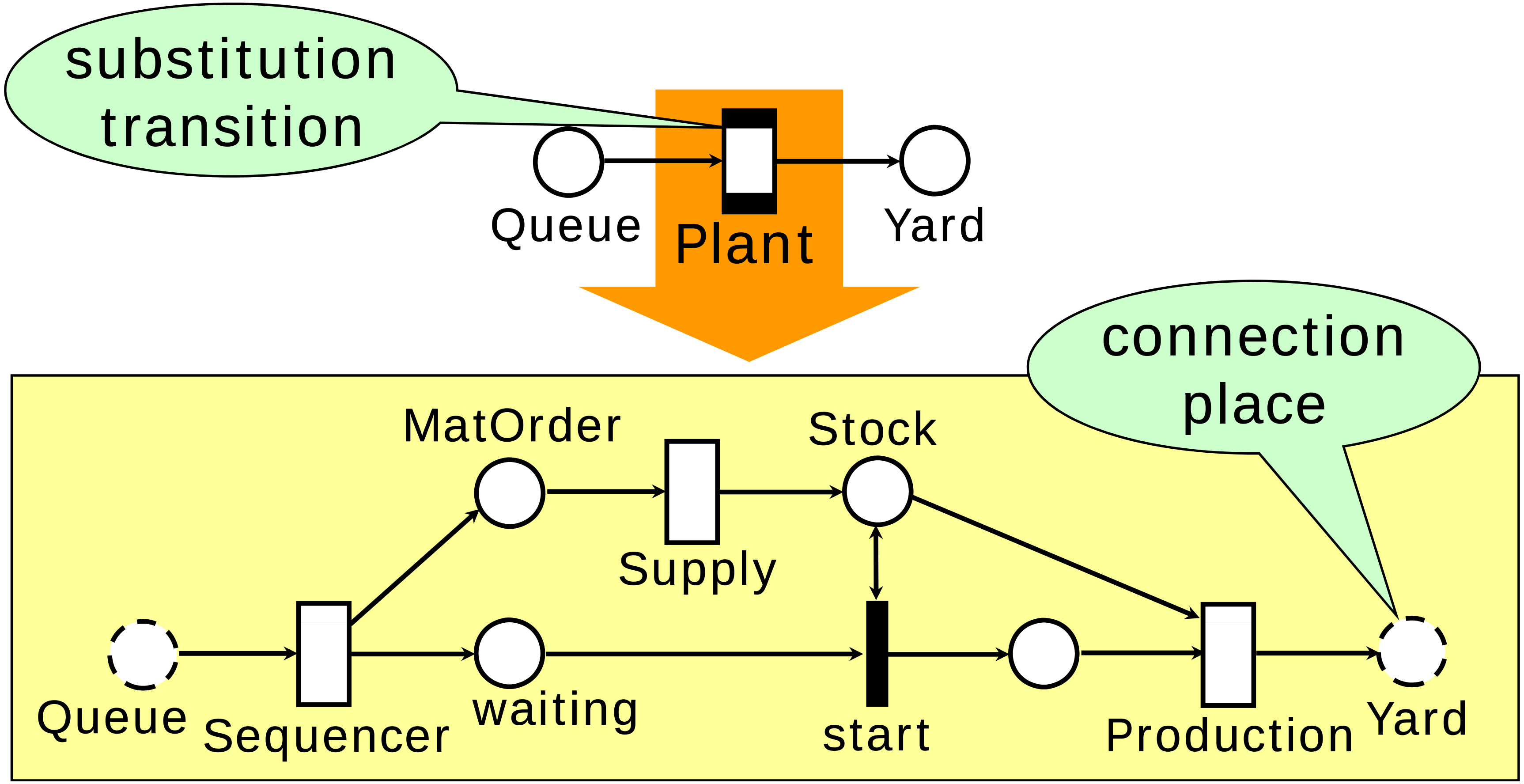
- Petri nets
  - Model of structure and behavior
  - Graphical representation and mathematical background
- Simple Petri nets
  - Not sufficient for complex applications (tokens not distinguishable)
- High-level Petri nets since early 80s

