SERA 2013

7th August 2013
Praha

Activity Diagrams Patterns for Modeling Business Processes

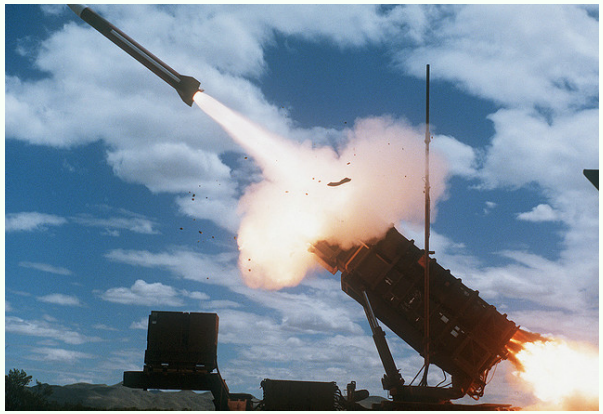
Étienne André[†], Christine Choppy[†], Gianna Reggio[‡]

[†] LIPN, Université Paris 13, Sorbonne Paris Cité, France
[‡] DIBRIS, Genova, Italy

Context: Verifying Complex Systems

- Need for early bug detection
  - Bugs discovered when final testing: expensive

  $\rightsquigarrow$ Need for a thorough modeling phase



UML Activity Diagrams

- Components of UML used to express the behavior of dynamic systems
- Specified in [Object Management Group, 2012]
- Often used to model business processes
  - Application to enterprise application (SOA paradigm)
- Rich syntax
  - ☹ Error-prone
- Informal semantics
  - ☹ Prevents formal verification

Our Contribution

- Set of activity diagram patterns for modeling business processes
- Modular mechanism to compose fragments into a UML activity diagram
  - ☺ Discarding ambiguous, rarely used or ill-formed activities
  - ☺ Easy design using composition
  - ☺ Helps to avoid common errors
- Formal semantics using a translation to colored Petri nets
  - ☺ Allowing for formal verification
  - ☺ Relatively natural since Petri nets are reasonably near activity diagrams

Outline

Outline

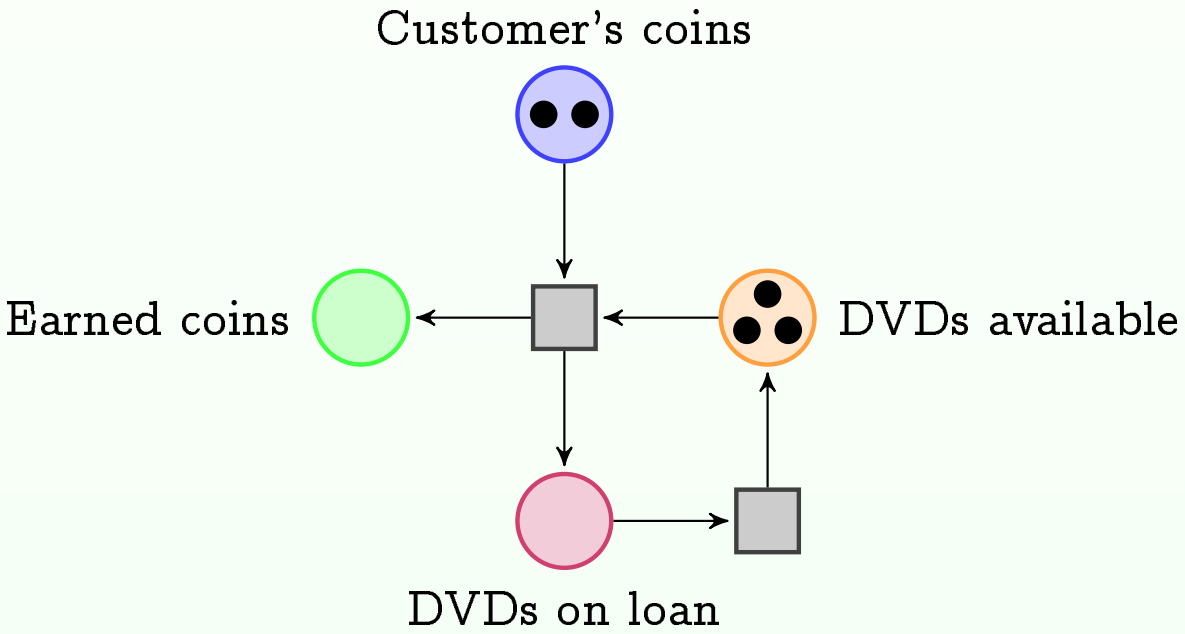
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

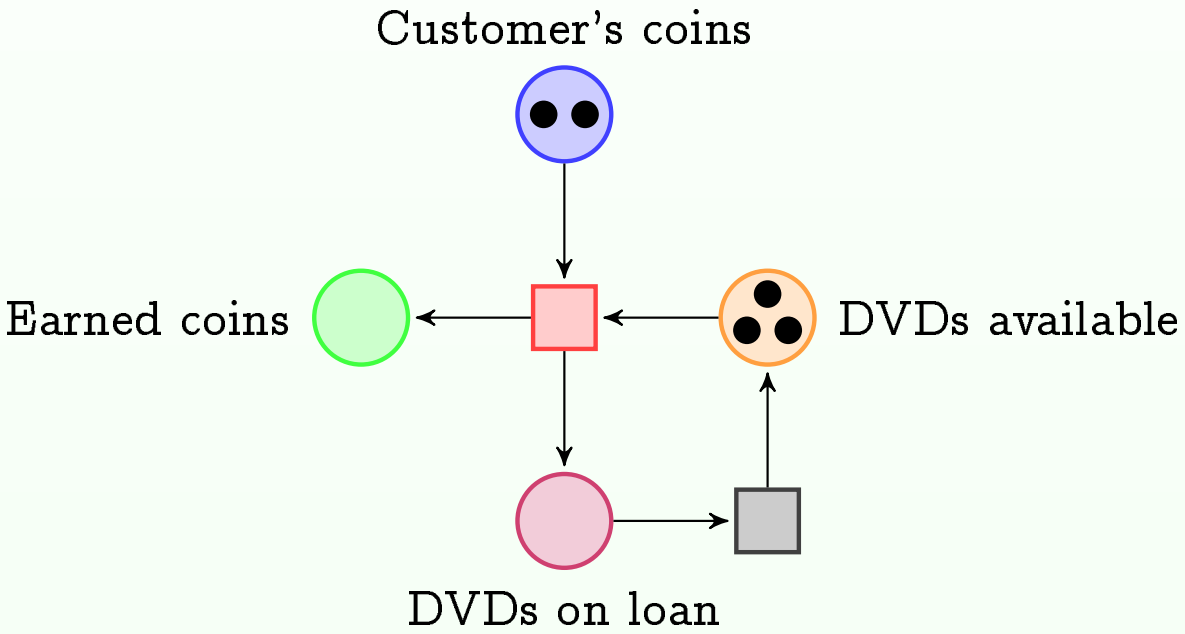
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

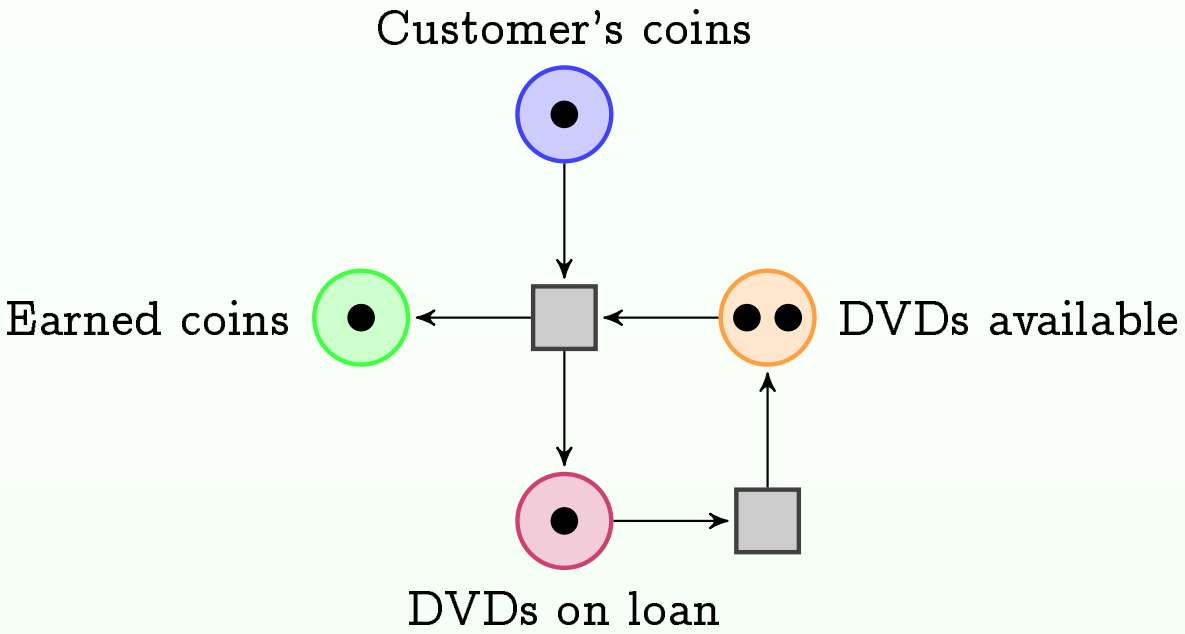
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

