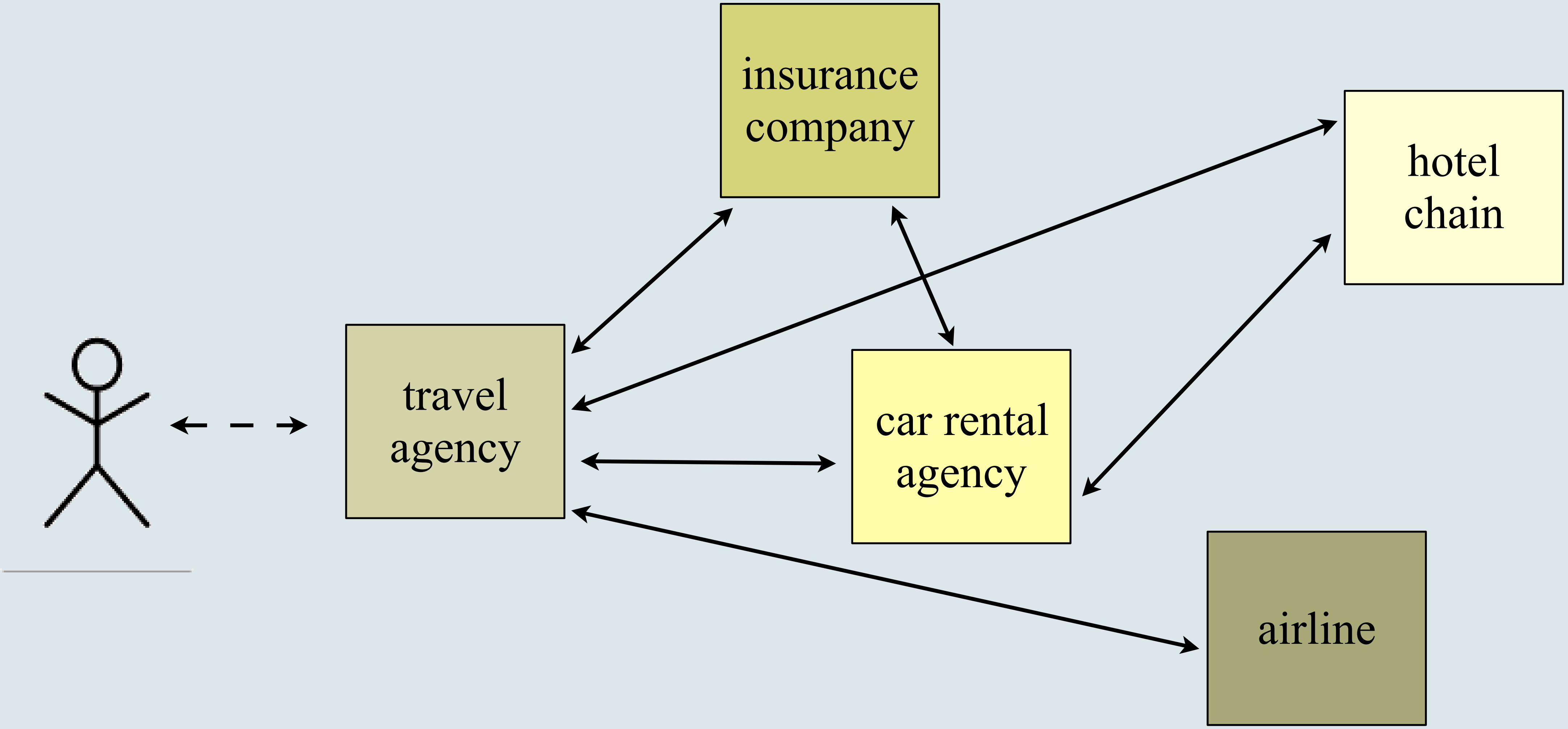
Conformance Checking of Service Behavior

W.M.P. van der Aalst[1,2] , M. Dumas[2] , C. Ouyang[2] , **A. Rozinat[1]** , and H.M.W. Verbeek[1]

[1] Eindhoven University of Technology
[2] Queensland University of Technology

Motivation

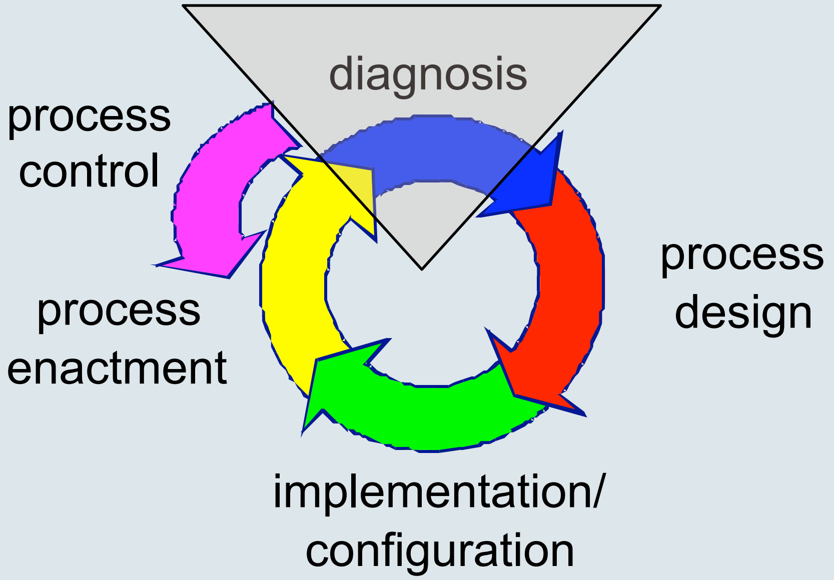


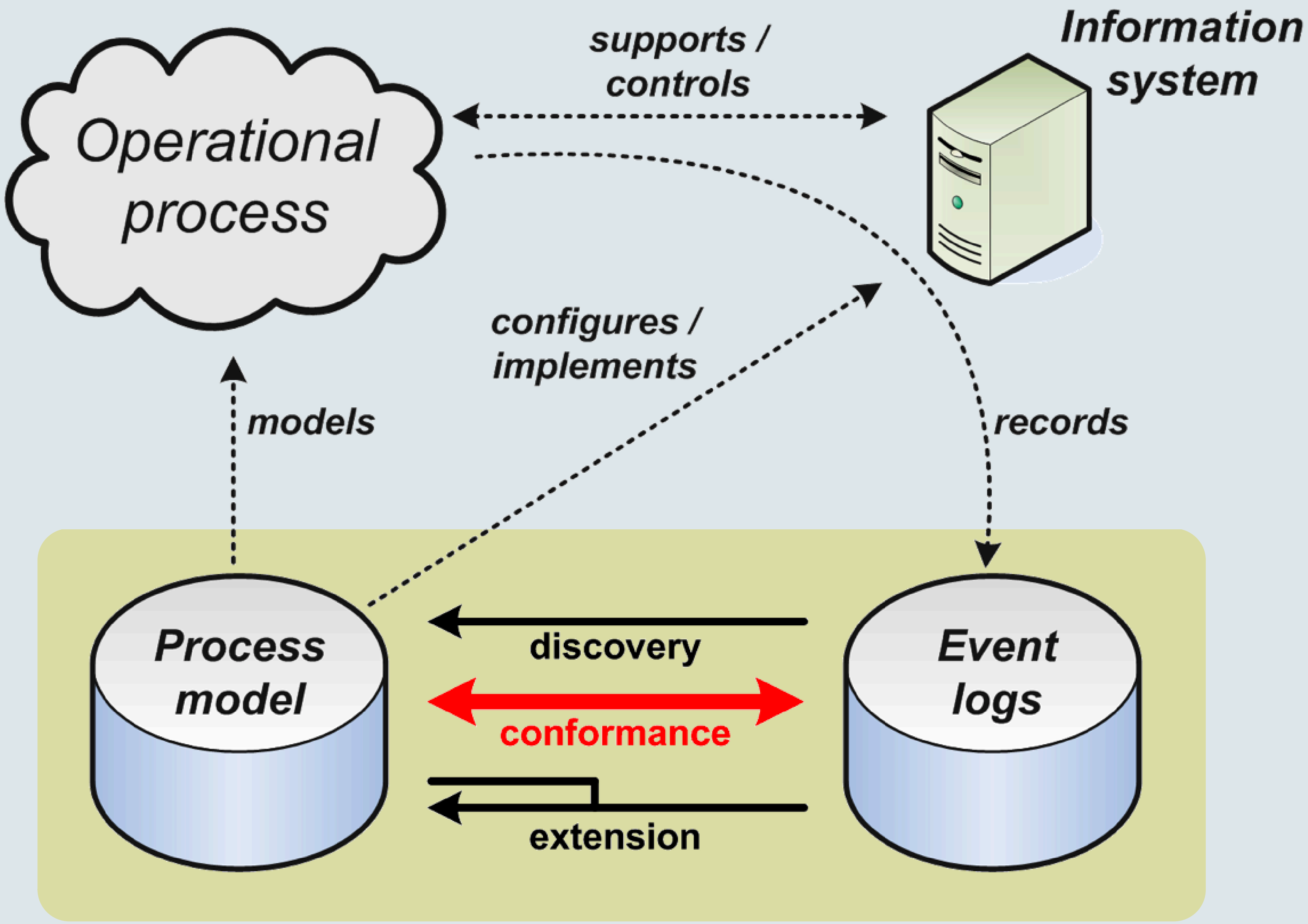
How to check whether interacting parties stick to what they agreed upon?

Outline

1. Introduction Process Mining
2. Conformance Checking
   1. Fitness
   2. Appropriateness
3. Application to Web Services
4. Feasibility Study
5. Conclusion

1. Introduction Process Mining



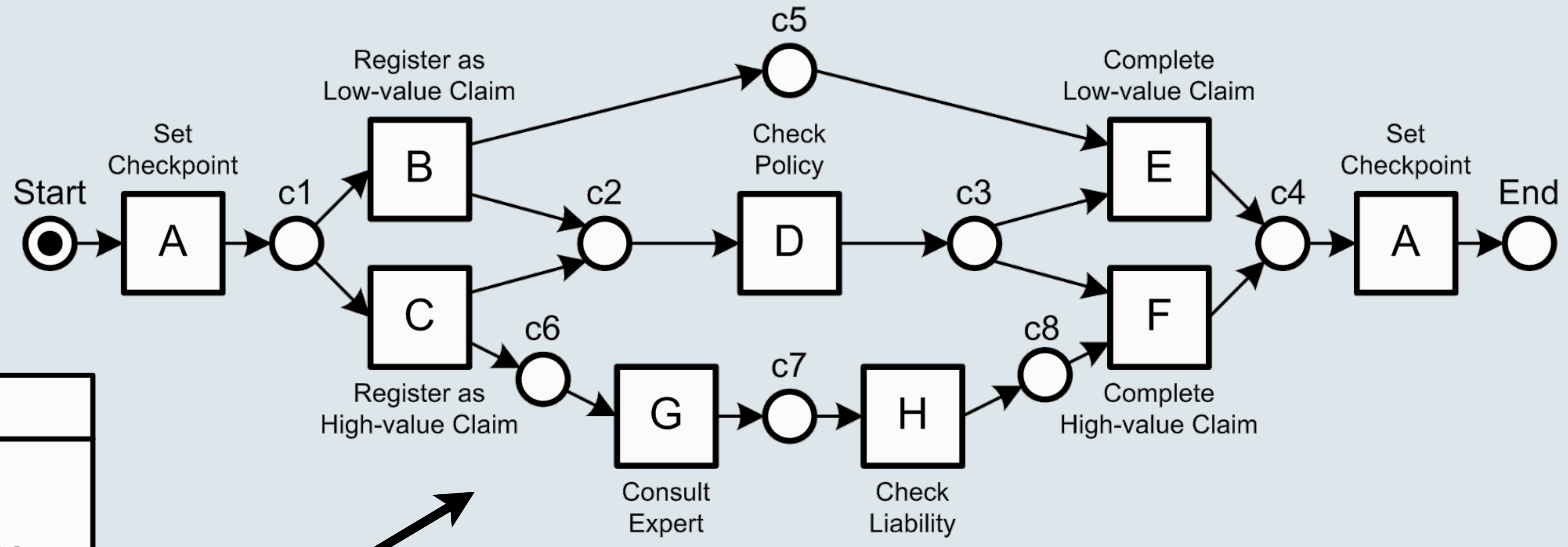


- analyze run-time data
- possibly relate to existing models and/or requirements
- process mining!

2. Conformance Checking

Objectives:

- quantitatively measure conformance (i.e., metrics)
- locate deviations



No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

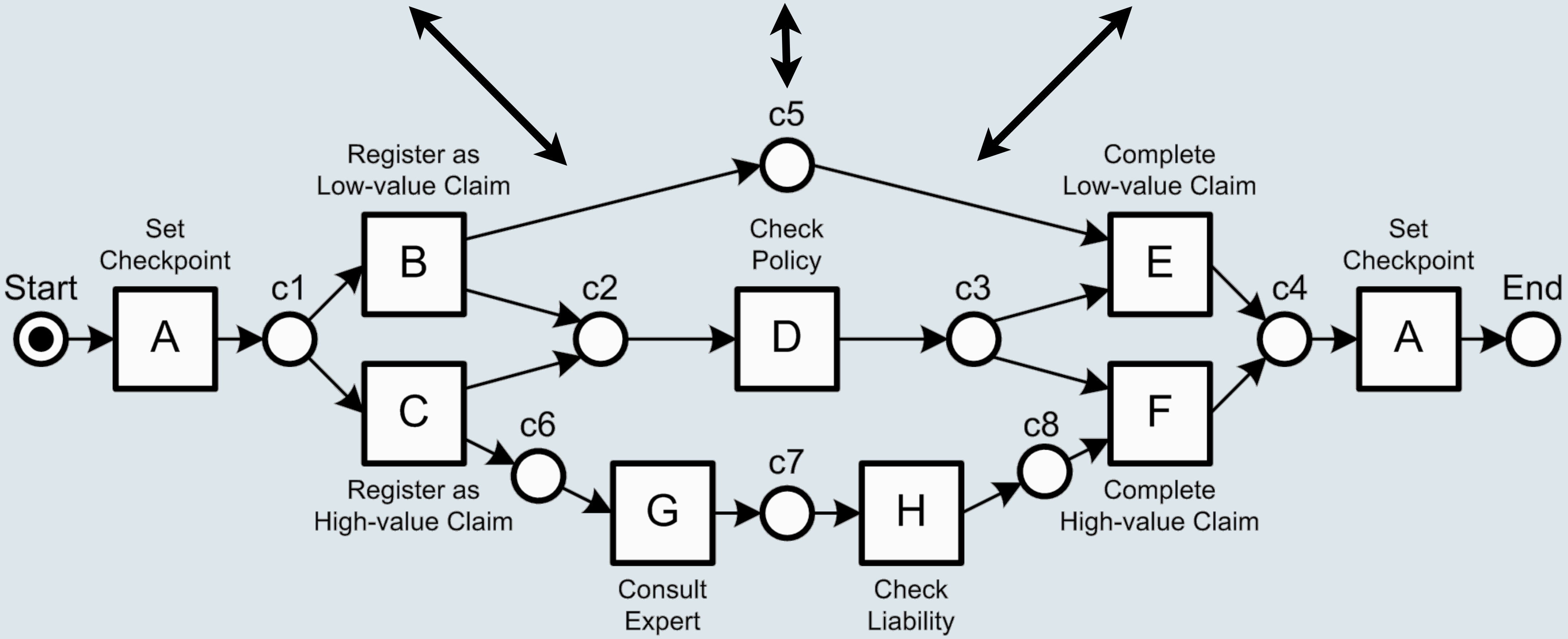
Conformance?