Colored Stochastic Petri Nets



Colored Stochastic Petri Nets

Hierarchical refinement and modularity

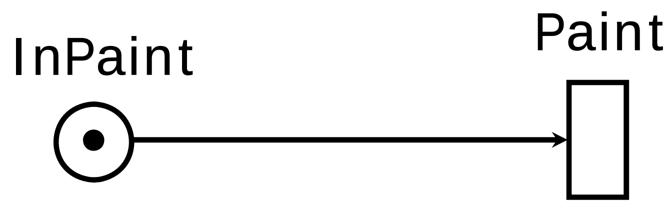


Colored Stochastic Petri Nets

- General performance measures
  - *reward variables* [Sanders, Meyer 1991]

Contribution in a state (*rate reward*)	Contribution of transition firing (*impulse reward*)

  - Example: **`-#InPaint`** **+** **`3*#Paint(make=“VW“)`**



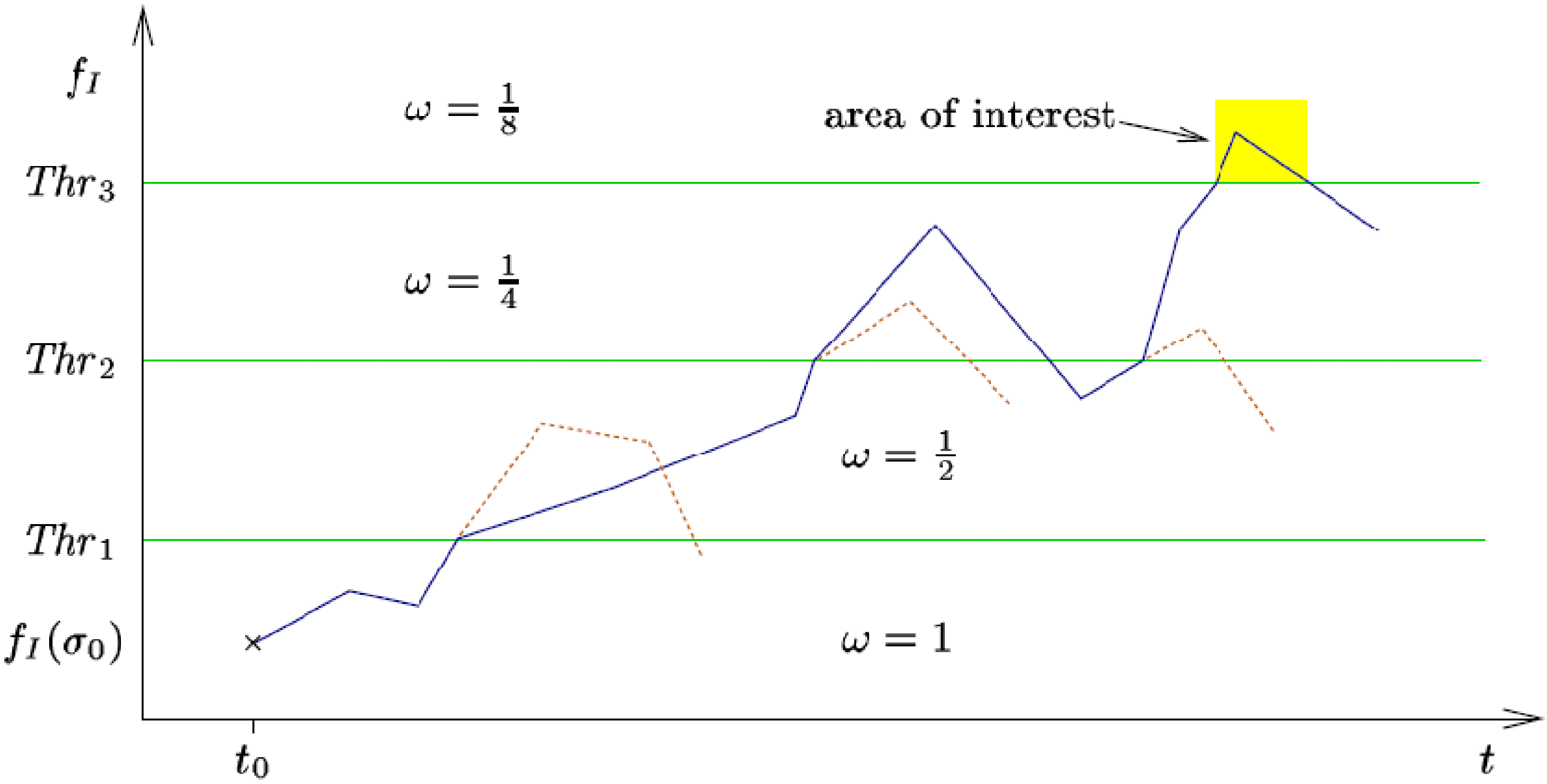
Outline

- Introduction
- Colored stochastic Petri nets
- RESTART simulation of CSPNs
- TimeNET prototype implementation
- An example

