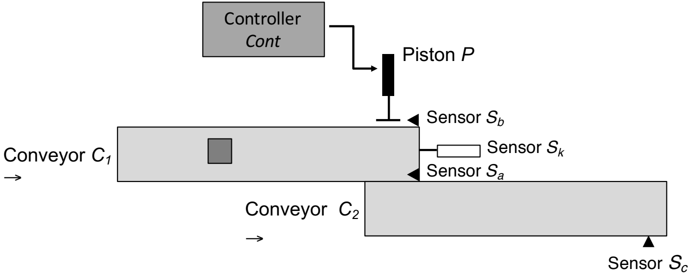
Diagnosing Discrete Event Systems Using Nominal Models Only

Yannick Pencolé[1], Gerald Steinbauer[2], Clemens Mühlbacher[2], Louise Travé-Massuyès[1]
[1]LAAS-CNRS, Toulouse, France
[2]Graz University of Technology, Graz, Austria

DX-17 25-29th September 2017

Introduction

Running example



A conveyor 1 receives a piece of luggage (left) and moves to the right. A sensor detects the presence of the luggage and a piston pushes it to a conveyor 2 that delivers it on the right. Empty conveyors move back to initial positions.

LAAS CNRS

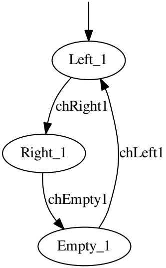
Basic assumptions

- Completeness of event types : a faulty component does not produce any other events than the ones already in the model,
- Stable structural model : the synchronization between components works always correctly,
- Event source is certain : any event e of a component is only generated by this component.

Model of a component

Definition: Model of a component

The model of a component c_i is an automaton $A_i = (Q_i, E_i, T_i, q_{0_i})$ where Q_i is a finite set of states, E_i is a set of events, the transition function $T_i : Q_i \times E_i \to Q_i$, and q_{0_i} is the initial state.



A conveyor belt :

- chRight1 : the belt starts moving from the left to right with an item
- chEmpty1 : the item is delivered
- chLeft1 : the belt starts moving from the right to left (no item)

Synchronization operator

Definition: Operator $\|_{\mathscr{R}}$

The synchronized product of $A_1, \ldots, A_n$ with respect to a set of synchronization rules $\mathscr{R}$ denoted by $A_1 \|_{\mathscr{R}} \cdots \|_{\mathscr{R}} A_n$ is defined as the automaton $\mathscr{A} = (Q_{\mathscr{A}}, E_{\mathscr{A}}, T_{\mathscr{A}}, q_{0_{\mathscr{A}}})$ with $Q_{\mathscr{A}} \subseteq Q_1 \times \cdots \times Q_n$, $E_{\mathscr{A}} = E_{\mathscr{R}}$, $q_{0_{\mathscr{A}}} = (q_{0_1}, \ldots, q_{0_n})$, and the transition function $T_{\mathscr{A}} : T_{\mathscr{A}}((q_1, \ldots, q_n), (e_1, \ldots, e_n)) = T_1^{\varepsilon}(q_1, e_1) \times \cdots \times T_n^{\varepsilon}(q_n, e_n)$ with $q_i \in Q_i$ and $e_i \in E_i \cup \{\varepsilon\}$ if all $T_i(q_i, e_i)$, $i = 1, \ldots, n$ are defined. $T_{\mathscr{A}}$ is undefined otherwise.

Property:

The synchronized product operator $\|_{\mathscr{R}}$ with respect to a set of synchronization rules $\mathscr{R}$ is commutative and associative.

