Specification of Asynchronous Component Systems with Modal I/O-Petri Nets

Serge Haddad[1] Rolf Hennicker[2] Mikael H. Møller[3]

[1]ENS Cachan & CNRS & INRIA, France

[2]Ludwig-Maximilians-Universität München, Germany

[3]Aalborg University, Denmark

Introduction

Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

Introduction

Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

Interested in:

- Infinite state systems

Introduction

Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

Interested in:

- Infinite state systems $\longrightarrow$ Petri nets

Introduction

Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

Interested in:

- Infinite state systems $\longrightarrow$ Petri nets
- Loose specifications

Introduction

Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

Interested in:

- Infinite state systems $\longrightarrow$ Petri nets
- Loose specifications $\longrightarrow$ modal transitions (may, must)

Introduction

Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

Interested in:

- Infinite state systems $\longrightarrow$ Petri nets
- Loose specifications $\longrightarrow$ modal transitions (may, must)
- Observational abstraction

Introduction

Context:

- Systems of asynchronously communicating components.
- Formal specification of component and system behaviors.

Interested in:

- Infinite state systems $\longrightarrow$ Petri nets
- Loose specifications $\longrightarrow$ modal transitions (may, must)
- Observational abstraction $\longrightarrow$ hiding and τ-transitions

Introduction

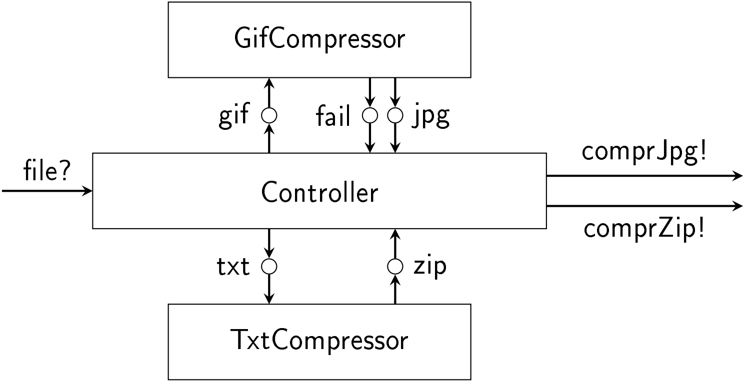
Context:

- Systems of asynchronously communicating components
- Formal specification of component and system behaviors

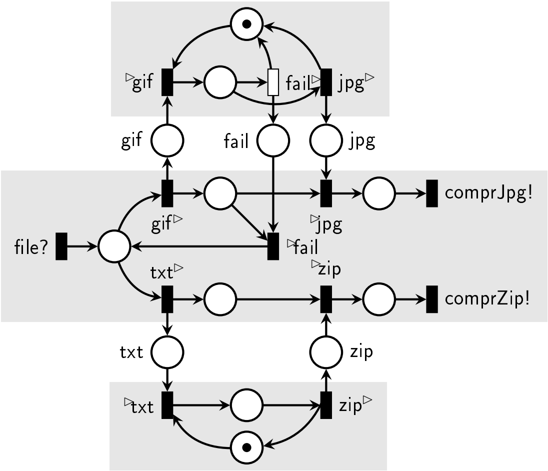
Interested in:

- Infinite state systems $\longrightarrow$ Petri nets
- Loose specifications $\longrightarrow$ modal transitions (may, must)
- Observational abstraction $\longrightarrow$ hiding and τ-transitions
- Refinement correctness
- Communication correctness
- Decidability results $\longrightarrow$ Petri nets
- Modular and incremental verification

Example: File Compressing System

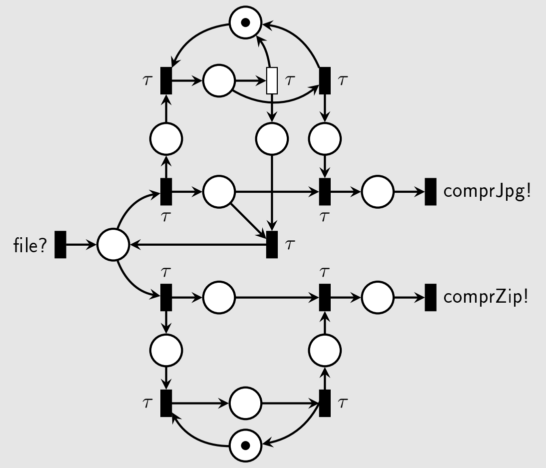


Example: CompressorAssembly =
GifCompressor $\otimes$ Controller $\otimes$ TxtCompressor



