

Designed by Showeet.com

Block Layout

For (accepting) Petri nets

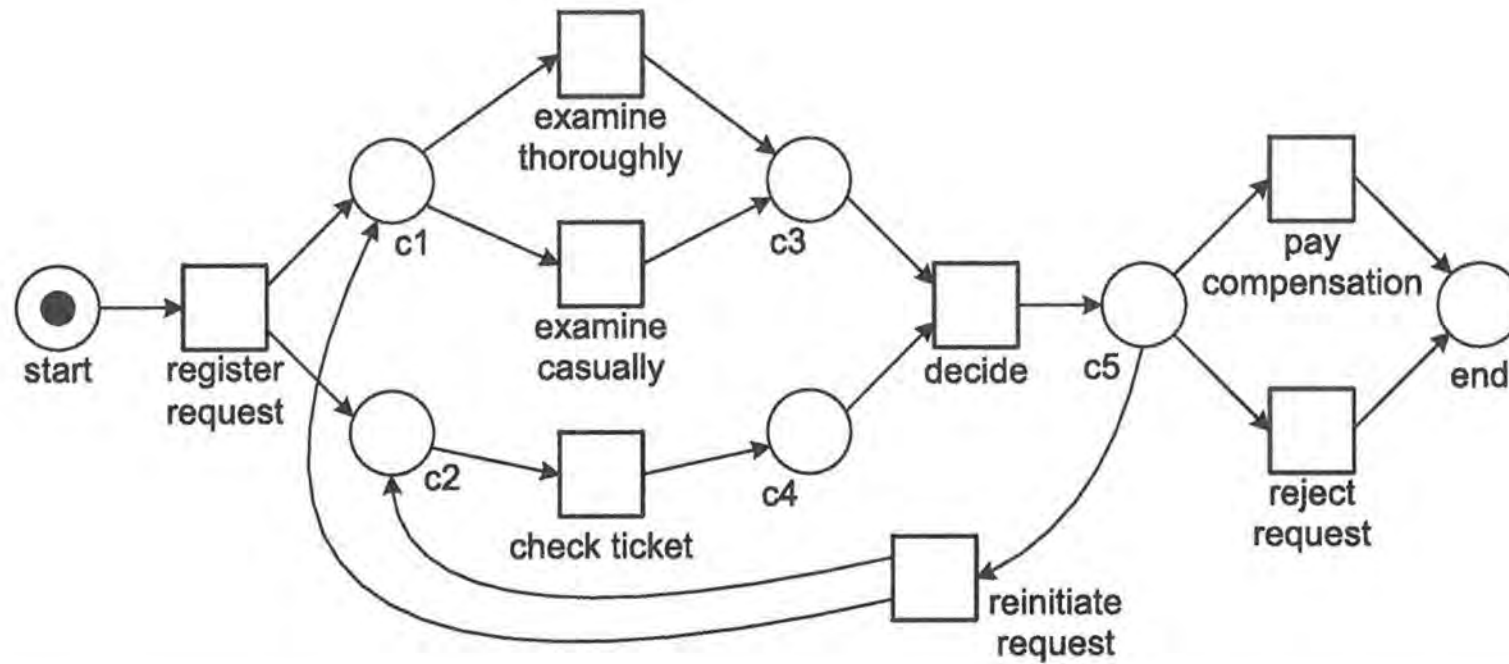
Eric Verbeek

Block Layout

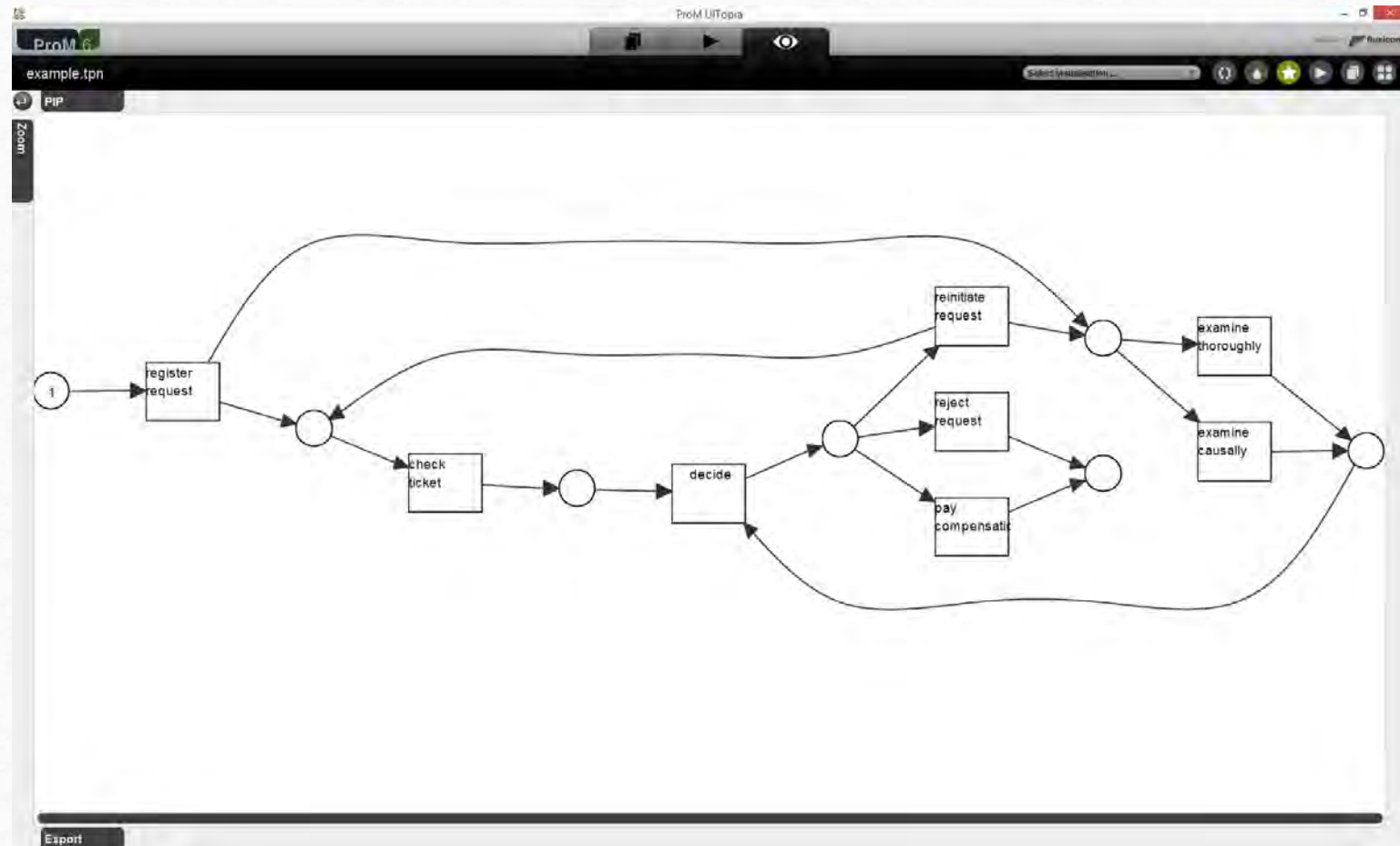
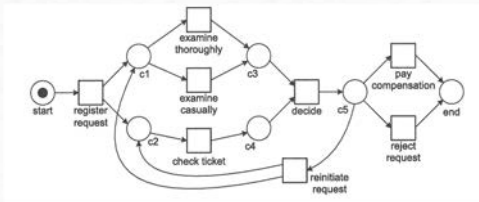
For (accepting) Petri nets

Eric Verbeek

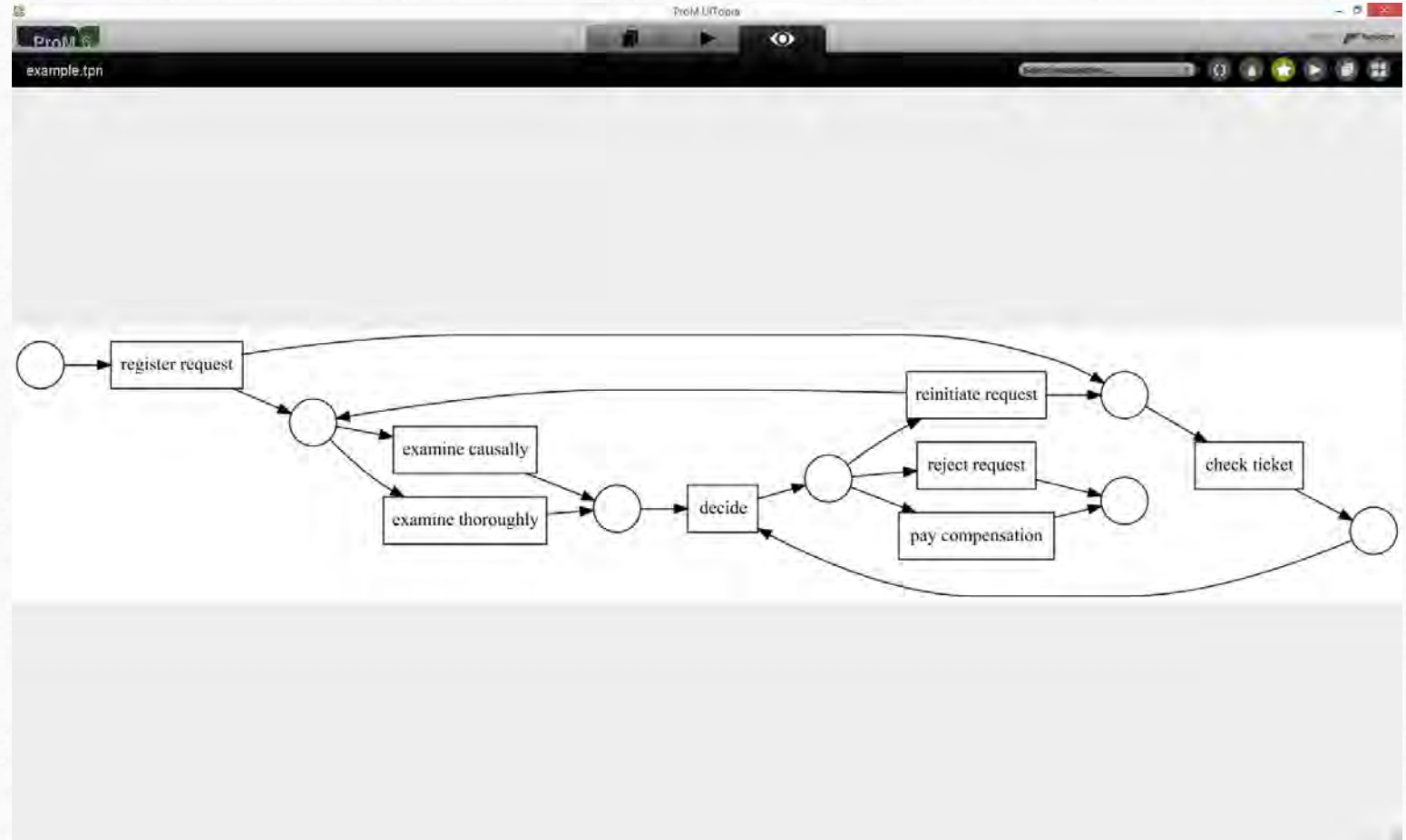
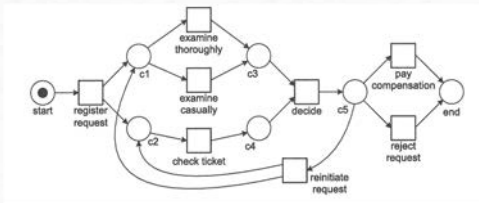
A well-known example



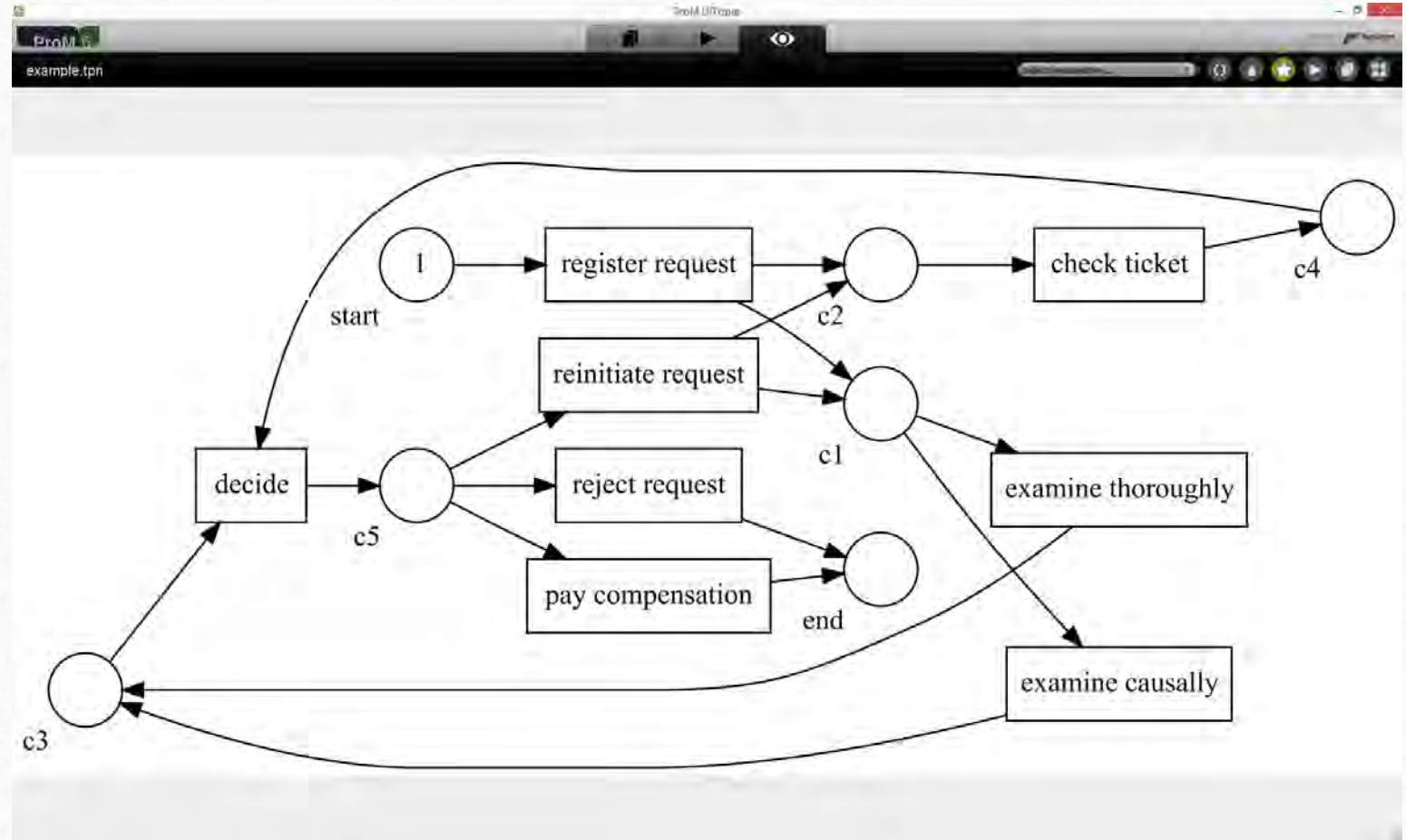
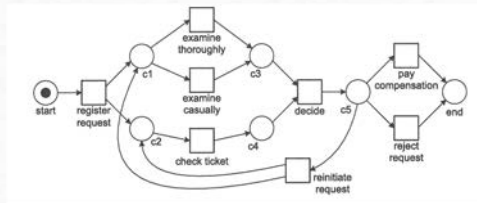
JGraph



