

# NoPain – Meeting

Berlin

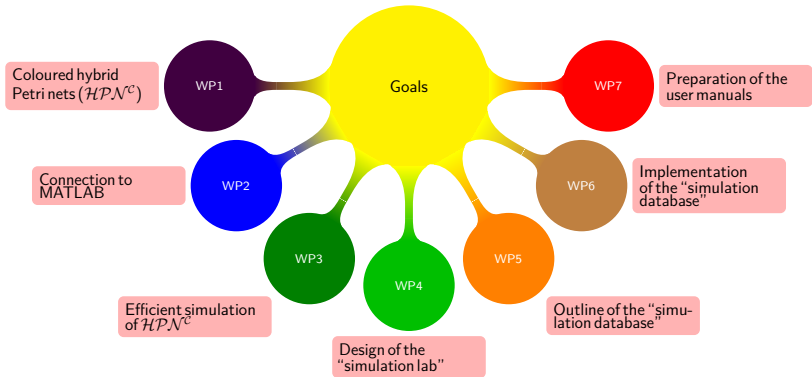
Monika Heiner    Christian Rohr

Department of Computer Science  
Brandenburg University of Technology Cottbus

<http://www-dssz.informatik.tu-cottbus.de>

September 24, 2014

# Work Packages





## Development of the “simulation lab”

- a. Predecessor WPs: BTU-WP3
- b. Successor WPs: BTU-WP5
  - Outline and implementation of the simulation laboratory
    - Support of professional simulation experiments
    - Execution of experiment scenarios for searching parameter spaces
    - Documentation and evaluation of the experiments

⇒ in cooperation with the BioModelKit

# User scenario



Snoopy

modelling & simulation only in Snoopy

## User scenario



BioModelKit



Snoopy

retrieve models from BioModelKit & simulate in Snoopy

## User scenario



BioModelKit



Snoopy



Simulation Lab

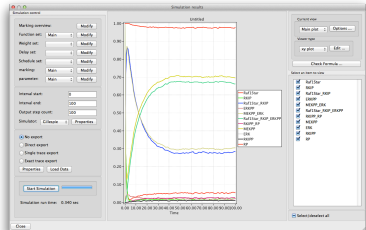
retrieve models from BioModelKit, create experiments in Snoopy,  
run in simulation lab

# Graphical User Interface



## Simulation Dialog

- all-in-one and overfilled dialog
- view area (plot/table) to small
- different properties scatter around
- on small screen resolutions not usable or broken



# Graphical User Interface



## Simulation Dialog

**Simulation Properties**

Start:  
End:  
Steps:  
Simulator:  
Runs:  
Threads:

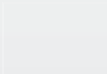
Net Properties

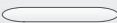
Views



Export

Analysis

History





**Simulation Results**

Default Plot

Plot 1

Plot 2

Plot 3

☐ Plot

☐ Table

☐ Histogramm



# Graphical User Interface



## Simulation Properties

- separate controls and view area
- only selected control is open, others collapsed  
↪ cleaner interface
- simulation properties is the default
- set simulation start and end time
- define output steps:
  - $N - N$  steps of equal length
  - $N_1 : T_1, N_2 : T_2, \dots - N_i$   
steps of equal length to  $T_j$
- select the simulator
- set simulator specific settings

A screenshot of a software window titled "Simulation Properties". The window has a light gray background and a vertical stack of controls. At the top, there's an orange header bar with the title. Below it, several labels are listed: "Start:", "End:", "Steps:", "Simulator:", "Runs:", and "Threads:". These are followed by a series of buttons: "Net Properties", "Views", "Export", "Analysis", and "History". Below these buttons is a large, empty rectangular area. At the bottom of the window, there are two buttons: "Start" with a play icon and "Save" with a floppy disk icon.

# Graphical User Interface



## Net Properties

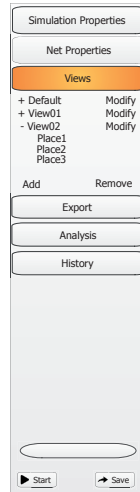
- list of net properties to modify
- constant groups
  - marking, parameters, ...
- rate functions
- deterministic times
- scheduled times

# Graphical User Interface



## Simulation Views

- list of views
- add new views
- remove views
- modify view settings
- open view by double click
- de-/select element of active view

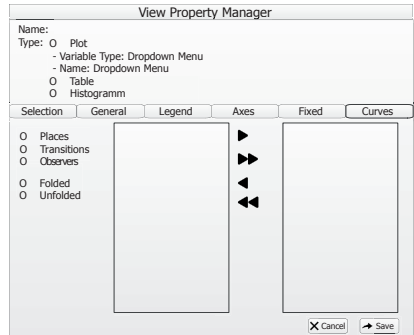


# Graphical User Interface



## Simulation View – Property Manager

- select elements to show out of available
  - places, transitions, observers
  - folded, unfolded
- select out of list
- use regular expression to select subset of elements
- how to display results:
  - plot, table, histogram
- define display settings:
  - title, axis labels, legend, line style, ...

