Kickoff Meeting NoPain

Work Schedule

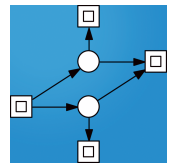
Monika Heiner Christian Rohr

Department of Computer Science
Brandenburg University of Technology Cottbus

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

February 5, 2013

Review

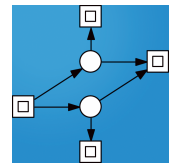


What has been achieved so far...

- before MoPS:
  - Snoopy a generic unifying Petri net framework
  - qualitative, stochastic, continuous Petri nets
  - hierarchical organisation by subnets, logical node
- during MoPS
  - coloured Petri nets
  - modular modelling concept by AG Marwan
- parallel to MoPS
  - hybrid Petri nets, but no coloured hybrid Petri nets

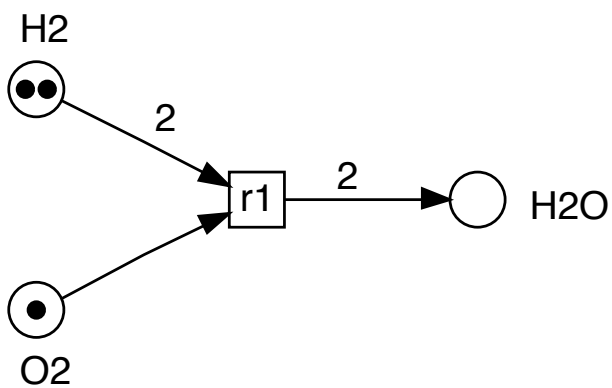
`http://www-dssz.informatik.tu-cottbus.de/snoopy.html`

Review



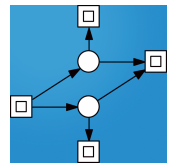
Petri net				**Bio network**
transitions	$\leftrightarrow$	atomic actions	$\leftrightarrow$	chemical reactions
places	$\leftrightarrow$	local conditions	$\leftrightarrow$	chemical compounds
arc weights	$\leftrightarrow$	multiplicities	$\leftrightarrow$	stoichiometric relations
token(s)	$\leftrightarrow$	condition's state	$\leftrightarrow$	available amount (e.g. mol)
marking	$\leftrightarrow$	system state	$\leftrightarrow$	compound distribution

Example



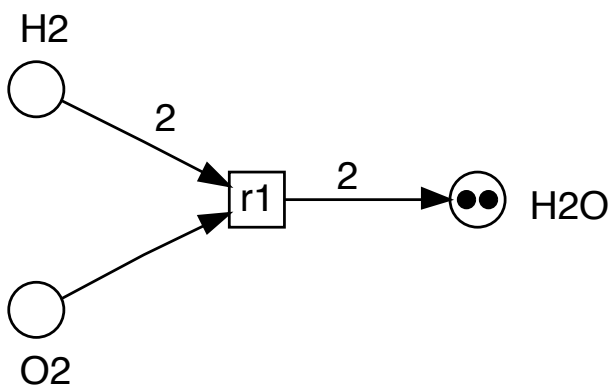
$$r_1 : 2H_2 + O_2 \rightarrow 2H_2O$$

Review

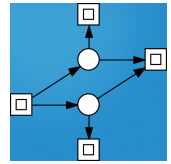


Petri net				**Bio network**
transitions	$\leftrightarrow$	atomic actions	$\leftrightarrow$	chemical reactions
places	$\leftrightarrow$	local conditions	$\leftrightarrow$	chemical compounds
arc weights	$\leftrightarrow$	multiplicities	$\leftrightarrow$	stoichiometric relations
token(s)	$\leftrightarrow$	condition's state	$\leftrightarrow$	available amount (e.g. mol)
marking	$\leftrightarrow$	system state	$\leftrightarrow$	compound distribution

Example

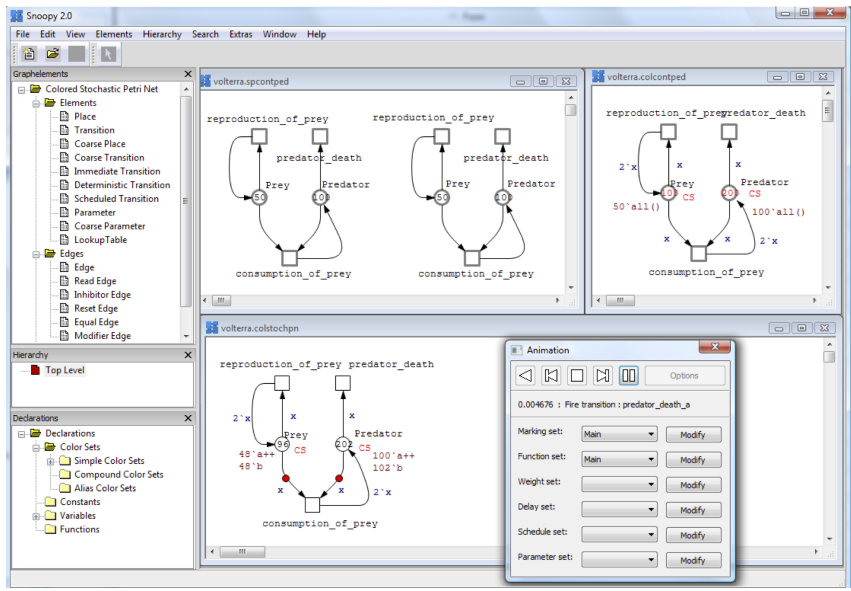


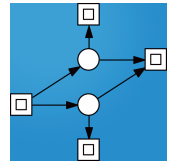
$$r_1 : 2H_2 + O_2 \rightarrow 2H_2O$$



Snoopy a generic unifying Petri net framework

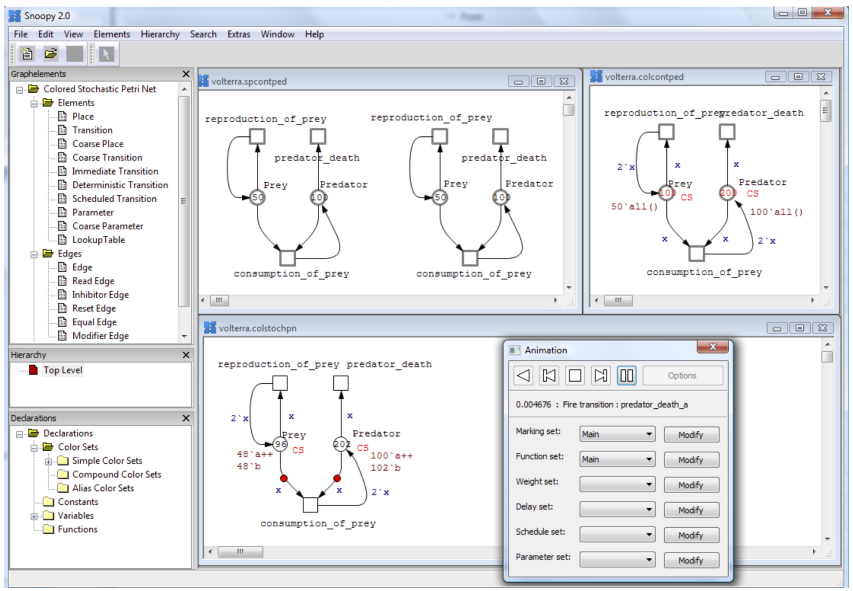
- graph constraints ensure syntactically correct models
- simultaneous use of several Petri net classes
- graphical user interface adapts dynamically to the active one
- consists of two levels: uncoloured and coloured
- built-in animation and simulation (depending on the net class)

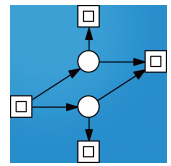




Snoopy a generic unifying Petri net framework

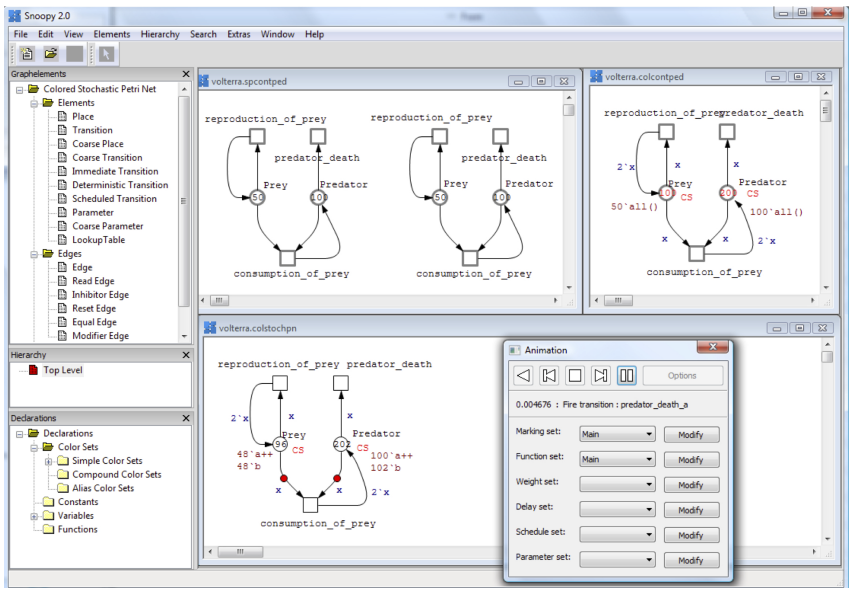
- graph constraints ensure syntactically correct models
- simultaneous use of several Petri net classes
- graphical user interface adapts dynamically to the active one
- consists of two levels: uncoloured and coloured
- built-in animation and simulation (depending on the net class)