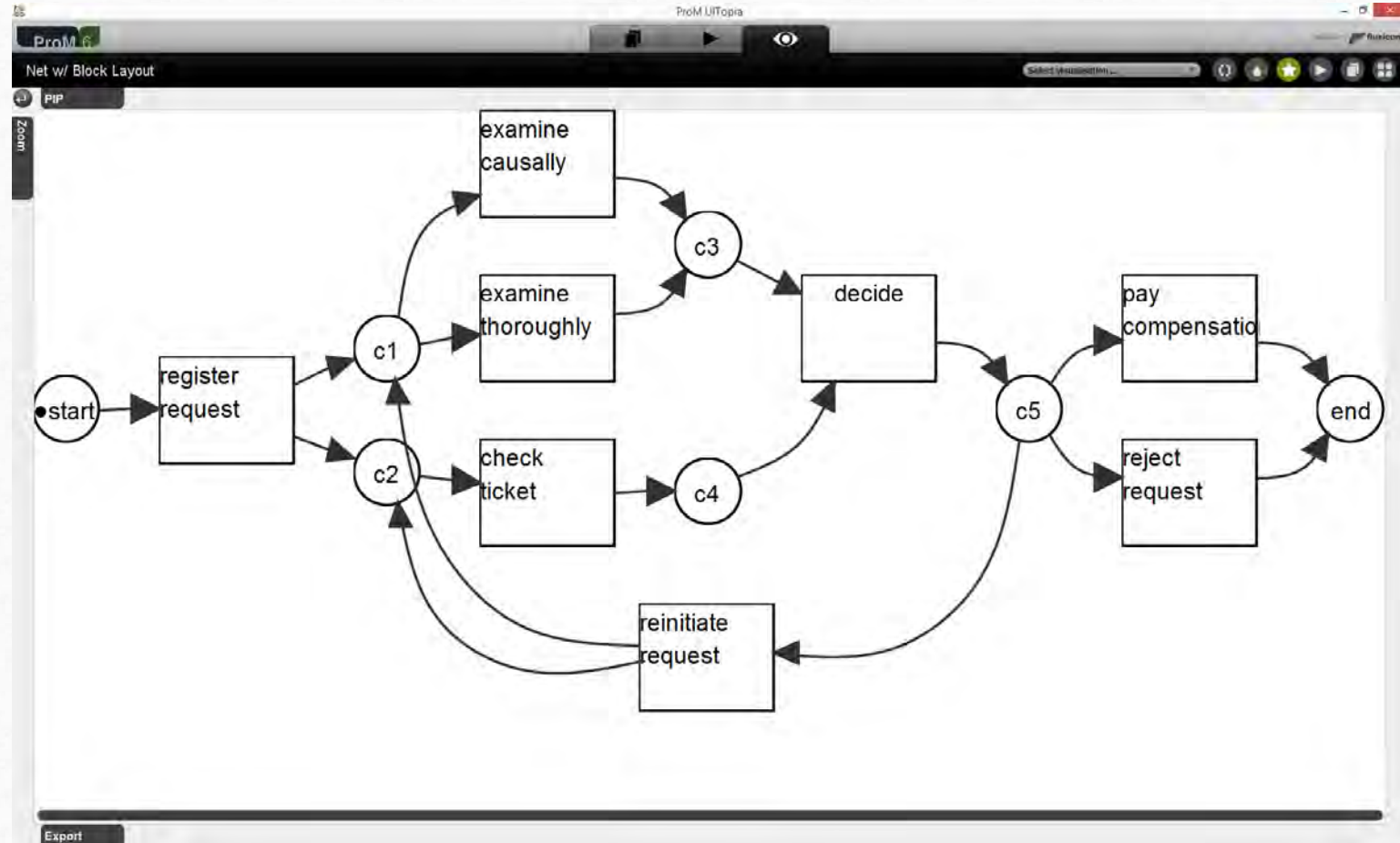
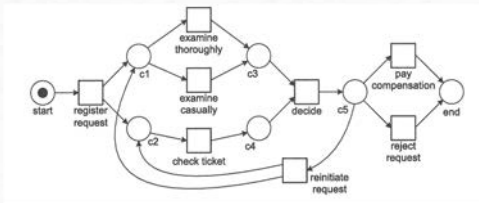
GraphViz



Dot



Block Layout

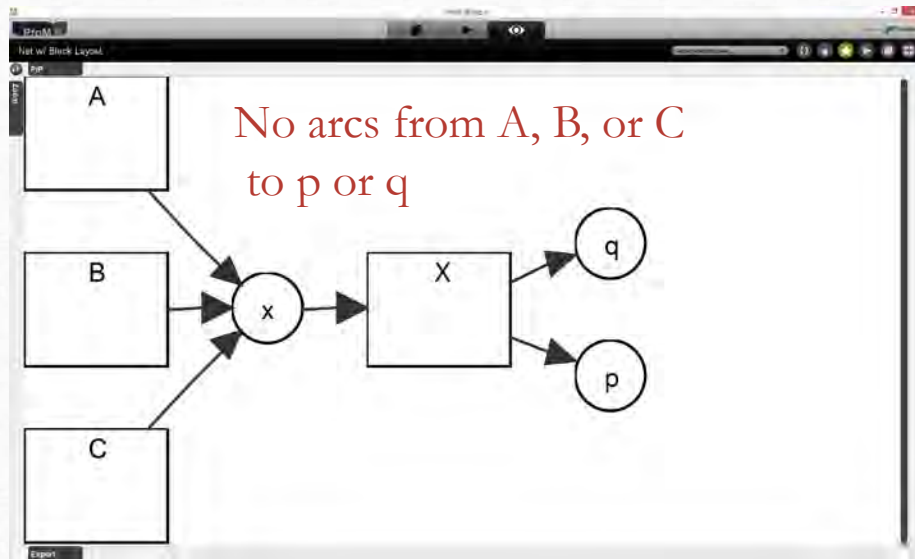


Overview

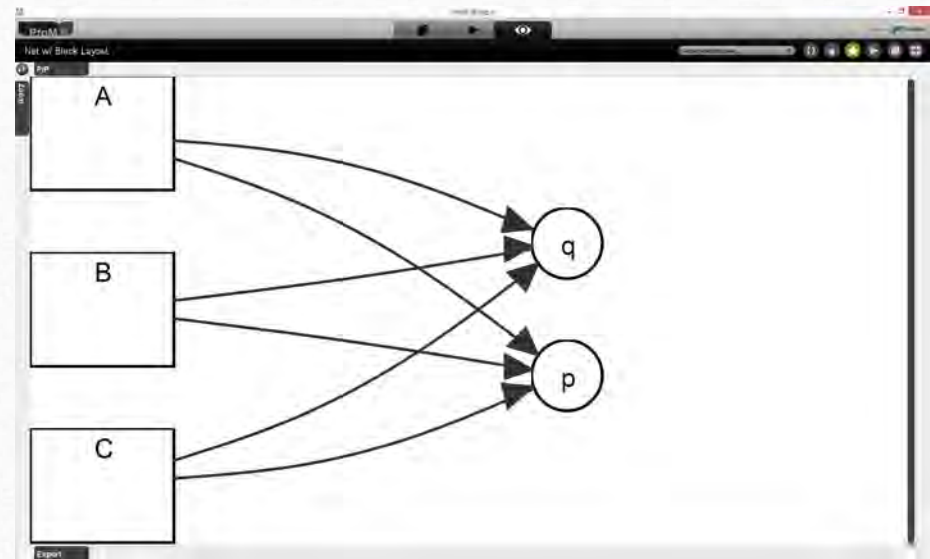
- Reduction (of well-formed free-choice nets)
 - Application
 - Sound free-choice workflow nets
 - Any workflow net
 - Demo
 - Discussions
 - Conclusions

Abstraction rule

Before

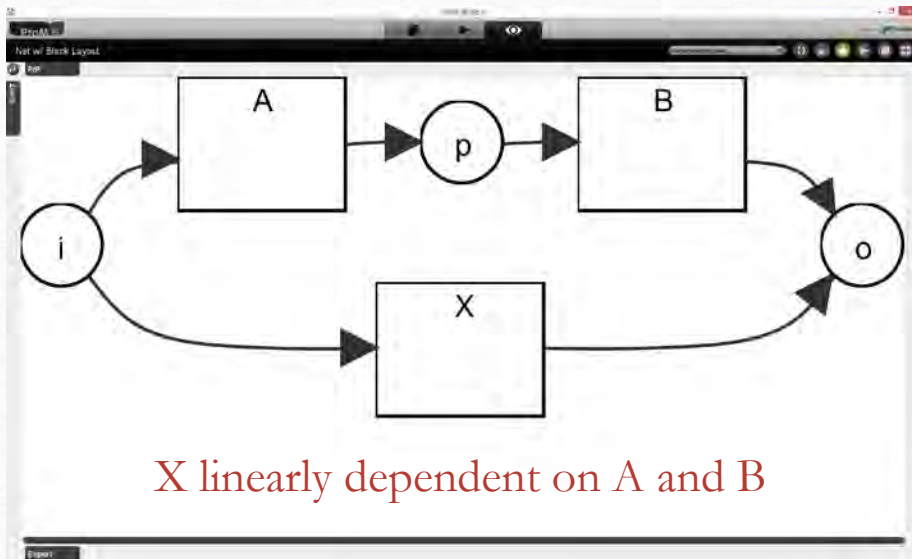


After

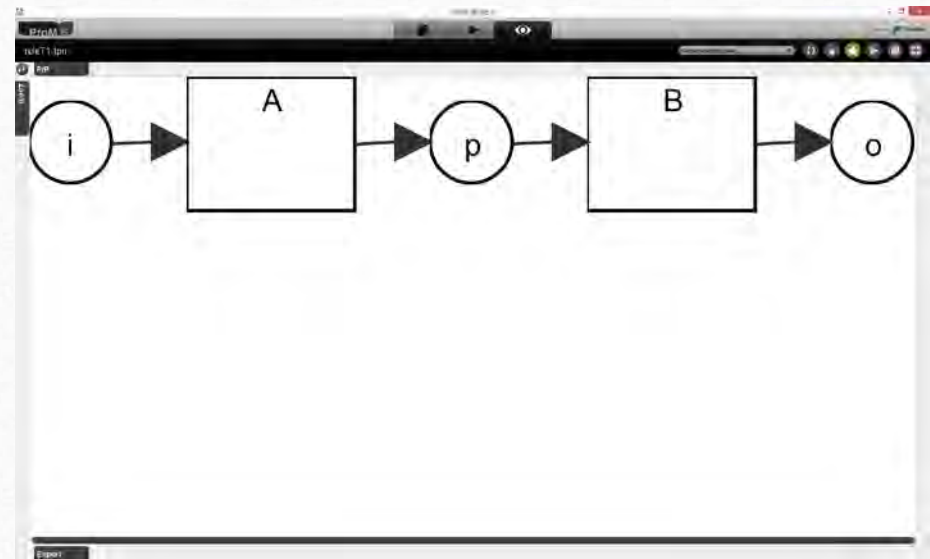


Transition (Place) rule

Before

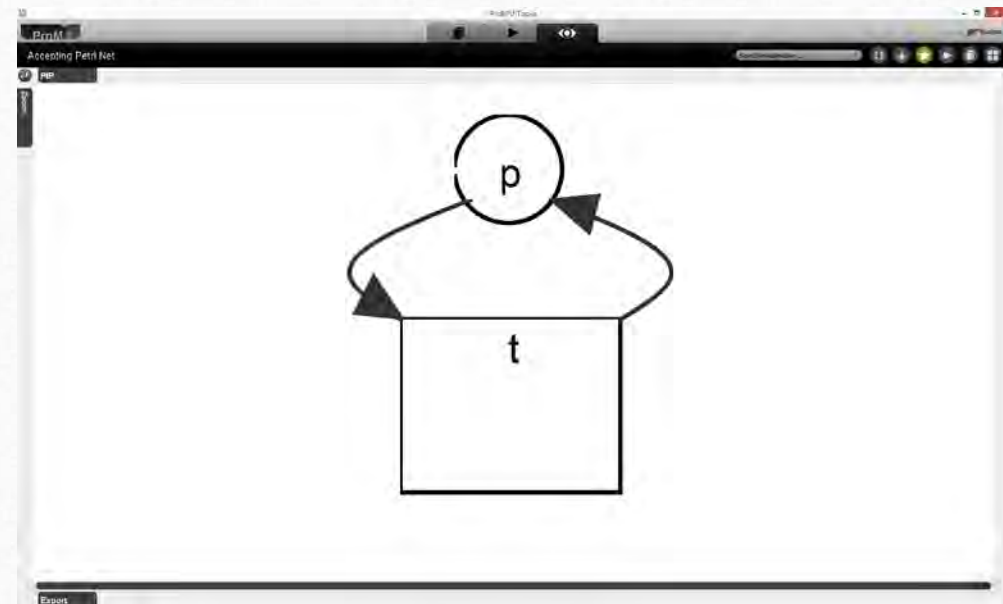


After



Reduction facts

- Can reduce a well-formed free-choice Petri net into a basic Petri net:
 - One transition
 - One place
 - Two arcs back and forth
- Sound and complete



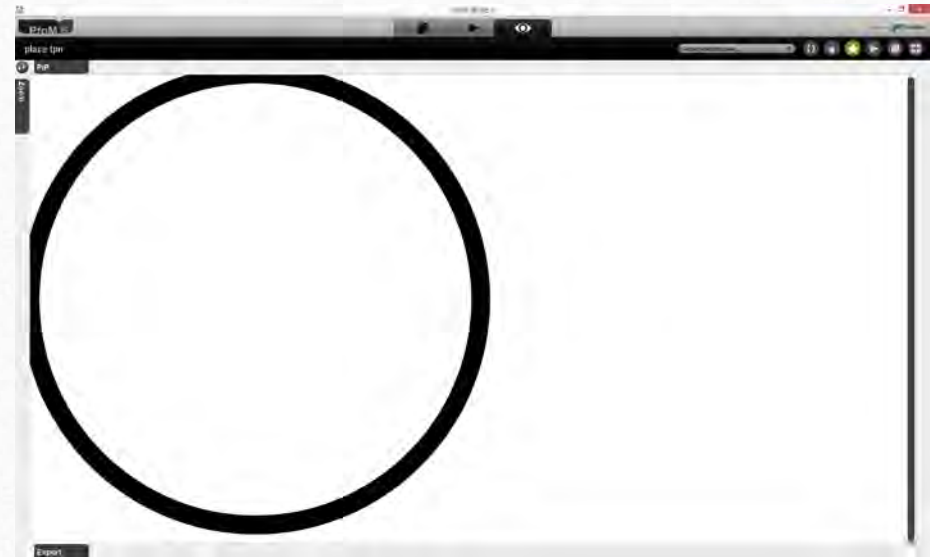
Overview

- ✓ Reduction (of well-formed free-choice nets)
- Application
 - Sound free-choice workflow nets
 - Any workflow net
- Demo
- Discussions
- Conclusions

Sound free-choice workflow nets

After short-circuiting:

- Live and bounded in initial marking, hence well-formed
- Reduction rules can be successfully applied
- Short-circuiting transition is not reduced
- Workflow net is reduced into a single place
- Which we can lay out easily:



Blocks

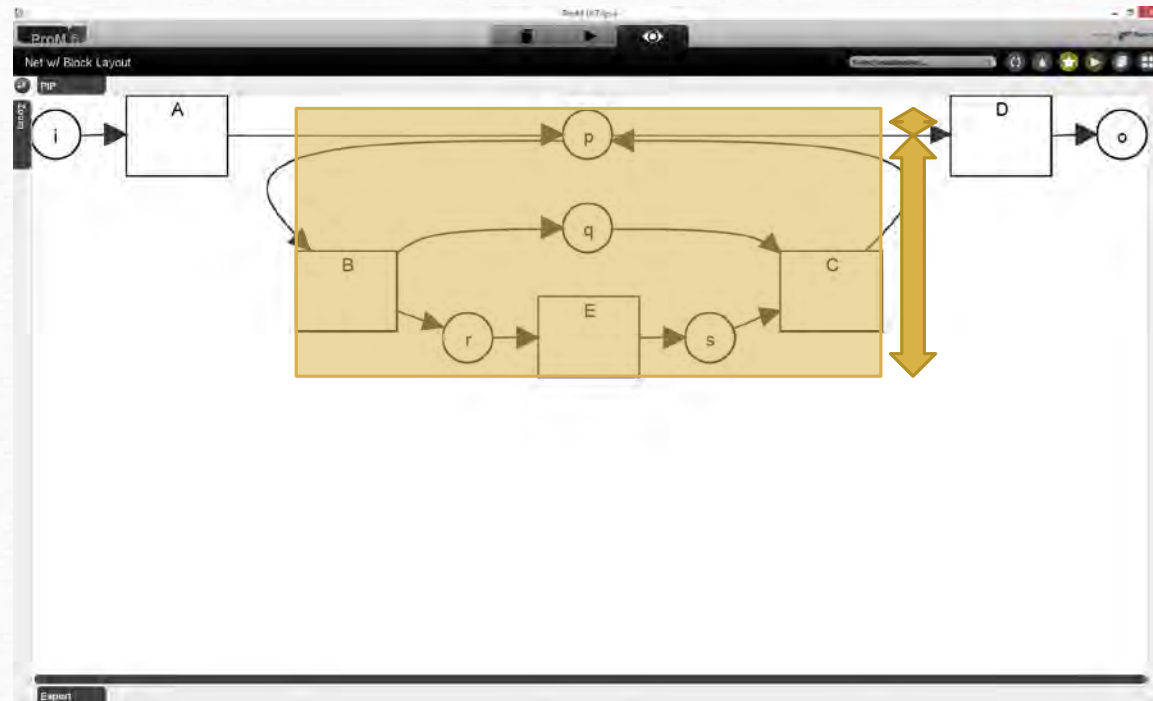
Properties

- Rectangular
- Position
- Dimension
 - Width
 - Upper and lower height

Types

- Elementary blocks
 - Node block
- Composite (hierarchical) blocks
 - Horizontal block
 - Vertical block

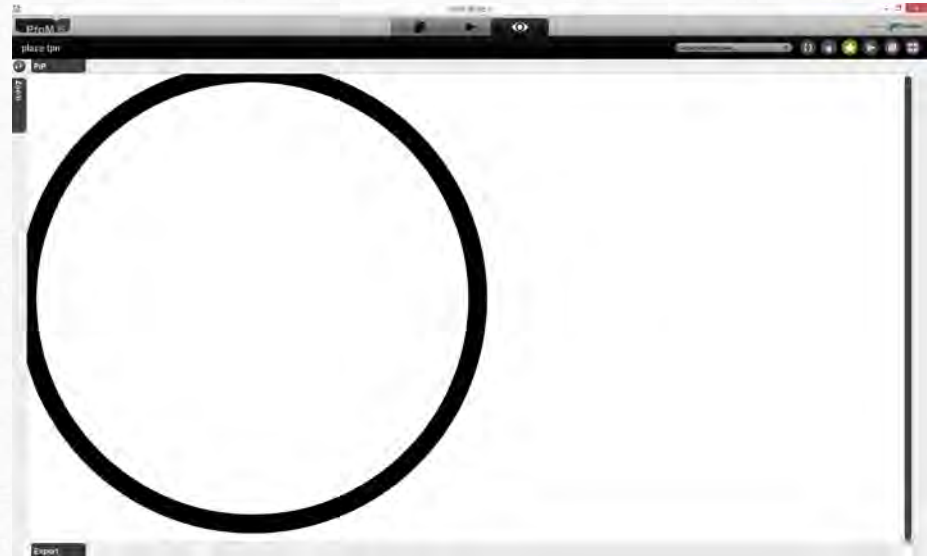
Why upper and lower height?



