Business Intelligence & Process Modelling

Frank Takes

Universiteit Leiden

Lecture 7 — Process Modelling & Petri nets

Recap

- **Business Intelligence**: anything that aims at providing actionable information that can be used to support business decision making
  - Business Intelligence
  - Visual Analytics
  - Descriptive Analytics
  - Predictive Analytics
- **Process Modelling** (April and May)

Data & Models

Business Process Management (recap)

- **Process**: a set of related actions and transactions to achieve a certain objective
- **Business process**: a sequence of activities aimed at producing something of value for the business (Morgan02)
  - Management processes
  - Operational processes
  - Supporting processes
- **Business Process Management**: the discipline that combines knowledge from information technology and knowledge from management sciences and applies this to operational business processes (v.d. Aalst)
  - Extension of WorkFlow Management (WFM)

Business Process Modelling (recap)

- **Business Process Model**: abstract representation of business processes, functionality is:
  - Descriptive: what is actually happening?
  - Prescriptive: what should be happening?
  - Explanatory: why is the process designed this way?
- In practice: **formalize** and **visualize** business processes
- **Process Discovery**: derive the process from a description of activities
- **Process Mining**: the task of converting **event** data into process models (discovery, conformance, enhancement)

Why Model Processes? (recap)

- insight
- discussion: the stakeholders use models to structure discussions;
- documentation for instructing people or certification purposes
- verification, for example to find errors in systems
- performance analysis
- animation: “play out” different scenarios
- specification: a “contract” between the developer and the end user/management; and
- configuration of a system.

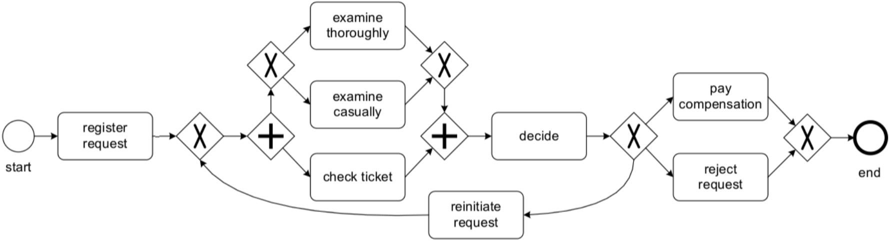
Business Process . . . Intelligence?

- M. Castellanos et al., **Business process intelligence**, *Handbook of research on business process modeling*, pp. 456–480, 2009.

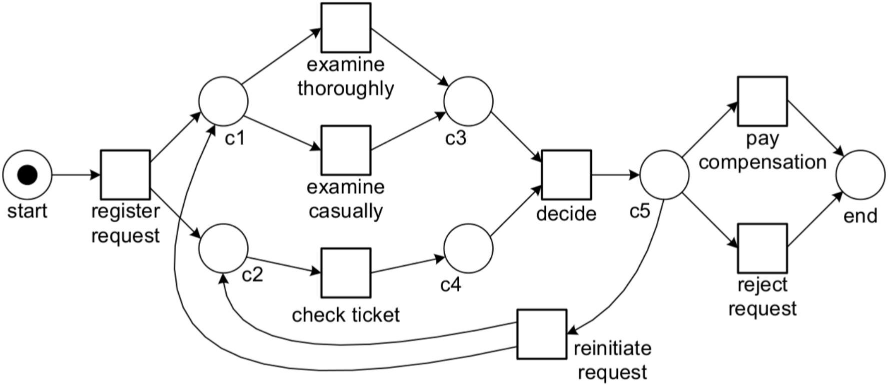
Process Modelling

- Informal models: used for discussion and documentation (process descriptions)
- Formal models: used for analysis or enactment
  - Petri Nets PN
  - Business Process Model Notation BPMN

Business Process Model Notation



Petri Nets


start
register request
c1
c2
examine thoroughly
examine casually
check ticket
c3
c4
decide
c5
pay compensation
reject request
end
reinitiate request

Event logs (1)

Case ID	Event ID	dd-mm-yyyy:hh.mm	Activity	Resource	Costs
1	35654423	30-12-2010:11.02	register request	Pete	50
1	35654424	31-12-2010:10.06	examine thoroughly	Sue	400
1	35654425	05-01-2011:15.12	check ticket	Mike	100
1	35654426	06-01-2011:11.18	decide	Sara	200
1	35654427	07-01-2011:14.24	reject request	Pete	200
2	35654483	30-12-2010:11.32	register request	Mike	50
2	35654485	30-12-2010:12.12	check ticket	Mike	100
2	35654487	30-12-2010:14.16	examine casually	Sean	400
2	35654488	05-01-2011:11.22	decide	Sara	200
2	35654489	08-01-2011:12.05	pay compensation	Ellen	200
3	35654521	30-12-2010:14.32	register request	Pete	50
3	35654522	30-12-2010:15.06	examine casually	Mike	400
3	35654524	30-12-2010:16.34	check ticket	Ellen	100
3	35654525	06-01-2011:09.18	decide	Sara	200
3	35654526	06-01-2011:12.18	reinitiate request	Sara	200
3	35654527	06-01-2011:13.06	examine thoroughly	Sean	400
3	35654530	08-01-2011:11.43	check ticket	Pete	100
3	35654531	09-01-2011:09.55	decide	Sara	200
3	35654533	15-01-2011:10.45	pay compensation	Ellen	200
4	35654641	06-01-2011:15.02	register request	Pete	50
4	35654643	07-01-2011:12.06	check ticket	Mike	100
4	35654644	08-01-2011:14.43	examine thoroughly	Sean	400
4	35654645	09-01-2011:12.02	decide	Sara	200
4	35654647	12-01-2011:15.44	reject request	Ellen	200
...					

