Some Simple Extensions of Petri’s Cycloids

PNSE’20

Bjarne Jessen and Daniel Moldt
June 23, 2020

http://www.informatik.uni-hamburg.de/TGI/
Contents

Introduction

Objective and Context

Cycloids

Extensions

Applications for (Extended) Cycloids

Summary and Future Work
Introduction
Introduction

• Cycloids are special nets with cyclic behaviour: The main idea is to fold a special subnet of an infinite causal net (Petri space) in such a way that the behaviour of this initially marked subnet (cycloid) can be repeated infinitely often without losing or gaining any subnet behaviour and to stay bounded.

• Already considered by Petri himself

• Kummer and Stehr (1997) and Valk (2018, 2019, 2020) conduct further work
Objective and Context
Objective and Context

- Cycloids contain concurrency, but no conflicts (alternative actions).
- We consider the modelling of conflicts in cycloids as an extension\(^1\).
- We give a context of Petri net-based software engineering.

\(^1\)Some more basic and more formal results will be discussed in the talk of Valk at the main Petri net conference.
Petri net-based software engineering

are context of

distributed systems

are special
cycloids

are not considered yet

alternative actions / conflicts

implementation through specification

provide

executable model

provide

model

Petri nets

provide

implementation through refinement

infrastructure

protocol

platform

agent

MULAN architecture

contains

used in

application of
Cycloids
Cycloids model recurrent behaviour containing causal dependencies and concurrency. Formally, they are foldings of an infinite Petri net (Petri space).

→ Four parameters specify how the infinite Petri space is folded.

→ Different cycloids can be generated / chosen from the same Petri Space. The choice represents the desired modelling perspective of the modellers.
Cycloids

Infinite model of driving cars
(Space horizontal, time vertical)

Cars blue, gaps grey
Depiction on the right hand side shows excerpt of the following Petri space:

Bjarne Jessen and Daniel Moldt (Some Simple Extensions of Petri’s Cycloids)
Infinite model of driving cars
(Space horizontal, time vertical)

Cars blue, gaps grey
Depiction on the right hand side shows
excerpt of the following Petri space:
Cycloids

The Petri space is a Petri net

\[ PR = (P_r, T_r, F_r, W_r), \]

which is defined as follows:

\[ P_r = P_r^\rightarrow \cup P_r^\leftarrow, \]

with \( P_r^\rightarrow = \{ p_{\xi,\eta}^\rightarrow | \xi, \eta \in \mathbb{Z} \} \)

and \( P_r^\leftarrow = \{ p_{\xi,\eta}^\leftarrow | \xi, \eta \in \mathbb{Z} \} \), \( P_r^\rightarrow \cap P_r^\leftarrow = \emptyset \),

\[ T_r = \{ t_{\xi,\eta} | \xi, \eta \in \mathbb{Z} \}, \]

\[ F_r = \{ (t_{\xi,\eta}, p_{\xi,\eta}^\rightarrow) | \xi, \eta \in \mathbb{Z} \} \]

\( \cup \{ (p_{\xi,\eta}^\rightarrow, t_{\xi+1,\eta}) | \xi, \eta \in \mathbb{Z} \} \)

\( \cup \{ (t_{\xi,\eta}, p_{\xi,\eta}^\leftarrow) | \xi, \eta \in \mathbb{Z} \} \)

\( \cup \{ (p_{\xi,\eta}^\leftarrow, t_{\xi,\eta+1}) | \xi, \eta \in \mathbb{Z} \} \),

\[ W_r(x, y) = 1, \ \forall (x, y) \in F_r. \]
For $\alpha, \beta, \gamma, \delta \in \mathbb{N} \setminus \{0\}$ we define an equivalence relation $\equiv$ on $X_r$ for $X_r = P_r \cup T_r$:

\begin{align*}
\equiv[P_r^\rightarrow] & \subseteq P_r^\rightarrow, \\
\equiv[P_r^\leftarrow] & \subseteq P_r^\leftarrow, \\
\equiv[T_r] & \subseteq T_r, \\
x_{\xi, \eta} & \equiv x_{\xi + m\alpha + n\gamma, \eta - m\beta + n\delta}, \forall \xi, \eta, m, n \in \mathbb{Z}.
\end{align*}
Cycloids

Then a particular net \((P, T, F, W)\) is defined as a cycloid \(C(\alpha, \beta, \gamma, \delta)\):

- \(P\) contains the places of \(X_r/\equiv\),
- \(T\) contains the transitions of \(X_r/\equiv\),

\((\llbracket x \rrbracket \equiv, \llbracket y \rrbracket \equiv) \in F \iff \exists x' \in \llbracket x \rrbracket \equiv \exists y' \in \llbracket y \rrbracket \equiv : (x', y') \in F_r,

\(W(x, y) = 1, \forall (x, y) \in F\).

Furthermore \(\alpha\) and \(\beta\) specify a standard initial marking for the cycloid. The maximal distance between cars is restricted by the number of available spaces (\(\text{cars} + \text{gaps} = \text{spaces}\)).

