Partial Orders Fit For Work

Jörg Desel FernUniversität in Hagen

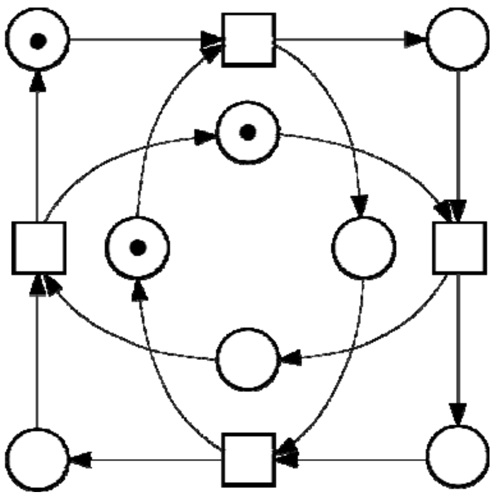
based on a talk at the
Carl Adam Petri Memorial Symposium
last week in Berlin

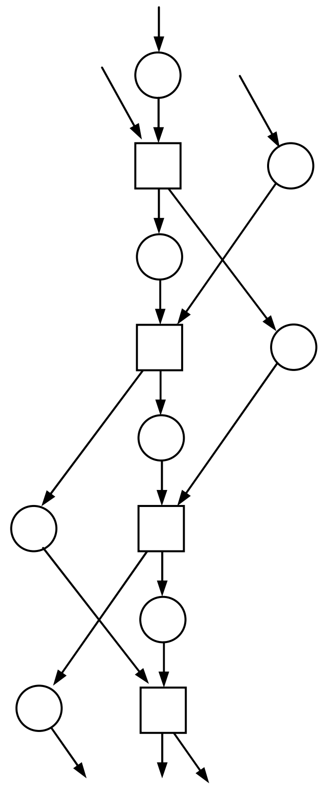
Partial Orders Fit For Work

PART I:
A proof using partial orders
(occurrence nets)
work done in 1988 at GMD

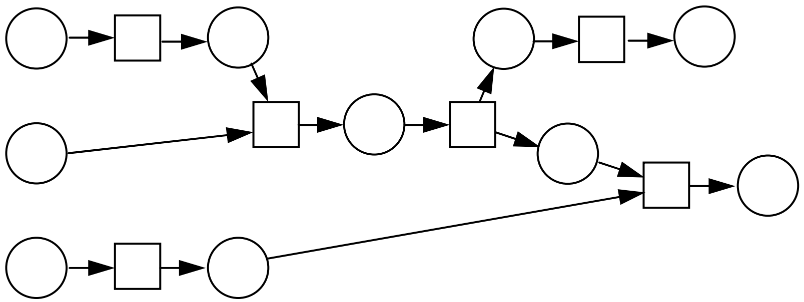
PART II:
Process Model Verification, Validation and Synthesis
Based on Partial Orders
(VIPtool)
work done from 1996 until today

PART I:
A proof using partial orders (occurrence nets) work done in 1988 at GMD



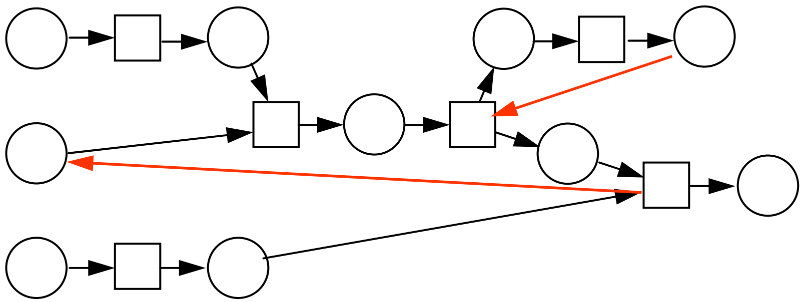


An **occurrence net** (B,E,K)



An **occurrence net** (B,E,K)

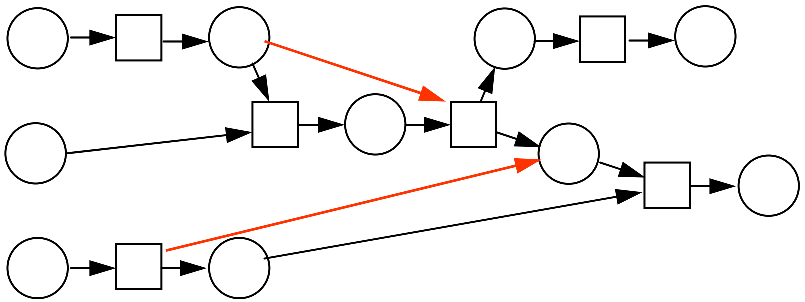
acyclic



An **occurrence net** (B,E,K)

acyclic

places unbranched

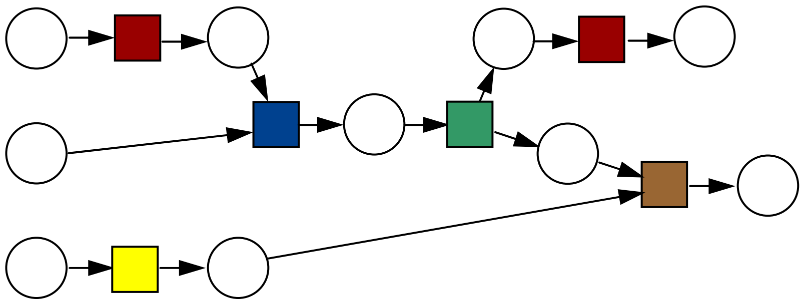


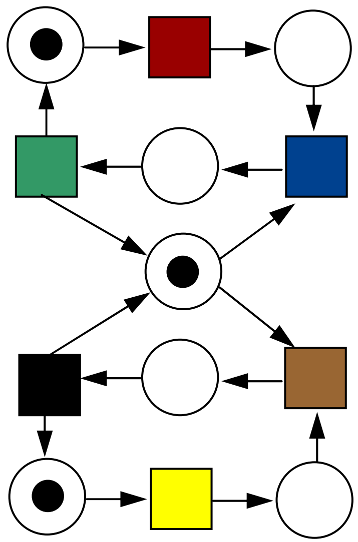
An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

acyclic

places unbranched

maps to the Petri net





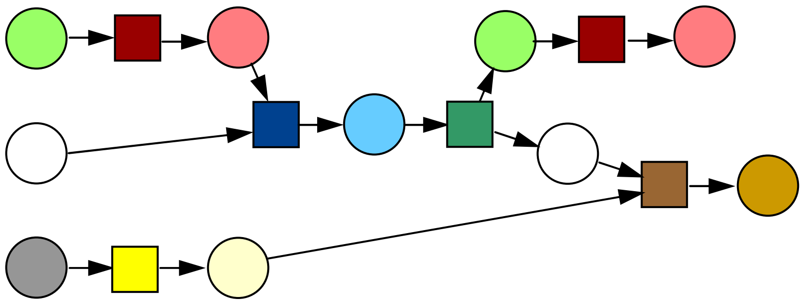
Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

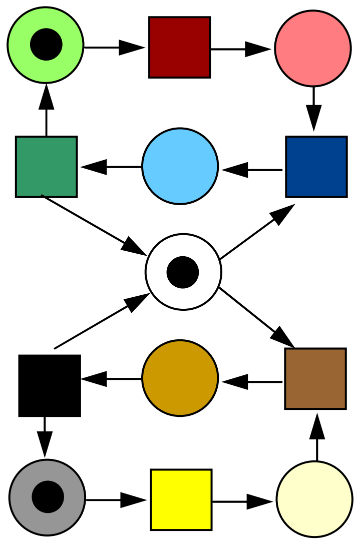
An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

acyclic

places unbranched

maps to the Petri net





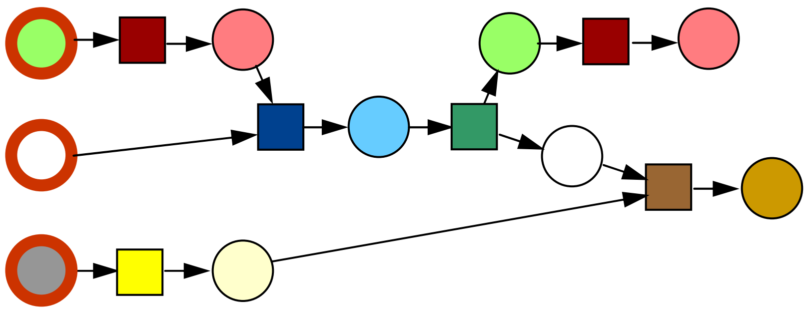
An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

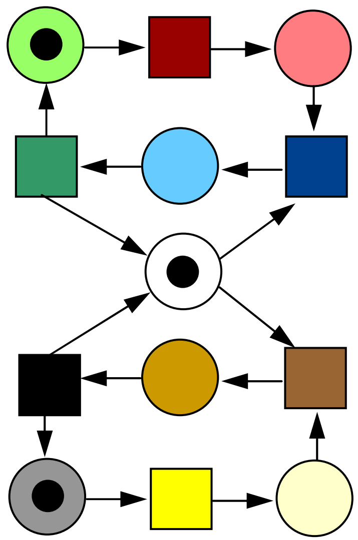
acyclic

places unbranched

maps to the Petri net

minimal places map to tokens





An **occurrence net** (B,E,K)

of a **Petri net** (S,T,F)

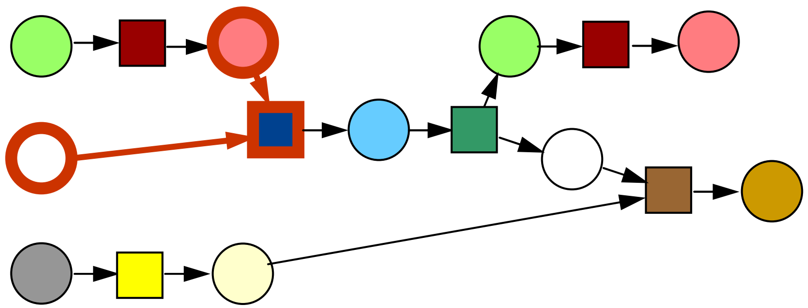
acyclic

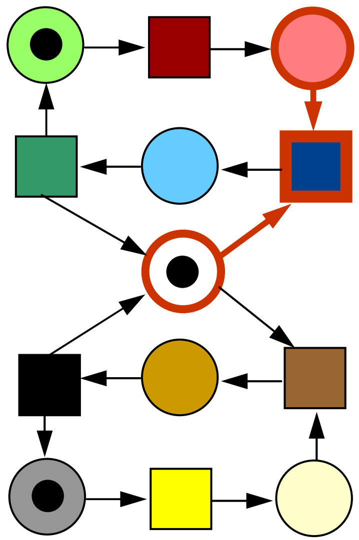
places unbranched

maps to the Petri net

minimal places map to tokens

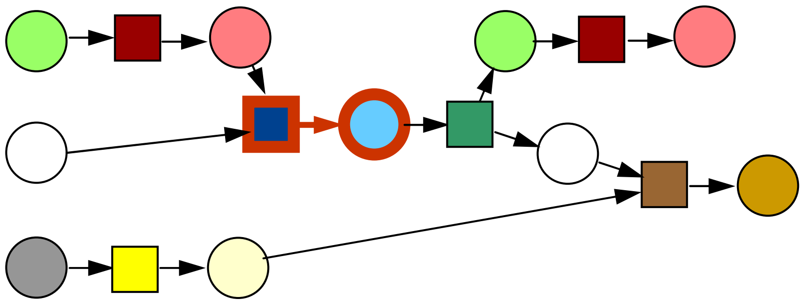
presets of transitions map to presets of transitions

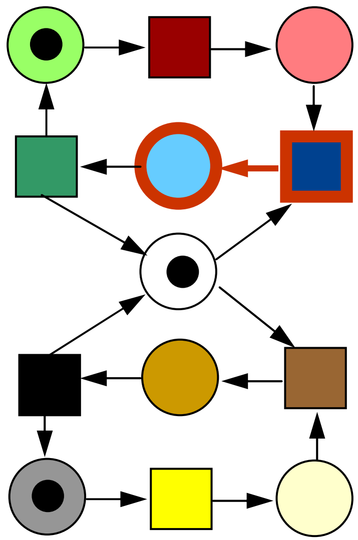




An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

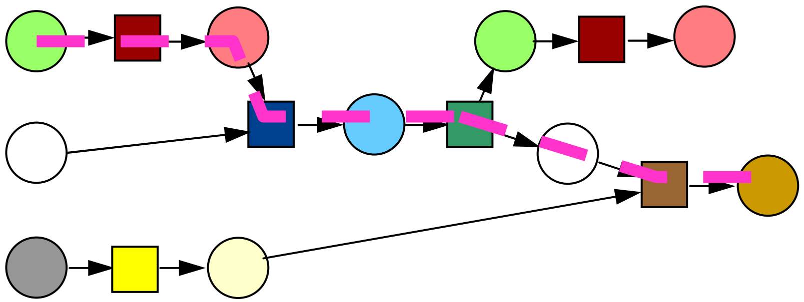
acyclic
places unbranched
maps to the Petri net
minimal places map to tokens
presets of transitions map to presets of transitions
postsets of transitions map to postsets of transitions

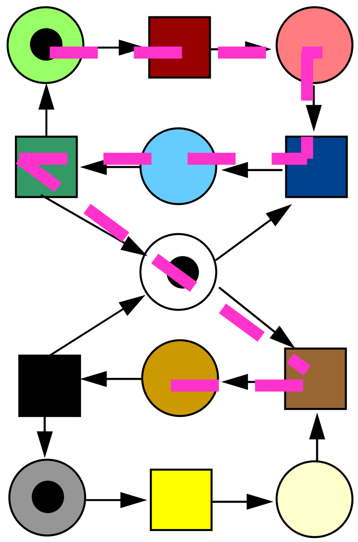




An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

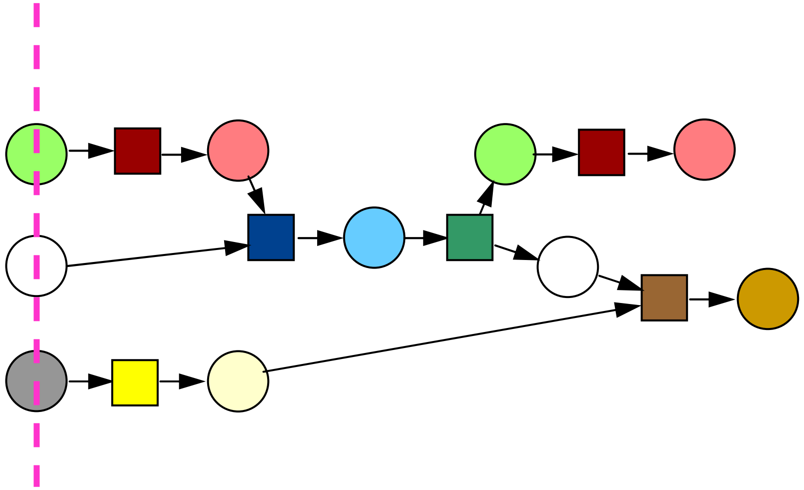
Lemma: paths map to paths

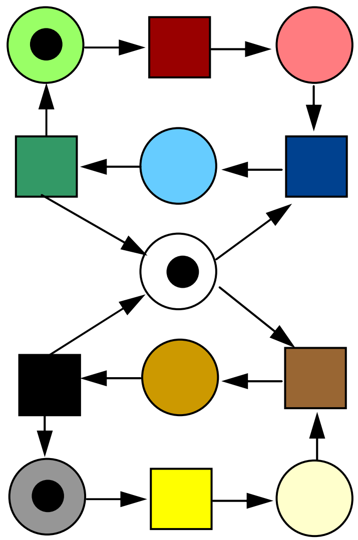




An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

Lemma: finite cuts
(max. sets of mutually concurrent places)
map to reachable markings

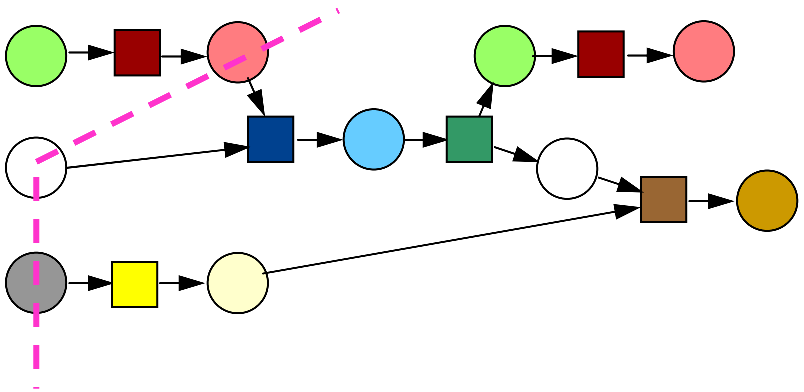


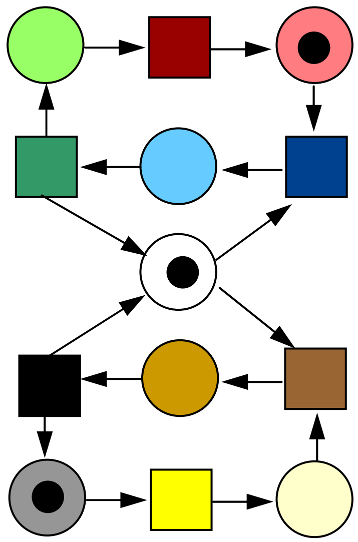


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

Lemma: finite cuts (max. sets of mutually concurrent places) map to reachable markings

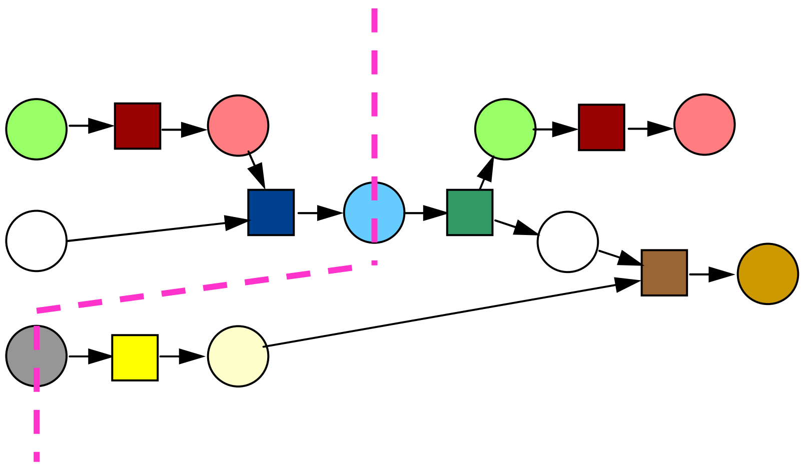


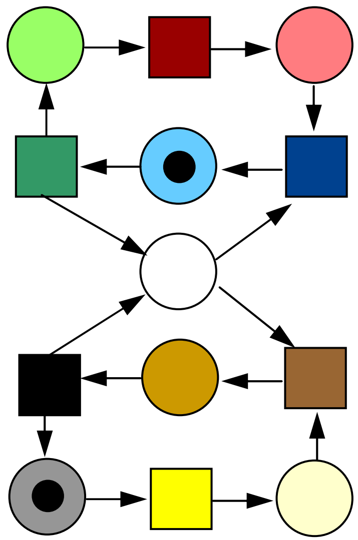


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

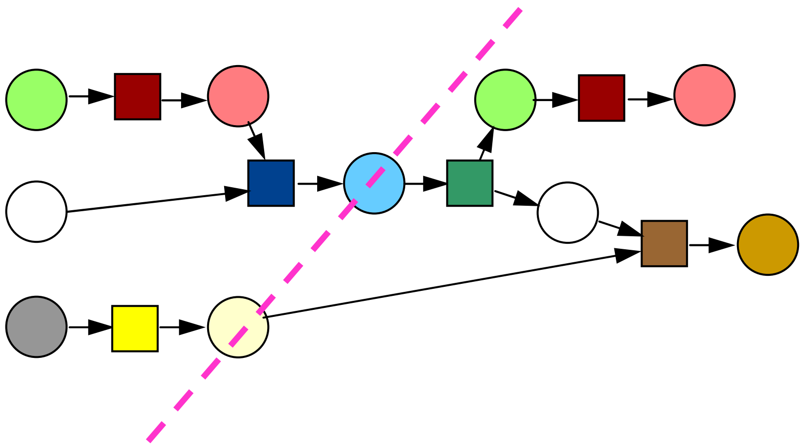
Lemma: finite cuts
(max. sets of mutually concurrent places)
map to reachable markings

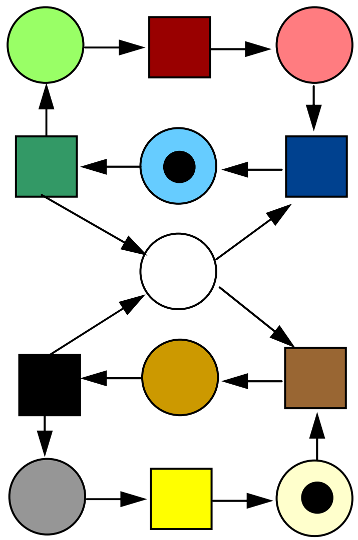




An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

Lemma: finite cuts
(max. sets of mutually concurrent places)
map to reachable markings



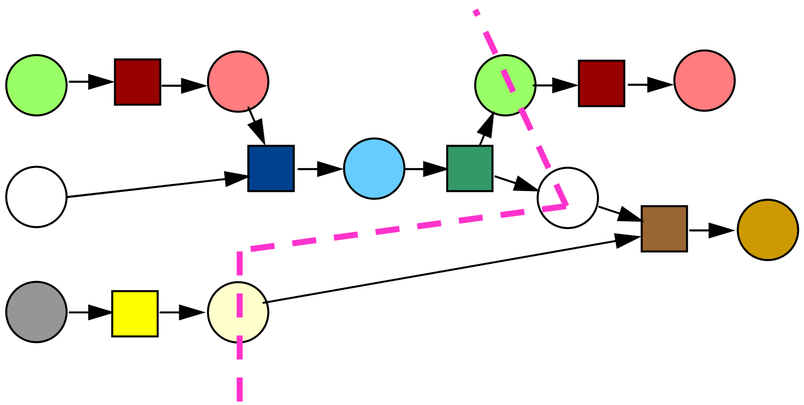


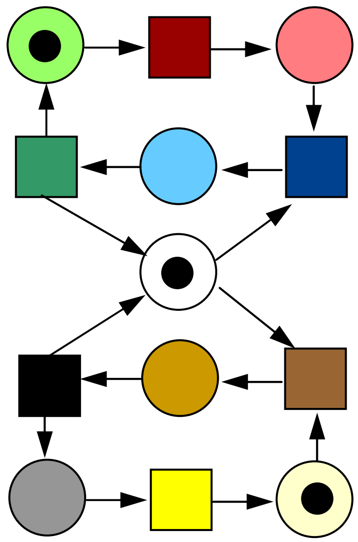
An **occurrence net** (B,E,K)

of a **Petri net** (S,T,F)

Lemma: finite cuts
(max. sets of mutually concurrent places)
map to

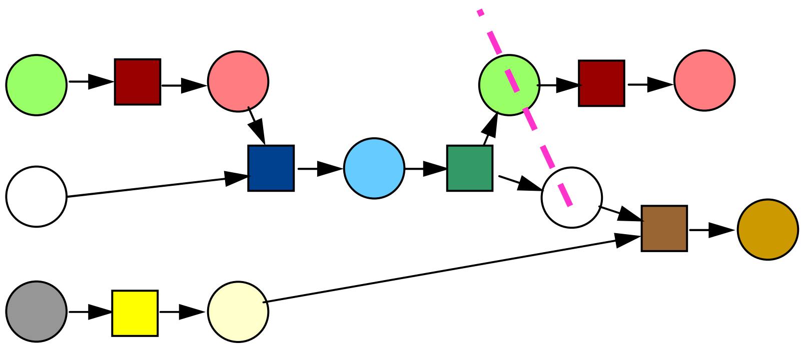
reachable markings

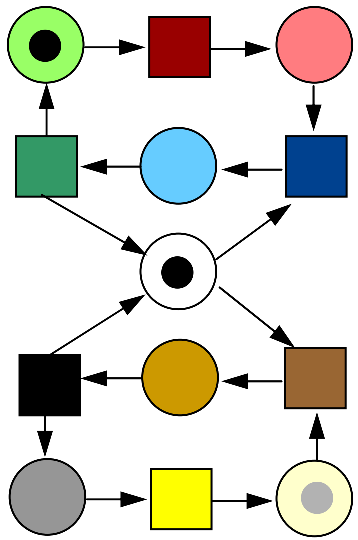




An **occurrence net** (B,E,K) of a **Petri net** (S,T,F)

Corollary: finite co-sets (sets of mutually concurrent places) map to reachable sub-markings



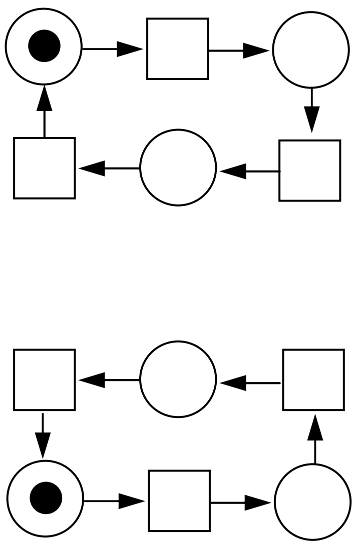


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

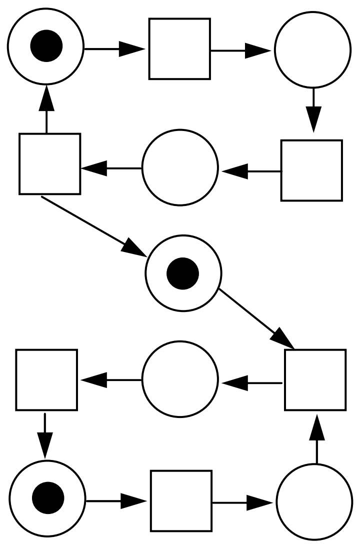
Theorem: each connected live and bounded Petri net is strongly connected

Theorem: each connected live and bounded Petri net is strongly connected

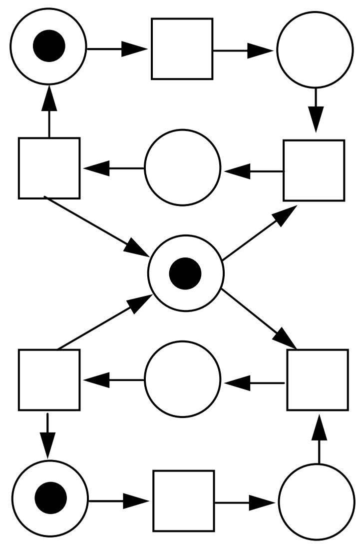
not connected



connected



strongly connected

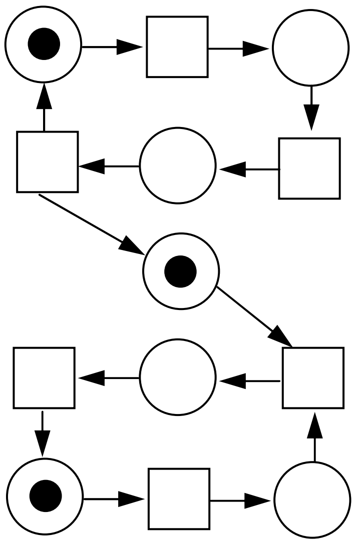


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

Theorem: each connected live and bounded Petri net is strongly connected

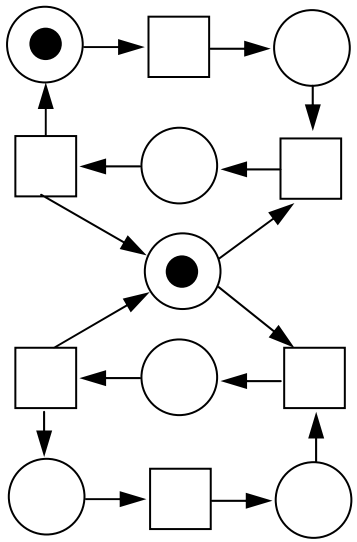
live, not bounded

each transition can
always occur again

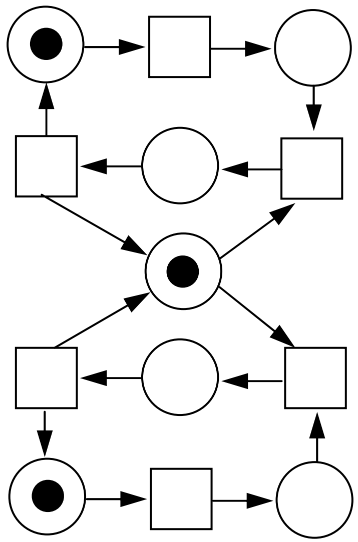


bounded, not live

each place has a bound
(maximal number of tokens)

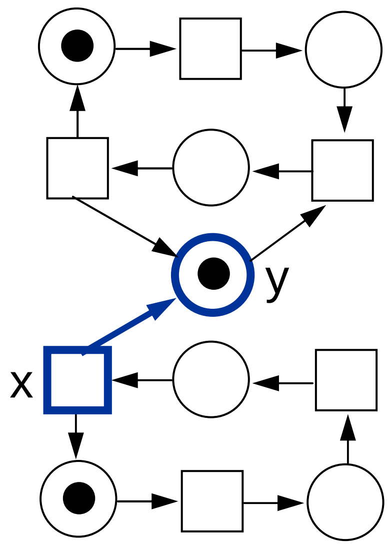


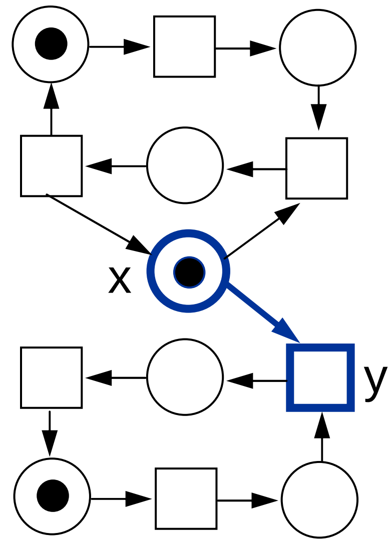
live and bounded



Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

Theorem: each connected live and bounded Petri net is strongly connected





Lemma: if a net is connected but not strongly connected then for some arc (*x*,*y*) there is no directed path from *y* to *x*

Corollary: if, in a connected net, for each arc (*x*,*y*) there is a path from *y* to *x*, then the net is strongly connected

Theorem: each connected live and bounded Petri net is strongly connected

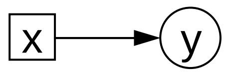
Proof: Consider a live and bounded connected net and an arbitrary arc (x,y).

We will show that there is a path from y to x.

Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (*x*,*y*).

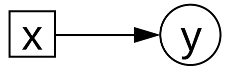
Case 1: *x* is a transition and *y* is a place.



Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (x,y).

Case 1: x is a transition and y is a place.

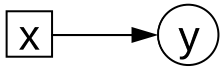


Let b be the bound of y (exists, because the net is bounded).

Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (x,y).

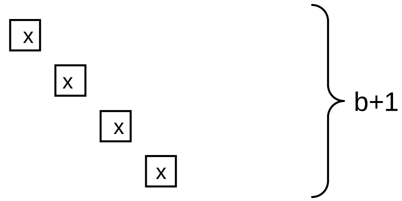
Case 1: x is a transition and y is a place.



Let b be the bound of y (exists, because the net is bounded).

Assume an occurrence net with $b + 1$ occurrences of the transition x.

(exists, because the net is live).

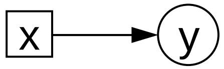


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

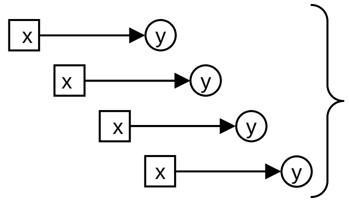
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (*x*,*y*).

Case 1: *x* is a transition and y is a place.



Let *b* be the bound of *y* (exists, because the net is bounded).
Assume an occurrence net with *b* + 1 occurrences of the transition *x*.
(exists, because the net is live).



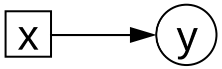
b+1

Since postsets of the occurrences of *x* are respected,
each occurrence of *x* has an occurrence of *y* in its postset.

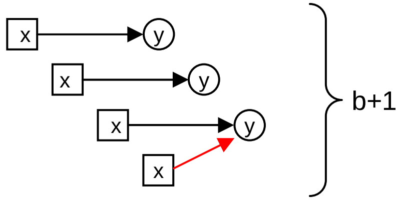
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (*x*,*y*).

Case 1: *x* is a transition and y is a place.



Let *b* be the bound of *y* (exists, because the net is bounded).
Assume an occurrence net with *b* + 1 occurrences of the transition *x*.
(exists, because the net is live).



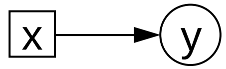
Since postsets of the occurrences of *x* are respected,
each occurrence of *x* has an occurrence of *y* in its postset.

Since places in occurrence nets are not branched,
all these occurrences of *y* are distinct.

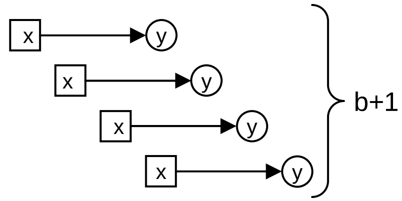
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (x,y).

Case 1: x is a transition and y is a place.



Let b be the bound of y (exists, because the net is bounded).
Assume an occurrence net with b + 1 occurrences of the transition x.
(exists, because the net is live).



Since postsets of the occurrences of x are respected,
each occurrence of x has an occurrence of y in its postset.

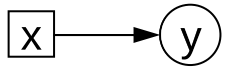
Since places in occurrence nets are not branched,
all these occurrences of y are distinct.

Since b is its bound, the place y never carries more than b tokens.
Hence no co-set contains all b+1 occurrences of y.

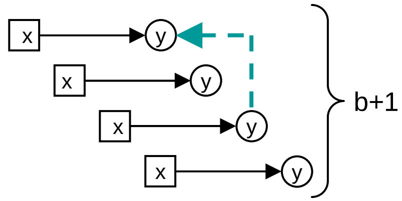
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (*x*,*y*).

Case 1: *x* is a transition and *y* is a place.



Let *b* be the bound of y (exists, because the net is bounded).
Assume an occurrence net with *b* + 1 occurrences of the transition *x*.
(exists, because the net is live).



Since postsets of the occurrences of *x* are respected,
each occurrence of *x* has an occurrence of *y* in its postset.

Since places in occurrence nets are not branched,
all these occurrences of *y* are distinct.

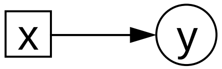
Since *b* is its bound, the place *y* never carries more than *b* tokens.
Hence no co-set contains all *b*+1 occurrences of *y*.

So at least two of these occurrences are connected by a path.

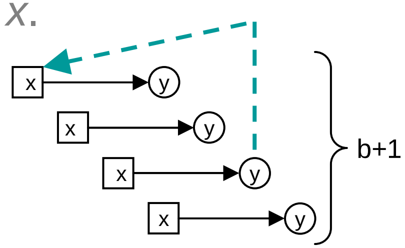
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (x,y).

Case 1: x is a transition and y is a place.



Let b be the bound of y (exists, because the net is bounded).
Assume an occurrence net with $b + 1$ occurrences of the transition x.
(exists, because the net is live).



Since postsets of the occurrences of x are respected,
each occurrence of x has an occurrence of y in its postset.

Since places in occurrence nets are not branched,
all these occurrences of y are distinct.

Since b is its bound, the place y never carries more than b tokens.
Hence no co-set contains all b+1 occurrences of y.

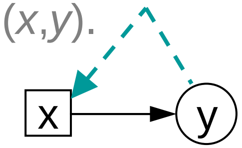
So at least two of these occurrences are connected by a path.

