Automatic generation of formal models for diagnosability of DES

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

R. Nardone[1], G. De Tommasi[1], N. Mazzocca[1], A. Pironti[1], V. Vittorini[1]

[1]Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
Università degli Studi di Napoli Federico II, Italy

Torino - 6 September 2018

DIETI. UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

Outline

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

Model Driven Engineering & Formal Methods - 1/2

- There is a research trend that integrates formal methods (FM) with MDE approaches, in order to take advantages from both

	Advantages	Disadvantages
MDE	* User-friendly notation * Derivative artifacts for tool development * Automated model transformations	* Lack of semantics * Unfit for model analysis
FM	* Rigorous mathematical foundation * Suitable for model analysis	* Hard notation * Lack of tools * Lack of integration

Figure: *Gargantini et al.*, ICSEA2009.

Model Driven Engineering & Formal Methods - 2/2

- **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)

Contribution - 1/3

- A model-driven approach for the automatic feneration 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

Contribution - 2/3

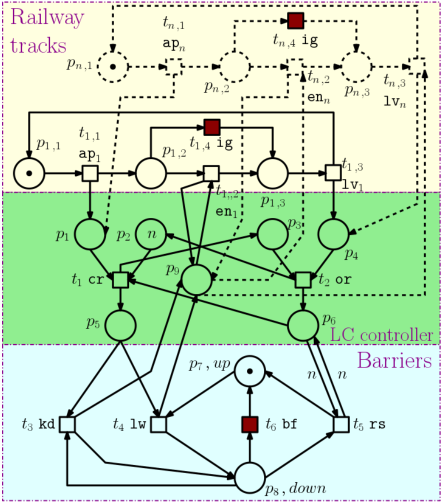
- The case study of a railway benchmark is used to deal with diagnosability of fault in DES
- The technique proposed in *Basile et al.*, Automatica 2012 that relies on Petri net (PN) models of the system is first used to assess diagnosability
- The proposed approach relies on the solution of Integer 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

Contribution - 3/3

- 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

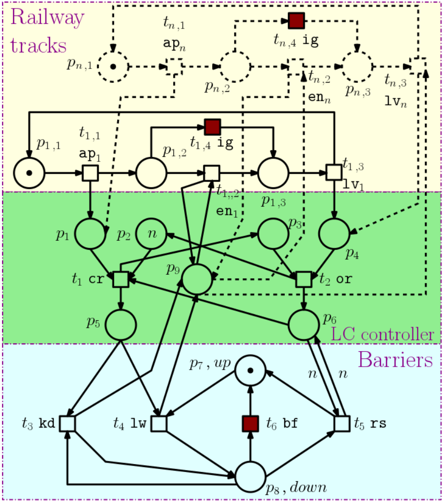
The railway benchmark - 1/2

- 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 railway benchmark 2/2

- The following fault events are modeled by unobservable transitions
  - the i-th transition ($t_{i,4}$, ig) indicates that the i-th train enters the LC zone before the controller lowers the barriers;
  - the transition (t_6 , 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 ($t_{i,4}$, ig) is not diagnosable when $n > 1$.
- **Only (t_6 , bf) will be considered for the comparison**



PN notation

- $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$ is the net system, where $N = (P, T, \mathbf{Pre}, \mathbf{Post})$
- $T = T_o \cup T_{uo}$, and $T_f \subset T_{uo}$
- Given a **firing count vector** $\boldsymbol{\sigma} \in \mathbb{N}^n$, we would like to consider only firings of either observable or unobservable transitions. The following notation is introduced:

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

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

Labeled PNs

- $G = \langle N, \boldsymbol{m}_0, \lambda \rangle$ 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* ε
  - $\lambda(t) = \varepsilon$ if $t \in T_{uo}$, while $\lambda(t) \neq \varepsilon$ otherwise
- We denote with

$$T^{\alpha} = \left\{ t \in T \mid \lambda(t) = \alpha \right\},$$

  the set of transitions associated with the same event $\alpha \in E$.
- w denotes a word of events associated with a sequence σ such that $w = \lambda(\sigma)$
- $|w|$ denotes the length of w, while $|w|_{\alpha}$ denotes the number of occurrences of the event α 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

$$\Psi(\hat{t}) = \left\{u \in L \text{ s.t. } \dot{u} = \hat{t}\right\}$$

Diagnosability - Definition 2/3

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

$$r \in Pr_L^{-1}\big(Pr(uv)\big) \Rightarrow t_f \in r\,.$$

Diagnosability - Definition 3/3

Definition ($\mathcal{K}$–diagnosable fault)

