



# Transforming BPEL into Petri Nets

Sebastian Hinz, Karsten Schmidt,  
Christian Stahl

# Motivation

---

## **BPEL:**

- modelling language for business processes
  - BPEL process can be very huge
- formal verification of BPEL processes needed

## **problems:**

- BPEL is specified informally
- formal verification of BPEL processes not possible

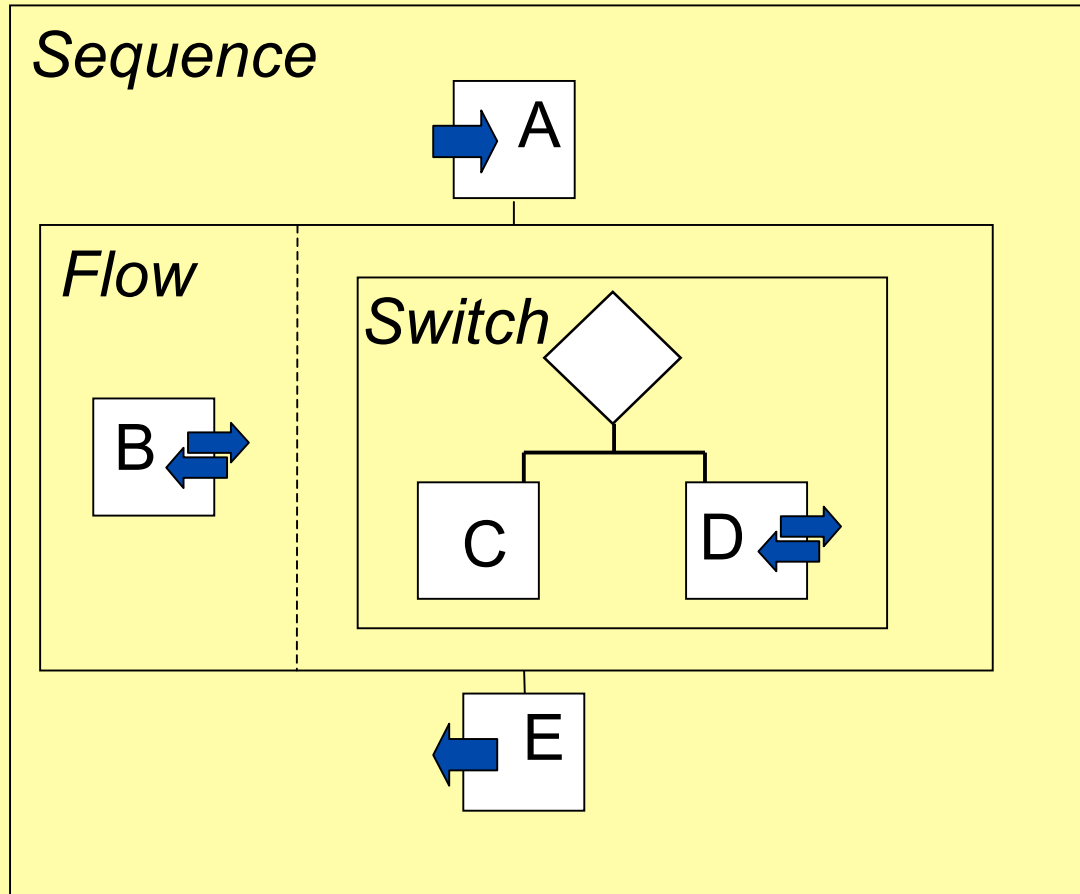
# Roadmap

---

1. Petri net semantics for BPEL
2. Automatic transformation
3. Computer-aided verification of BPEL processes
4. Improved transformation