2.1 Conformance Checking - Fitness

No. of Instances	Log Traces
4070	ABDEA
245	ACDGHFA
56	ACGDHFA

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

No. of Instances	Log Traces
24	BDE
7	AABHF
15	CHF
6	ADBE
1	ACBGDFAA
8	ABEDA

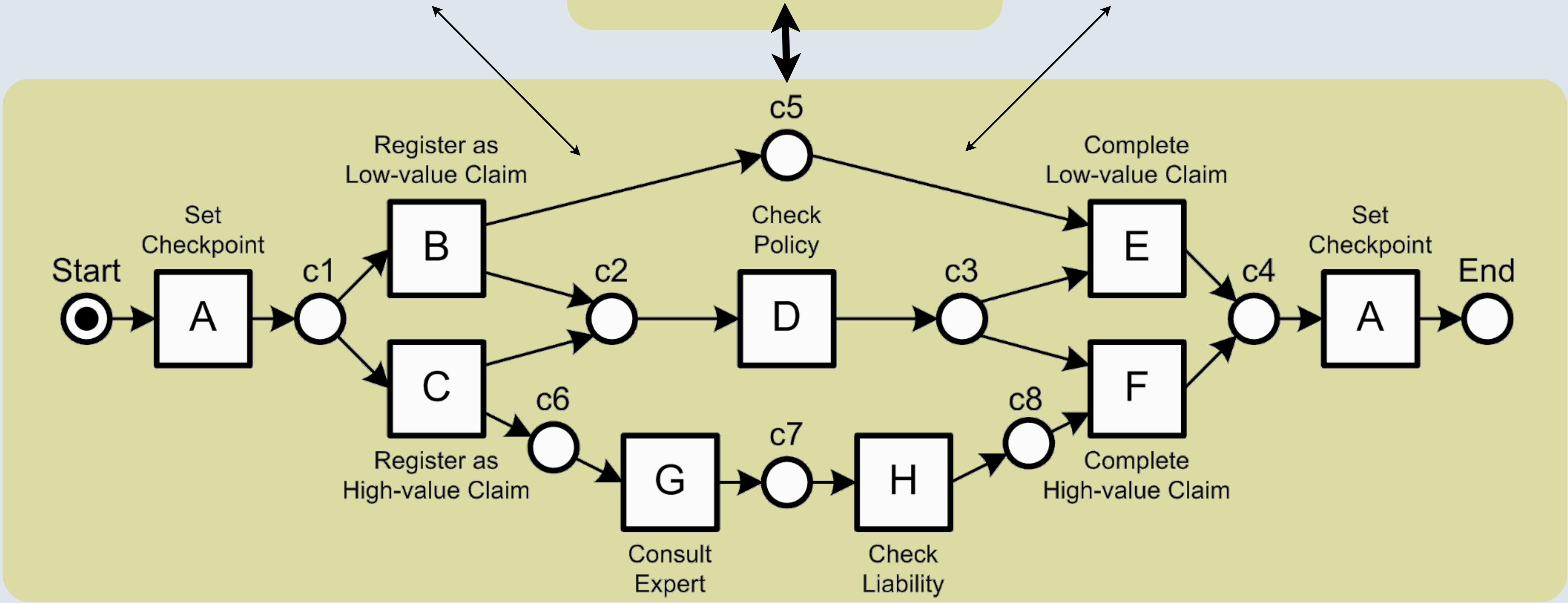


2.1 Conformance Checking - Fitness

No. of Instances	Log Traces
4070	ABDEA
245	ACDGHFA
56	ACGDHFA

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

No. of Instances	Log Traces
24	BDE
7	AABHF
15	CHF
6	ADBE
1	ACBGDFAA
8	ABEDA



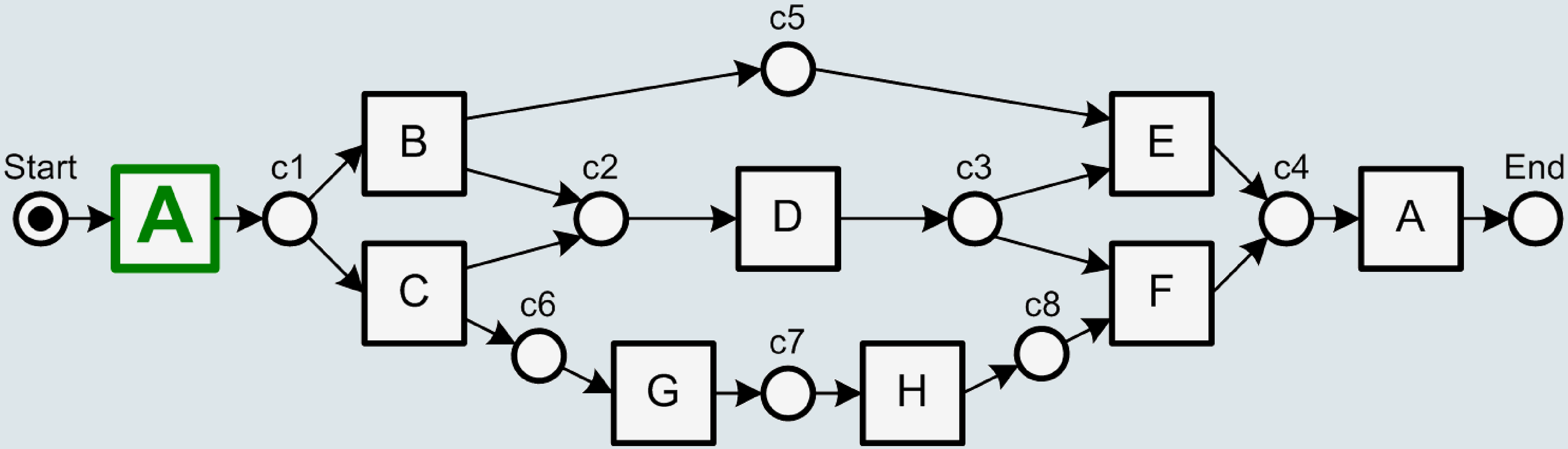
2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

missing tokens = 0 consumed tokens = 0

remaining tokens = 0 produced tokens = 1

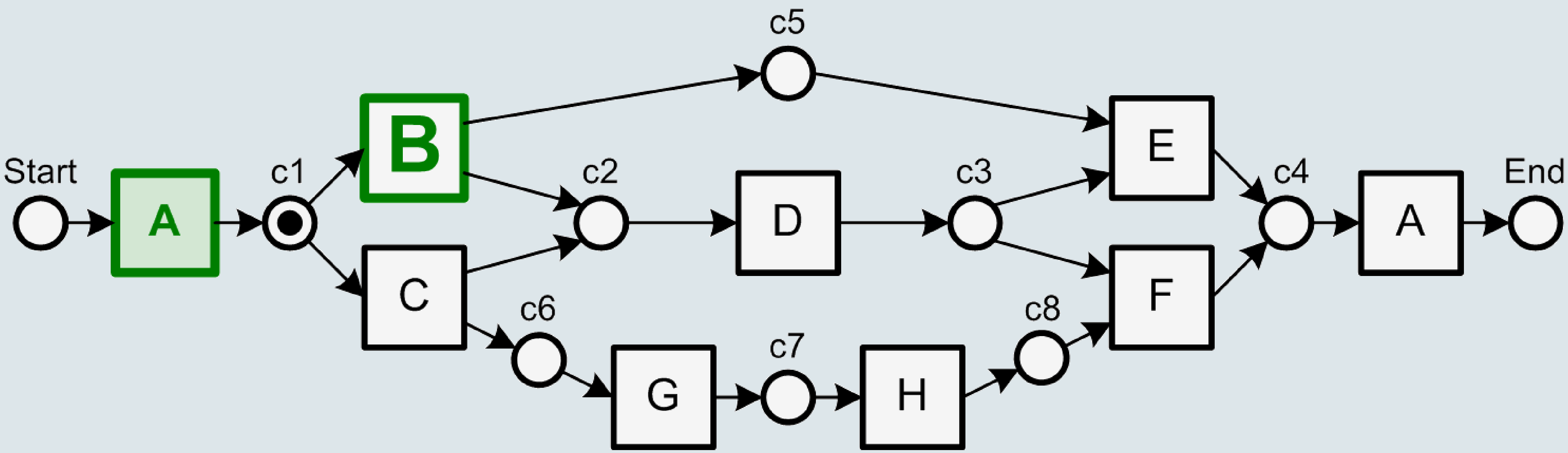


2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$\boxed{f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)}$$

missing tokens	$= 0$	consumed tokens	$= 1$
remaining tokens	$= 0$	produced tokens	$= 2$



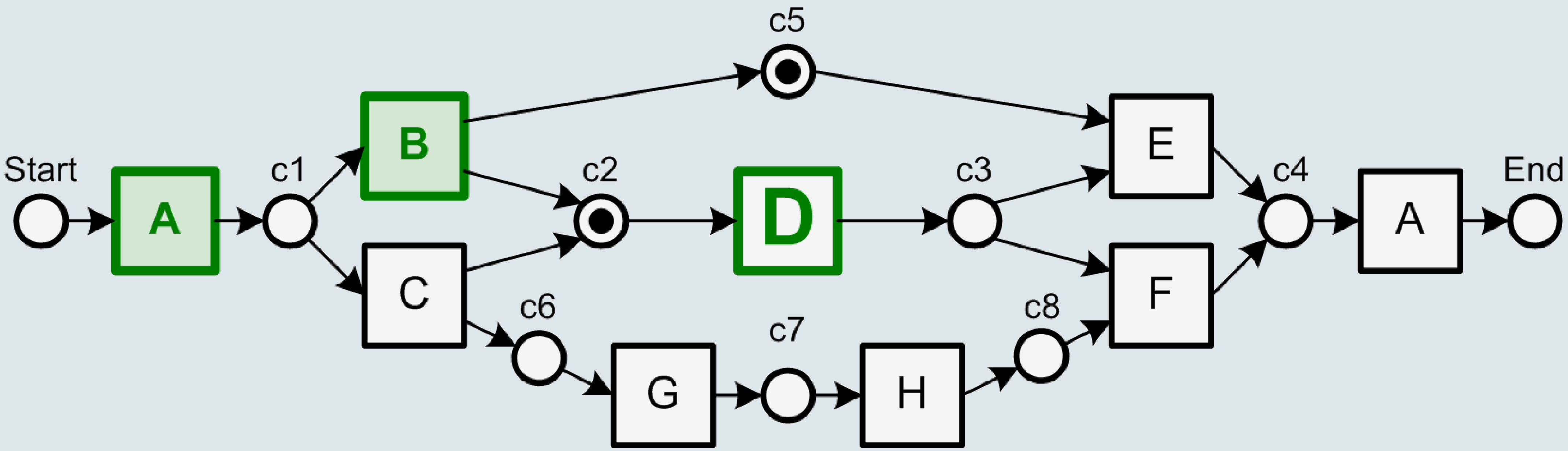
2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

missing tokens $= 0$ consumed tokens $= 2$

remaining tokens $= 0$ produced tokens $= 4$

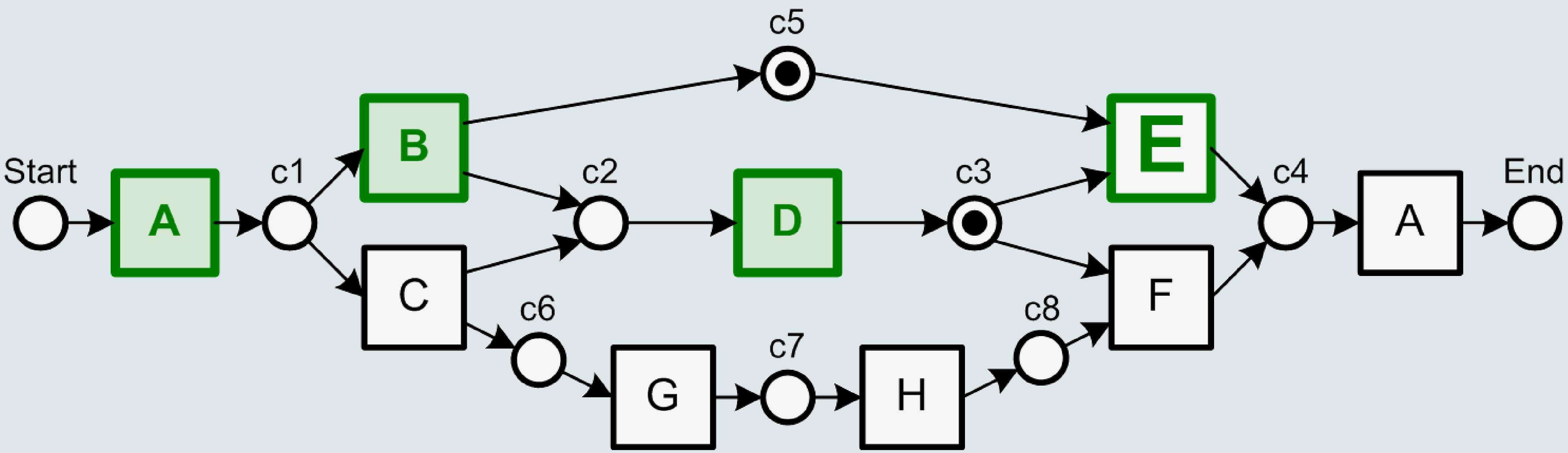


2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

missing tokens	= 0	consumed tokens	= 3
remaining tokens	= 0	produced tokens	= 5

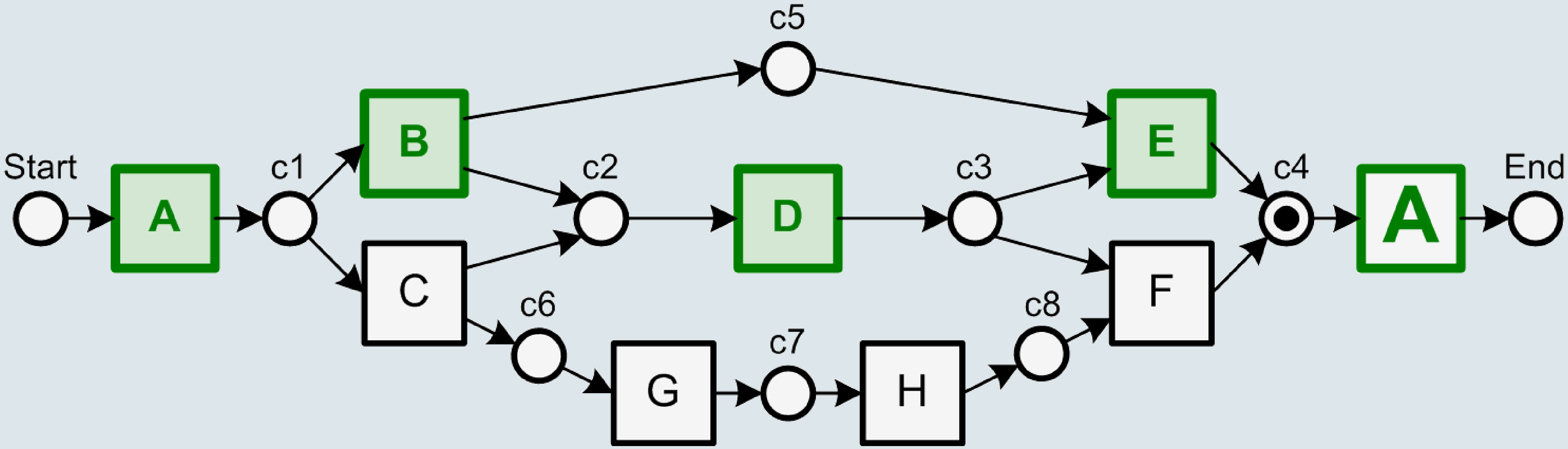


2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	→ **ABDEA**
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

missing tokens	$= 0$	consumed tokens	$= 5$
remaining tokens	$= 0$	produced tokens	$= 6$