Hiding of Channel Places

Example: CompressorAssembly $\setminus$ {gif,fail,jpg,txt,zip}



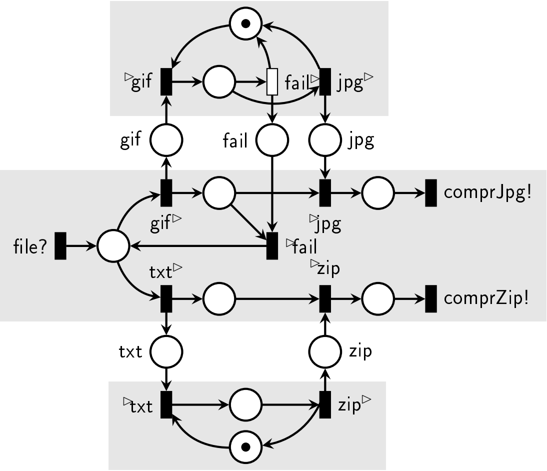
Semantics

Modal Asynchronous I/O-Transition Systems =

Modal I/O-Transition Systems [Larsen et al. 1988,2007]
extended by Communication Channels

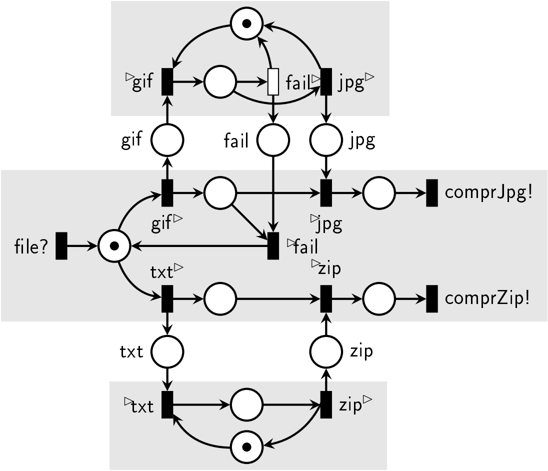
- states = reachable markings,
- initial state = initial marking,
- may-transitions $m \overset{a}{\dashrightarrow} m'$,
- must-transitions $m \xrightarrow{a} m'$,
  such that $m \xrightarrow{a} m' \implies m \overset{a}{\dashrightarrow} m'$