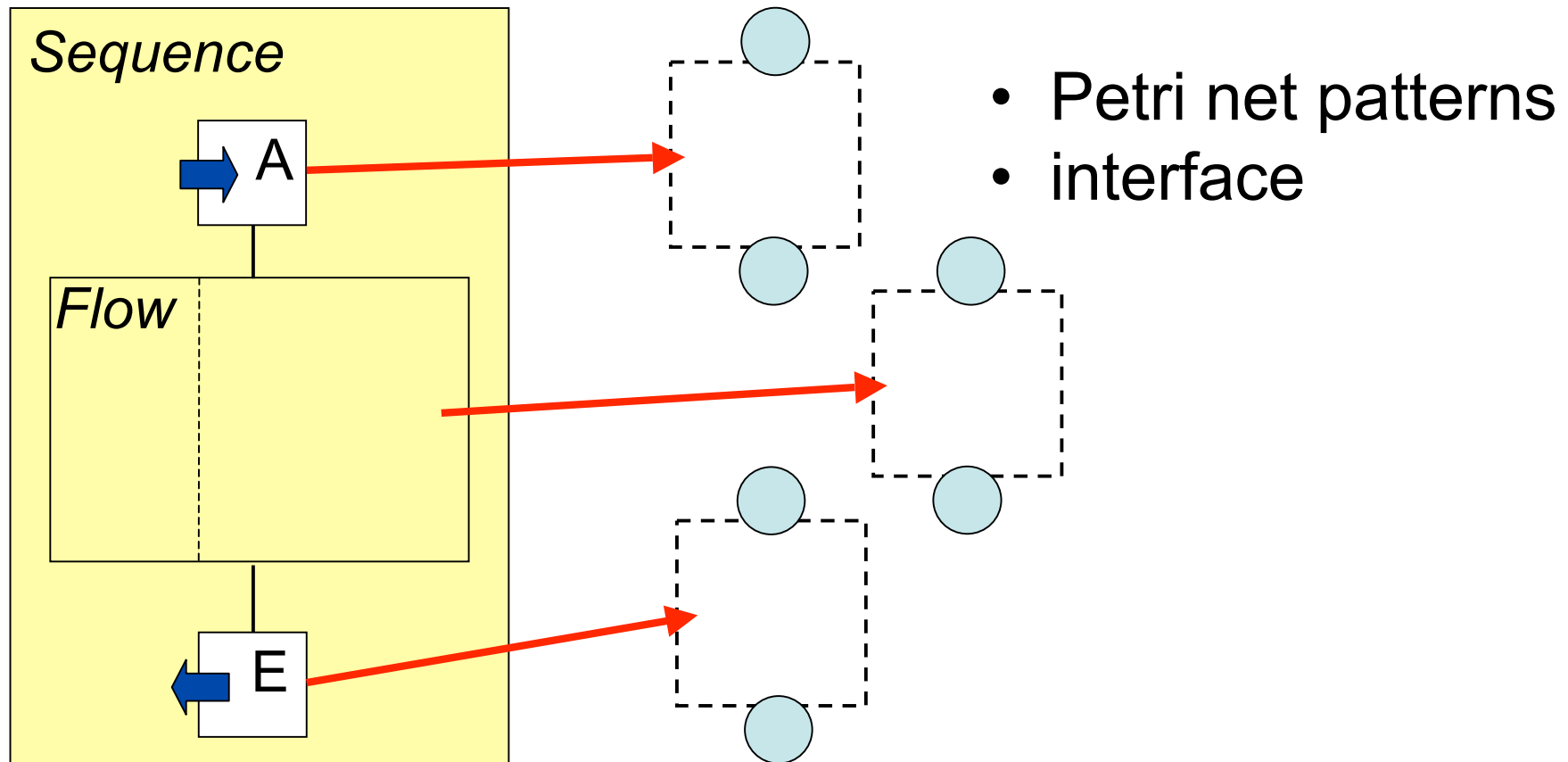
# 1. Petri net semantics for BPEL

# Process



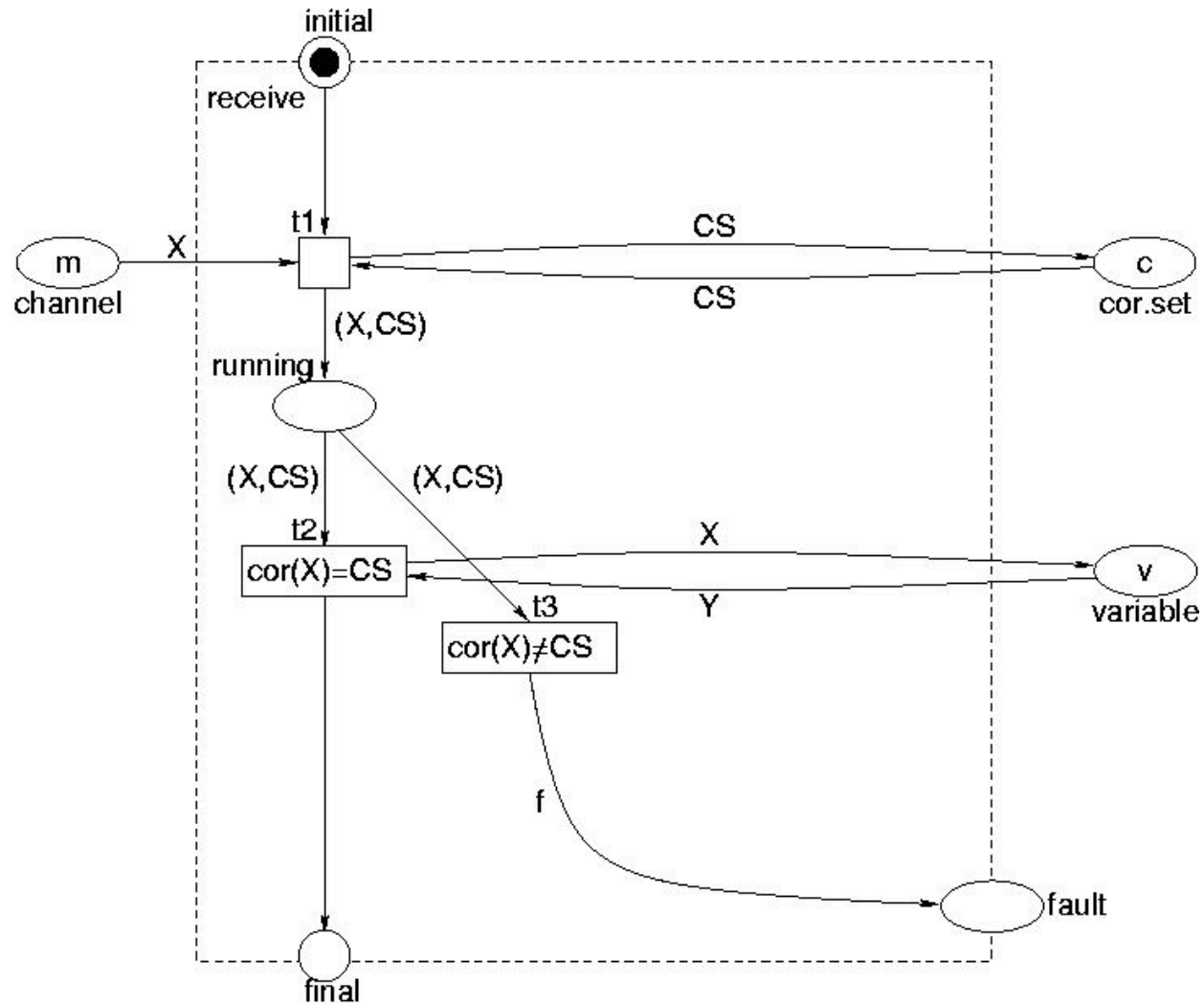
# Idea of the Petri net semantics

---



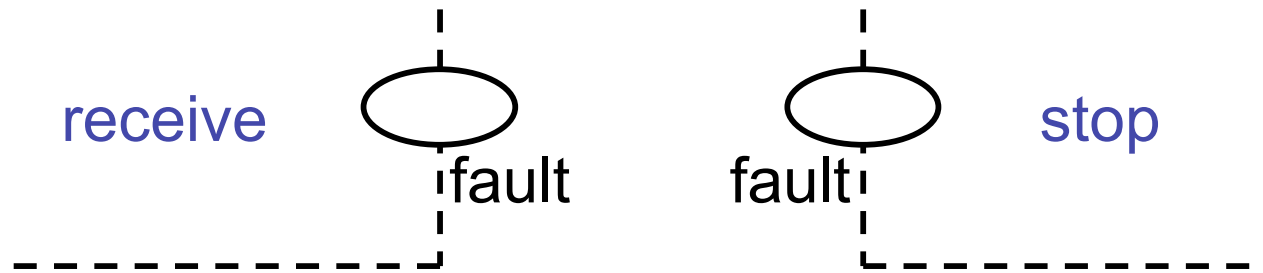
- complete Petri net semantics for BPEL v1.1