LAAS CNRS

Synchronization rules of the running example

r_1	$< chRight1, sensK >$
r_2	$< \mathbf{k}, \mathbf{pistonOut}, moveOut >$
r_3	$< endOut, sensA, chEmpty1, chLeft2 >$
r_4	$< \mathbf{a}, \mathbf{pistonIn}, moveIn >$
r_5	$< endIn, sensB >$
r_6	$< chRight2, sensC >$
r_7	$< \mathbf{c}, chEmpty2, chLeft1 >$
r_8	$< wcc, wcb >$

Example : synchronization rule r_2

- **k** : Sensor S_k detects the presence of a baggage
- **pistonOut** : Controller sends the command for the piston to move out
- *moveOut* : Piston receives the command to move out.

System

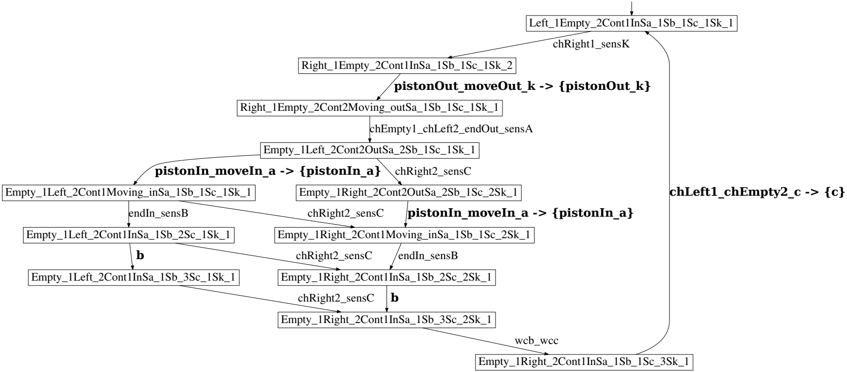
Definition: System

A system comprises :

1. a set of components $C = \{c_1, \ldots, c_n\}$,
2. a system description $SD = (\{A_1, \ldots, A_n\}, \mathscr{R})$ where the A_i's are the automata representing the normal behavior of the components c_i's, and $\mathscr{R}$ is a set of synchronization rules.

LAAS CNRS

Running example : global behaviour



Observation mask

Definition: Observation mask

The observation mask $obs : E_{\mathscr{R}} \rightarrow \prod_{i=1}^{n}(E_{O_i} \cup \{\varepsilon\})$ maps a synchronized event to a synchronized observable event or to $(\varepsilon, \ldots, \varepsilon)$:

$$obs((e_1, \ldots, e_n)) = (obs_1(e_1), \ldots, obs_n(e_n)).$$

LAAS CNRS

Consistency

Definition: [

Consistency] A system description SD is consistent with a sequence of observations OBS if $obs^{-1}(OBS) \cap \mathscr{L}(SD) \neq \emptyset$.

LAAS CNRS

Universal behaviour

Definition: Universal behavior

The universal behavior Ub_i of a component c_i is an automaton that represents the language of the Kleene closure of the component's events E_i.

Diagnosis

Definition: [

Diagnosis] A diagnosis for the diagnosis problem (SD, C, OBS) is a set $\Delta \subseteq C$ such that $\|\{SD \backslash \{A_{c_i} | c_i \in \Delta\} \cup \{Ub_{c_i} | c_i \in \Delta\}\}$ is consistent with OBS.

Definition: Minimal diagnosis

A diagnosis Δ is minimal if there is no strict subset $\Delta' \subset \Delta$ that is a diagnosis.

Conflicts/Minimal conflicts

Definition: Conflict

A conflict set is a set of components $\Gamma := \{c_1, \ldots, c_k\} \subseteq C$ such that $\|\{A_\Gamma \cup \{Ub_i | c_i \in C \backslash \Gamma\}\}$ is inconsistent with *OBS*.

Definition: A

conflict Γ is minimal if no subset $\Gamma' \subset \Gamma$ is a conflict.

LAAS CNRS

Minimal conflicts and algorithm

Property: Link between diagnosis and conflicts

Let *MCS* be the set of minimal conflicts, a diagnosis Δ is minimal iff Δ is a minimal hitting set of *MCS*.

Conclusions

Perspectives