Again since places in occurrence nets are not branched,
this path goes through an occurrence of x.

So there is a path from an occurrence of y to an occurrence of x.

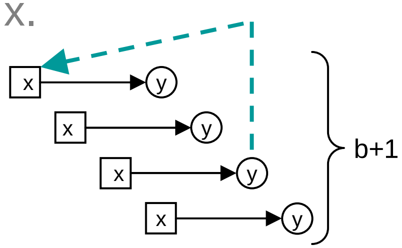
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (*x*,*y*).



Case 1: *x* is a transition and *y* is a place.

Let *b* be the bound of *y* (exists, because the net is bounded).
Assume an occurrence net with *b* + 1 occurrences of the transition x.
(exists, because the net is live).



Since postsets of the occurrences of *x* are respected,
each occurrence of *x* has an occurrence of *y* in its postset.

Since places in occurrence nets are not branched,
all these occurrences of *y* are distinct.

Since *b* is its bound, the place *y* never carries more than *b* tokens.
Hence no co-set contains all *b*+1 occurrences of *y*.

So at least two of these occurrences are connected by a path.

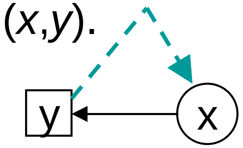
Again since places in occurrence nets are not branched,
this path goes through an occurrence of *x*.

So there is a path from an occurrence of *y* to an occurrence of *x*.

Since paths are mapped to paths, there is a path from *y* to *x*.

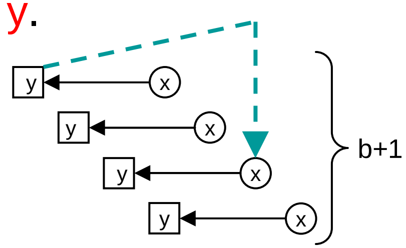
Theorem: each connected live and bounded Petri net is strongly connected

Proof: Consider a live and bounded connected net and an arbitrary arc (*x*,*y*).



Case 2: *x* is a place and y is a transition.

Let *b* be the bound of x (exists, because the net is bounded).
Assume an occurrence net with *b* + 1 occurrences of the transition y.
(exists, because the net is live).



Since presets of the occurrences of *y* are respected,
each occurrence of *y* has an occurrence of *x* in its preset.

Since places in occurrence nets are not branched,
all these occurrences of *x* are distinct.

Since *b* is its bound, the place *x* never carries more than *b* tokens.
Hence no co-set contains all b+1 occurrences of *x*.

So at least two of these occurrences are connected by a path.

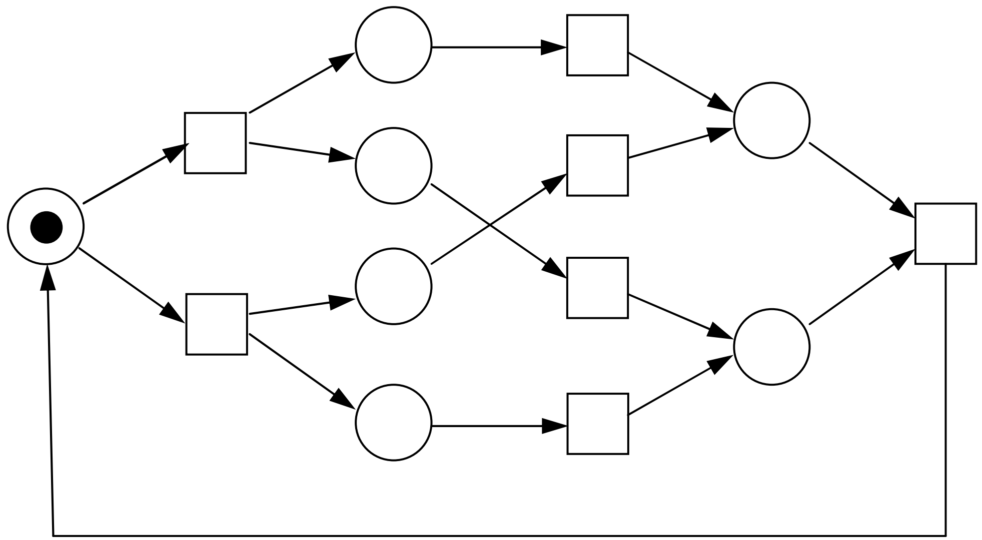
Again since places in occurrence nets are not branched,
this path goes through an occurrence of *y*.

So there is a path from an occurrence of *y* to an occurrence of *x*.

Since paths are mapped to paths, there is a path from *y* to *x*.

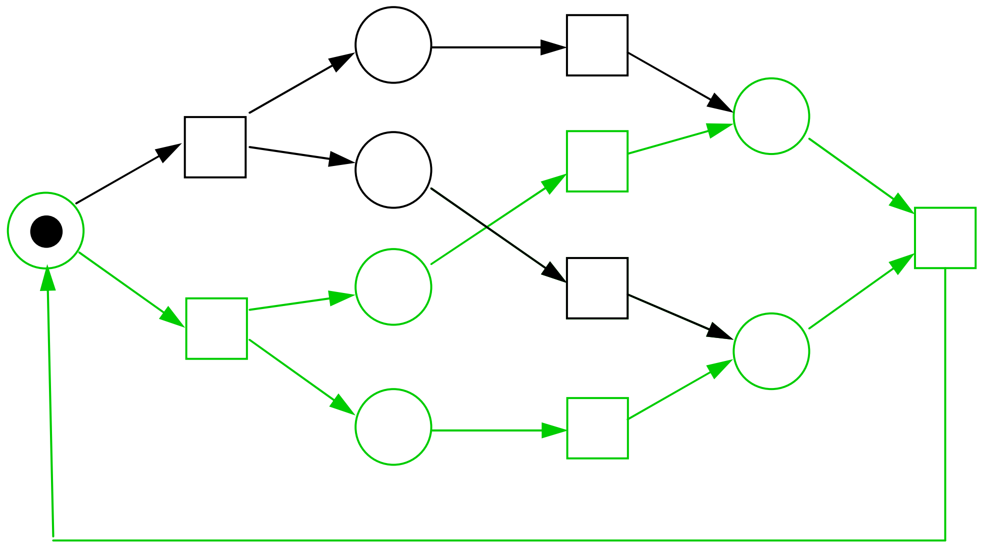
Further Theorems, using the same idea:

each live and bounded extended free-choice net is covered by T-components



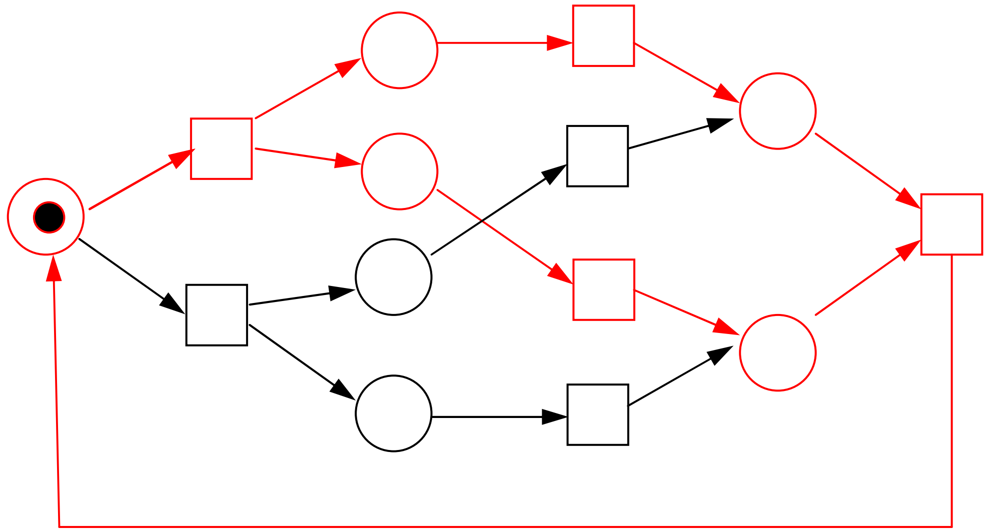
Further Theorems, using the same idea:

each live and bounded extended free-choice net is covered by T-components



Further Theorems, using the same idea:

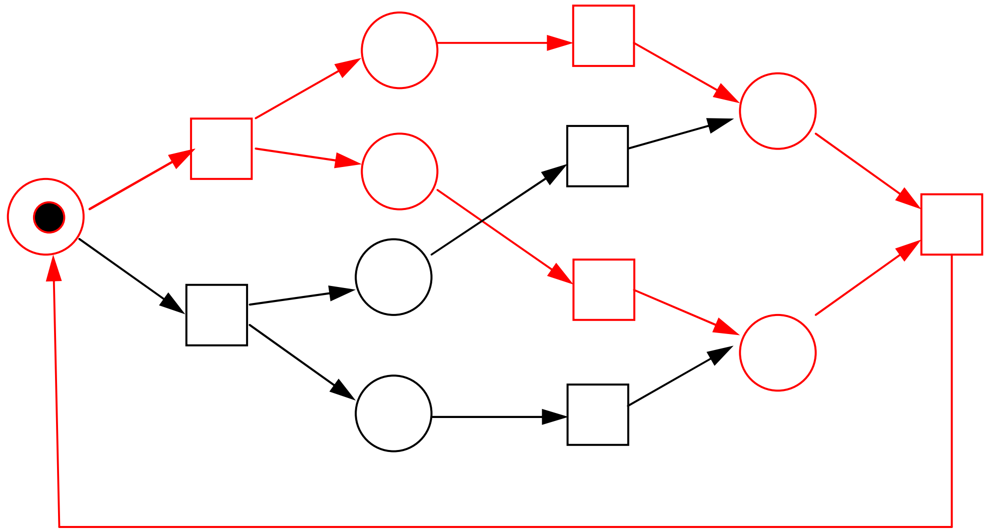
each live and bounded extended free-choice net is covered by T-components



Further Theorems, using the same idea:

each live and bounded extended free-choice net is covered by T-components

each T-invariant is realizable, i.e., corresponds to a cyclic process



PART II:

Process Model Synthesis From Partial Orders (VIPtool)

PART II:

Process Model Synthesis
From Partial Orders
(VIPtool)

VIPtool was originally created in 1996 -1998
by my group and the group of Andreas Oberweis,
Institute AIFB, University of Karlsruhe
(Carl-Adam-Petri award !!)

Simulation for Analysis

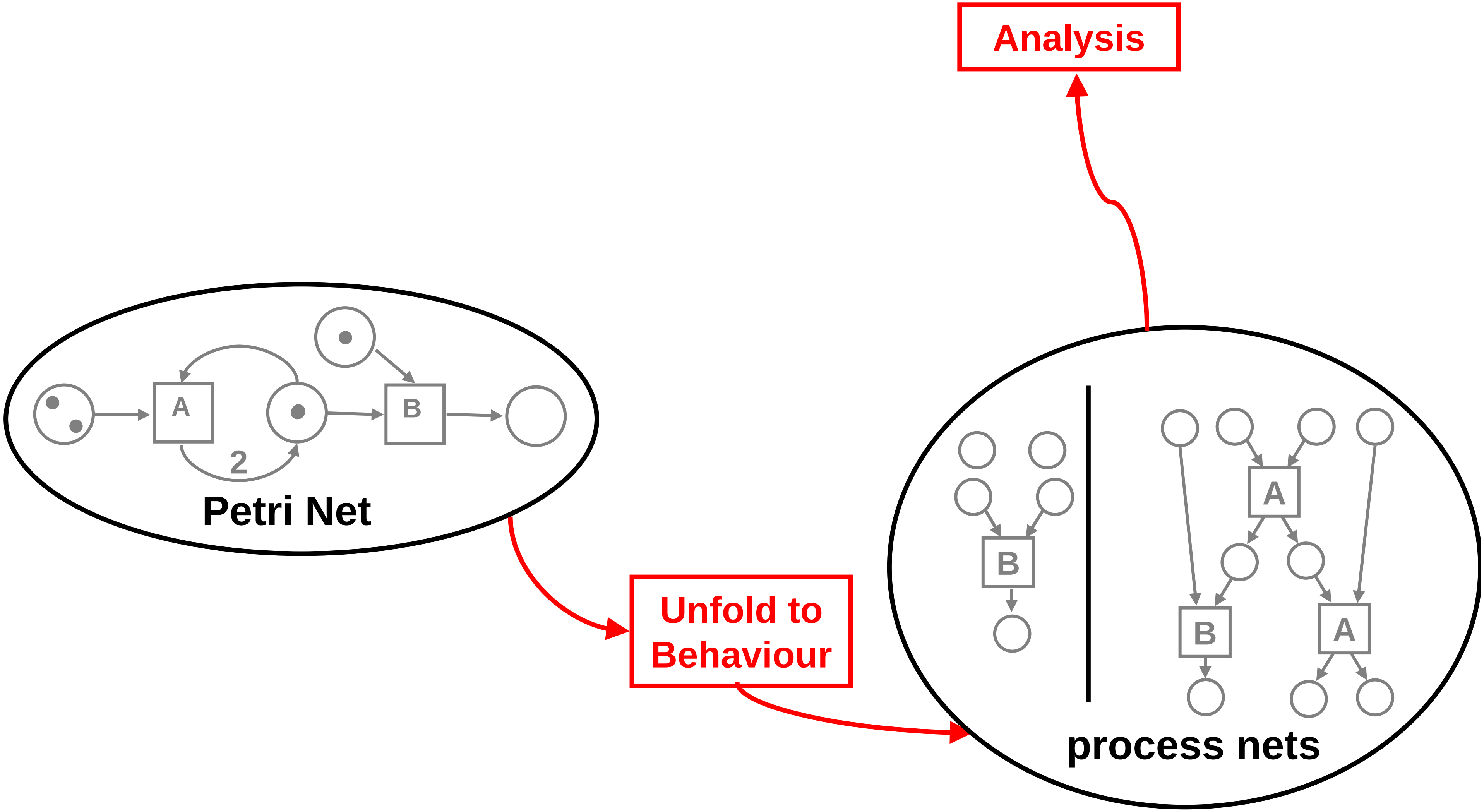
means generation of runs.

- **sequential runs** (occurrence sequences), combined with graphical animation

  $\Rightarrow$ the usual approach

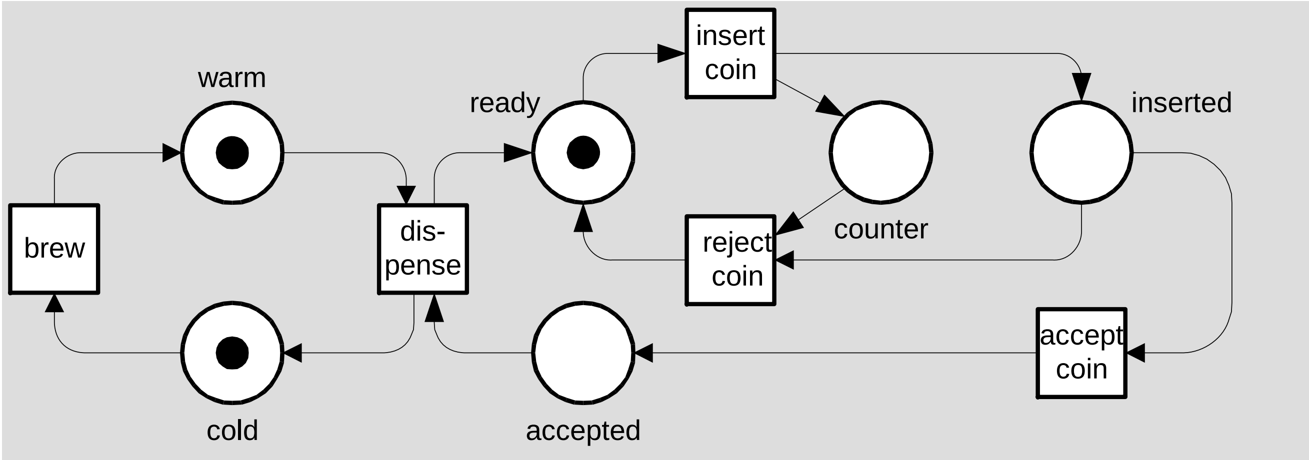
- **non-sequential, causal runs** (process nets)

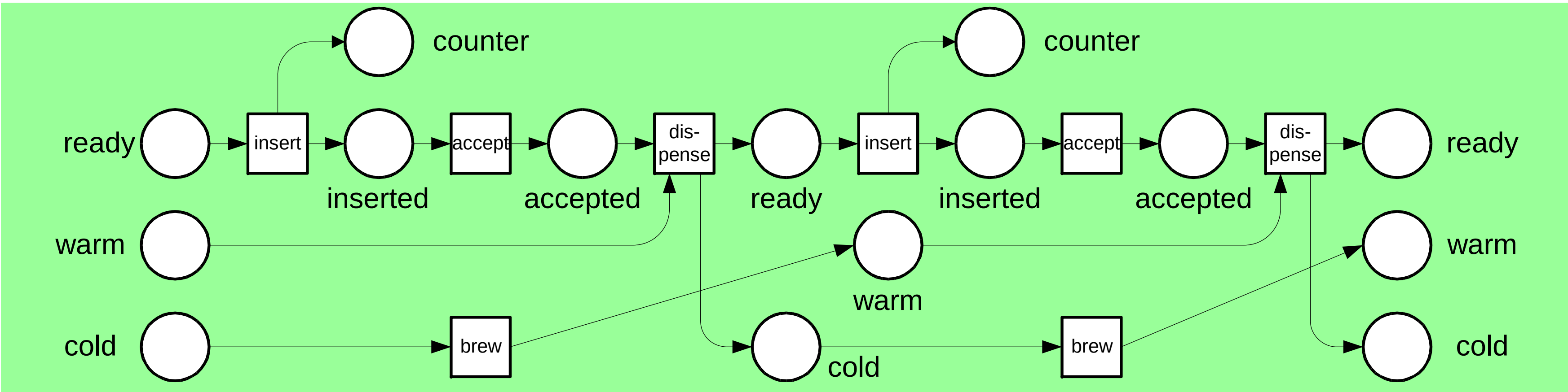
  $\Rightarrow$ the **VIP-approach**



Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

A Coffee Machine with one Process





Efficiency

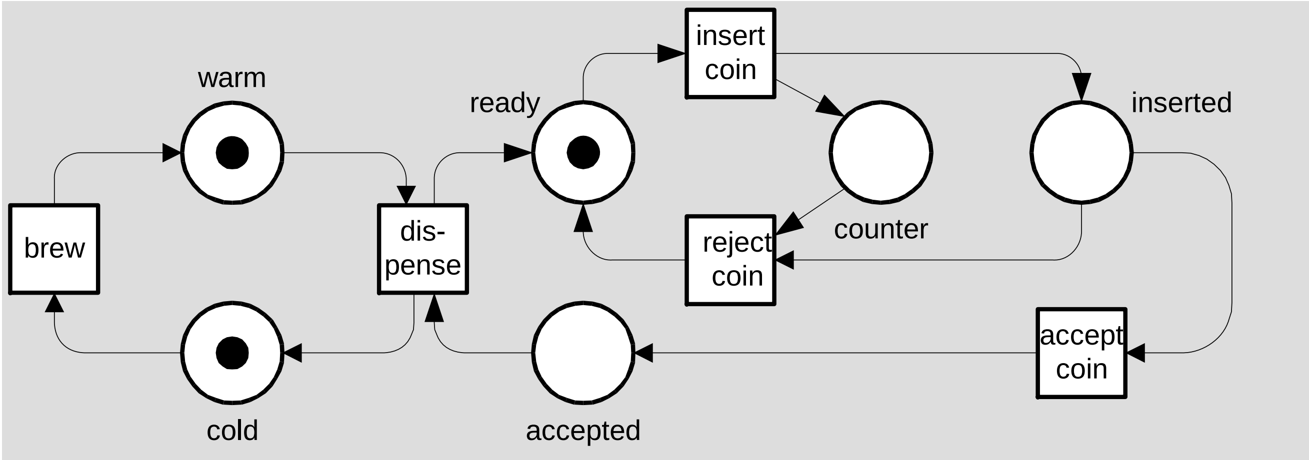
- every occurrence sequence is represented
  by an occurrence sequence of a process net

- every occurrence sequence of a process net
  corresponds to an occurrence sequence of the model

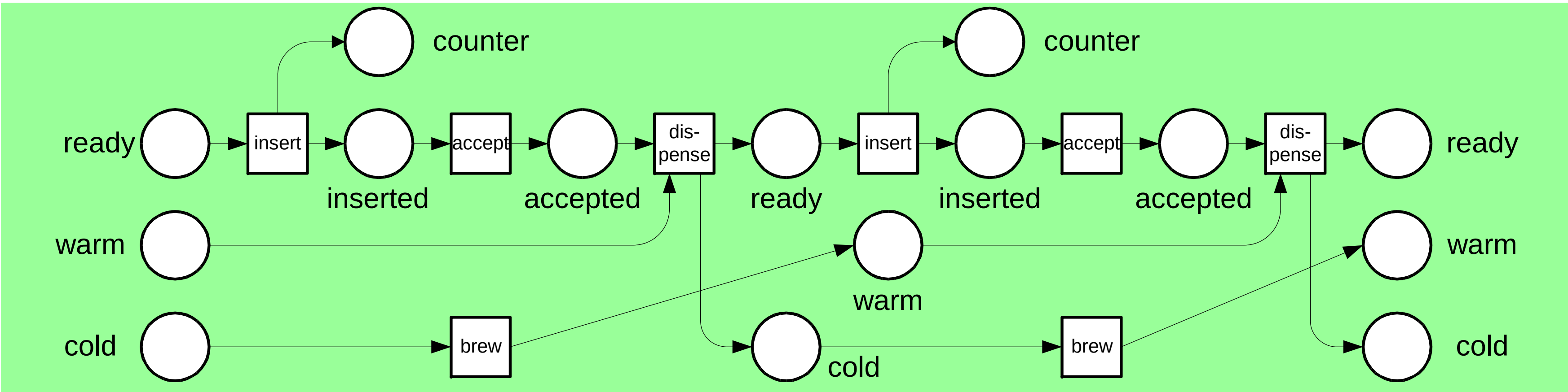
- the number of process nets exceeds the number of
  occurrence sequences significantly
  (exponential in the degree of parallelism)

Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

Advantage 1: Efficiency

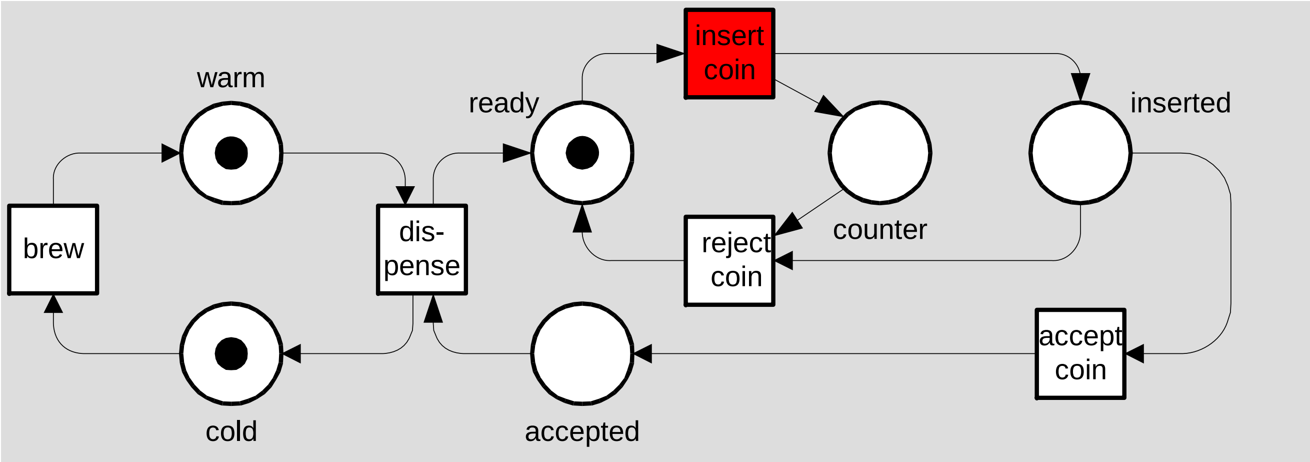


The process net represents 21 different occurrence sequences

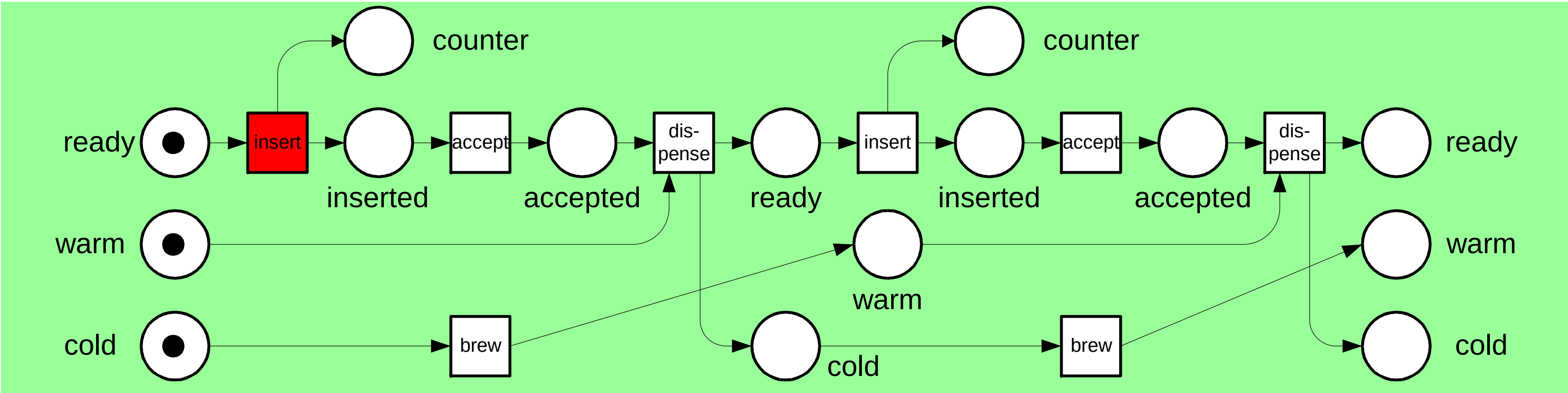


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence

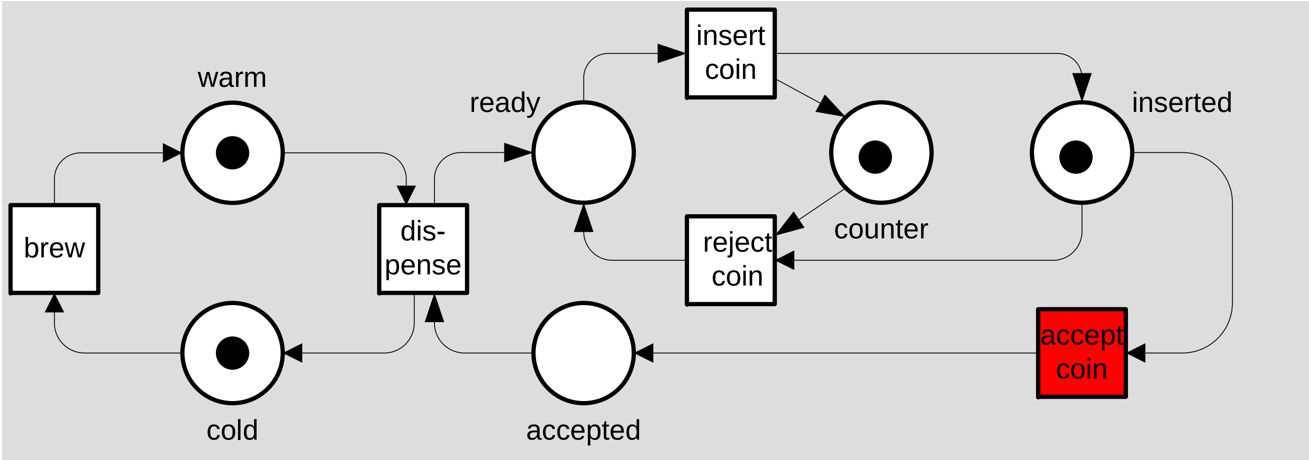


insert

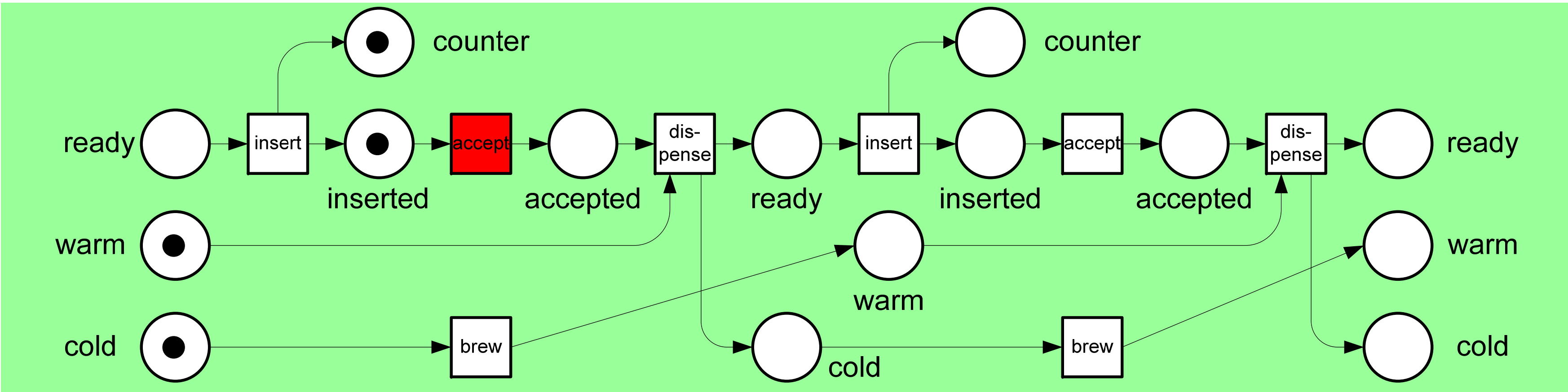


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence

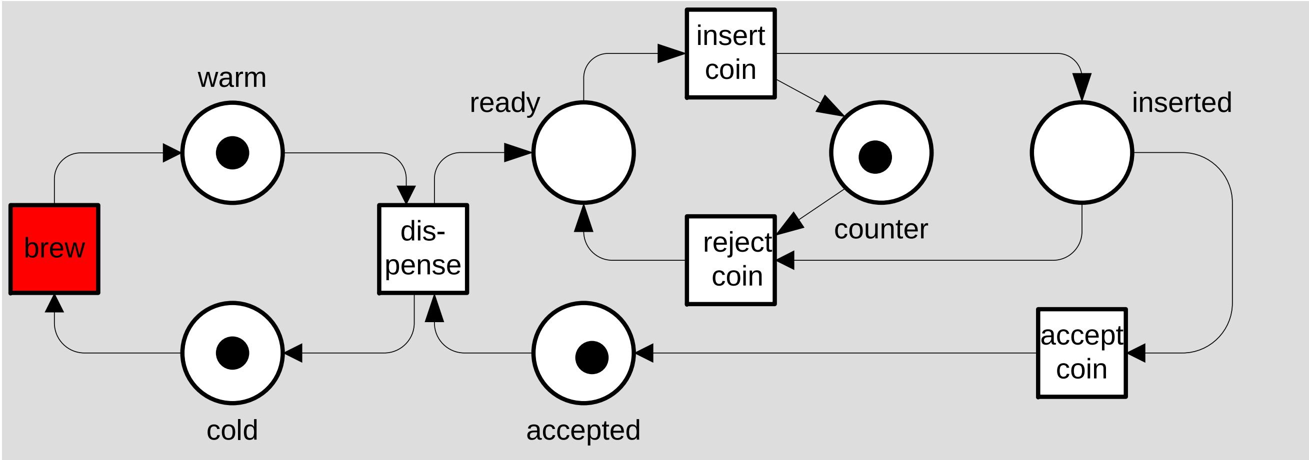


insert, accept

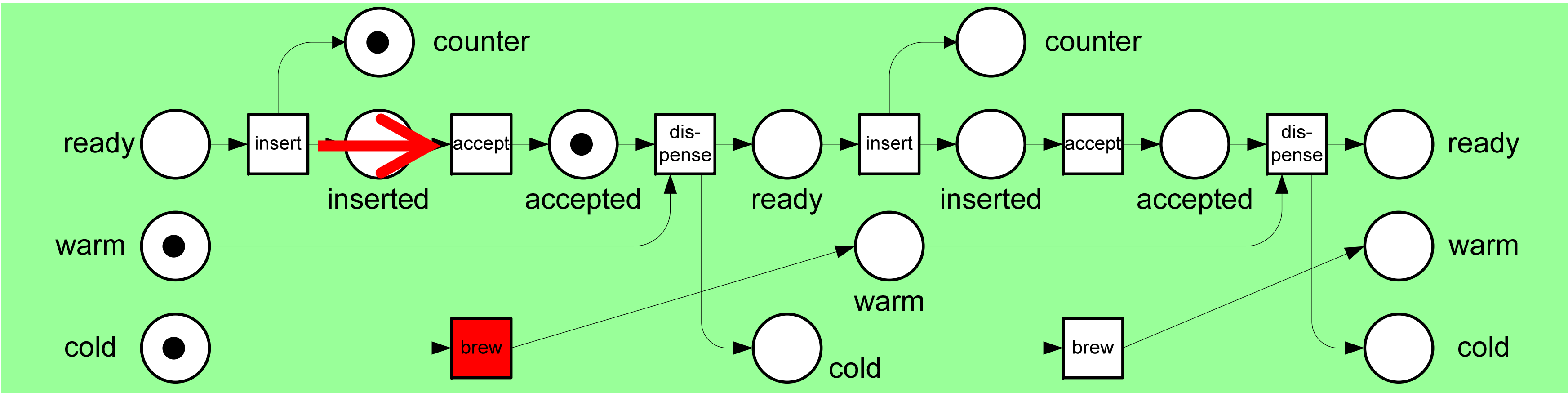


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence

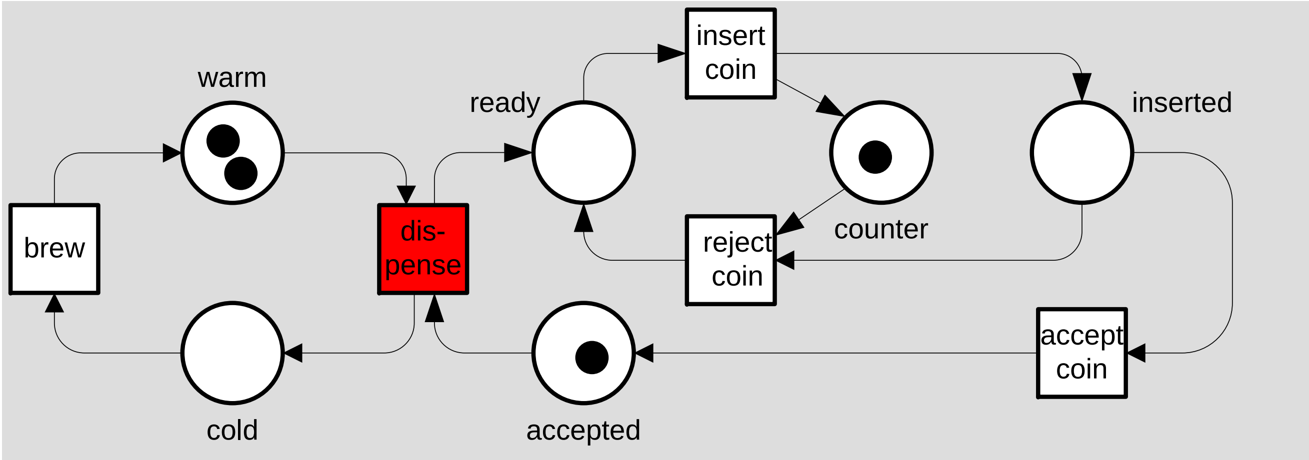


insert, accept, brew

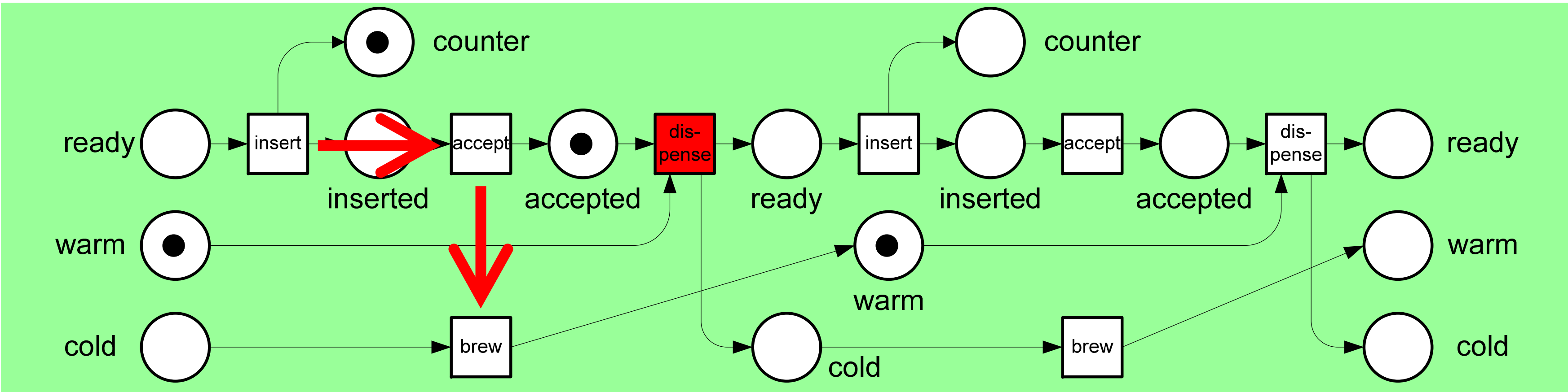


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence

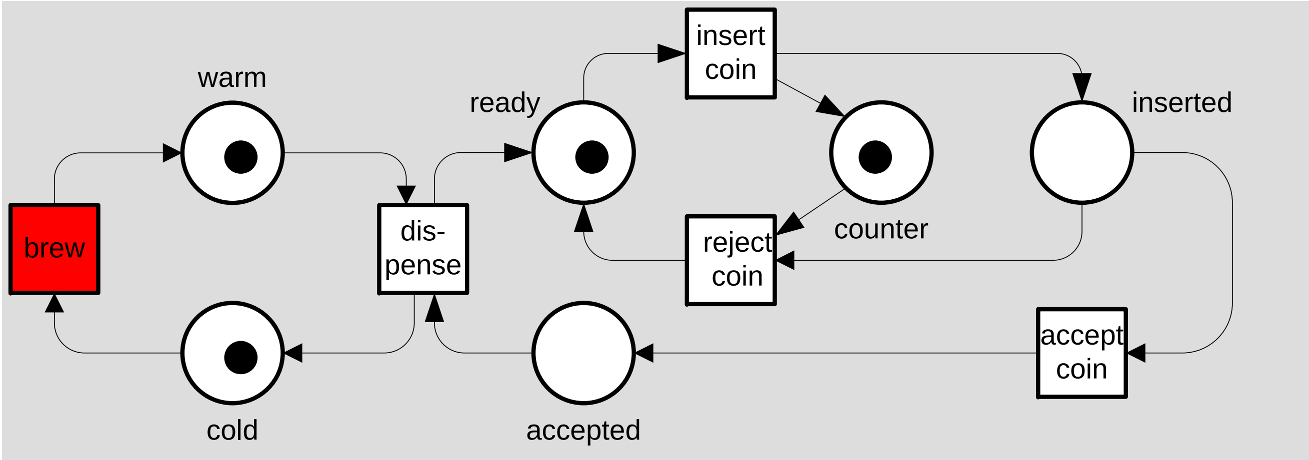


insert, accept, brew, dispense

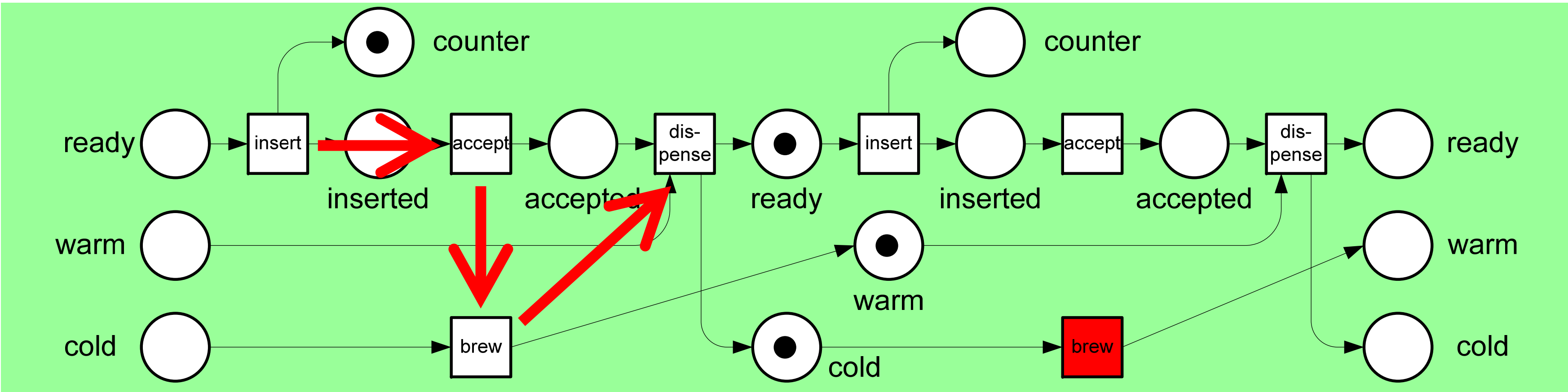


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence

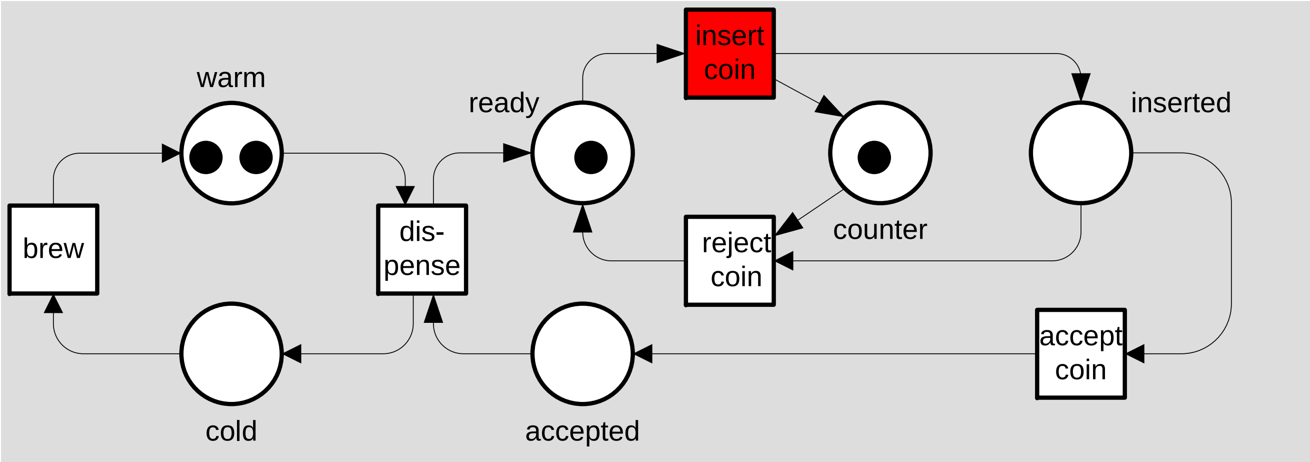


insert, accept, brew, dispense, brew

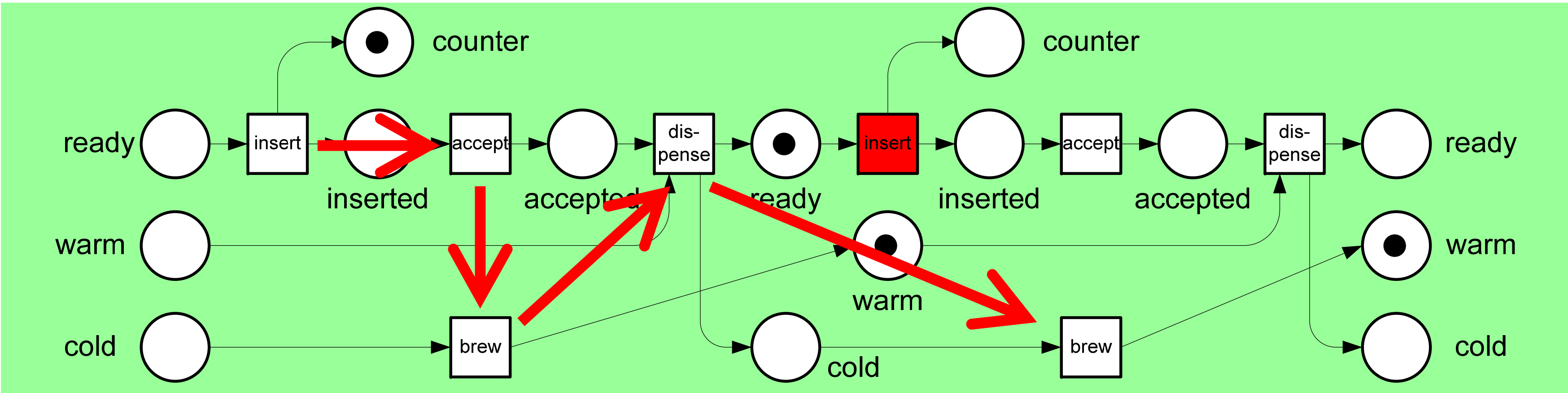


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

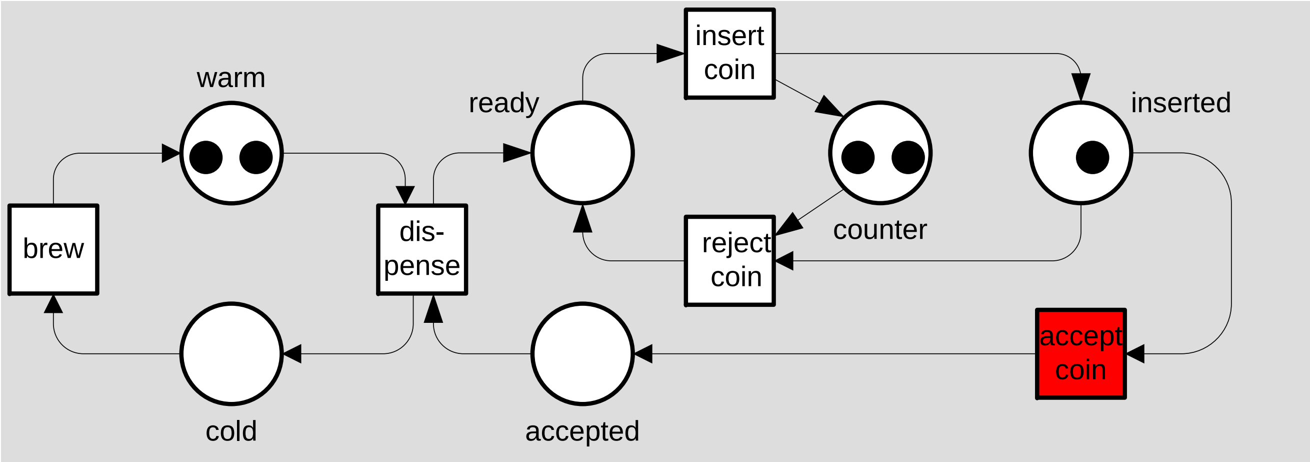
One common occurrence sequence



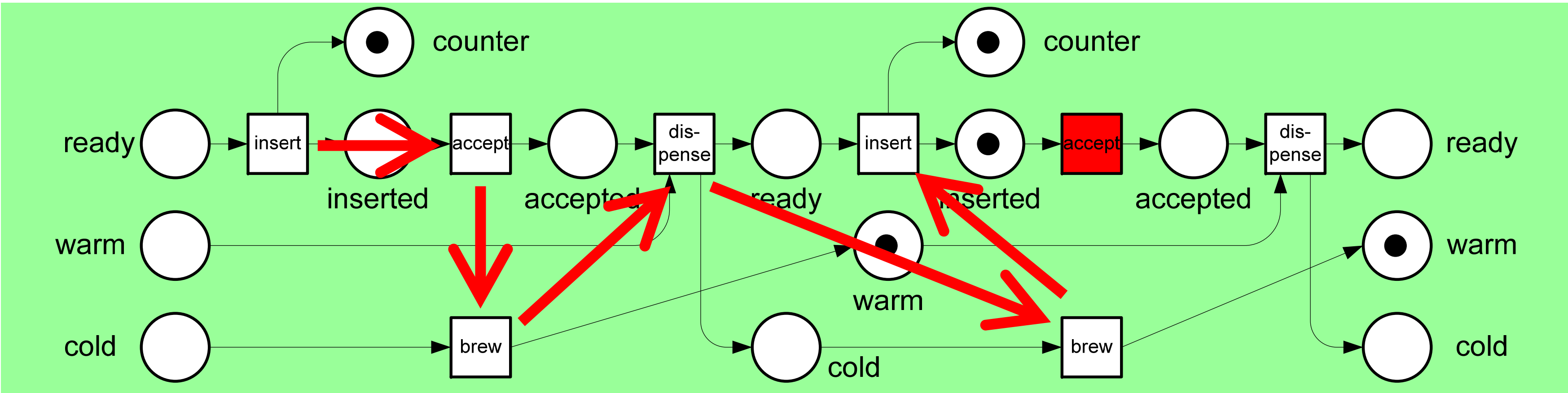
insert, accept, brew, dispense, brew, insert



One common occurrence sequence

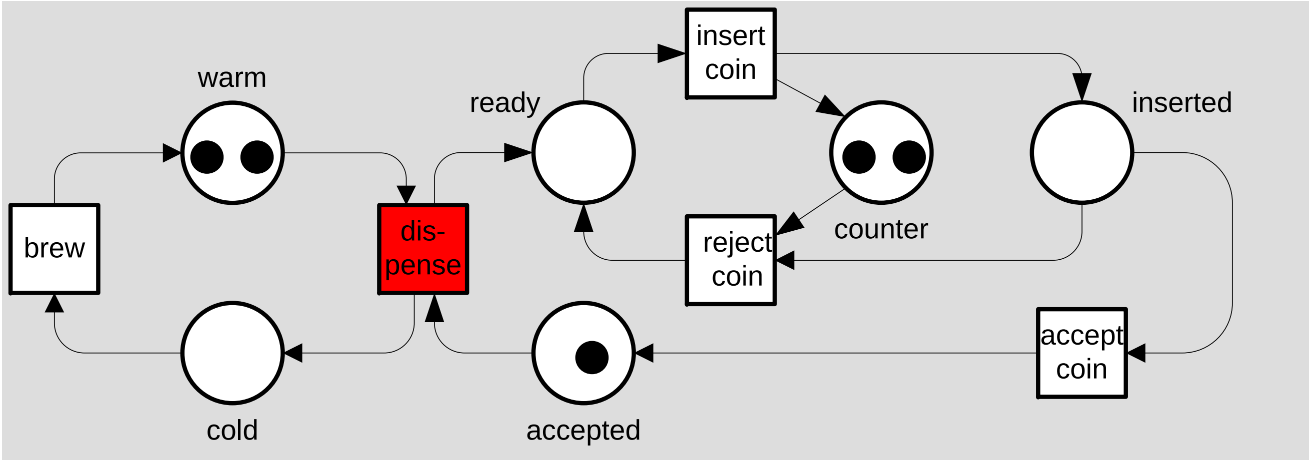


insert, accept, brew, dispense, brew, insert, accept

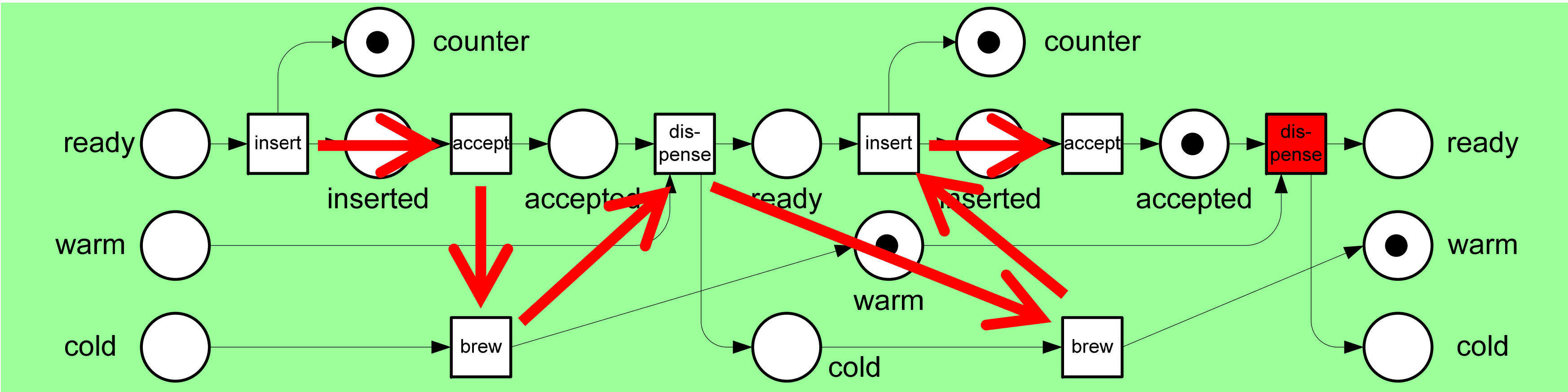


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence

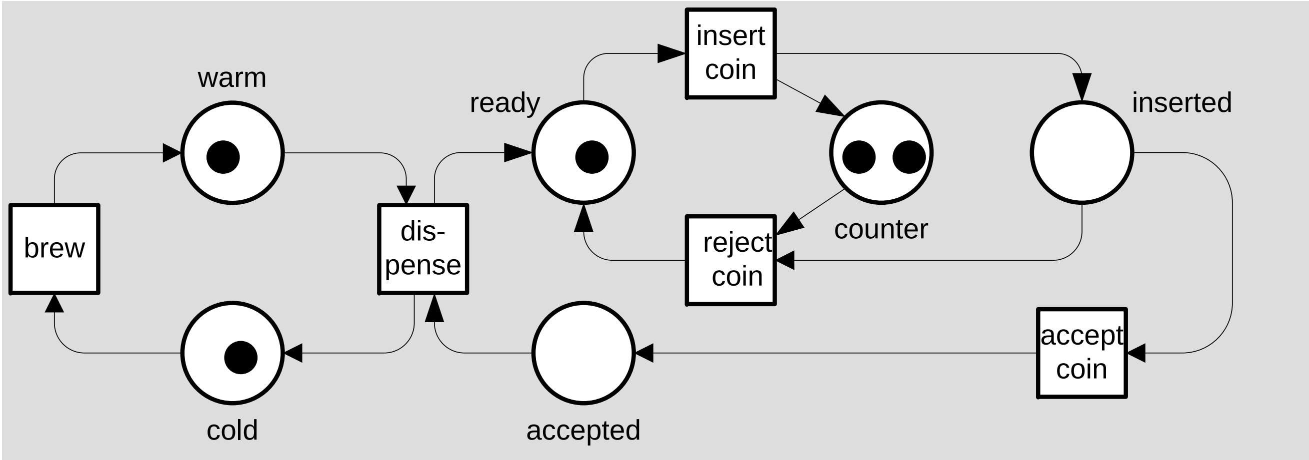


insert, accept, brew, dispense, brew, insert, accept, dispense

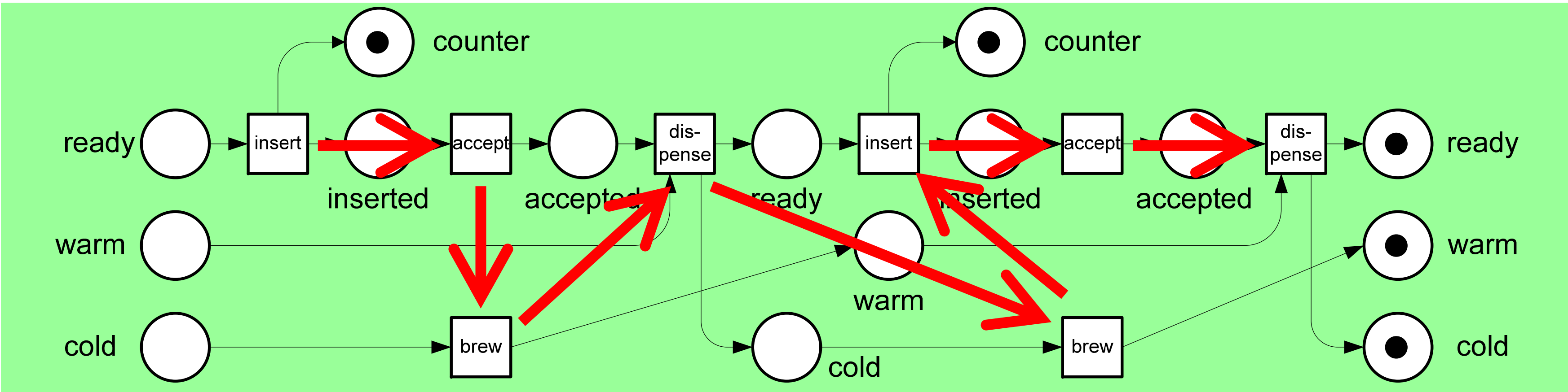


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

One common occurrence sequence



insert, accept, brew, dispense, brew, insert, accept, dispense



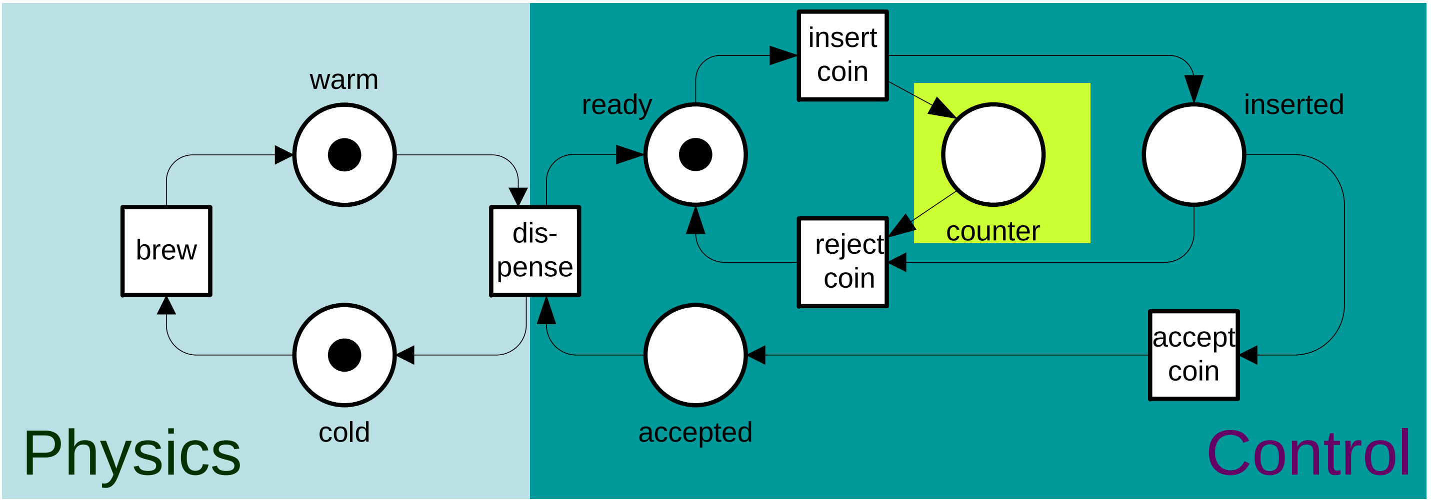
a linearization of the partial order of events

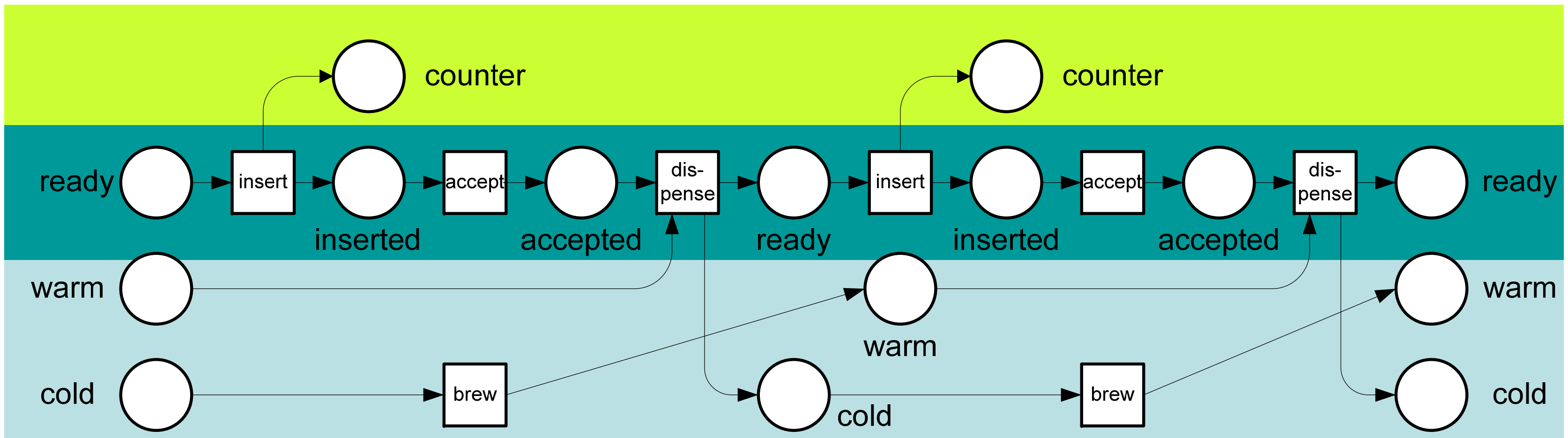
Advantage 2: Expressiveness

components of models, token flow etc.

can easily be identified in process nets

Advantage 2: Expressiveness



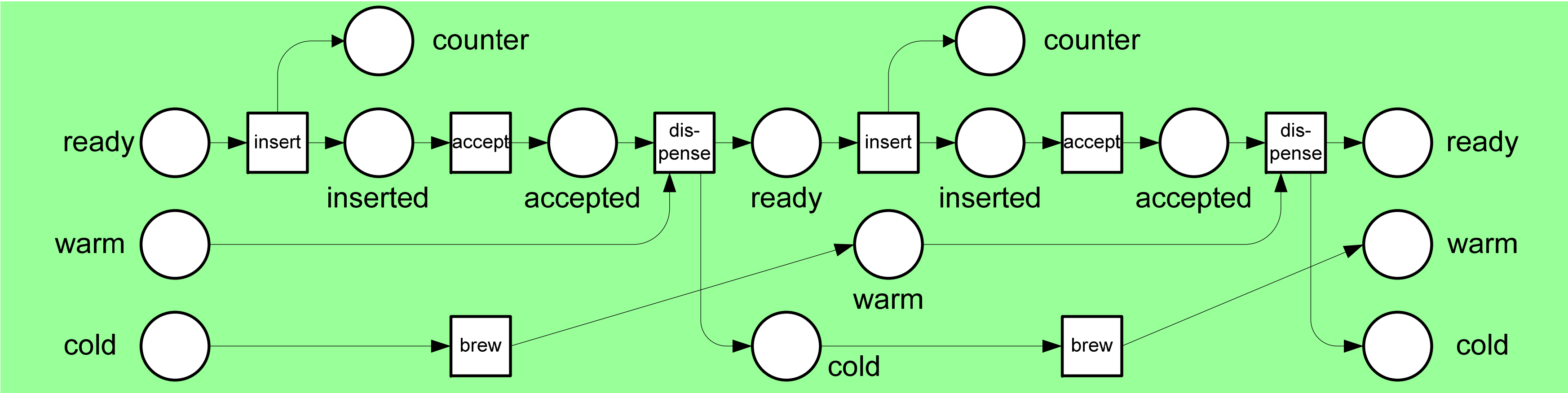


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

Advantage 3: Expressiveness (again)

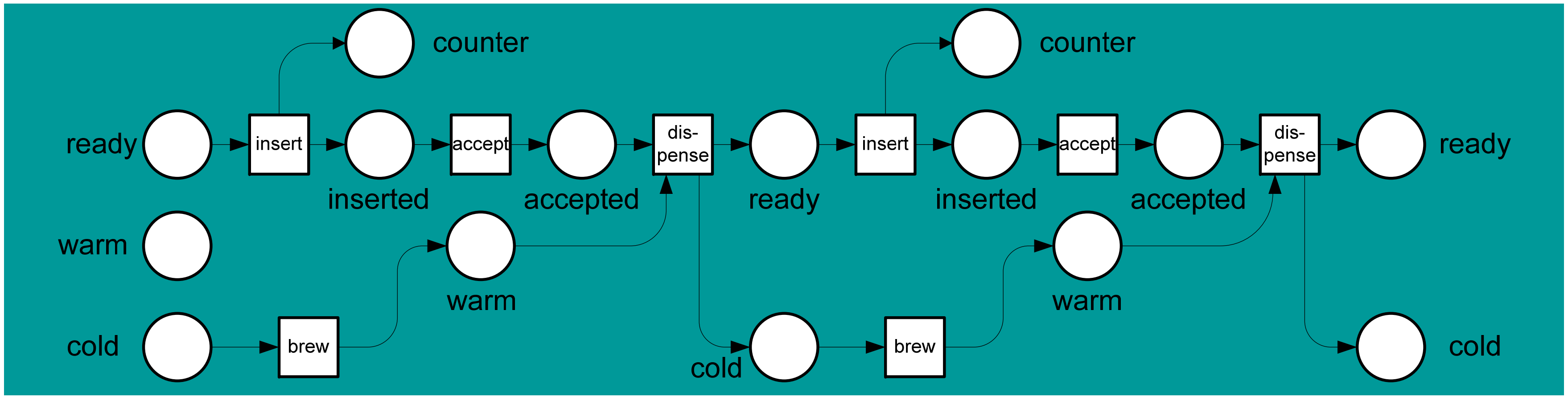
causal relationships are explicitly represented
(which is not the case for occurrence sequences
of non-safe place/transition nets)

Advantage 3: Expressiveness (again)



a common occurrence sequence of both process nets:

insert, accept, brew, dispense, brew, insert, accept, dispense



Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

Validating Requirements

What type of requirements?