Two-step layout approach (1)

Bottom-up: Creating blocks

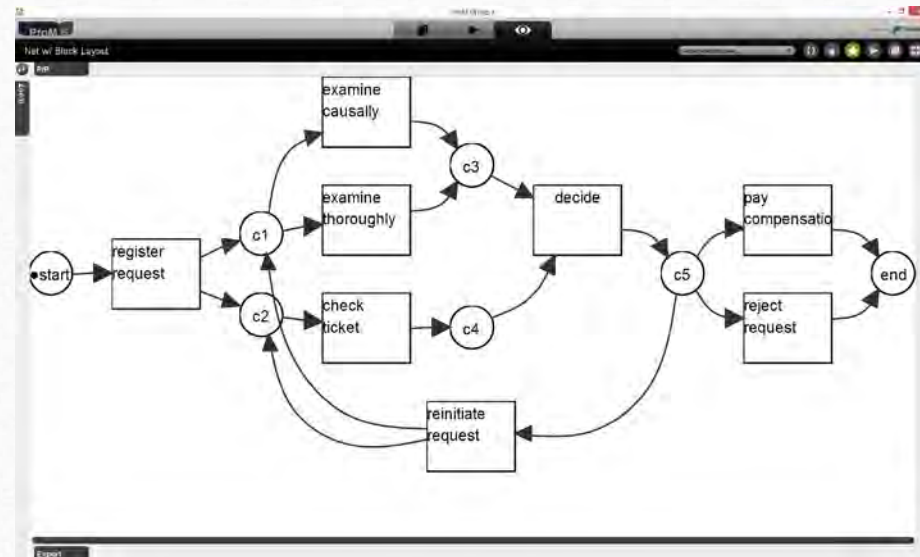
- For every node in the net, create a node block containing its initial dimension
- Reduce workflow net
- For every node in the net, maintain a current block



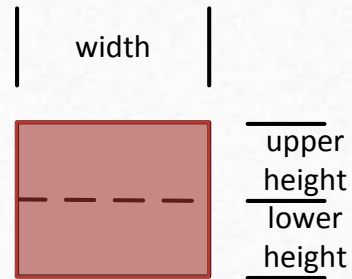
Two-step layout approach (2)

Top-down: Fixing positions

- Fix position (0, 0) for block of remaining node
- Recursively, fix position of subblocks
- Transfer position of every node block to the corresponding node



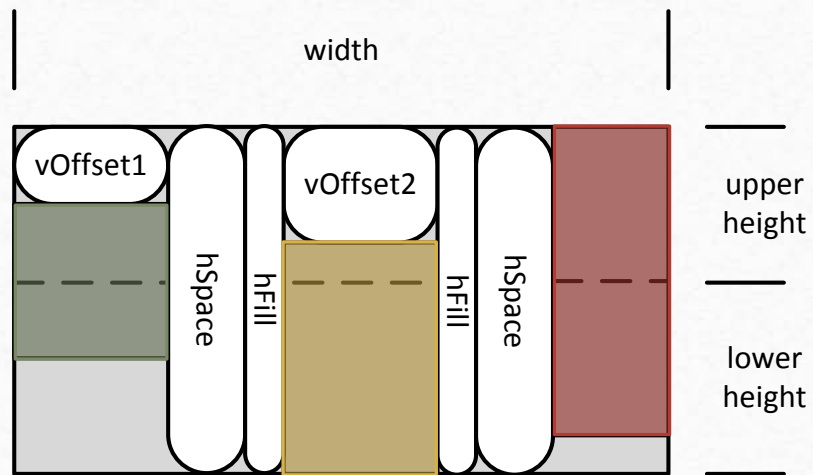
Node block



NODE BLOCK

- Contains position and dimension of a single node
- Initially, every node corresponds to a node block with proper dimension and unknown position
- Finally, all positions of the node blocks will be set.

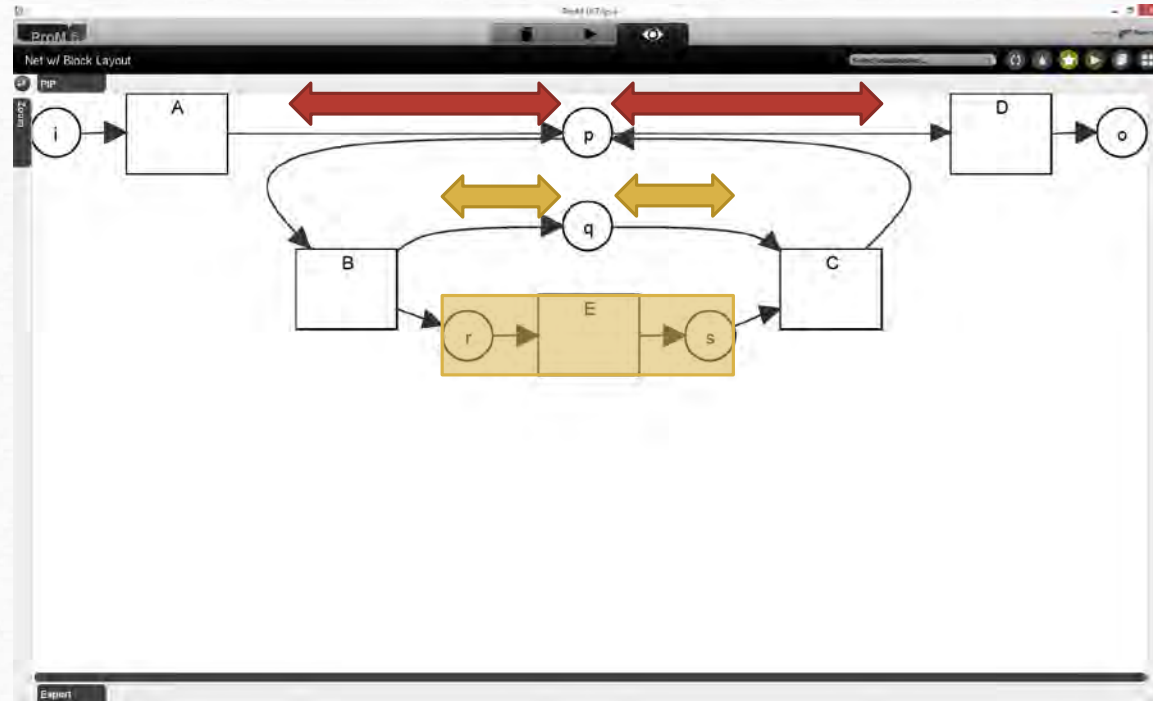
Horizontal block



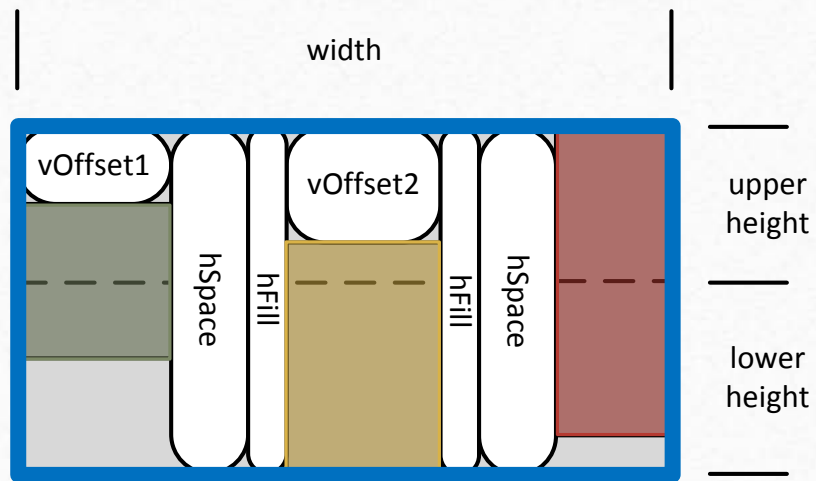
HORIZONTAL BLOCK
use hFills to widen block if needed

- Three subblocks
 - Green, yellow, and red rectangles
- Spacers
 - White rounded rectangles
 - hSpace has adjustable (manual) width
 - hFill has variable (automatic) width
 - vOffsets used to align subblocks

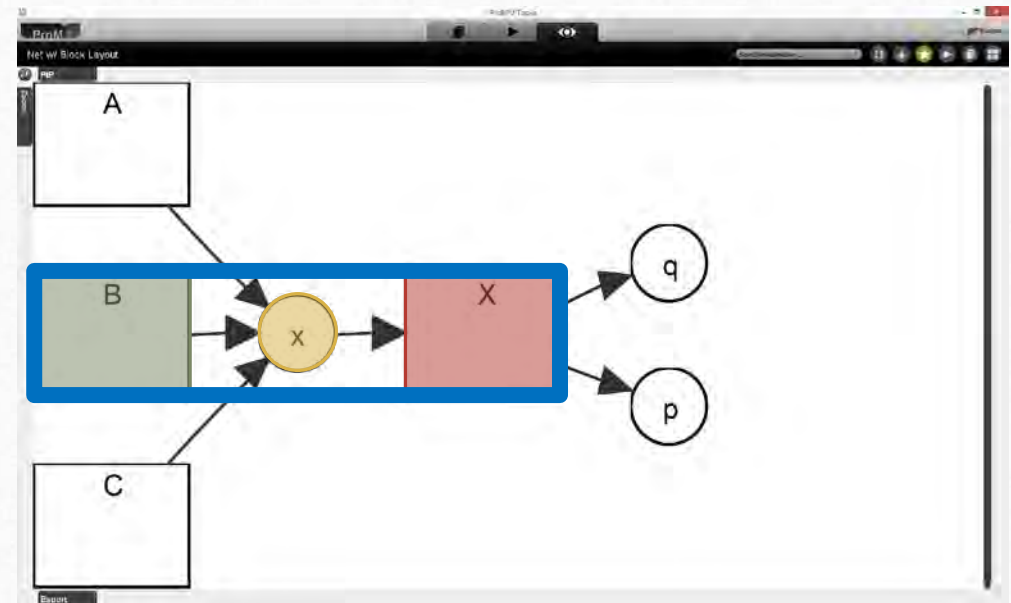
Horizontal block: Why hFill?



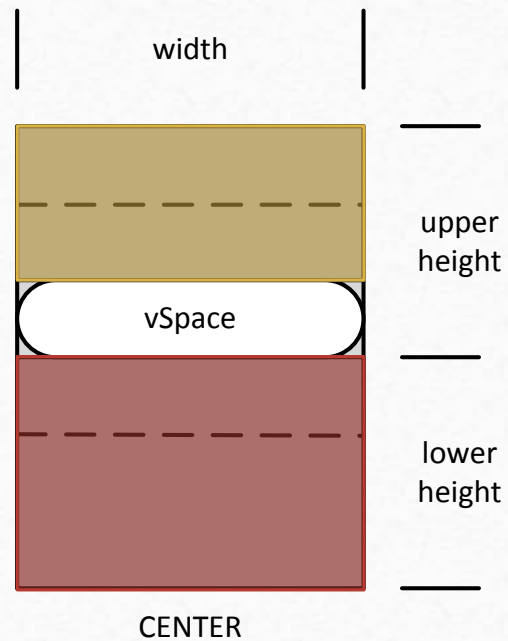
Horizontal block: Abstraction rule



HORIZONTAL BLOCK
use hFills to widen block if needed

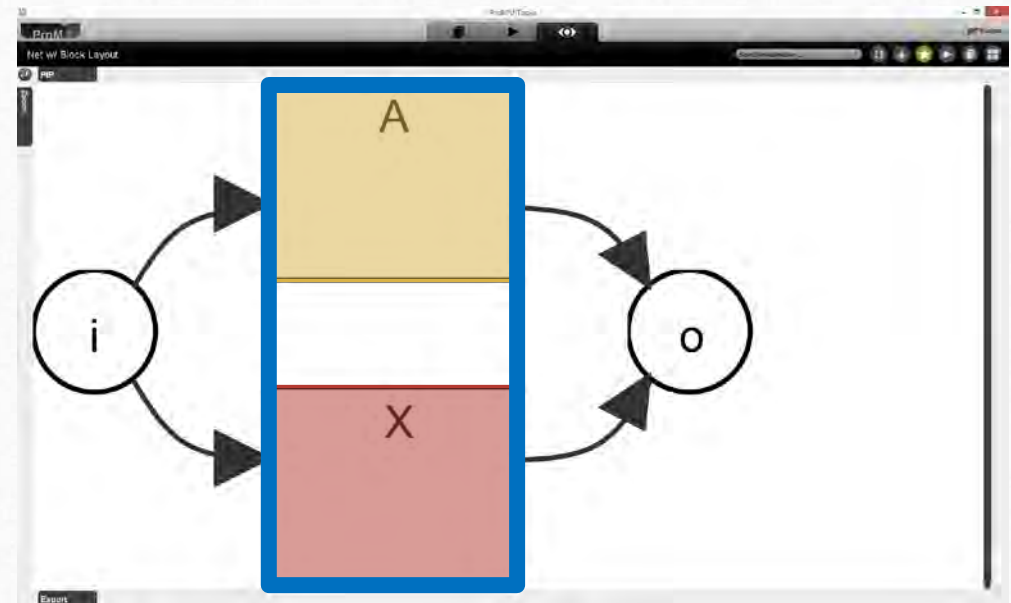
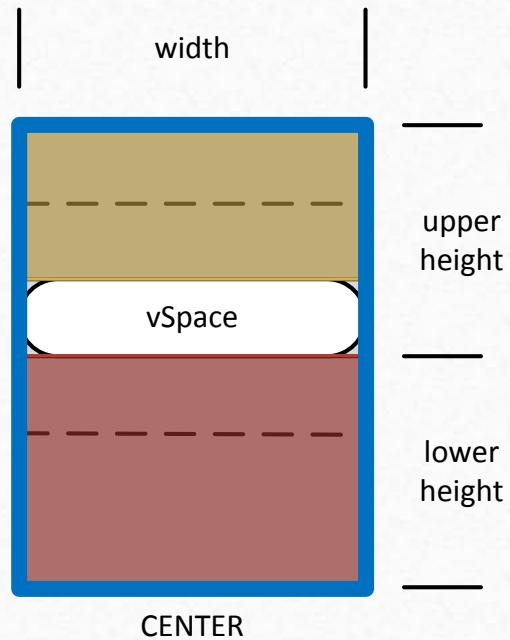


Vertical block



- Contains two subblocks
- vSpace has adjustable (manual) height
- Three settings for heights
 - Center
 - Top
 - Bottom
- Blocks widths are aligned
 - Using hFill
- Self-loops