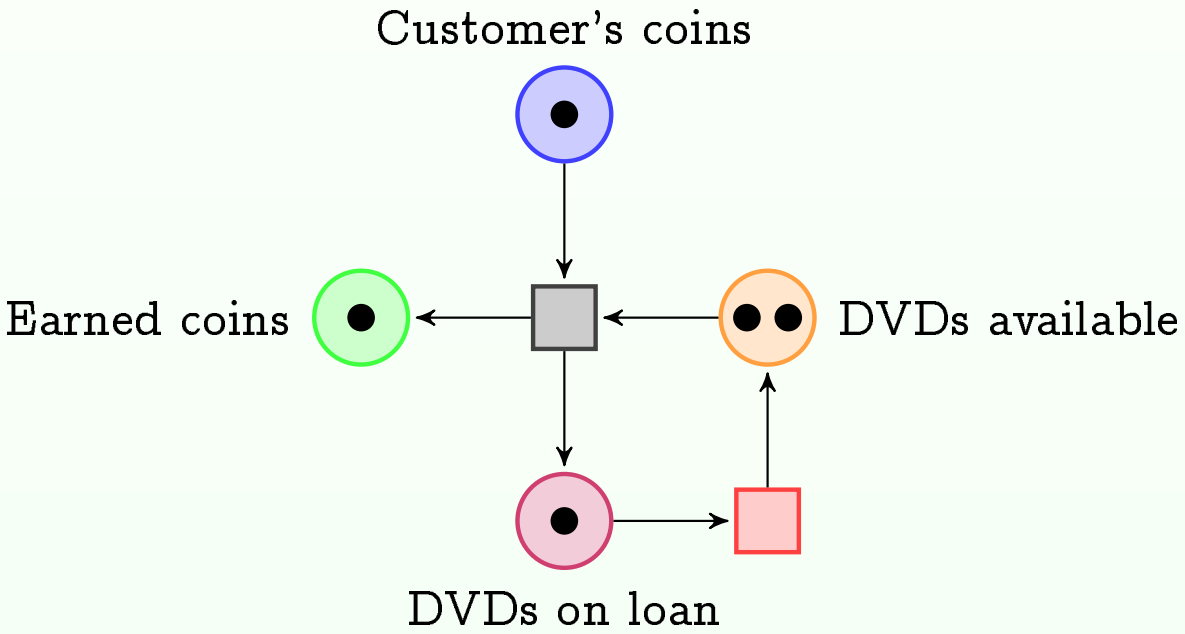
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

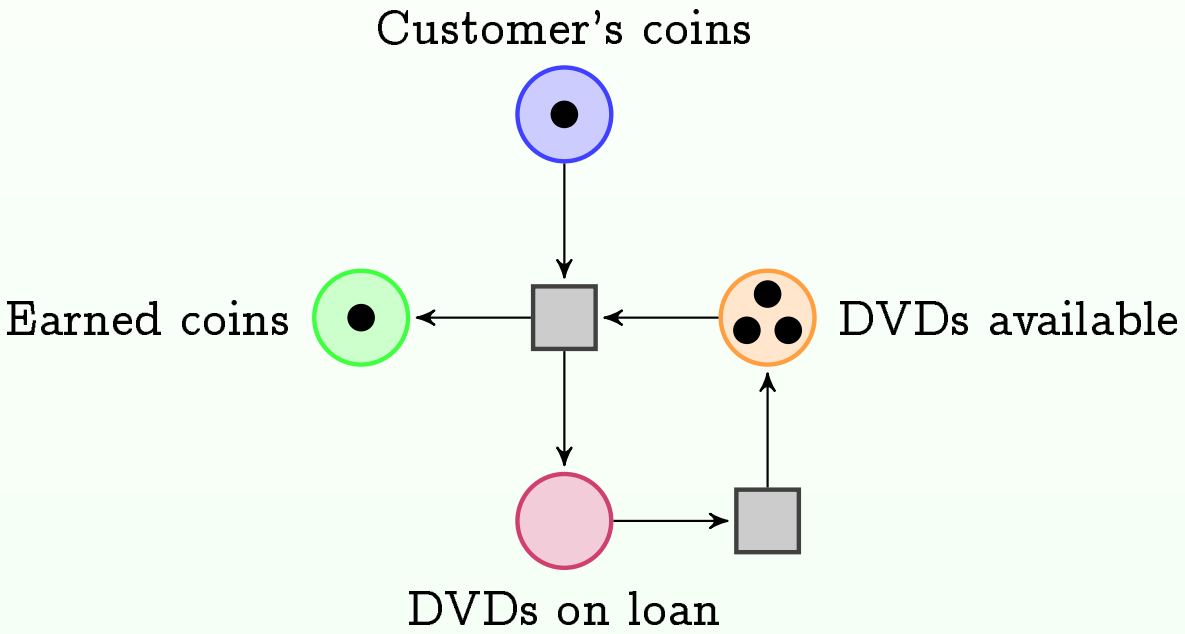
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

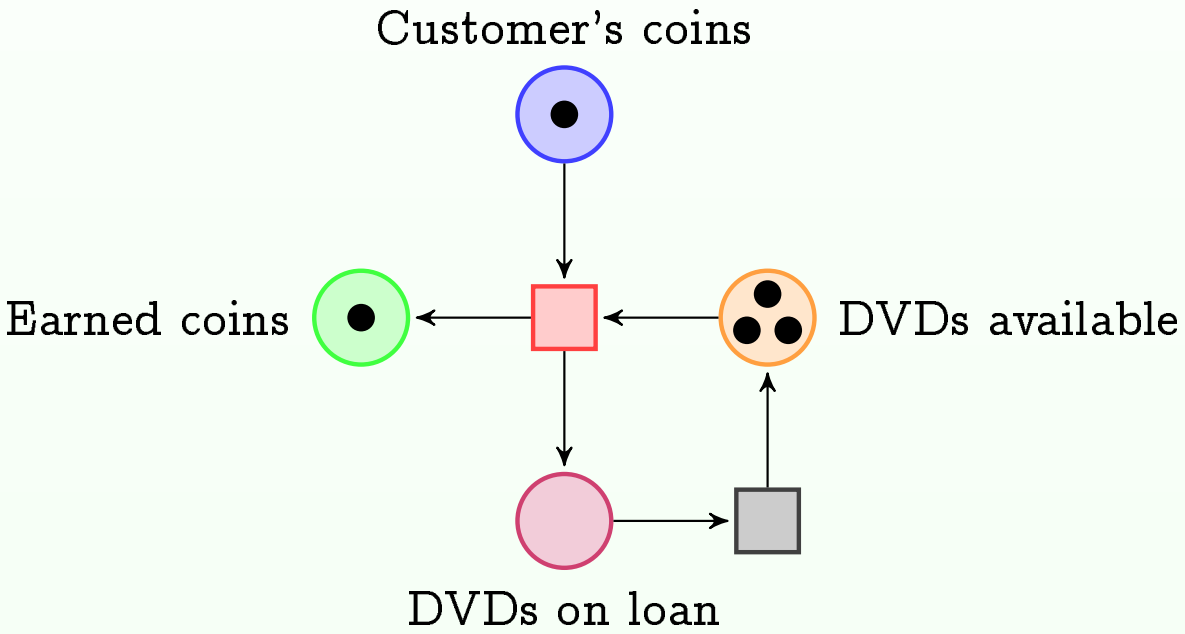
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