Semantics: Example



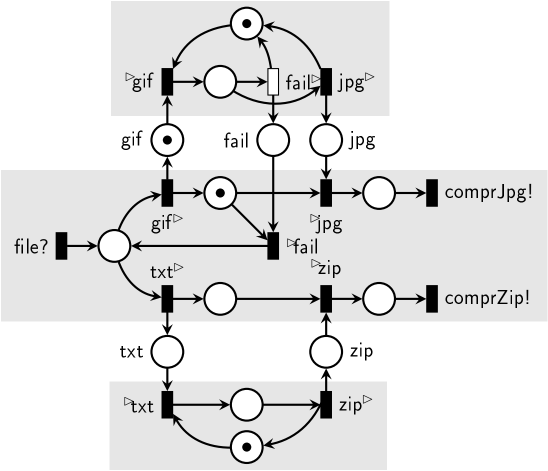
m^0

Semantics: Example



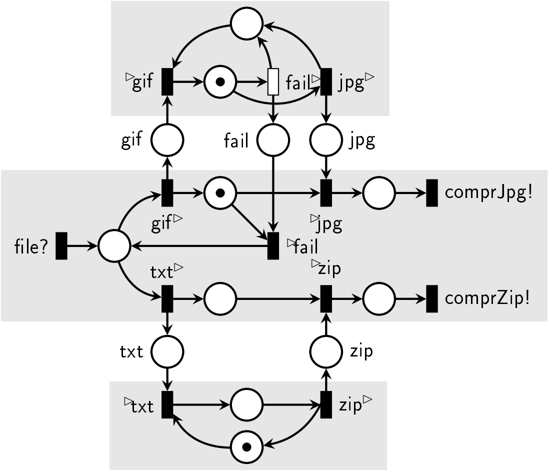
$$m^0 \xrightarrow{\text{file?}} m^1$$

Semantics: Example



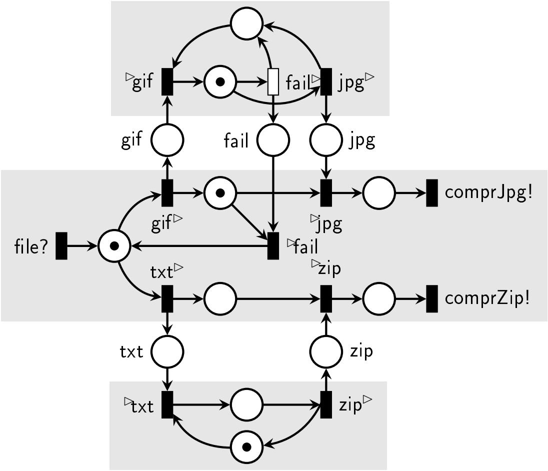
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2$$

Semantics: Example



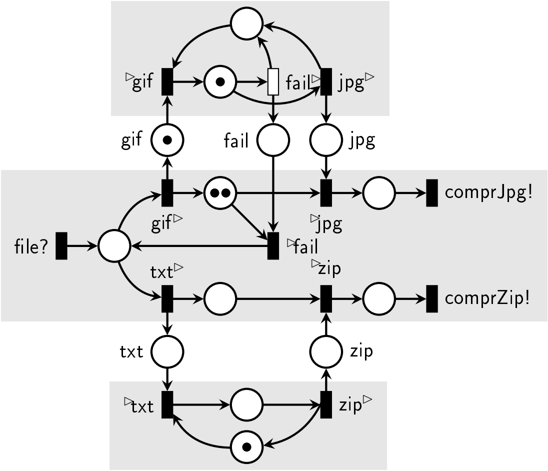
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3$$

Semantics: Example



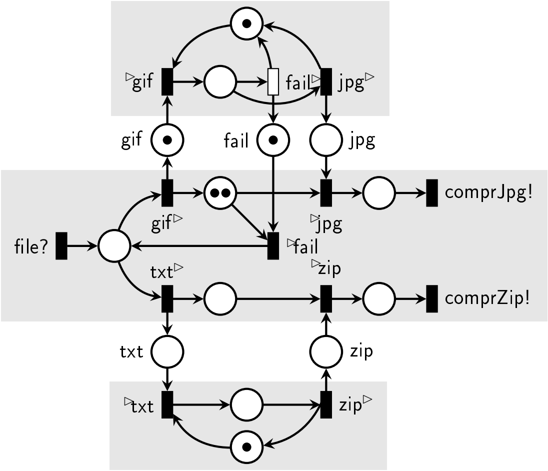
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3 \xrightarrow{\text{file?}} m^4$$

Semantics: Example



$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3 \xrightarrow{\text{file?}} m^4 \xrightarrow{\text{gif}^{\triangleright}} m^5$$

Semantics: Example

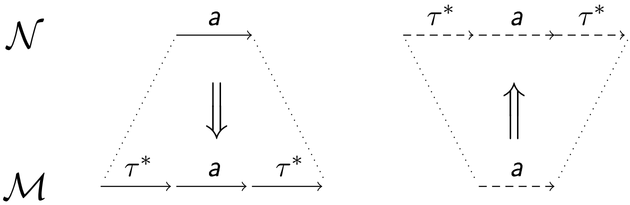


$$m^0 \xrightarrow{\mathsf{file?}} m^1 \xrightarrow{\mathsf{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\mathsf{gif}} m^3 \xrightarrow{\mathsf{file?}} m^4 \xrightarrow{\mathsf{gif}^{\triangleright}} m^5 \overset{\mathsf{fail}^{\triangleright}}{\dashrightarrow} m^6$$

Weak Modal Refinement [Hüttel, Larsen 1989]

concrete MAIOPN abstract MAIOPN

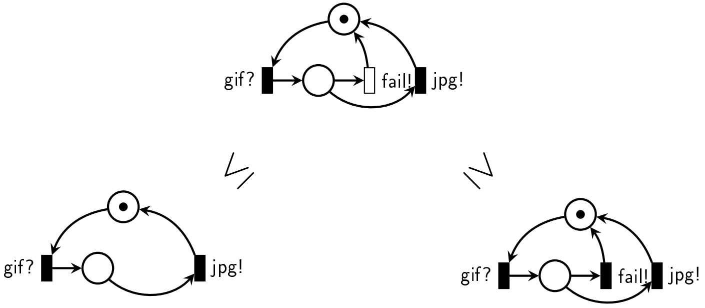
$$\mathcal{M} \leq \mathcal{N}$$



(a) Must-transitions of the abstract net are preserved (up to τs).