RESTART for CSPNs

- Fixed splitting
  - Thresholds and importance function
  - Result **`ImportanceFunction`** → integer levels
- Global step
- Weight-Based Variant
  - Each path has an individual weight according to its splitting history [Tuffin, Trivedi 2000]

RESTART for CSPNs



Levels and weights

RESTART for CSPNs

- Performance measures
  - Instantaneous reward at time *t*

$$R_{inst}(t) = r_{rate}\left(\sigma(t)\right) + \Delta \sum_{e \in E(t)} r_{imp}(e)$$

  - Simulation estimator for RESTART

$$\widehat{rv(\mathrm{S})} = \frac{1}{T} \int_{0}^{T} \omega(t)\, R_{inst}(t)\, \mathrm{d}t$$

RESTARTPATH $(lvl, \omega, \sigma, \text{EventList}, t)$

Input: Level lvl, Weight ω, state σ, event list, time t
Output: Final state of the simulation: new $(\sigma, \text{EventList}, t)$

```
(* main simulation loop *)
repeat
    (* get new events *)
    UPDATEEVENTLIST(σ, EventList, t)
    (* select executed event *)
    (a, binding, t') := SELECTEVENT(EventList)
    Event := (a, binding)
    (* update performance measure rvar = (rrate, rimp, ·, ·) *)
    Reward_rvar += ω * ((t' − t) * rrate(σ) + rimp(Event))
    (* execute state change *)
    t := t'; σ := Exec(Event, σ)
    (* RESTART level control *)
    lvl' := Level(σ)
    if lvl' > lvl then (* split *)
        for i = 1 ... R_lvl' do (σ', EventList', t') :=
                RESTARTPATH(lvl', ω/R_lvl', σ, EventList, t)
        (* Continue the final path *)
        σ := σ'; EventList := EventList'; t := t'
until (lvl' < lvl) or (stop condition reached, e.g. t ≥ MaxSimTime)
return (σ, EventList, t)
```