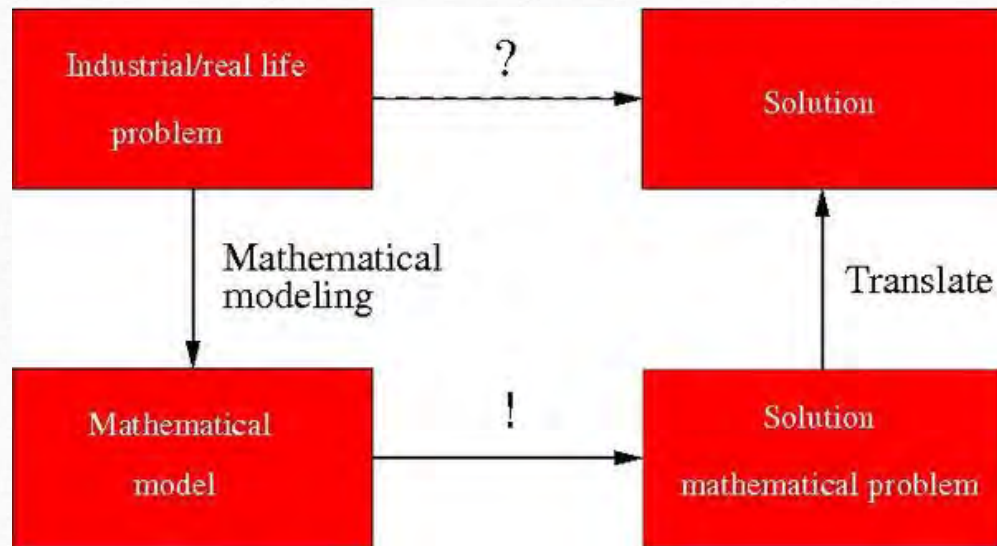
Vertical block: Transition (Place) rule



Overview

- ✓ Reduction (of well-formed free-choice nets)
- Application
 - ✓ Sound free-choice workflow nets
 - Any workflow net
- Demo
- Discussions
- Conclusions

Any workflow net (1)



- Industrial/real life problem
 - Any workflow net
- Solution
 - Layout for any workflow net
- Mathematical model
 - Sound free-choice workflow net
- Solution mathematical problem
 - Layout for sound free-choice workflow net

Any workflow net (2)

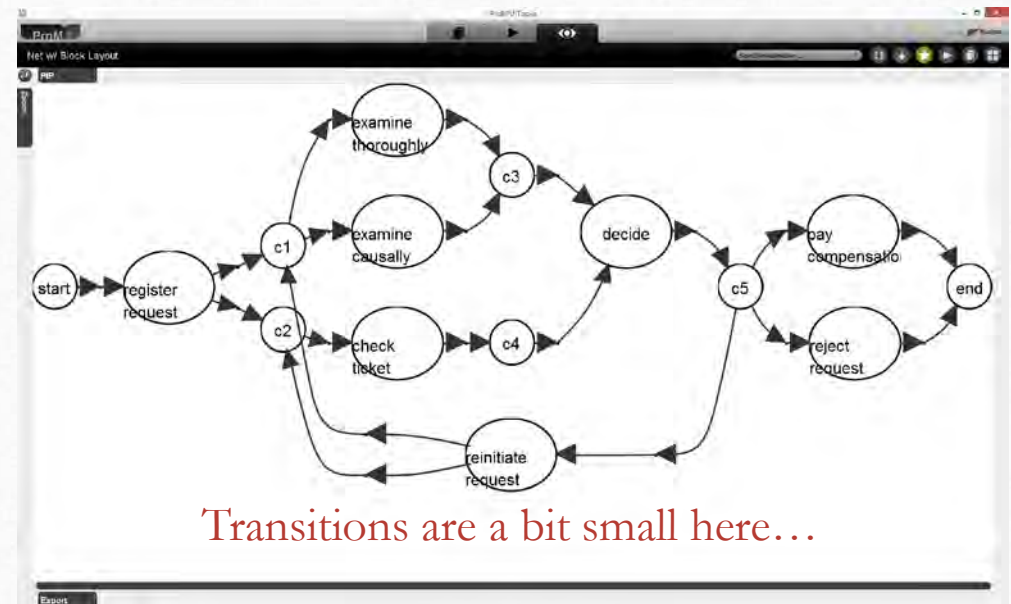
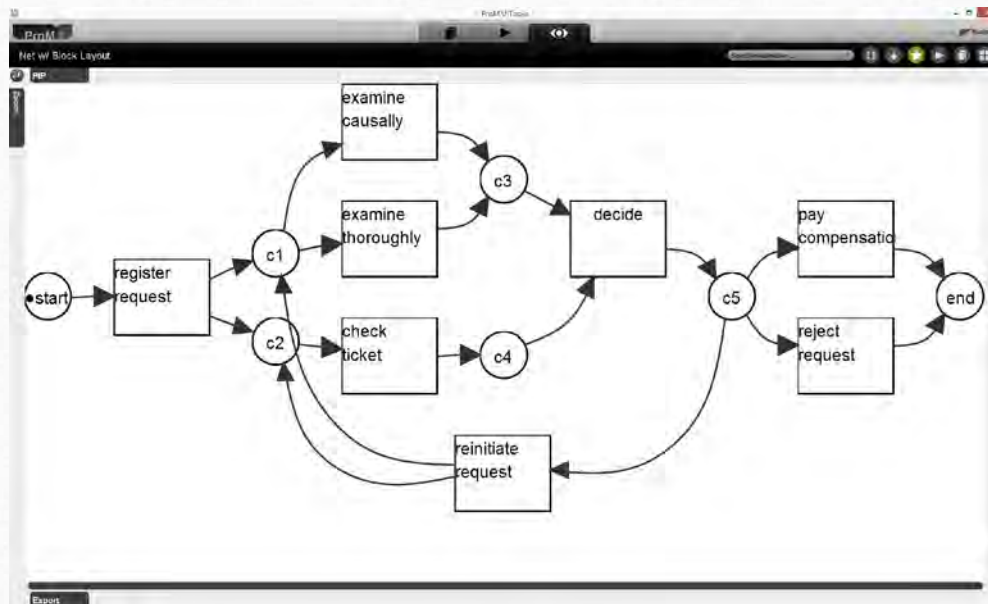
Mathematical modeling

- Create state machine workflow net with similar structure
 - Basic idea by Boudewijn
 - Replace
 - transitions by places and
 - arcs by transitions (with two arcs)
 - Is sound and free-choice

Translate

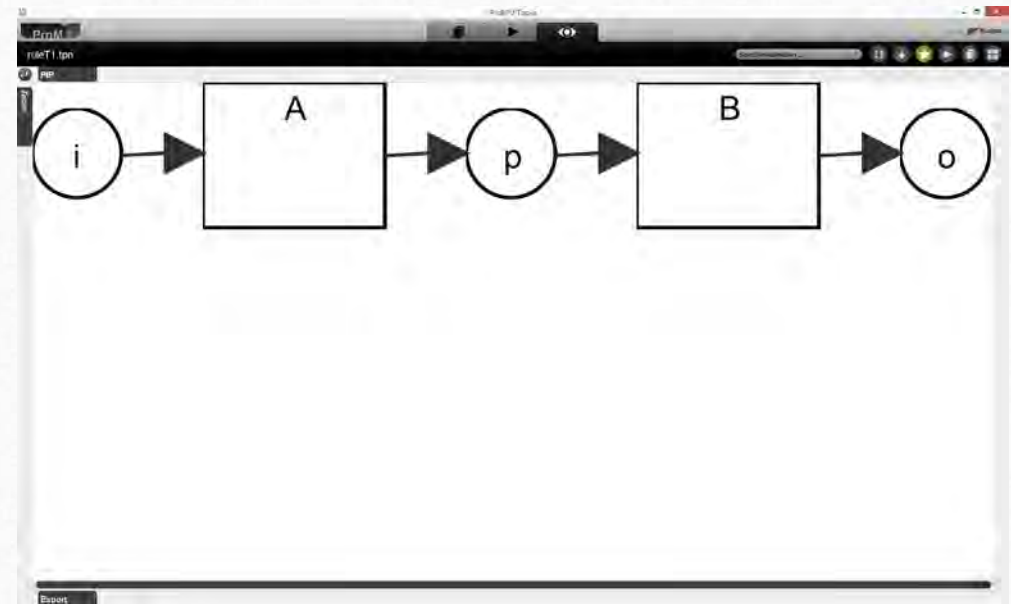
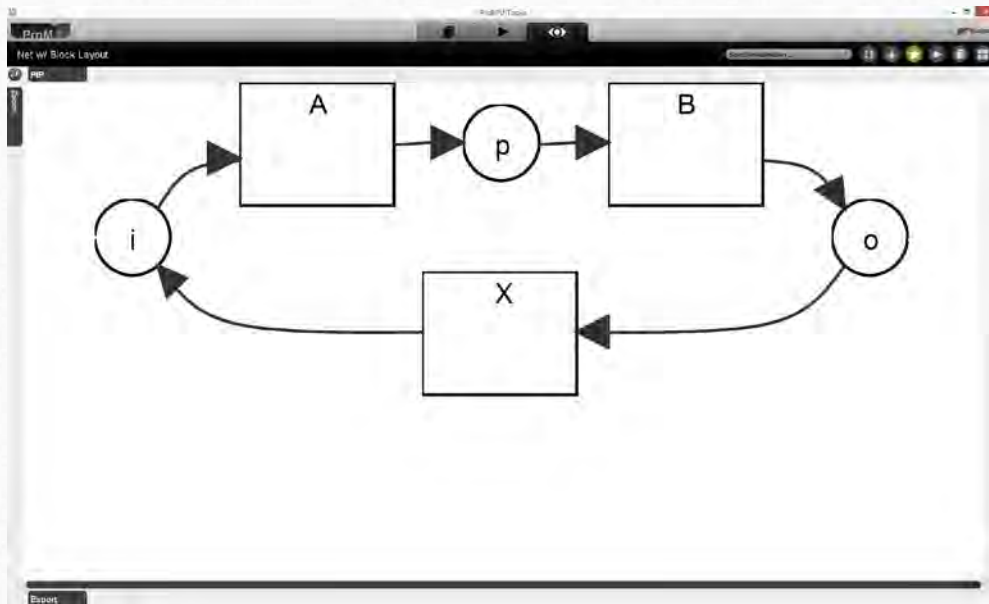
- Copy obtained positions to original workflow net
- Use transitions to obtain arc points

State machine workflow net

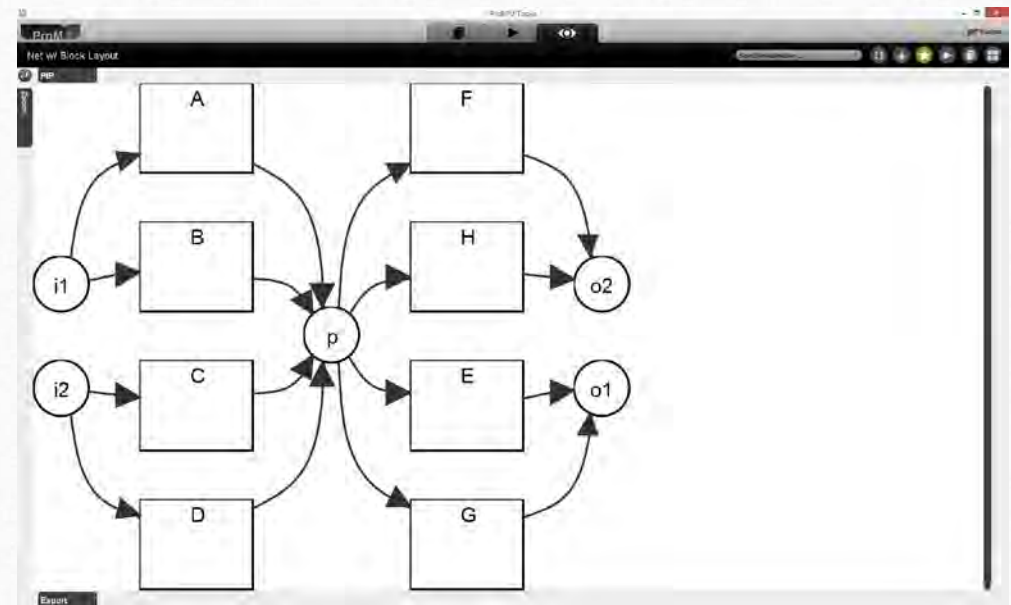
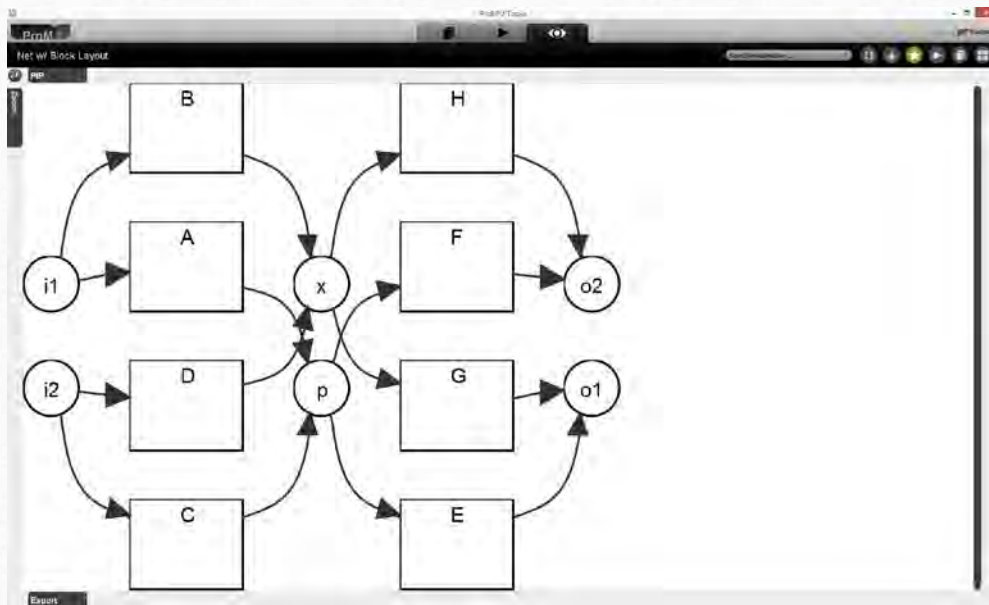


Transitions are a bit small here...