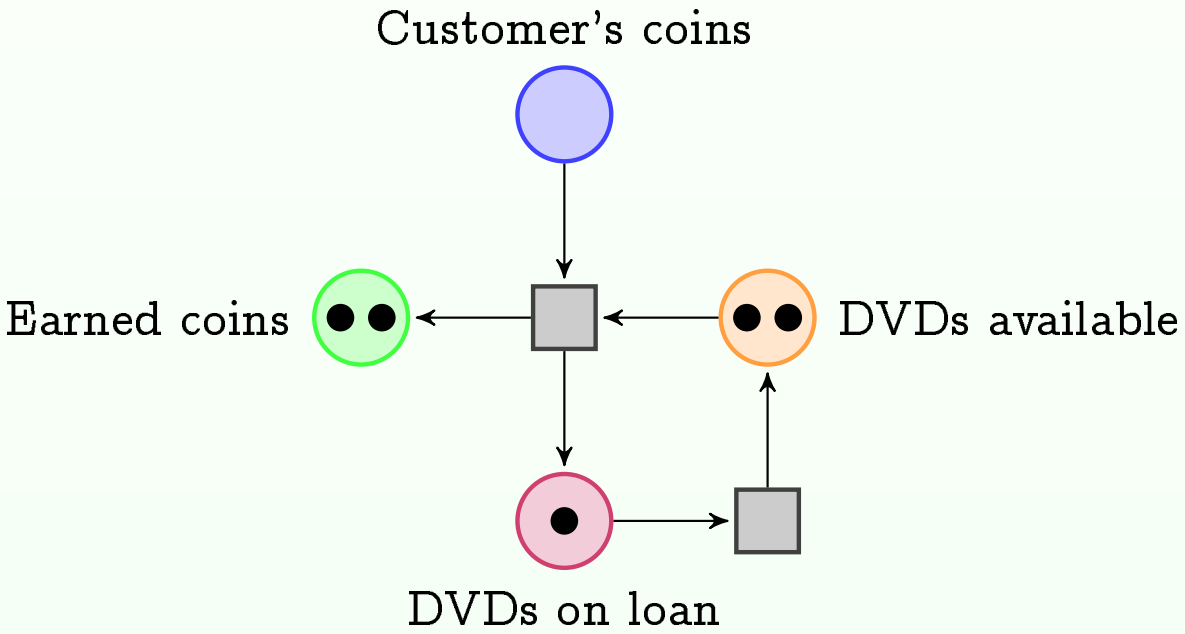
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

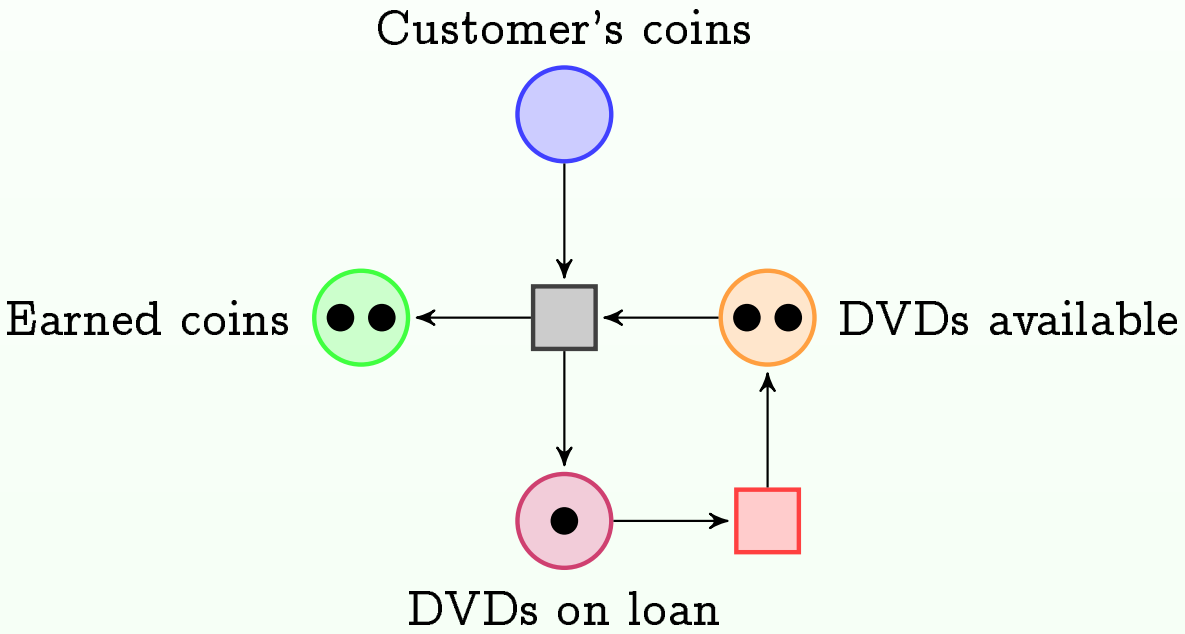
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

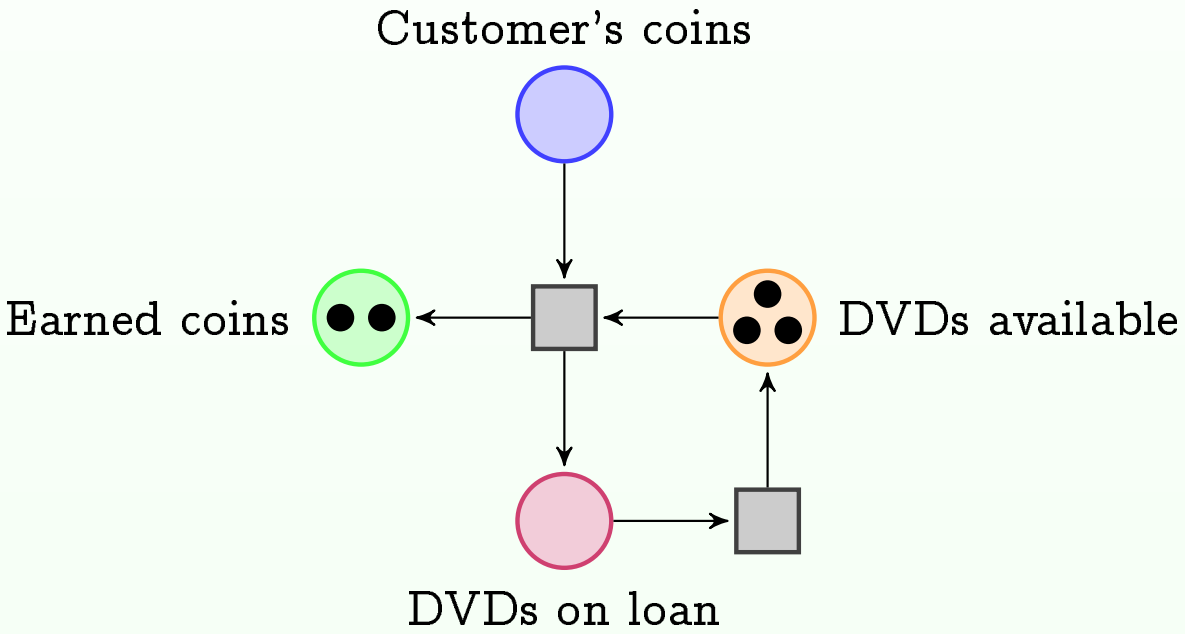
- Example: A DVD renting machine



Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation
  - A formal semantics
  - Powerful tools to test and check the model

- Example: A DVD renting machine



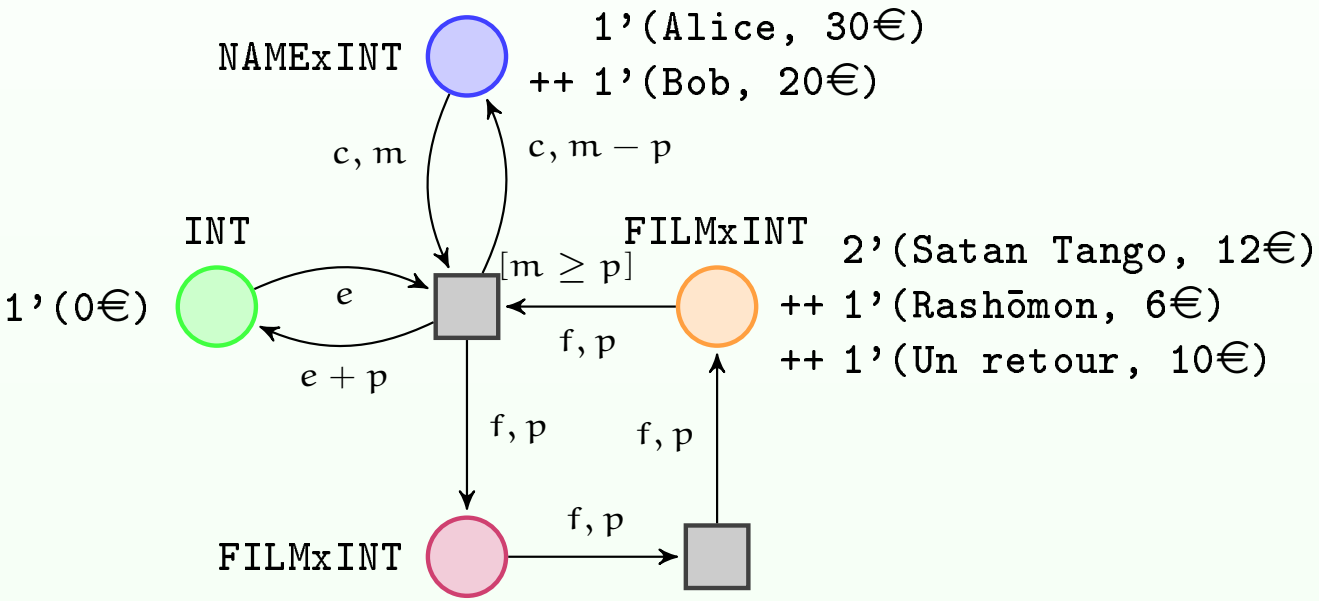
Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