Example: \(C(2, 3, 1, 1)\)

Bjarne Jessen and Daniel Moldt (Some Simple Extensions of Petri’s Cycloids)
Extensions
### Overview

<table>
<thead>
<tr>
<th>Model</th>
<th>Focus on</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$ cars, $n$ gaps, one Lane</td>
<td>concurrency</td>
</tr>
<tr>
<td>multiple independent lanes, no crossing of lanes</td>
<td>concurrency</td>
</tr>
<tr>
<td>additional overtaking bay of length one, one car overtakes $m$ cars</td>
<td>conflict</td>
</tr>
<tr>
<td>additional overtaking lane of length $l$, one car overtakes $m$ cars</td>
<td>conflict</td>
</tr>
<tr>
<td>guaranteed cut in</td>
<td>conflict</td>
</tr>
<tr>
<td>forced sheering out</td>
<td>conflict</td>
</tr>
<tr>
<td>guaranteed cut in + forced sheering out</td>
<td>conflict</td>
</tr>
<tr>
<td>multiple cars can overtake</td>
<td>conflict</td>
</tr>
<tr>
<td>multiple cars must overtake</td>
<td>conflict</td>
</tr>
</tbody>
</table>
## Overview

<table>
<thead>
<tr>
<th>Model</th>
<th>Focus on</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiple cars are located on one section</td>
<td>coarsening</td>
</tr>
<tr>
<td>sliding window of overtaking</td>
<td>conflict</td>
</tr>
<tr>
<td>crossing lanes freely</td>
<td>conflict</td>
</tr>
<tr>
<td>coloured cars</td>
<td>distinguishability</td>
</tr>
<tr>
<td>coloured cars + coloured gaps</td>
<td>distinguishability</td>
</tr>
<tr>
<td>oncoming traffic on additional lane</td>
<td>conflict</td>
</tr>
<tr>
<td>simultaneous actions of cars</td>
<td>synchronization</td>
</tr>
</tbody>
</table>
Beige places are modelled as virtual places for cars and gaps respectively.

Figure 1: Additional Lane

Bjarne Jessen and Daniel Moldt (Some Simple Extensions of Petri’s Cycloids)
Beige and red places are ”copied” for cars and gaps respectively.

Figure 2: Additional Lane
Crossing of Lanes

Figure 3: Crossing of Lanes

Beige and green places are "copied" for cars and gaps respectively.

Bjarne Jessen and Daniel Moldt (Some Simple Extensions of Petri’s Cycloids)
Figure 4: Crossing of Lanes

Beige and green places are "copied" for cars and gaps respectively.

Bjarne Jessen and Daniel Moldt (Some Simple Extensions of Petri’s Cycloids)
Beige and red places are "copied" for cars and gaps respectively.
Application: autonomously driving cars

Figure 5: Guaranteed Cut In
Beige and red places are "copied" for cars and gaps respectively.
Light blue and white places are "copied" for cars and gaps.
Applications for (Extended) Cycloids
Relevant modelling aspects of cycloids are:

- true concurrency
- no conflicts or other extensions
- simple structure
- difficult to represent
- require tool support
- low expressiveness
- formal background
- applicable for basic modelling
Relevant modelling aspects of extended cycloids are:

- true concurrency
- conflict place / virtual places / colours / synchronous channels etc.
- still simple structure
- even more difficult to represent
- require tool support
- increased expressiveness
- no formal background (yet)
- more applicable for modelling
Currently envisioned modelling areas

- basis for understanding concurrency
- object behaviour
- protocols between agents
- incremental scenario modelling support
- proof by construction principle in software engineering
- secure systems
What are the benefits of conflicts for cycloids?

- single cycloid: no alternative for leaving out behaviour, just contacts hinder tokens to move forward
- two cycloids: independent behaviour of two cycloids
- High redundancy when only two transitions differ in the set of transitions $T$
- solution: alternative parts of the cycloid
- general assumption: repetitive behaviour
Construction methods for extended cycloids

- replication (colouring)
- composition
- folding
- fusion
- synchronization
Generalisation:
A Unit or an entity has some behaviour creating a lifeline (Minkowski/Petri).

Idea:
Units are identified within the Petri space and modelled by (extended) cycloids.
Units are combined to units with the above mentioned methods.
Modellers identify the relevant units of a system.
Software Engineer can follow the proof-by-construction principle for secure systems.
Summary and Future Work
• Cycloids as special nets
• Options for modelling of recurrent behaviour described
• Conceptual extensions of cycloids are sketched
• General modelling paradigm for secure systems proposed
Future Work

- Tools for the generation of processes of nets
- Investigating formal properties of extended cycloids
- Modelling approach for provable systems based on proof by construction principles
- Tools for the support of modelling, simulation and analysis of extended cycloid models

Olaf Kummer and Mark-Oliver Stehr.  
**Petri’s axioms of concurrency- A selection of recent results.**  

Olaf Kummer.  
**Referenznetze.**  
Wolfgang Reisig.

*Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien.*

Rüdiger Valk.

*On the structure of cycloids introduced by carl adam petri.*
Rüdiger Valk.  
**Formal properties of petri’s cycloid systems.**  

Rüdiger Valk.  
**Circular traffic queues and Petri’s cycloids.**  