(b) May-transitions of the concrete net are simulated (up to τs).

Weak Modal Refinement: Example



Results for Refinement

Decidability:

(1) $\mathcal{M} \leq \mathcal{N}$ is decidable
if both $\mathcal{M}$ and $\mathcal{N}$ are modally weakly deterministic.

(2) Modal weak determinism is decidabe as well.

Modular verification:

(1) $\mathcal{M} \leq \mathcal{N}$ and $\mathcal{E} \leq \mathcal{F} \implies \mathcal{M} \otimes \mathcal{E} \leq \mathcal{N} \otimes \mathcal{F}$.

(2) $\mathcal{M} \leq \mathcal{N} \implies (\mathcal{M} \setminus H) \leq (\mathcal{N} \setminus H)$.

Communication Requirements

Goal: Avoid communication errors!

Typical communication errors:
Message not taken, message not delivered.

Variants: * synchronous vs asynchronous communication,
* message queues vs message pools,
* delayed vs undelayed reception,
* optimistic vs pessimistic view of the environment, ...

Literature:
* specified reception in CFSMs [Brand, Zafiropulo 1983],
* compatibility of interface automata [de Alfaro, Henzinger 2005],
* communication-safe assemblies [ICTAC 2011],
* I/O-compatibility in team automata [Carmona, Kleijn 2013], ...

We consider:
Message not taken, asynchronous communication with message pools, delayed reception, pessimistic view.

Consumption Properties for Channels

Assume given a MAIOPN $\mathcal{M}$ and a subset B of the channels of $\mathcal{M}$.

- $\mathcal{M}$ is **message consuming** w.r.t. B if for all channels $c \in B$ and for all reachable markings m of $\mathcal{M}$:
  if c is not empty, then *there exists* a path of autonomous must-transitions

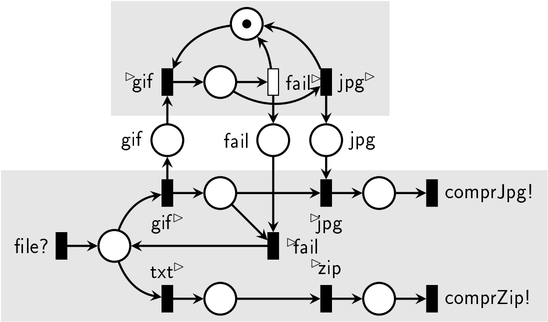
  $$m \xrightarrow{a_1} \dots \xrightarrow{a_n} m' \xrightarrow{\rhd c}$$

  Autonomous means: No open input action under the a_i.

- $\mathcal{M}$ is **necessarily message consuming** w.r.t. B if for all channels $c \in B$ and for all reachable markings m of $\mathcal{M}$:
  if c is not empty, then $\xrightarrow{\rhd c}$ will "always" eventually be executed.

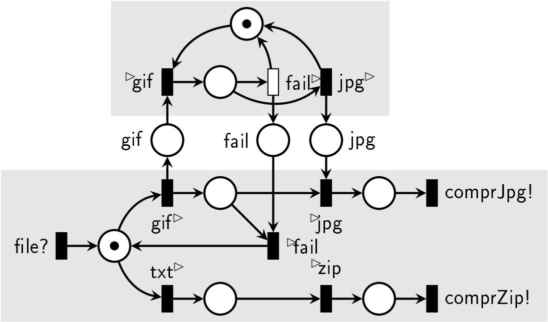
  What means "always"?