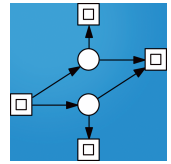




Snoopy a generic unifying Petri net framework

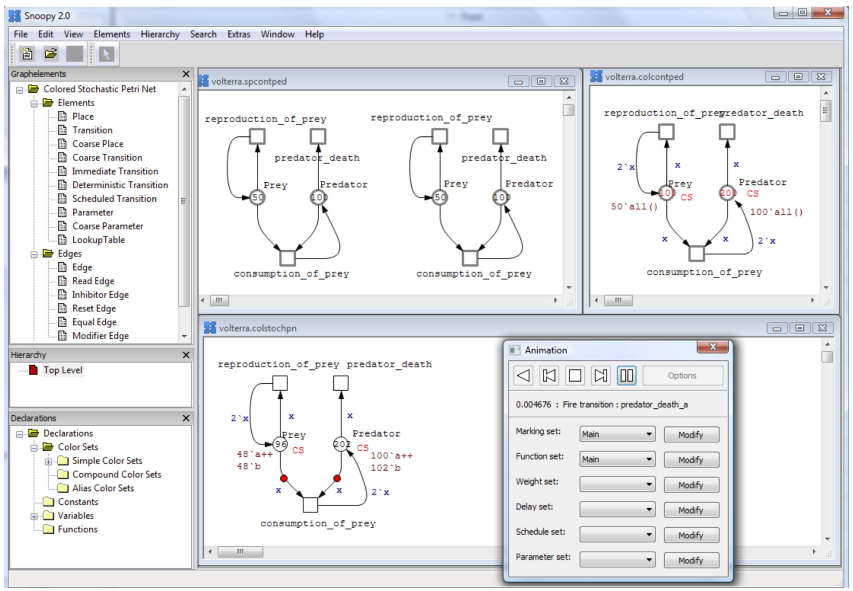
- graph constraints ensure syntactically correct models
- simultaneous use of several Petri net classes
- graphical user interface adapts dynamically to the active one
- consists of two levels: uncoloured and coloured
- built-in animation and simulation (depending on the net class)

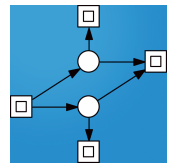




Snoopy a generic unifying Petri net framework

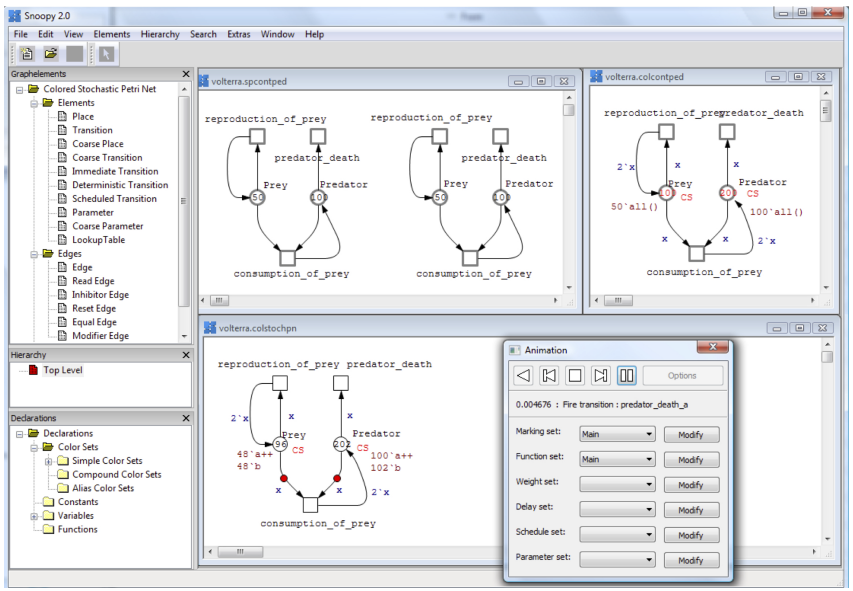
- graph constraints ensure syntactically correct models
- simultaneous use of several Petri net classes
- graphical user interface adapts dynamically to the active one
- consists of two levels: uncoloured and coloured
- built-in animation and simulation (depending on the net class)



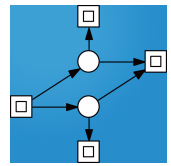


Snoopy a generic unifying Petri net framework

- graph constraints ensure syntactically correct models
- simultaneous use of several Petri net classes
- graphical user interface adapts dynamically to the active one
- consists of two levels: uncoloured and coloured
- built-in animation and simulation (depending on the net class)

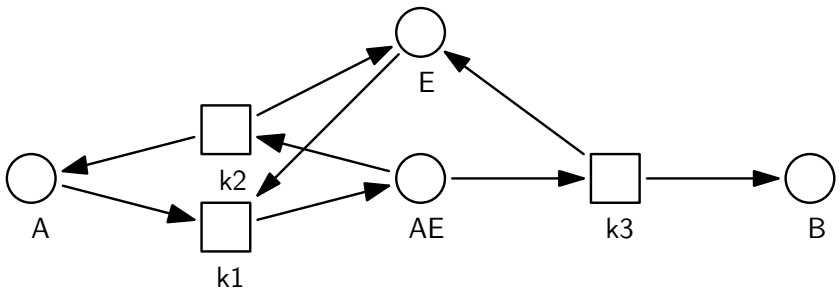


Review



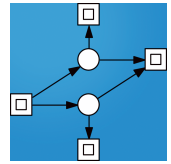
Logical Nodes

$$A + E \leftrightharpoons A|E \rightarrow B$$



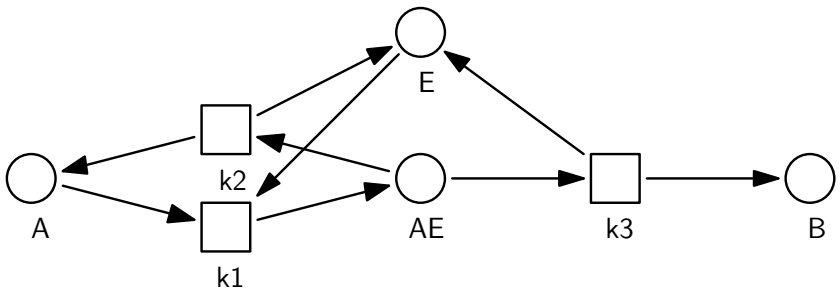
- serve as connectors
- logical places & logical transitions
- place-oriented nets
- transition-oriented nets
- connect subnets