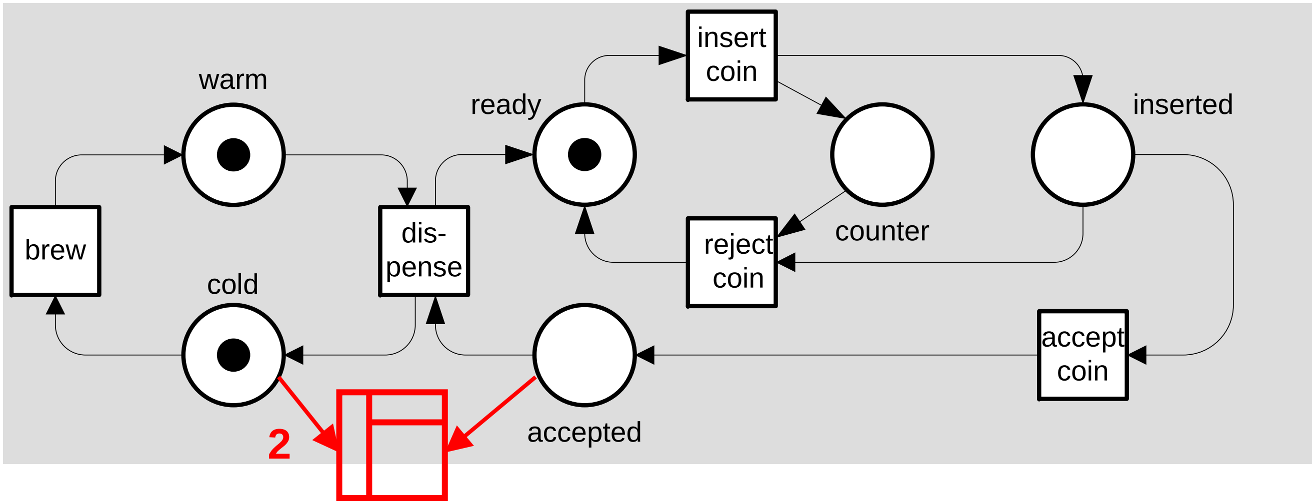
- linear-time properties that can be interpreted on single process nets

Two examples:

- facts representing invariant properties
- goals representing “leads-to”-properties

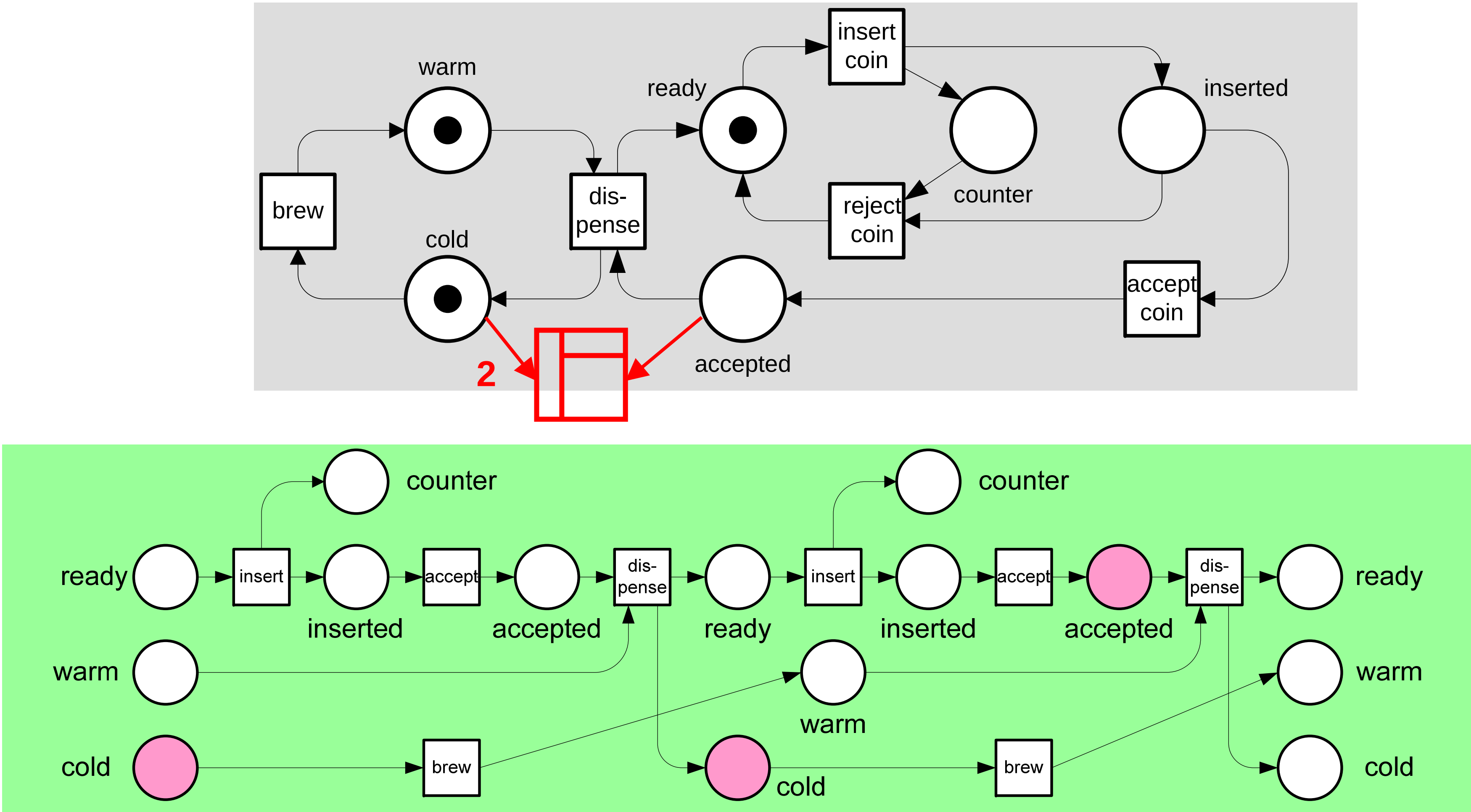
Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

A Fact



identify processes with a reachable marking
marking ***cold*** twice and marking ***accepted***

A Fact


warm
ready
insert
coin
inserted
brew
dis-
pense
counter
reject
coin
cold
accept
coin
2
accepted
counter
counter
ready
insert
accept
dis-
pense
insert
accept
dis-
pense
ready
inserted
accepted
ready
inserted
accepted
warm
warm
warm
cold
brew
cold
brew
cold

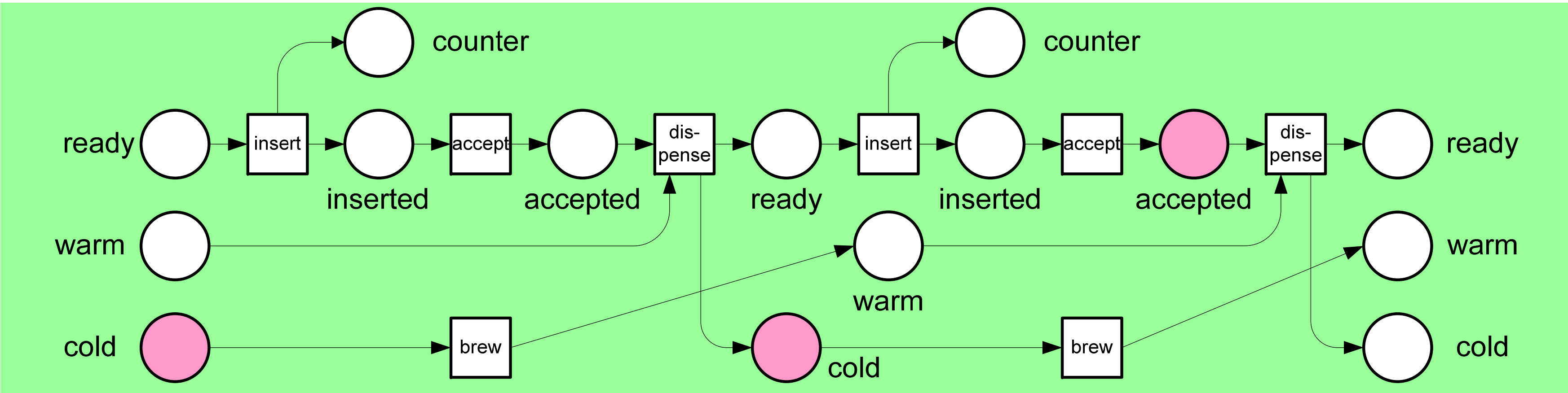
A Fact

A co-set of appropriately marked places of the process…

... is included in a cut

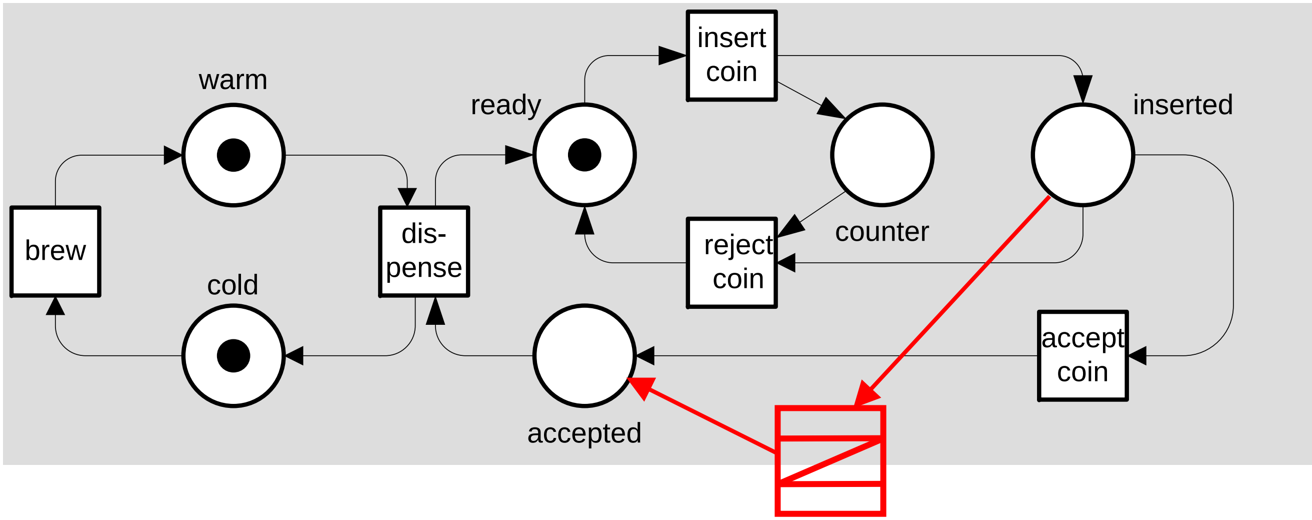
… which corresponds to a reachable marking of the process

… which corresponds to a reachable marking of the net.



Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

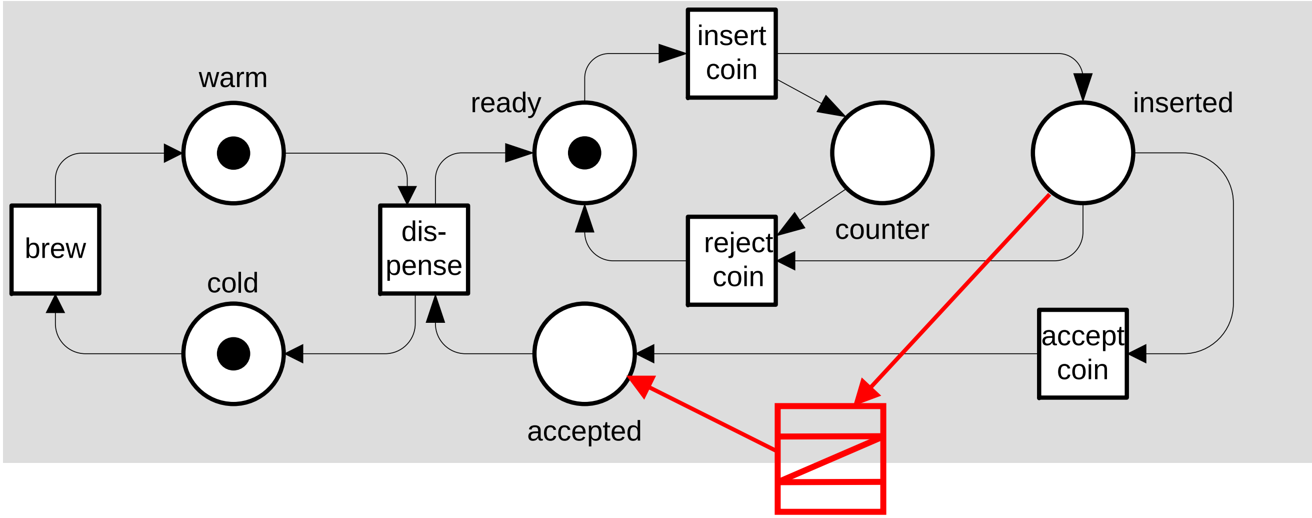
A Goal

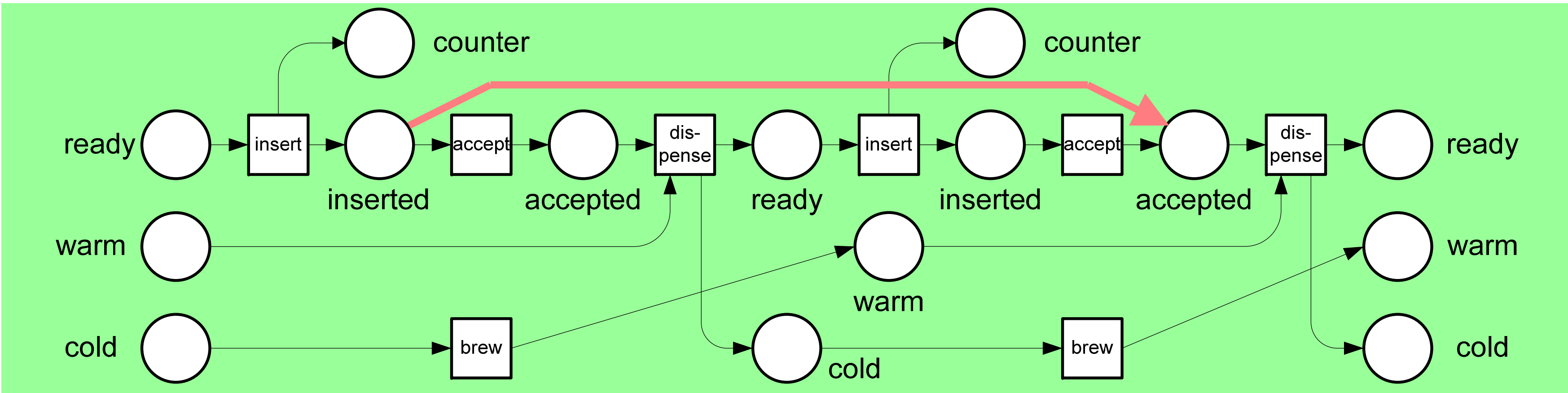


identify processes such that:

If there is a token on *inserted* then eventually there will be a token on *accepted*

A Goal





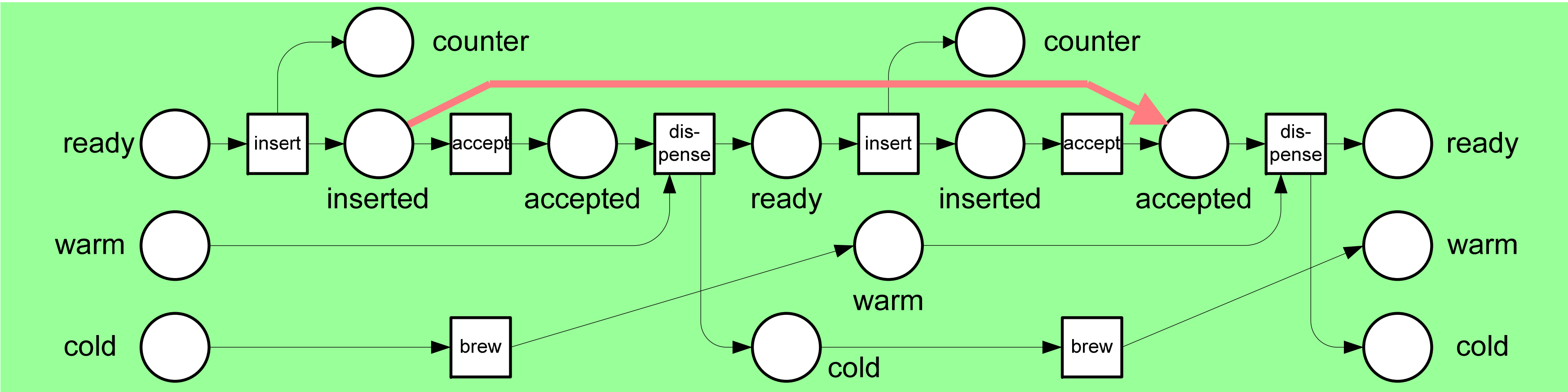
Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

A Goal

accepted causally depends on ***inserted***

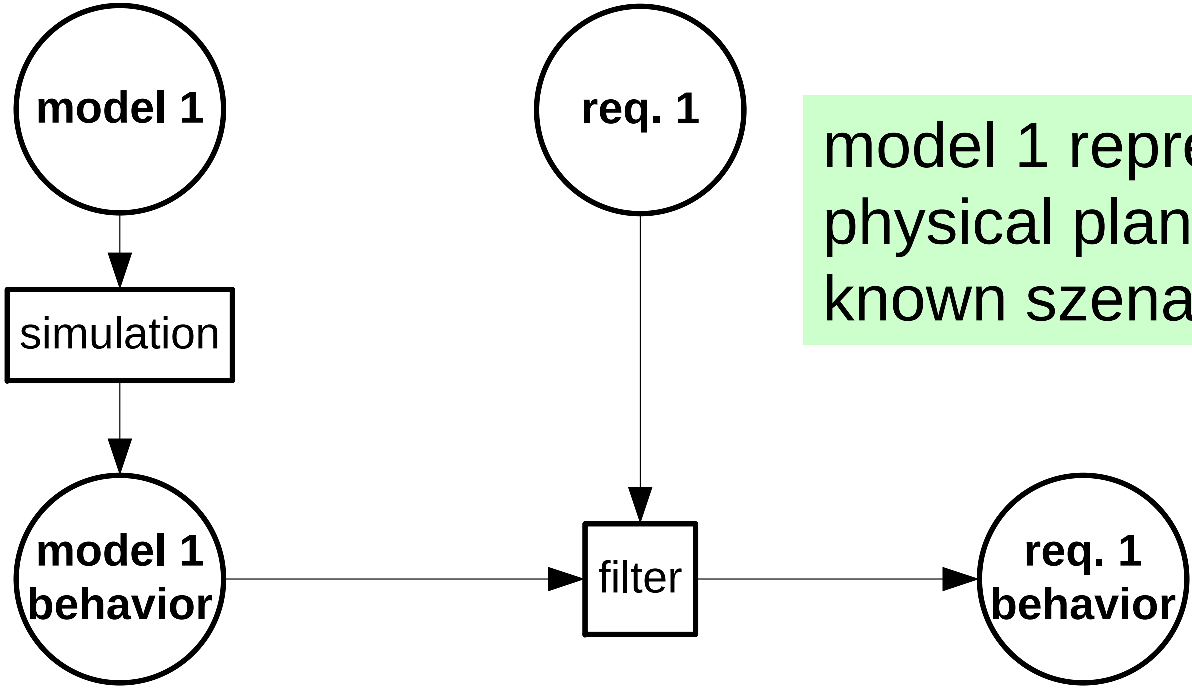
… hence from each marking of the process
that marks the condition ***inserted***
a marking will be reached that marks condition ***accepted.***

So for this run, the place
accepted will eventually be marked after ***inserted.***



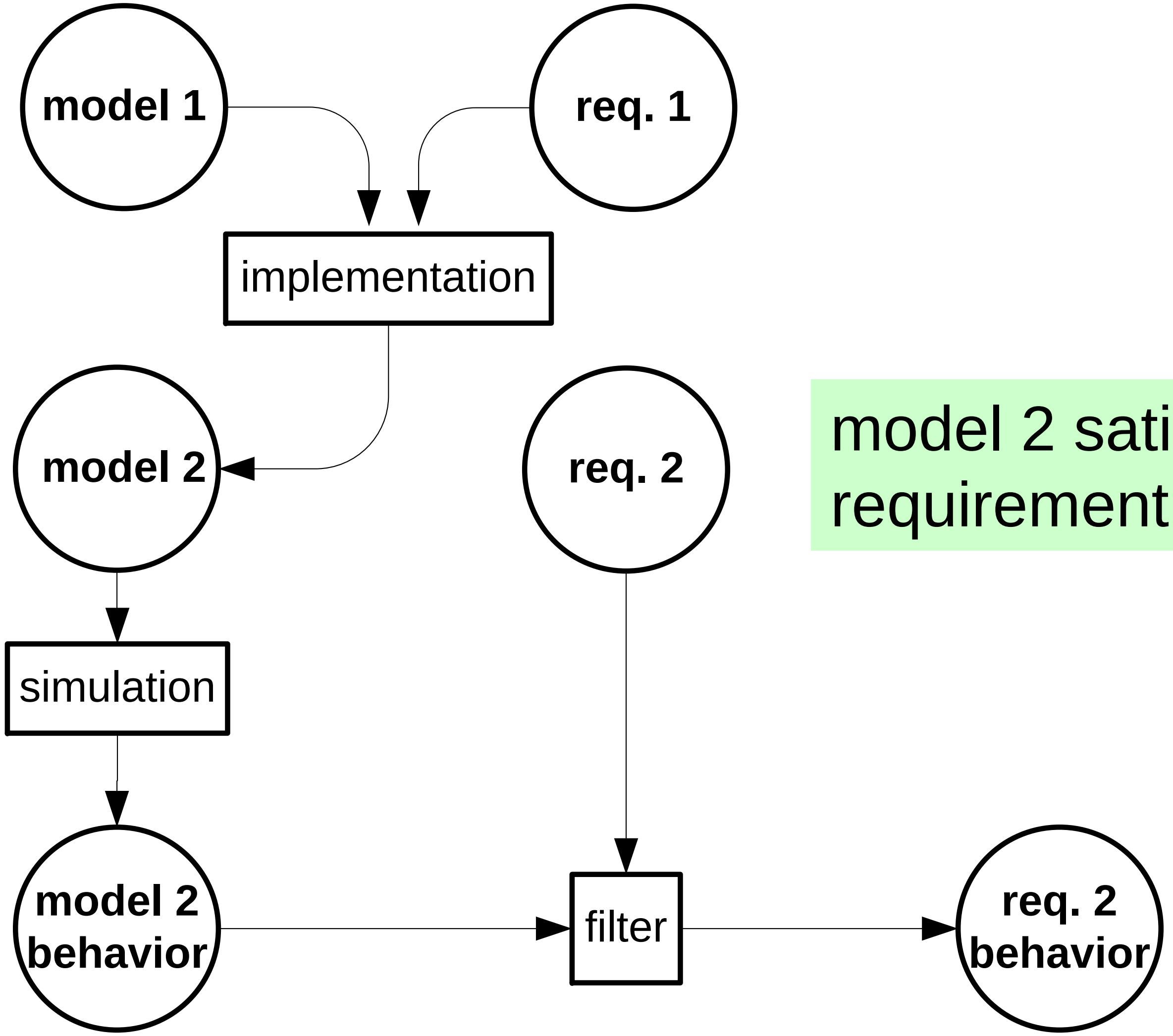
Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

Stepwise Validation of Requirements



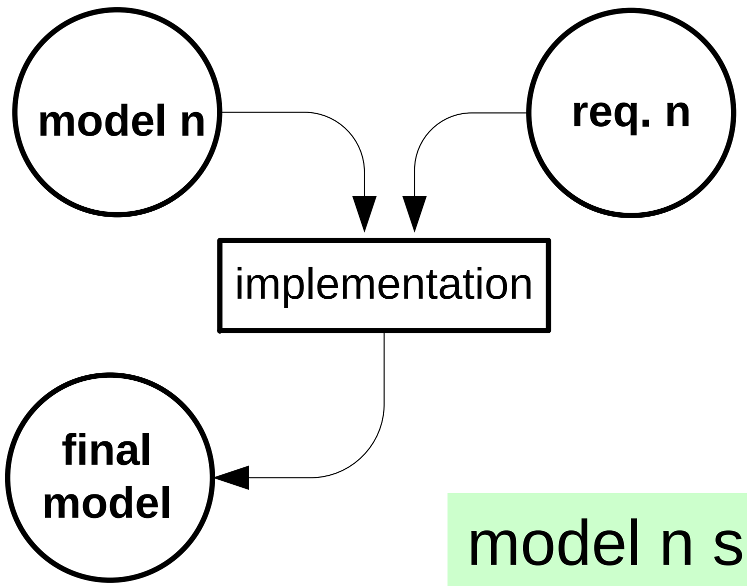
model 1 represents the physical plant,
known szenarios etc.

Stepwise Validation of Requirements



model 2 satisfies requirement 1
requirement 2 is validated

Stepwise Validation of Requirements

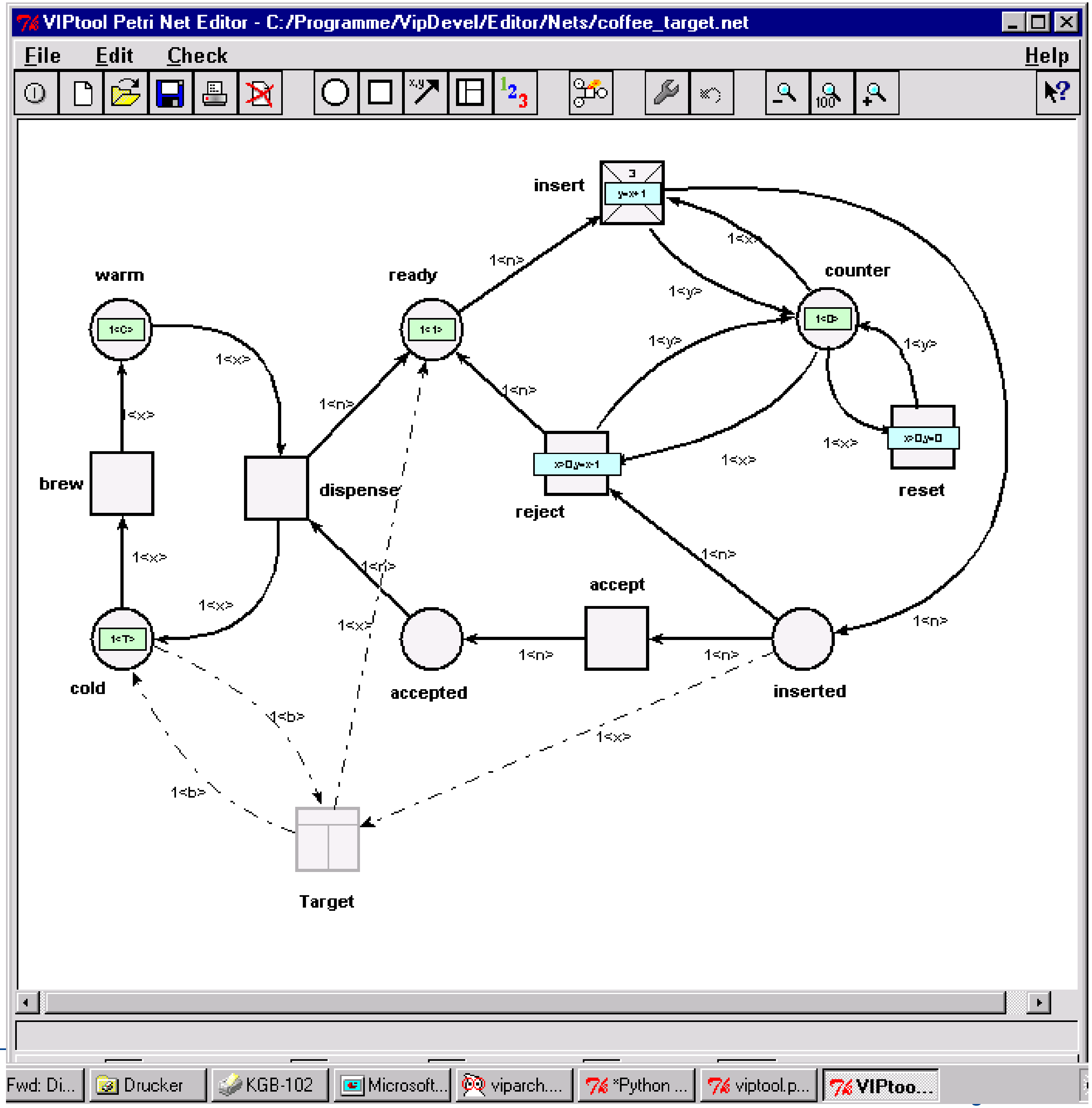


model n satisfies requirements 1, ... , n-1
final model satisfies requirements 1, ... , n

This approach only works in general if all requirements restrict behavior

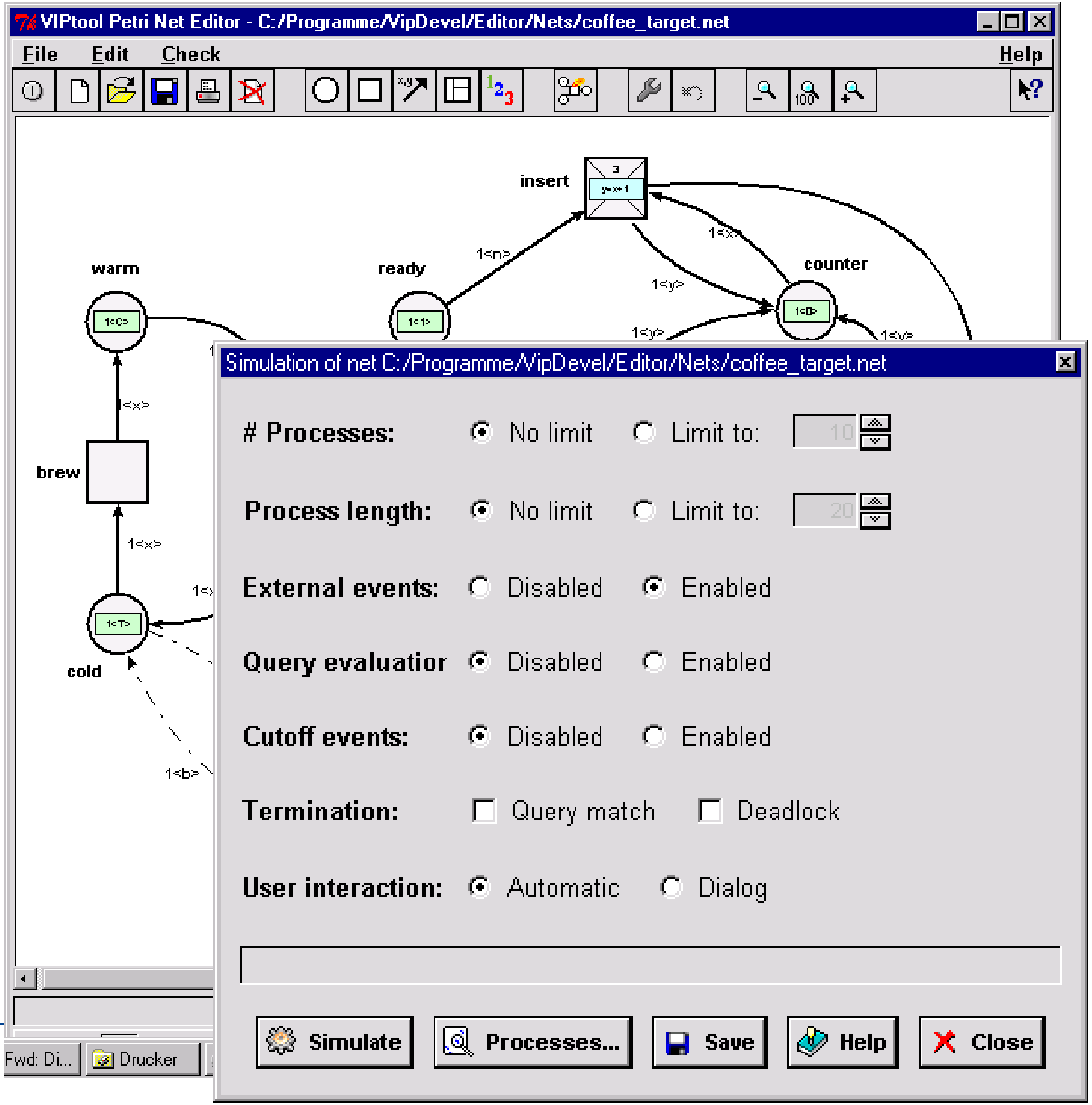
VipTool

simulation
generates
causal nets



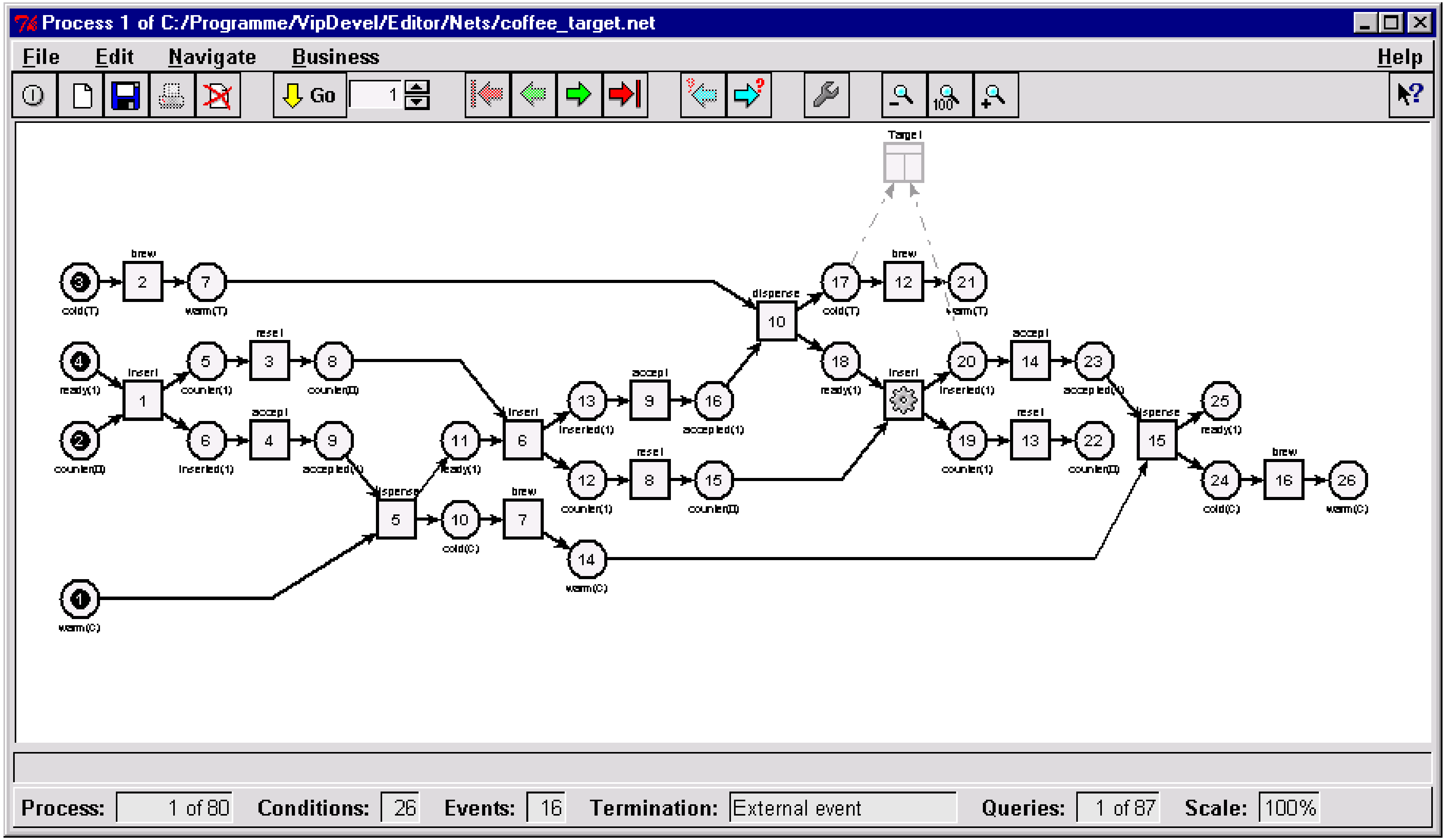
VipTool

control of the simulation



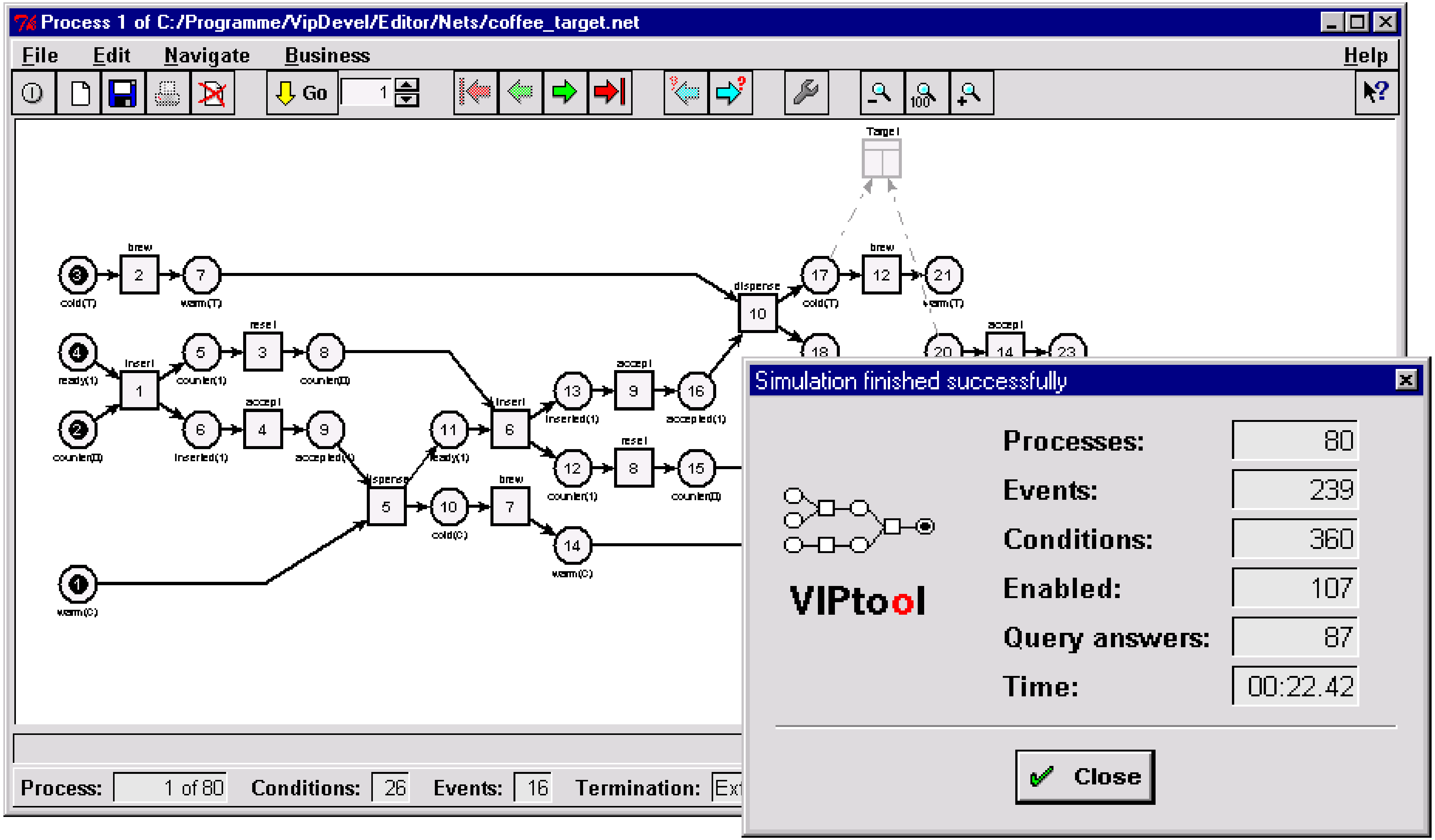
VipTool

a generated and visualized run



VipTool

analysis of runs



Synthesis

means generation of nets from runs.

