Automatic generation of formal models for diagnosability of DES

23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2018)

R. Nardone\textsuperscript{1}, G. De Tommasi\textsuperscript{1}, N. Mazzocca\textsuperscript{1}, A. Pironti\textsuperscript{1}, V. Vittorini\textsuperscript{1}

\textsuperscript{1}Dipartimento di Ingegneria Elettrica e delle Tecnologie dell’Informazione
Università degli Studi di Napoli Federico II, Italy

Torino - 6 September 2018
Model Driven Engineering

- **Model-Driven Engineering (MDE)** is a software engineering paradigm where models are the key entities to implement a software system throughout the development process.

- MDE relies on:
  - modeling languages to describe a system at different levels of abstraction
  - **Model-to-Model (M2M) and Model-to-Text (M2T)** transformations to create bridges between different abstraction levels and/or technological spaces → to provide efficient and automated procedure to produce artifacts from other artifacts.

- During the last two decades MDE approaches have been promoted in different fields:
  - manufacturing systems
  - electronic systems
  - automotive
  - embedded and control systems
There is a research trend that integrates formal methods (FM) with MDE approaches, in order to take advantages from both

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MDE</strong></td>
<td>* Lack of semantics</td>
</tr>
<tr>
<td>* User-friendly notation</td>
<td>* Unfit for model analysis</td>
</tr>
<tr>
<td>* Derivative artifacts for</td>
<td></td>
</tr>
<tr>
<td>tool development</td>
<td></td>
</tr>
<tr>
<td>* Automated model transformations</td>
<td></td>
</tr>
<tr>
<td><strong>FM</strong></td>
<td>* Hard notation</td>
</tr>
<tr>
<td>* Rigorous mathematical</td>
<td>* Lack of tools</td>
</tr>
<tr>
<td>foundation</td>
<td>* Lack of integration</td>
</tr>
<tr>
<td>* Suitable for model analysis</td>
<td></td>
</tr>
</tbody>
</table>

Figure: Gargantini et al., ICSEA2009.
Aim: definition of *model-driven processes* that can be applied to automatically generate and analyze formal models in many application domains

This direction is hard to go, as FM development is a not fully engineered field, unlike software development, this despite the scientific community has been working for decades for a more widespread adoption of FM in industry and the need for FM has always been declared (especially in critical system development)
A model-driven approach for the automatic generation of FM for diagnosability in the discrete event systems (DES) context.

The ultimate goal of the proposal is to enable the analysis of critical systems by supporting modelers in the definition of a high-level specification of the system.

Starting from this specification, FM for different kinds of analysis can be generated by exploiting automatic transformation chains.
The case study of a railway benchmark is used to deal with diagnosability of fault in DES.

The technique proposed in Basile et al., Automatica 2012 that relies on Petri net (PN) models of the system is first used to assess diagnosability.

The proposed approach relies on the solution of Integer Linear Programming (ILP) problems.

- Although the approach proved to be numerically efficient, it cannot be used to detect non diagnosable faults.
- It cannot be used to assess diagnosability of all the faults in the considered benchmark.
This motivated the presented model-driven approach

- It enables the generation of models that use different FM wrt to PNs
- It can be used to apply different analysis techniques
- A Promela benchmark is derived to apply model checking techniques

- Dynamic STate Machine (DSTM) is used as source specification language *Benerecetti et al.*, SCP-2017, which permits to derive different target models
Originally proposed in *Leveson and Stolzy*, IEEE TSE, and recently adopted in *Boussif et al.*, DX2017 as a benchmark to assess the performance of different diagnosability algorithms.

**Modular** PN model of a railway system with:
- *n* tracks
- level crossing (LC) controller
- the barriers
The following fault events are modeled by unobservable transitions:

- The $i$-th transition $(t_i, 4, \text{ig})$ indicates that the $i$-th train enters the LC zone before the controller lowers the barriers;
- The transition $(t_6, \text{bf})$ indicates a defect in the barriers that results in a premature raising.

The proposed *optimization-based* approach cannot be used to assess non-diagnosability.

- The fault $(t_i, 4, \text{ig})$ is not diagnosable when $n > 1$.
- Only $(t_6, \text{bf})$ will be considered for the comparison.
Motivation & Contribution
The railway benchmark Diagnosability of PNs
MD Generation Approach Conclusions

PN notation

- $S = \langle N, m_0 \rangle$ is the net system, where $N = (P, T, \text{Pre}, \text{Post})$
- $T = T_o \cup T_{uo}$, and $T_f \subset T_{uo}$
- Given a firing count vector $\sigma \in \mathbb{N}^n$, we would like to consider only firings of either observable or unobservable transitions. The following notation is introduced:

\[
\sigma|_{T_o} \in \mathbb{N}^n, \text{ with } \sigma|_{T_o}(t) = \begin{cases} 
\sigma(t) & \text{if } t \in T_o \\
0 & \text{if } t \notin T_o
\end{cases}
\]

\[
\sigma|_{T_{uo}} \in \mathbb{N}^n, \text{ with } \sigma|_{T_{uo}}(t) = \begin{cases} 
\sigma(t) & \text{if } t \in T_{uo} \\
0 & \text{if } t \notin T_{uo}
\end{cases}
\]
Labeled PNs