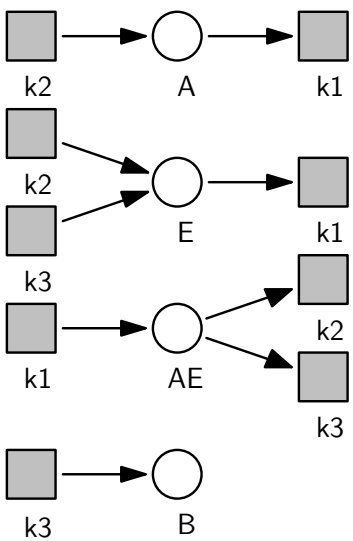
Review



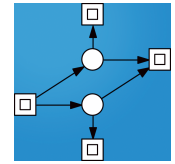
Logical Nodes

$$A + E \leftrightharpoons A|E \rightarrow B$$



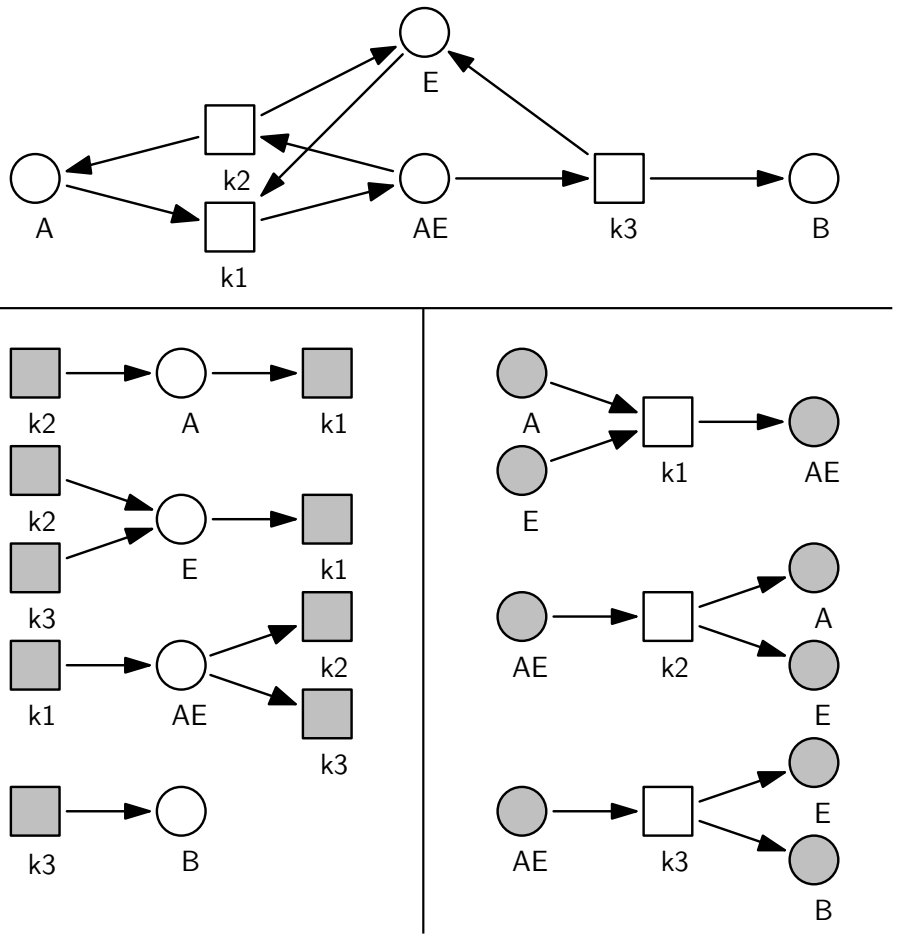


- serve as connectors
- logical places & logical transitions
- place-oriented nets
- transition-oriented nets
- connect subnets



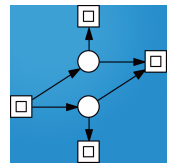
Logical Nodes

$$A + E \leftrightharpoons A|E \rightarrow B$$



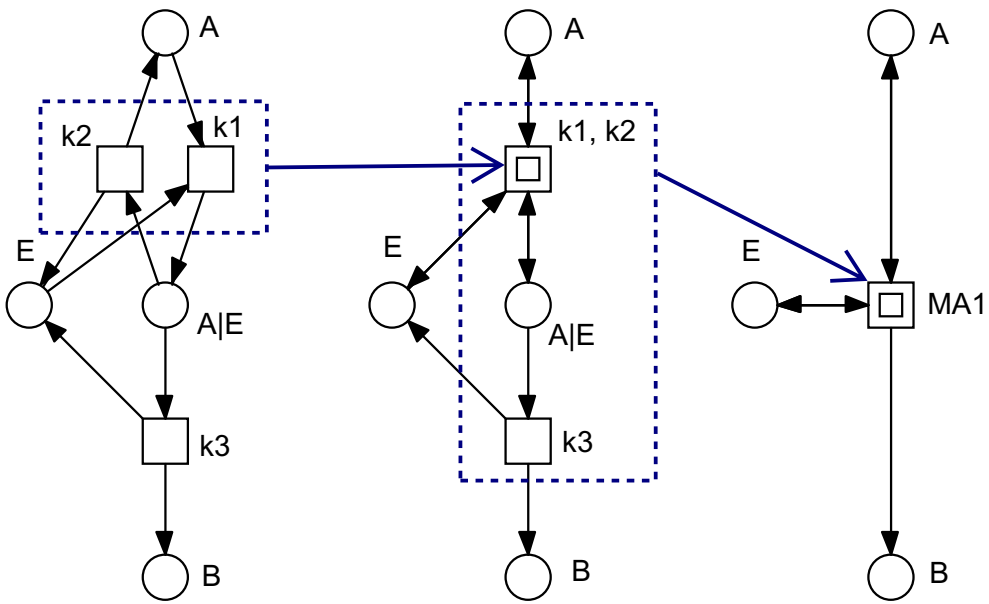
- serve as connectors
- logical places & logical transitions
- place-oriented nets
- transition-oriented nets
- connect subnets

Review



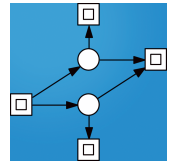
Hierarchy

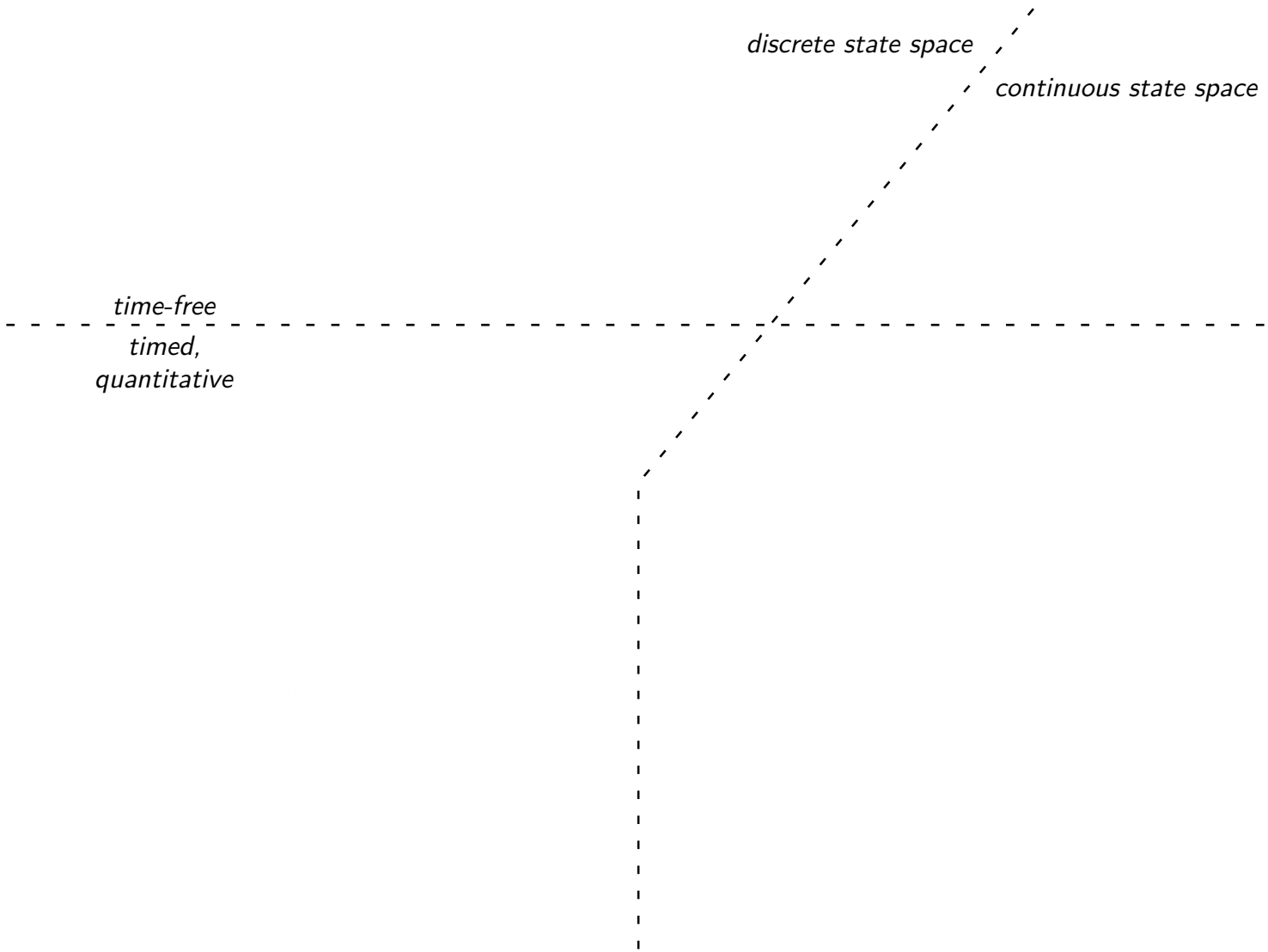
$$A + E \leftrightharpoons A|E \rightarrow B$$