# Example: receive



# Controlled stop of a process

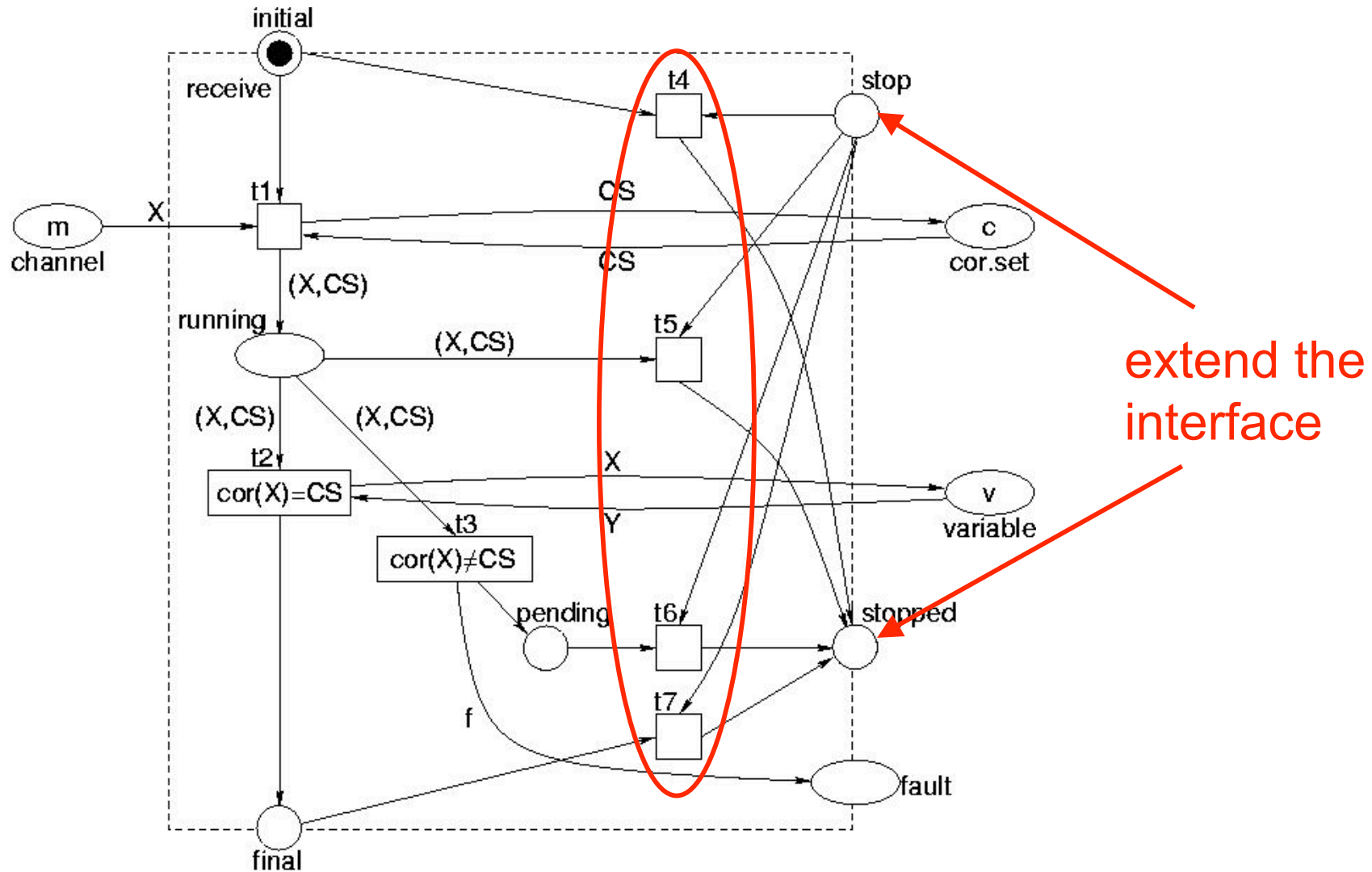
---



- process is forced to stop, e.g. when fault occurs
- specification cloudy
- modelling decisions:
  - process is extended by a stop pattern
  - stop pattern controls the stop procedure
  - removes all tokens
  - each activity pattern is extended by a stop component



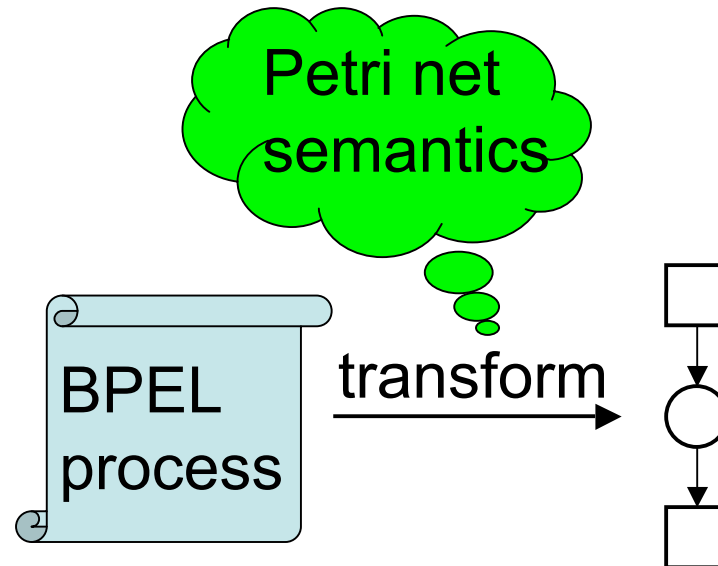
# Example: receive (cont.)