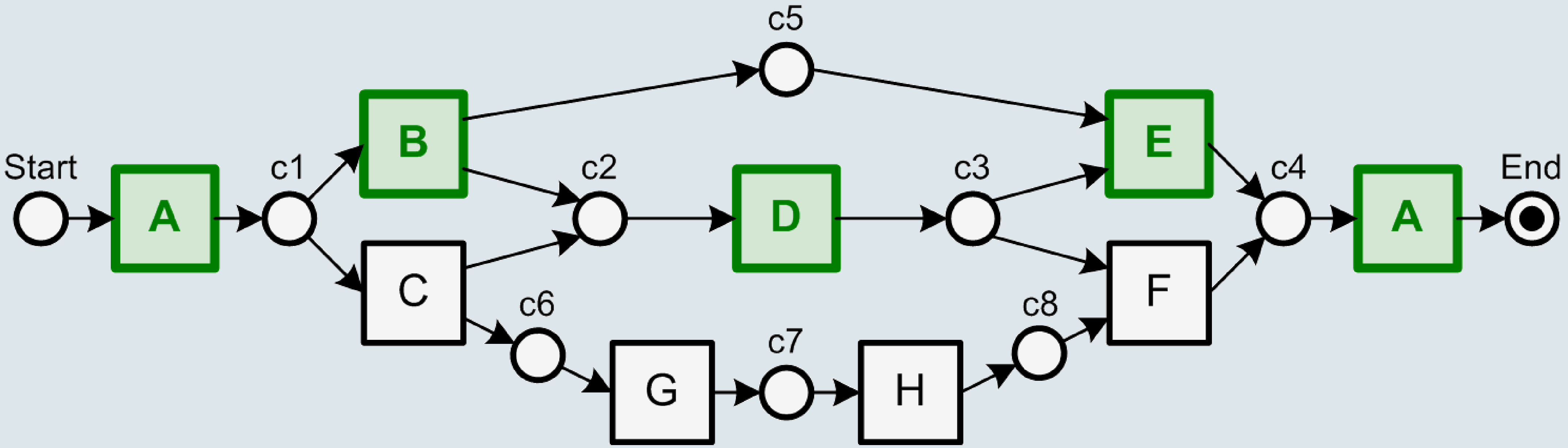
2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	→ **ABDEA**
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

missing tokens = 0 consumed tokens = 6

remaining tokens = 0 produced tokens = 7



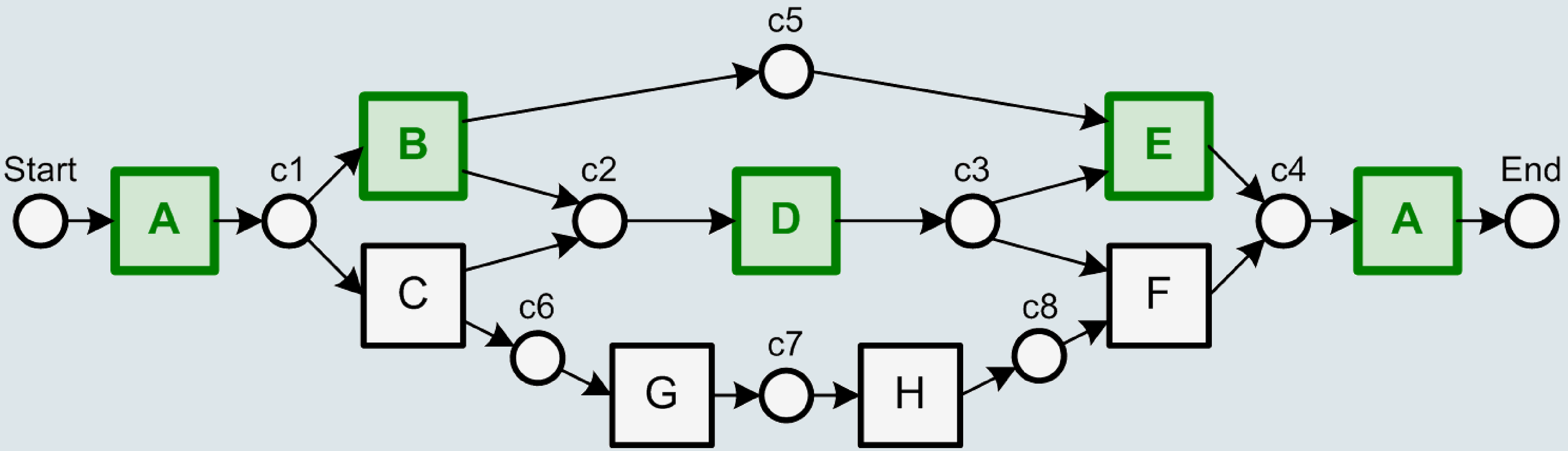
$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	→ **ABDEA**
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}(1 - \frac{0+}{(1207 \cdot 7)+}) + \frac{1}{2}(1 - \frac{0+}{(1207 \cdot 7)+})$$

$$\text{missing tokens} = 0 \qquad \text{consumed tokens} = 7$$
$$\text{remaining tokens} = 0 \qquad \text{produced tokens} = 7$$



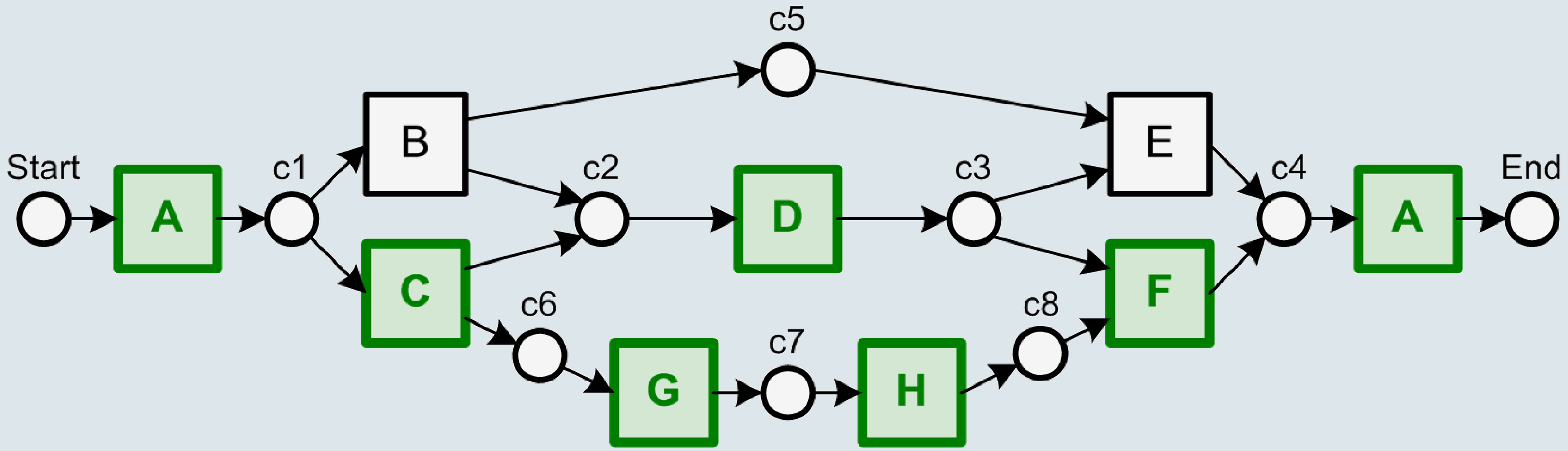
$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	→ **ACDGHFA**
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}(1 - \frac{0+0+}{(1207 \cdot 7)+(145 \cdot 9)+}) + \frac{1}{2}(1 - \frac{0+0+}{(1207 \cdot 7)+(145 \cdot 9)+})$$

missing tokens $= 0$ consumed tokens $= 9$
remaining tokens $= 0$ produced tokens $= 9$



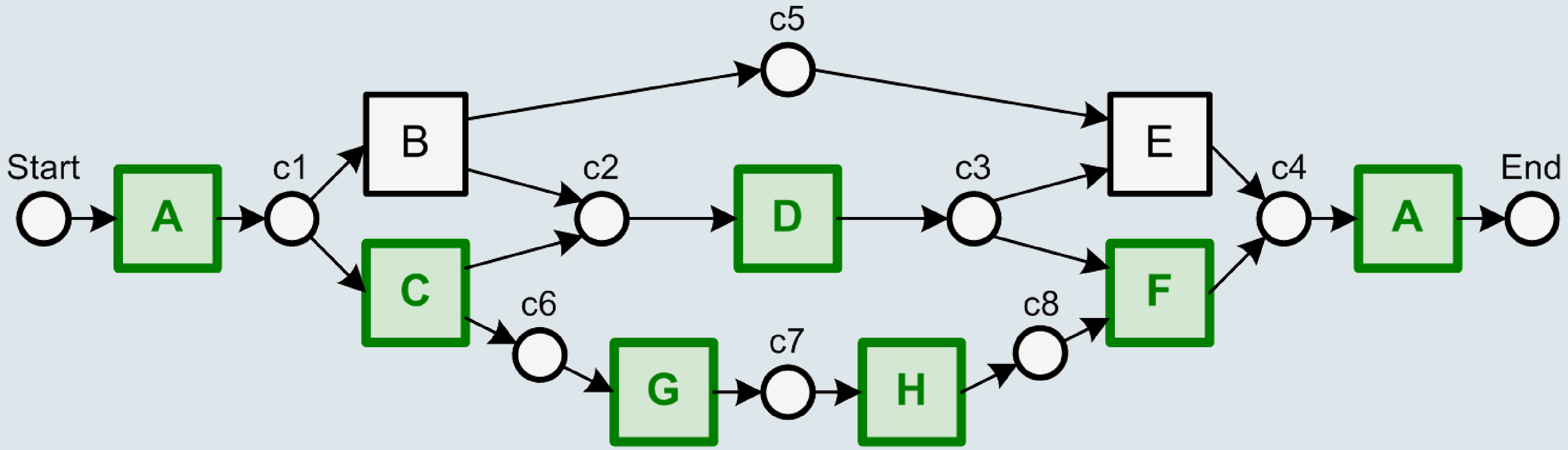
$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	→ **ACGDHFA**
23	ACHDFA
28	ACDHFA

$$\begin{aligned} f =& \frac{1}{2}(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}) \\ &+ \frac{1}{2}(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}) \end{aligned}$$

$$\begin{aligned} &\text{missing tokens} &&= 0 \qquad &&\text{consumed tokens} &&= 9 \\ &\text{remaining tokens} &&= 0 \qquad &&\text{produced tokens} &&= 9 \end{aligned}$$



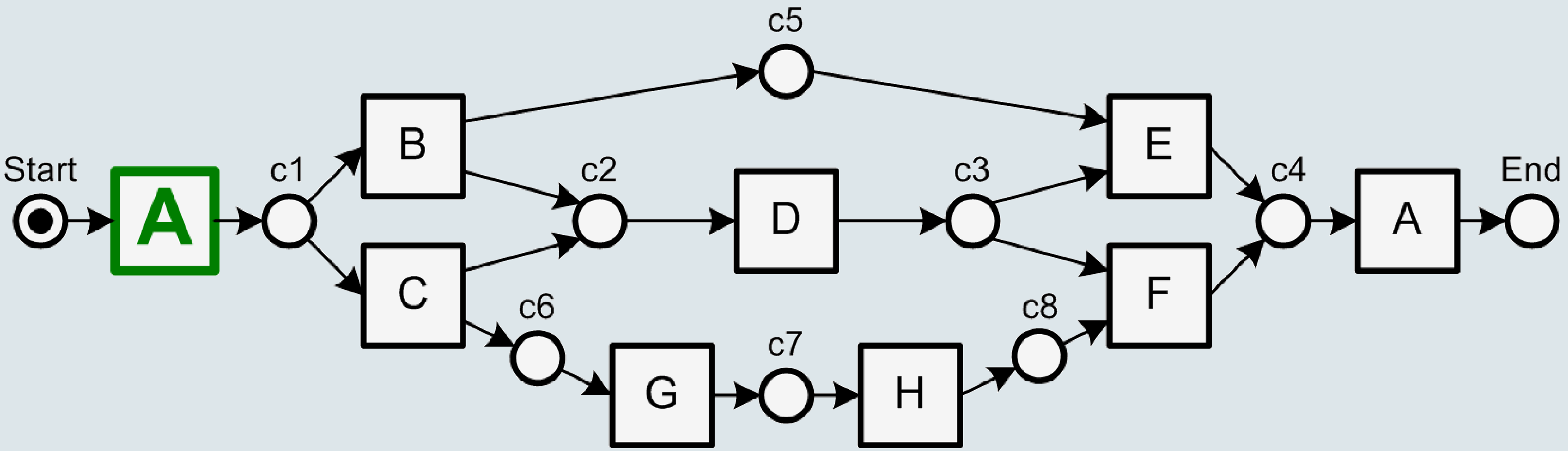
$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$\begin{aligned} f =& \frac{1}{2}(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9)+}) \\ &+ \frac{1}{2}(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9)+}) \end{aligned}$$

$$\begin{aligned} &\text{missing tokens} &&= 0 \qquad &&\text{consumed tokens} &&= 0 \\ &\text{remaining tokens} &&= 0 \qquad &&\text{produced tokens} &&= 1 \end{aligned}$$



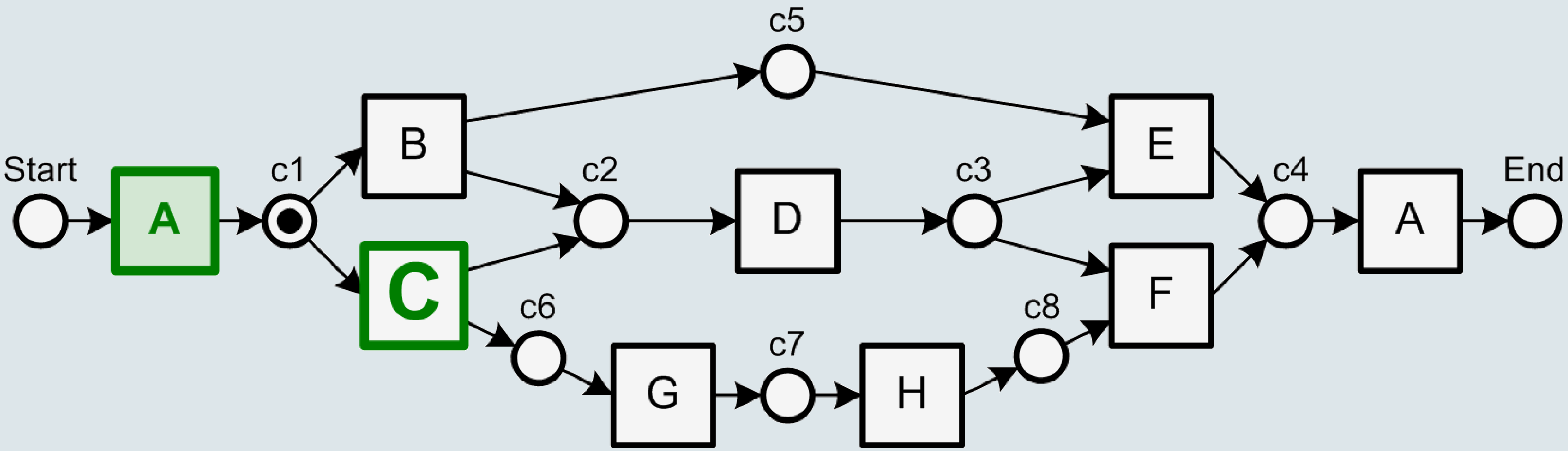
$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$f = \frac{1}{2}(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9)+}) + \frac{1}{2}(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9)+})$$

$$\begin{array}{llll} \text{missing tokens} & = 0 & \text{consumed tokens} & = 1 \\ \text{remaining tokens} & = 0 & \text{produced tokens} & = 2 \end{array}$$