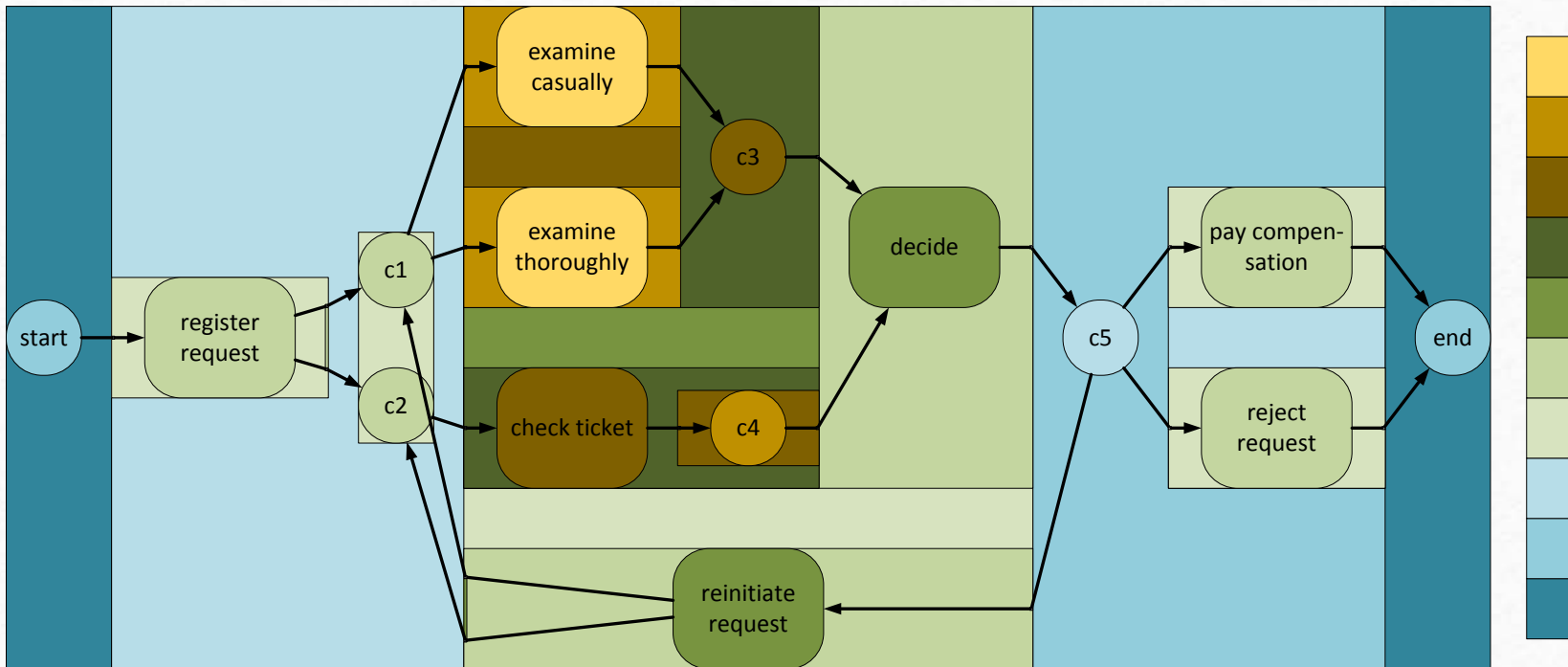
Reverse Transition rule



Alternative Place rule

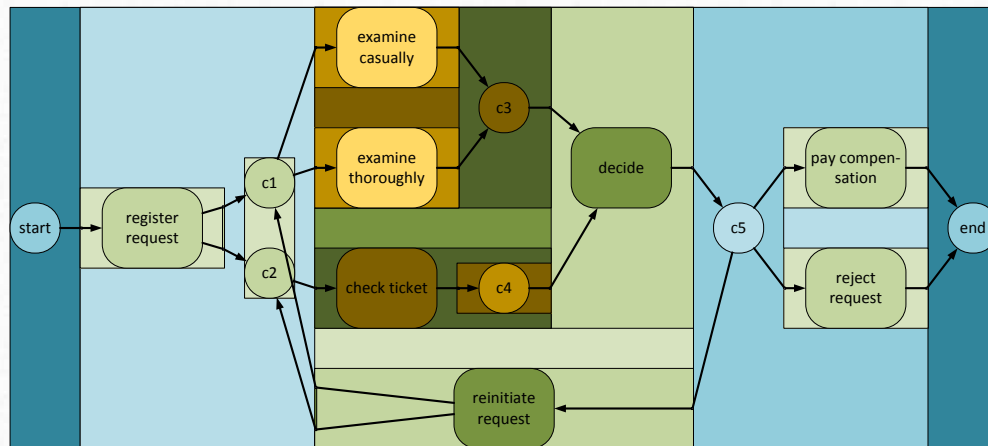


Block hierarchy for the example

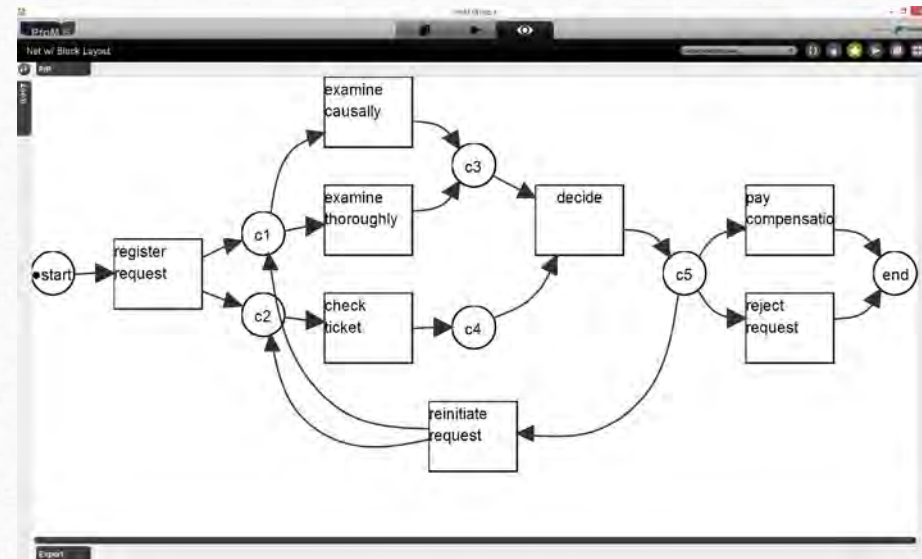


Layout for the original workflow net

Block Hierarchy



Layout

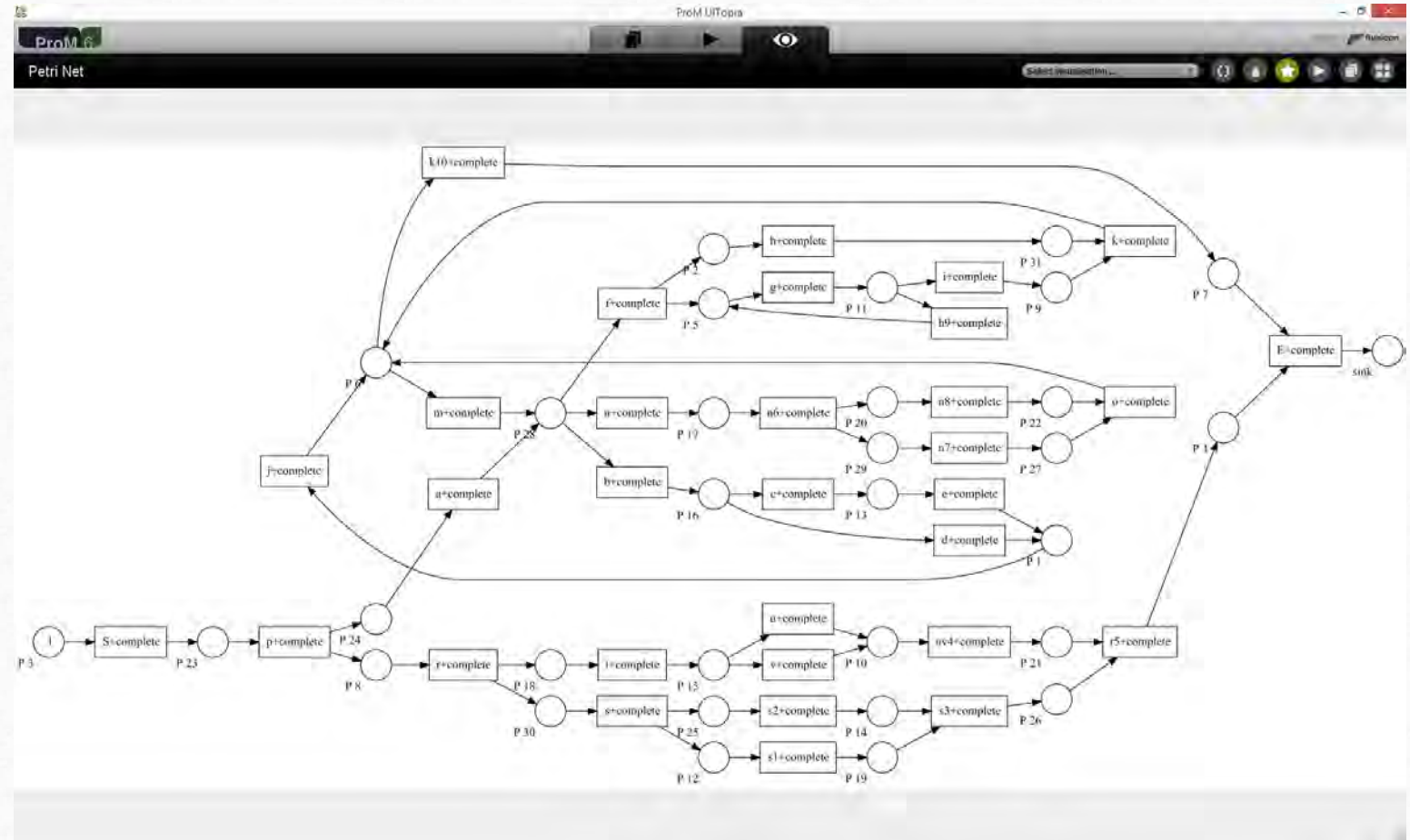


Overview

- ✓ Reduction (of well-formed free-choice nets)
- ✓ Application
 - ✓ Sound free-choice workflow nets
 - ✓ Any workflow net
- Demo
- Discussions
- Conclusions