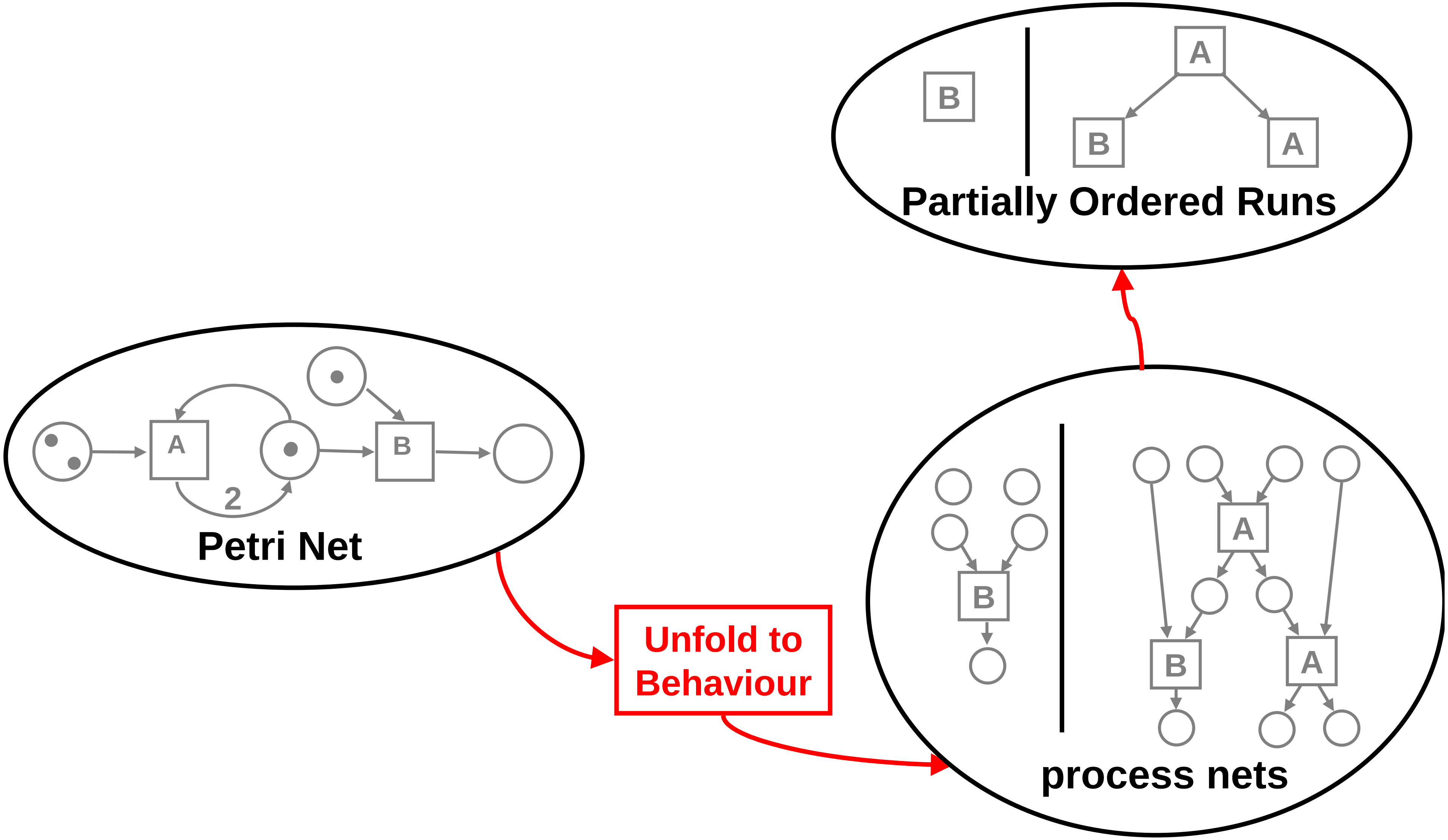
from

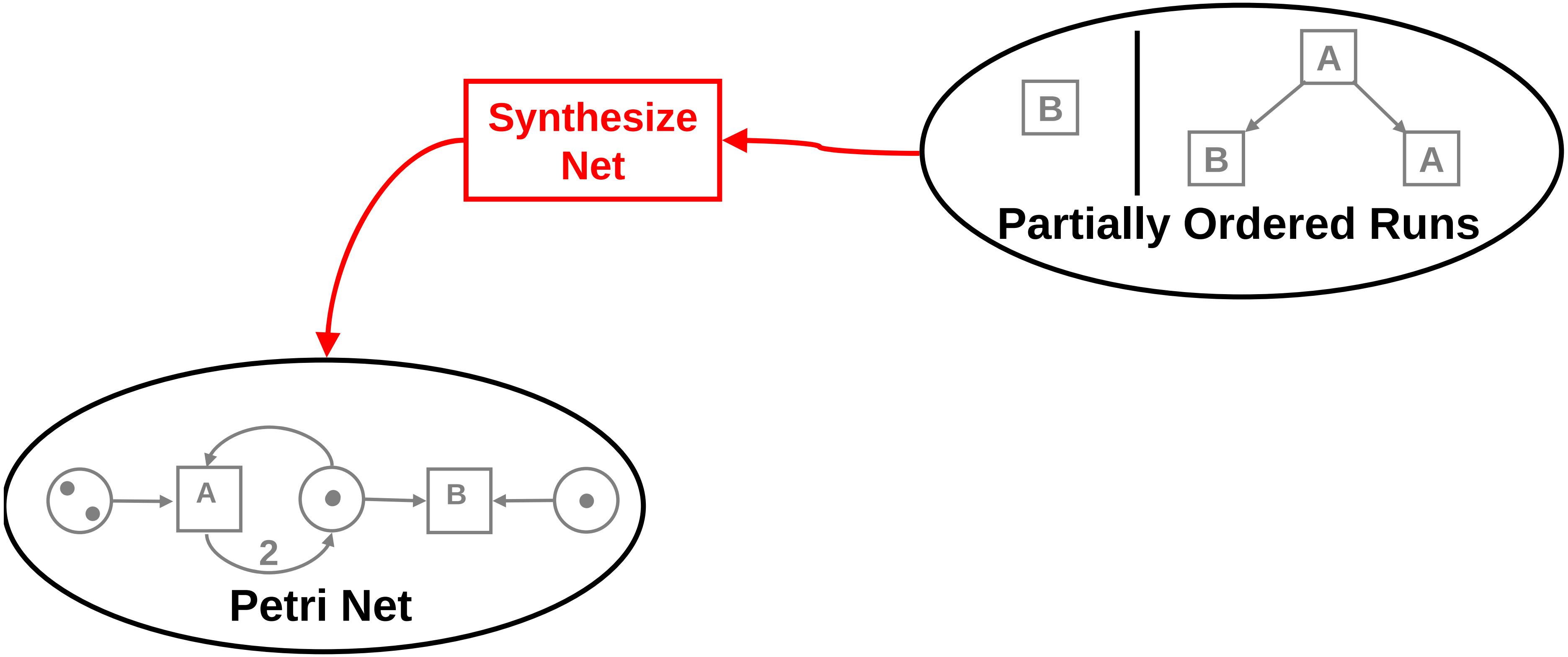
- **sequential runs** (occurrence sequences),

  $\Rightarrow$ well-known region theory

- **non-sequential, partially ordered runs**

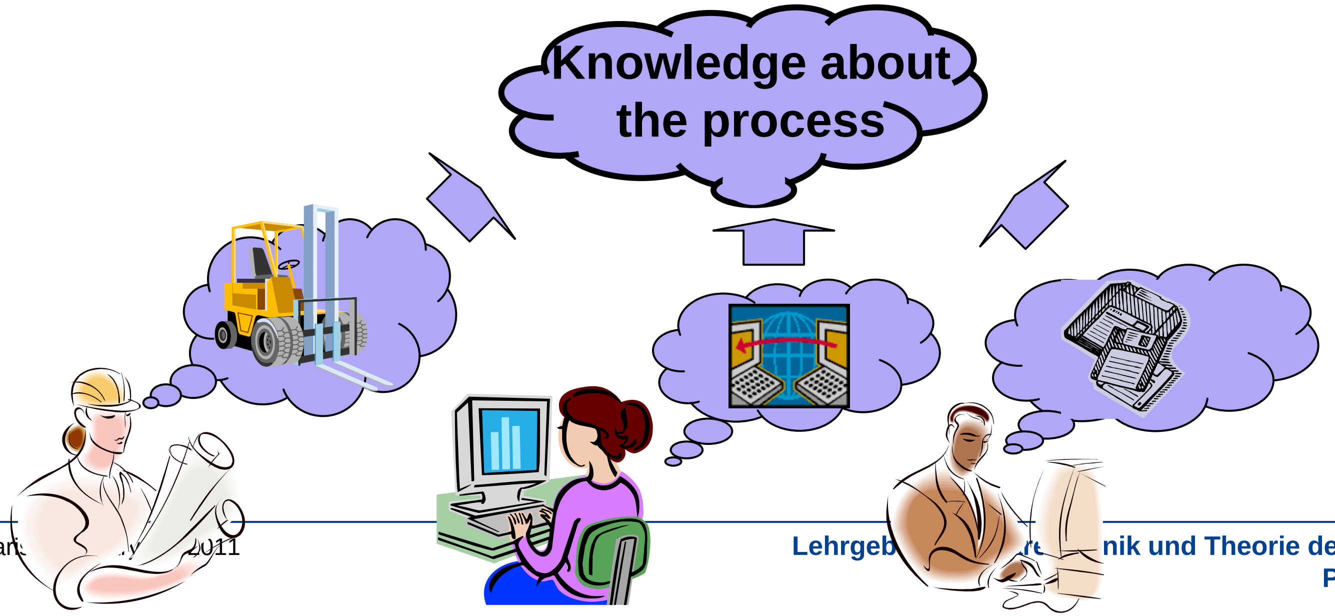
  $\Rightarrow$ the **<u>VIP-approach</u>**

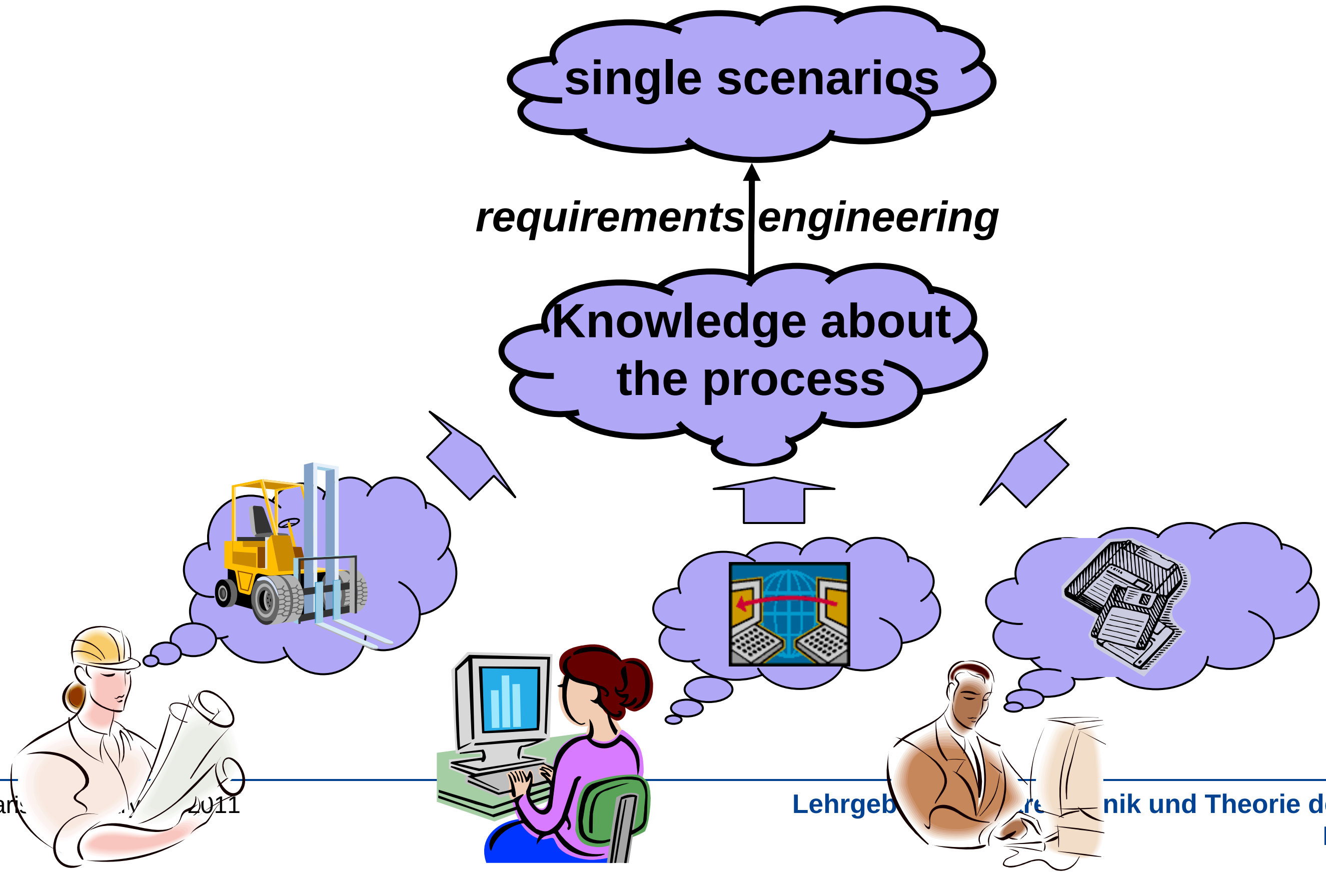

B
A
B
A
Partially Ordered Runs
A
2
B
Petri Net
Unfold to
Behaviour
B
A
B
A
process nets

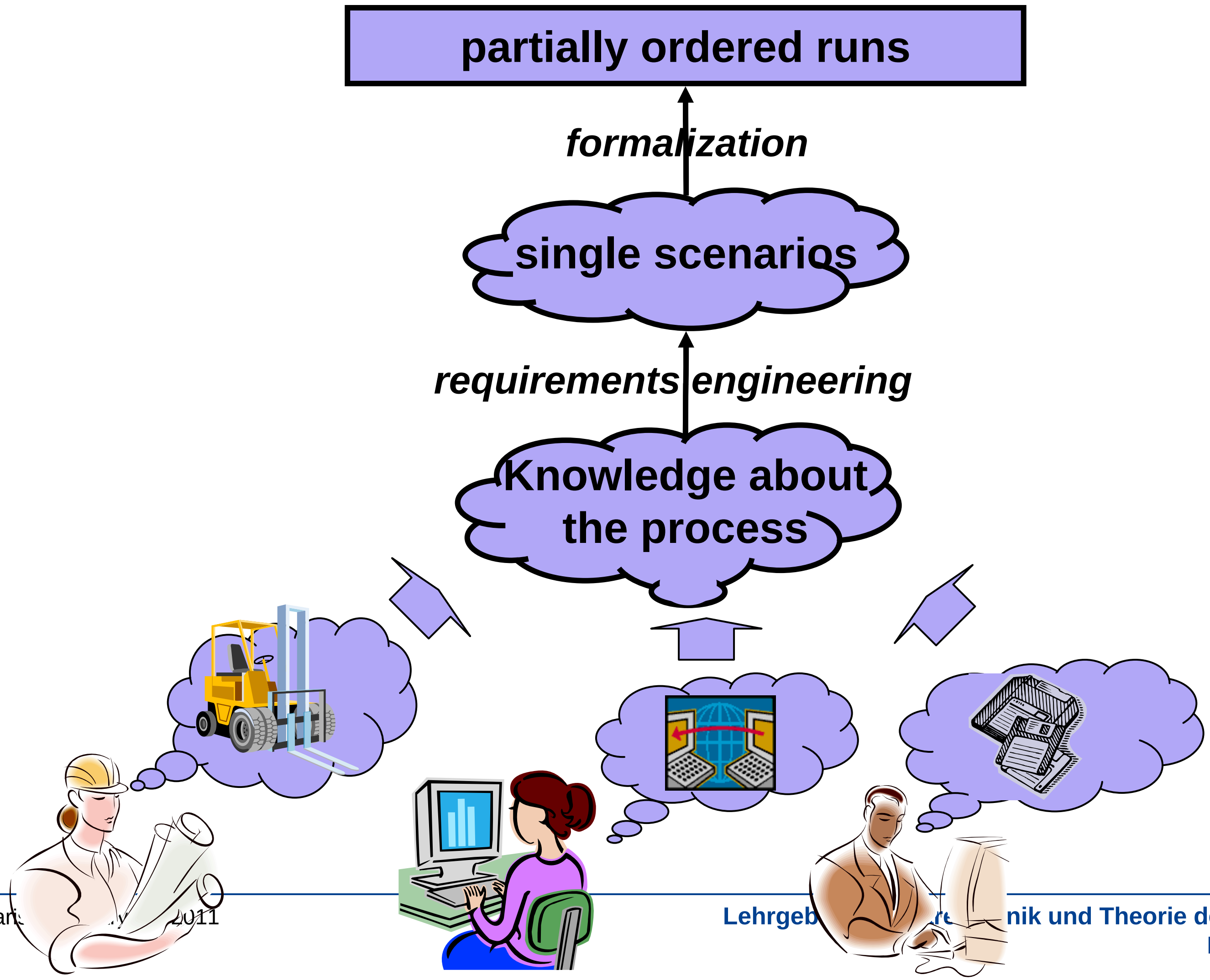

Synthesize
Net
B
A
B
A
Partially Ordered Runs
A
2
B
Petri Net

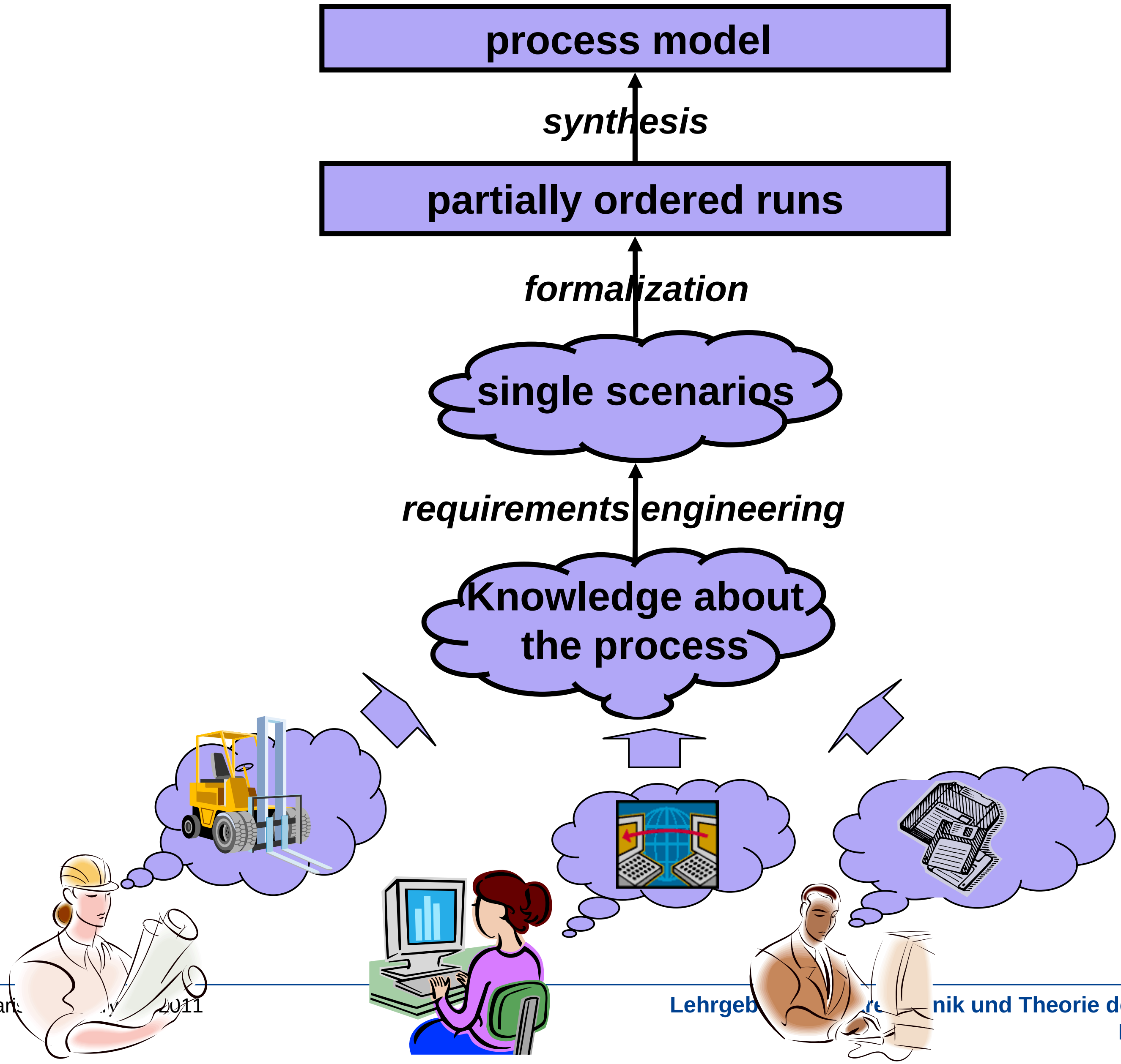
Initial Situation:

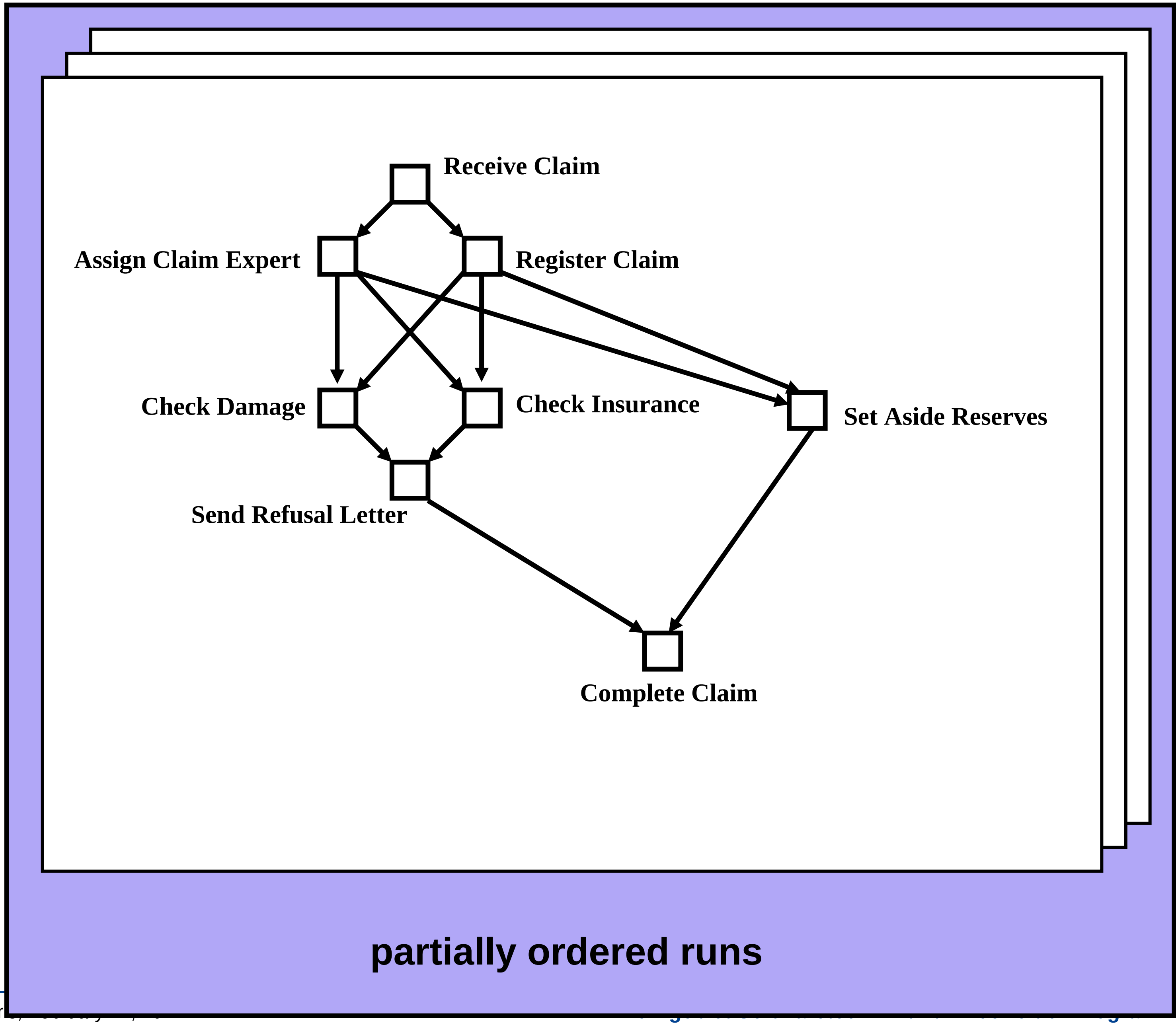
Knowledge about a process is distributed in several peoples‘ mind in an informal environment

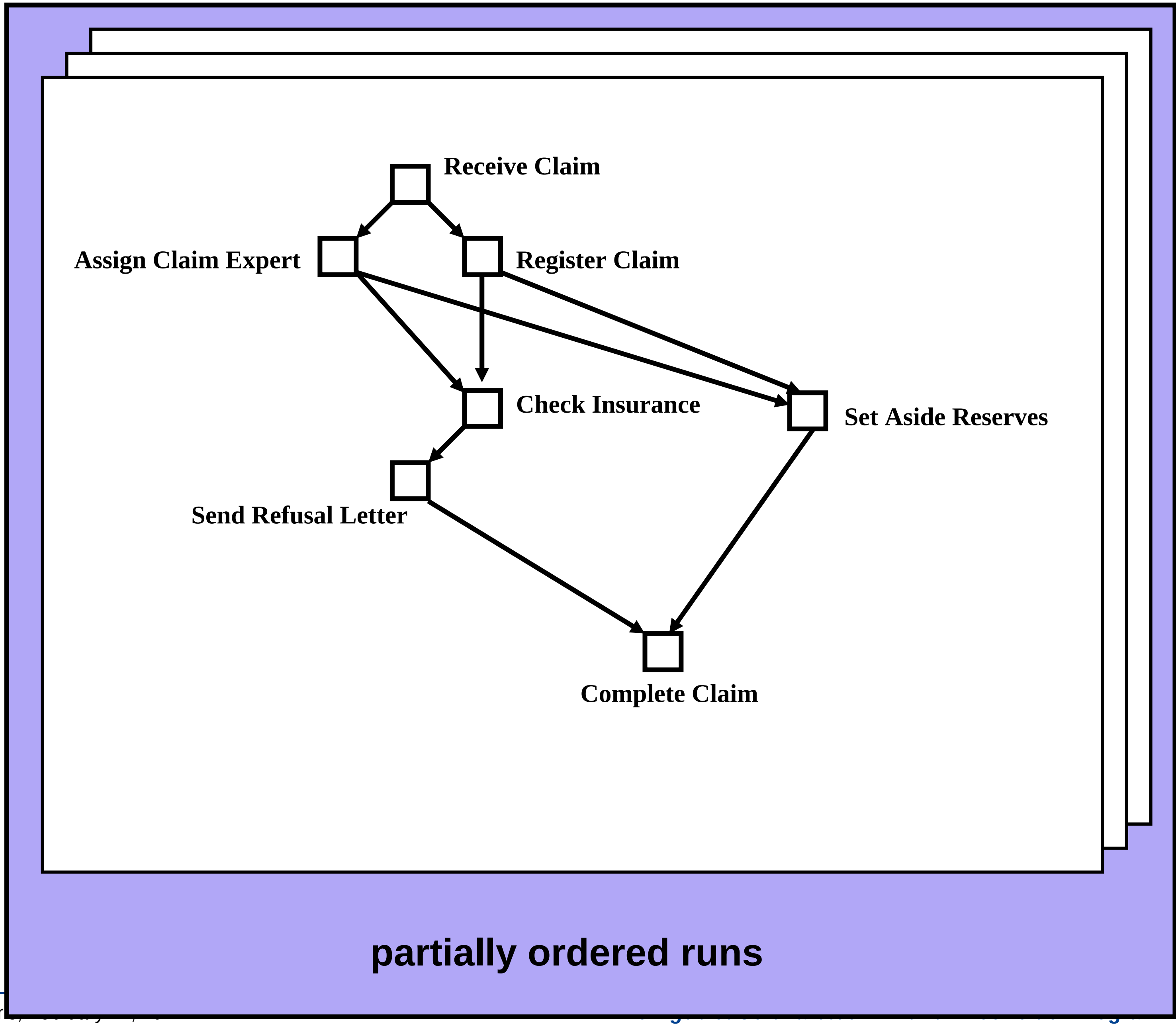


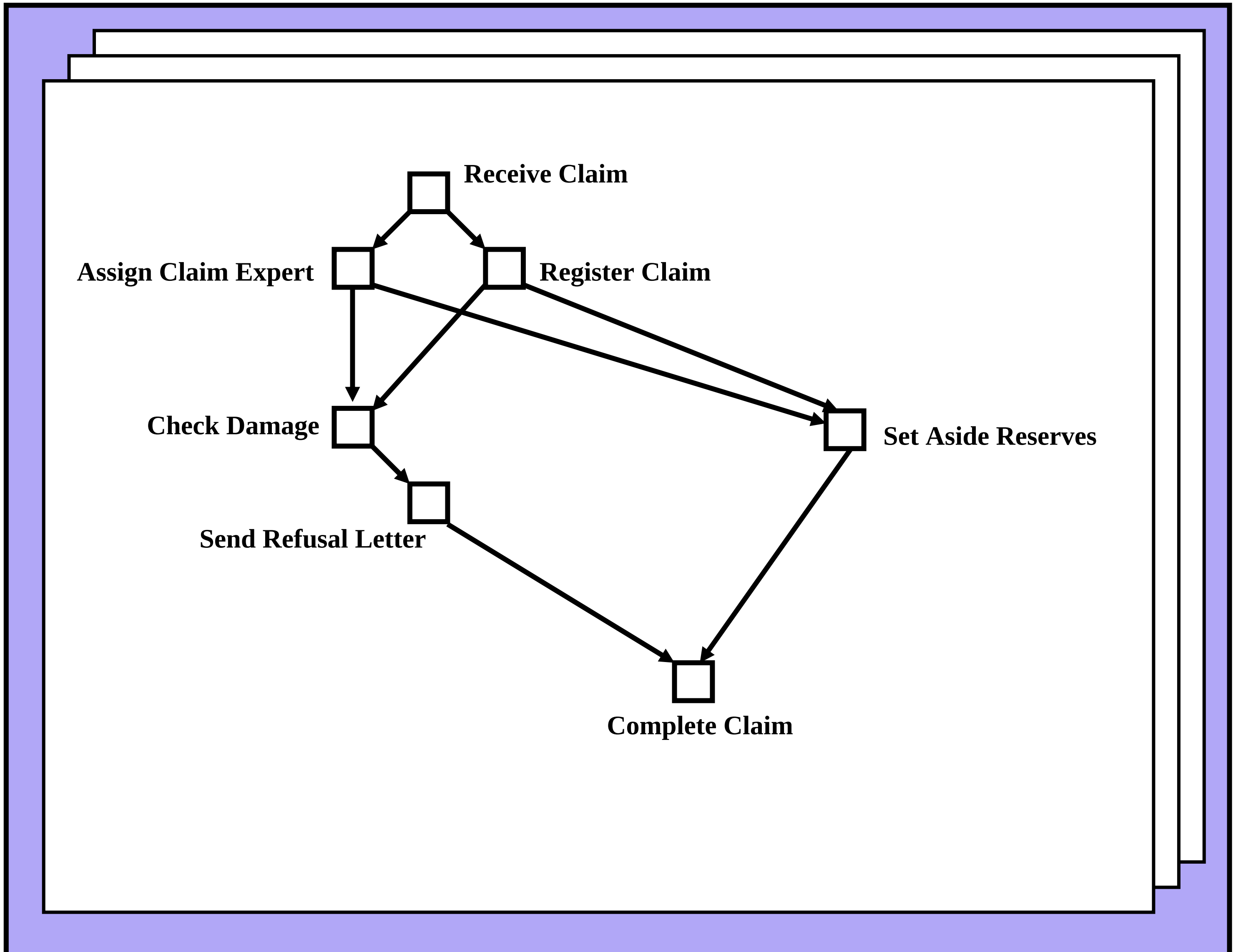

single scenarios
requirements engineering
Knowledge about the process


partially ordered runs
formalization
single scenarios
requirements engineering
Knowledge about the process

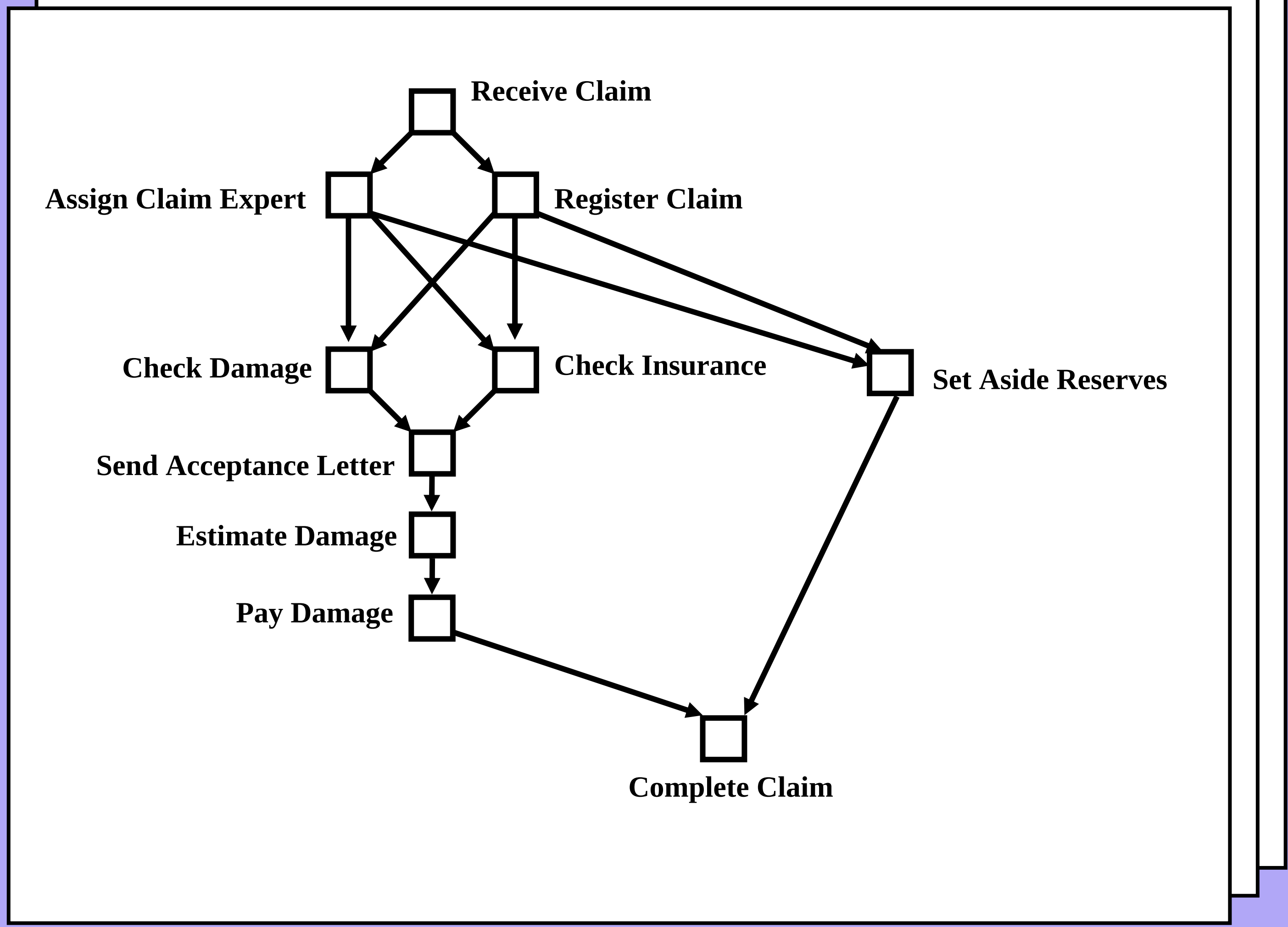

process model
synthesis
partially ordered runs
formalization
single scenarios
requirements engineering
Knowledge about the process


Receive Claim
Assign Claim Expert
Register Claim
Check Damage
Check Insurance
Set Aside Reserves
Send Refusal Letter
Complete Claim
partially ordered runs