## 2. Automatic transformation

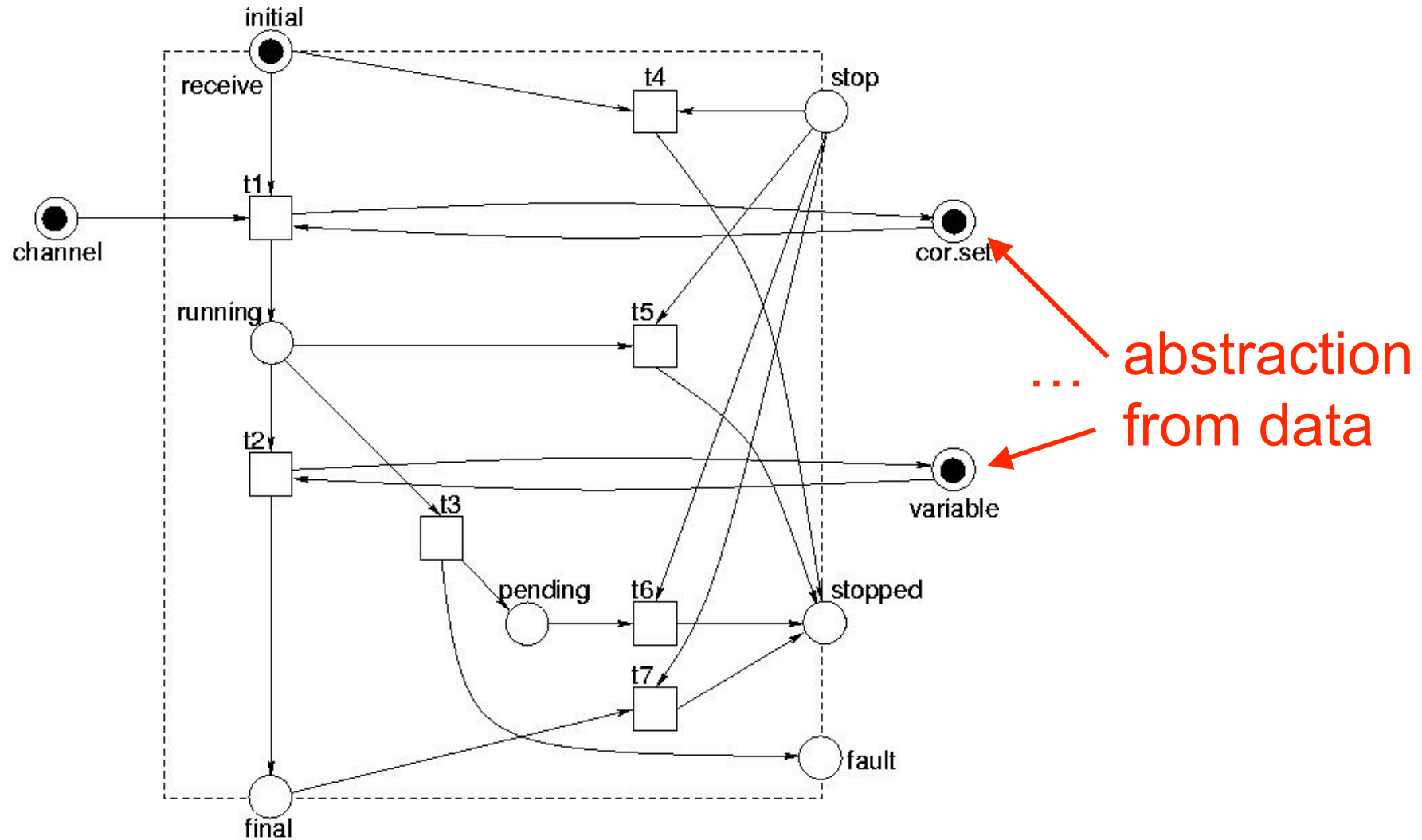
# Schema of transformation

---



- transformation can be done automatically  
→ compiler *BPEL2PN*
- abstraction from data