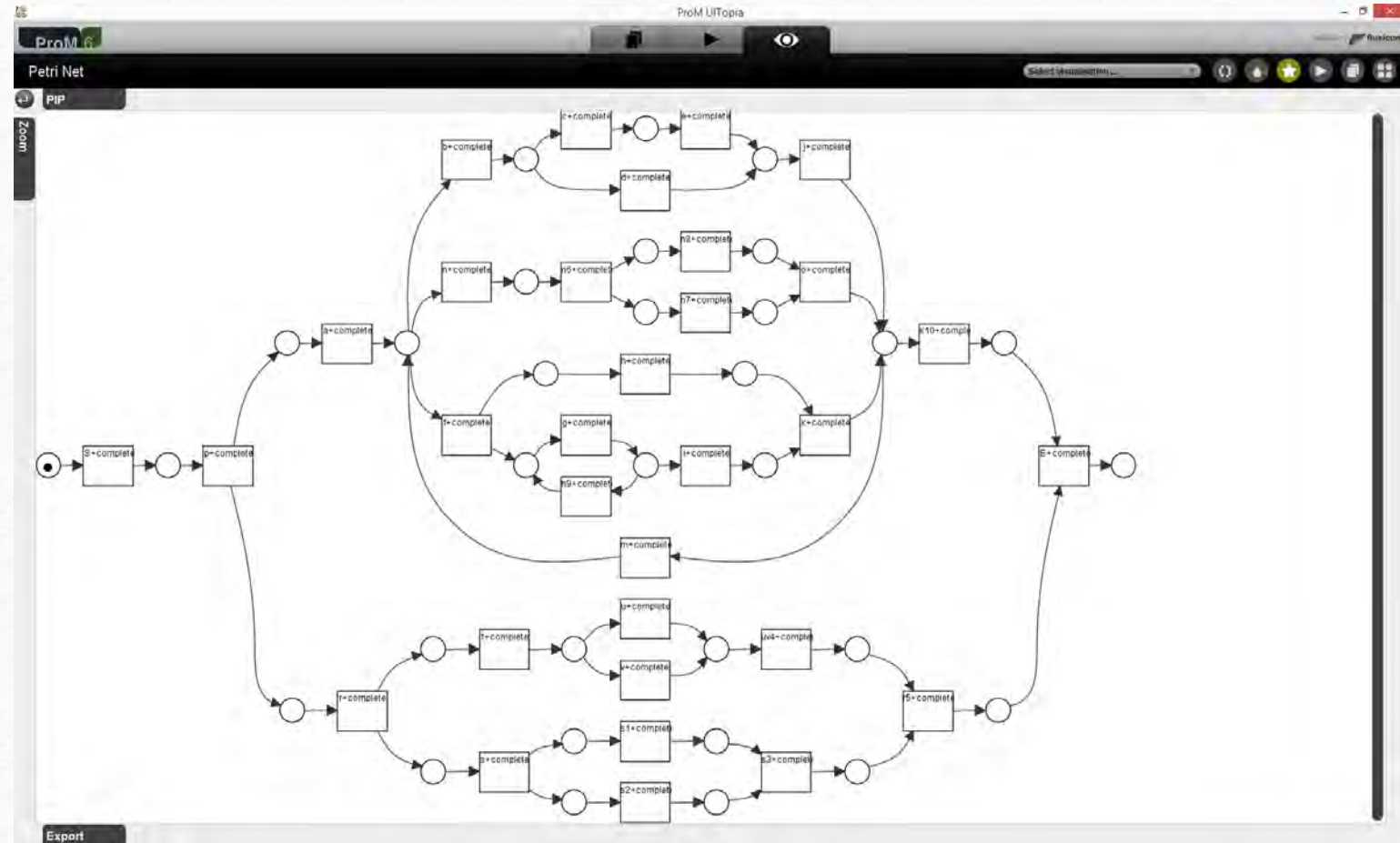
a32

Dot



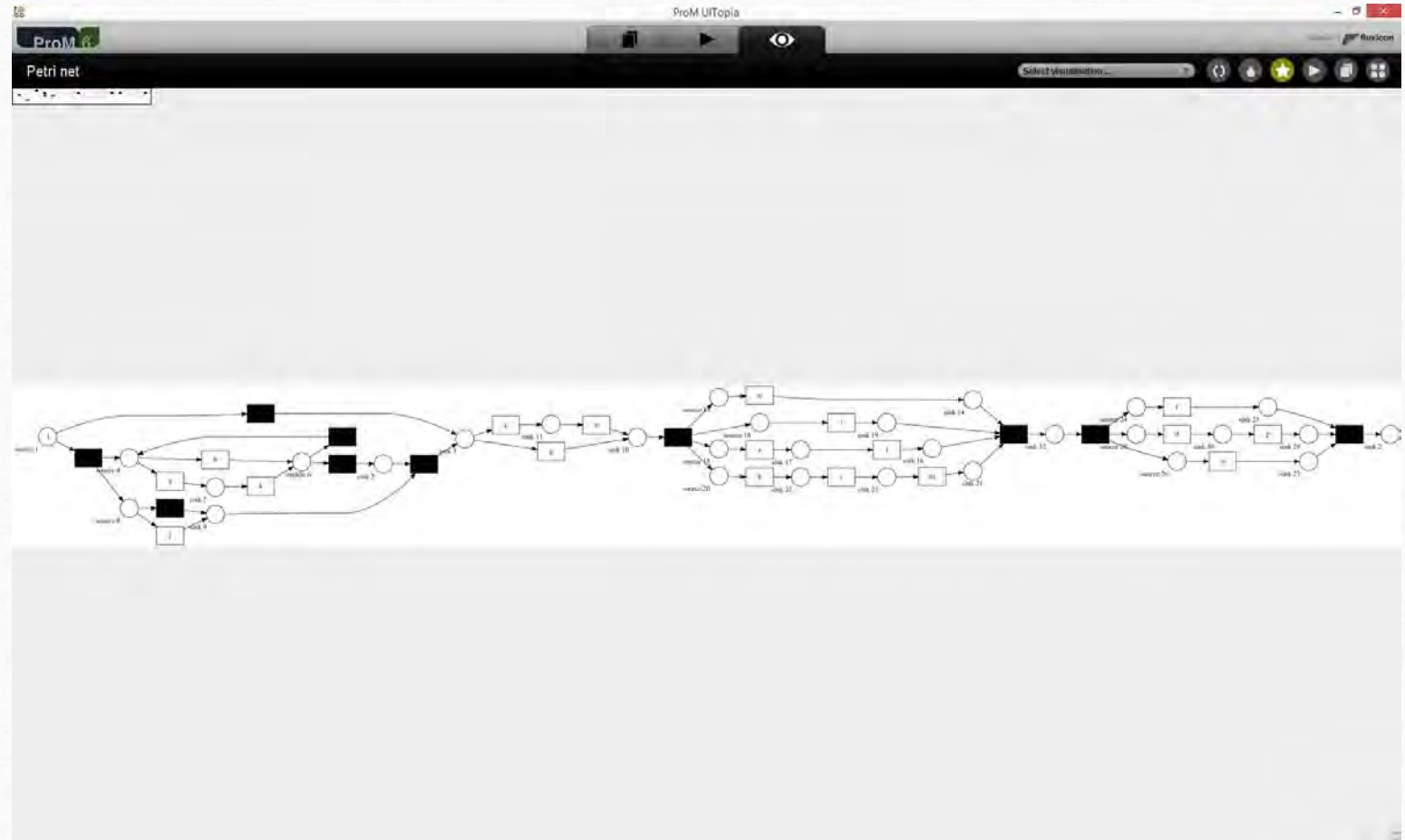
a32

Block Layout



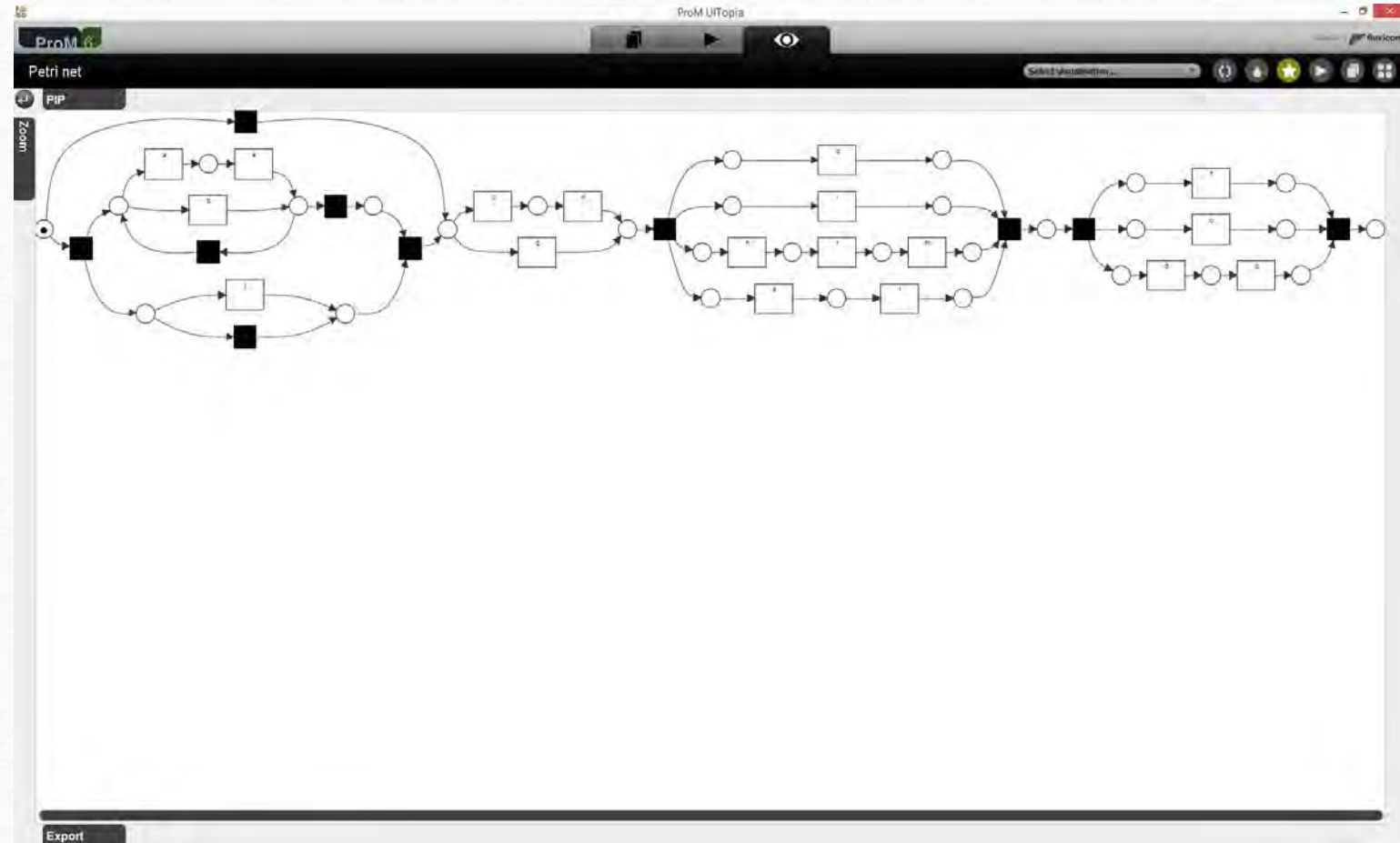
Inductive
Miner
(on Contest
2016 training
log 1)

Dot



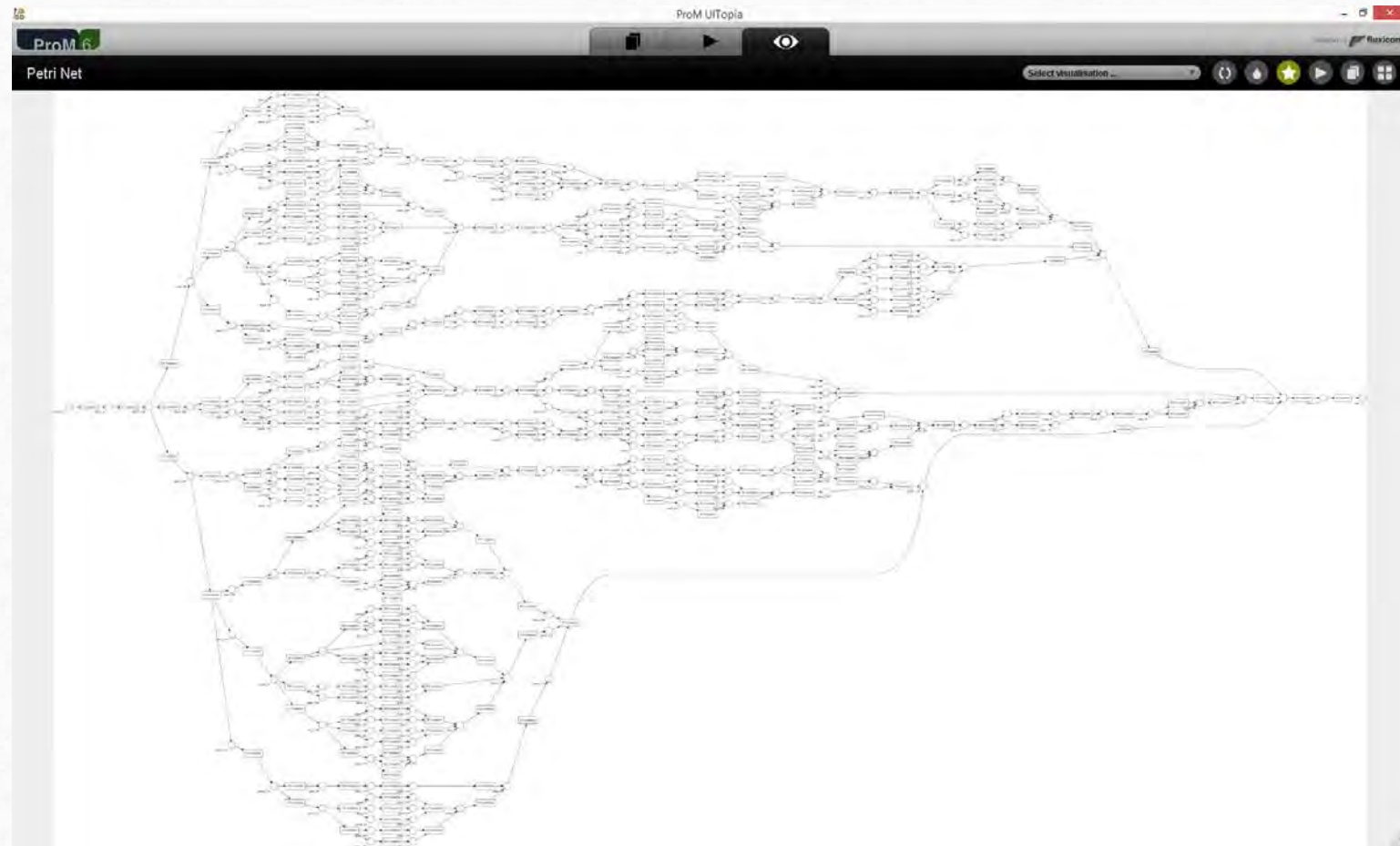
Inductive
Miner
(on Contest
2016 training
log 1)

Block Layout



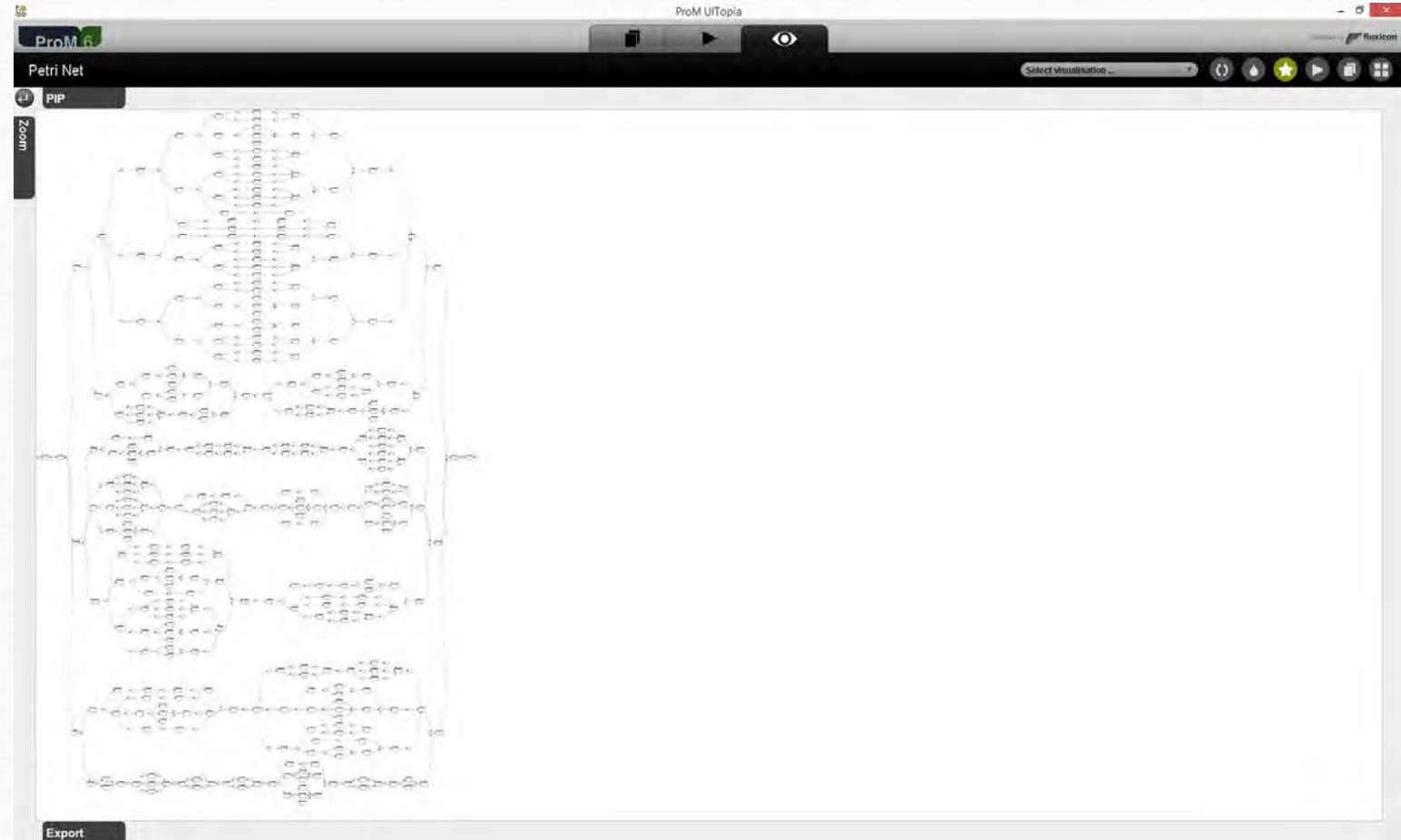
BPM 2013
data set, model
prAm6
(Jorge Munoz)

Dot



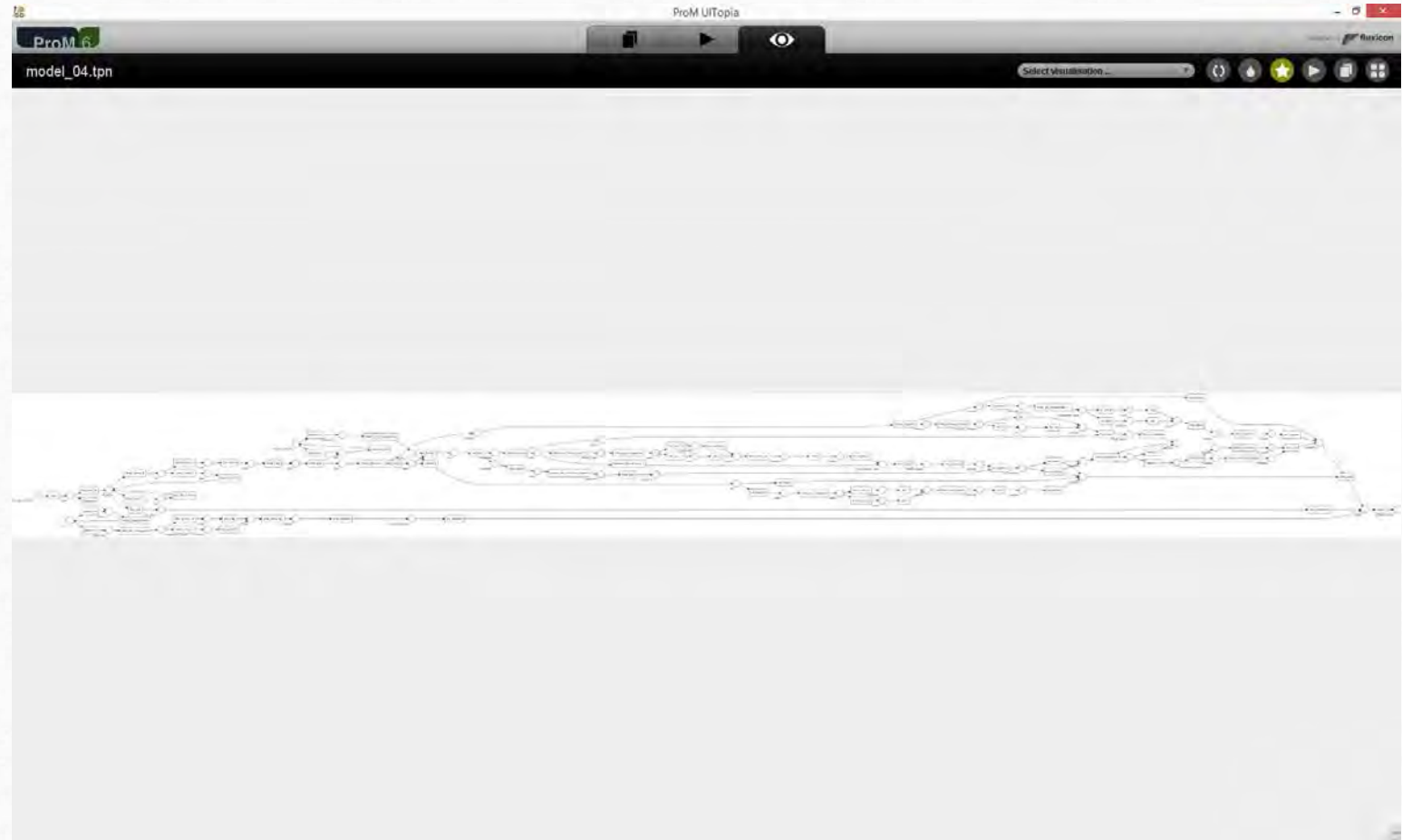
BPM 2013
data set, model
prAm6
(Jorge Munoz)

