High-level Petri nets
Extending Petri nets with color and time

prof.dr.ir. Wil van der Aalst
www.vdaalst.com
Classical Petri Nets => High-Level Petri Nets

Service help desk

Classical Petri net may be too abstract, may become too complex, or is simply unable to express relevant behavior.

extensions needed to model data and time and to structure larger models
Limitations of classical Petri nets
Large network size

Duplicate structure to distinguish individual products → size linear in the number of products
Limited expressive power (1)

Inability to model choices well.

- condition_holds
- condition_doesNotHold
Limited expressive power (2)

Zero testing: solutions only if $p$ is bounded

\[ t \rightarrow \square \rightarrow ? \]

\[ \square \rightarrow \bigcirc \rightarrow \square \]

\[ p \]
No explicit modeling of time

Duration of each phase is highly relevant
High-level Petri nets

- Extend Petri nets with
  - Color (i.e., data)
  - Time
  - Hierarchy (later, to structure large models)

- This lecture:
  - No concrete language but focus on the main concepts
  - Concepts appear in most modeling languages

- Next lecture:
  - A concrete language, **CPN**
  - Concepts supported by many variants of CPN, including ExSpect and CPN AMI
Initial Example
Classical Petri net
Adding Color

Auxiliary
Create
Declare
Hierarchy
Monitoring
Net
Simulation
State space
Style
View

Help
Options

deadoralive-color.cpn
Step: 0
Time: 0
Options
History

Declarations
Standard priorities
Standard declarations

\texttt{colset UNIT = unit;}
\texttt{colset INT}
\texttt{colset BOOL}
\texttt{colset STRING}
\texttt{colset Name = string;}
\texttt{colset Age = int;}
\texttt{colset Person = product Name * Age;}
\texttt{var n:Name;}
\texttt{var a:Age;}
\texttt{var p:Person;}

Monitors

\texttt{colset Name = string;}
\texttt{colset Age = int;}
\texttt{colset Person = product Name * Age;}
\texttt{var n:Name;}
\texttt{var a:Age;}
\texttt{var p:Person;}

TU/e
Technische Universiteit Eindhoven
University of Technology
Adding Time

[Diagram showing a CPN model with transitions and places labeled such as 'alive', 'death', 'birthday', 'dead', 'funeral', 'grave', 'holy', and 'open gate'.]

- **Simulator**
- **Step:** 0
- **Time:** 0
- **Options**
- **History**
- **Declarations**
  - **Standard priorities**
  - **Standard declarations**
    - `colset UNIT = unit timed;`
    - `colset INT`
    - `colset BCOL`
    - `colset STRING`
    - `colset Name = string;`
    - `colset Age = int;`
    - `colset Person = product Name * Age timed;`
    - `var n:Name;`
    - `var a:Age;`
    - `var p:Person;`
- **Monitors**
  - `main`
Adding Hierarchy
Adding color (informal)
Running example: Service help desk

- Waiting patients
- Patient being helped by an employee
- Served patients
- Free desk employees

Diagram:
- Wait
- Start
- Stop
- Done
- Busy
- Free
Every token has a **color** (i.e., a data value)

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)
Every place has a color set (i.e., place type)

(641112, 7) free

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)

Patient = Id \times Name \times Address \times DateOfBirth \times Gender

Employee = EmpNo \times Experience
Enabledness

Transition is **enabled** if enough tokens reside in its input places

(Employee, Patient)

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)
Transition firing: start

Consume a patient from place wait and an employee from place free, and produce a pair (Employee, Patient) in place busy.

Consume a pair (Employee, Patient) from place busy, and produce the employee in place free and the patient in place done.

((641112, 7), (12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male))

Consume a patient from place wait and an employee from place free, and produce a pair (Employee, Patient) in place busy.

Consume a pair (Employee, Patient) from place busy, and produce the employee in place free and the patient in place done.
Transition firing: stop

(641112, 7) free

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)
Product = ProdType × Quality

If there is a low-quality product in place unchecked, then consume it and produce it in place low.

(A, low) (A, low) (B, low) (C, high)
unchecked

check_low low

check_high high

If there is a high-quality product in place unchecked, then consume it and produce it in place high.
General firing rule

Enabledness condition (guard)

Which token? (based on token value)
How many tokens?

How many tokens?
Which token value?
(data manipulation)
Adding color
(a bit more formal)
Markings and net structure formally

**Boolean expression**

- **transition guard**
- **marking** *(multiset over place type)*
- **place type**
- **arc inscription**

**expression evaluated to a multiset**

may contain constants and variables; by assigning a value to each variable, the value of this expression can be calculated

Although arc inscriptions evaluate to multisets, the inscriptions typically evaluate to singletons. Hence, notation does not enforce the multiset notation in case only one token is consumed or produced.
Tokens in a place form a multiset

\[
\begin{align*}
(A, \text{low}), & (B, \text{low}), (C, \text{high}) \\
\end{align*}
\]
Consider a wallet.