$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

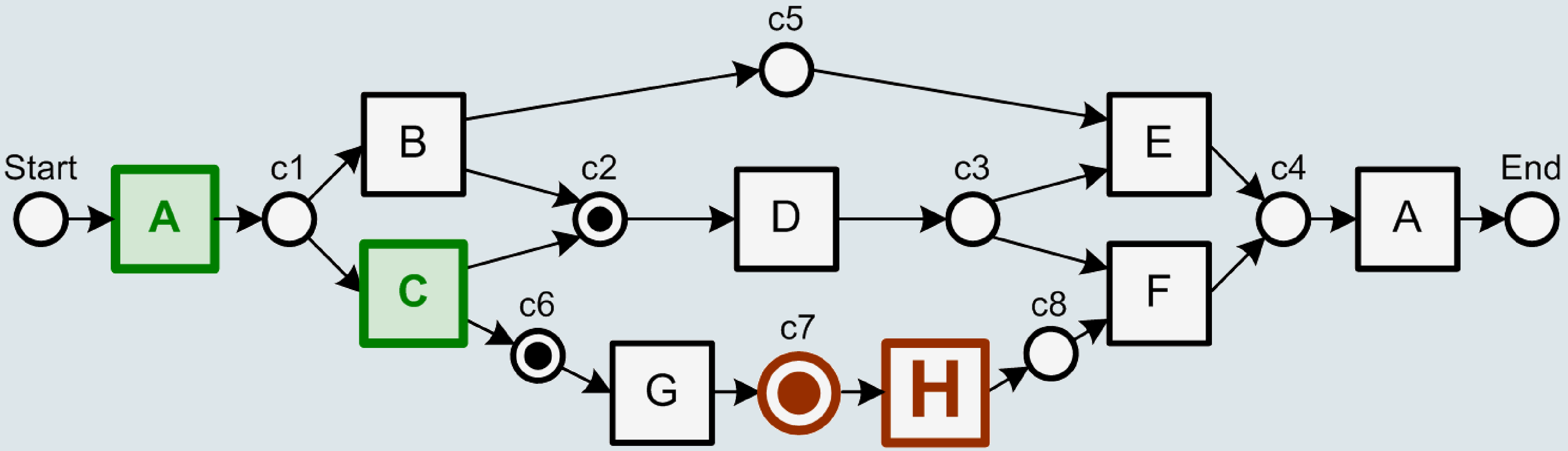
2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$f = \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}\right) + \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}\right)$$

missing tokens = 1 consumed tokens = 2

remaining tokens = 0 produced tokens = 4



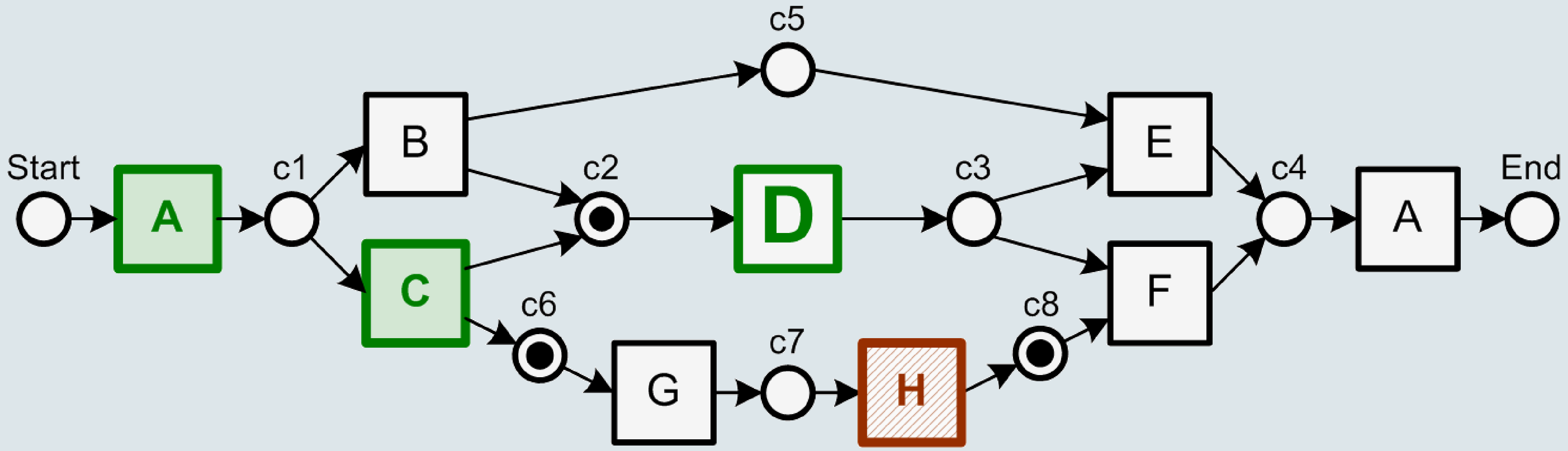
$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$\begin{aligned} f =& \frac{1}{2}(1 - \frac{0+0+0+}{(1207\cdot 7)+((145+56)\cdot 9)+}) \\ &+ \frac{1}{2}(1 - \frac{0+0+0+}{(1207\cdot 7)+((145+56)\cdot 9)+}) \end{aligned}$$

$$\begin{aligned} &\text{missing tokens} &= 1 \qquad &\text{consumed tokens} &= 3 \\ &\text{remaining tokens} &= 0 \qquad &\text{produced tokens} &= 5 \end{aligned}$$



$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

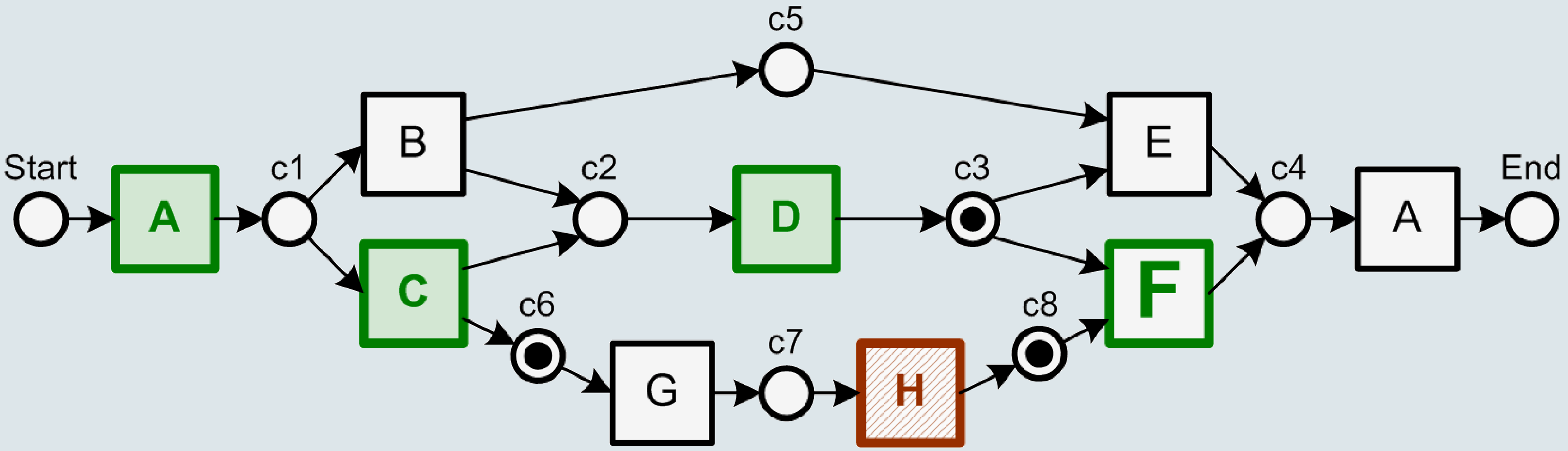
2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$\begin{aligned} f =& \frac{1}{2}(1 - \frac{0+0+0+}{(1207\cdot 7)+((145+56)\cdot 9)+}) \\ &+ \frac{1}{2}(1 - \frac{0+0+0+}{(1207\cdot 7)+((145+56)\cdot 9)+}) \end{aligned}$$

missing tokens $= 1$ consumed tokens $= 4$

remaining tokens $= 0$ produced tokens $= 6$



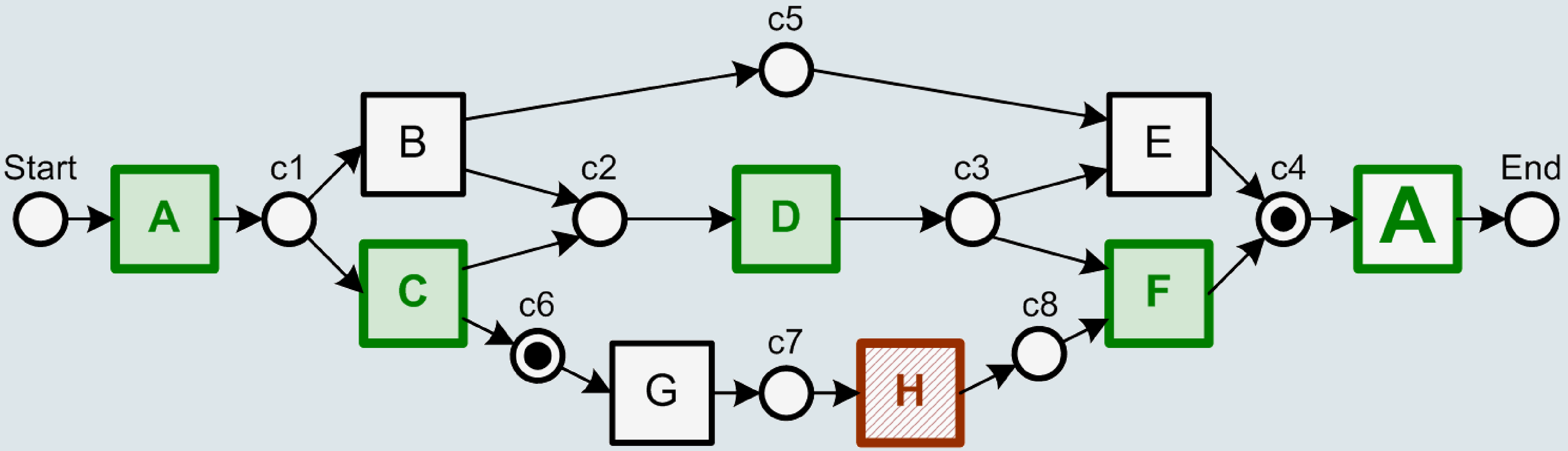
$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$f = \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}\right) + \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207 \cdot 7) + ((145+56) \cdot 9) +}\right)$$

missing tokens	$= 1$	consumed tokens	$= 6$
remaining tokens	$= 0$	produced tokens	$= 7$



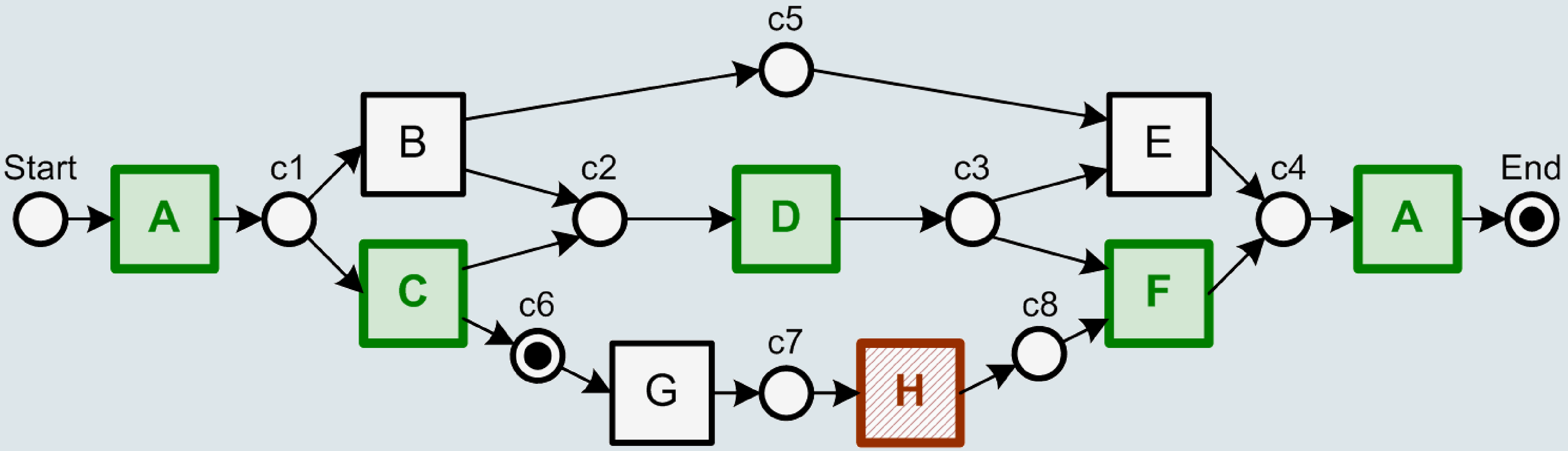
$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$\begin{aligned} f =& \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207\cdot 7)+((145+56)\cdot 9)+}\right) \\ &+ \frac{1}{2}\left(1 - \frac{0+0+0+}{(1207\cdot 7)+((145+56)\cdot 9)+}\right) \end{aligned}$$

missing tokens $= 1$ consumed tokens $= 7$
remaining tokens $= 0$ produced tokens $= 8$