Table : Event logs of a helpdesk handling customer compensations

Event logs (2)

Case ID	Event ID	dd-mm-yyyy:hh.mm	Activity	Resource	Costs
...					
5	35654711	06-01-2011:09.02	register request	Ellen	50
5	35654712	07-01-2011:10.16	examine casually	Mike	400
5	35654714	08-01-2011:11.22	check ticket	Pete	100
5	35654715	10-01-2011:13.28	decide	Sara	200
5	35654716	11-01-2011:16.18	reinitiate request	Sara	200
5	35654718	14-01-2011:14.33	check ticket	Ellen	100
5	35654719	16-01-2011:15.50	examine casually	Mike	400
5	35654720	19-01-2011:11.18	decide	Sara	200
5	35654721	20-01-2011:12.48	reinitiate request	Sara	200
5	35654722	21-01-2011:09.06	examine casually	Sue	400
5	35654724	21-01-2011:11.34	check ticket	Pete	100
5	35654725	23-01-2011:13.12	decide	Sara	200
5	35654726	24-01-2011:14.56	reject request	Mike	200
6	35654871	06-01-2011:15.02	register request	Mike	50
6	35654873	06-01-2011:16.06	examine casually	Ellen	400
6	35654874	07-01-2011:16.22	check ticket	Mike	100
6	35654875	07-01-2011:16.52	decide	Sara	200
6	35654877	16-01-2011:11.47	pay compensation	Mike	200

Table : Event logs of a support desk handling customer compensations

Simplified event log

Case ID	Trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$

Table : Simplified event log of a support desk handling customer compensations (a = register request, b = examine thoroughly, c = examine casually, d = check ticket, e = decide, f = reinitiate request, g = pay compensation, h = reject request)

Simplified event log

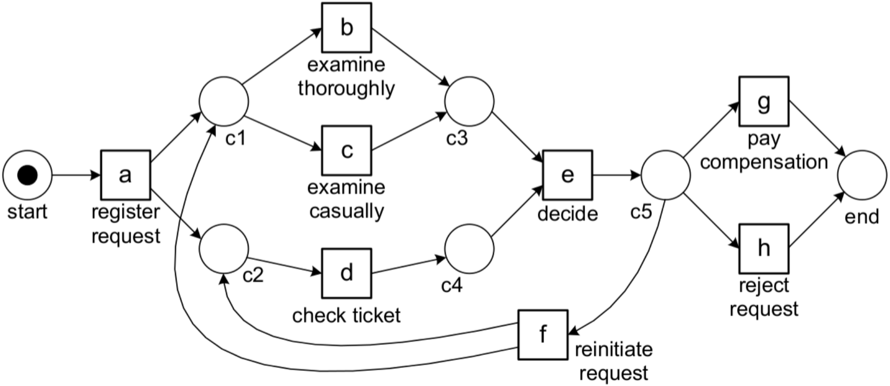
Case ID	Trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$

Table : Simplified event log of a support desk handling customer compensations (a = register request, b = examine thoroughly, c = examine casually, d = check ticket, e = decide, f = reinitiate request, g = pay compensation, h = reject request)

In short: $\{\langle a, b, d, e, h \rangle, \langle a, d, c, e, g \rangle, \langle a, c, d, e, f, b, d, e, g \rangle, \langle a, d, b, e, h \rangle, \langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle, \langle a, c, d, e, g \rangle\}$

Example (1)

Case ID	Trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$



Example (2)