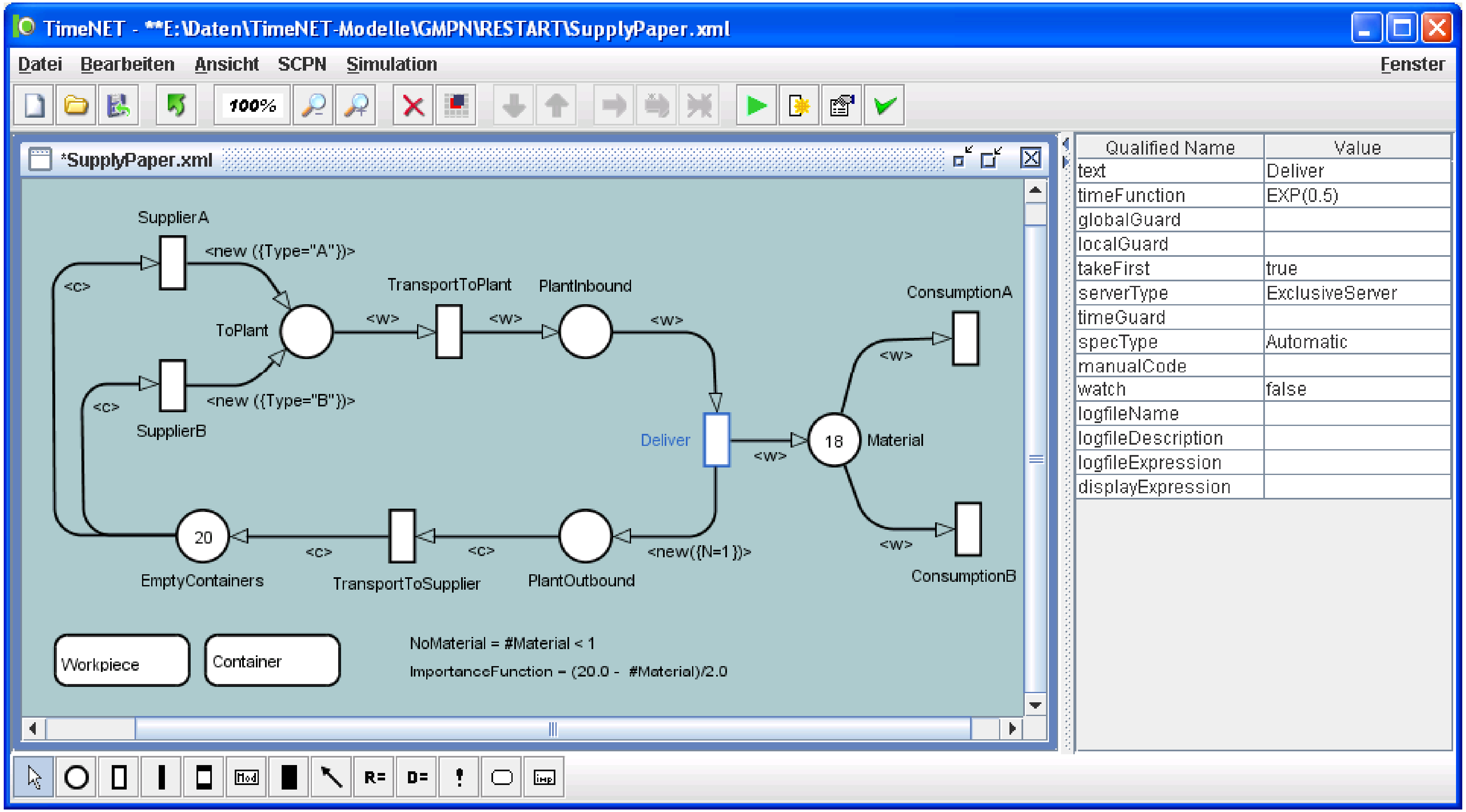
Outline

- Introduction
- Colored stochastic Petri nets
- RESTART simulation of CSPNs
- TimeNET prototype implementation
- An example