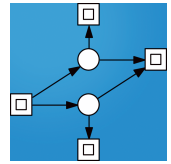
- hierarchy by subgraphs
- coarse transitions
- coarse places
- hide transition-bordered subnets
- hide place-bordered subnets
- design hierarchically structured nets

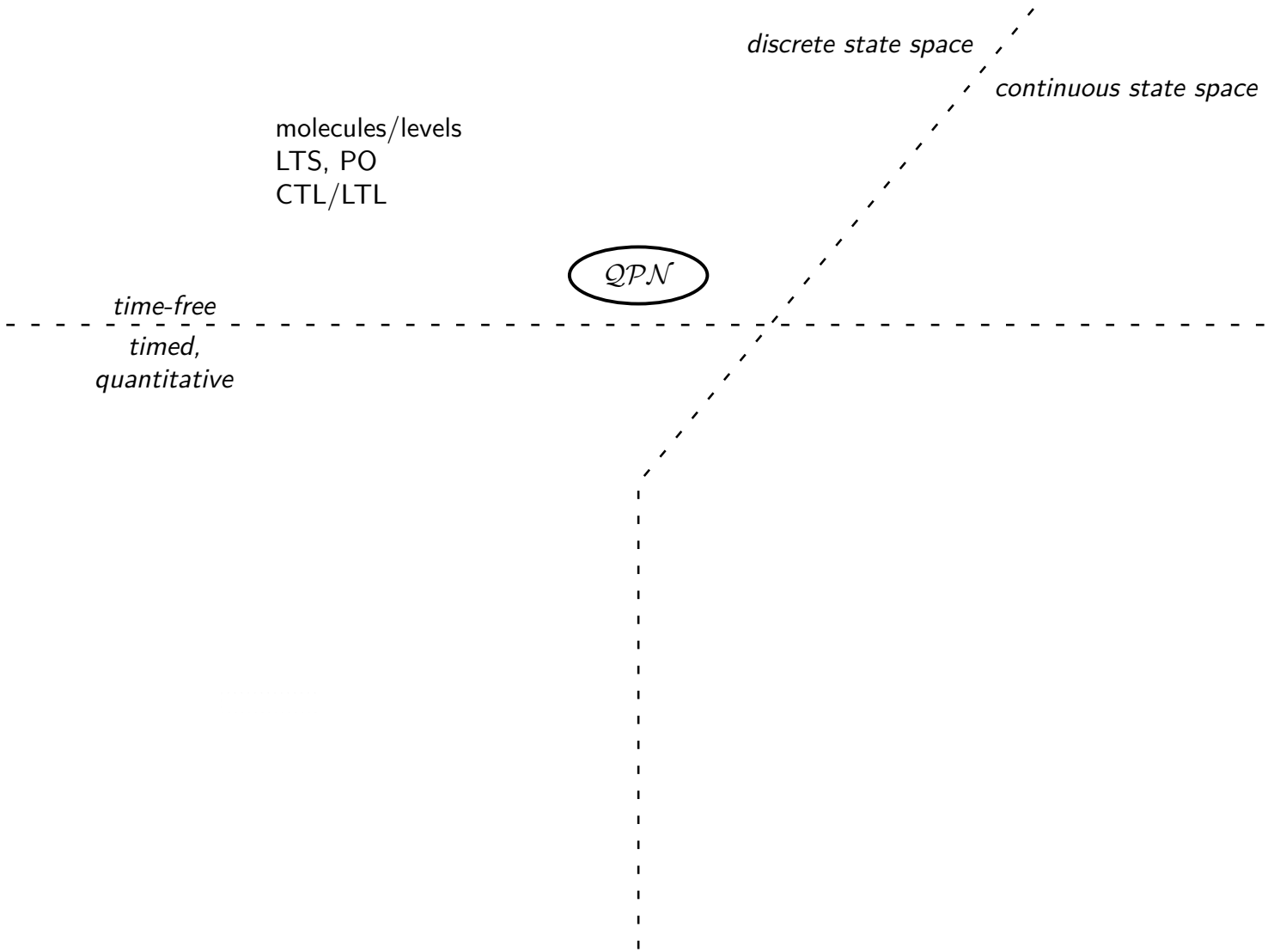
Review



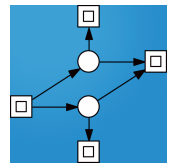

discrete state space
continuous state space
time-free
timed,
quantitative

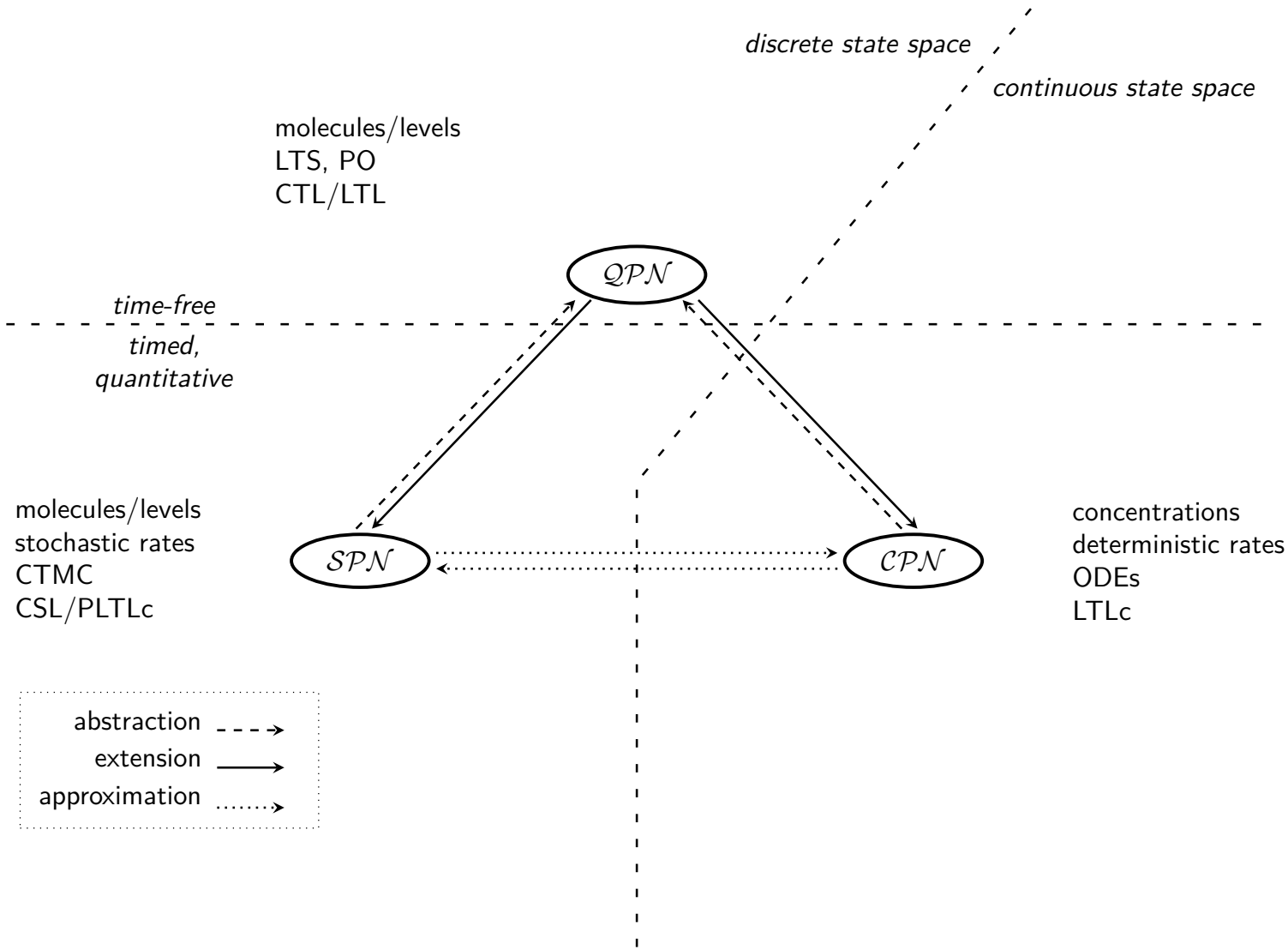
Review



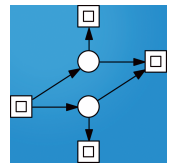

discrete state space
continuous state space
molecules/levels
LTS, PO
CTL/LTL
$\mathcal{QPN}$
time-free
timed,
quantitative