A universe $U$ is a set containing all possible values.

$$U = \{ \text{€2}, \text{€1}, \text{€50}, \text{€20}, \text{€10}, \text{€5}, \text{€2}, \text{€1} \}$$

A multiset over $U$ is a mapping $f : U \rightarrow \mathbb{N}$.

Examples:

- $f(\text{€2}) = 2$
- $f(\text{€10}) = 1$
Markings: Only one place type

Implies that all places have the **same** universe

\[ U = \{ (A, \text{low}), (B, \text{low}), (C, \text{low}), (A, \text{high}), (B, \text{high}), (C, \text{high}) \} \]

unchecked: \( f((A, \text{low})) = 2; f((B, \text{low})) = 1; f((C, \text{high})) = 1; f((A, \text{high})) = 0; \ldots \)
Markings: More than one place type

For each place type, we have a universe

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)

3 universes: Employee, Patient, Employee × Patient with sets of multisets $M(\text{Employee}), M(\text{Patient}), M(\text{Employee} \times \text{Patient})$

Let $M = M(\text{Employee}) \cup M(\text{Patient}) \cup M(\text{Employee} \times \text{Patient})$

Then, marking $m: P \rightarrow M$
Arc inscriptions

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)

\( y \) evaluates to multiset \([(12345, \ldots, \text{male})]\)

\( x \) evaluates to multiset \([(641112, 7)]\)

variable of type Patient

variable of type Employee

(\( x, y \) \( (x,y) \) (Employee, Patient))
Transition guard

Evaluates to true if a low/high-quality product is assigned to $x$
Behavior formally

Boolean expression

transition guard

marking (multiset over place type)

place type

arc inscription

expression evaluated to a multiset

may contain constants and variables; by assigning a value to each variable, the value of this expression can be calculated
Enabledness at a binding

- **Idea**: we can assign more than one value to a variable (because places may contain tokens of different value)
- **Binding** is one concrete assignment
Enabledness at a binding

Binding = concrete assignment of tokens to variables

(employee, patient)

(12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)

(start, \{ x = (641112, 7), y = (12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male) \})

How many bindings if wait and free both contain three tokens

Treat value of x and y as a multiset

(employee, patient)
Recall: for all places $p \in P$, 

$$m'(p) = m(p) - w((p, t)) + w((t, p))$$

$$m'(\text{free}) = m(\text{free}) - [(641112, 7)] + []$$
$$= [(641112, 7)] - [(641112, 7)] + []$$
$$= []$$

Recall: (641112, 7), (12345, Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male)
Enabledness in the presence of guards

Enabledness requires guard to be evaluated to true

Three bindings of transition check_high:

\[(\text{check\_high}, \langle x = (A, \text{low}) \rangle)\] not enabled
\[(\text{check\_high}, \langle x = (B, \text{low}) \rangle)\] not enabled
\[(\text{check\_high}, \langle x = (C, \text{high}) \rangle)\] enabled
Firing at enabled binding \((\text{check\_high}, \langle x = (C, \text{high}) \rangle)\) yields:

1. **[x.Quality=low]**
   - (A, low)
   - (A, low)
   - (B, low)
   - (C, high)

2. **[x.Quality=high]**
   - (A, low)
   - (A, low)
   - (B, low)
   - (C, high)
Guard may only use variables appearing in the surrounding arc inscriptions.

\[ x.\text{Quality}=\text{low} \]

\[ x.\text{Quality}=\text{high} \]

\( y.\text{Quality}=\text{high} \) would be invalid, for instance
Definition: Colored Petri net

A colored Petri net is a Petri net in which every place has a type, and every token has a value (i.e., a color) that complies to the place type.

An arc in a colored Petri net may have an arc inscription. An arc inscription is an expression with some variables that evaluates to a multiset.

A transition can have a guard. A guard is a Boolean expression, and it may have variables in exactly the same way than arc inscriptions (of arcs surrounding that transition) have.
Definition: Binding

Let $t$ be a transition of a colored Petri net. A binding of $t$ assigns a concrete value to each variable that occurs in the arc inscriptions of arcs surrounding $t$. These values should be of the corresponding type.

Transition $t$ is enabled at a binding if there are tokens matching the values of the arc inscriptions and the guard of $t$ evaluates to true.

An enabled transition fires while consuming and producing the corresponding tokens.
Example: Inventory management system