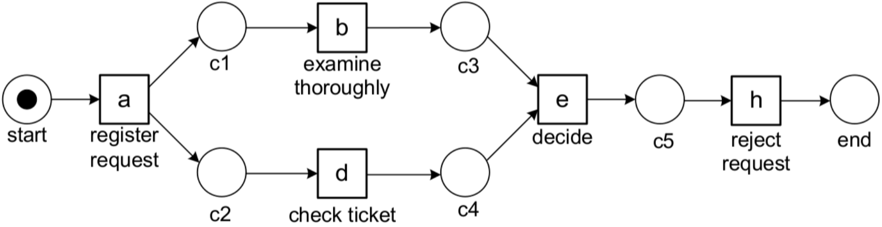
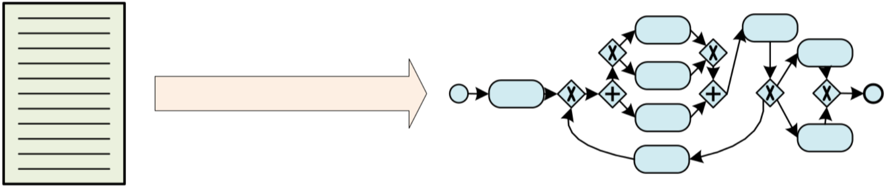


Figure : Petri net based on event log $\{\langle a, b, d, e, h\rangle, \langle a, d, b, e, h\rangle\}$

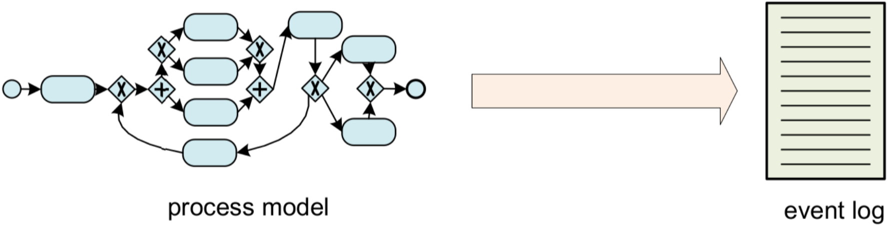
Play in



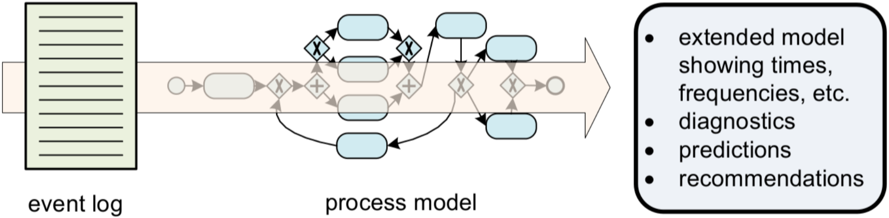
event log

process model

Play out



Replay



event log

process model

- extended model showing times, frequencies, etc.
- diagnostics
- predictions
- recommendations

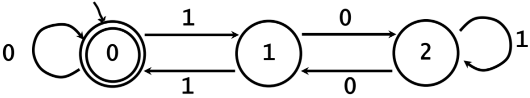
Replay

- Connecting models to real events is crucial
- Possible uses
  - Conformance checking
  - Repairing models
  - Extending the model with frequencies and temporal information
  - Constructing predictive models
  - Operational support (prediction, recommendation, etc.)

Petri Nets

Automata (remember?)

- Finite automaton $FA = (Q, \Sigma, q_o, A, \delta)$
  - Q is a finite set of states
  - Σ is a finite alphabet of input symbols
  - $q_o \in Q$ is the initial state
  - $A \subseteq Q$ is the set of accepting states
  - $\delta : Q \times \Sigma \rightarrow Q$ is the transition function



Automata (remember?)

- Finite automaton $FA = (Q, \Sigma, q_o, A, \delta)$
  - Q is a finite set of states
  - Σ is a finite alphabet of input symbols
  - $q_o \in Q$ is the initial state
  - $A \subseteq Q$ is the set of accepting states
  - $\delta : Q \times \Sigma \rightarrow Q$ is the transition function

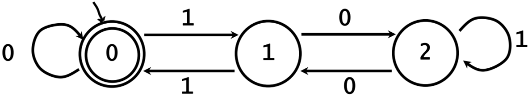
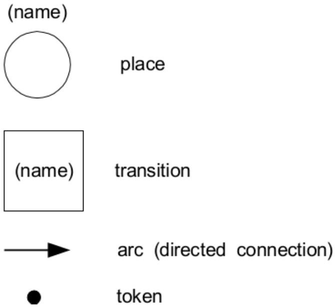


Figure : Deterministic Finite Automaton for the function $x \mod 3 = 0$

Petri Nets

- Petri net $N = (P, T, F)$
- P is a finite set of **places**
- T is a finite set of **transitions**
- $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed **arcs** called the **flow relation**