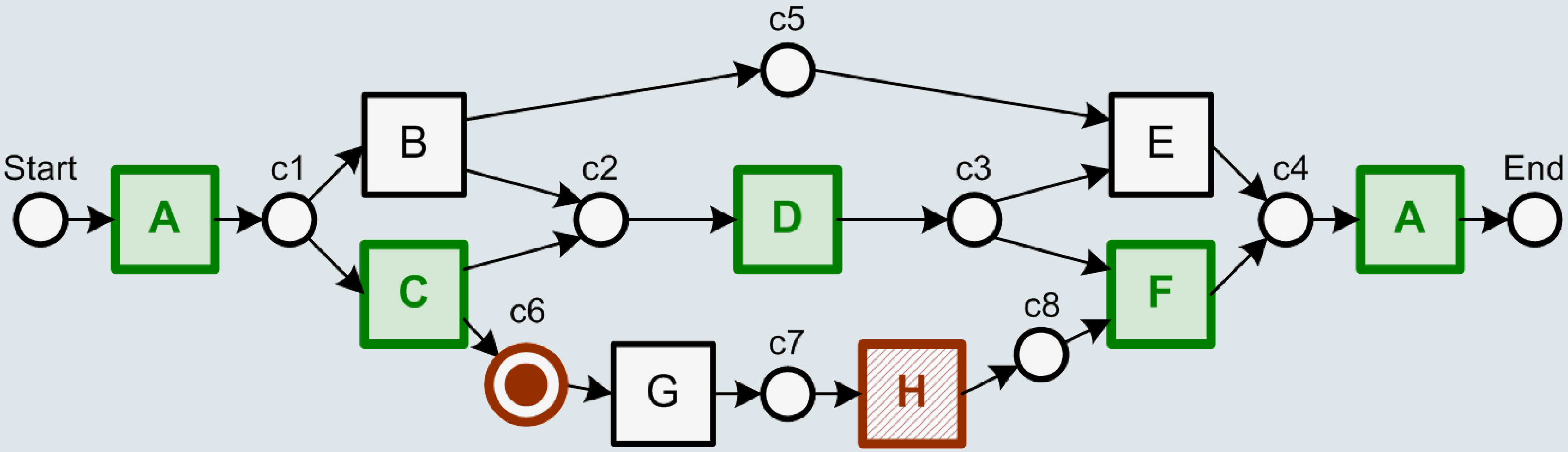
$$f=\frac{1}{2}(1-\frac{\sum_{i=1}^{k}n_i m_i}{\sum_{i=1}^{k}n_i c_i})+\frac{1}{2}(1-\frac{\sum_{i=1}^{k}n_i r_i}{\sum_{i=1}^{k}n_i p_i})$$

2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

$$\begin{aligned} f =& \frac{1}{2}(1-\frac{0+0+0+(23\cdot 1)+}{(1207\cdot 7)+((145+56)\cdot 9)+(23\cdot 8)+}) \\ &+\frac{1}{2}(1-\frac{0+0+0+(23\cdot 1)+}{(1207\cdot 7)+((145+56)\cdot 9)+(23\cdot 8)+}) \end{aligned}$$

missing tokens	$= 1$	consumed tokens	$= 8$
remaining tokens	$= 1$	produced tokens	$= 8$



$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

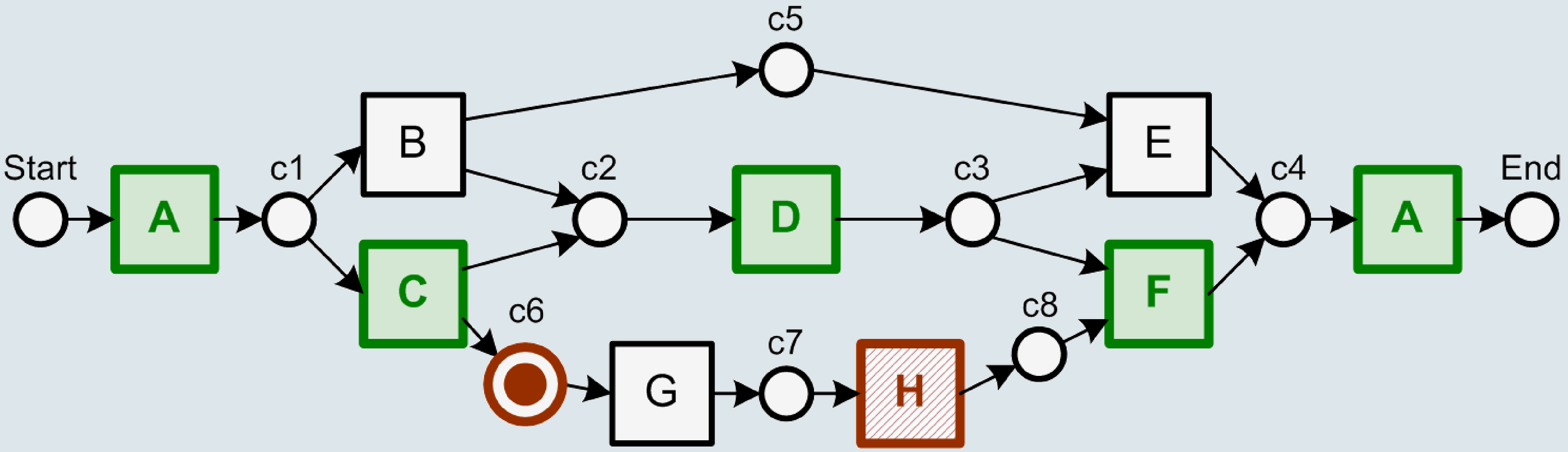
2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	→ **ACDHFA**

$$f = \frac{1}{2}(1 - \frac{0+0+0+(23\cdot 1)+(28\cdot 1)}{(1207\cdot 7)+((145+56)\cdot 9)+((23+28)\cdot 8)}) + \frac{1}{2}(1 - \frac{0+0+0+(23\cdot 1)+(28\cdot 1)}{(1207\cdot 7)+((145+56)\cdot 9)+((23+28)\cdot 8)})$$

missing tokens $= 1$ consumed tokens $= 8$

remaining tokens $= 1$ produced tokens $= 8$

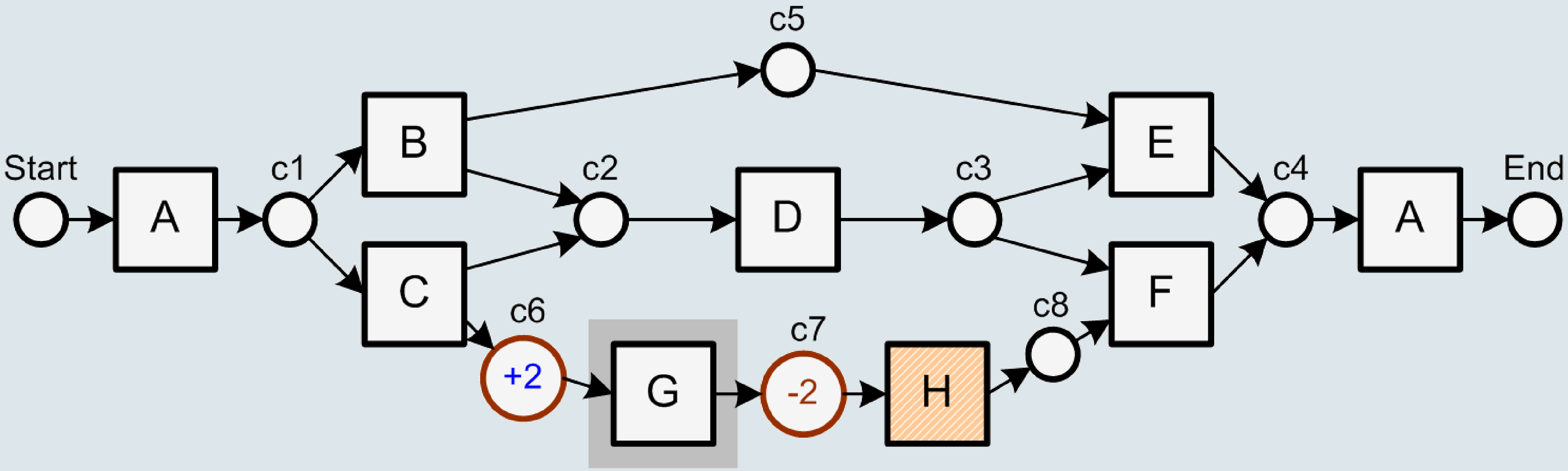


2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = 1 - \frac{23 + 28}{(1207 \cdot 7) + ((145 + 56) \cdot 9) + ((23 + 28) \cdot 8)} \approx 0.995$$

$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$



2.1 Measuring fitness: Log replay analysis

No. of Instances	Log Traces
4070	ABDEA
245	ACDGHFA
56	ACGDHFA

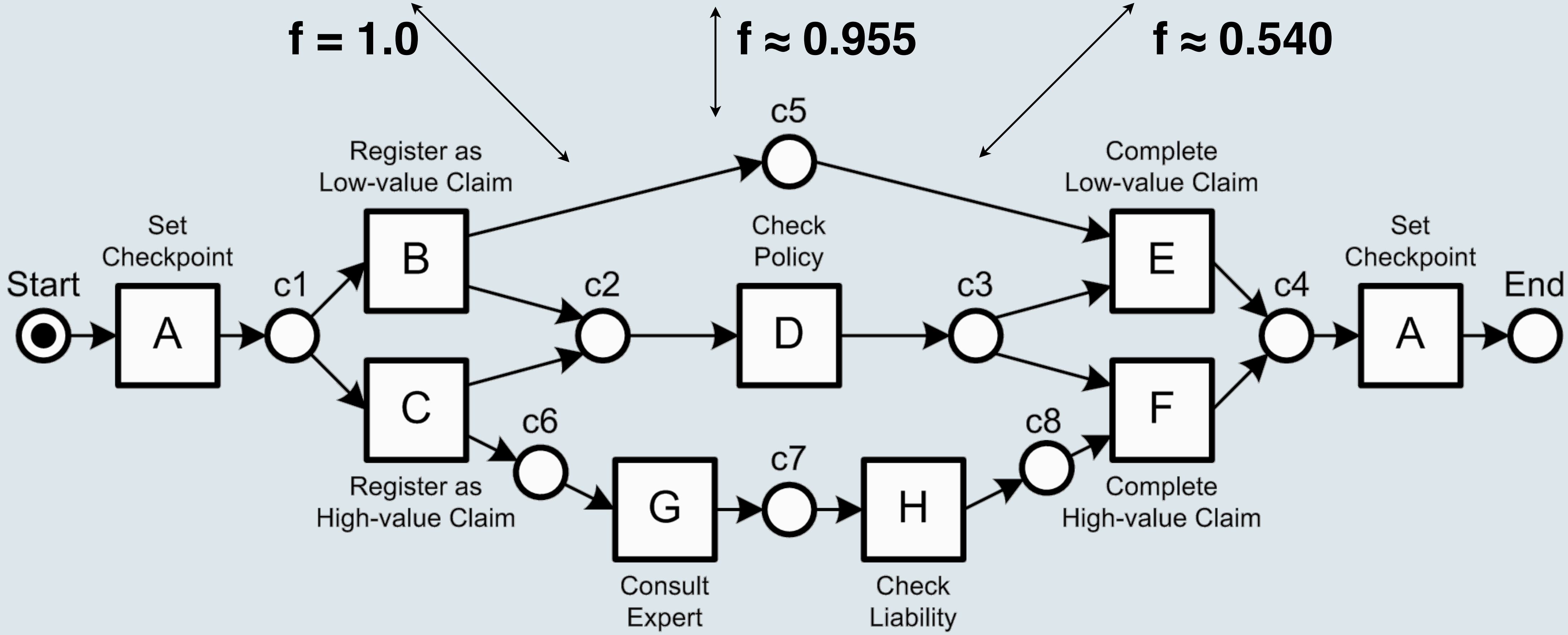
No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

No. of Instances	Log Traces
24	BDE
7	AABHF
15	CHF
6	ADBE
1	ACBGDFAA
8	ABEDA

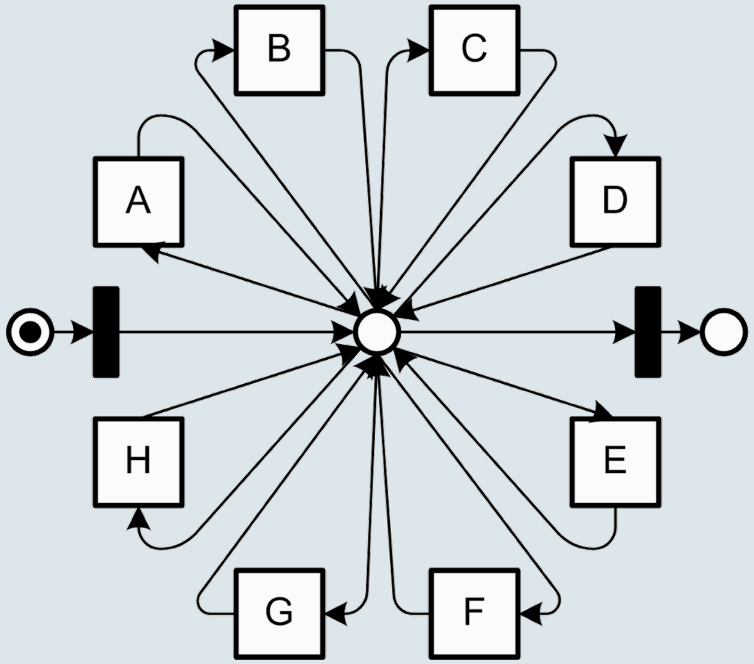
f = 1.0

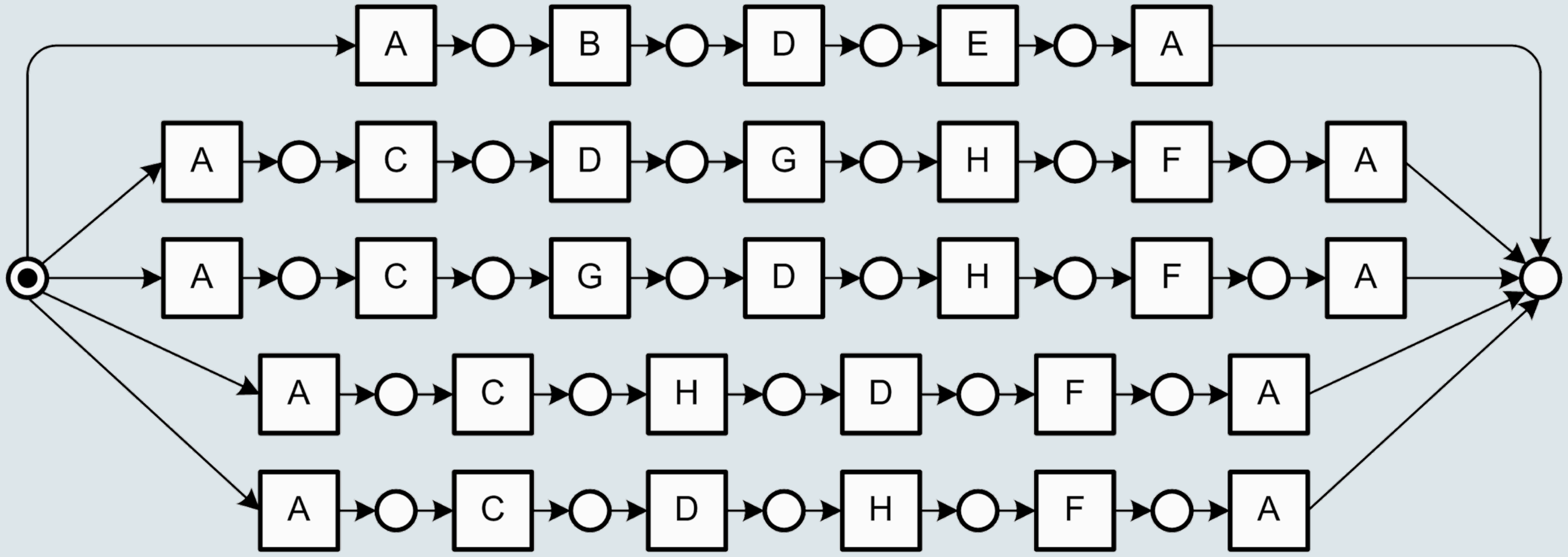
f ≈ 0.955

f ≈ 0.540



2.2 Conformance Checking - Appropriateness





100 % fitness
but not sufficiently
specific from
behavioral point of view.