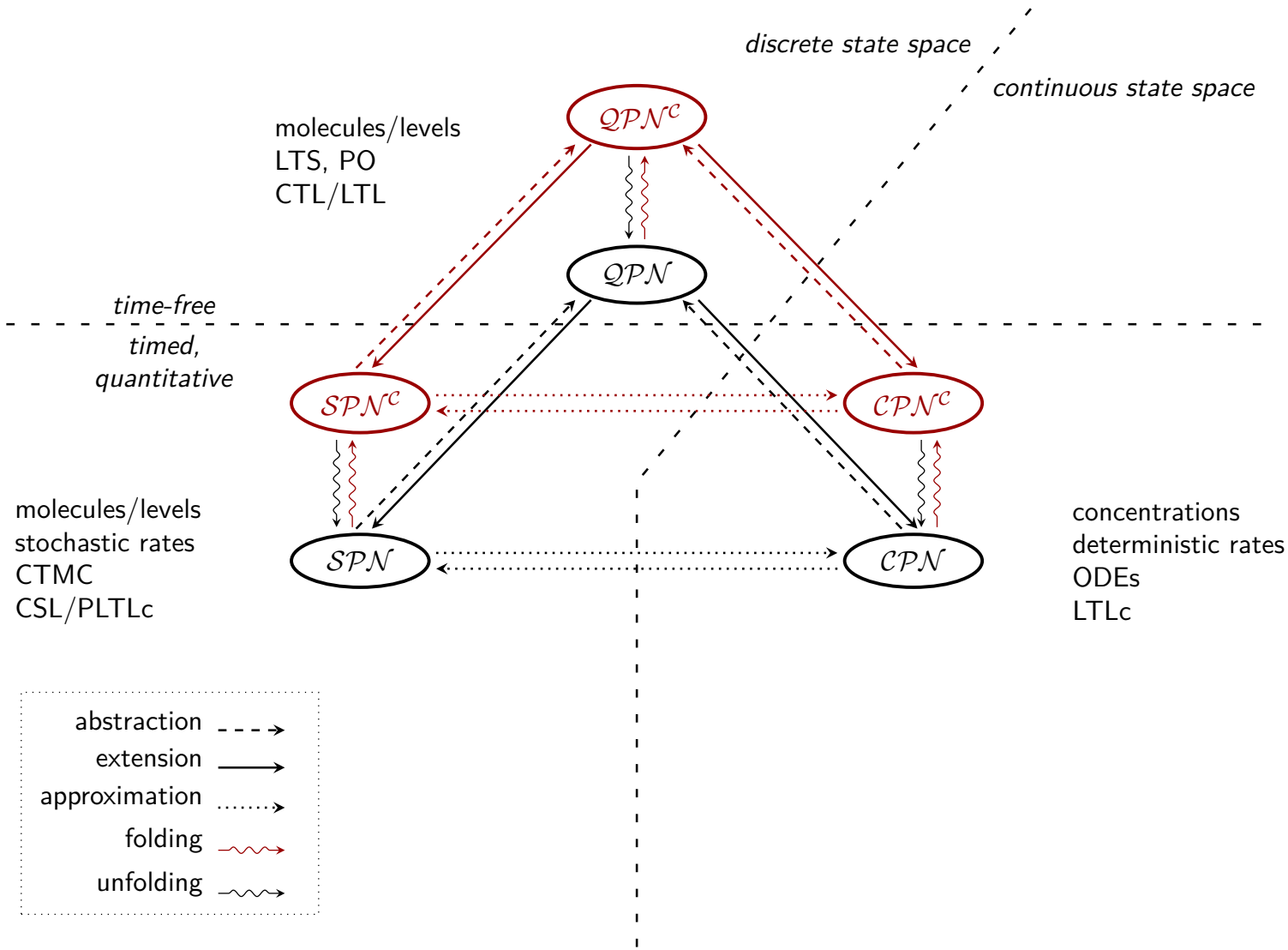
Review



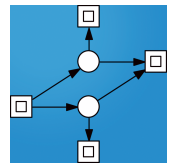

discrete state space
continuous state space
molecules/levels
LTS, PO
CTL/LTL
$\mathcal{QPN}$
time-free
timed,
quantitative
molecules/levels
stochastic rates
CTMC
CSL/PLTLc
$\mathcal{SPN}$
$\mathcal{CPN}$
concentrations
deterministic rates
ODEs
LTLc
abstraction
extension
approximation

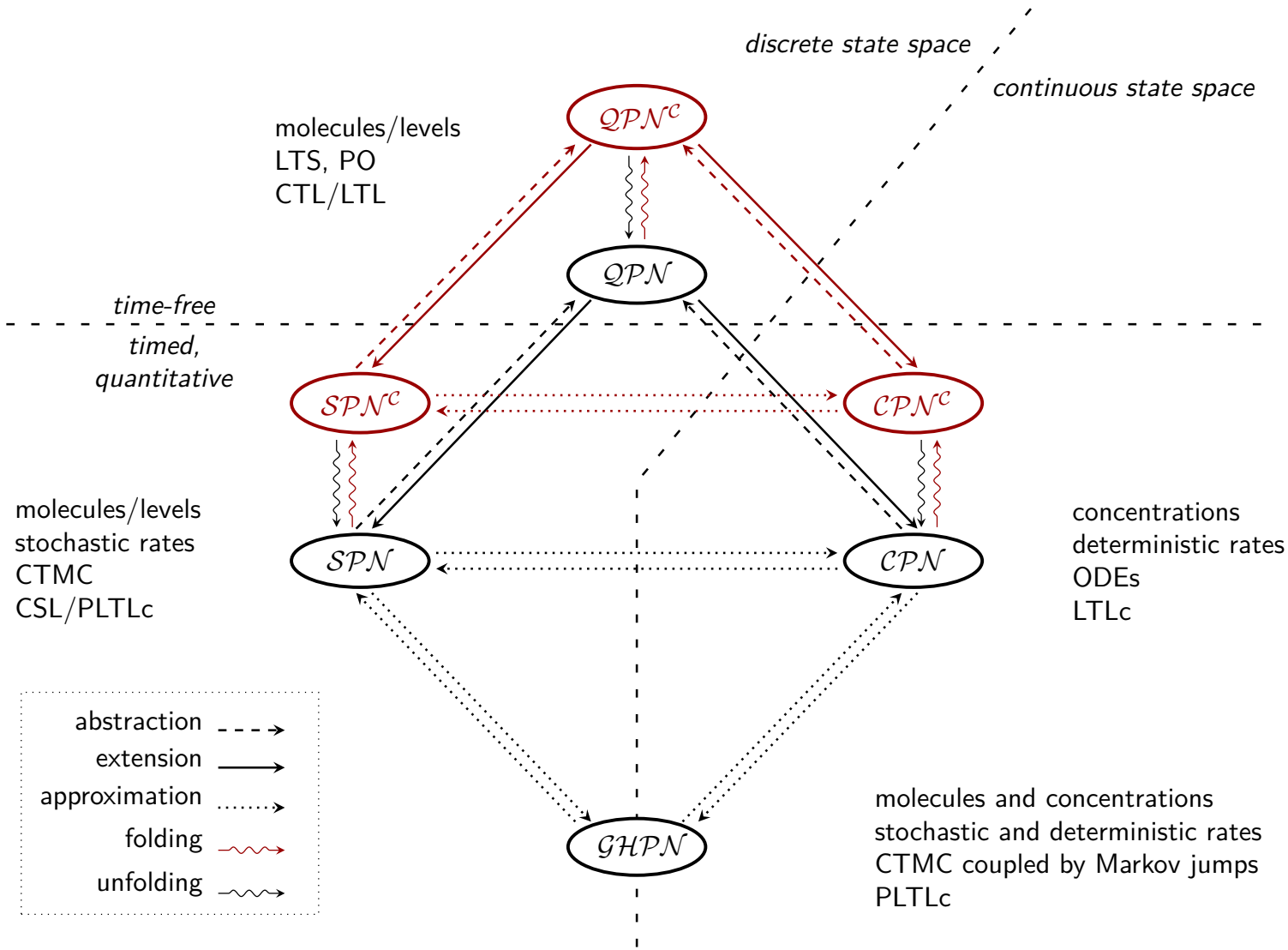
Review



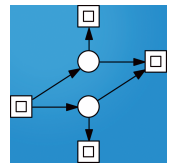

discrete state space
continuous state space
molecules/levels
LTS, PO
CTL/LTL
$\mathcal{QPN}^{\mathcal{C}}$
$\mathcal{QPN}$
time-free
timed,
quantitative
$\mathcal{SPN}^{\mathcal{C}}$
$\mathcal{CPN}^{\mathcal{C}}$
molecules/levels
stochastic rates
CTMC
CSL/PLTLc
$\mathcal{SPN}$
$\mathcal{CPN}$
concentrations
deterministic rates
ODEs
LTLc
abstraction
extension
approximation
folding
unfolding