- $G = \langle N, m_0, \lambda \rangle$ is a labeled Petri net (LPN) system
- $\lambda : T \mapsto E \cup \{\varepsilon\}$ is the labeling function
  - $\lambda(\cdot)$ assigns to each transition $t \in T$ either an event in $E$ or the silent event $\varepsilon$
  - $\lambda(t) = \varepsilon$ if $t \in T_{uo}$, while $\lambda(t) \neq \varepsilon$ otherwise
- We denote with $$T^\alpha = \{ t \in T \mid \lambda(t) = \alpha \} ,$$ the set of transitions associated with the same event $\alpha \in E$.
- $w$ denotes a word of events associated with a sequence $\sigma$ such that $w = \lambda(\sigma)$
- $|w|$ denotes the length of $w$, while $|w|_\alpha$ denotes the number of occurrences of the event $\alpha$ in $w$
Diagnosability - Definition 1/3

- \( L/u = \{ v \in T^* \text{ s.t. } uv \in L \} \), is the post-language of \( L \) after the sequence of transitions \( u \).

- \( Pr : T^* \mapsto T_o^* \) is the usual projection that erases the unobservable transitions in a sequence \( u \).

- The inverse projection operator \( Pr_L^{-1} \) is defined as

  \[
  Pr_L^{-1}(r) = \{ u \in L \text{ s.t. } Pr(u) = r \}
  \]

- Let \( \dot{u} \) be the final transition of sequence \( u \) and define

  \[
  \Psi(\hat{t}) = \{ u \in L \text{ s.t. } \dot{u} = \hat{t} \}
  \]
### Definition (Diagnosable fault)

A fault transition $t_f \in T_f$ is said to be diagnosable if

$$\exists \ h \in \mathbb{N} \ \text{such that} \ \forall \ u \in \Psi(t_f) \ \text{and} \ \forall \ v \in L/u \ \text{with} \ |v| \geq h,$$

it is

$$r \in Pr_L^{-1}(Pr(uv)) \Rightarrow t_f \in r.$$
<table>
<thead>
<tr>
<th>Definition ((\mathcal{K})-diagnosable fault)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given ( t_f \in T_f ) and ( \mathcal{K} \in \mathbb{N} ) (i.e., the maximum length of the postfix is given), ( t_f ) is said to be ( \mathcal{K})-diagnosable if</td>
</tr>
<tr>
<td>( \forall \ u \in \Psi(t_f) ) and ( \forall \ v \in L/u ) such that (</td>
</tr>
<tr>
<td>then it is</td>
</tr>
<tr>
<td>( r \in \Pr_{L}^{-1}(\Pr(uv)) \Rightarrow t_f \in r ).</td>
</tr>
</tbody>
</table>
\(K\)-diagnosability via solution of ILP problems 1/3

- Originally proposed in Basile et al., Automatica-2012
- Gives a necessary and sufficient condition to check \(K\)-diagnosability in **bounded and live** labeled net systems
- Cannot be used to assess non-diagnosability
\(\mathcal{K}\)-diagnosability via solution of ILP problems 2/3

- A labeled bounded and live net system \(G = \langle N, m_0, \lambda \rangle\)
- A fault transition \(t_f\)
- A positive integer \(J\) such that inequalities (1) (denoted with \(\mathcal{F}(m_0, t_f, J, K)\)) describe the set

\[
\mathcal{M}(t_f) = \left\{ m \in \mathbb{N}^m \mid (m_0[u]m) \land (t_f \notin u) \land (m[t_f]) \right\}
\]

\[
m_0 \geq \text{Pre} \cdot u_1
\]
\[
m_0 + C \cdot u_1 \geq \text{Pre} \cdot u_2
\]
\[
\ldots
\]
\[
m_0 + C \cdot \sum_{i=1}^{J-1} u_i \geq \text{Pre} \cdot u_J
\]  
(1a)

\[
m_0 + C \cdot \sum_{i=1}^{J} u_i \geq \text{Pre}(\cdot, \hat{t})
\]

\[
m_0 + C \cdot \sum_{i=1}^{J} u_i + C(\cdot, \hat{t}) \geq \text{Pre} \cdot v_1
\]

\[
\text{Pre} \cdot v_2
\]
\[
\ldots
\]

\[
m_0 + C \cdot \sum_{i=1}^{J} u_i + C(\cdot, \hat{t}) + C \cdot \sum_{j=1}^{J-1} v_j \geq \text{Pre} \cdot v_K
\]

\[
\sum_{i=1}^{J} u(\hat{t}) = 0
\]  
(1b)