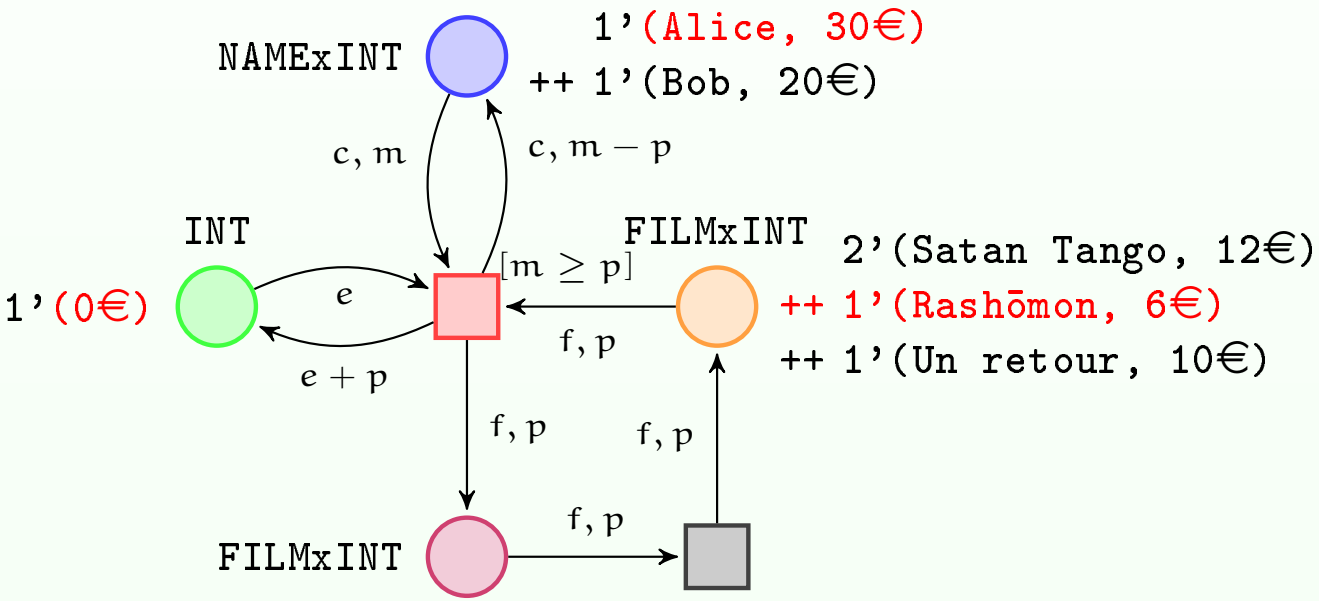
- Example: A more complex version of the DVD renting machine



Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

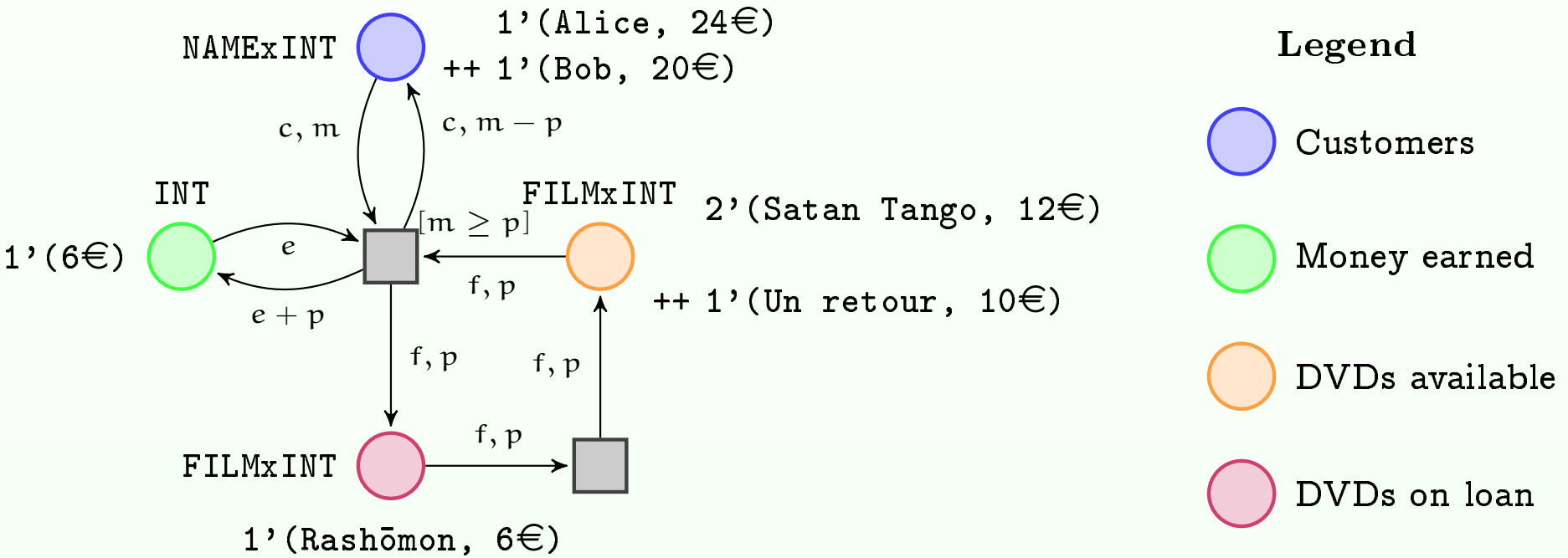
- Example: A more complex version of the DVD renting machine



Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

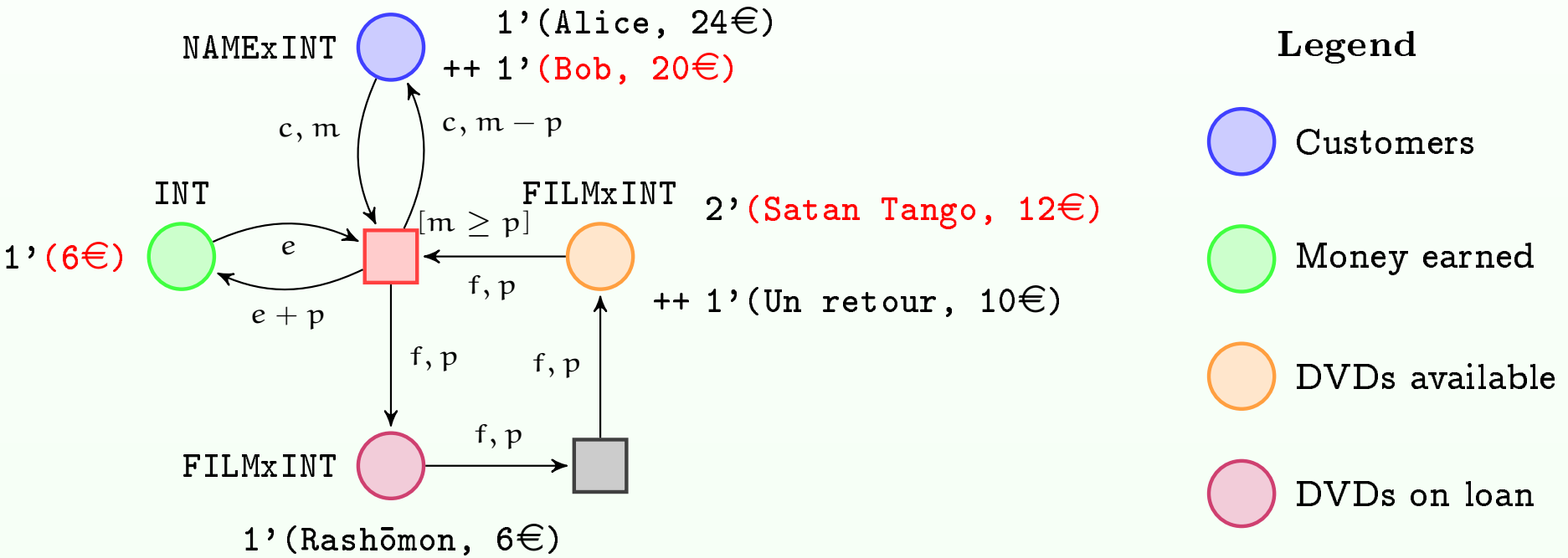
- Example: A more complex version of the DVD renting machine



Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

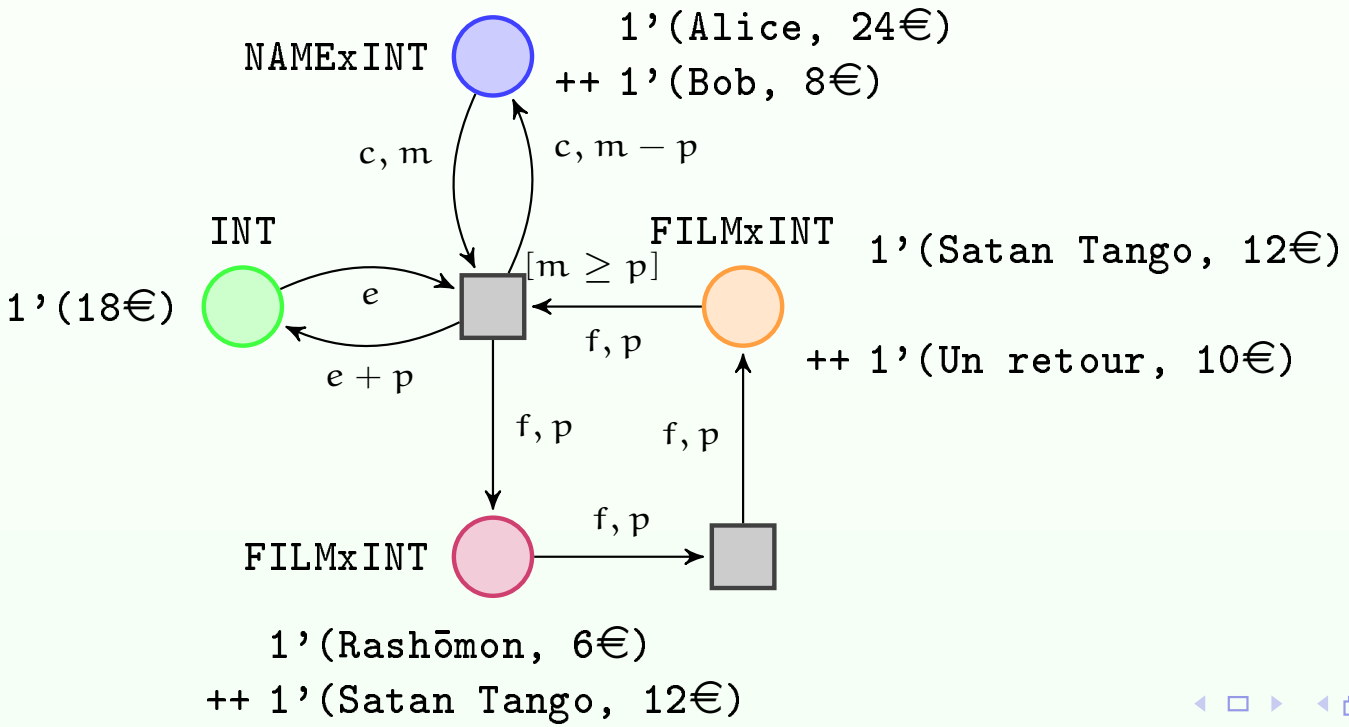
- Example: A more complex version of the DVD renting machine



Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

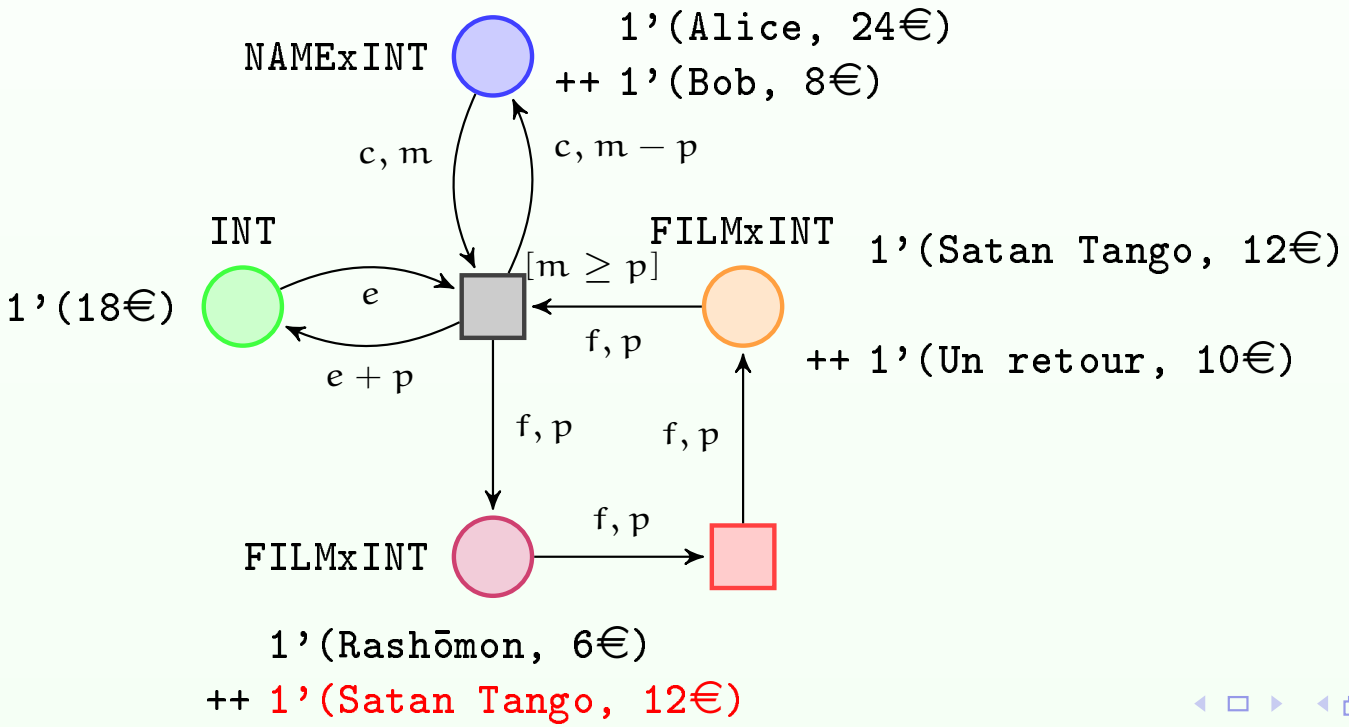
- Example: A more complex version of the DVD renting machine



Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

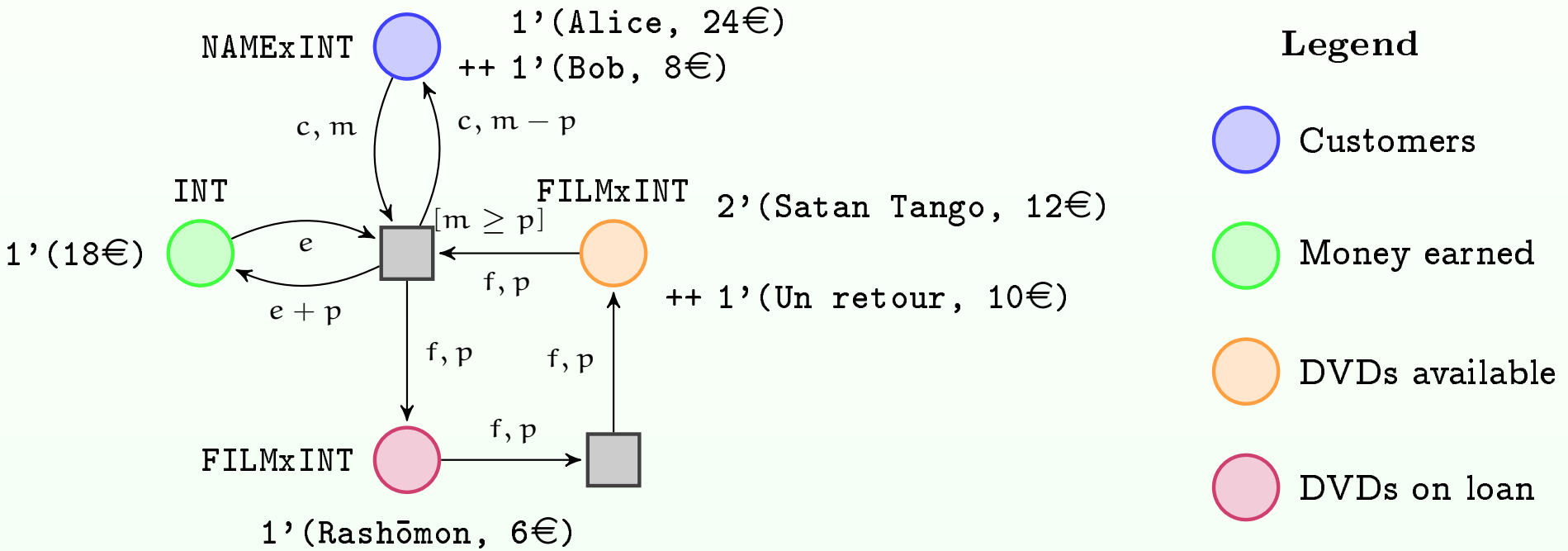
- Example: A more complex version of the DVD renting machine