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

100 % fitness
but not represented in
structurally suitable way.

3. Application to Web Services

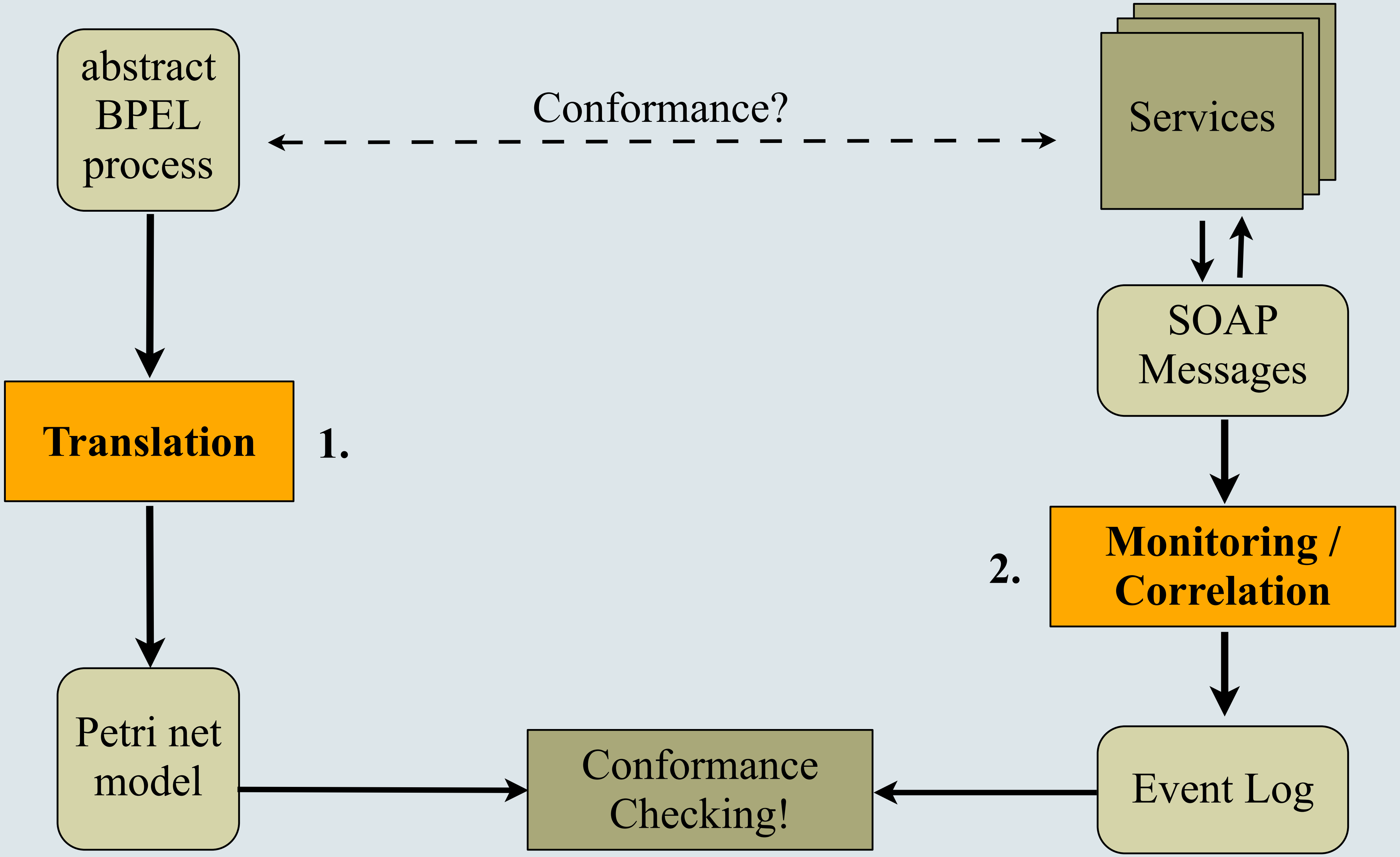
- Service-oriented systems are composed of services that are:
  - independently developed and operated
  - interact with one another exclusively through message exchanges

➡ Expected behavior may deviate

- e.g., service receives reply of the wrong type, messages may be received in the wrong order, ...

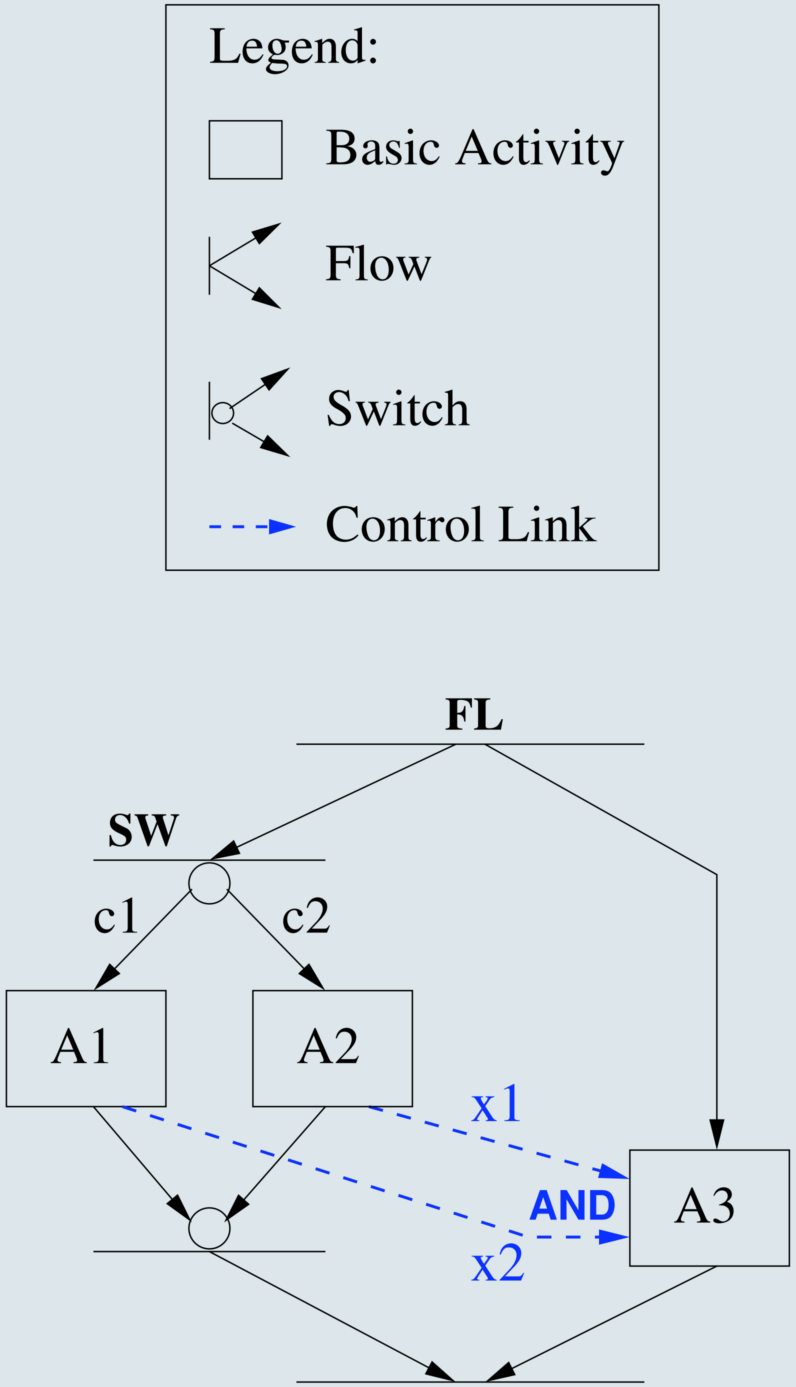
Question: “Do all services in a service-oriented system operate as expected?”

3. Application to Web Services - Approach



3.1 From BPEL to Petri Nets

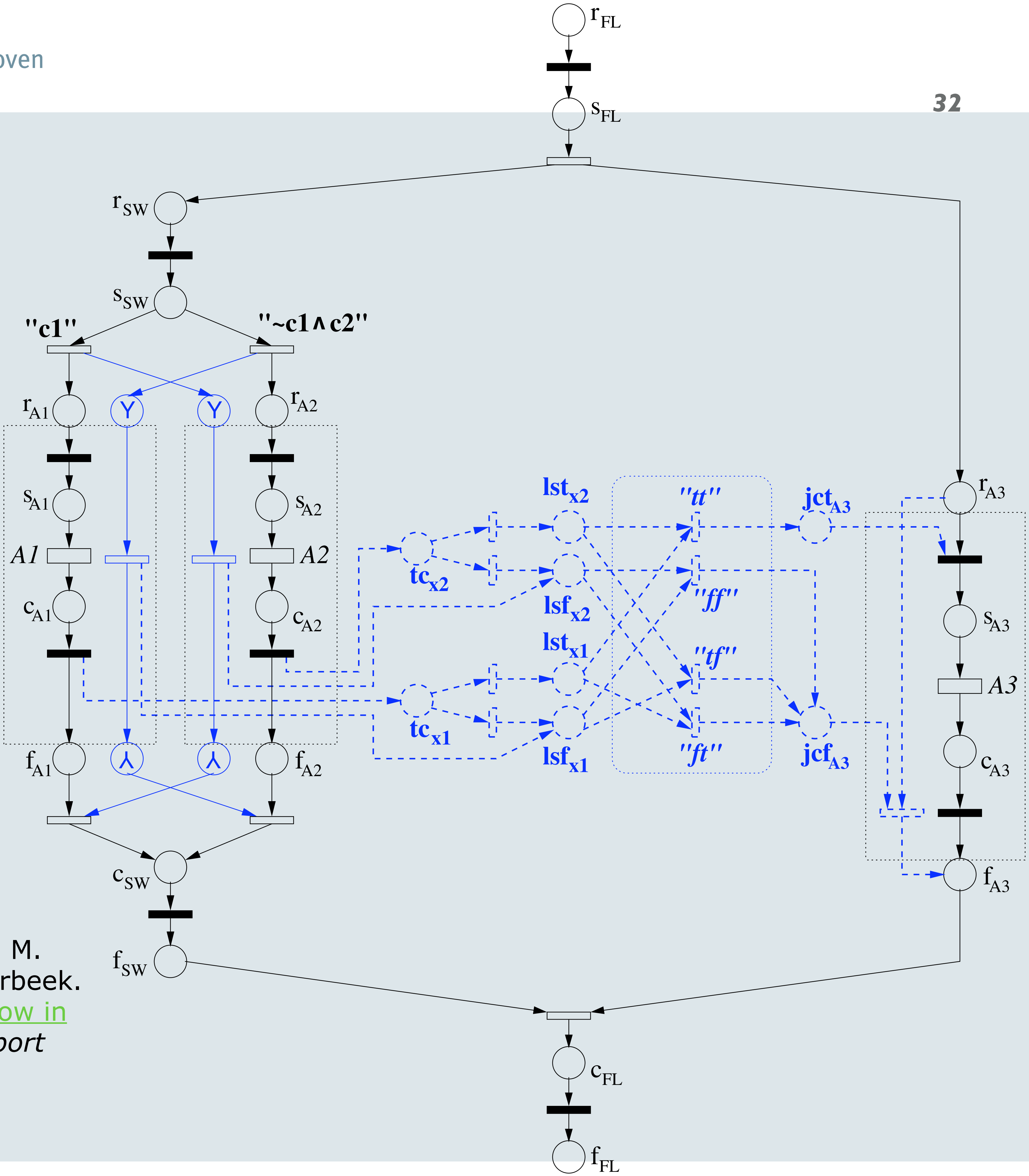
```
<process name="unreachableTask"
    targetNamespace="http://samples.otn.com"
    suppressJoinFailure="yes"
    xmlns:tns="http://samples.otn.com"
    xmlns:services="http://services.otn.com"
    xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
  <flow name="FL" suppressJoinFailure="yes">
  <links>
    <link name="x1"/>
    <link name="x2"/>
  </links>
  <switch name="SW">
    <case>
      <invoke name="A1">
        <sources>  <source linkName="x1"/>   </sources>
      </invoke>
    </case>
    <otherwise>
      <invoke name="A2">
        <sources>  <source linkName="x2"/>   </sources>
      </invoke>
    </otherwise>
  </switch>
  <invoke name = "A3">
   <targets>
    <joinCondition>
      bpws:getLinkStatus(‘x1’) and bpws:getLinkStatus(‘x2’)
    </joinCondition>
    <target linkName="x1"/>
    <target linkName="x2"/>
   </targets>
  </invoke>
  </flow>
</process>
```



3.1 From BPEL to Petri Nets

Translation is *feature-complete*:

➡ maps all BPEL aspects on a Petri-net model

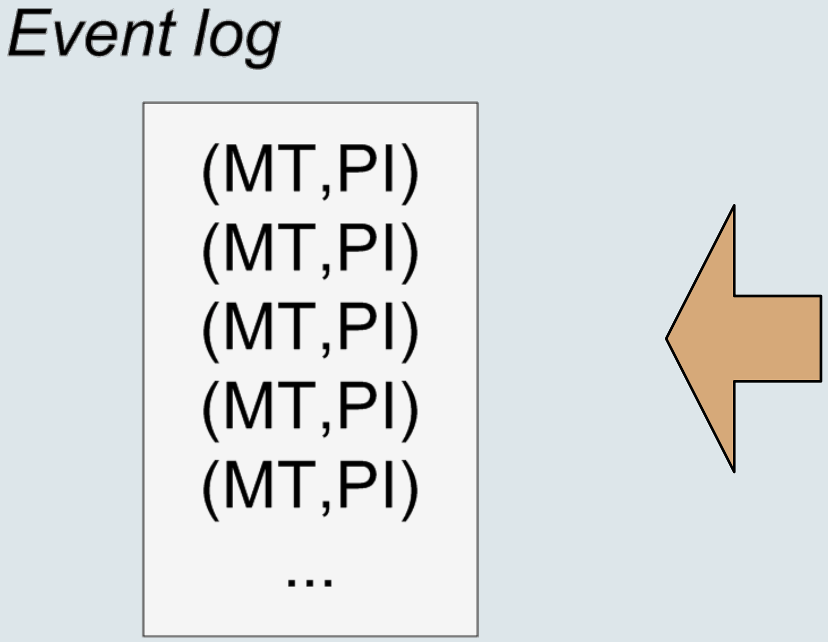


C. Ouyang, W.M.P. van der Aalst, S. Breutel, M. Dumas, A.H.M. ter Hofstede, and H.M.W. Verbeek. Formal Semantics and Analysis of Control Flow in WS-BPEL (Revised version). *BPM Center Report BPM-05-15*, BPMcenter.org, 2005.