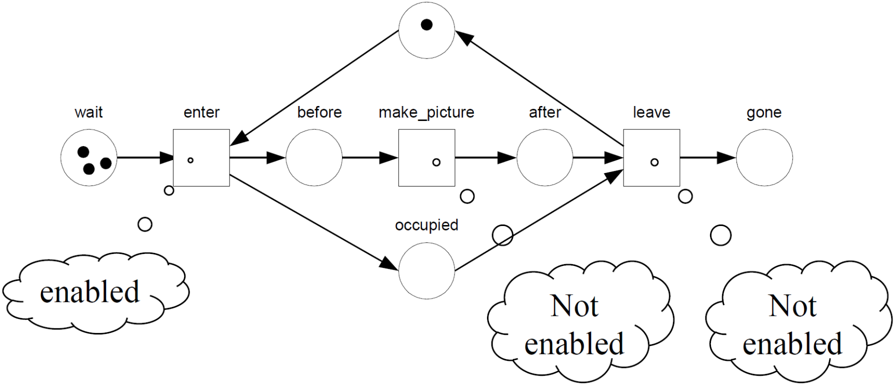


Labeled Petri Nets

- Petri net $N = (P, T, F, A, \ell)$
- P is a finite set of **places**
- T is a finite set of **transitions**
- $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed **arcs** called the **flow relation**
- A is a set of activity **labels**
- $\ell : T \rightarrow A$ is a **labeling function**

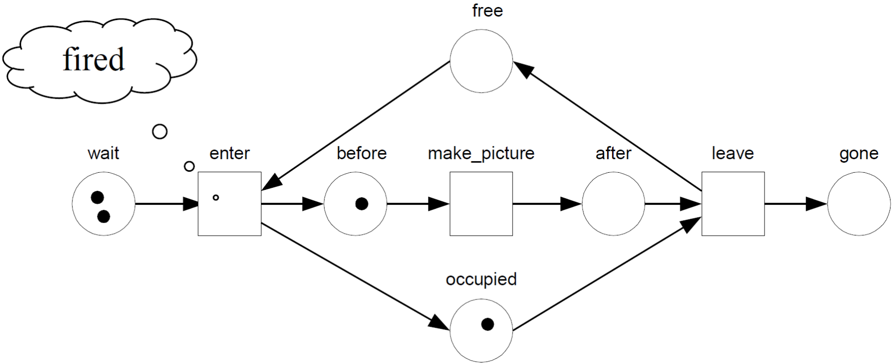
Enabling

- A transition is **enabled** if each of its input places contains at least one token



Firing

- An enabled transition can **fire** (i.e., it occurs), **consuming** a token from **each** input place and **producing** a token for **each** output place.



Petri Nets

- Connections are directed
- No connections between two places or two transitions
- Places may hold zero or more tokens
- At most one arc between nodes (for now)
- Firing is **atomic**
- Multiple transitions may be enabled, but only one fires at a time
- During execution, the number of tokens may vary if there are transitions for which the number of input places is not equal to the number of output places
- The network is static

Example (1)

- Petri net for a **traffic light**

Example (1)

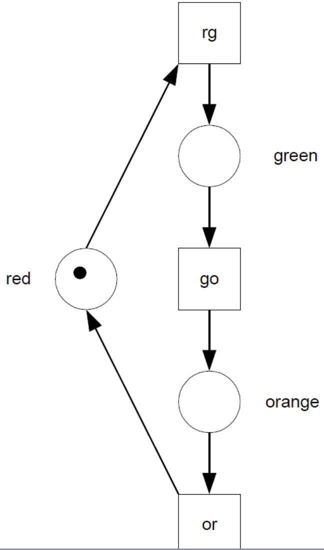
- Petri net for a **traffic light**
- **States**: red, orange and green

Example (1)

- Petri net for a **traffic light**
- **States**: red, orange and green
- **Transitions** from red to green, green to orange, and orange to red

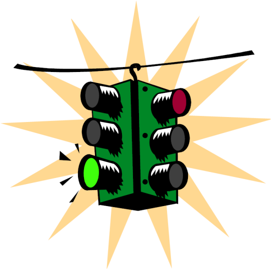
Example (1)

- Petri net for a **traffic light**
- **States**: red, orange and green
- **Transitions** from red to green, green to orange, and orange to red



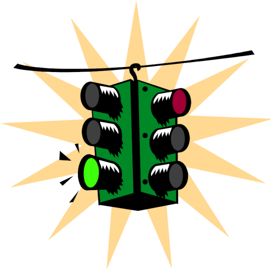
Example (2)

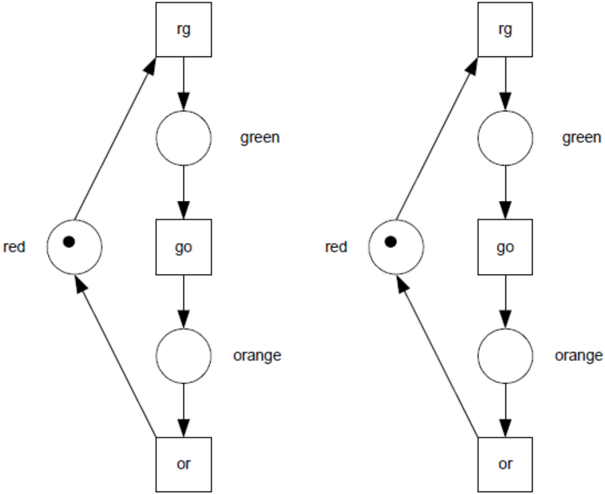
- Petri net for **2 traffic lights**



Example (2)