Block Layout



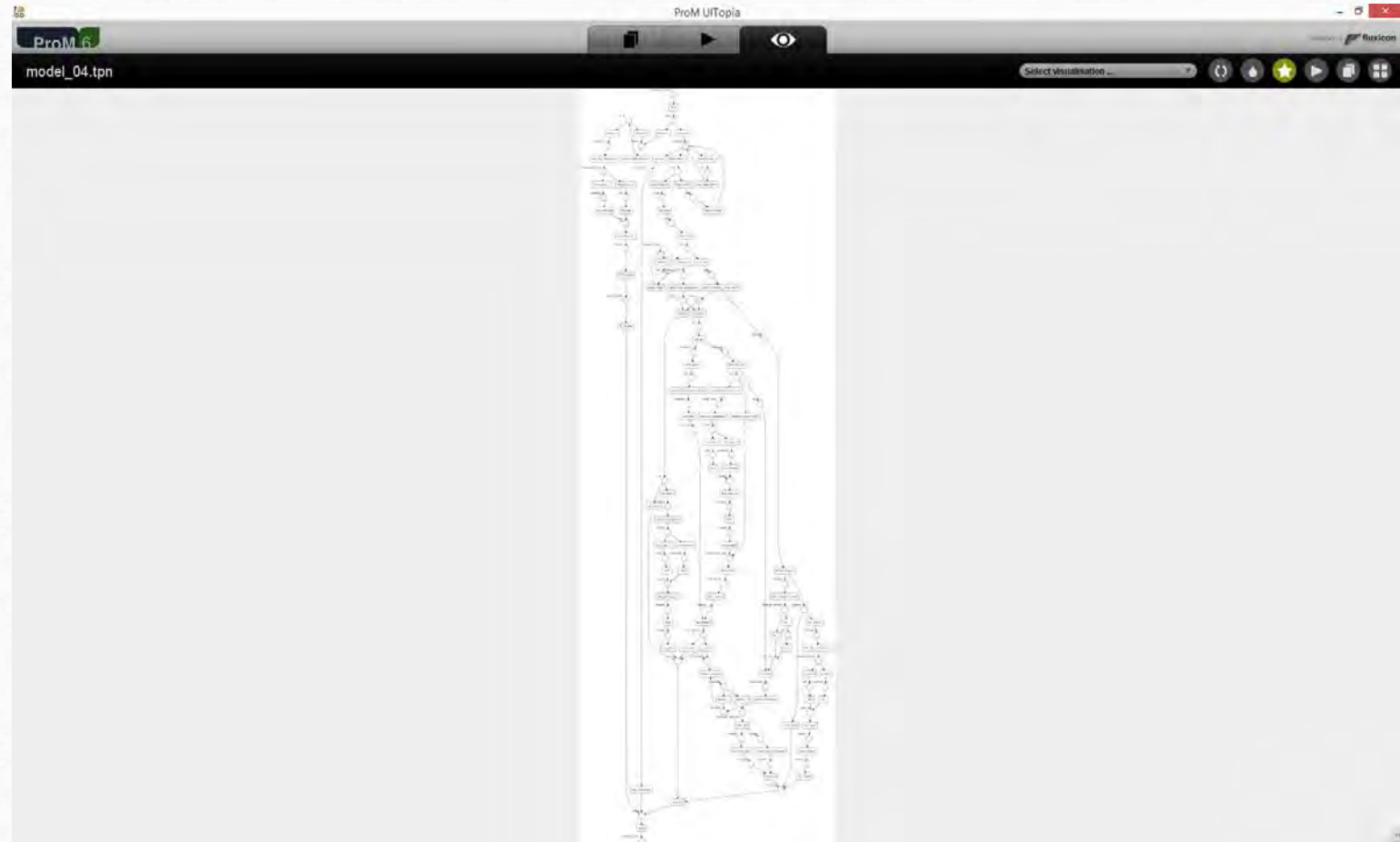
1R420,
model_04

Dot



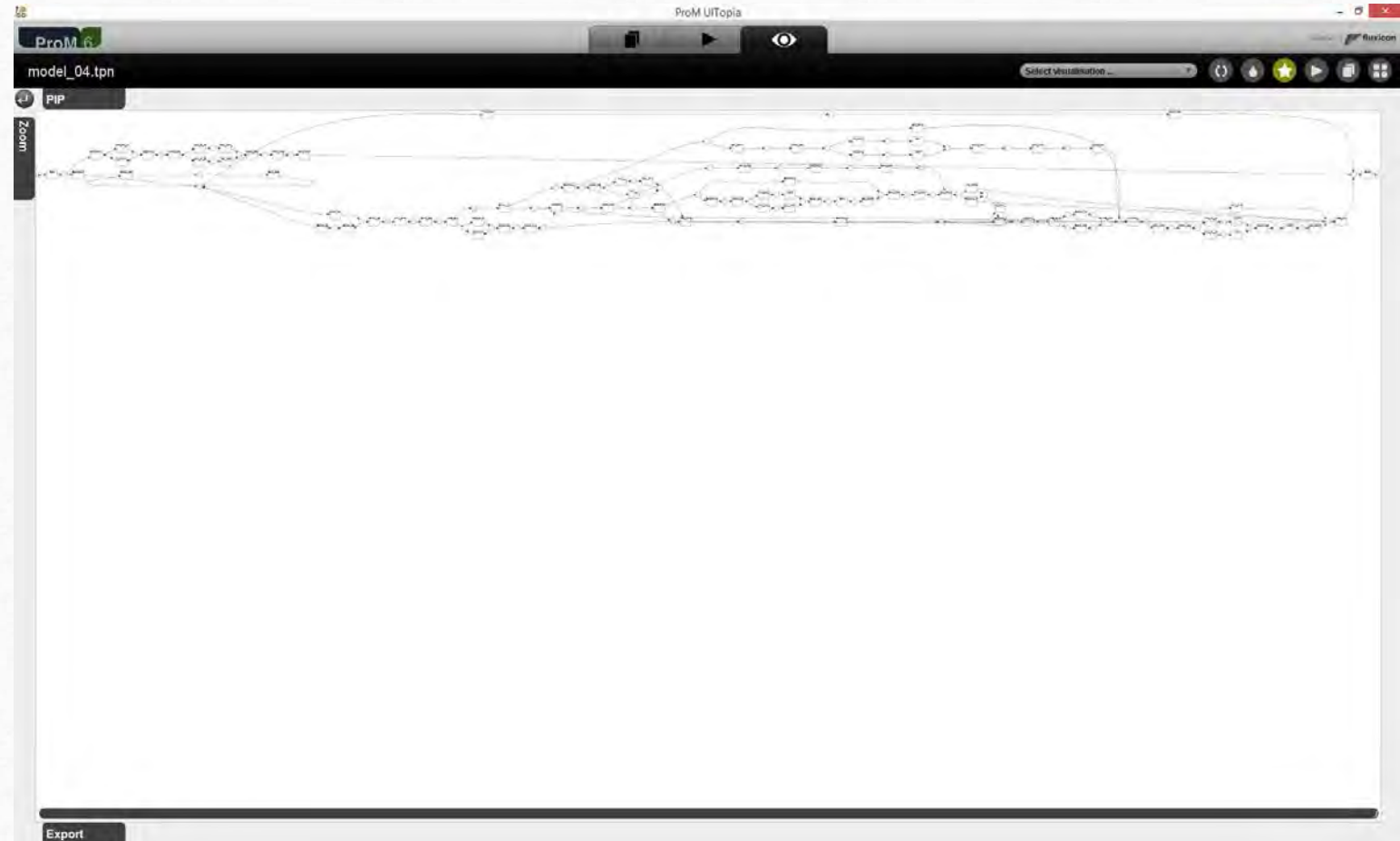
1R420,
model_04

Dot, top-down



1R420, model_04

Block Layout

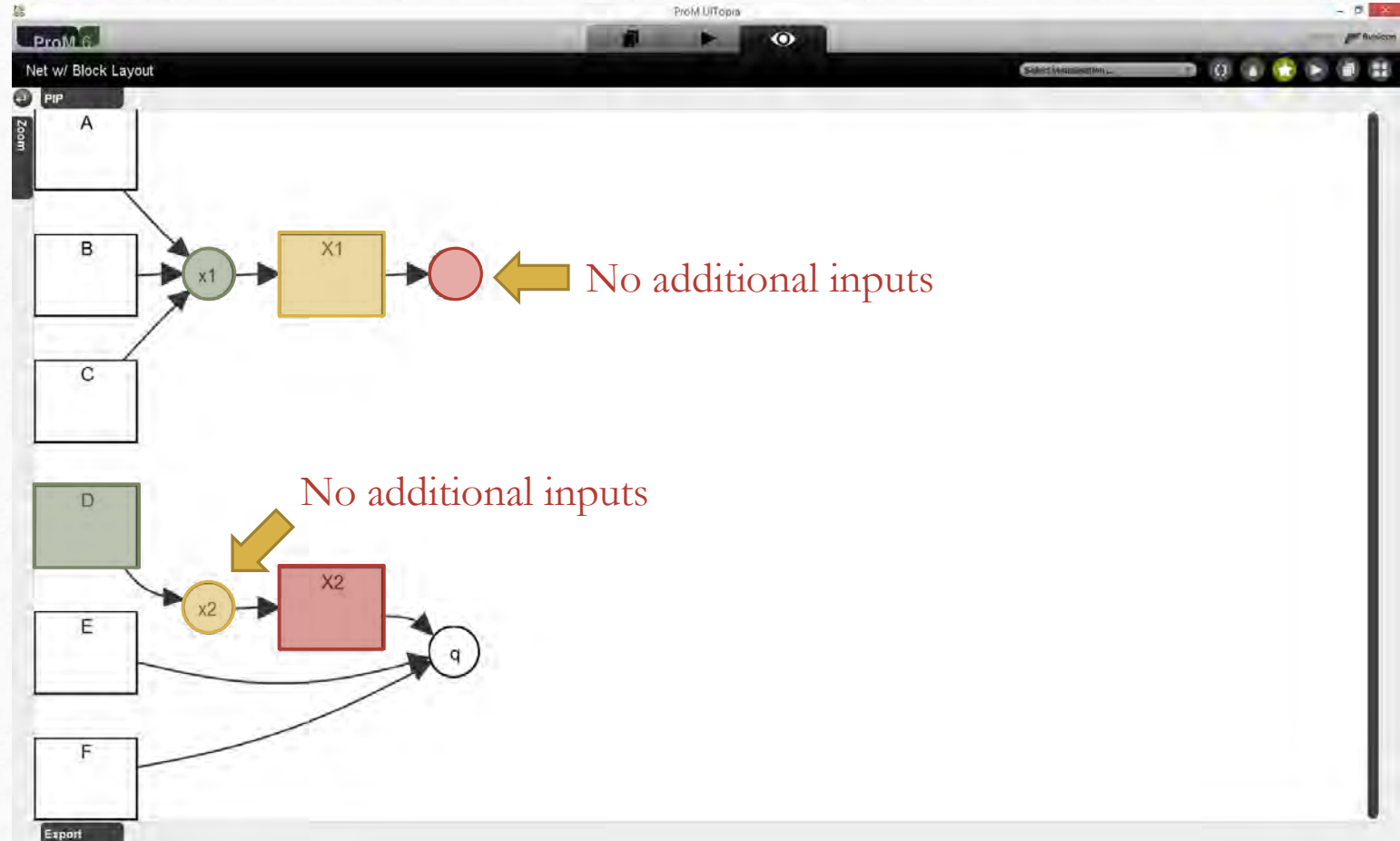


Overview

- ✓ Reduction (of well-formed free-choice nets)
- ✓ Application
 - ✓ Sound free-choice workflow nets
 - ✓ Any workflow net
- ✓ Demo
- Discussions
- Conclusions

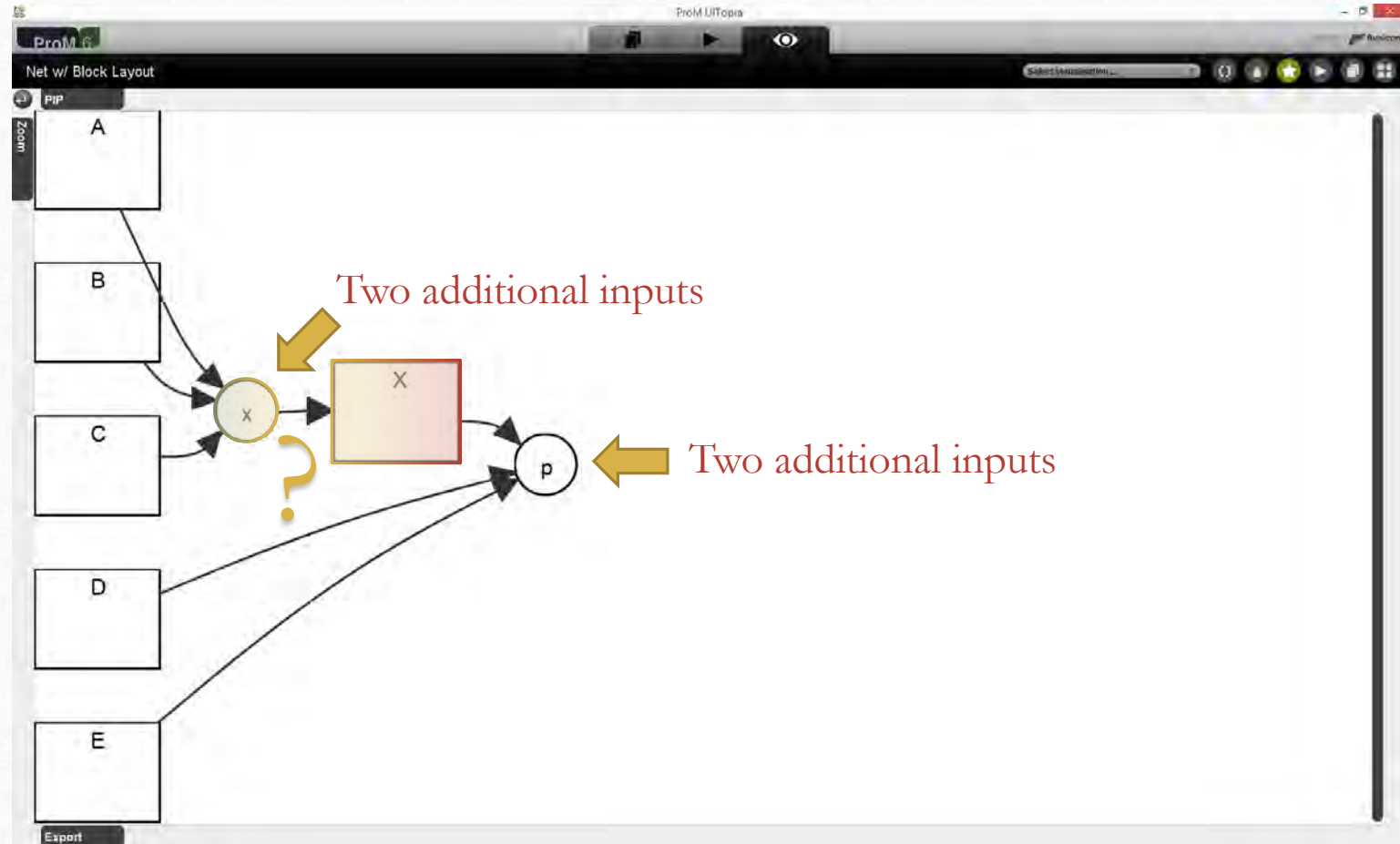
Abstraction rule

Easy



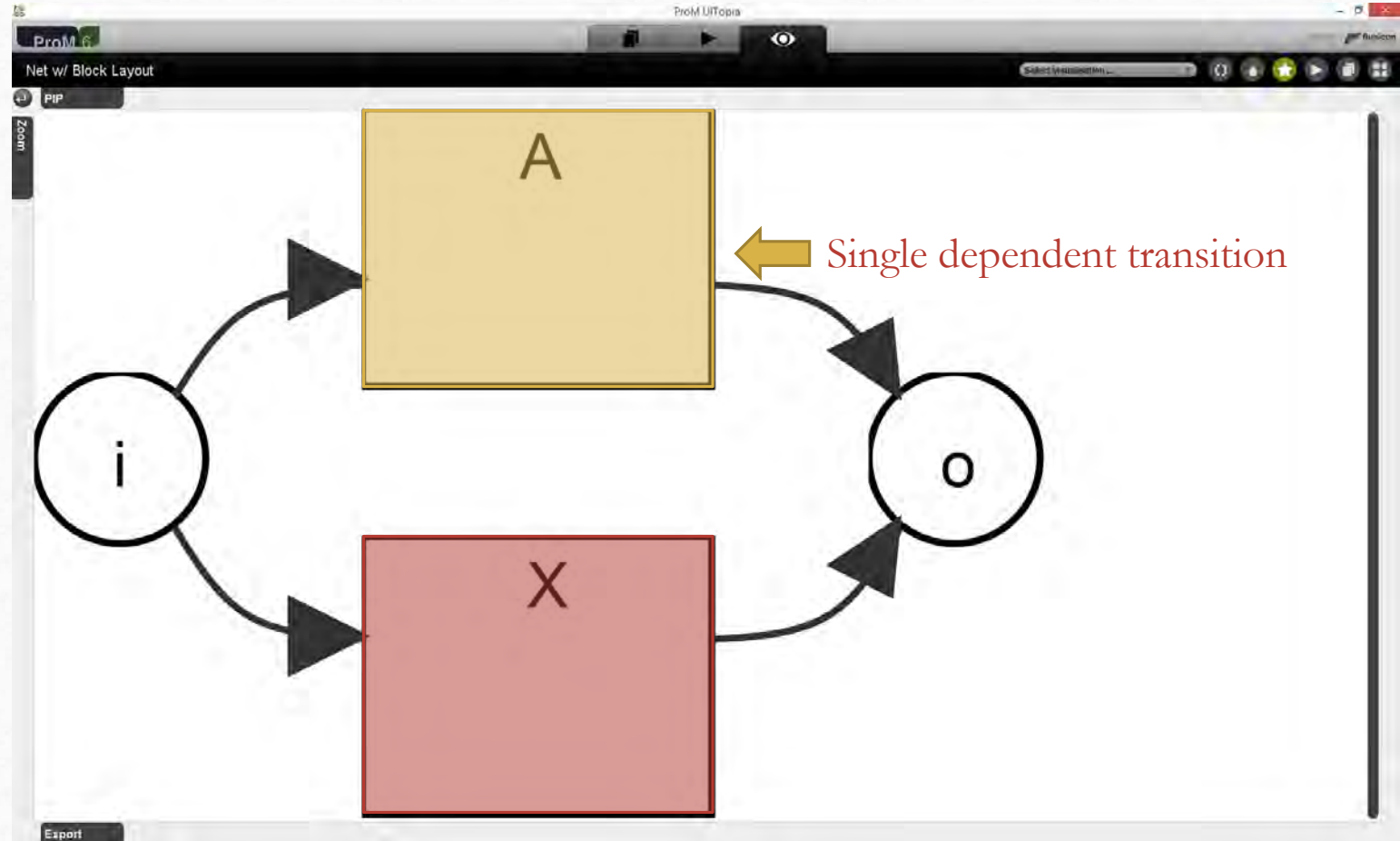
Abstraction rule

Hard
Prefer the place?