Colored Petri Nets [Jensen and Kristensen, 2009]

- Extension of Petri nets with colors
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

- Example: A more complex version of the DVD renting machine

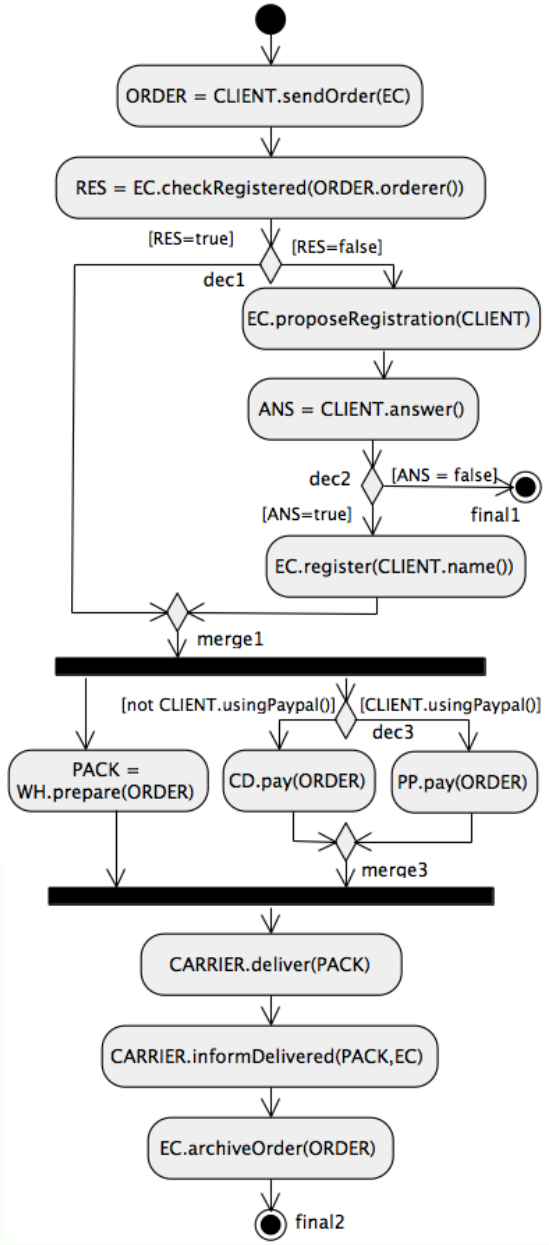


Outline

UML Activity Diagrams

- Initial nodes, final nodes (activity final, flow final)
- Choice and merge
  - Ability to follow one path among several possibilities
- Fork and join
  - Ability to split the flow into different activities in parallel

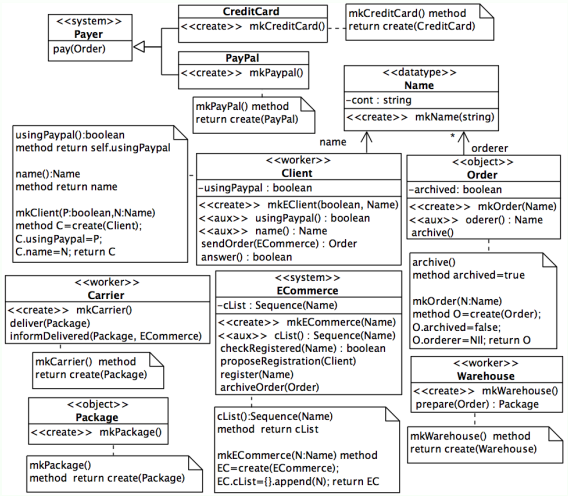
On the right hand-side: Example of an E-commerce



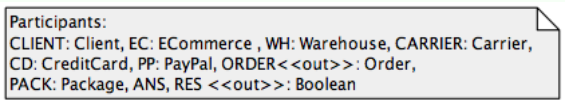
Precise Business Process Models

- Precise: defined in a sharper way than usual [Reggio et al., 2011]

- Precise Business Process Models
  1. A precise UML activity diagram (composed of patterns)
  2. A static view

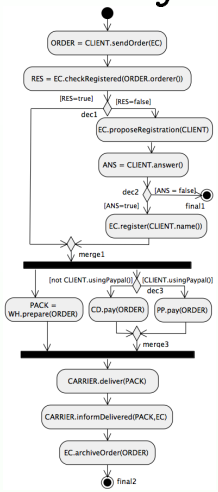


  3. A list of the participants

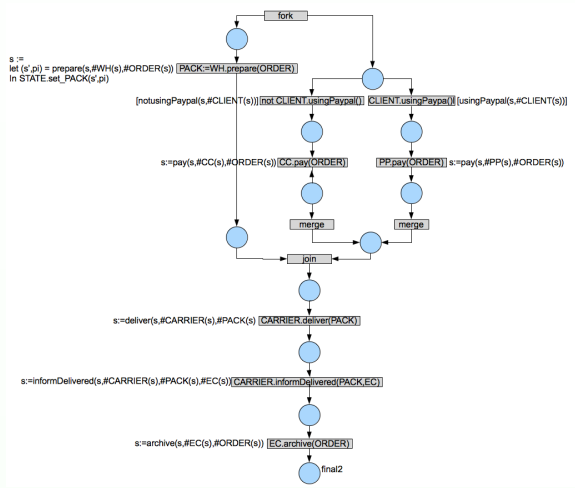


Translation General Scheme

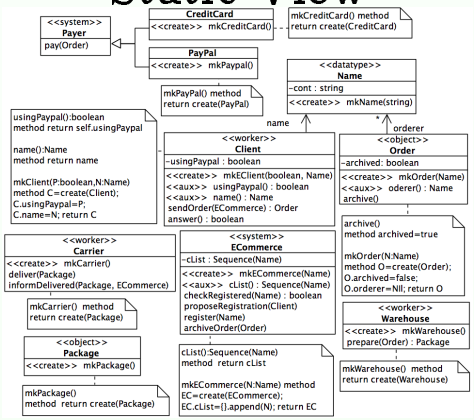
Precise activity diagram



Colored Petri net



Static View



Participants

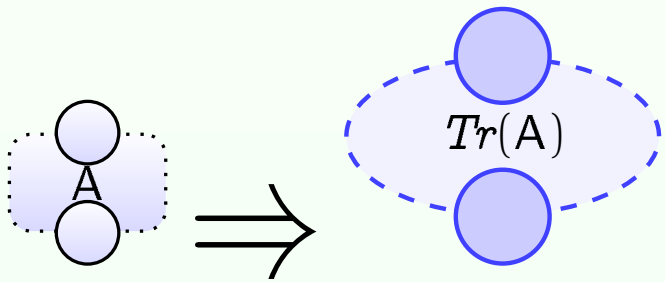
Participants:
CLIENT: Client, EC: ECommerce , WH: Warehouse, CARRIER: Carrier,
CD: CreditCard, PP: PayPal, ORDER<<out>>: Order,
PACK: Package, ANS, RES <<out>>: Boolean

Set of declarations and functions

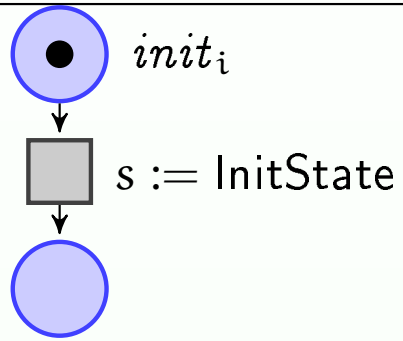
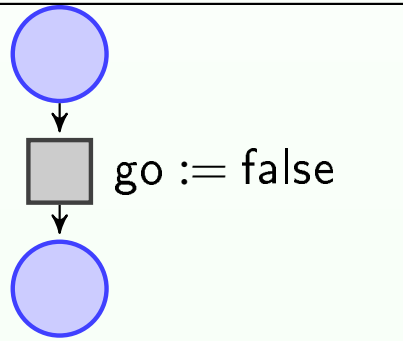
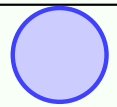
```
(*************************************************************
* Initial Value for State
*************************************************************)
        val initState = {}
        CLIENT = 0,
        EC = 0,
        WH = 0,
        CARRIER = 0,
        CC = 0,
        PP = 0,
        ORDER = 0,
        PACK = 0,
        ANS = false,
        RES = false,
        clients = [(0, {usingPaypal = false, name = mkName "init" , control =
Client0 })],

fun upClient(nil,ci,cs) = [(ci,cs)]
        | upClient((ci',cs')::cls, ci, cs) =
        if ci'=ci then
                (ci,cs)::cls
        else
                (ci',cs'):: (upClient(cls,ci,cs))
```