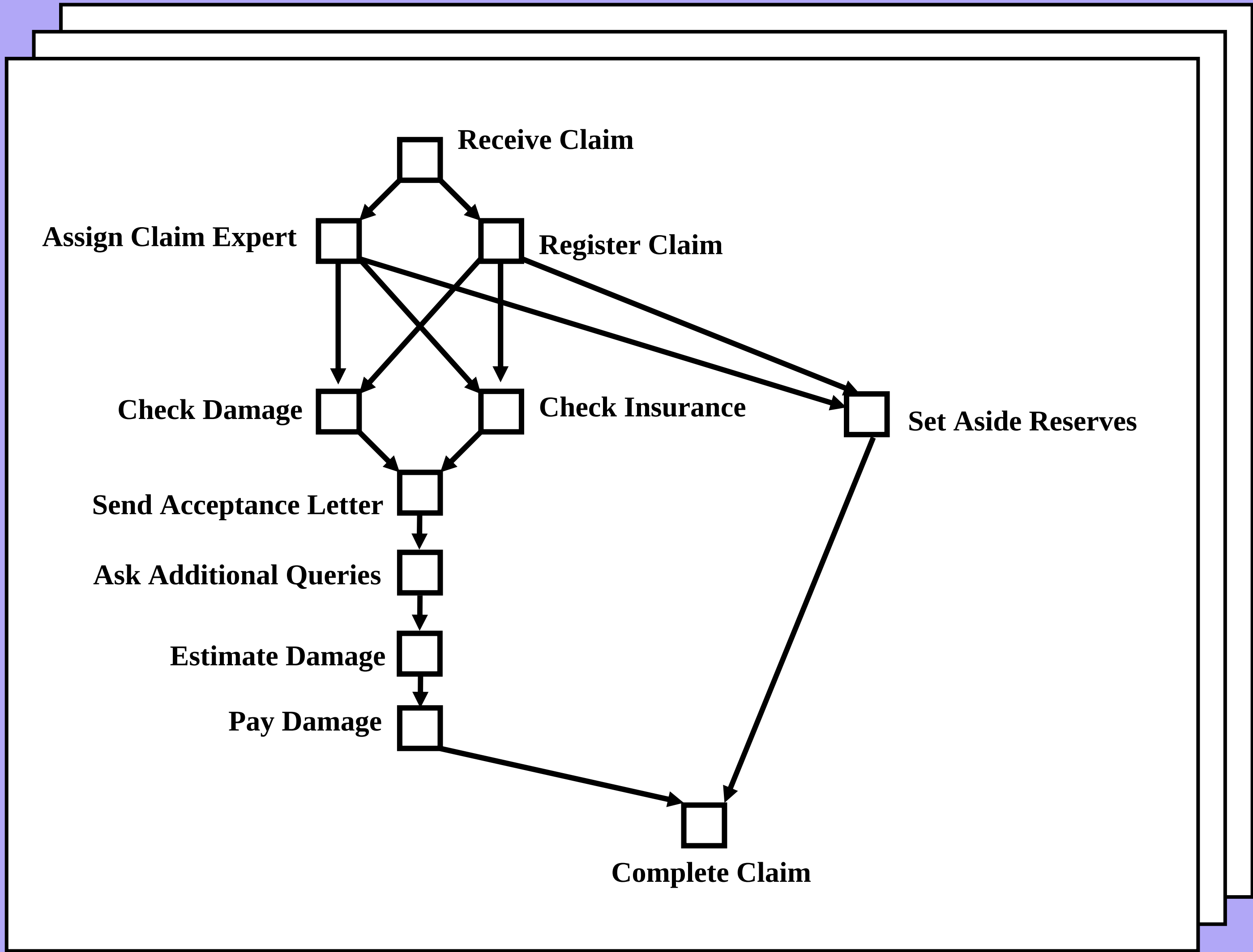

Receive Claim
Assign Claim Expert
Register Claim
Check Insurance
Set Aside Reserves
Send Refusal Letter
Complete Claim
partially ordered runs



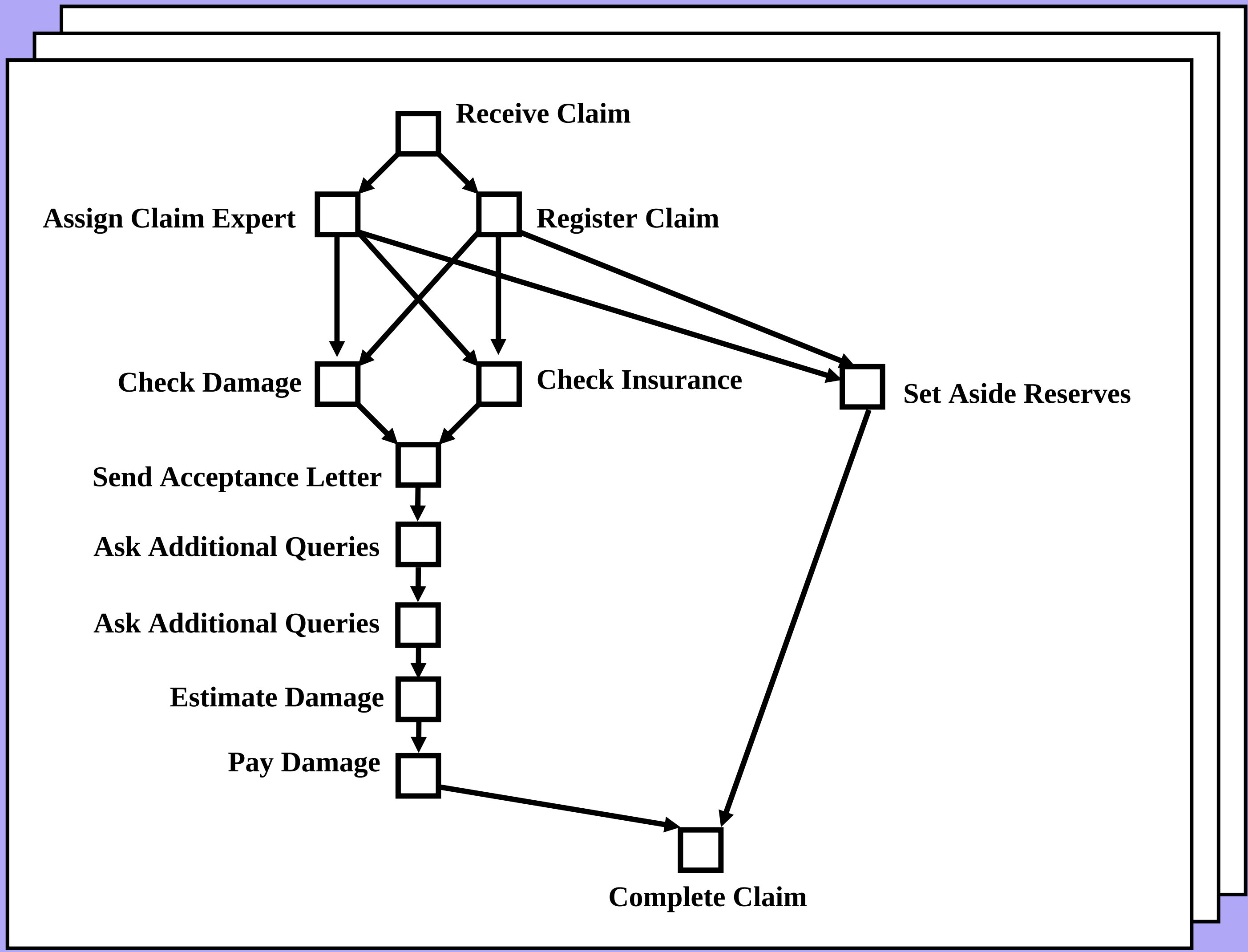
partially ordered runs



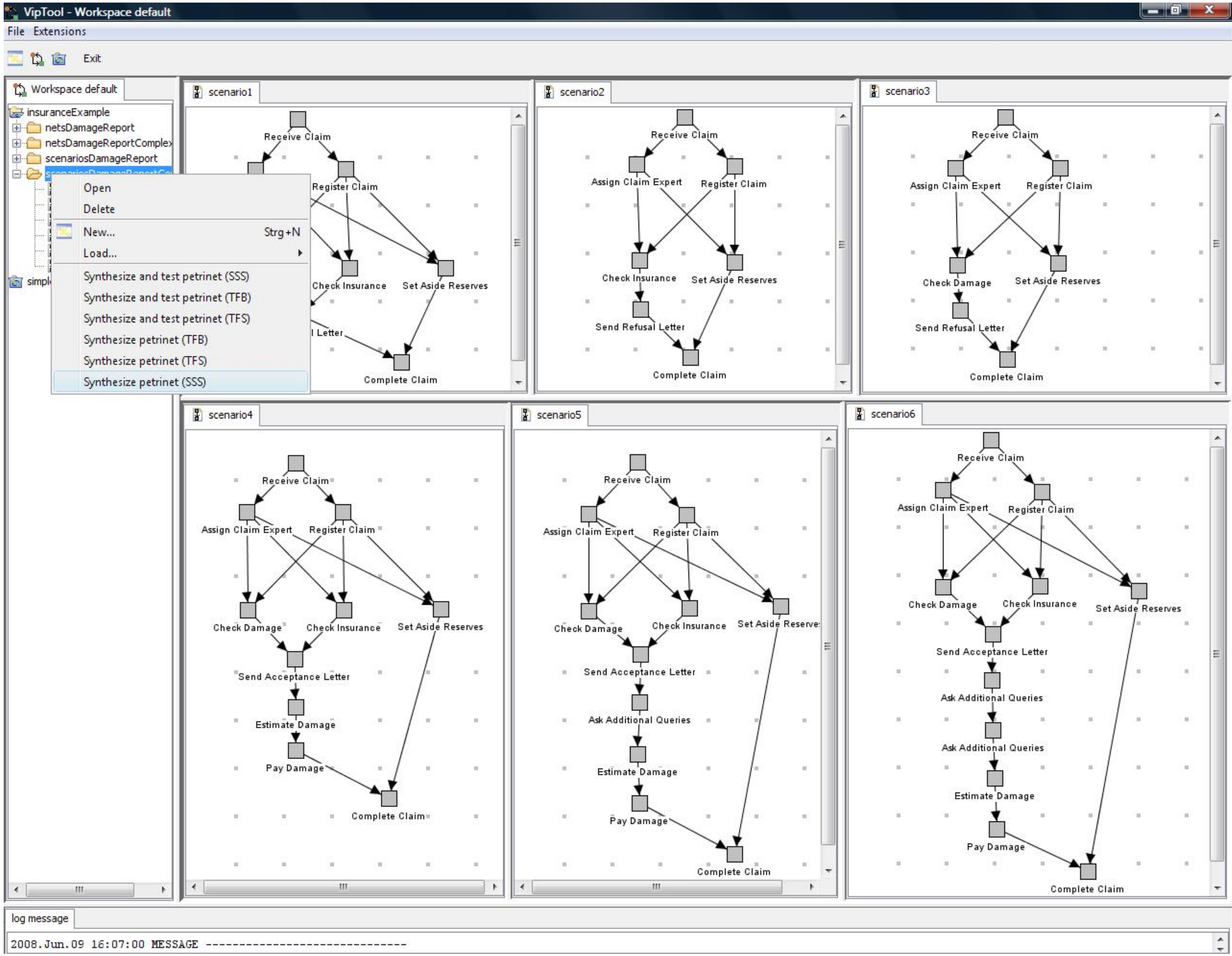
partially ordered runs

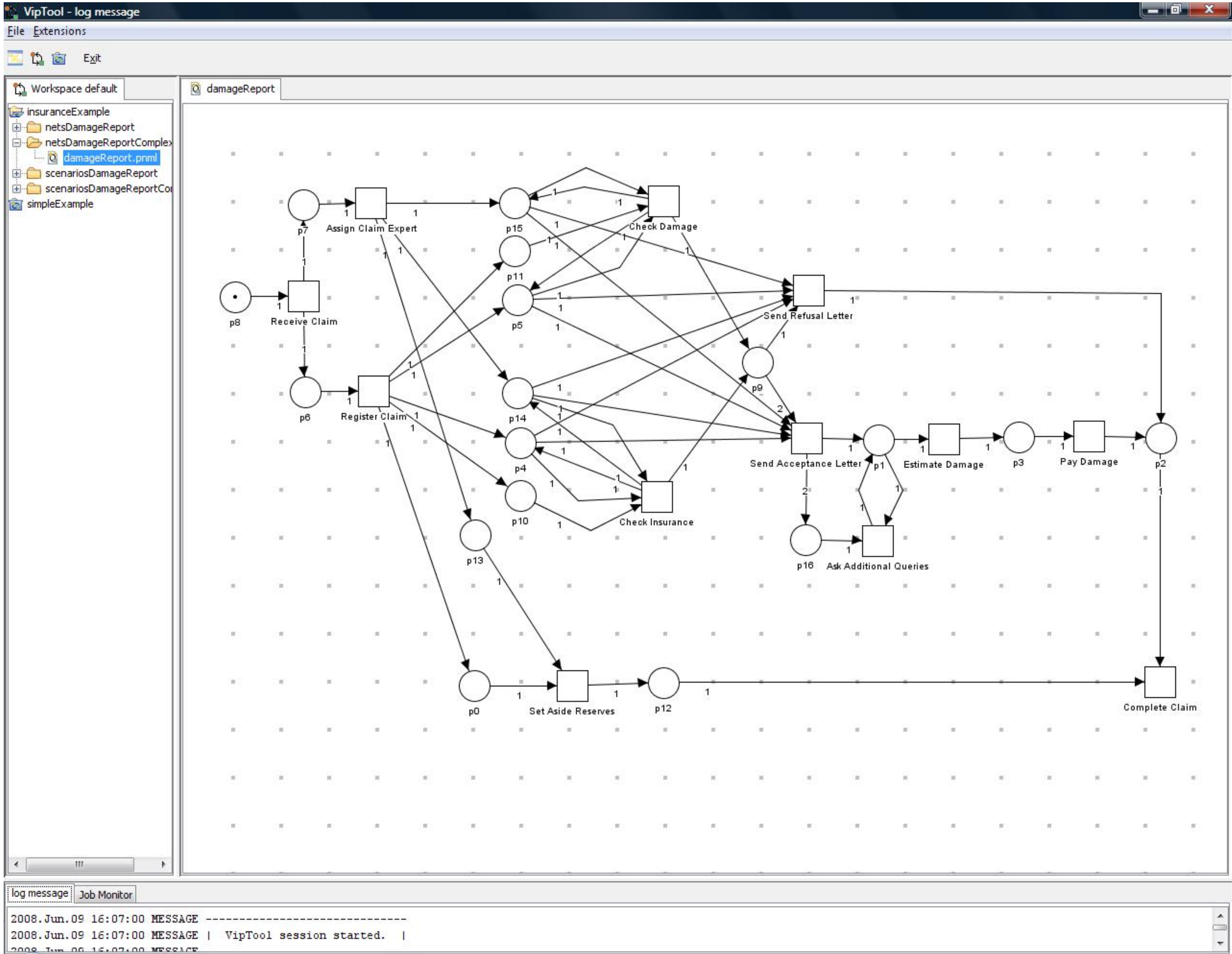


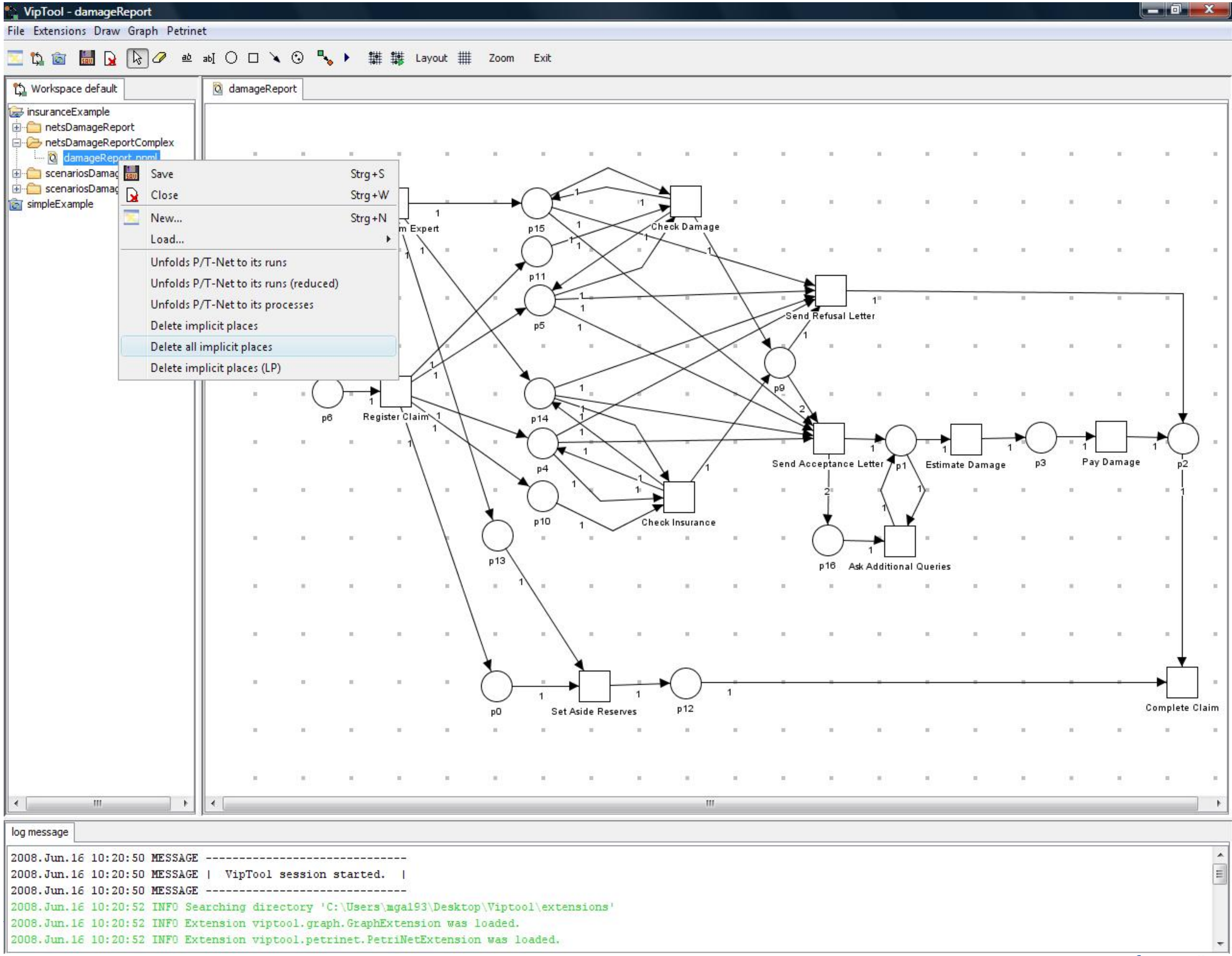
partially ordered runs

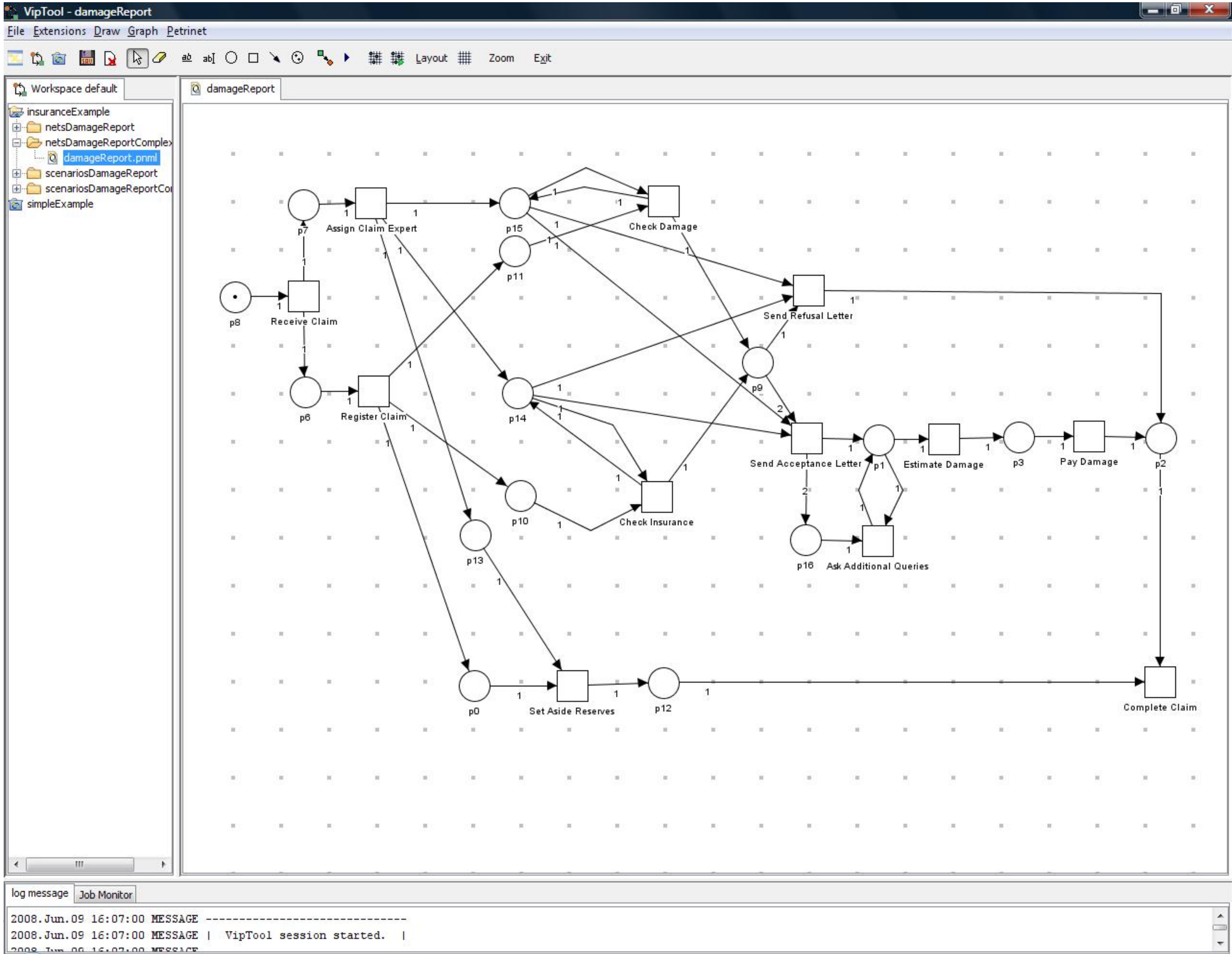


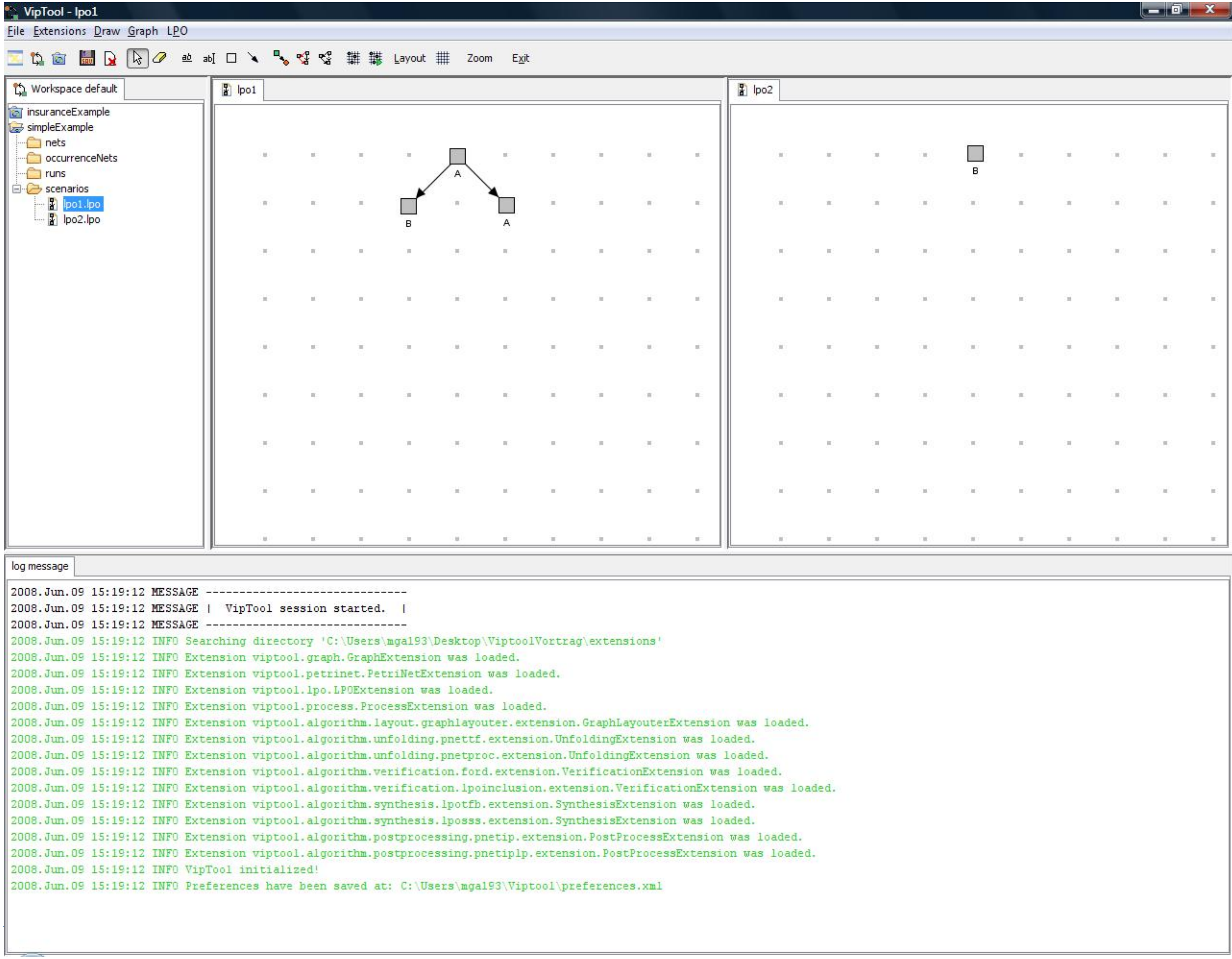
partially ordered runs

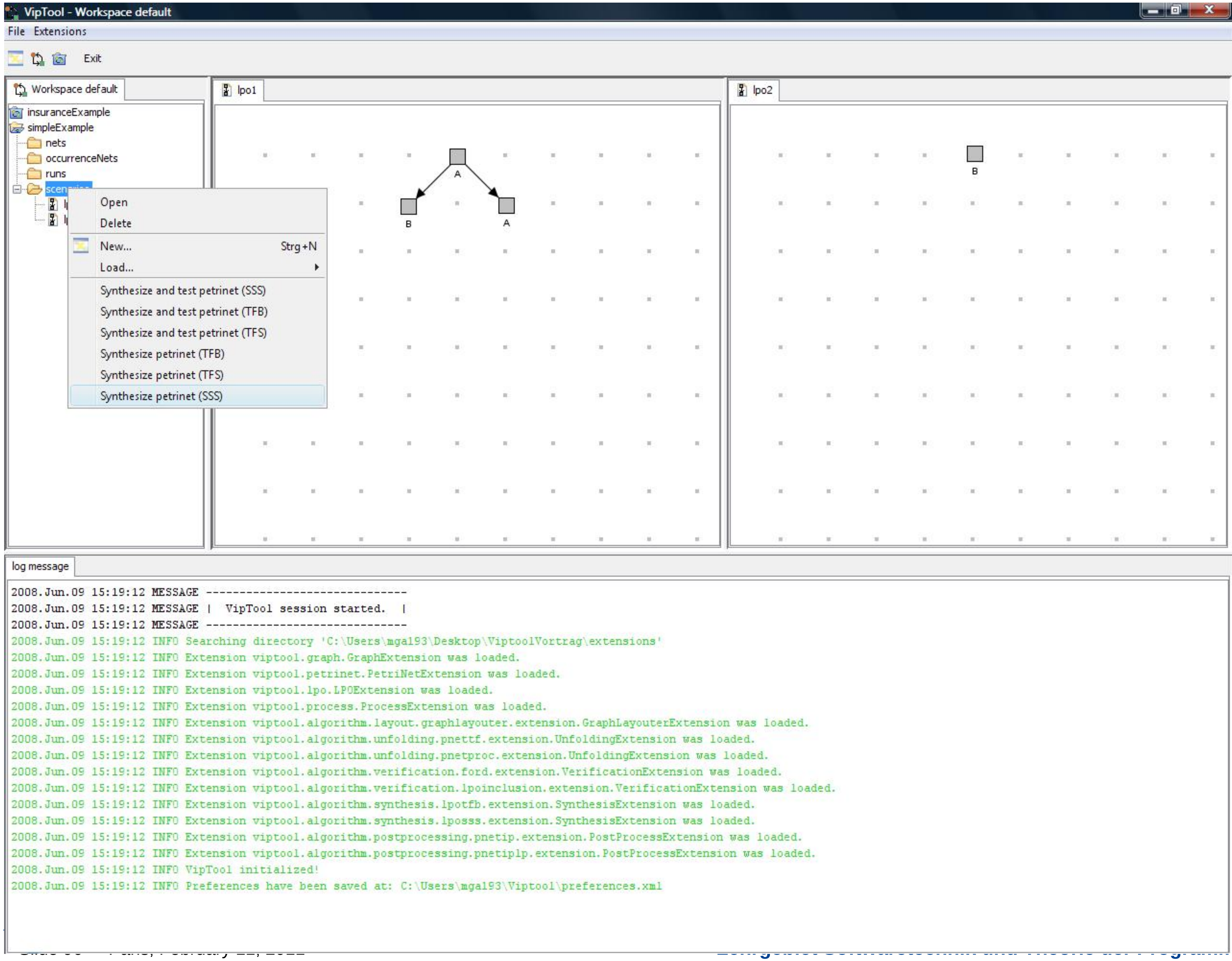

VipTool - Workspace default
File Extensions
Exit
Workspace default
insuranceExample
netsDamageReport
netsDamageReportComplex
scenariosDamageReport
Open
Delete
New... Strg+N
Load...
Synthesize and test petrinet (SSS)
Synthesize and test petrinet (TFB)
Synthesize and test petrinet (TFS)
Synthesize petrinet (TFB)
Synthesize petrinet (TFS)
Synthesize petrinet (SSS)
scenario1
scenario2
scenario3
scenario4
scenario5
scenario6
Receive Claim
Assign Claim Expert
Register Claim
Check Insurance
Check Damage
Set Aside Reserves
Send Refusal Letter
Send Acceptance Letter
Ask Additional Queries
Estimate Damage
Pay Damage
Complete Claim
log message
2008.Jun.09 16:07:00 MESSAGE ------------------------------

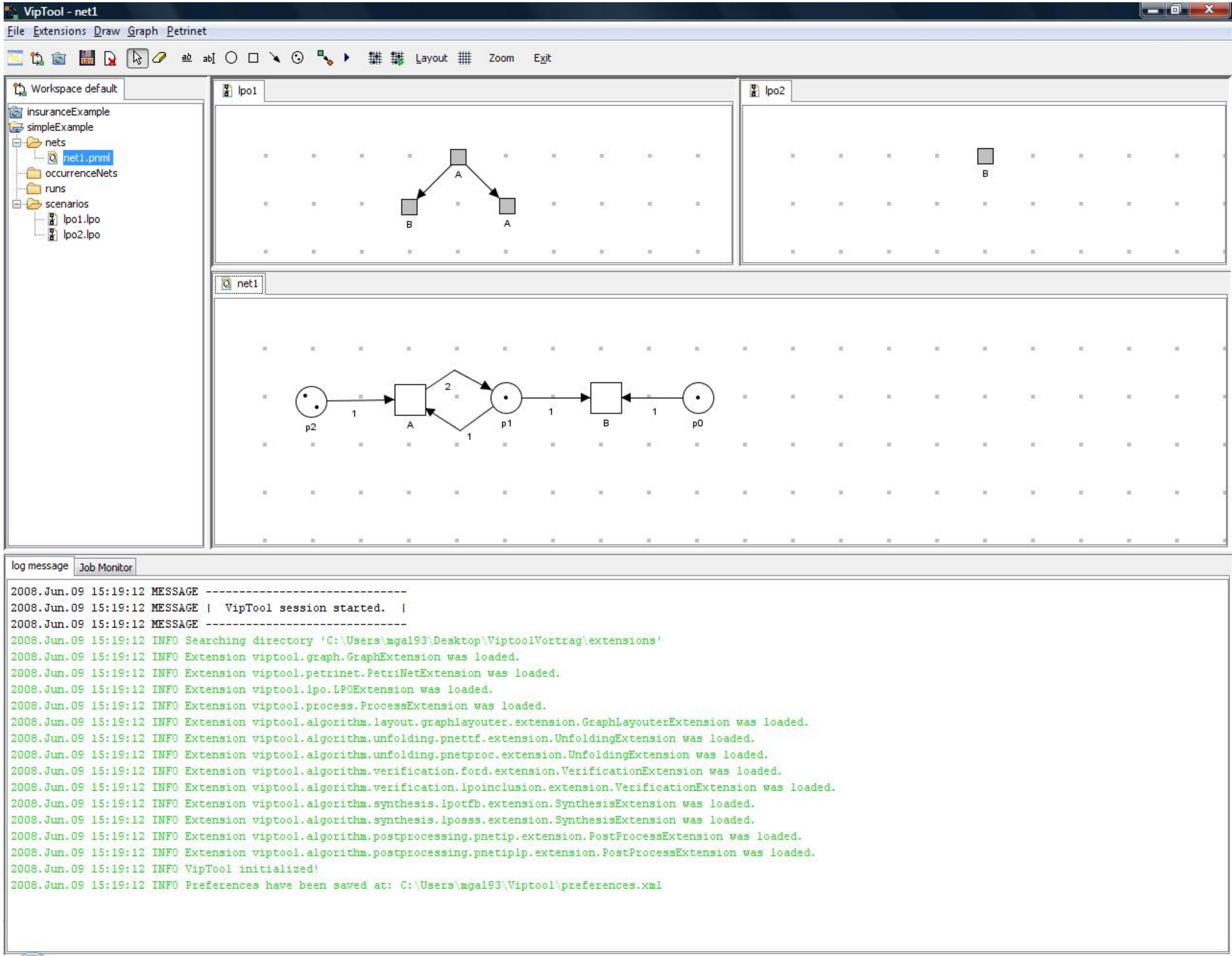

VipTool - log message
File Extensions
Exit
Workspace default
insuranceExample
netsDamageReport
netsDamageReportComplex
damageReport.pnml
scenariosDamageReport
scenariosDamageReportCo
simpleExample
damageReport
Assign Claim Expert
Check Damage
Receive Claim
Send Refusal Letter
Register Claim
Send Acceptance Letter
Estimate Damage
Pay Damage
Check Insurance
Ask Additional Queries
Set Aside Reserves
Complete Claim
log message
Job Monitor
2008.Jun.09 16:07:00 MESSAGE ------------------------------
2008.Jun.09 16:07:00 MESSAGE | VipTool session started. |

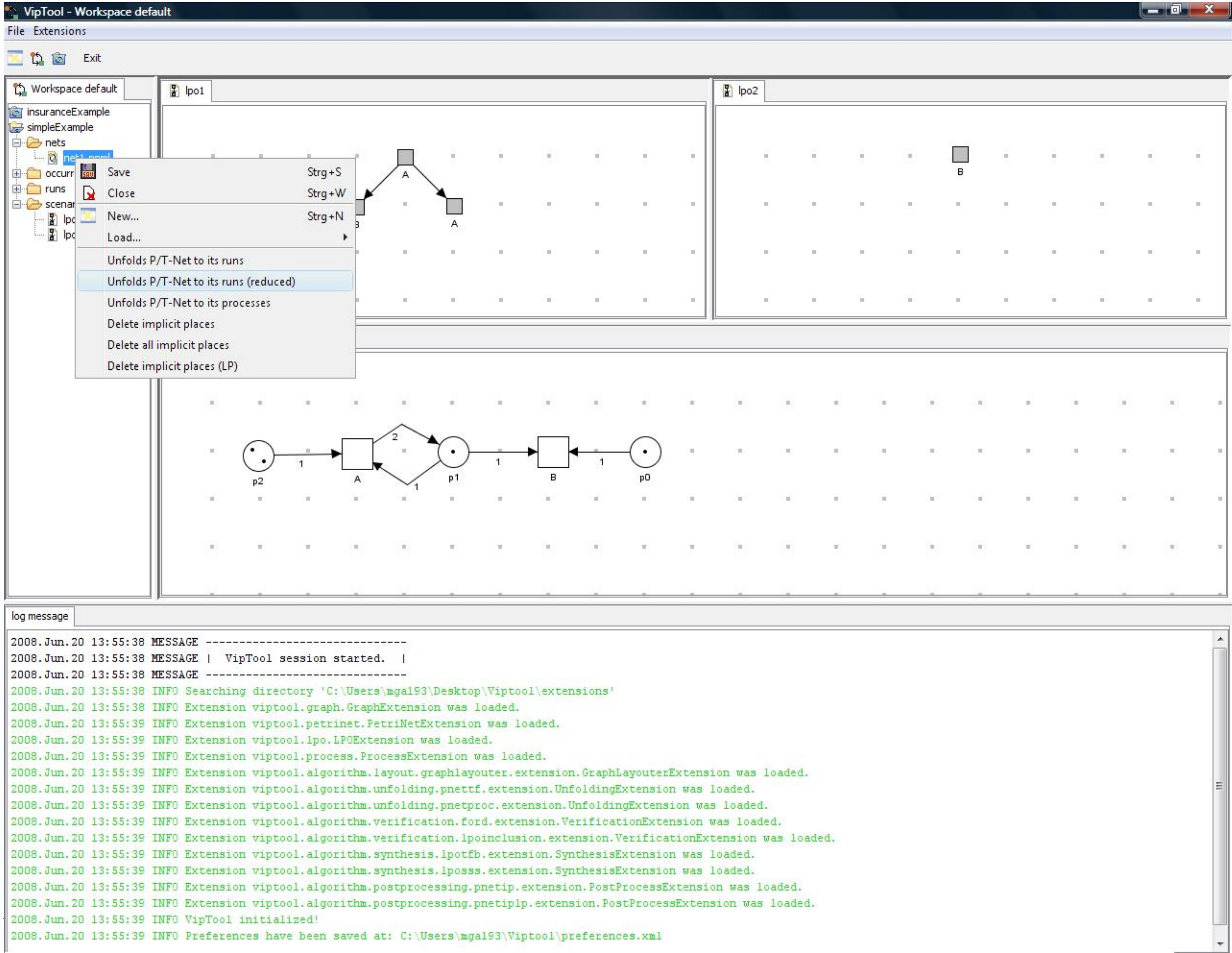

VipTool - damageReport
File Extensions Draw Graph Petrinet
Layout
Zoom
Exit
Workspace default
insuranceExample
netsDamageReport
netsDamageReportComplex
damageReport.pnml
scenariosDamag
scenariosDamag
simpleExample
Save Strg+S
Close Strg+W
New... Strg+N
Load...
Unfolds P/T-Net to its runs
Unfolds P/T-Net to its runs (reduced)
Unfolds P/T-Net to its processes
Delete implicit places
Delete all implicit places
Delete implicit places (LP)
damageReport
m Expert
Check Damage
Send Refusal Letter
Register Claim
Send Acceptance Letter
Estimate Damage
Pay Damage
Check Insurance
Ask Additional Queries
Set Aside Reserves
Complete Claim
p0
p1
p2
p3
p4
p5
p6
p9
p10
p11
p12
p13
p14
p15
p16
log message
2008.Jun.16 10:20:50 MESSAGE ------------------------------
2008.Jun.16 10:20:50 MESSAGE | VipTool session started. |
2008.Jun.16 10:20:50 MESSAGE ------------------------------
2008.Jun.16 10:20:52 INFO Searching directory 'C:\Users\mga193\Desktop\Viptool\extensions'
2008.Jun.16 10:20:52 INFO Extension viptool.graph.GraphExtension was loaded.
2008.Jun.16 10:20:52 INFO Extension viptool.petrinet.PetriNetExtension was loaded.