Message Consuming: Example



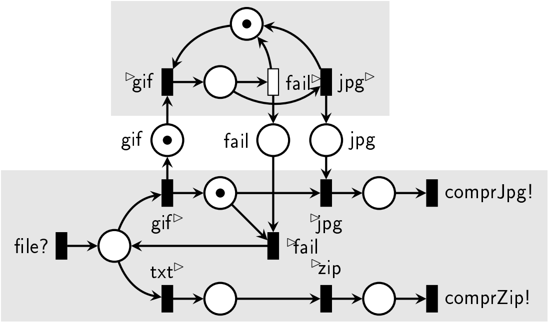
m^0

Message Consuming: Example



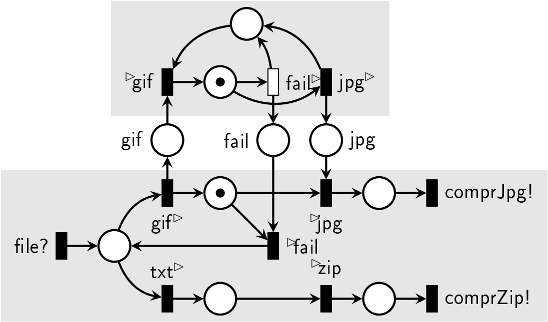
$m^0 \xrightarrow{\text{file?}} m^1$

Message Consuming: Example



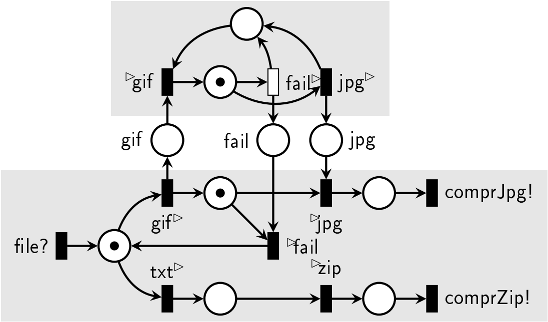
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2$$

Message Consuming: Example



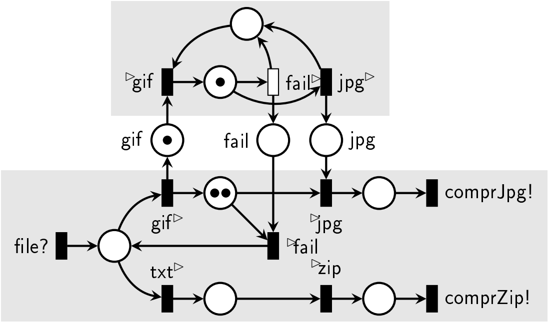
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3$$

Message Consuming: Example



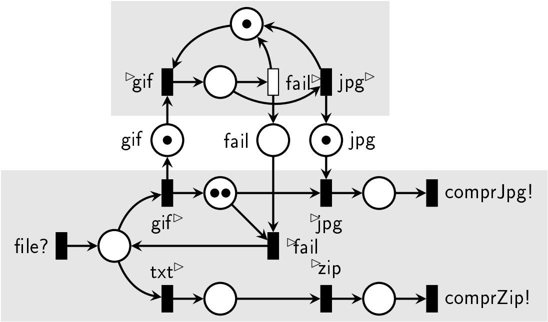
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{^{\triangleright}\text{gif}} m^3 \xrightarrow{\text{file?}} m^4$$

Message Consuming: Example



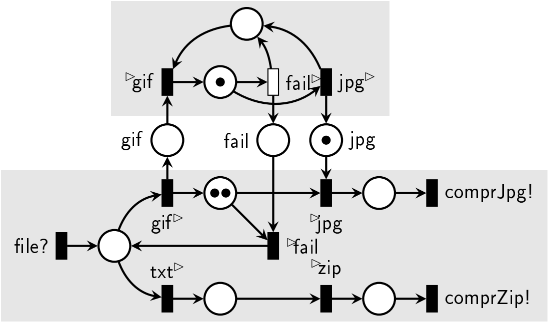
$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3 \xrightarrow{\text{file?}} m^4 \xrightarrow{\text{gif}^{\triangleright}} m^5$$

Message Consuming: Example



$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3 \xrightarrow{\text{file?}} m^4 \xrightarrow{\text{gif}^{\triangleright}} m^5 \xrightarrow{\text{jpg}^{\triangleright}} m^6$$

Message Consuming: Example



$$m^0 \xrightarrow{\text{file?}} m^1 \xrightarrow{\text{gif}^{\triangleright}} m^2 \xrightarrow{{}^{\triangleright}\text{gif}} m^3 \xrightarrow{\text{file?}} m^4 \xrightarrow{\text{gif}^{\triangleright}} m^5 \xrightarrow{\text{jpg}^{\triangleright}} m^6 \xrightarrow{{}^{\triangleright}\text{gif}} m^7$$

Results for Message Consuming

Decidability:

The property of message consuming is decidable.

Preservation by refinement:

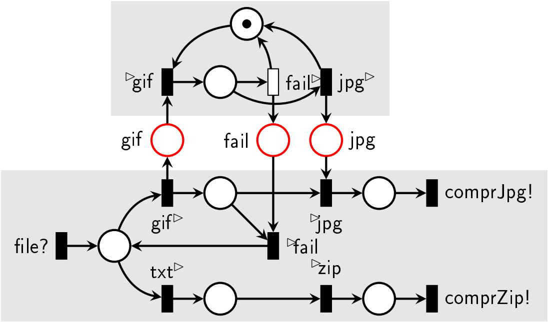
$\mathcal{N}$ message consuming and $\mathcal{M} \leq \mathcal{N} \implies \mathcal{M}$ message consuming.