Give a colored Petri net modeling this system!
Solution

```
Solution

stock

```

```
wheel
steering_wheel
frame

```

```
bike
wheel
bell

```

```
bell

```

```
increase

```

```
decrease

```

```
out

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```
Petri nets vs Colored Petri nets

One place type: BlackDot

$U = \{ \bullet \}$

$m: P \rightarrow M(\{\bullet\})$

$m(\text{wait}) = [1 \cdot \bullet]$  

Suffices to specify the number of tokens in each place: $m : P \rightarrow \mathbb{N}$

Petri nets are a special case of colored Petri nets!
Adding time
• **Global clock**

• **Time passes in time units** (concrete value of a time unit depends on specification)

• **At every time unit, check whether transition can fire**
What one would intuitively expect, … but is not true

Firing of each transition takes time (e.g., 2 time units)

Problem:
- Firing not atomic
- Only partial marking visible
- Several analysis techniques not applicable

→ Timestamps
Each token has a **timestamp**

Timestamp specifies the **earliest time when token is available (can be consumed)**
When is transition start enabled?

Enabledness as for (untimed) Petri nets; now, also consider availability of tokens

At time 2 (this is the enabling time of this transition)
Enabling time

The enabling time of a transition $t$ is the time point at which its input places contain enough available tokens such that $t$ is enabled.

$\rightarrow$ The enabling time of a transition is the maximum timestamp of the tokens to be consumed.
Enabledness and firing

- If there are multiple tokens in a place, only the ones **available** can be consumed.
- A transition with the **smallest** enabling time will fire first.
- As soon as a transition is enabled it **must** fire (unless disabled).

The **firing time** of a transition $t$ is maximal timestamp of the tokens consumed.
Example: firing start

Time 0:

Passing of time does not necessarily change the marking (e.g., Time 0 and 1)

Time 2:

\[
\text{max}(0, \min(2, 4, 4))
\]
Example: Different enabling times

Time 0

Time 1

Time 2

Time 4
Goal: Firing of transitions should take time

Realization: atomic firing but delay the availability of produced tokens by assigning a timestamp = firing time + specified delay to them
Delay by example: start fires

\[ 5 = \text{firing time} + \text{delay} = 2 + 3 \]
What is the final marking?
Runs in Petri nets with and without time

Is every run in a Petri net with time also a run in the Petri net without time?

And the other way around?

\[ \langle t_1, t_1, t_2 \rangle \] shows a counterexample
Petri nets with color and time
Running example

Time for making a punch card depends on the experience of the employee

\[
d(x) = \begin{cases} 
3 & \text{if } x.\text{Experience} > 5 \\
4 & \text{else }
\end{cases}
\]

start fires at time 2, yielding a token \((641112, 7), (12345, \text{Peter, "Kerkstraat 10, Amsterdam", 13-Dec-1962, male})\) with timestamp 5 in busy
Definition: Enabledness at a binding

A transition $t$ in a Petri net with color and time is enabled at a binding if there does not exist any other binding (of $t$ or of any other transition) with a smaller enabling time.
System model

data model

process model
**Data and process model of the help desk**

Data model represents basic entities of a system together with its interrelationships.

Tokens and places of a colored Petri net are related to the data model.
Modeling time

- Time may be part of a system specification (e.g., 3 seconds after pushing the red button, the door opens).
- Time may be added to model the environment of a system, e.g., for business process simulation purposes (e.g., the average time between two subsequent arriving customers is 4 minutes).
- Time is often related to routing probabilities, e.g., there may be race conditions (if the customer does not respond within three days his account is closed).
Sneak Preview
Extend the example

<table>
<thead>
<tr>
<th>name of train</th>
<th>capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>whale</td>
<td>250</td>
</tr>
<tr>
<td>dolphin</td>
<td>50</td>
</tr>
<tr>
<td>sea horse</td>
<td>50</td>
</tr>
</tbody>
</table>

one train per station
Add time

waiting time at

station 1  10
station 2  30
station 3  20
station 4  10

travel time from

station 1  40
station 2  50
station 3  30
station 4  45
Note that people only hop on or hop off just before train leaves.
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lectures</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>29-4-2013</td>
<td>Inst.</td>
<td>TU/e closed Modeling with Petri nets (3)</td>
</tr>
<tr>
<td></td>
<td>2-5-2013</td>
<td></td>
<td>Modeling with Petri nets</td>
</tr>
<tr>
<td></td>
<td>3-5-2013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>6-5-2013</td>
<td>Lect.</td>
<td>Extending Petri nets with color and time (4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TU/e closed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TU/e closed</td>
</tr>
<tr>
<td>20</td>
<td>13-5-2013</td>
<td>Exam</td>
<td>Colored Petri Nets (5)</td>
</tr>
<tr>
<td></td>
<td>16-5-2013</td>
<td></td>
<td>Colored Petri Nets (6)</td>
</tr>
<tr>
<td></td>
<td>17-5-2013</td>
<td>Inst.</td>
<td>Pre-exam focusing on classical Petri nets (1 point)</td>
</tr>
<tr>
<td></td>
<td>(14.00-15.30)</td>
<td></td>
<td>Explanation “CPN assignment” (3 points)</td>
</tr>
</tbody>
</table>

**Prepare for pre-exam**

Read Chapter 4 of book. Make all exercises in Section 2.

Read Chapter 5 of book.

Read Chapter 6 of book. Study Chapters 1-4 and all exercises in Sections 1-2. Start making exercises in Section 3.
After this lesson you should be able to

• Explain why Petri nets are not suitable for modeling complex information systems.
• Explain the concepts: place type, token value, arc inscription, guard, binding, time stamp, delay, and enabling time.
• Determine the reachable markings of a given Petri net extended with color and time for a given initial marking.
• Design a simple Petri net with color and time given a textual description.