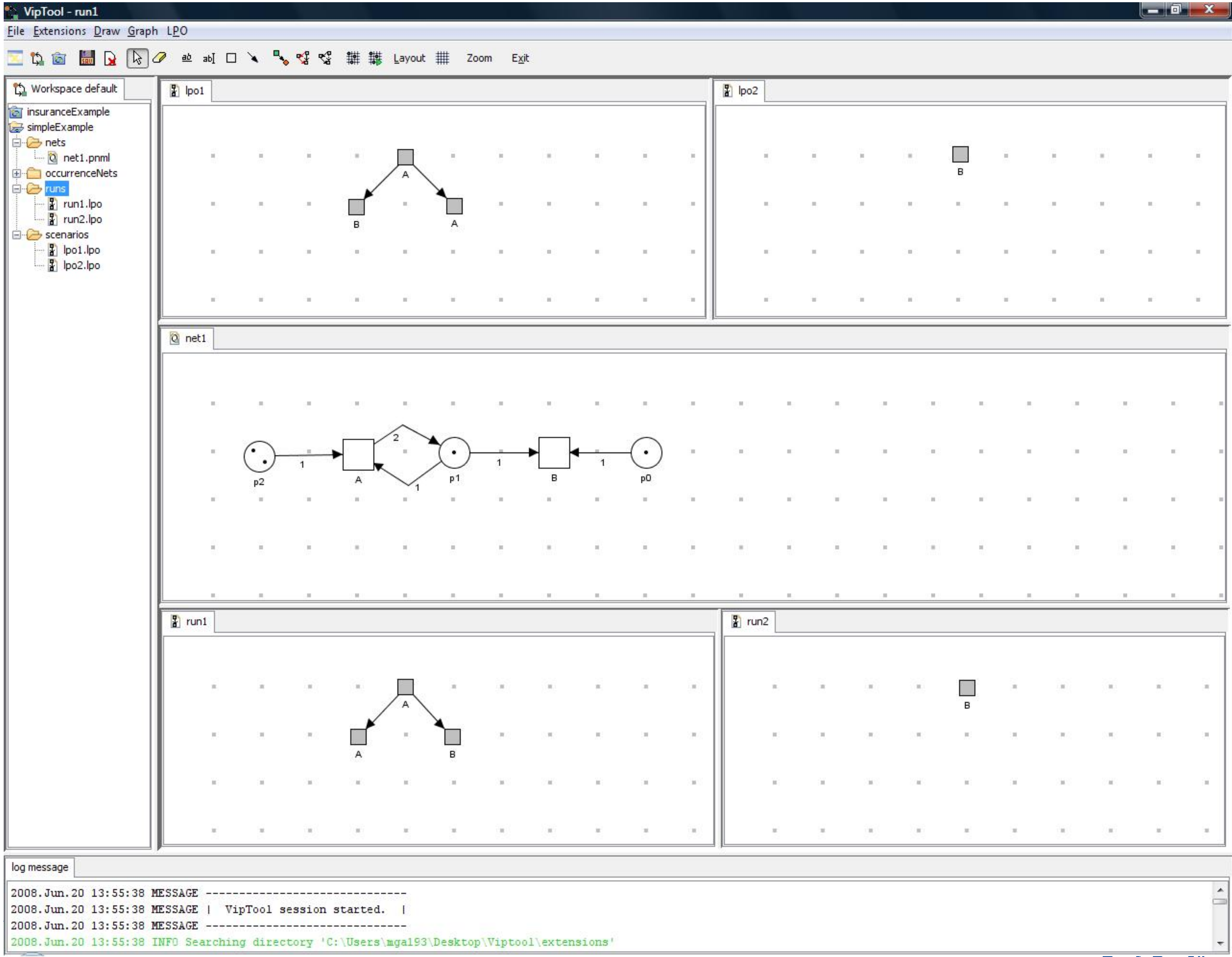

VipTool - damageReport
File Extensions Draw Graph Petrinet
Layout
Zoom
Exit
Workspace default
insuranceExample
netsDamageReport
netsDamageReportComplex
damageReport.pnml
scenariosDamageReport
scenariosDamageReportCo
simpleExample
damageReport
p8
Receive Claim
p7
Assign Claim Expert
p6
Register Claim
p15
p11
Check Damage
p14
p10
Check Insurance
p13
p0
Set Aside Reserves
p12
Send Refusal Letter
p9
Send Acceptance Letter
p1
Estimate Damage
p3
Pay Damage
p2
p16
Ask Additional Queries
Complete Claim
log message
Job Monitor
2008.Jun.09 16:07:00 MESSAGE ------------------------------
2008.Jun.09 16:07:00 MESSAGE | VipTool session started. |

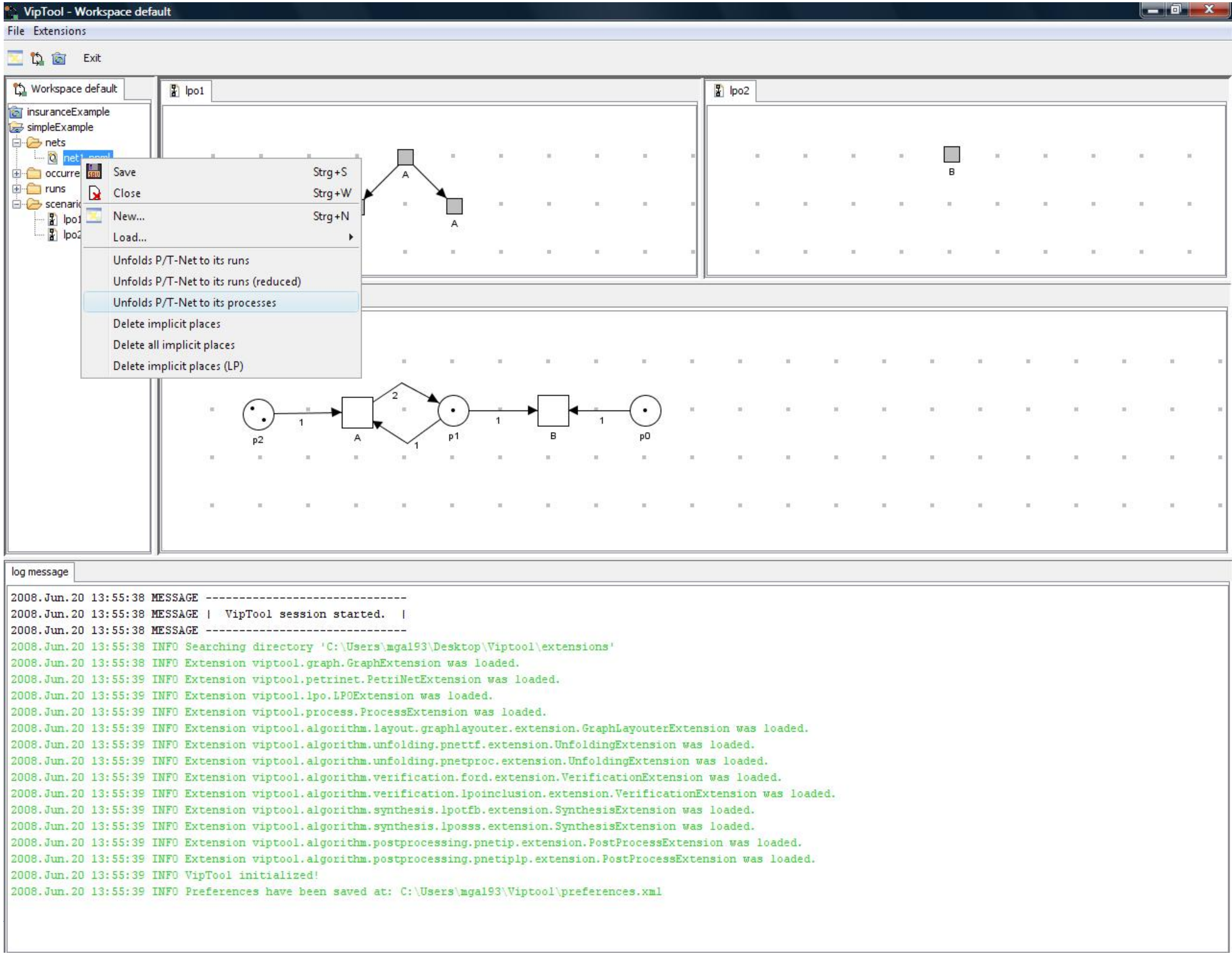

VipTool - lpo1
File Extensions Draw Graph LPO
Layout
Zoom
Exit
Workspace default
insuranceExample
simpleExample
nets
occurrenceNets
runs
scenarios
lpo1.lpo
lpo2.lpo
lpo1
A
B
A
lpo2
B
log message
2008.Jun.09 15:19:12 MESSAGE ------------------------------
2008.Jun.09 15:19:12 MESSAGE | VipTool session started. |
2008.Jun.09 15:19:12 MESSAGE ------------------------------
2008.Jun.09 15:19:12 INFO Searching directory 'C:\Users\mga193\Desktop\ViptoolVortrag\extensions'
2008.Jun.09 15:19:12 INFO Extension viptool.graph.GraphExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.petrinet.PetriNetExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.lpo.LPOExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.process.ProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.layout.graphlayouter.extension.GraphLayouterExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.unfolding.pnettf.extension.UnfoldingExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.unfolding.pnetproc.extension.UnfoldingExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.verification.ford.extension.VerificationExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.verification.lpoinclusion.extension.VerificationExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.synthesis.lpotfb.extension.SynthesisExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.synthesis.lposss.extension.SynthesisExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.postprocessing.pnetip.extension.PostProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.postprocessing.pnetiplp.extension.PostProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO VipTool initialized!
2008.Jun.09 15:19:12 INFO Preferences have been saved at: C:\Users\mga193\Viptool\preferences.xml

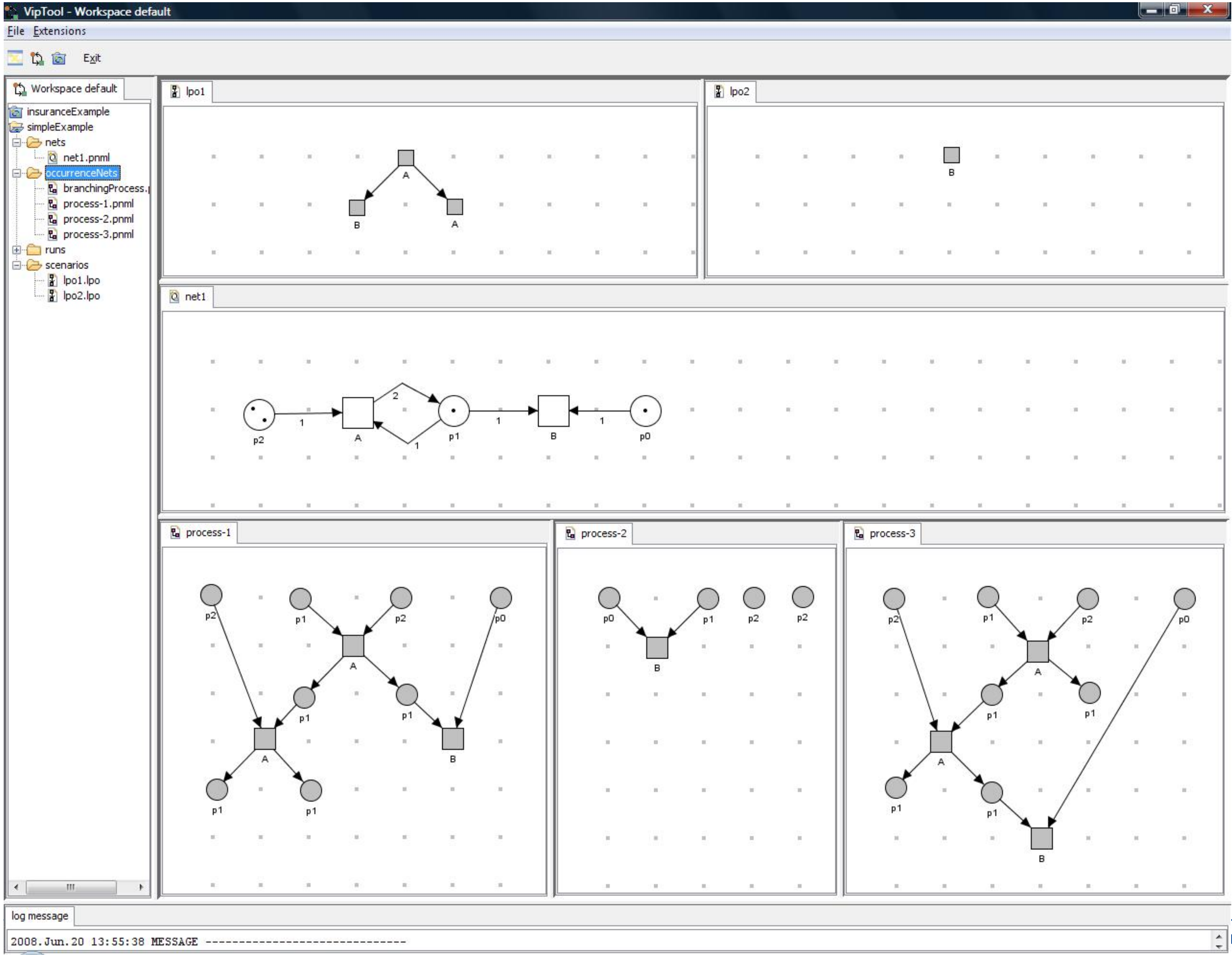

VipTool - Workspace default
File Extensions
Exit
Workspace default
insuranceExample
simpleExample
nets
occurrenceNets
runs
lpo1
lpo2
Open
Delete
New...
Strg+N
Load...
Synthesize and test petrinet (SSS)
Synthesize and test petrinet (TFB)
Synthesize and test petrinet (TFS)
Synthesize petrinet (TFB)
Synthesize petrinet (TFS)
Synthesize petrinet (SSS)
A
B
A
B
log message
2008.Jun.09 15:19:12 MESSAGE -----------------------------
2008.Jun.09 15:19:12 MESSAGE | VipTool session started. |
2008.Jun.09 15:19:12 MESSAGE -----------------------------
2008.Jun.09 15:19:12 INFO Searching directory 'C:\Users\mgal93\Desktop\ViptoolVortrag\extensions'
2008.Jun.09 15:19:12 INFO Extension viptool.graph.GraphExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.petrinet.PetriNetExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.lpo.LPOExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.process.ProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.layout.graphlayouter.extension.GraphLayouterExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.unfolding.pnettf.extension.UnfoldingExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.unfolding.pnetproc.extension.UnfoldingExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.verification.ford.extension.VerificationExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.verification.lpoinclusion.extension.VerificationExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.synthesis.lpotfb.extension.SynthesisExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.synthesis.lposss.extension.SynthesisExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.postprocessing.pnetip.extension.PostProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.postprocessing.pnetiplp.extension.PostProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO VipTool initialized!
2008.Jun.09 15:19:12 INFO Preferences have been saved at: C:\Users\mgal93\Viptool\preferences.xml


VipTool - net1
File Extensions Draw Graph Petrinet
Layout
Zoom
Exit
Workspace default
insuranceExample
simpleExample
nets
net1.pnml
occurrenceNets
runs
scenarios
lpo1.lpo
lpo2.lpo
lpo1
lpo2
net1
p2
A
p1
B
p0
log message
Job Monitor
2008.Jun.09 15:19:12 MESSAGE ------------------------------
2008.Jun.09 15:19:12 MESSAGE | VipTool session started. |
2008.Jun.09 15:19:12 MESSAGE ------------------------------
2008.Jun.09 15:19:12 INFO Searching directory 'C:\Users\mga193\Desktop\ViptoolVortrag\extensions'
2008.Jun.09 15:19:12 INFO Extension viptool.graph.GraphExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.petrinet.PetriNetExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.lpo.LPOExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.process.ProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.layout.graphlayouter.extension.GraphLayouterExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.unfolding.pnettf.extension.UnfoldingExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.unfolding.pnetproc.extension.UnfoldingExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.verification.ford.extension.VerificationExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.verification.lpoinclusion.extension.VerificationExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.synthesis.lpotfb.extension.SynthesisExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.synthesis.lposss.extension.SynthesisExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.postprocessing.pnetip.extension.PostProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO Extension viptool.algorithm.postprocessing.pnetiplp.extension.PostProcessExtension was loaded.
2008.Jun.09 15:19:12 INFO VipTool initialized!
2008.Jun.09 15:19:12 INFO Preferences have been saved at: C:\Users\mga193\Viptool\preferences.xml


VipTool - Workspace default
File Extensions
Exit
Workspace default
insuranceExample
simpleExample
nets
runs
lpo1
lpo2
Save Strg+S
Close Strg+W
New... Strg+N
Load...
Unfolds P/T-Net to its runs
Unfolds P/T-Net to its runs (reduced)
Unfolds P/T-Net to its processes
Delete implicit places
Delete all implicit places
Delete implicit places (LP)
A
B
p2
p1
p0
log message
2008.Jun.20 13:55:38 MESSAGE ------------------------------
2008.Jun.20 13:55:38 MESSAGE | VipTool session started. |
2008.Jun.20 13:55:38 MESSAGE ------------------------------
2008.Jun.20 13:55:38 INFO Searching directory 'C:\Users\mga193\Desktop\Viptool\extensions'
2008.Jun.20 13:55:38 INFO Extension viptool.graph.GraphExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.petrinet.PetriNetExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.lpo.LPOExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.process.ProcessExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.layout.graphlayouter.extension.GraphLayouterExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.unfolding.pnettf.extension.UnfoldingExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.unfolding.pnetproc.extension.UnfoldingExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.verification.ford.extension.VerificationExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.verification.lpoinclusion.extension.VerificationExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.synthesis.lpotfb.extension.SynthesisExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.synthesis.lposss.extension.SynthesisExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.postprocessing.pnetip.extension.PostProcessExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.postprocessing.pnetiplp.extension.PostProcessExtension was loaded.
2008.Jun.20 13:55:39 INFO VipTool initialized!
2008.Jun.20 13:55:39 INFO Preferences have been saved at: C:\Users\mga193\Viptool\preferences.xml

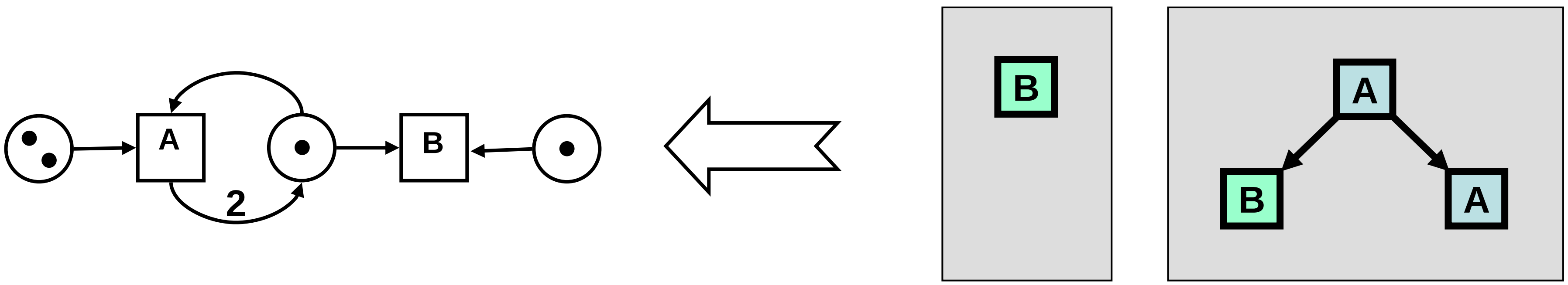

VipTool - run1
File Extensions Draw Graph LPO
Layout Zoom Exit
Workspace default
insuranceExample
simpleExample
nets
net1.pnml
occurrenceNets
runs
run1.lpo
run2.lpo
scenarios
lpo1.lpo
lpo2.lpo
lpo1
lpo2
net1
run1
run2
log message
2008.Jun.20 13:55:38 MESSAGE ------------------------------
2008.Jun.20 13:55:38 MESSAGE | VipTool session started. |
2008.Jun.20 13:55:38 MESSAGE ------------------------------
2008.Jun.20 13:55:38 INFO Searching directory 'C:\Users\mga193\Desktop\Viptool\extensions'


VipTool - Workspace default
File Extensions
Exit
Workspace default
insuranceExample
simpleExample
nets
runs
lpo1
lpo2
Save Strg+S
Close Strg+W
New... Strg+N
Load...
Unfolds P/T-Net to its runs
Unfolds P/T-Net to its runs (reduced)
Unfolds P/T-Net to its processes
Delete implicit places
Delete all implicit places
Delete implicit places (LP)
A
B
p2
p1
p0
1
2
log message
2008.Jun.20 13:55:38 MESSAGE -----------------------------
2008.Jun.20 13:55:38 MESSAGE | VipTool session started. |
2008.Jun.20 13:55:38 MESSAGE -----------------------------
2008.Jun.20 13:55:38 INFO Searching directory 'C:\Users\mgal93\Desktop\Viptool\extensions'
2008.Jun.20 13:55:38 INFO Extension viptool.graph.GraphExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.petrinet.PetriNetExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.lpo.LPOExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.process.ProcessExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.layout.graphlayouter.extension.GraphLayouterExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.unfolding.pnettf.extension.UnfoldingExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.unfolding.pnetproc.extension.UnfoldingExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.verification.ford.extension.VerificationExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.verification.lpoinclusion.extension.VerificationExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.synthesis.lpotfb.extension.SynthesisExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.synthesis.lposss.extension.SynthesisExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.postprocessing.pnetip.extension.PostProcessExtension was loaded.
2008.Jun.20 13:55:39 INFO Extension viptool.algorithm.postprocessing.pnetiplp.extension.PostProcessExtension was loaded.
2008.Jun.20 13:55:39 INFO VipTool initialized!
2008.Jun.20 13:55:39 INFO Preferences have been saved at: C:\Users\mgal93\Viptool\preferences.xml

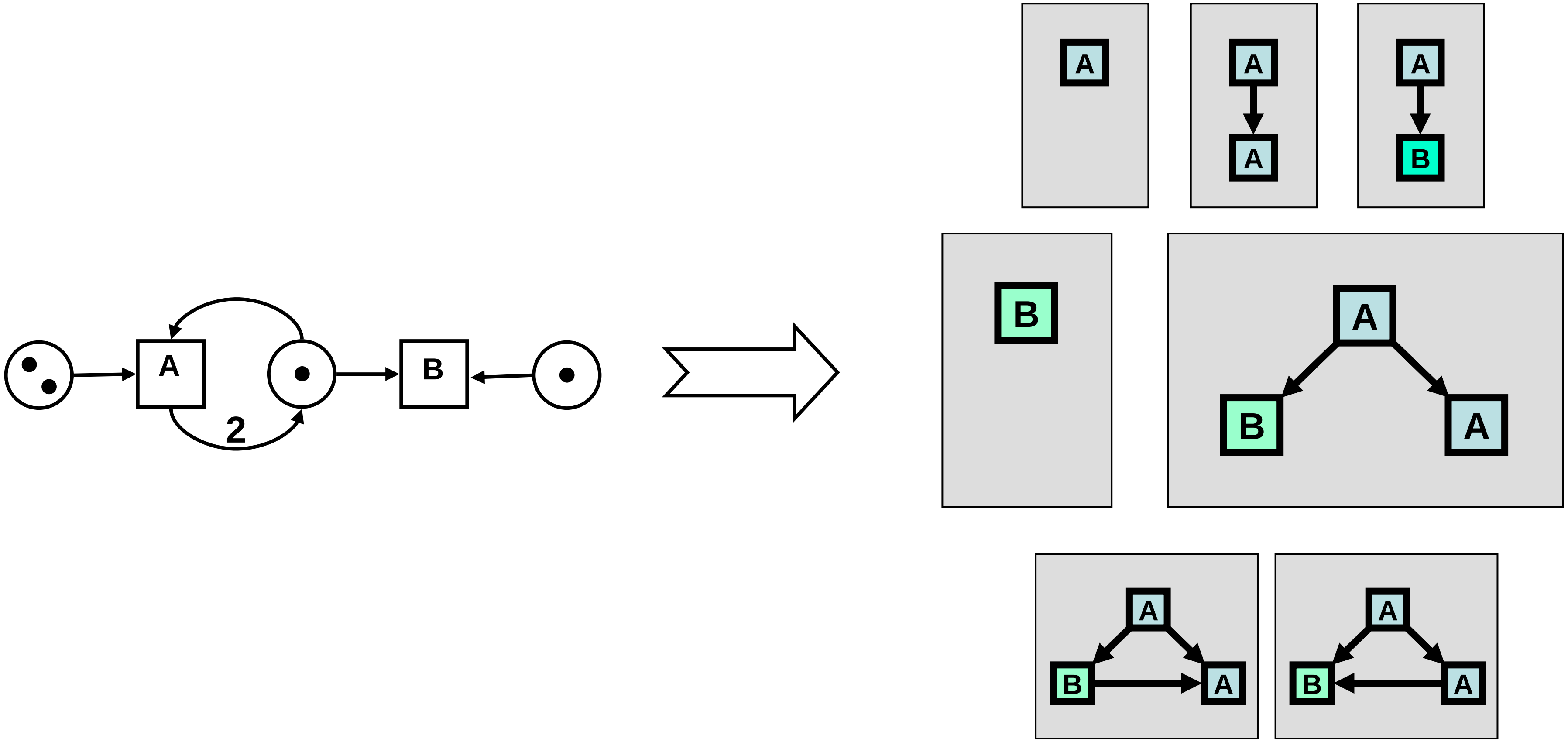


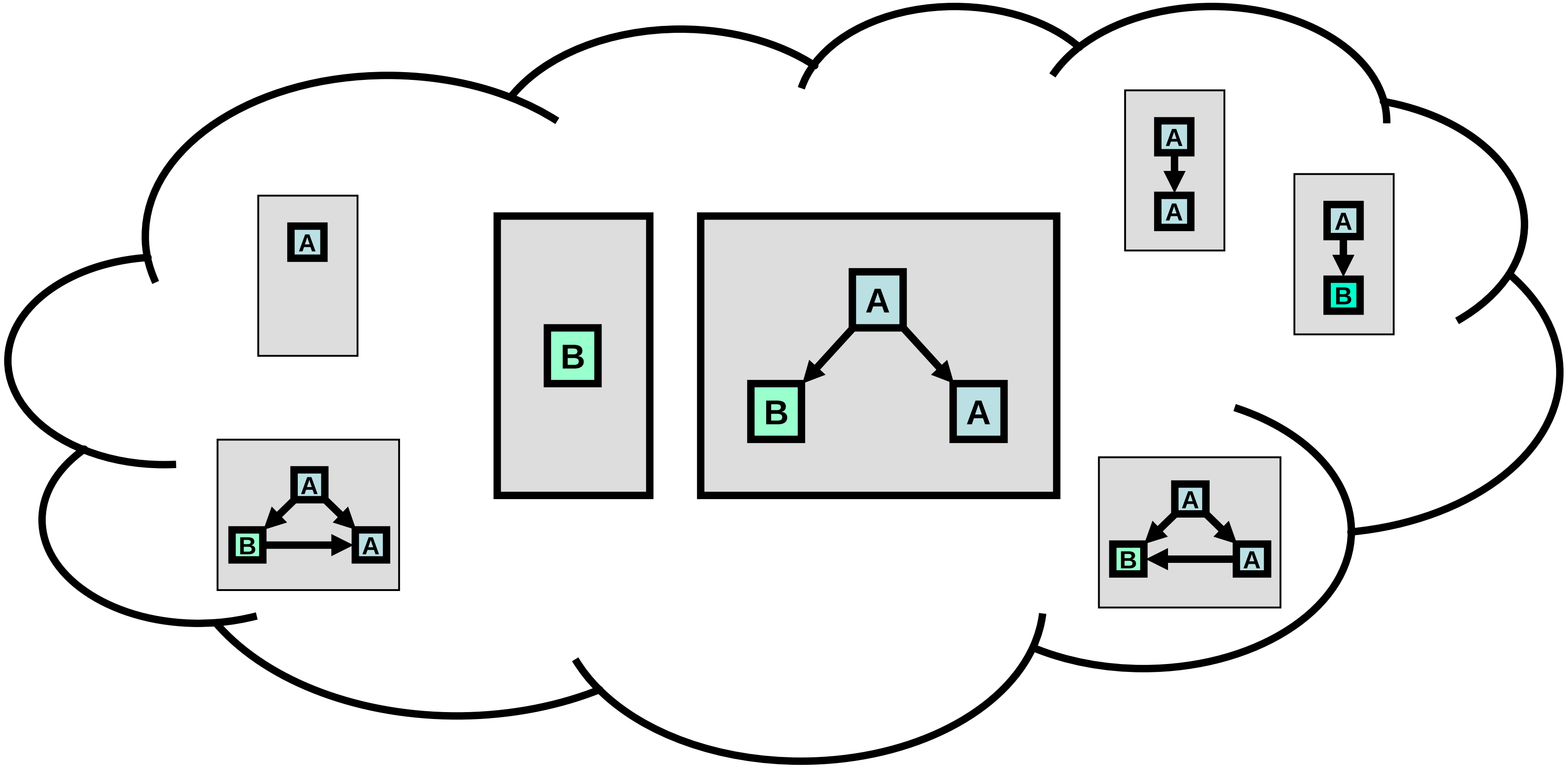
VipTool - Workspace default
File Extensions
Exit
Workspace default
insuranceExample
simpleExample
nets
net1.pnml
occurrenceNets
branchingProcess.
process-1.pnml
process-2.pnml
process-3.pnml
runs
scenarios
lpo1.lpo
lpo2.lpo
lpo1
lpo2
net1
process-1
process-2
process-3
log message
2008.Jun.20 13:55:38 MESSAGE ------------------------------