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

$$\forall\ u \in \Psi(t_f) \text{ and } \forall\ v \in L/u \text{ such that } |v| \geq \mathcal{K}\,,$$

then it is

$$r \in Pr_L^{-1}\big(Pr(uv)\big) \Rightarrow t_f \in r\,.$$

$\mathcal{K}$-diagnosability via solution of ILP problems 1/3

- Originally proposed in *Basile et al.*, Automatica-2012
- Gives a necessary and sufficient condition to check $\mathcal{K}$-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 $G = \langle N, \boldsymbol{m}_0, \lambda \rangle$
- A fault transition t_f
- A positive integer $\mathcal{J}$ such that inequalities (1) (denoted with $\mathcal{F}(\boldsymbol{m}_0, \hat{t}, \mathcal{J}, \mathcal{K})$) describe the set

$$\mathcal{M}(t_f) = \left\{ \boldsymbol{m} \in \mathbb{N}^m \mid \left(\boldsymbol{m}_0 [u\rangle \boldsymbol{m}\right) \bigwedge \left(t_f \notin u\right) \bigwedge \left(\boldsymbol{m}[t_f\rangle\right) \right\}$$

$$\begin{cases}
\boldsymbol{m}_0 \geq \mathbf{Pre} \cdot \boldsymbol{u}_1 \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{u}_1 \geq \mathbf{Pre} \cdot \boldsymbol{u}_2 \\
\dots & \text{(1a)} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}-1} \boldsymbol{u}_i \geq \mathbf{Pre} \cdot \boldsymbol{u}_{\mathcal{J}} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}} \boldsymbol{u}_i \geq \mathbf{Pre}(\cdot, \hat{t}) & \text{(1b)} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}} \boldsymbol{u}_i + \boldsymbol{C}(\cdot, \hat{t}) \geq \mathbf{Pre} \cdot \boldsymbol{v}_1 \\
\mathbf{Pre} \cdot \boldsymbol{v}_2 \\
\dots & \text{(1c)} \\
\boldsymbol{m}_0 + \boldsymbol{C} \cdot \sum_{i=1}^{\mathcal{J}} \boldsymbol{u}_i + \boldsymbol{C}(\cdot, \hat{t}) + \boldsymbol{C} \cdot \sum_{j=1}^{\mathcal{K}-1} \boldsymbol{v}_j \geq \mathbf{Pre} \cdot \boldsymbol{v}_{\mathcal{K}} \\
\sum_{i=1}^{\mathcal{J}} \boldsymbol{u}(\hat{t}) = 0 & \text{(1d)} \\
\left\| \sum_{j=1}^{\mathcal{K}} \boldsymbol{v}_j \right\|_1 \geq \mathcal{K} & \text{(1e)}
\end{cases}$$

$\mathcal{K}$-diagnosability via solution of ILP problems 3/3

Theorem

Given a positive integer $\mathcal{K}$, t_f is $\mathcal{K}$-diagnosable if and only if there exist $3(\mathcal{J}+\mathcal{K})$ vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\mathcal{J}}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\mathcal{K}}$, $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{\mathcal{J}+\mathcal{K}}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{\mathcal{J}+\mathcal{K}} \in \mathbb{N}^n$ such that

$$\min_{\text{s.t. } \mathcal{LD}\left(\boldsymbol{m}_0, t_f, \mathcal{J}, \mathcal{K}\right)} \sum_{r=1}^{\mathcal{J}+\mathcal{K}} \epsilon_r(t_f) \neq 0,$$

where the set $\mathcal{LD}(\boldsymbol{m}_0, t_f, \mathcal{J}, \mathcal{K})$ includes $\mathcal{F}\left(\boldsymbol{m}_0, \hat{t}, \mathcal{J}, \mathcal{K}\right)$ and other similar linear constraints.

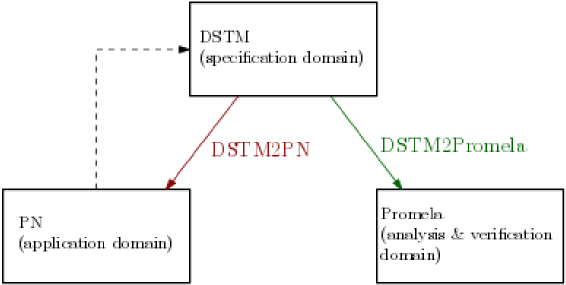
Numerical results

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

Figure: Results of the numerical experiments run to assess diagnosability of the fault $(t_6, \texttt{bf})$ 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

Proposed workflow



- 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

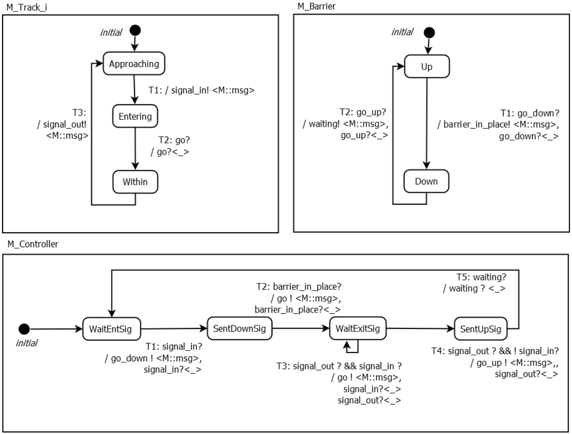


Figure: DSTM specification of the benchmark components.

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;
      :: (((!(TrackSignalIn?[_])) && (TrackSignalOut?[_]))) ->
            TrackSignalOut?_;
            GoUp!msg;
            state = controller_sentUpSignal;
      fi;
}
:: (state == controller_sentUpSignal && HasToken[_pid]==1) ->
      ...
}od unless {(chTerm?[1]); ... }
```

Figure: Promela Controller (excerpt).

Conclusions

- 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

Future developments

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?