TimeNET

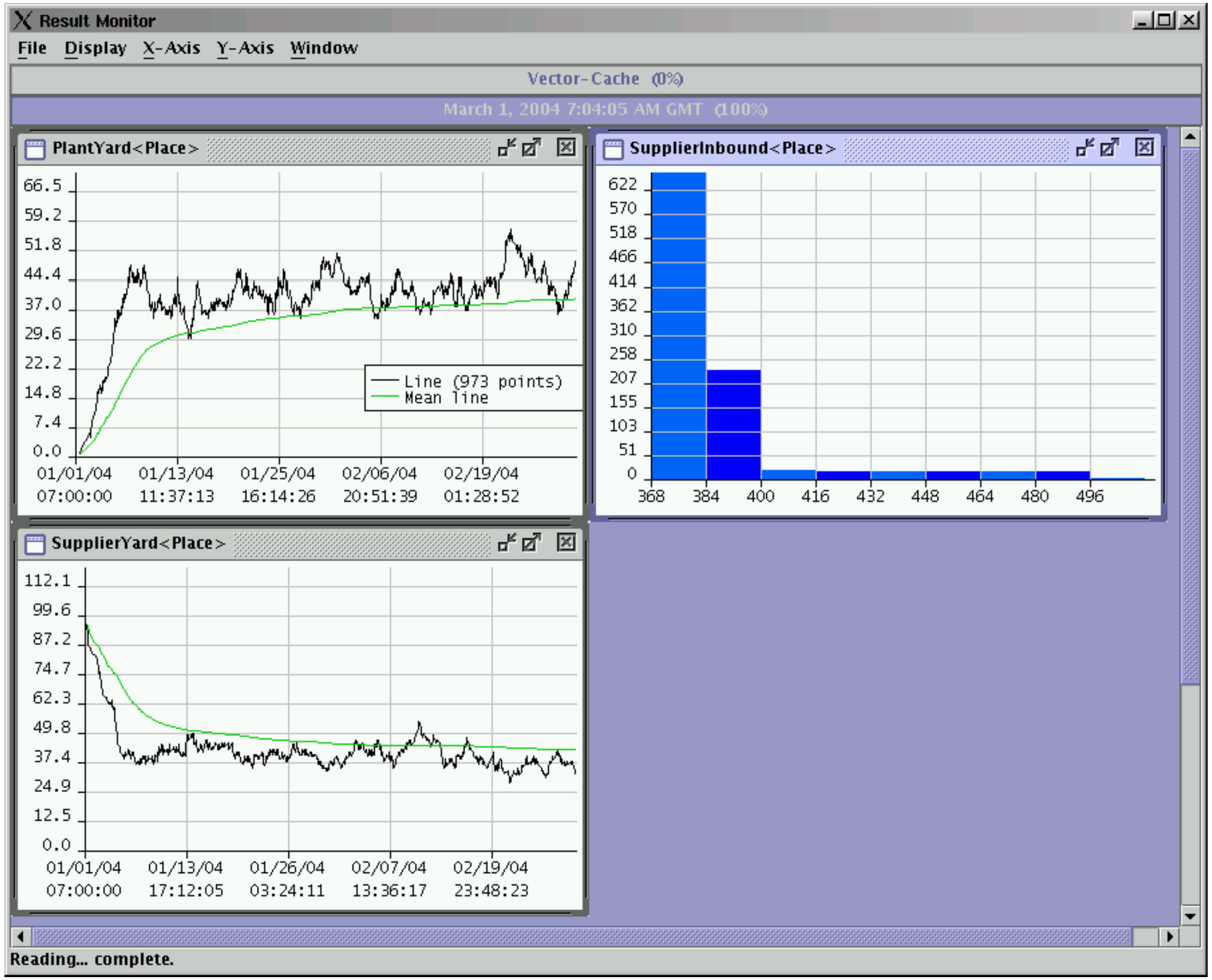
- Software tool
  - Simple and colored stochastic Petri nets
    - Run-time simulation code generation
  - Non-exponentially distributed firing delays
  - Numerical analysis and simulation
  - Steady-state and transient evaluation, Token game, distributed simulation
  - Runs on Windows XP, Debian Linux 3
  - http://www.tu-ilmenau.de/TimeNET