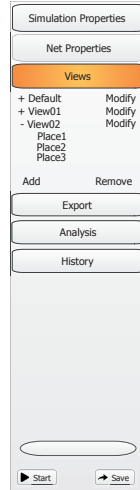


# Graphical User Interface



## Simulation Export

- export to comma separated values
  - manual export by user
  - automatic export after simulation
  - automatic export of every single run
  - automatic export of every single run and every transition firing
- export plot/histogram to image
- export either selected elements or all of them

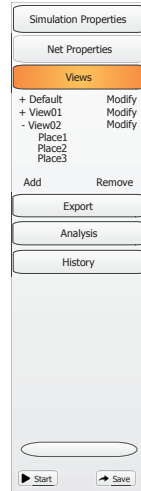


# Graphical User Interface



## Simulation Analysis

- expectations, e.g. averaged number of tokens
- probability distributions
  - transient, cumulative, steady state
- token distributions
  - actual values, normalized values

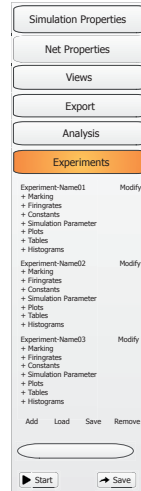


# Graphical User Interface



## Simulation Experiments

- list of created/loaded experiments
- show settings of each experiment
- add new experiments
- load experiments from external file
- save selected experiments to external file
- experiments always stored with snoopy file
- remove experiments
- modify experiment settings





## Experiments

- define experiments to run:
  - 1 select net parameters to iterate over:
    - constant, initial marking, multiplicity
  - 2 define the range of values and steps for each parameter
  - 3 select analysis/simulation method and properties:
    - transient, steady-state, #runs, time, ...
  - 4 what results should be stored:
    - expectations, probabilities, ...
  - 5 save each experiment in separate file together with model file
- run experiments using:
  - 1 Snoopy
  - 2 simulation lab
  - 3 Marcie
- view results in Snoopy





## Images

- import images into Snoopy
- only raster graphics images supported (png, jpg, bmp, gif)
  - ↪ are resolution dependent
- new metadataclass *images*
- moveable, scalable
- placed behind any other element
- everybody needs to have the image file!  
possible solutions:
  - 1 separate files and copy images separately
  - 2 include in snoopy's XML file
    - ↪ but handling binary data in XML is difficult
  - 3 package XML file and image files together
    - ↪ new file format



## Current

- single file containing all information
- text file using XML syntax
  - well structured, but redundancies
- utf-8 encoding
- uncompressed
- each netclass with different filename extension

# File Format



```
<?xml version="1.0" encoding="utf-8"?>
<Snoopy version="2" revision="1.13">
  <netclass name="Petri_Net"/>
  <nodeclasses count="4">
    <nodeclass count="0" name="Place"/>
    <nodeclass count="0" name="Transition"/>
    <nodeclass count="0" name="Coarse_Place"/>
    <nodeclass count="0" name="Coarse_Transition"/>
  </nodeclasses>
  <edgeclasses count="1">
    <edgeclass count="0" name="Edge"/>
  </edgeclasses>
  <metadataclasses count="3">
    <metadataclass count="1" name="General">
      ...
    </metadataclass>
    <metadataclass count="0" name="Comment"/>
    <metadataclass count="0" name="Constant_Class"/>
    <metadataclass count="0" name="Function_Class"/>
  </metadataclasses>
</Snoopy>
```

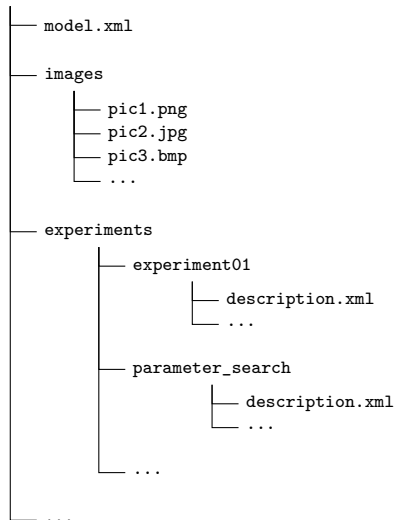
# File Format



## New

- a collection of several sub-documents within a package
- package is a standard ZIP file with a defined structure of sub-documents
- *model.xml* containing the net remains the same
- folder *images* contains pictures used in the model
- folder *experiments* contains subfolders for each experiment and a XML file with the description of it

example.pn



## Next steps...



- refine and finish outline
- implement redesign, new features, new file format

# Milestones



	2013				2014				2015			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
WP1		M1										
WP2				M2								
WP3						M3						
WP4								M4				
WP5								M4				
WP6												M5
WP7												M6



Thank you for your attention!