Incremental verification:

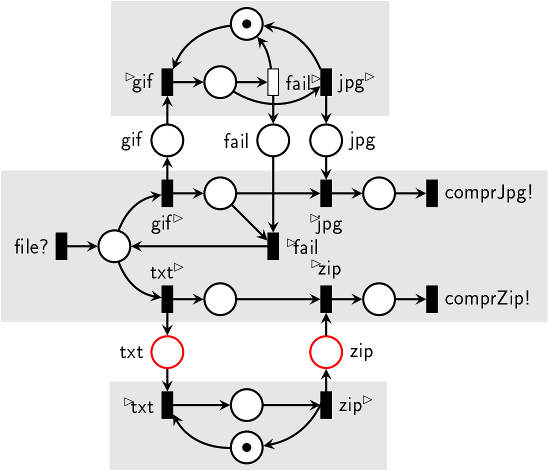
$\mathcal{M} \otimes \mathcal{N}$ message consuming and
$(\mathcal{M} \otimes \mathcal{N}) \otimes \mathcal{E}$ message consuming w.r.t. the new channels.

Then: $\mathcal{M} \otimes \mathcal{N} \otimes \mathcal{E}$ is message consuming.

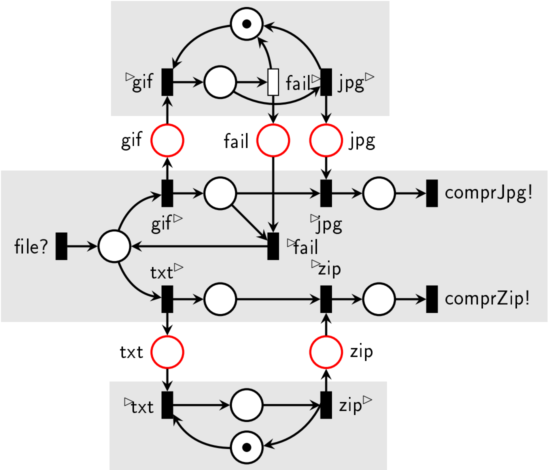
Incremental Verification: Example



Incremental Verification: Example



Incremental Verification: Example



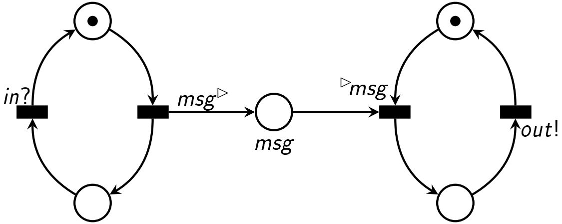
Necessarily Message Consuming (1)

$\mathcal{M}$ is **necessarily message consuming** w.r.t. B if for all channels $c \in B$ and for all reachable markings m of $\mathcal{M}$:

if c is not empty, then $\xrightarrow{\triangleright c}$ will "always" eventually be executed.

What means "always"?

Necessarily Message Consuming (2)

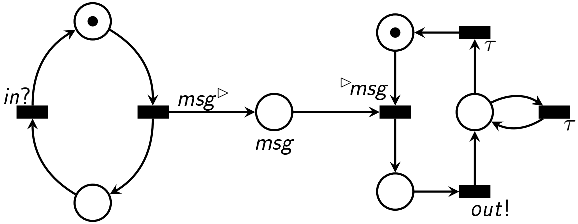


Then this is a run: $(\xrightarrow{msg^{\triangleright}} \xrightarrow{in?})^{\infty}$

Hence: This component is not consuming on all runs.

BUT: This component is consuming on all *weakly fair* runs.