TimeNET



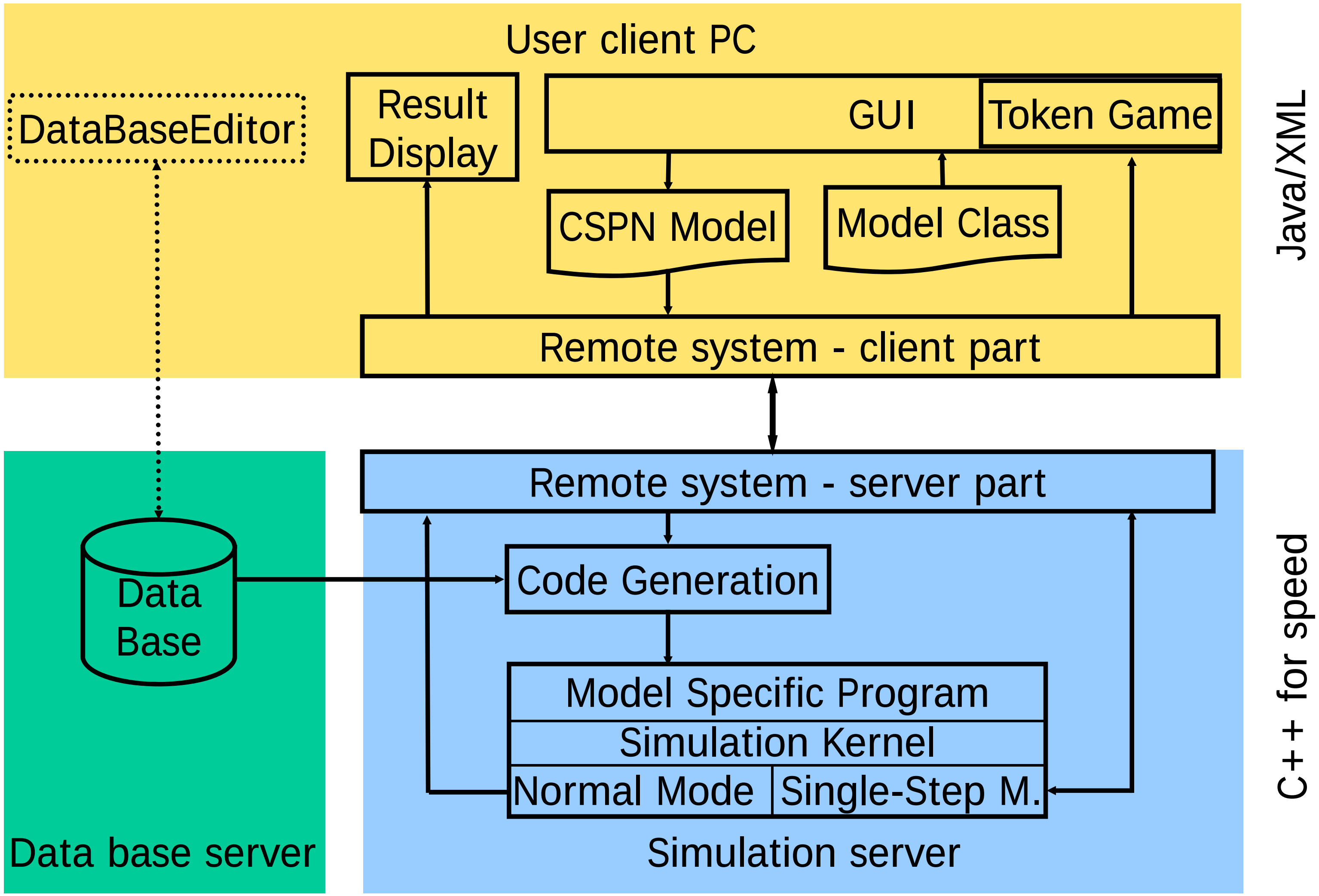
TimeNET - **E:\Daten\TimeNET-Modelle\GMPN\RESTART\SupplyPaper.xml
Datei Bearbeiten Ansicht SCPN Simulation
Fenster
100%
*SupplyPaper.xml
SupplierA
<new ({Type="A"})>
<c>
TransportToPlant
PlantInbound
ConsumptionA
ToPlant
<w>
<w>
<w>
<w>
<c>
<new ({Type="B"})>
SupplierB
Deliver
18
Material
<w>
20
<c>
<c>
<new({N=1})>
<w>
EmptyContainers
TransportToSupplier
PlantOutbound
ConsumptionB
Workpiece
Container
NoMaterial = #Material < 1
ImportanceFunction = (20.0 - #Material)/2.0
Qualified Name | Value
text | Deliver
timeFunction | EXP(0.5)
globalGuard
localGuard
takeFirst | true
serverType | ExclusiveServer
timeGuard
specType | Automatic
manualCode
watch | false
logfileName
logfileDescription
logfileExpression
displayExpression
R=
D=

TimeNET



Result Monitor
File Display X-Axis Y-Axis Window
Vector-Cache (0%)
March 1, 2004 7:04:05 AM GMT (100%)
PlantYard<Place>
66.5
59.2
51.8
44.4
37.0
29.6
22.2
14.8
7.4
0.0
Line (973 points)
Mean line
01/01/04 07:00:00
01/13/04 11:37:13
01/25/04 16:14:26
02/06/04 20:51:39
02/19/04 01:28:52
SupplierInbound<Place>
622
570
518
466
414
362
310
258
207
155
103
51
0
368 384 400 416 432 448 464 480 496
SupplierYard<Place>
112.1
99.6
87.2
74.7
62.3
49.8
37.4
24.9
12.5
0.0
01/01/04 07:00:00
01/13/04 17:12:05
01/26/04 03:24:11
02/07/04 13:36:17
02/19/04 23:48:23
Reading... complete.