Activity Diagram Patterns

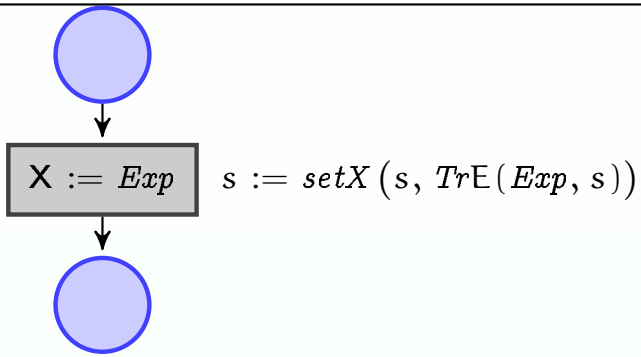
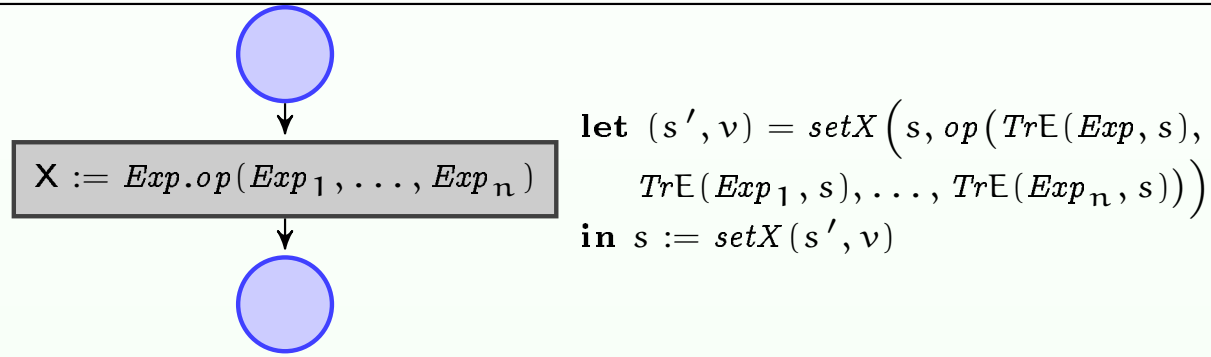
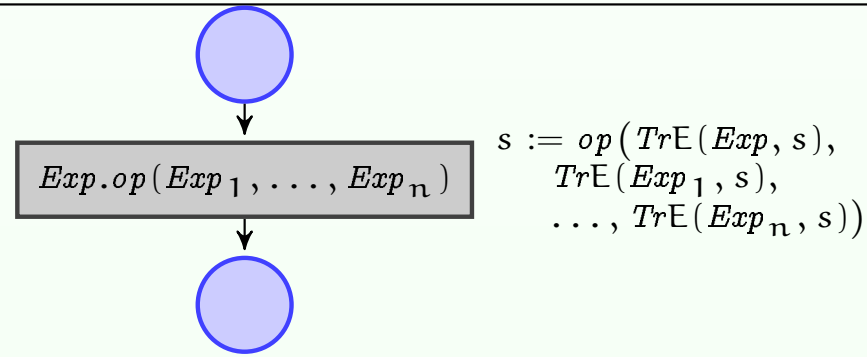
- Restricted expressiveness
  - Discards rarely used, ambiguous or ill-formed constructs
  - Gathered from years of experience in modeling
- Can be composed with each other to build a precise activity diagram
- General shape
  - A begin node (possibly undefined)
  - An end node (possibly undefined)
- Each pattern is translated into a colored Petri net fragment
  - Using various variables (in particular the current state s)



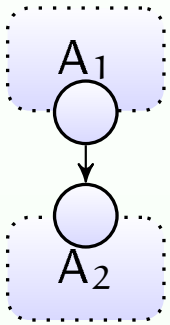
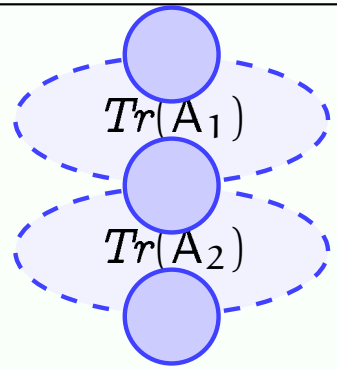
Simple Patterns: Initial and Final

Pattern	Activity Diagram	Colored Petri net
Initial		
Activity final		
Flow final		

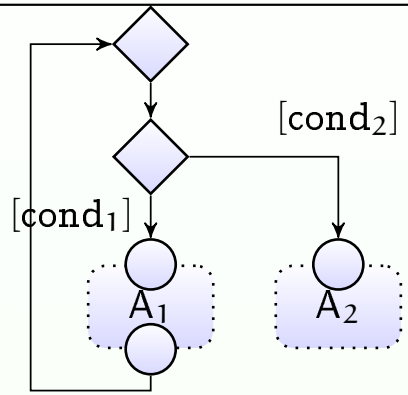
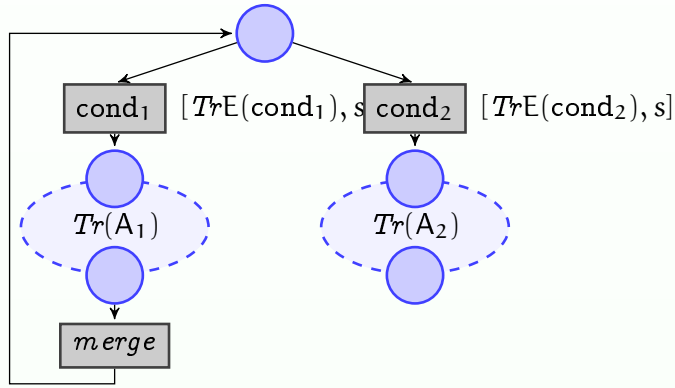
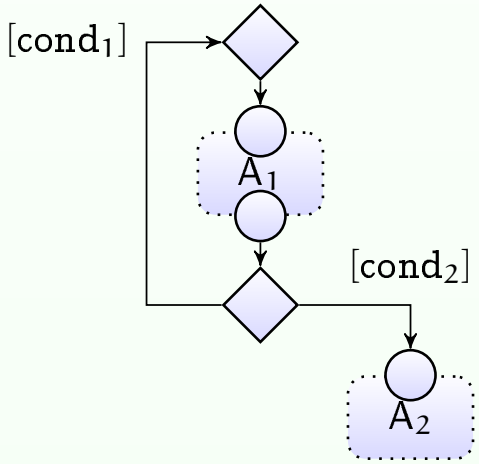
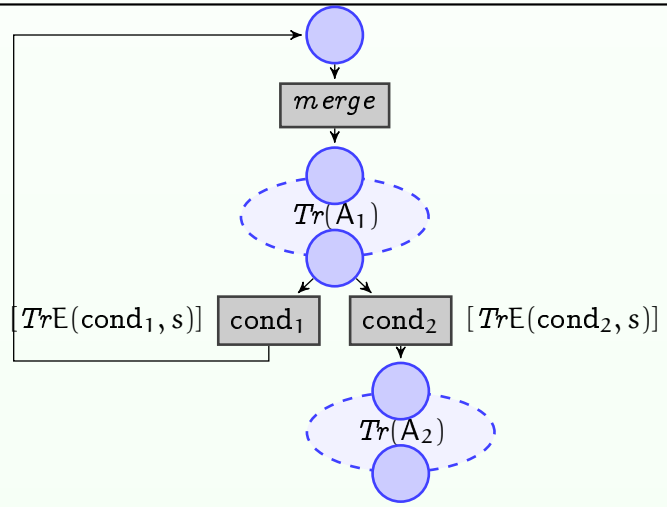
Simple Patterns: Actions

Activity Diagram	Colored Petri net
$X := Exp$	
$X := Exp.op(Exp_1, \ldots, Exp_n)$	
$Exp.op(Exp_1, \ldots, Exp_n)$	

Complex Patterns: Sequence

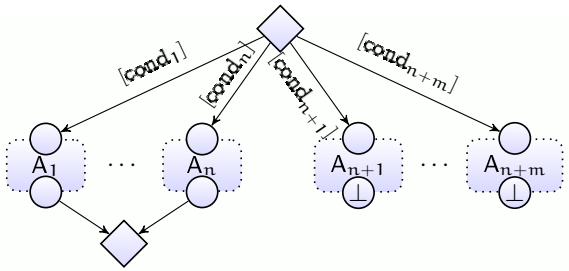
Pattern	Activity Diagram	Colored Petri net
Sequence		