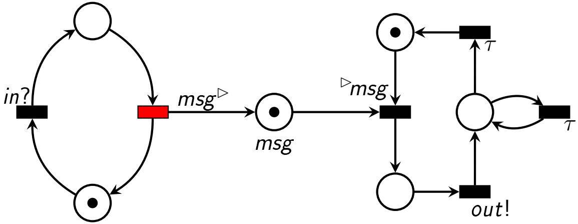
Necessarily Message Consuming (3)



Then this is a weakly fair run:

m^0

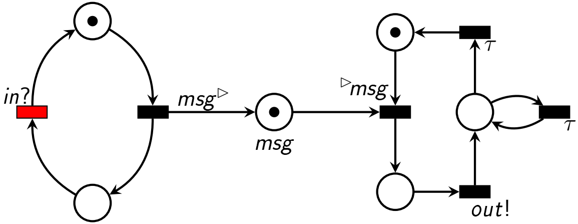
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\mathsf{msg}^{\triangleright}}$

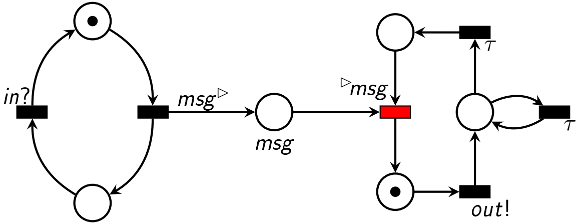
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\mathsf{msg}^{\triangleright}} \xrightarrow{\mathsf{in?}}$

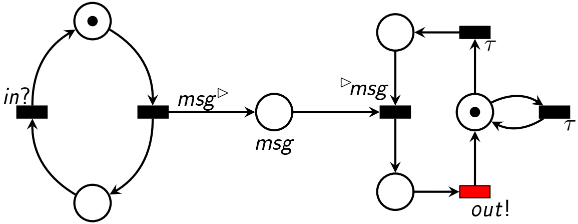
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\text{msg}^{\triangleright}} \xrightarrow{\text{in?}} \xrightarrow{{}^{\triangleright}\text{msg}}$

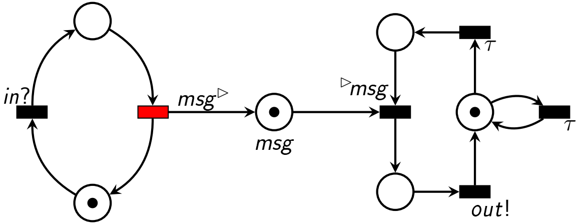
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\text{msg}^{\triangleright}} \xrightarrow{\text{in?}} \xrightarrow{{}^{\triangleright}\text{msg}} \xrightarrow{\text{out!}}$

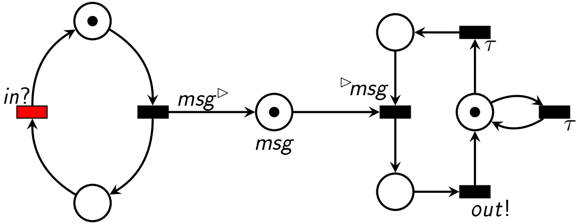
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\text{msg}^{\triangleright}} \xrightarrow{\text{in?}} \xrightarrow{{}^{\triangleright}\text{msg}} \xrightarrow{\text{out!}} (\xrightarrow{\text{msg}^{\triangleright}}$

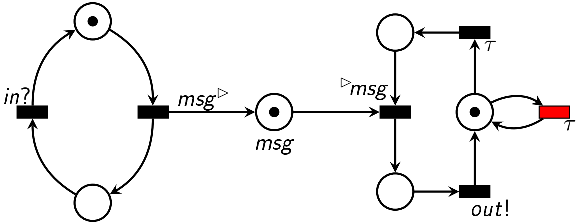
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\text{msg}^{\triangleright}} \xrightarrow{\text{in?}} \xrightarrow{{}^{\triangleright}\text{msg}} \xrightarrow{\text{out!}} (\xrightarrow{\text{msg}^{\triangleright}} \xrightarrow{\text{in?}}$

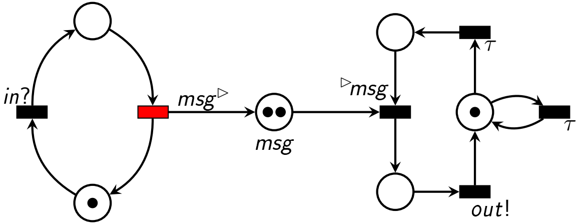
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\mathsf{msg}^{\triangleright}} \xrightarrow{\mathsf{in?}} \xrightarrow{{}^{\triangleright}\mathsf{msg}} \xrightarrow{\mathsf{out!}} (\xrightarrow{\mathsf{msg}^{\triangleright}} \xrightarrow{\mathsf{in?}} \xrightarrow{\tau})$