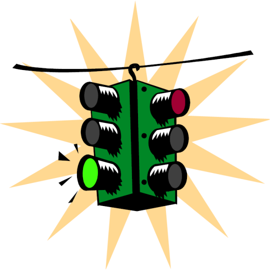
- Petri net for **2 traffic lights**

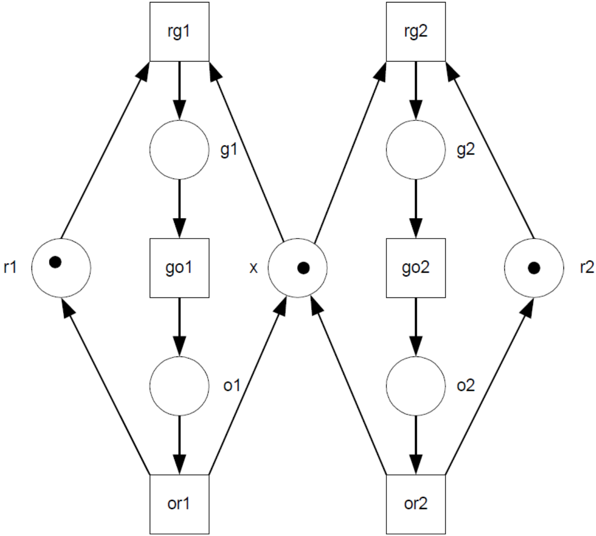




Example (3)

- Petri net for **2 traffic lights**



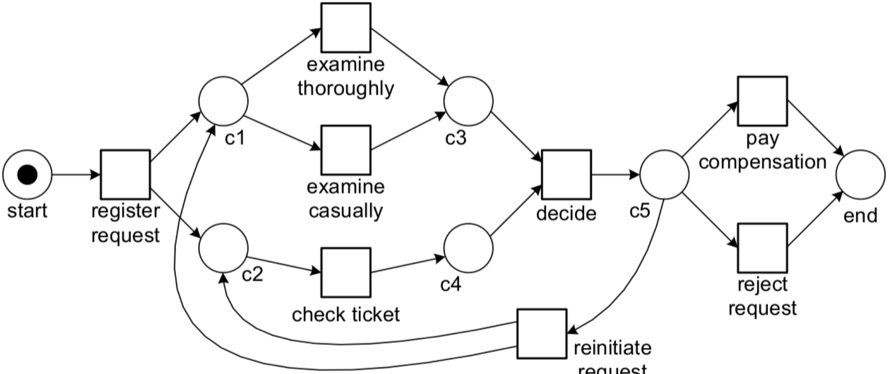


Marked Petri nets

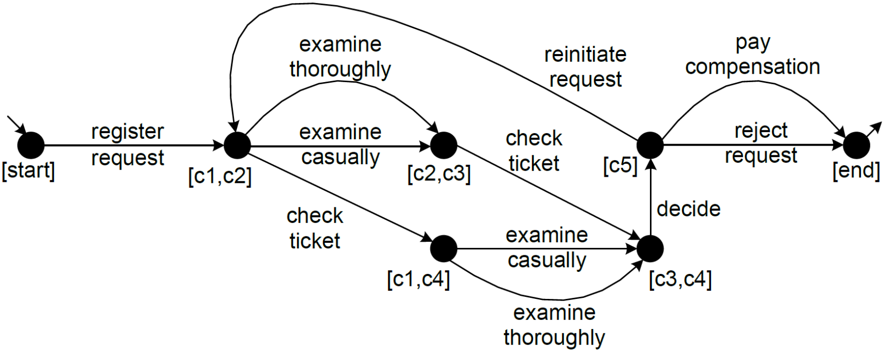
- **Marking**: the distribution of **tokens** over places in a Petri net
- Marked Petri net (M, N):
  - Petri net $N = (P, T, F)$
  - Marking $M \in \mathbb{B}(P)$ is a multi-set over P indicating the marking of N, denoted for example by $[c1^2, c2, c3^4]$
- Net **behavior**: set of all possible state sequences

Reachability graph

Reachability graph



Reachability graph



Different types of states

- **Initial state**: initial distribution of tokens.
- **Reachable state**: reachable from initial state.
- **Final/dead state**: no transition is enabled.
- **Home state/marking**: it is always possible to return (i.e., it is reachable from any reachable state).

Roles in Modelling

- **Place**: passive element ○
- **Transition**: active element □
- **Arc**: causal relation →
- **Token**: elements subject to change •

Role of a token •

- a **physical object**, for example a product, a part, a drug, a person;
- an **information object**, for example a message, a signal, a report;
- a **collection of objects**, for example a truck with products, a warehouse with parts, or an address file;
- an **indicator of a state**, for example the indicator of the state in which a process is, or the state of an object;
- an **indicator of a condition**: the presence of a token indicates whether a certain condition is fulfilled.