3.2 Monitoring and Correlating Messages

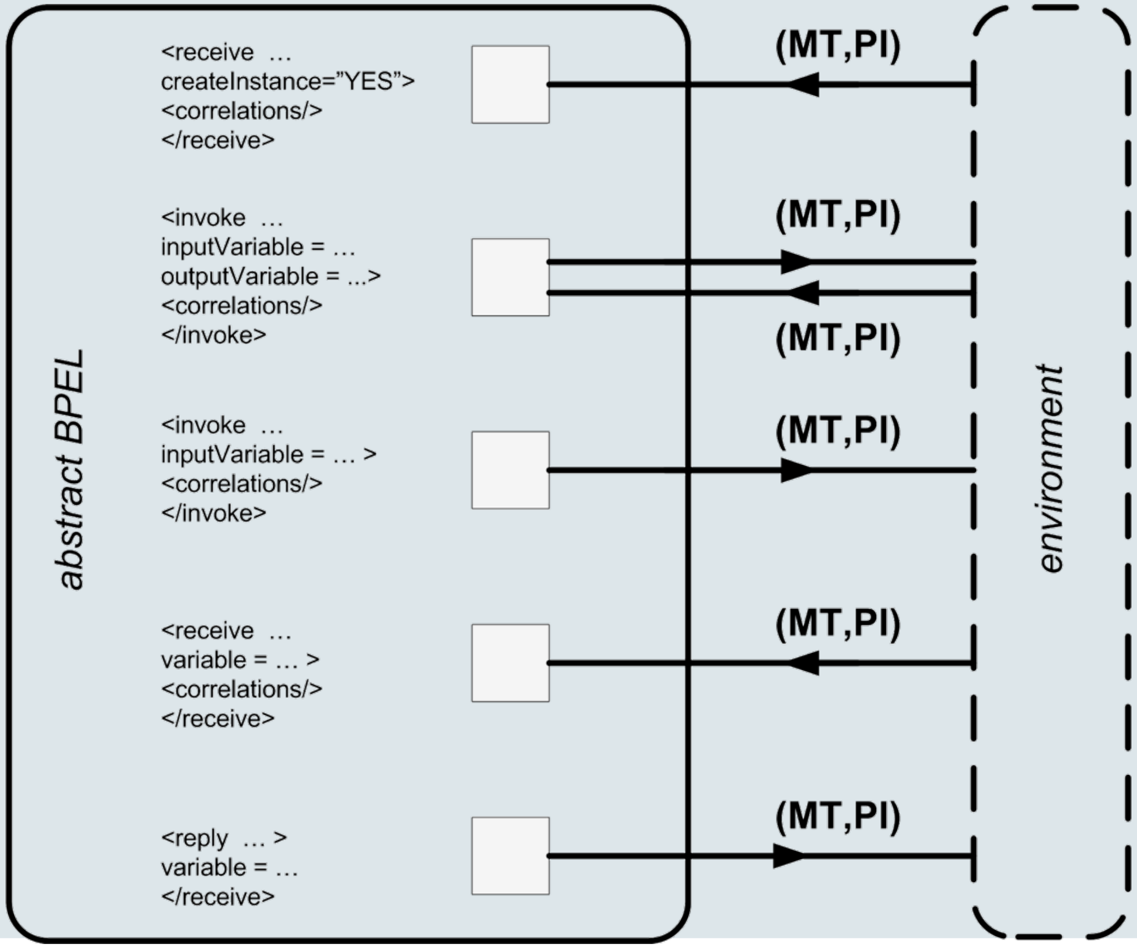
From SOAP messages to Event log:

- Process instance (PI)
- Label denoting the BPEL communication action (MT)



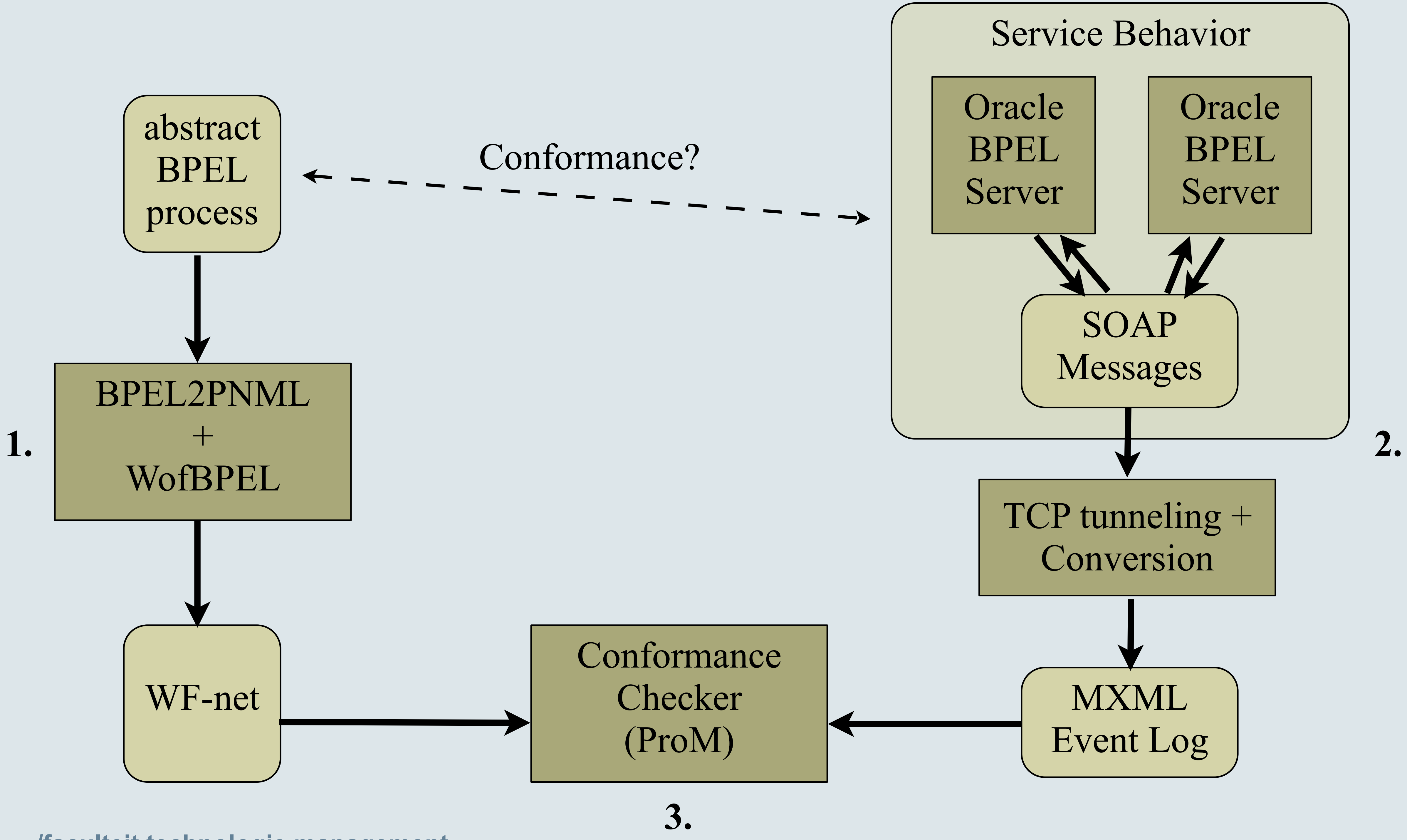
Event log

(MT,PI)
(MT,PI)
(MT,PI)
(MT,PI)
(MT,PI)
...



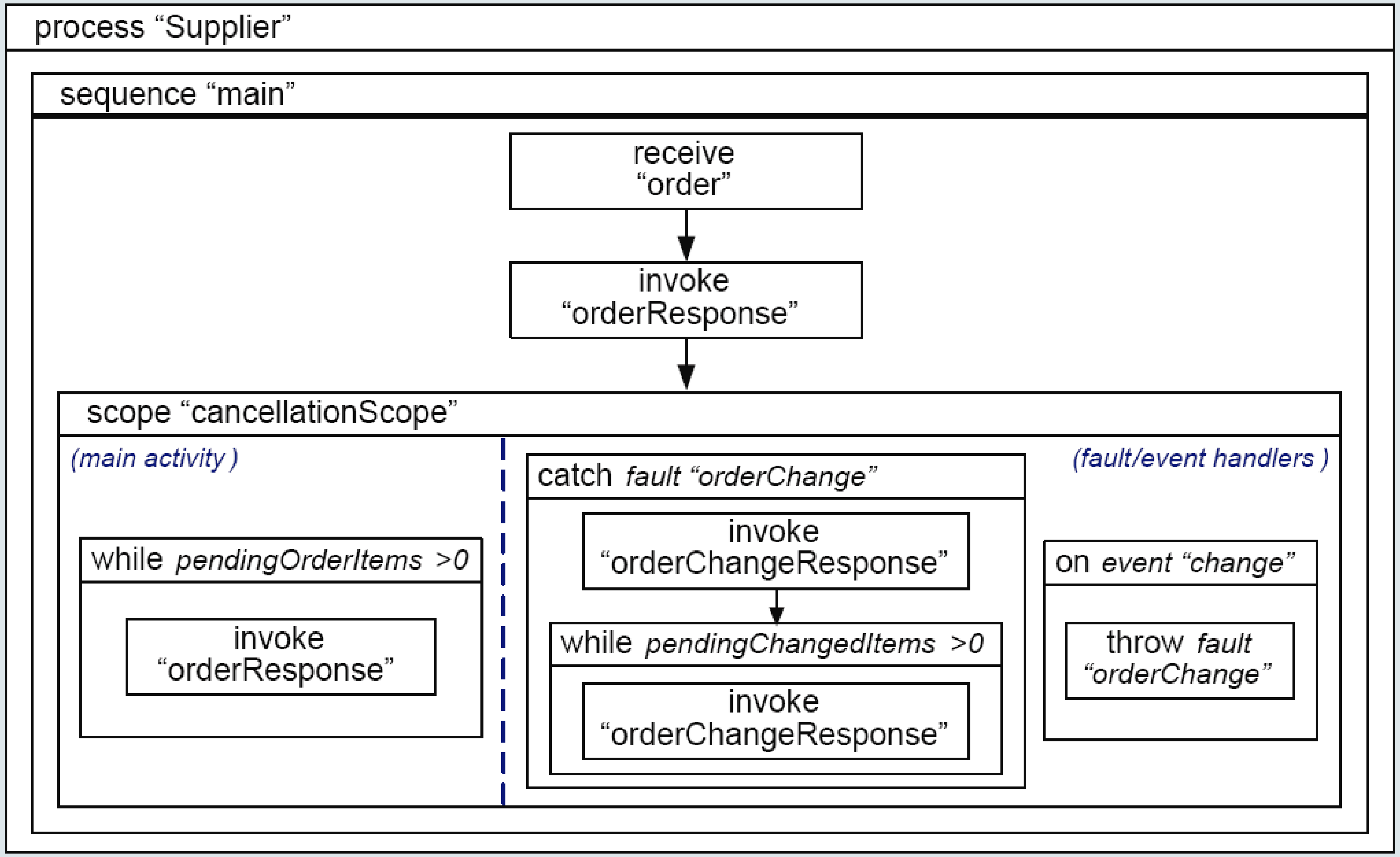
```
<receive  …
createInstance="YES">
<correlations/>
</receive>

<invoke  …
inputVariable = …
outputVariable = ...>
<correlations/>
</invoke>

<invoke  …
inputVariable = … >
<correlations/>
</invoke>

<receive  …
variable = … >
<correlations/>
</receive>

<reply  … >
variable = …
</receive>
```

4. Feasibility Study - Overview



4.1 From BPEL to WF-nets (1)

- Translation: BPEL2PNML



process "Supplier"

sequence "main"

receive "order"

invoke "orderResponse"

scope "cancellationScope"

(main activity)

while *pendingOrderItems >0*

invoke "orderResponse"

catch *fault "orderChange"*

invoke "orderChangeResponse"

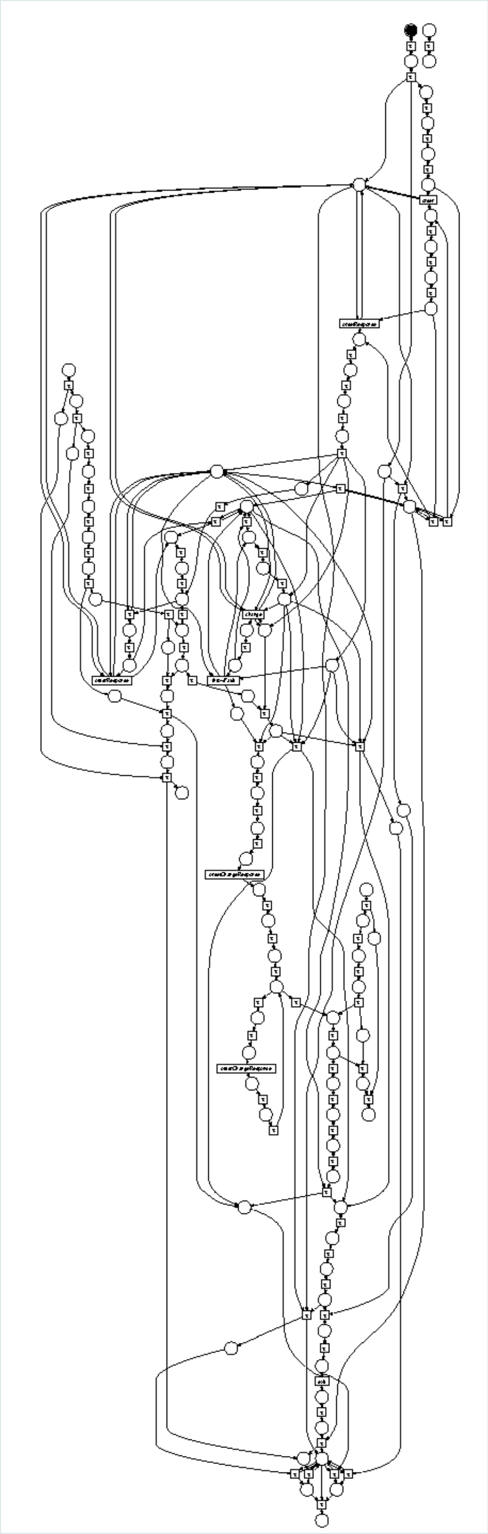
while *pendingChangedItems >0*

invoke "orderChangeResponse"

(fault/event handlers)