Review




discrete state space
continuous state space
molecules/levels
LTS, PO
CTL/LTL
$\mathcal{QPN}^c$
$\mathcal{QPN}$
time-free
timed,
quantitative
$\mathcal{SPN}^c$
$\mathcal{CPN}^c$
molecules/levels
stochastic rates
CTMC
CSL/PLTLc
$\mathcal{SPN}$
$\mathcal{CPN}$
concentrations
deterministic rates
ODEs
LTLc
abstraction
extension
approximation
folding
unfolding
$\mathcal{GHPN}$
molecules and concentrations
stochastic and deterministic rates
CTMC coupled by Markov jumps
PLTLc

Goals

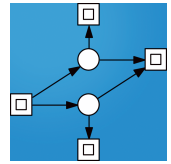


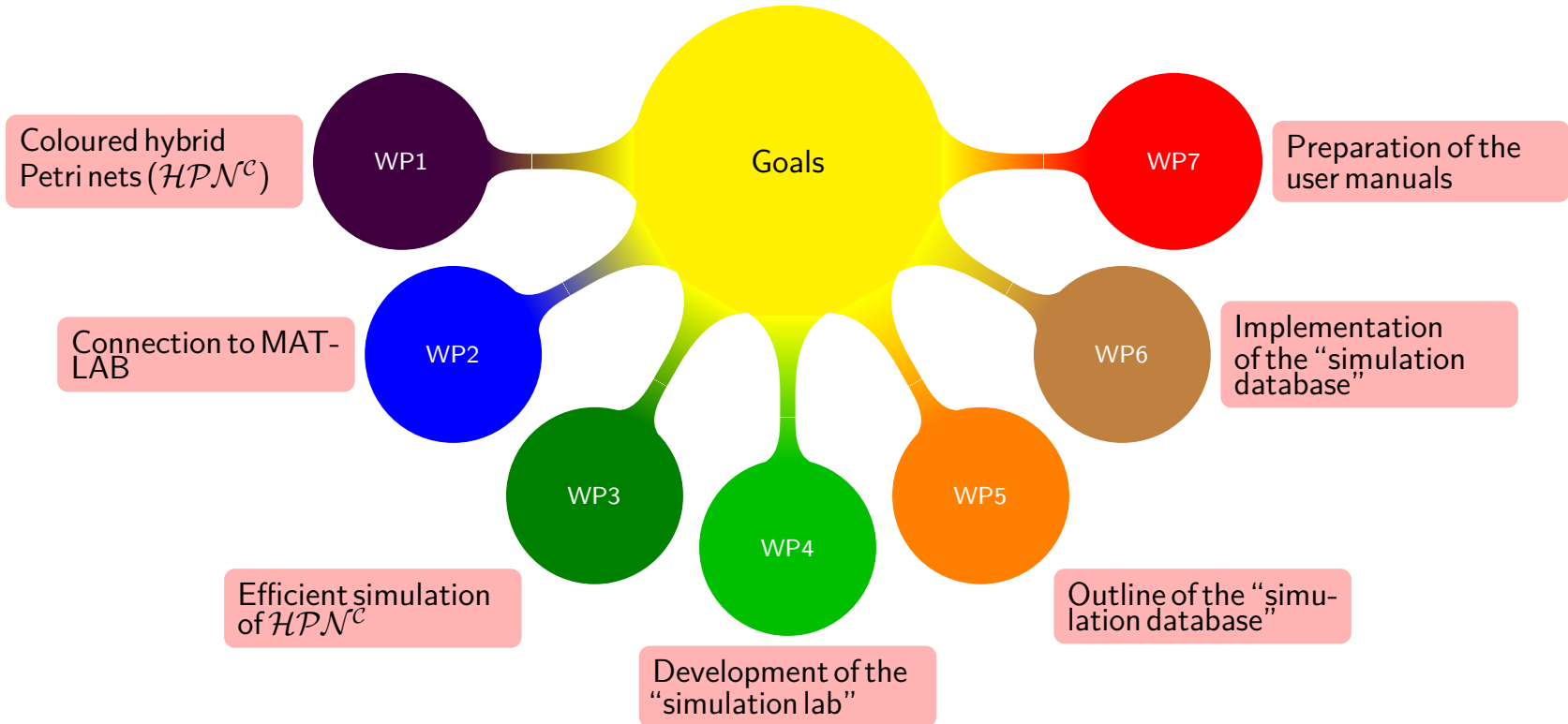
What shall be achieved...

- Harmonisation and extension of Snoopy
- Optimisation of simulation algorithms in terms of parallelisation
- Development of tools for performing automated simulation experiments and extended analysis of the simulation results

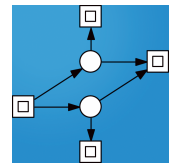
  $\Longrightarrow$ in cooperation with the BioModelKit

Work Packages





WP1



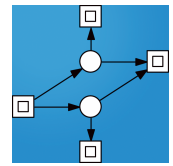
Coloured hybrid Petri nets ($\mathcal{HPN}^{\mathcal{C}}$)

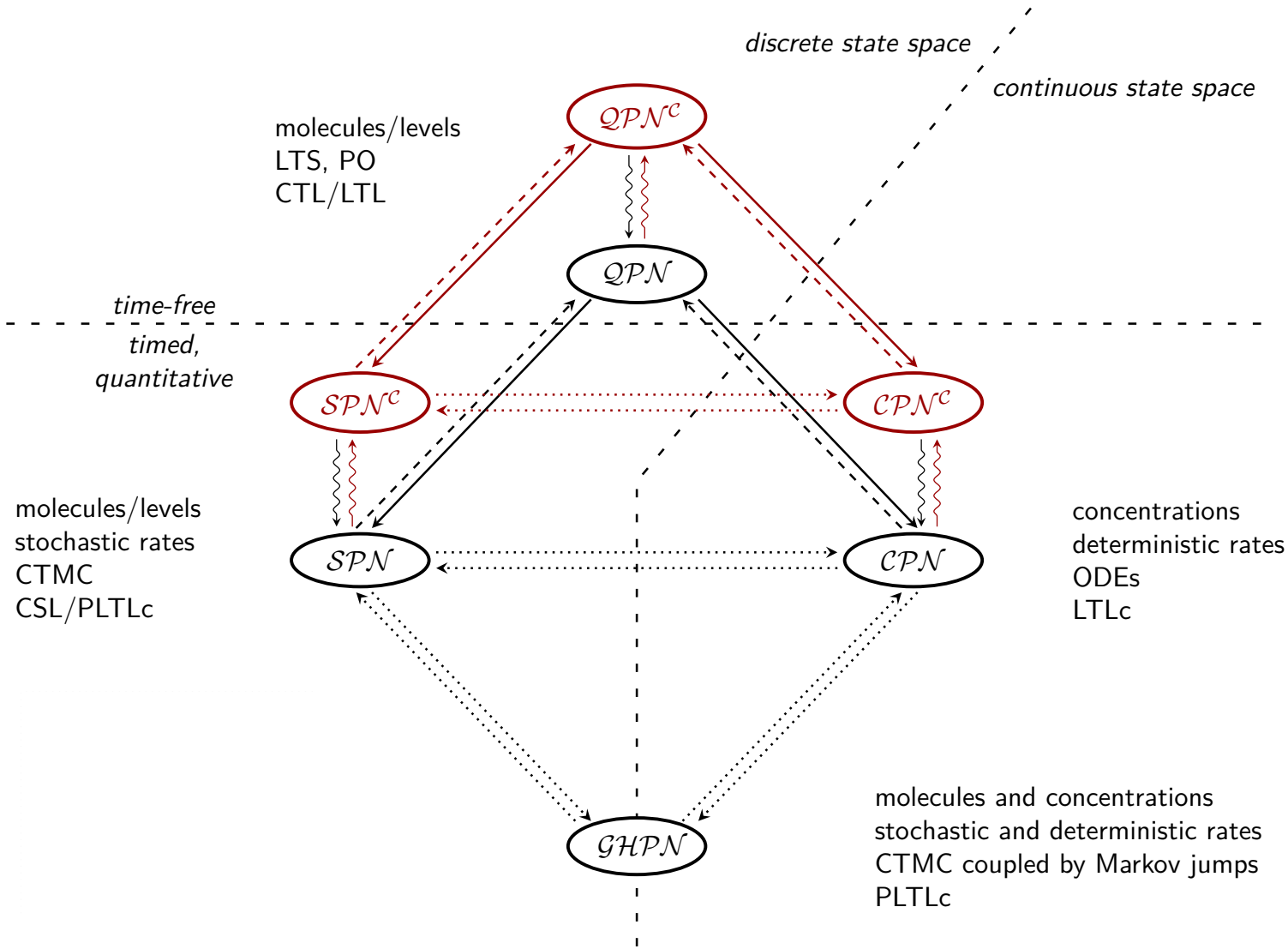
a. Predecessor WPs: -/-

b. Successor WPs: BTU-WP2, BTU-WP7

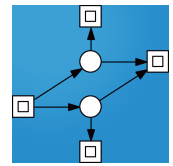
- Uniting the modelling concepts coloured and hybrid Petri nets in Snoopy
- Expansions around new special modelling features like constants and self-modifying edges with marking-dependent arc weights