Synthesis Algorithm



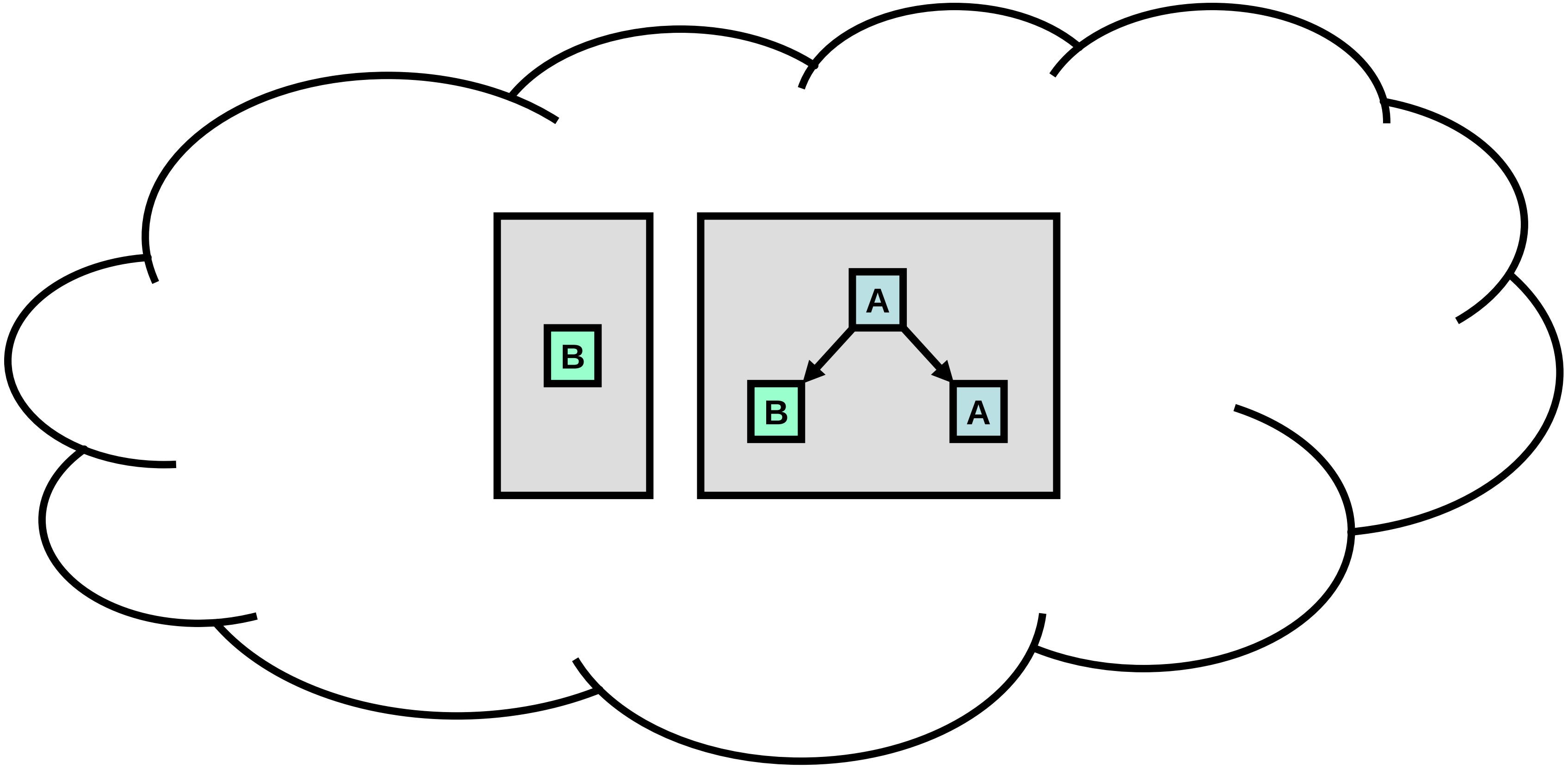
Prefixes and Sequentialization





Partial Language: finite set of LPOs, closed under prefixing and sequalization

Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel



Partial Language: … represented by
maximal LPOs with **minimal** order

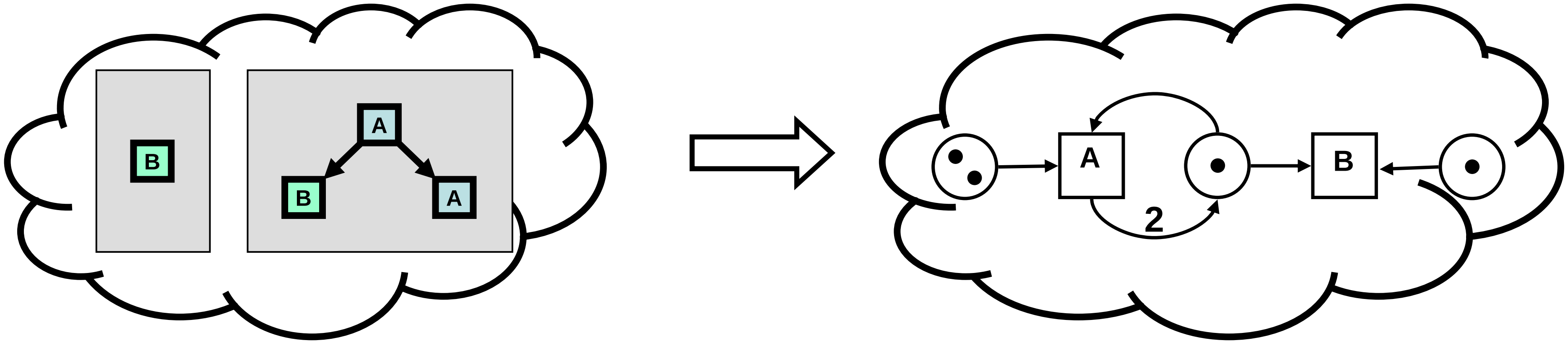
Synthese

given:

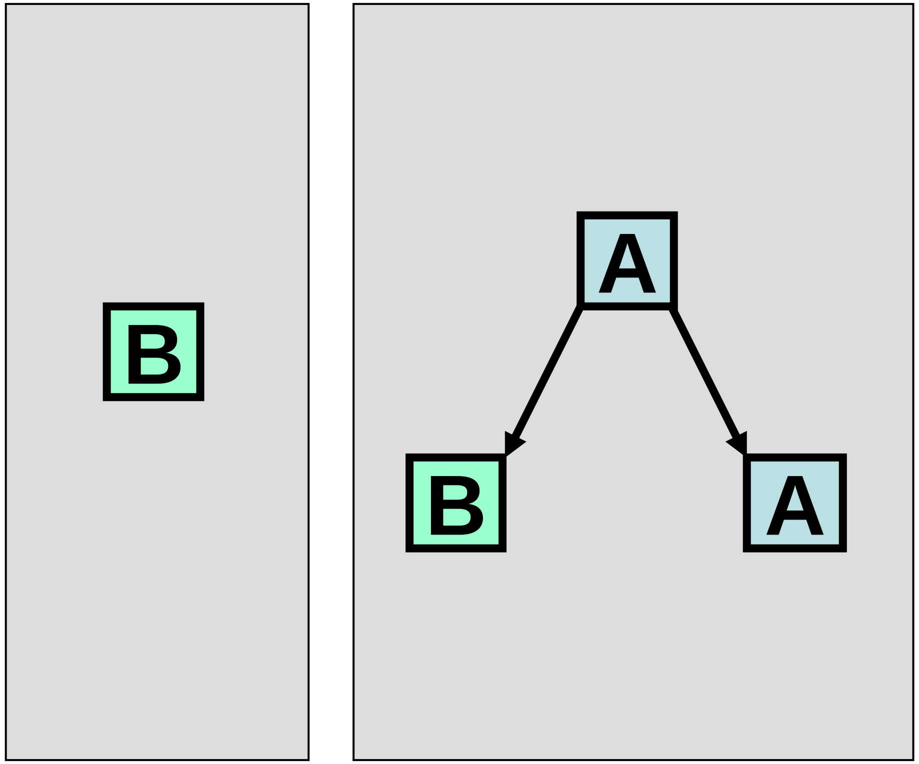
A finite partial language L

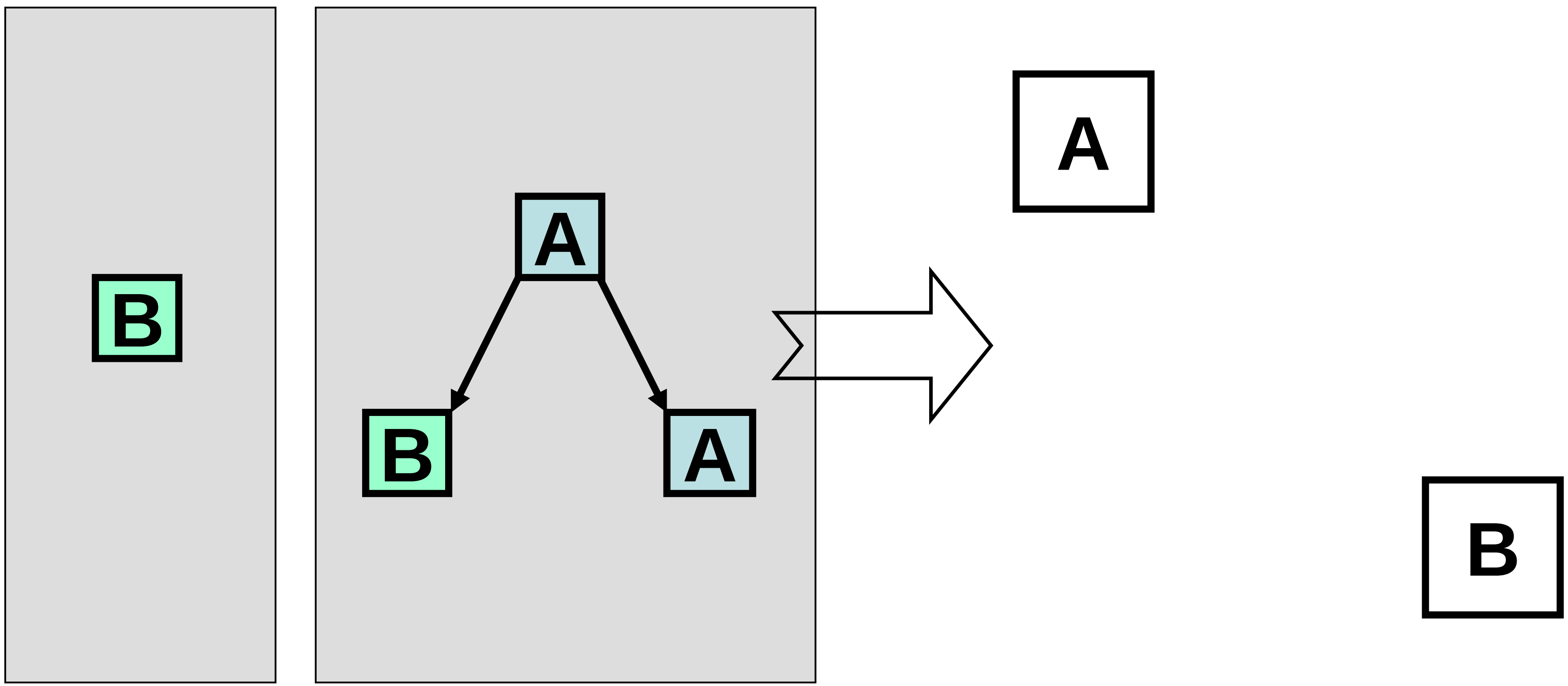
searched:

Petri net N with behaviour L

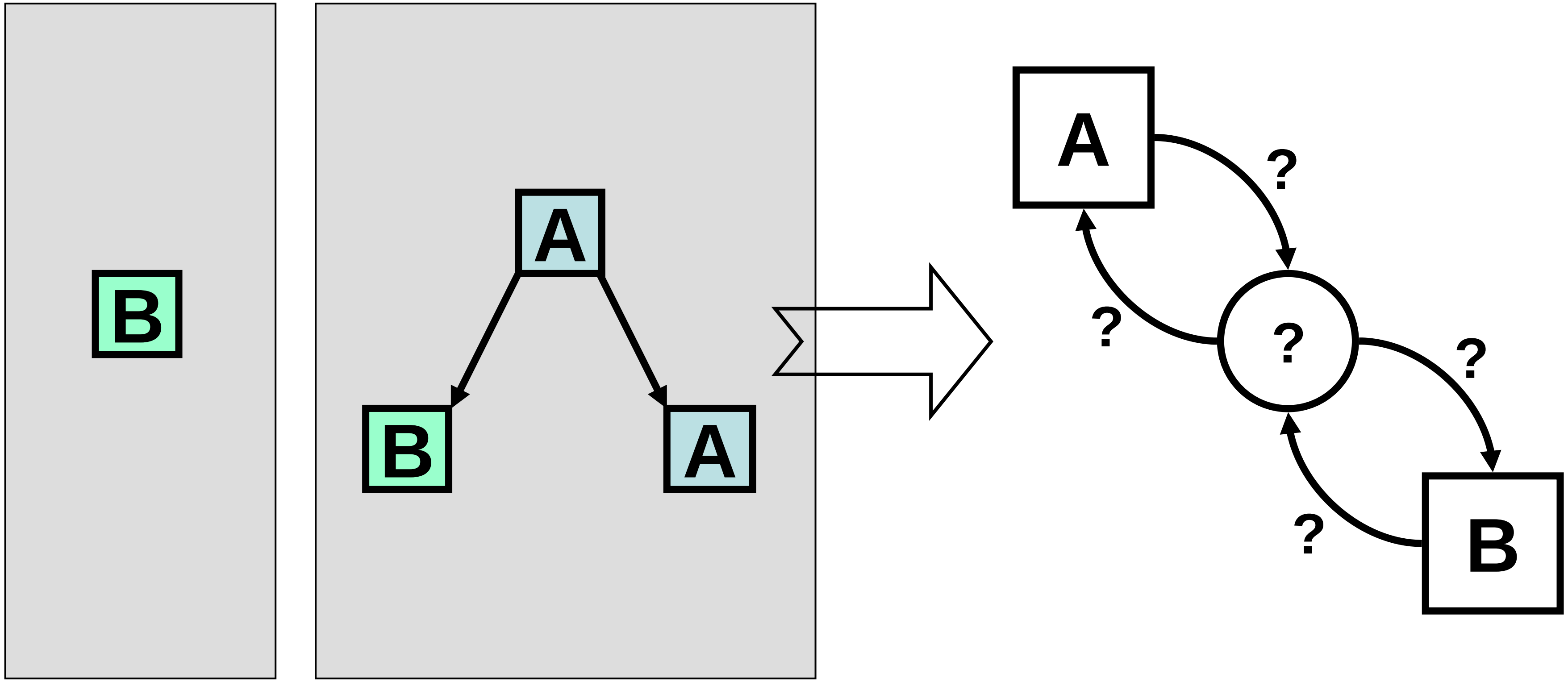


Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel


B
A
B
A

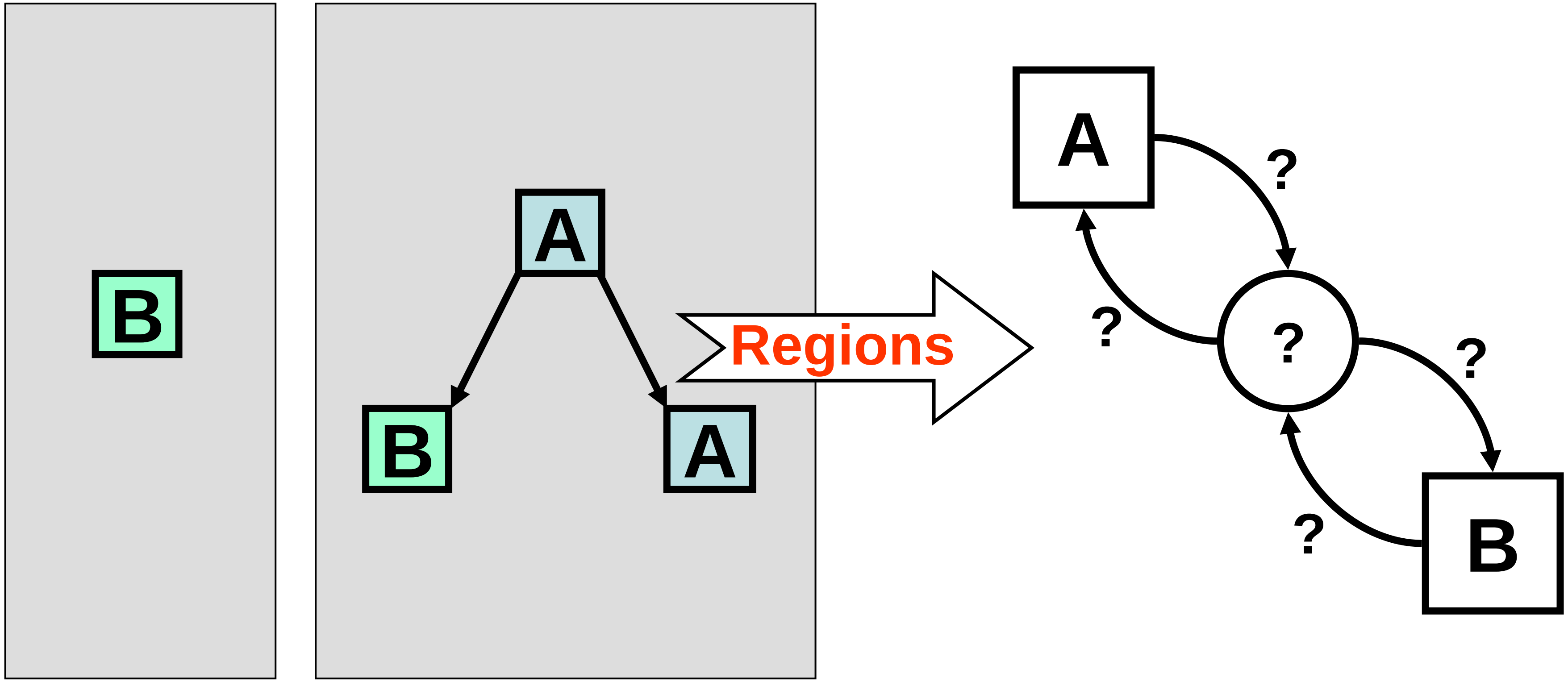


event labels ⇒ transitions of the net



how do we find the places ?

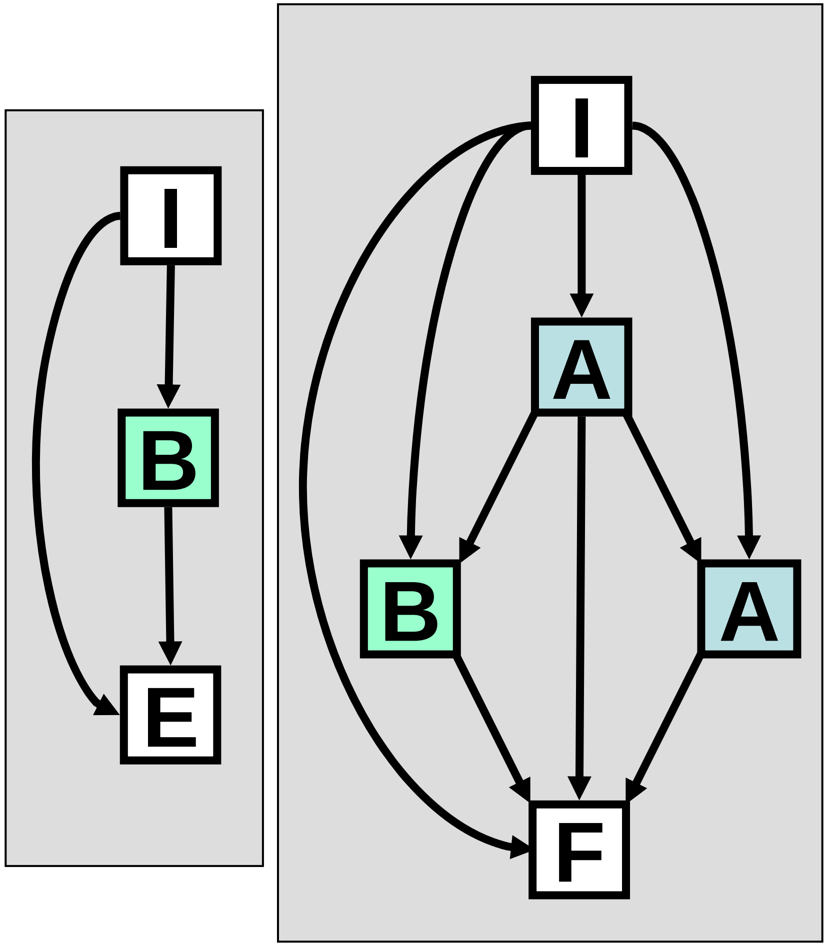
Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel



how do we find the places ?

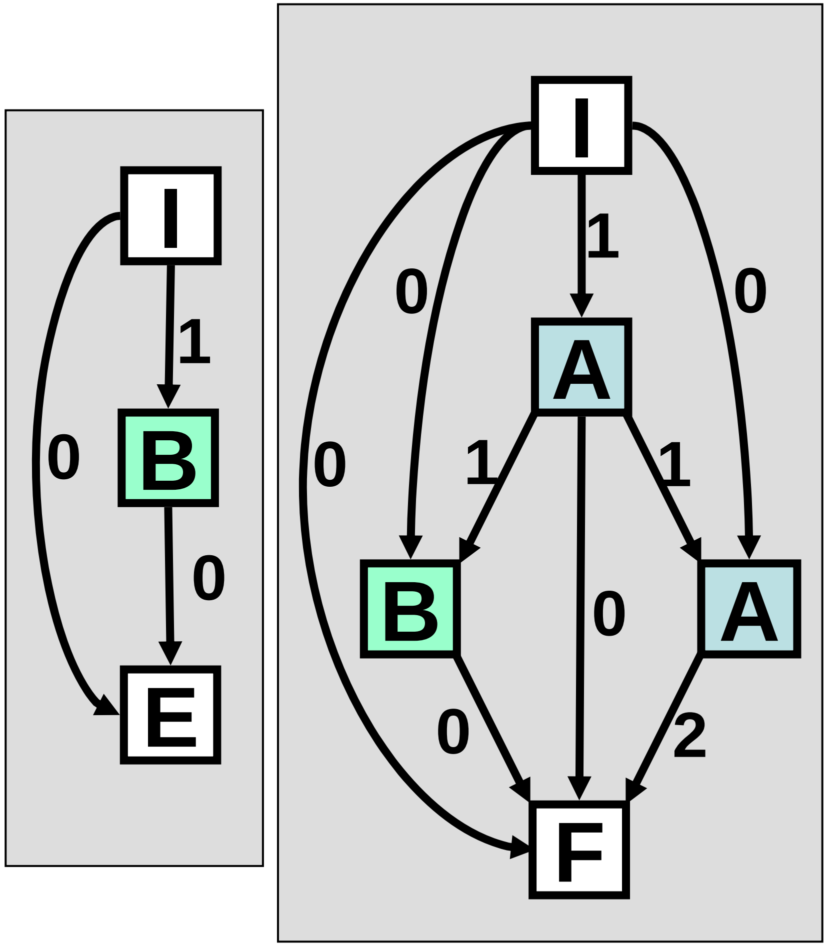
Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

token flow regions



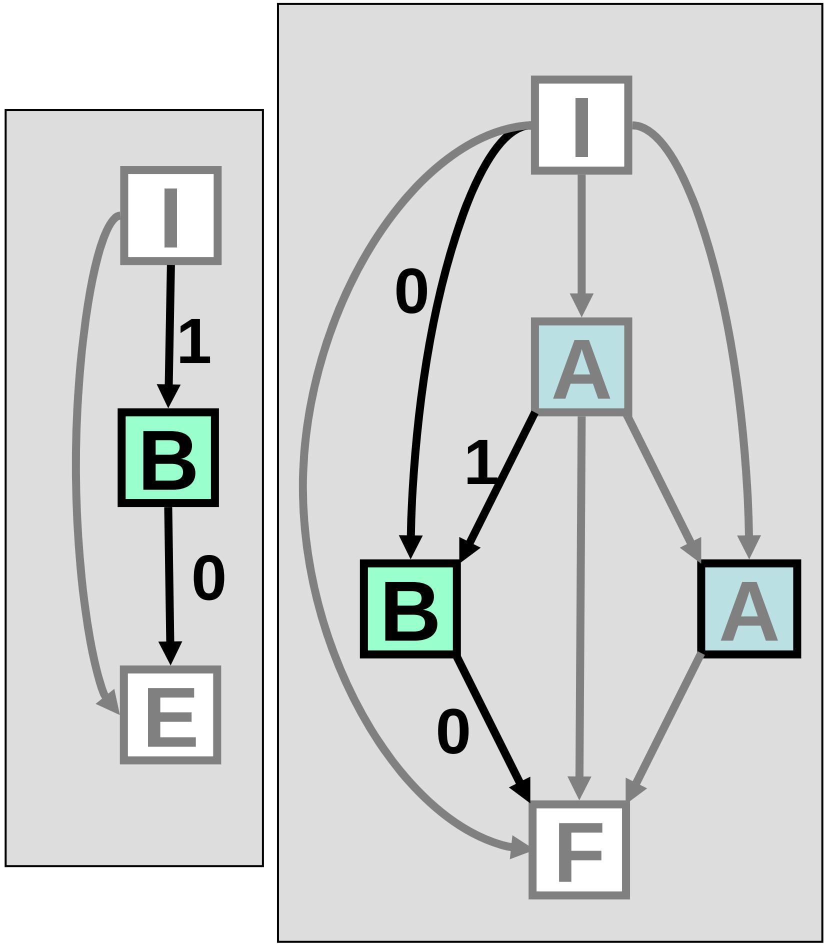
addition of initial event (I) (generates initial marking)
and final events (E, F) (consume final marking)

token flow regions



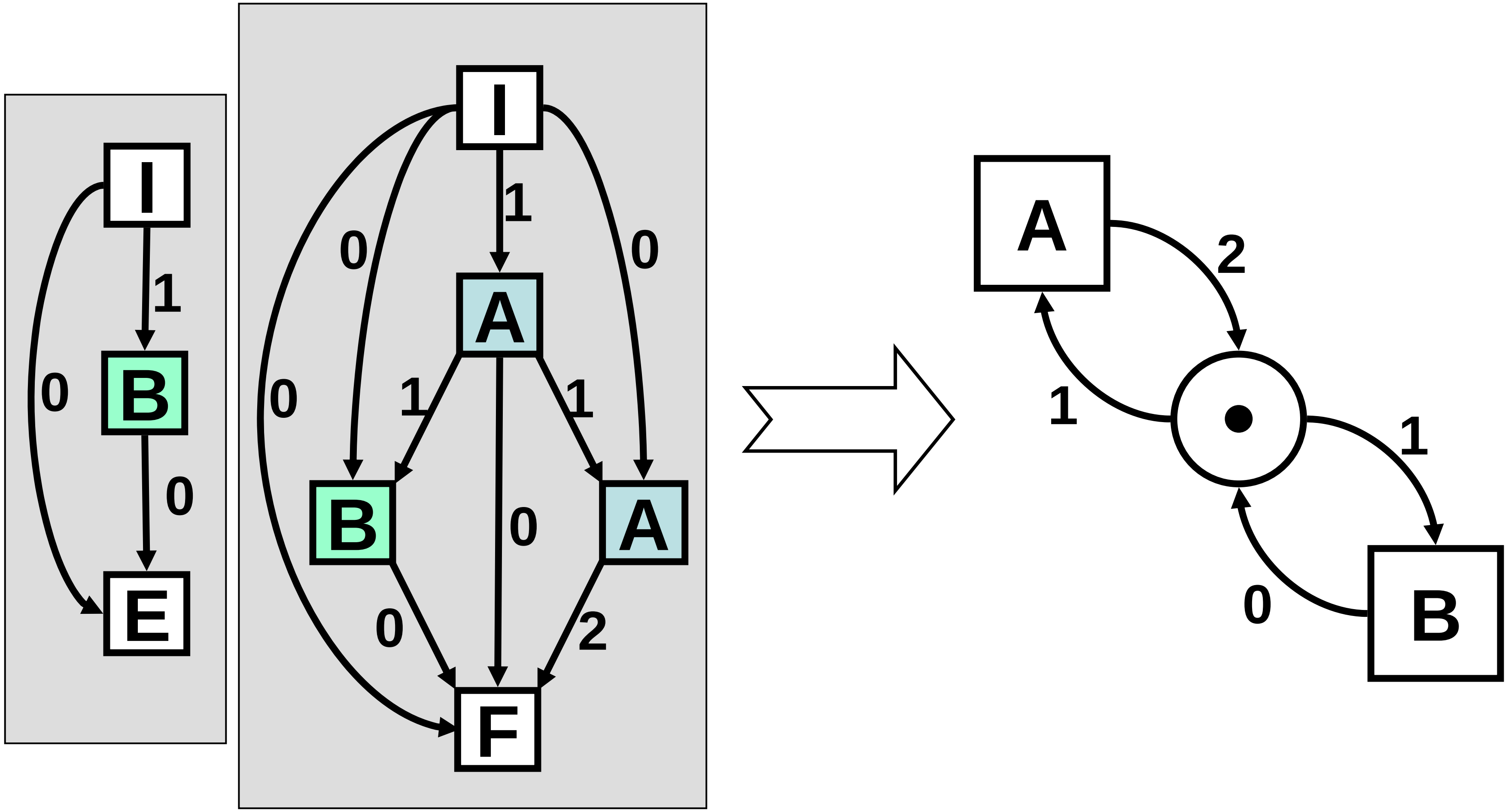
equally labelled events have equal input and output flow

token flow regions



equally labelled events have equal input and output flow

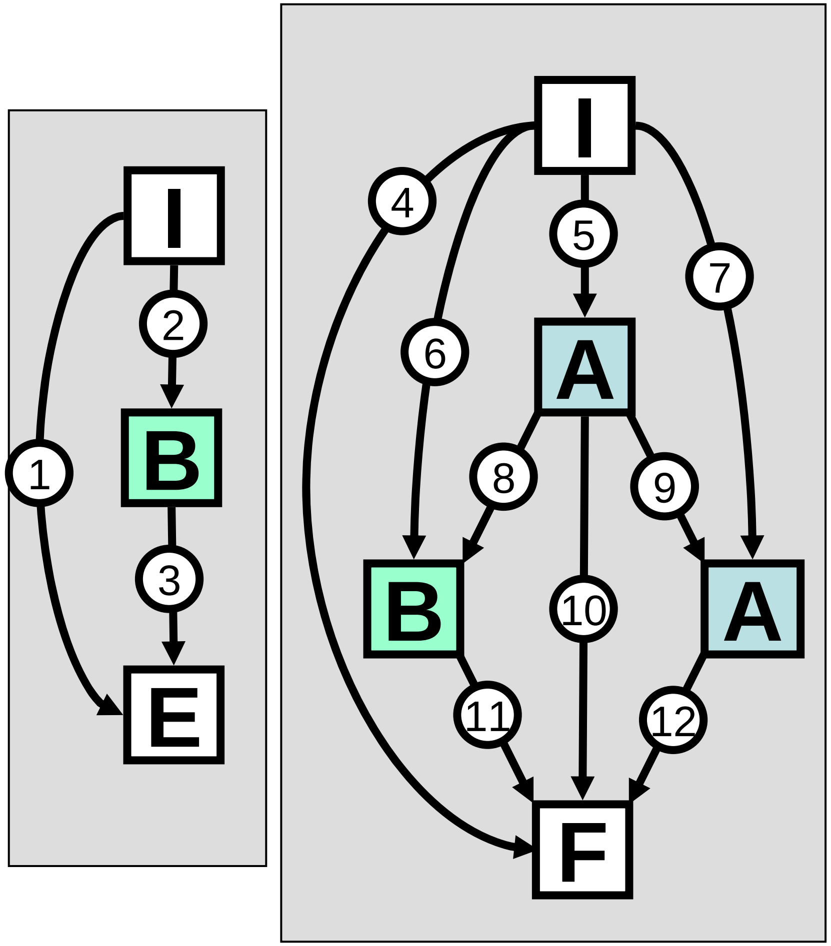
token flow regions



each token flow region generates a possible place

Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

token flow regions



① + ② - ④ - ⑤ - ⑥ - ⑦ = 0

② - ⑥ - ⑧ = 0

③ - ⑪ = 0

⑤ - ⑦ - ⑨ = 0

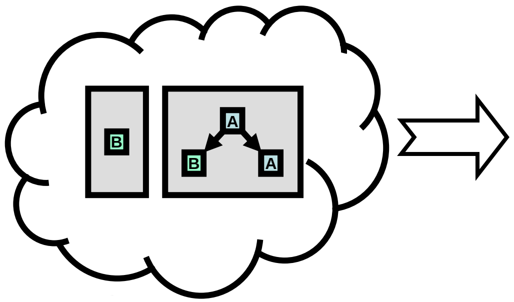
⑧ + ⑨ + ⑩ - ⑫ = 0

① ≥ 0	⑤ ≥ 0	⑨ ≥ 0
② ≥ 0	⑥ ≥ 0	⑩ ≥ 0
③ ≥ 0	⑦ ≥ 0	⑪ ≥ 0
④ ≥ 0	⑧ ≥ 0	⑫ ≥ 0

each token flow region is an integral solution of an inequation system

Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

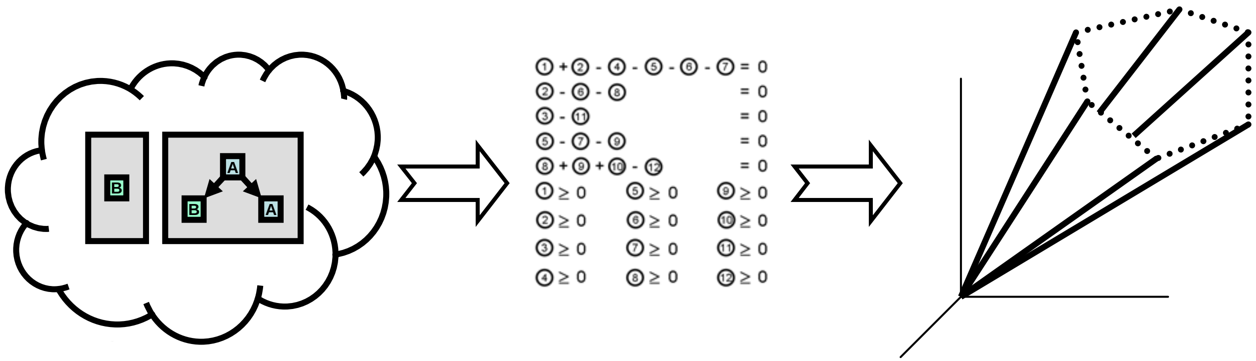
which possible places should we take?



$$\textcircled{1} + \textcircled{2} - \textcircled{4} - \textcircled{5} - \textcircled{6} - \textcircled{7} = 0$$
$$\textcircled{2} - \textcircled{6} - \textcircled{8} = 0$$
$$\textcircled{3} - \textcircled{11} = 0$$
$$\textcircled{5} - \textcircled{7} - \textcircled{9} = 0$$
$$\textcircled{8} + \textcircled{9} + \textcircled{10} - \textcircled{12} = 0$$
$$\textcircled{1} \geq 0 \quad \textcircled{5} \geq 0 \quad \textcircled{9} \geq 0$$
$$\textcircled{2} \geq 0 \quad \textcircled{6} \geq 0 \quad \textcircled{10} \geq 0$$
$$\textcircled{3} \geq 0 \quad \textcircled{7} \geq 0 \quad \textcircled{11} \geq 0$$
$$\textcircled{4} \geq 0 \quad \textcircled{8} \geq 0 \quad \textcircled{12} \geq 0$$

Lehrgebiet Softwaretechnik und Theorie der Programmierung
Prof. Dr. Jörg Desel

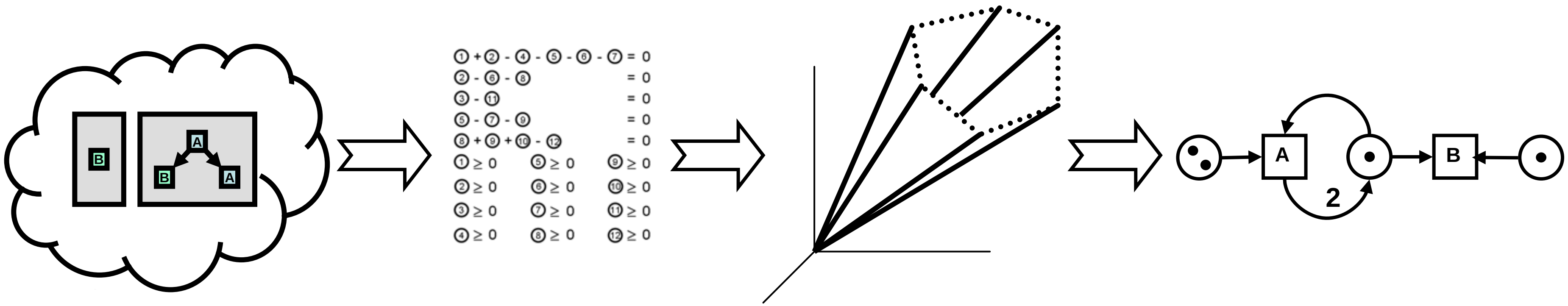
which possible places should we take?



solution space of the inequation system:

- **pointed polyhedrical cone**
- **generated by a finite set of rays**

which possible places should we take?



this set of rays generates a finite set of places

Complete Algorithm

for each label generate a transition

generate inequation system for possible places

calculate all rays of the solution space

for each plave, generate a place

If the synthesis problem has a solution,
then the synthesized net is a solution

If the synthesis problem has no solution,
then the synthesized net is a best upper approximation

Partial Orders Fit For Work

Jörg Desel FernUniversität in Hagen