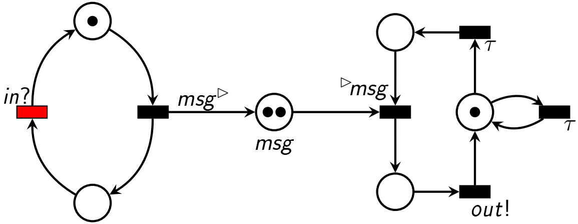
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\text{msg}^\triangleright} \xrightarrow{\text{in?}} \xrightarrow{{}^\triangleright\text{msg}} \xrightarrow{\text{out!}} (\xrightarrow{\text{msg}^\triangleright} \xrightarrow{\text{in?}} \xrightarrow{\tau}) (\xrightarrow{\text{msg}^\triangleright}$

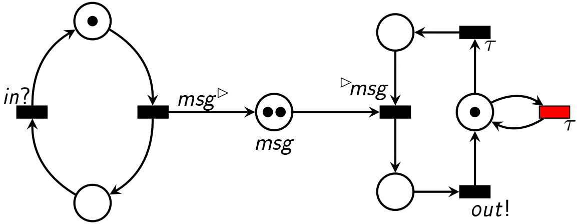
Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\mathsf{msg}^\triangleright} \xrightarrow{\mathsf{in?}} \xrightarrow{{}^\triangleright\mathsf{msg}} \xrightarrow{\mathsf{out!}} (\xrightarrow{\mathsf{msg}^\triangleright} \xrightarrow{\mathsf{in?}} \xrightarrow{\tau}) (\xrightarrow{\mathsf{msg}^\triangleright} \xrightarrow{\mathsf{in?}}$

Necessarily Message Consuming (3)



Then this is a weakly fair run:

$m^0 \xrightarrow{\mathsf{msg}^{\triangleright}} \xrightarrow{\mathsf{in?}} \xrightarrow{{}^{\triangleright}\mathsf{msg}} \xrightarrow{\mathsf{out!}} (\xrightarrow{\mathsf{msg}^{\triangleright}} \xrightarrow{\mathsf{in?}} \xrightarrow{\tau}) (\xrightarrow{\mathsf{msg}^{\triangleright}} \xrightarrow{\mathsf{in?}} \xrightarrow{\tau}) \ldots$

Hence: This component is not consuming on all weakly fair runs.

BUT: This component is consuming on all *observationally weakly fair* runs.

Additionally: A run can stop if it reaches a potential deadlock

Necessarily Message Consuming (4)

Definition:

$\mathcal{M}$ is **necessarily message consuming** w.r.t. B if for all channels $c \in B$ and for all reachable markings m of $\mathcal{M}$:

if c is not empty, then $\xrightarrow{\triangleright c}$ will eventually be executed on all observationally weakly fair runs starting in m.

Results:

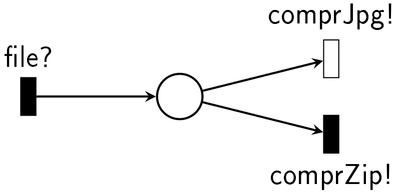
Decidability holds (relies on [Jancar 1990]).

Preservation by refinement holds.

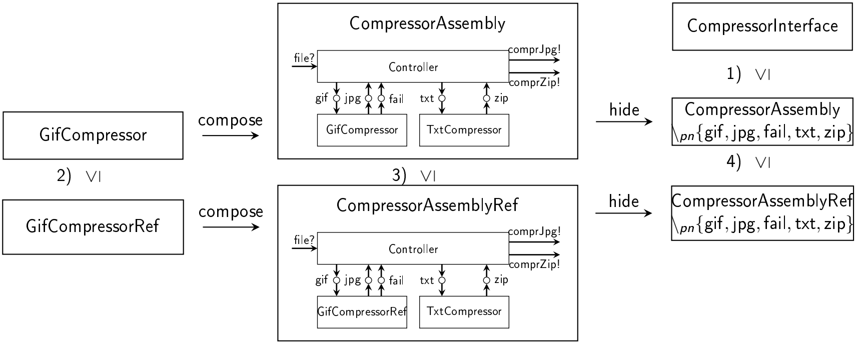
Incremental verification holds.

System Development Methodology (1)

An interface specification for the file compressing system



System Development Methodology (2)



Conclusion: Next steps

- Larger case-studies
- Multi-cast communication (e.g. broadcasting)
- “Message not provided” communication properties
- Tools
- Component model with ports and assume/guarantee reasoning
- Thread-based implementations