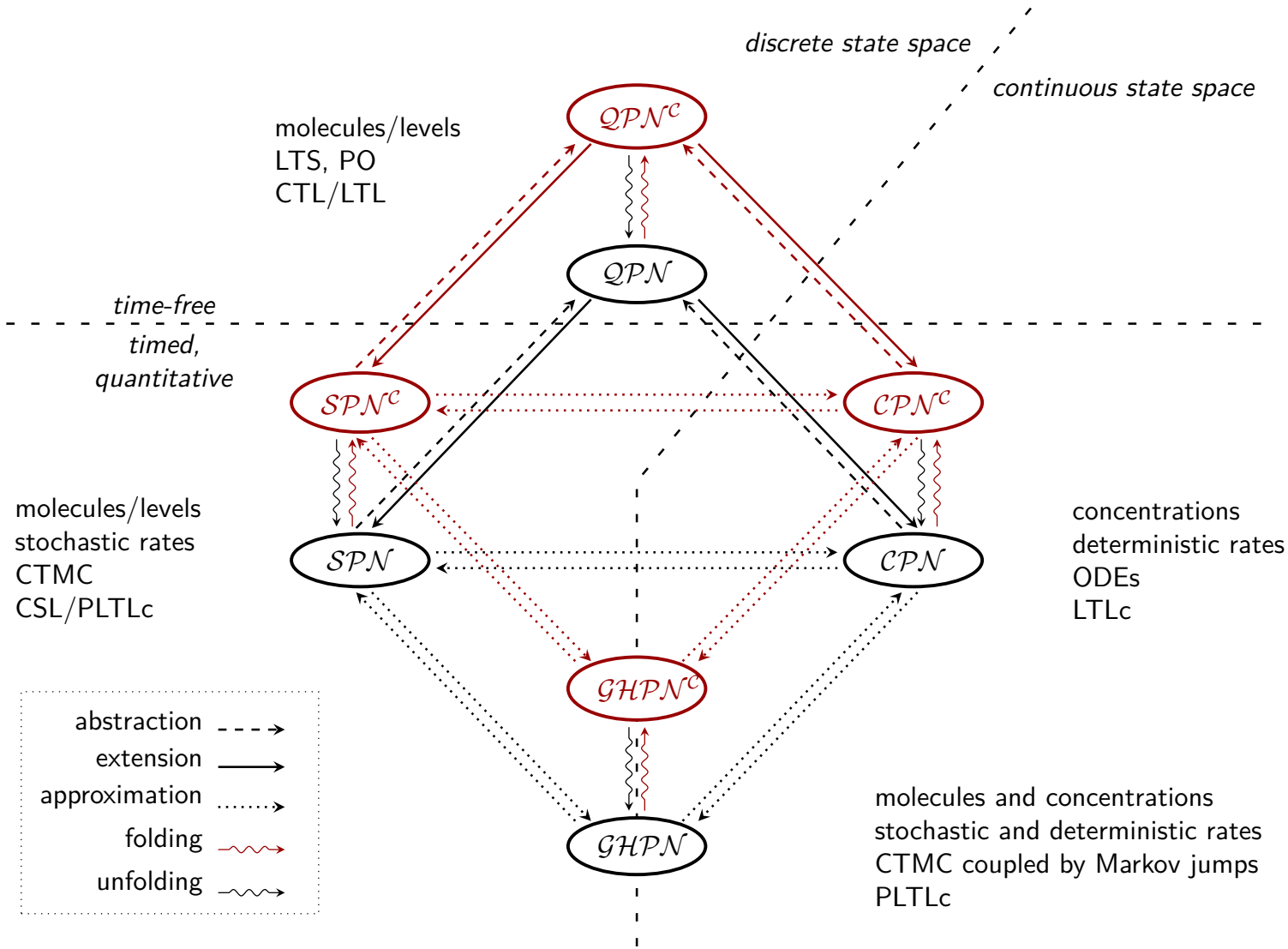
WP1



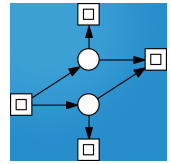

discrete state space
continuous state space
molecules/levels
LTS, PO
CTL/LTL
$\mathcal{QPN}^{\mathcal{C}}$
$\mathcal{QPN}$
time-free
timed,
quantitative
$\mathcal{SPN}^{\mathcal{C}}$
$\mathcal{CPN}^{\mathcal{C}}$
molecules/levels
stochastic rates
CTMC
CSL/PLTLc
$\mathcal{SPN}$
$\mathcal{CPN}$
concentrations
deterministic rates
ODEs
LTLc
$\mathcal{GHPN}$
molecules and concentrations
stochastic and deterministic rates
CTMC coupled by Markov jumps
PLTLc

WP1




discrete state space
continuous state space
molecules/levels
LTS, PO
CTL/LTL
$\mathcal{QPN}^{\mathcal{C}}$
$\mathcal{QPN}$
time-free
timed,
quantitative
$\mathcal{SPN}^{\mathcal{C}}$
$\mathcal{CPN}^{\mathcal{C}}$
molecules/levels
stochastic rates
CTMC
CSL/PLTLc
concentrations
deterministic rates
ODEs
LTLc
$\mathcal{SPN}$
$\mathcal{CPN}$
$\mathcal{GHPN}^{\mathcal{C}}$
$\mathcal{GHPN}$
abstraction
extension
approximation
folding
unfolding
molecules and concentrations
stochastic and deterministic rates
CTMC coupled by Markov jumps
PLTLc

WP2



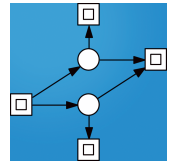
Connection to MATLAB

a. Predecessor WPs: BTU-WP1

b. Successor WPs: BTU-WP7

- Connection of the MATLAB software package to the hybrid Petri nets as well as to relevant net classes
  1. Export the net structure in MATLAB format
  2. Direct connection from Snoopy to MATLAB through its API, e.g. call MATLAB functions from within Snoopy

WP3

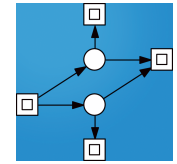


Efficient simulation of $\mathcal{HPN}^{\mathcal{C}}$

a. Predecessor WPs: BTU-WP1, OvGUM-WP1

b. Successor WPs: BTU-WP4

- Examination of the Petri net models with regard to parallelisation potential
- Investigation of optimisation possibilities and performance comparisons with alternative tools, i.e. MATLAB, Cain...



Development of the "simulation lab"