\[
\sum_{j=1}^{K} v_j \geq K
\]  
(1c)

\[
\| \sum_{j=1}^{K} v_j \|_1 \geq K
\]  
(1d)

\[
Gianmaria De Tommasi – detommas@unina.it
\]  
17 of 26
Theorem

Given a positive integer $K$, $t_f$ is $K$-diagnosable if and only if there exist $3(J + K)$ vectors $u_1, \ldots, u_J, v_1, \ldots, v_K, \epsilon_1, \ldots, \epsilon_{J+K}, s_1, \ldots, s_{J+K} \in \mathbb{N}^n$ such that

$$\min \sum_{r=1}^{J+K} \epsilon_r(t_f) \neq 0,$$

where the set $LD(m_0, t_f, J, K)$ includes $F(m_0, \hat{t}, J, K)$ and other similar linear constraints.
Numerical results

<table>
<thead>
<tr>
<th>nr. of tracks</th>
<th>nr. places</th>
<th>nr. of transitions</th>
<th>$\mathcal{K}$</th>
<th>Time needed to assess $\mathcal{K}$-diagnosability (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td>3.2</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>7.3</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>15.7</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>22</td>
<td>25</td>
<td>30.8</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>26</td>
<td>31</td>
<td>56</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>30</td>
<td>37</td>
<td>95.1</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>34</td>
<td>43</td>
<td>148.5</td>
</tr>
<tr>
<td>8</td>
<td>33</td>
<td>38</td>
<td>49</td>
<td>226.2</td>
</tr>
<tr>
<td>9</td>
<td>36</td>
<td>42</td>
<td>55</td>
<td>331.6</td>
</tr>
<tr>
<td>10</td>
<td>39</td>
<td>46</td>
<td>61</td>
<td>468.8</td>
</tr>
</tbody>
</table>

Figure: Results of the numerical experiments run to assess diagnosability of the fault $(t_6, b\bar{f})$ by solving the ILP problems. The ILP problems have been solved by using FICO™ Xpress on a standard PC equipped with an Intel® i7 processor at 3.4 GHz, and 8 GB of RAM running Windows 10 at 64 bit.
The proposed model-driven approach proposed in this paper relies on Dynamic STate Machine (DSTM) as specification language.

Exploiting the modularity of the original PN model, three DSTM sub-models are provided (one for the railway controller, one for the barrier and one for a generic track).

Multi-track benchmark is easily realized (instantiating as many track sub-models as needed).

Once the DSTM model of the railway traffic is defined, it is translated both into a PN model and a Promela model by defining and applying two M2M transformations.

The transformation from DSTM to Petri Nets (DSTM2PN) has been defined as part of this preliminary work (more details in the paper).
In the DSTM domain it is possible to specify different control strategies and reflect them to the original PN domain.

In the Promela domain it is possible to perform test and verification that are not enabled in the original domain (check non diagnosability, by using diagnoser-based approaches such as the one proposed in Sampath et al., IEEE TAC-1995 or Cabasino et al., IEEE TAC-2012).
DSTM models

Figure: DSTM specification of the benchmark components.
Example of Promela model

```
proctype controller(pid parent; mtype initial;chan chTerm) {
    ...
    do
        :: (state == controller_init && HasToken[_pid]==1) ->
            ...
        :: (state == controller_waitEntSignal && HasToken[_pid]==1) ->
            ...
        :: (state == controller_waitExitSignal && HasToken[_pid]==1) ->
            atomic{
                HasToken[_pid]=0;
                if
                    :: (((TrackSignalIn?[]) && (TrackSignalOut?[[]])) ->
                        TrackSignalIn);
                    TrackSignalOut;
                    TrackGo!msg;
                    state = controller_waitExitSignal;
                fi;
            } ->
                :: (state == controller_sentUpSignal && HasToken[_pid]==1) ->
                    ...
            } od unless {{chTerm?[1]; ... }
```

Figure: Promela Controller (excerpt).

Gianmaria De Tommasi – detommas@unina.it
A preliminary results related to the realization of a model-driven approach to perform analysis of DES have been presented.

The proposed approach permits to perform analysis of the same system at different levels of abstraction, by means automatic transformations that start from the DSTM high-level model.

The case study of the railway Petri net model has been considered.
The work can be extended in several directions:

1. The derived model can be used to investigate advantages (and disadvantages) of combining different modelling and analysis techniques (example: to compare the efficiency of the state space exploration performed at both Petri Net and Promela level in order to identify the best trade-offs between the usage of these models).

2. The transformational approach will be enhanced and extended to consider the asynchronous instantiation of machines and allow for exploiting all the high-level features of the DSTM formalism.

3. Develop a complete model-driven analysis approach for diagnosability also allowing for partial specification, removing the necessity of defining the high-level models of all the components of the system under study (example: model just the controller and neither the barrier or the tracks, which could be represented by a set of constraints over the external environment).
Questions?