Role of a place ○

- a type of **communication medium**, like a telephone line, a middleman, or a communication network;
- a **buffer**: for example, a depot, a queue or a post bin;
- a **geographical location**, like a place in a warehouse, office or hospital;
- a possible **state or state condition**: for example, the floor where an elevator is, or the condition that a specialist is available.

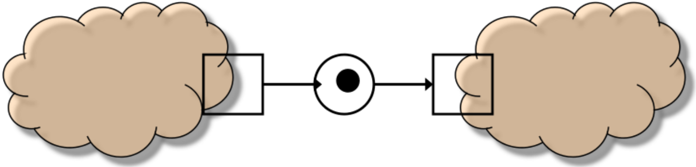
Role of a transition □

- an **event**: for example, starting an operation, the death of a patient, a change seasons or the switching of a traffic light from red to green;
- a **transformation of an object**, like adapting a product, updating a database, or updating a document;
- **transport of an object**: for example, transporting goods, or sending a file.

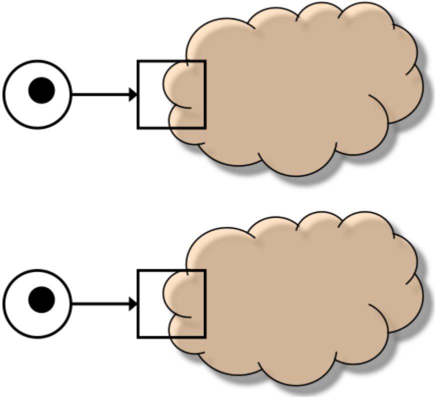
Typical net structures

- Causality
- Parallelism (AND-split, AND-join)
- Choice (XOR-split, XOR-join)
- Iteration (XOR-join, XOR-split)
- Capacity constraints
  - Feedback loop
  - Mutual exclusion
  - Alternating