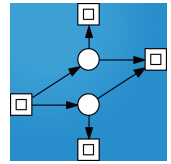
a. Predecessor WPs: BTU-WP3

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

$\Longrightarrow$ in cooperation with the BioModelKit

WP4

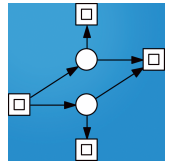


User scenario

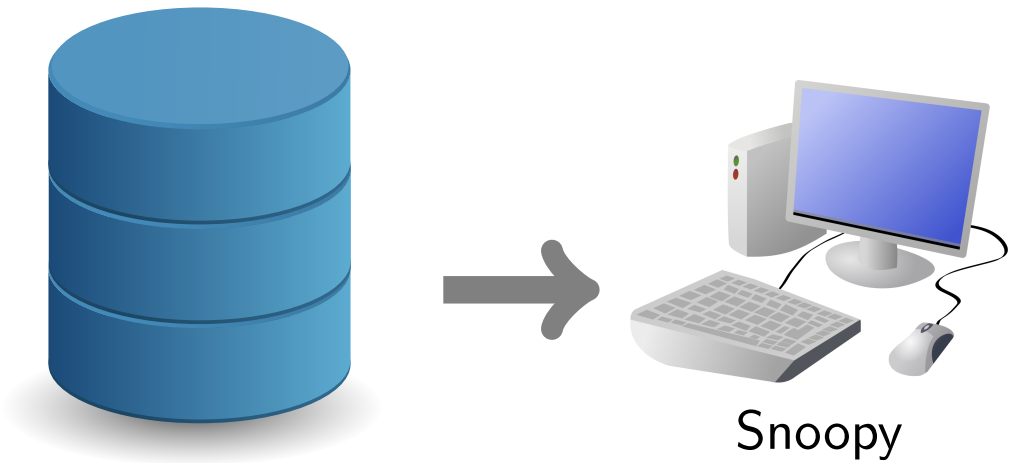
Snoopy

modelling & simulation only in Snoopy

WP4

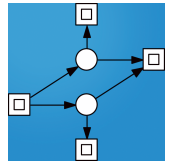


User scenario

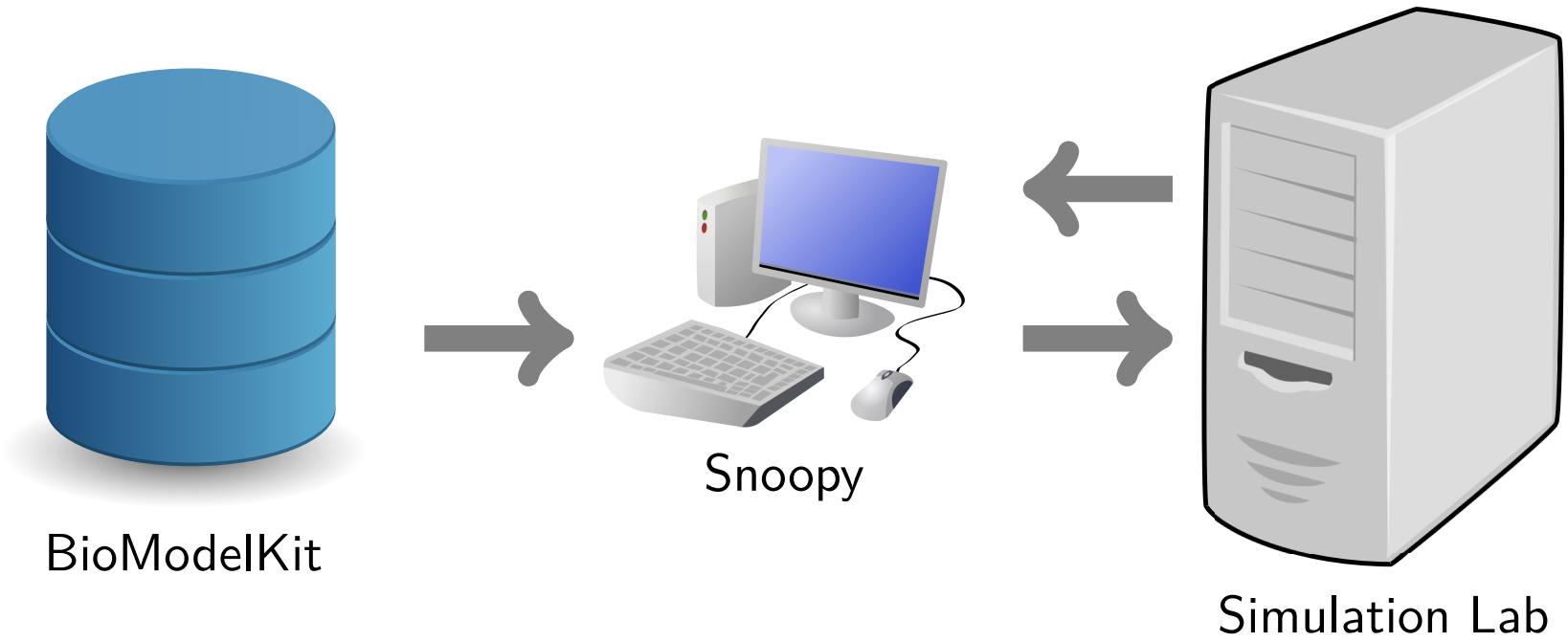


retrieve models from BioModelKit & simulate in Snoopy

WP4

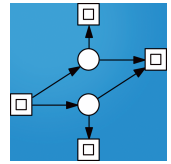


User scenario



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

WP5



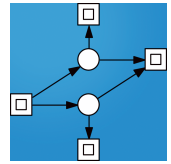
Outline of the “simulation database”

a. Predecessor WPs: BTU-WP4

b. Successor WPs: BTU-WP6

- Design of a server for the database based processing of results of simulation
  - Requirements analysis
  - Technology selection
  - Design of the database schema
  - Design of the user interface

WP6



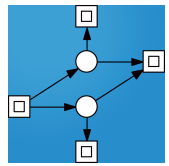
Implementation of the “simulation database”

a. Predecessor WPs: BTU-WP5, OvGUM-WP2

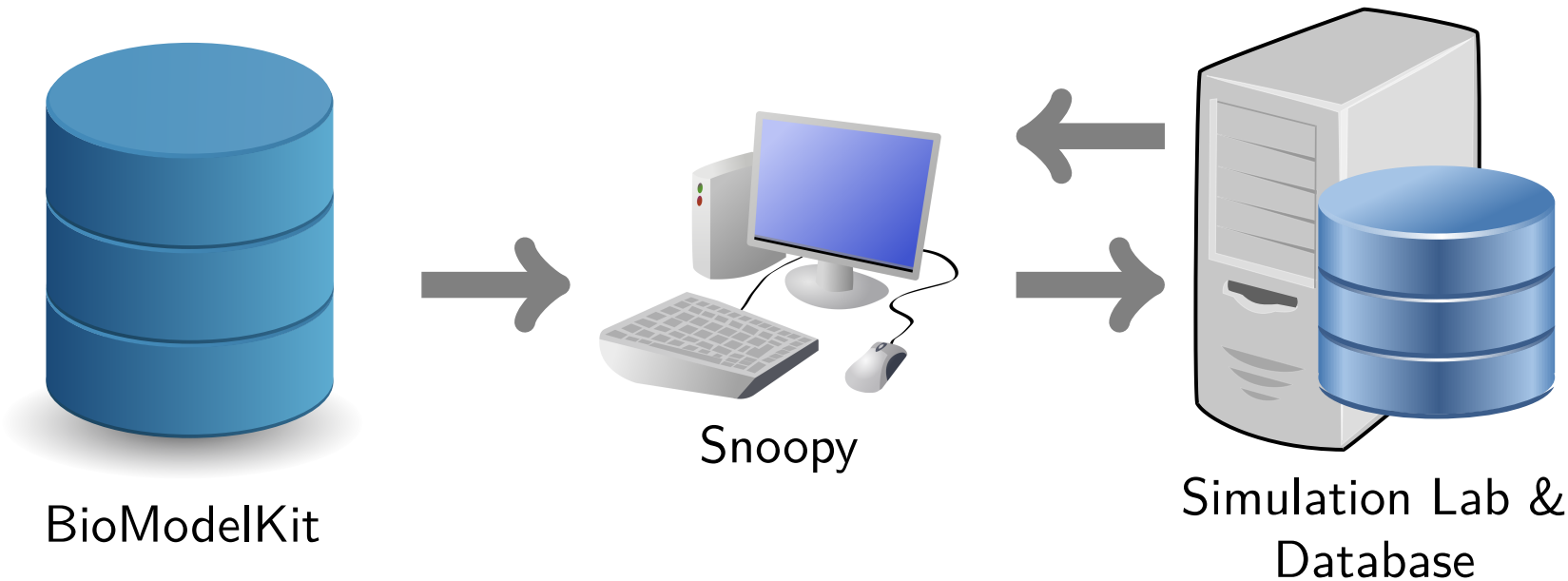
b. Successor WPs: BTU-WP7

- Implementation of the developed concept from WP5
  - Setting up database management system
  - Transferring database schema into the DBMS
  - Implementation of the user interface and database queries
  - Test and optimisation of the database with test or real data
  - Specifications to the data saving

WP6

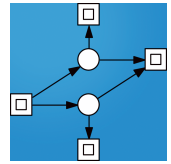


User scenario



retrieve wild-type & alternative models from BioModelKit, create simulation experiments in Snoopy, run experiments in simulation lab & returns clustered results from database

WP7



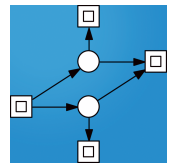
Preparation of the user manuals

a. Predecessor WPs: BTU-WP1, BTU-WP6

b. Successor WPs: -/-

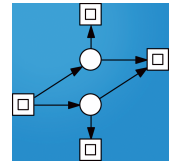
- Preparation of manuals and online tutorials for the use of coloured hybrid Petri nets as well as the simulation laboratory, incl. database for the processing of the results of simulation

Milestones



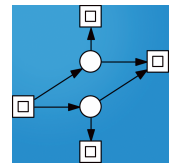
	2013				2014				2015			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
WP1		M1										
WP2				M2								
WP3						M3						
WP4								M4				
WP5								M4				
WP6												M5
WP7												M6

Next steps...



Coloured hybrid Petri nets ($\mathcal{HPN}^{\mathcal{C}}$)

- Uniting the modelling concepts coloured and hybrid Petri nets in Snoopy
- Expansions around new special modelling features like constants and self-modifying edges with marking-dependent arc weights



Thank you for your attention!