(Reverse) Transition rule

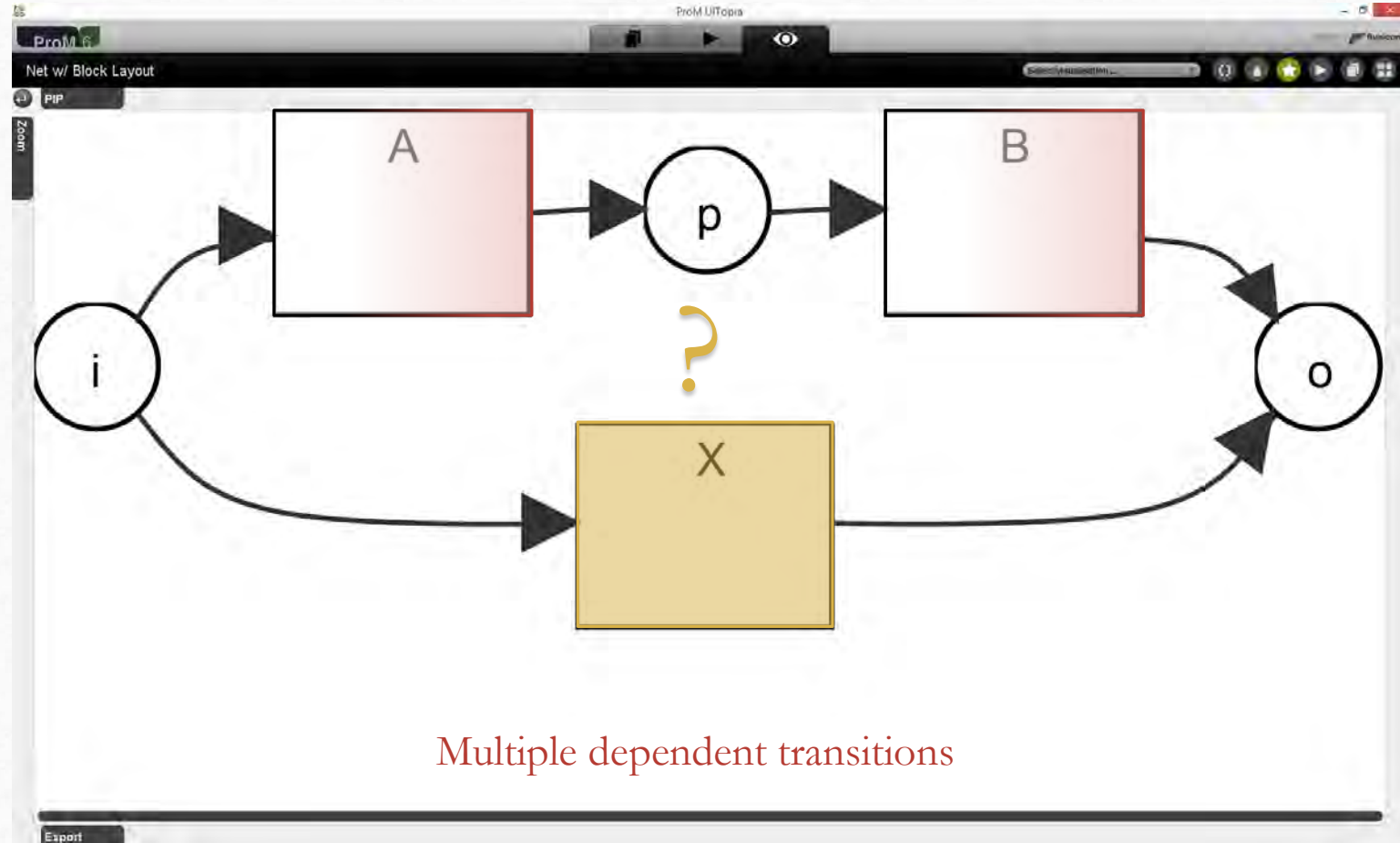
Easy



(Reverse) Transition rule

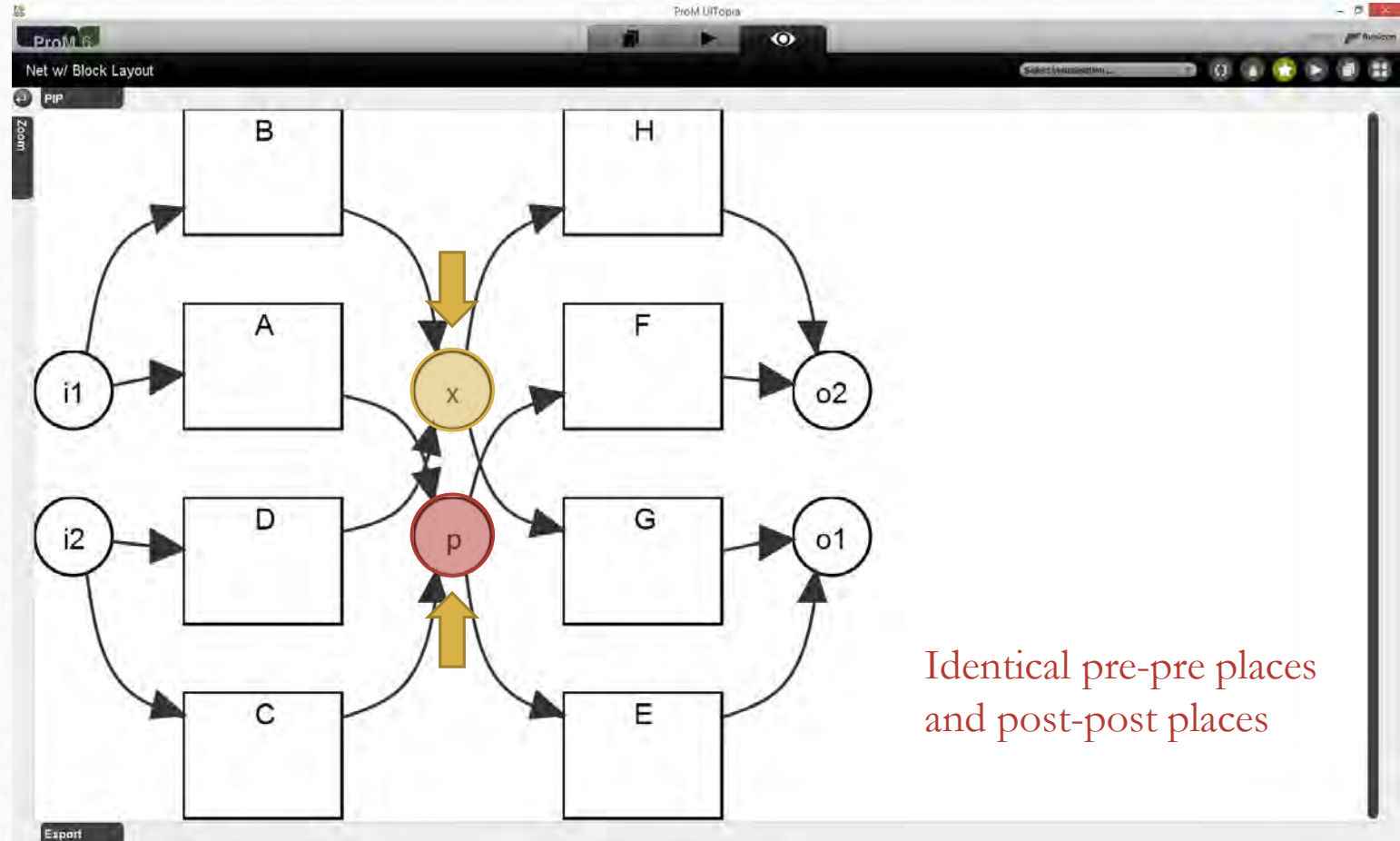
Hard

Prefer max height,
then max width?



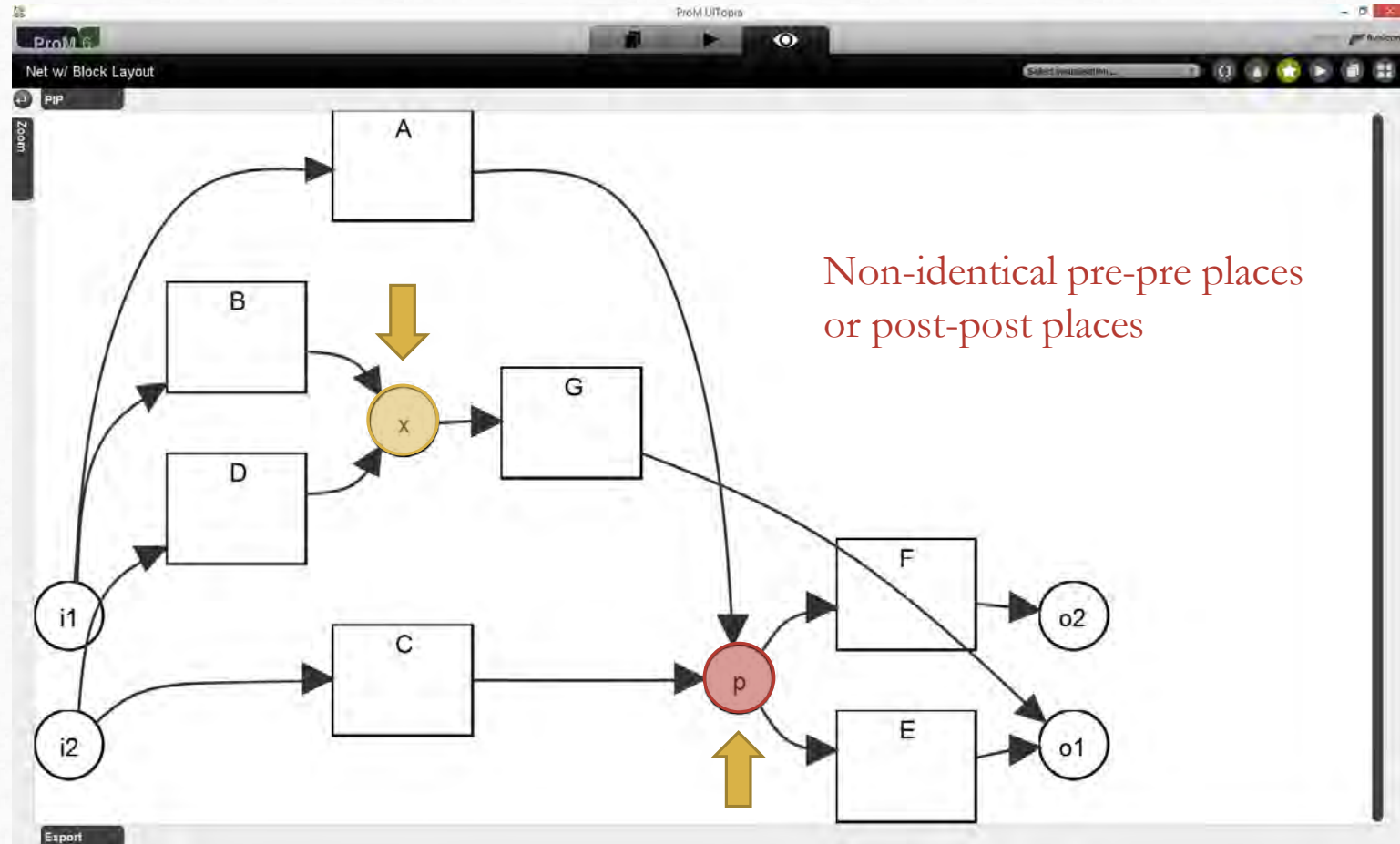
Alternative place rule

Easy



Alternative place rule

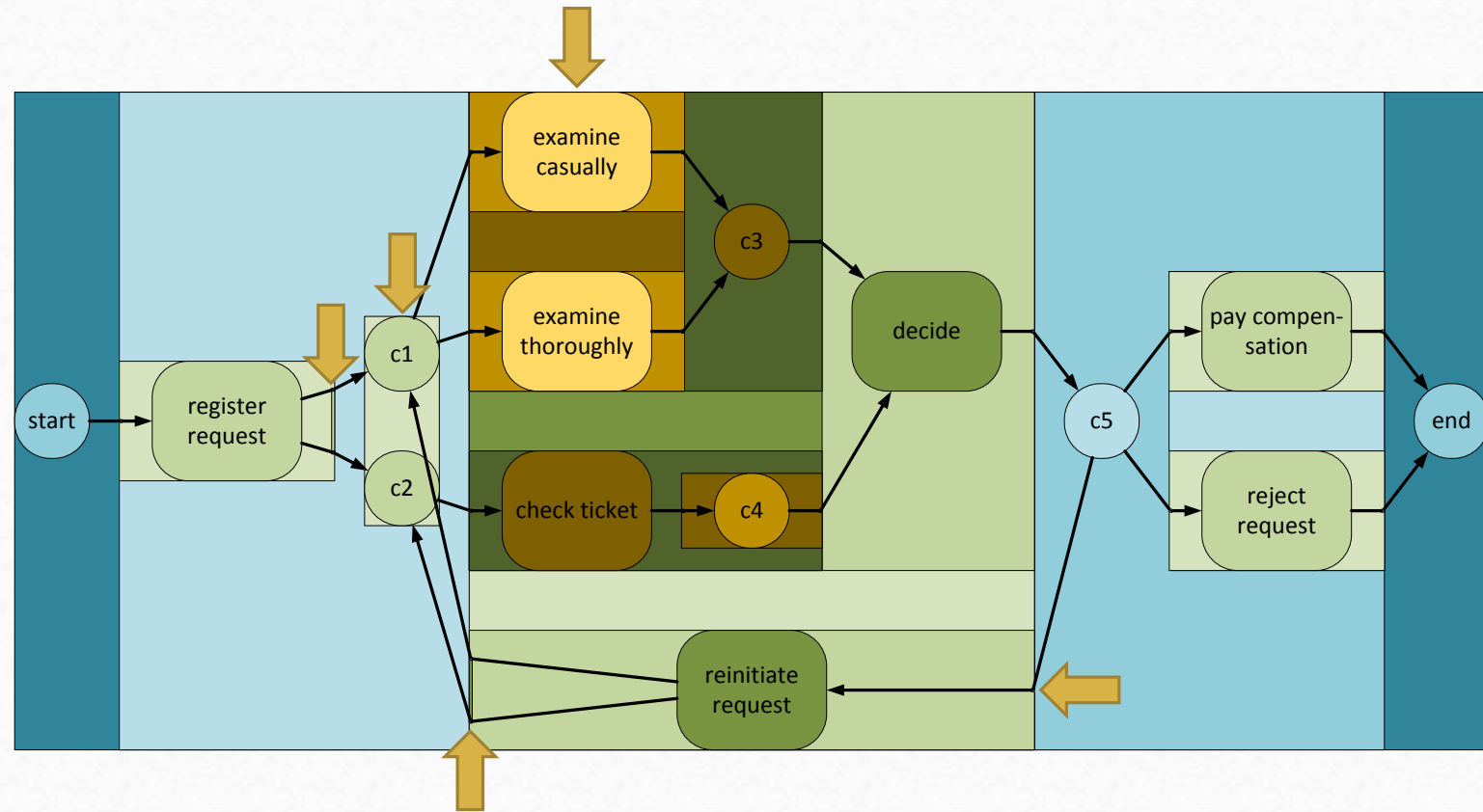
Hard
(H removed)



What to do if nothing is easy?

Vertical sort

- Loop-backs sorted at the bottom
- Based on current vertical sort of neighbors



Overview

- ✓ Reduction (of well-formed free-choice nets)
- ✓ Application
 - ✓ Sound free-choice workflow nets
 - ✓ Any workflow net
- ✓ Demo
- ✓ Discussions
- Conclusions

Conclusions

Pro

- No layers needed
- No need to revert some arcs
- Nice layouts for nicely structured nets
 - Not yet clear what “nicely structured” means here... “easy” does it...
 - Inductive Miner results typically qualify
- Almost deterministic (ILP solver is not)

Con

- No checking for crossings
 - Arcs may even overlap with nodes
- Not-so-nice layouts for not-so-nicely structured nets
 - Portal arcs?

Questions?



Priorities

Rule

- Abstraction
- Transition
- Place
- $0.99 / |\text{preset transitions}|$
- $1.00 / |\text{dependent transitions}|$
- $0.98 \times (|\cap| / |\cup|)$