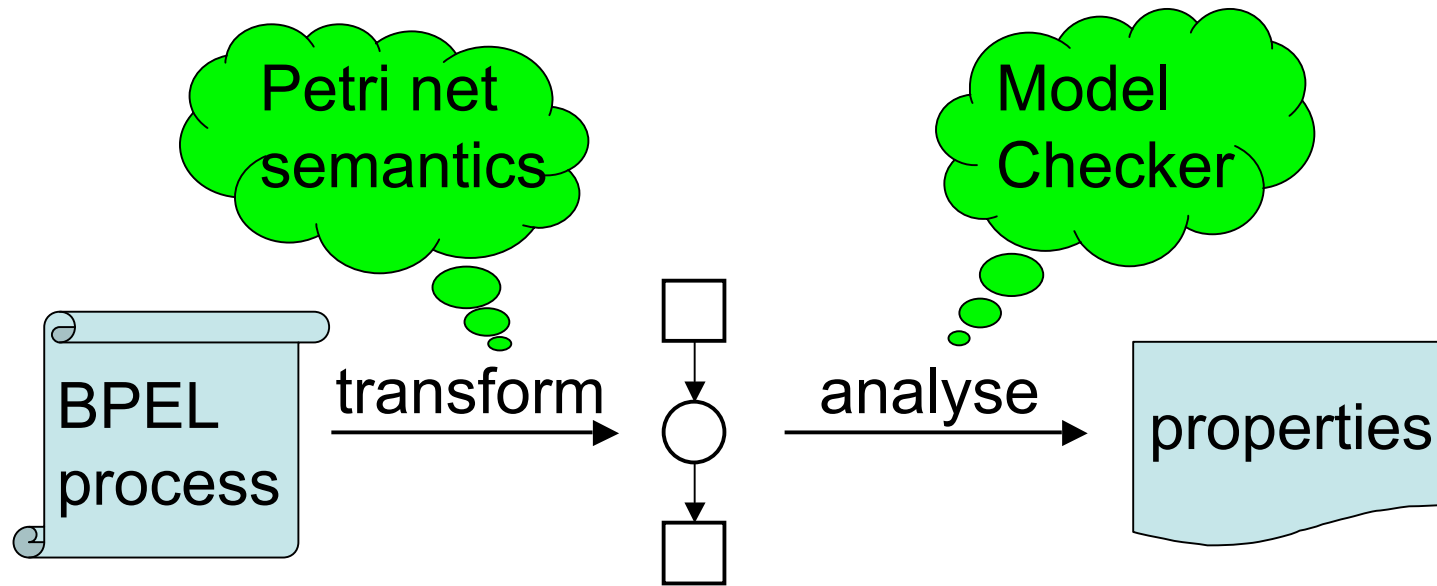
# Example: receive



# 3. Computer-aided verification of BPEL processes

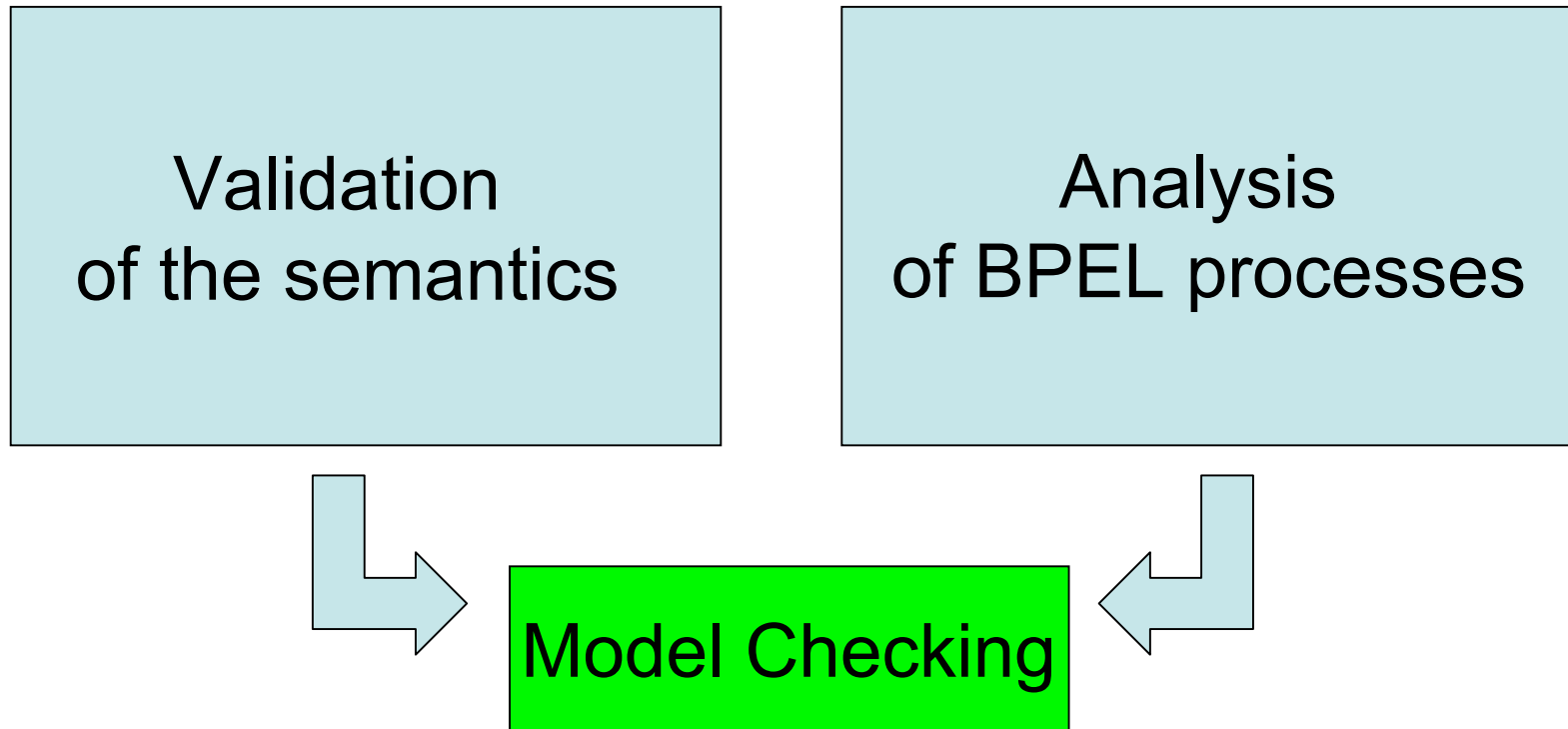
# Schema

---



# Validation and Analysis

---



# Analysis results

---

	PO	Shop	Bank
activities	17	53	132
places	158	410	1,890
transitions	249	1,069	21,193
states	10,000	6,300,000	?