TimeNET software architecture



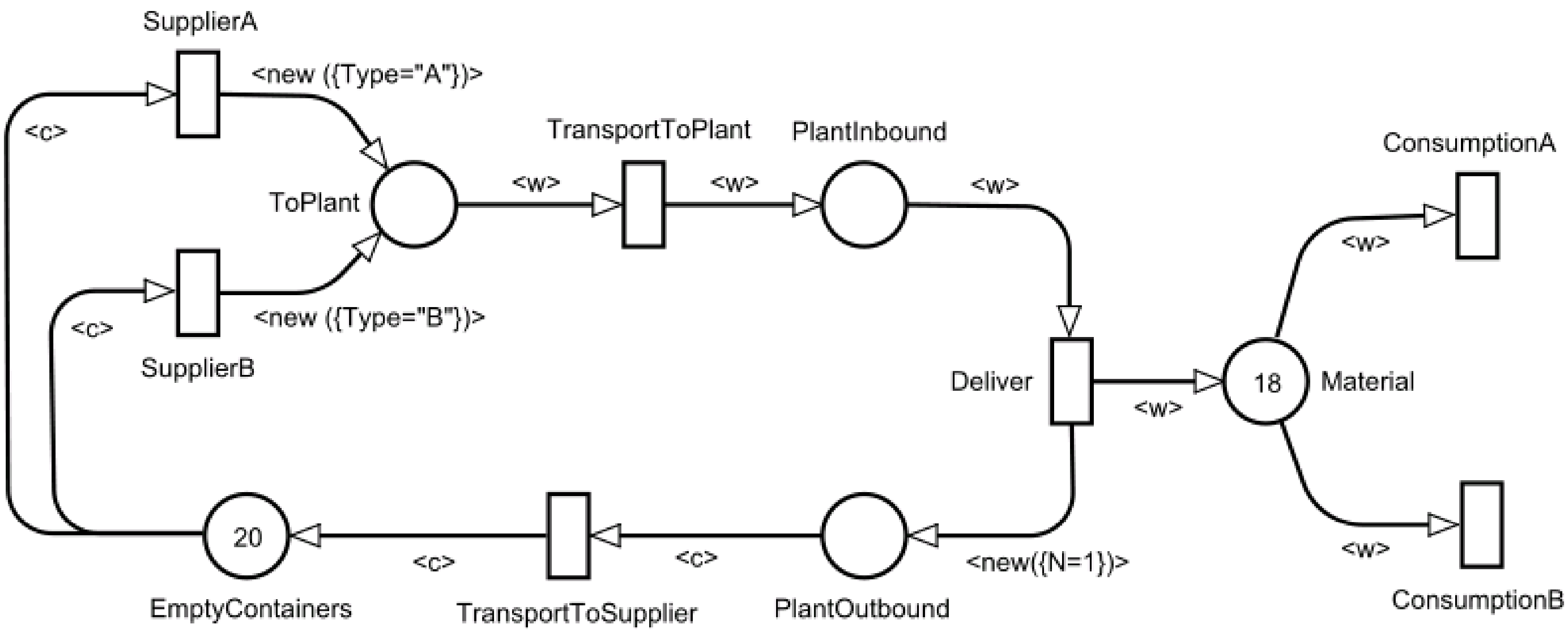
Outline

- Introduction
- Colored stochastic Petri nets
- RESTART simulation of CSPNs
- TimeNET prototype implementation
- An example

Validation Example

- M/M/1 queue
  - $\lambda/\mu = 0.5$
  - P{#customers > 20} ≈ $9.536 \cdot 10^{-7}$
  - 10,000 events RESTART
    → relative error 6%
  - 50,000 events standard
    → rare state not hit

Supply Chain Model Example



Workpiece

Container

NoMaterial = #Material < 1

ImportanceFunction = (20.0 - #Material)/2.0

Supply Chain Model Example

- Results
  - Measure of interest: probability of running out of material **`#Material < 1`**
    - capacity 20
  - **`ImportanceFunction = (20-#Material)/2`**
    - Levels 0..10
    - Splitting factor N=4
  - RESTART with 50,000 events: $2.6659 \; 10^{-6}$
    - 99% confidence interval width: $5.322 \; 10^{-8}$
    - 19 seconds CPU time on a 1.86GHz Pentium mobile

Conclusion

- Model-based design of complex systems
  - Complex systems
    → colored Petri nets
  - Non-functional properties
    → general performance measures
  - Highly reliable systems
    → RESTART simulation
  - Software tool
    → TimeNET prototype

- However: further obstacles!

Thank you for the attention!

More information

http://www.tu-ilmenau.de/sse

http://www.tu-ilmenau.de/TimeNET