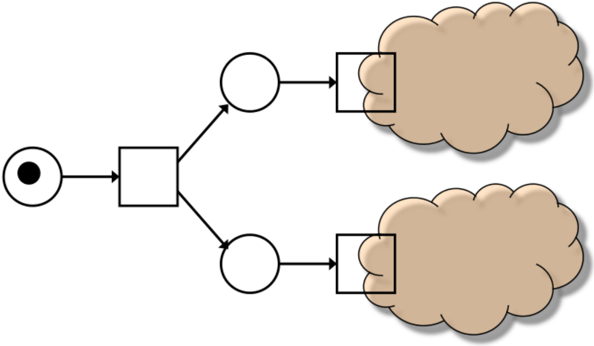
Causality



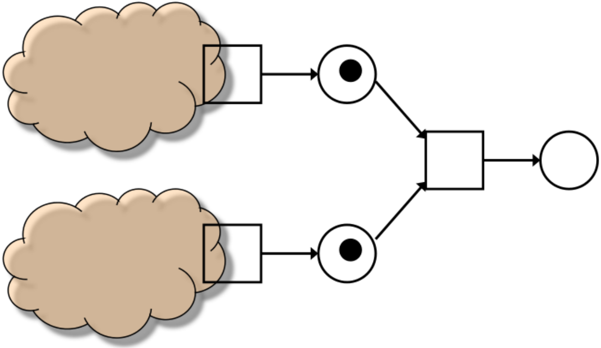
Parallelism



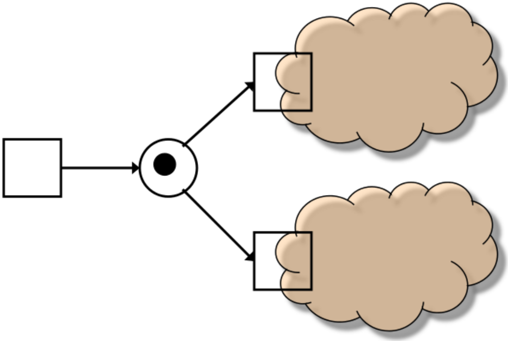
Parallelism: AND-split



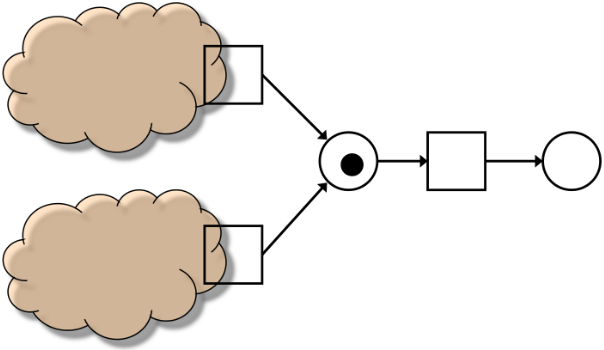
Parallelism: AND-join



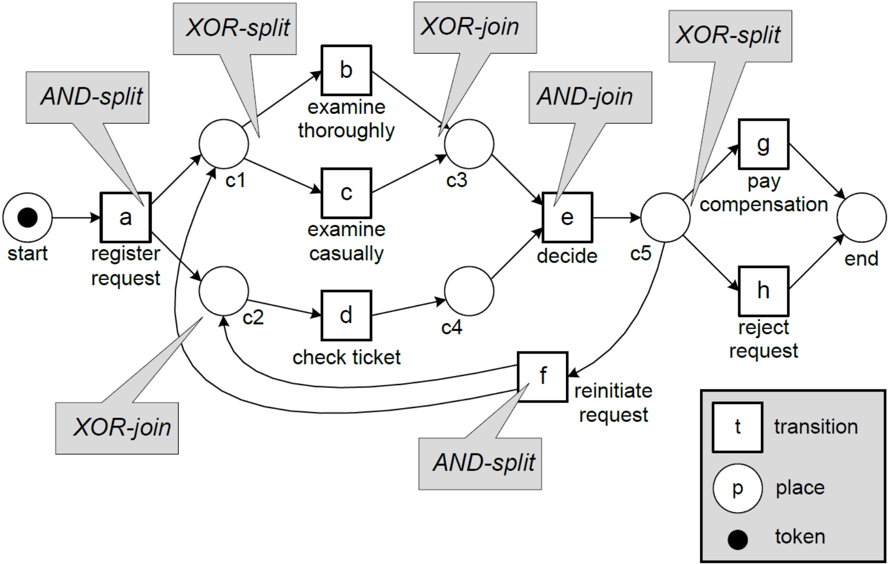
Choice: XOR-split



Choice: XOR-join



Logic with Petri nets



Iteration: 1 or more times

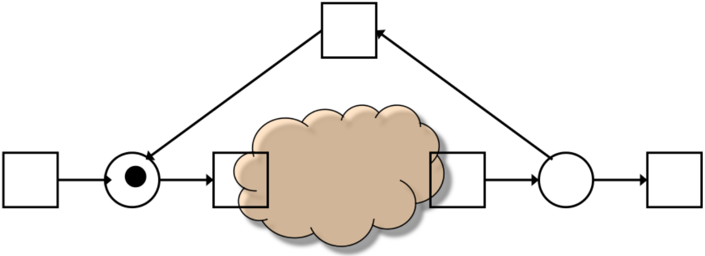


Figure : XOR-join before XOR-split

Iteration: 0 or more times

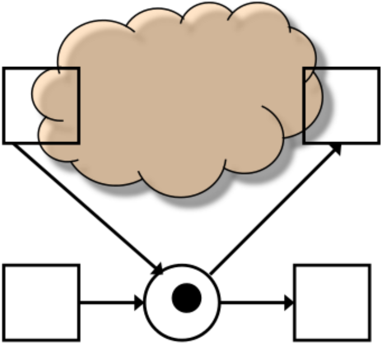


Figure : XOR-join before XOR-split

Capacity constraints: feedback loop

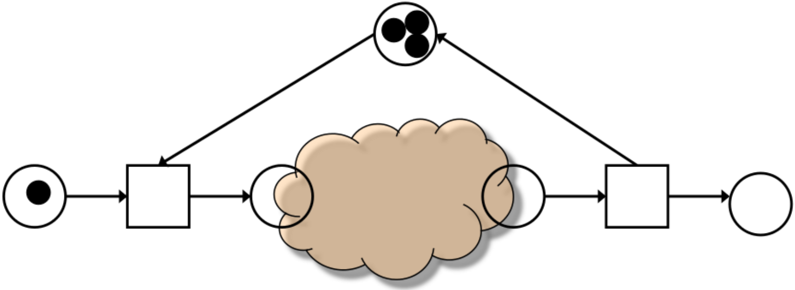


Figure : AND-join before AND-split

Capacity constraints: mutual exclusion

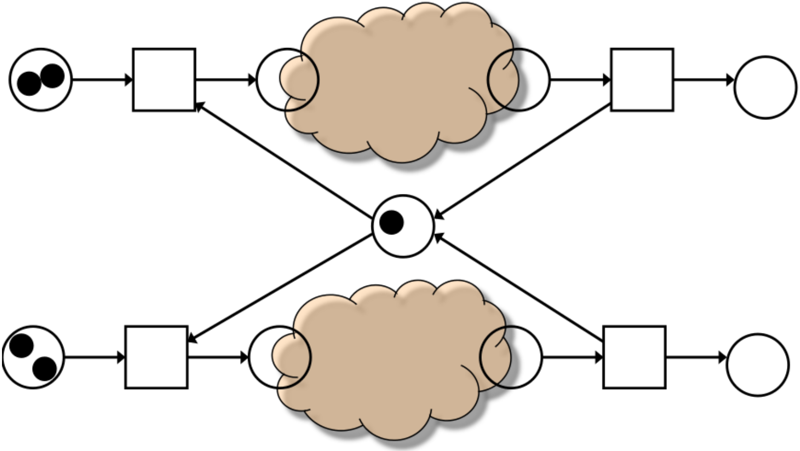
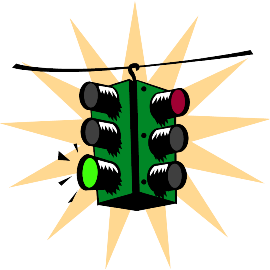
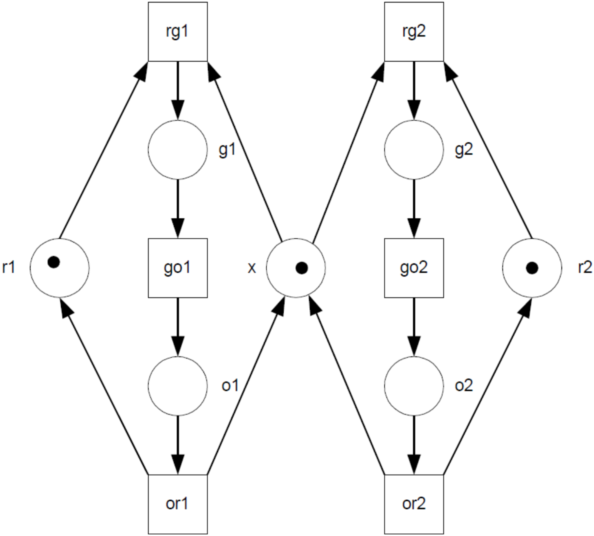


Figure : AND-join before AND-split

Two traffic lights: mutual exclusion

- Petri net for **2 traffic lights**





Capacity constraints: alternating

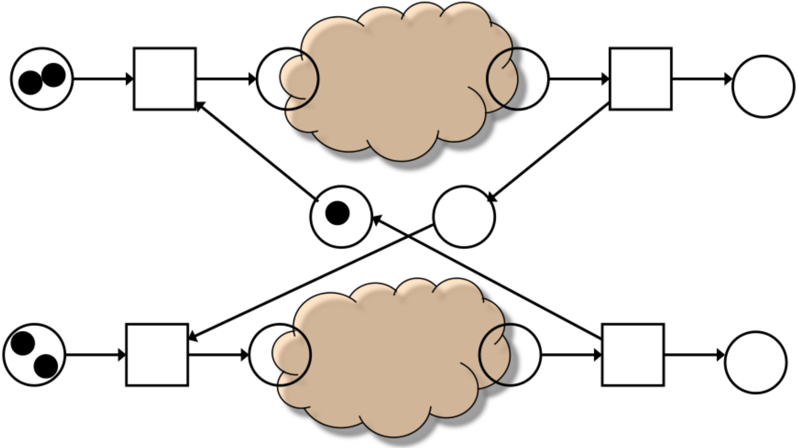
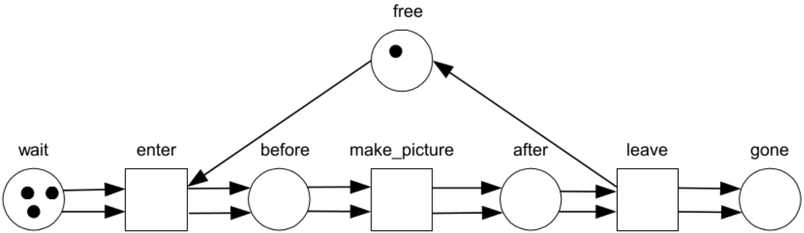


Figure : AND-join before AND-split

Petri nets with multiple arcs

- Petri net $N = (P, T, F')$ with multiple arcs
  - P is a finite set of **places**
  - T is a finite set of **transitions**
  - F' is a multiset over F, i.e., $F' \in \mathbb{B}(F)$ or a finite set $F' \subseteq (P \times T \times \mathbb{N}) \cup (T \times P \times \mathbb{N})$



Some properties

- In k-**bounded** Petri nets, no place ever holds more than k tokens
- A marked Petri net is **safe** if it is 1-bounded
- A marked Petri net is **deadlock-free** if at every reachable marking at least one transition is enabled
- A transition t in a marked Petri net is **live** if from every reachable marking, it is possible to enable t
- **WorkFlow-net** (WF-net): Petri net with fixed source $i \in P$ (without inputs) and target $o \in P$ (without outputs)

Variants and extensions

- Colored Petri nets (K. Jensen)
- Petri nets with inhibitor arcs (realizing Turing completeness)
- Prioritized Petri nets
- Timed Petri nets
- . . . and many more!

Lab session April 6

- Assess constructed features
- Create feature correlation plot
- Insert features and target attribute into SCIKIT-LEARN
- Perform final machine learning steps
- Next week: last chance to ask questions

Credits

Lecture partially based on (slides of the (previous edition of the)) course book: W. van der Aalst, *Process Mining: Data Science in Action*, 2nd edition, Springer, 2016.