red. states	1,300	440,000	?

using the model checker LoLA

## problem:

- real-life processes cause **state-explosion**  
→ smaller models needed



## 4. Improved transformation

# Improved transformation

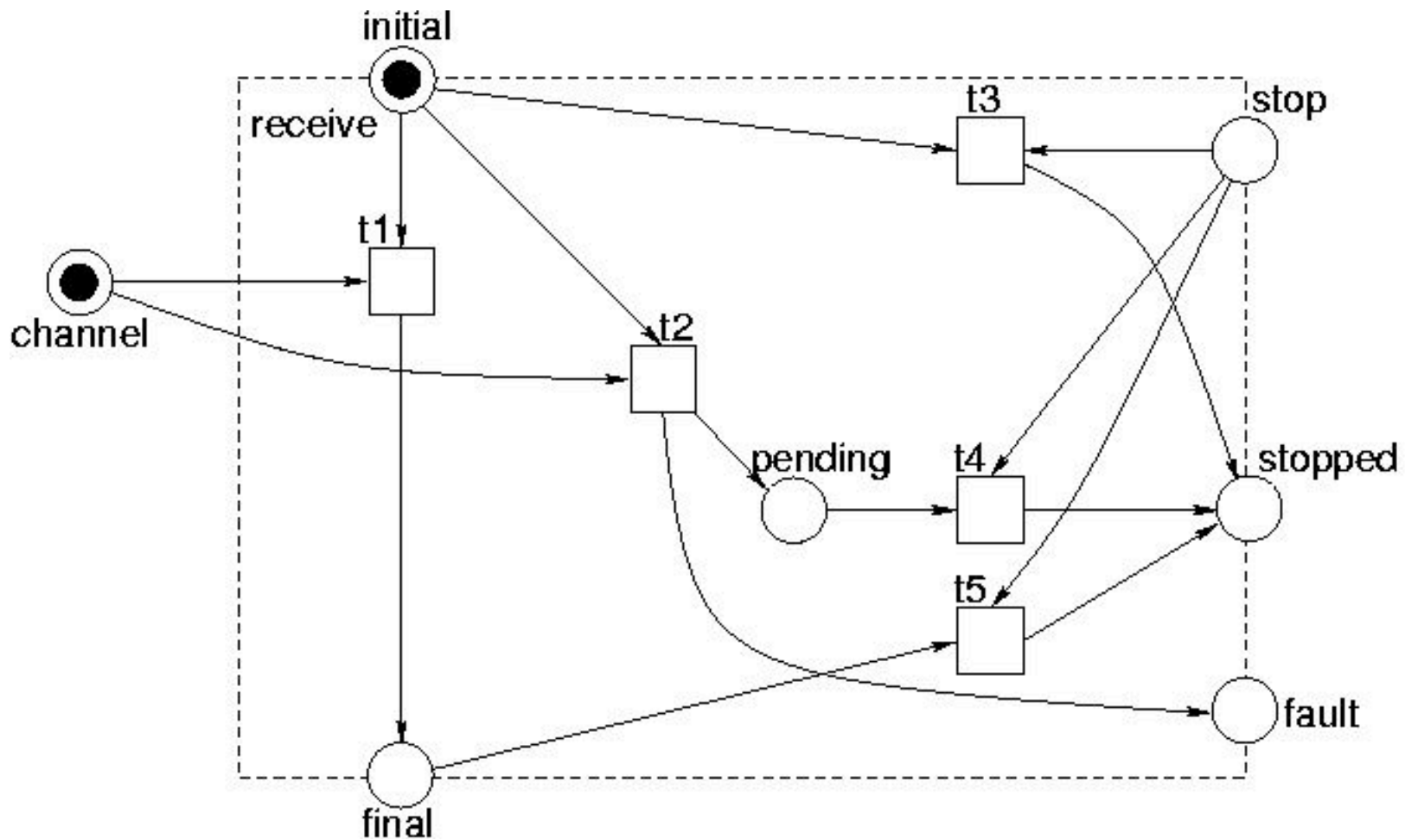
---

- so far patterns applicable in every context
- many of the modelled features are unused