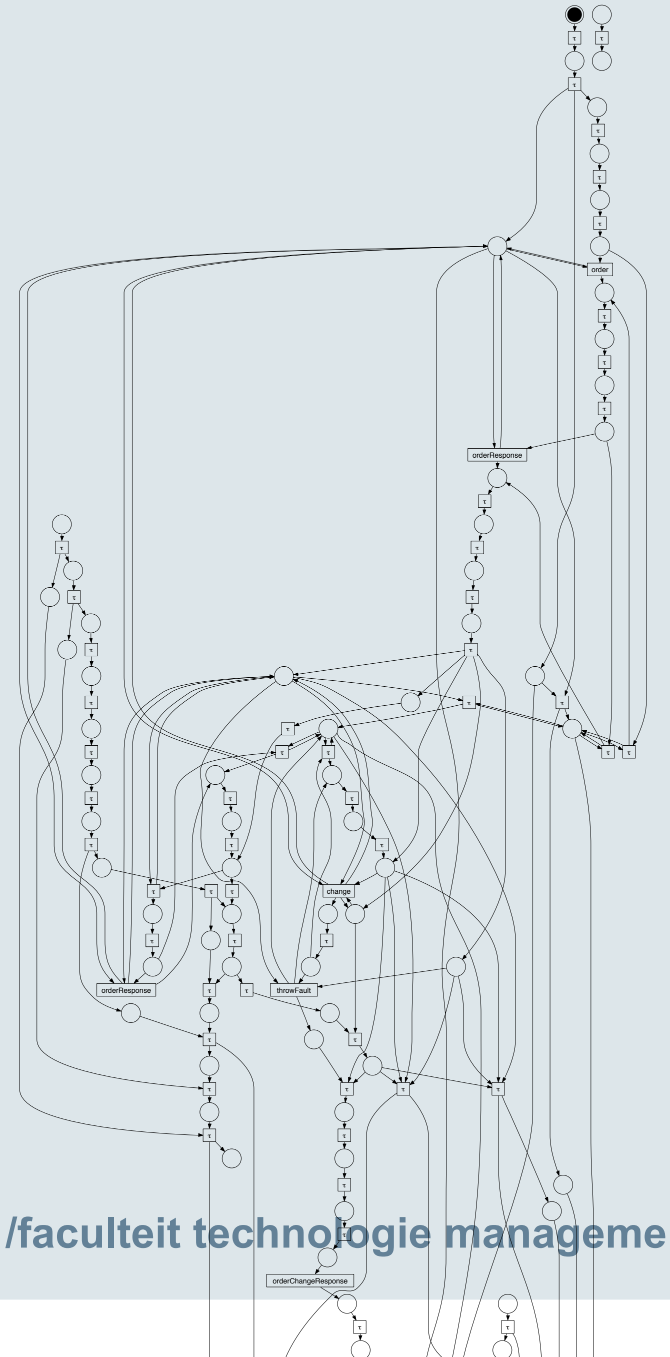
on *event "change"*

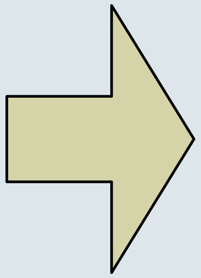
throw *fault "orderChange"*

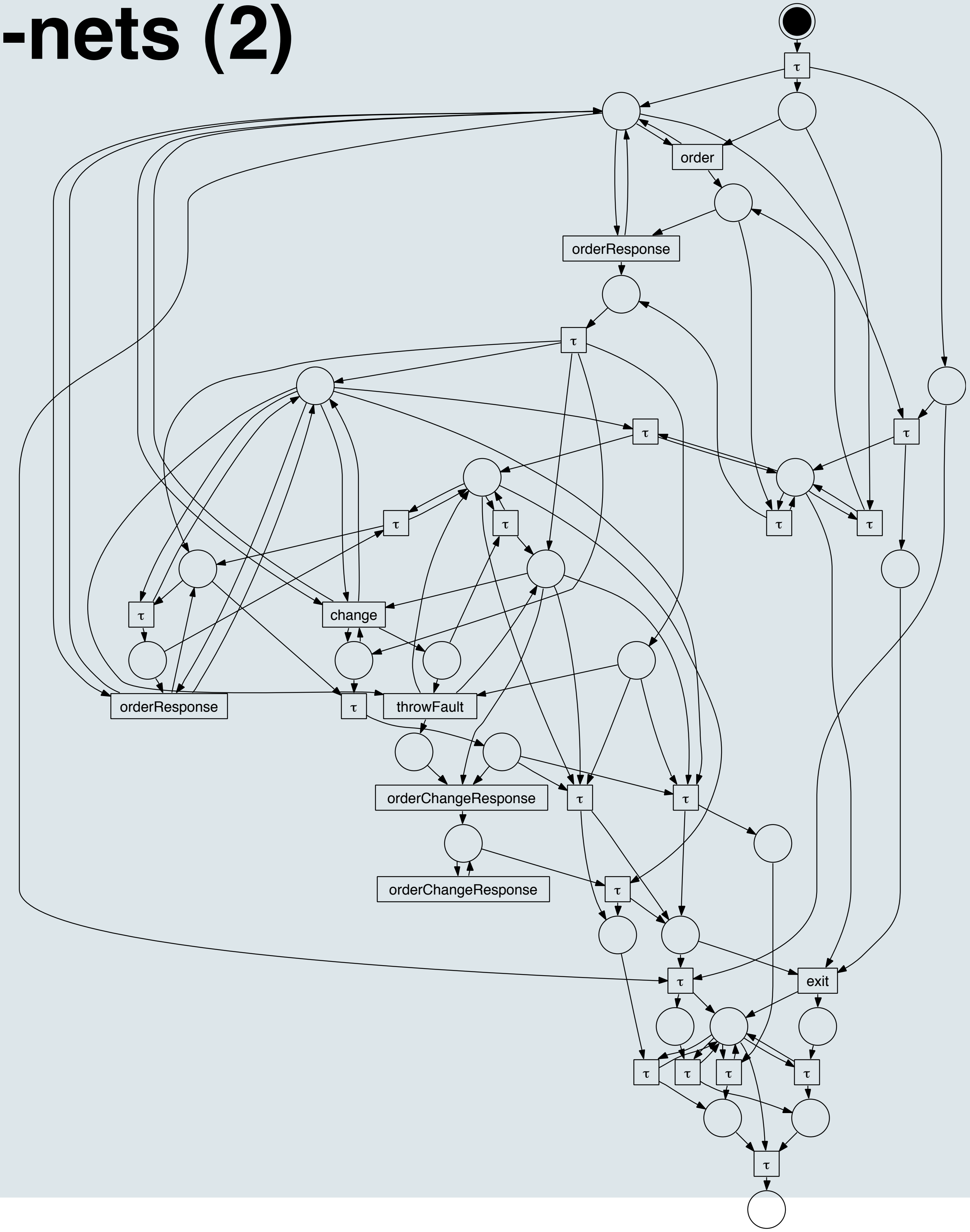


4.1 From BPEL to WF-nets (2)

- Reduction: WofBPEL





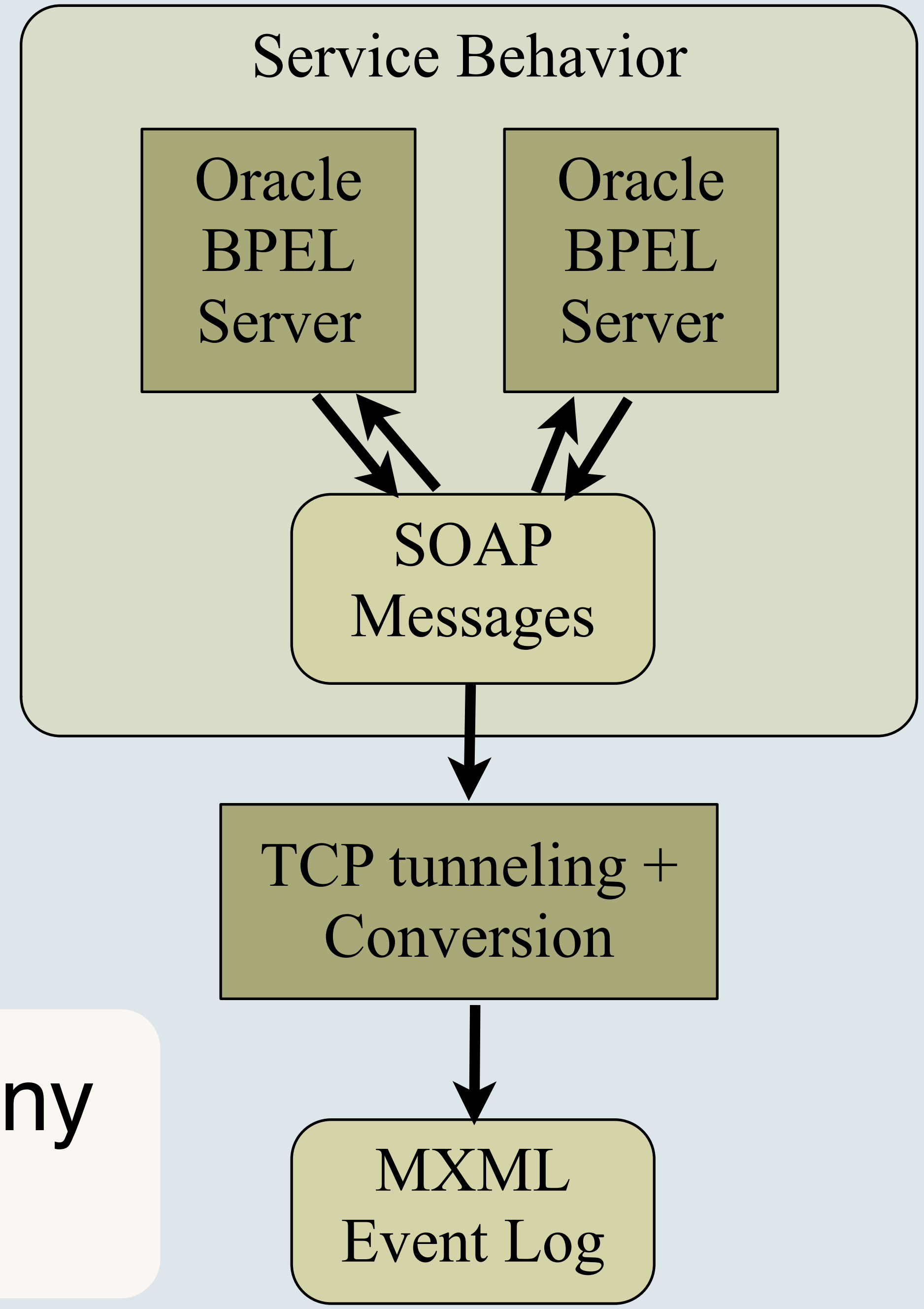


4.2 Observing Service Behavior

Implementation via Oracle BPEL:

- Supplier service
- Customer service

➡ could have been implemented on any other platform! (e.g., Java, .NET, etc.)



Result: Observation of one out of many possible communication scenarios

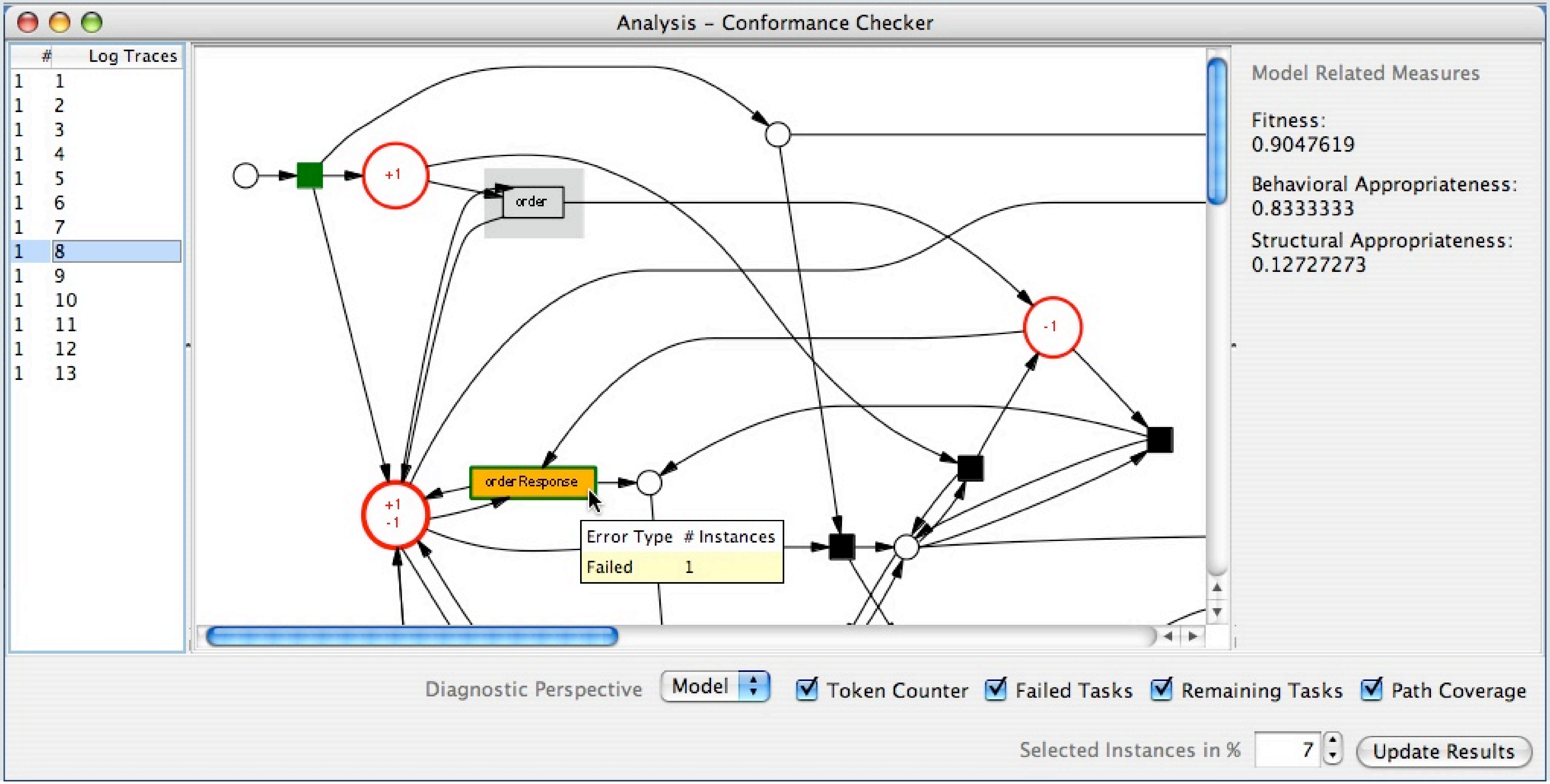
4.3 Conformance Checking

- Desirable and undesirable scenarios for the supplier service execution

	Scenario	Fitness	Log trace
desirable behavior	1	1.0	(order, orderResponse)
	2	1.0	(order, orderResponse, orderResponse, orderResponse)
	3	1.0	(order, orderResponse, change, orderChangeResponse)
	4	1.0	(order, orderResponse, orderResponse, change, orderChangeResponse)
	5	1.0	(order, orderResponse, change, orderChangeResponse, orderChangeResponse)
undesirable behavior	6	0.625	(order)
	7	0.749	(order, orderResponse, change)
	8	0.905	(orderResponse)
	9	1.0	(order, orderResponse, change, orderResponse, orderChangeResponse)
	10	0.759	(order, change, orderChangeResponse)
	11	0.0	(change)
	12	0.914	(order, orderResponse, change, orderChangeResponse, change)
	13	0.971	(order, orderResponse, change, change, orderChangeResponse)

4.3 Conformance Checking

- Analysis scenario 8: (orderResponse)



Analysis – Conformance Checker

#	Log Traces
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10
1	11
1	12
1	13

Model Related Measures

Fitness: 0.9047619

Behavioral Appropriateness: 0.8333333

Structural Appropriateness: 0.12727273

Error Type	# Instances
Failed	1

Diagnostic Perspective: Model

☑ Token Counter ☑ Failed Tasks ☑ Remaining Tasks ☑ Path Coverage

Selected Instances in %: 7

4. Conclusion

- Conformance checking particularly relevant in a service-based environment due to the independence of services
- Feasible to check conformance of an abstract BPEL process using existing tool chain (BPEL2PNML, WofBPEL, ProM Conformance Checker)

W.M.P. van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, and H.M.W. Verbeek. Choreography Conformance Checking: An Approach based on BPEL and Petri Nets (extended version). *BPM Center Report BPM-05-25*, BPMcenter.org, 2005.

Thank you very much!

Questions?