Complex Patterns: Loops

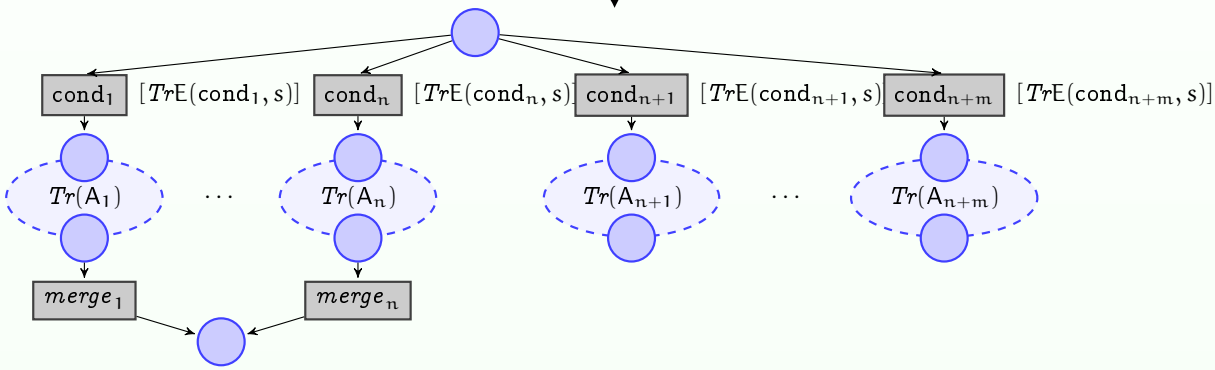
Pattern	Activity Diagram	Colored Petri net
While		
Repeat until		

Complex Patterns: Decision/merge

Activity diagram



Colored Petri net

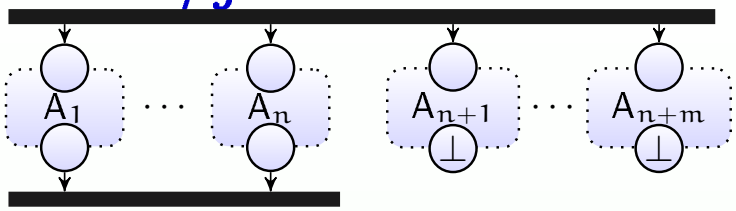


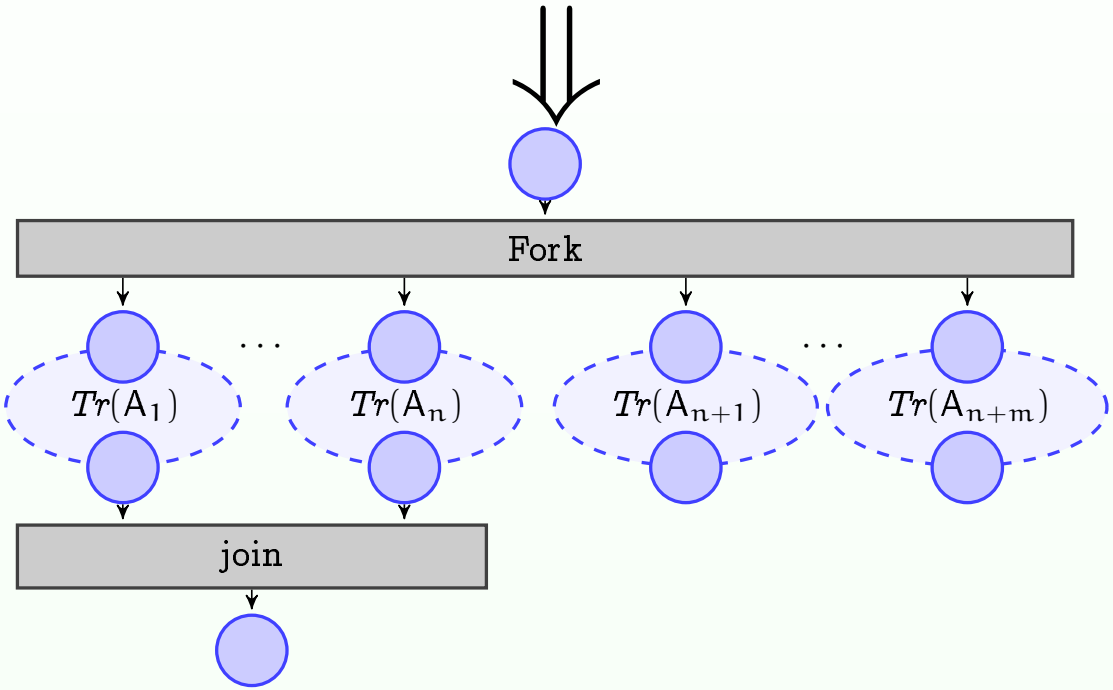
Remarks:

- Combines a decision node and a merge node
- Avoids errors such as decision with no merge

Complex Patterns: Fork/join

Activity diagram





Colored Petri net

Remarks:

- Combines a fork node and a join node
- Avoids errors such as fork with no join

Outline

Application to an E-Commerce Example

- Manual translation following our rules
  - Colored Petri net: 26 places, 25 transitions
  - Declarations and functions: 540 lines of PNML code
- Automatic verification performed using CPN Tools
  [Jensen and Kristensen, 2009]
- Properties verified
  - E.g.: For any execution, exactly one final state is reached

Outline

Conclusion

- Set of activity diagram patterns for modeling business processes
- Modular mechanism to compose fragments into a UML activity diagram
  - Discarding ambiguous, rarely used or ill-formed activities
  - Easy composition avoiding common errors
- Formal semantics using a translation to colored Petri nets
  - Formal verification using CPN Tools

Perspectives

- Implementation
  - Translation of the activity diagram: ongoing
    (using model transformation technologies)
  - Translation of the static view and participants: to do
- Comparison of our translation with existing semantics
  (e.g., [Kraemer and Herrmann, 2010, Grönniger et al., 2010])
- Simplification of the resulting colored Petri net (including the functions)
  - E.g., in case of a "merge" with only one node
- Integration of timed events

Bibliography

References I

Grönniger, H., Reiss, D., and Rumpe, B. (2010).
Towards a semantics of activity diagrams with semantic variation points.
In *MODELS*, volume 6394 of *LNCS*, pages 331–345. Springer.

Jensen, K. and Kristensen, L. M. (2009).
Coloured Petri Nets – Modelling and Validation of Concurrent Systems.
Springer.

Kraemer, F. A. and Herrmann, P. (2010).
Reactive semantics for distributed UML activities.
In *FMOODS/FORTE*, volume 6117 of *LNCS*, pages 17–31. Springer.

Object Management Group (2012).
OMG unified language superstructure specification(formal). version 2.4.1, 2011-08-06.
`http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/`.

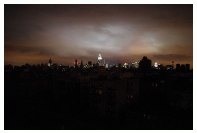
Petri, C. A. (1962).
Kommunikation mit Automaten.
PhD thesis, Darmstadt University of Technology, Germany.

References II

Reggio, G., Ricca, F., Scanniello, G., Cerbo, F., and Dodero, G. (2011).
A precise style for business process modelling: Results from two controlled experiments.
In *Model Driven Engineering Languages and Systems*, volume 6981 of *LNCS*, pages 138–152. Springer.

Additional explanation

Explanation for the 4 pictures in the beginning



Allusion to the Northeast blackout (USA, 2003)
Computer bug
Consequences: 11 fatalities, huge cost
(Picture actually from the Sandy Hurricane, 2012)

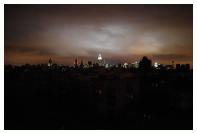
Allusion to any plane crash
(Picture actually from the happy-ending US Airways Flight 1549, 2009)

Allusion to the sinking of the Sleipner A offshore platform (Norway, 1991)
No fatalities
Computer bug: inaccurate finite element analysis modeling
(Picture actually from the Deepwater Horizon Offshore Drilling Platform)

Allusion to the MIM-104 Patriot Missile Failure (Iraq, 1991)
28 fatalities, hundreds of injured
Computer bug: software error (clock drift)
(Picture of an actual MIM-104 Patriot Missile, though not the one of 1991)

Licensing

Source of the graphics used



Title: Hurricane Sandy Blackout New York Skyline
Author: David Shankbone
Source: https://commons.wikimedia.org/wiki/File:Hurricane_Sandy_Blackout_New_York_Skyline.JPG
License: CC BY 3.0

Title: Miracle on the Hudson
Author: Janis Krums (cropped by Étienne André)
Source: https://secure.flickr.com/photos/davidwatts1978/3199405401/
License: CC BY 2.0

Title: Deepwater Horizon Offshore Drilling Platform on Fire
Author: ideum
Source: https://secure.flickr.com/photos/ideum/4711481781/
License: CC BY-SA 2.0

Title: DA-SC-88-01663
Author: imcomkorea
Source: https://secure.flickr.com/photos/imcomkorea/3017886760/
License: CC BY-NC-ND 2.0

License of this document

Author: Étienne André