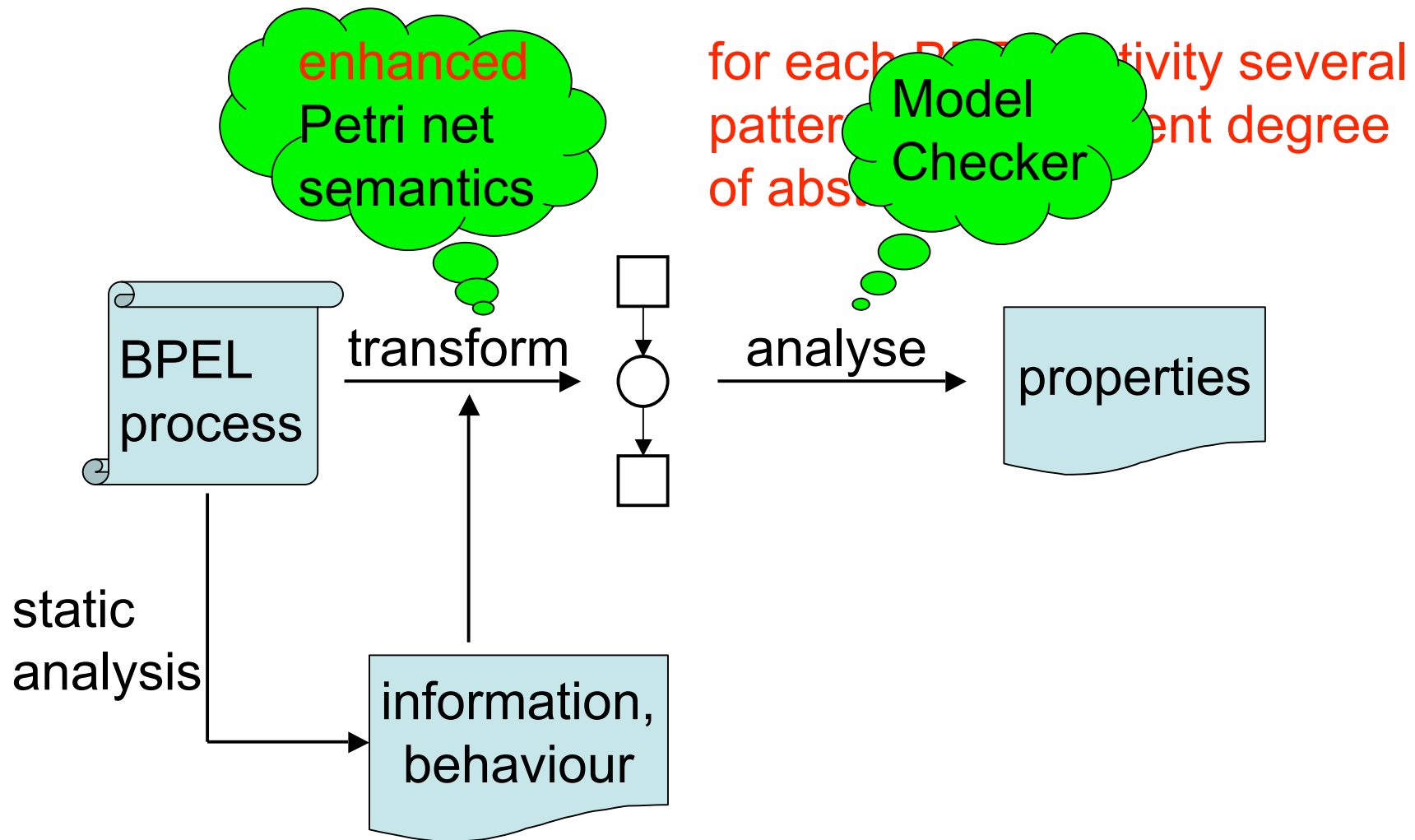
## **solution:**

1. get knowledge of an activity's context
  - Can an activity throw a fault?
2. derive aspects needed to prove a property
  - external behaviour: communication only

# receive: external behaviour



# Flexible model generation



# Further Work

---

- flexible model generation
- extend the semantics to BPEL v2.0
  - termination handler
  - ...

**Many thanks**