A General Algorithm for Pattern Diagnosability of Distributed Discrete Event Systems

Lina YE, Philippe DAGUE
LRI, Univ. Paris-Sud and CNRS

August 2, 2012
Outline

1. Introduction
2. Recognition
3. Diagnosability
4. Implementation
5. Conclusion
Diagnosis and diagnosability

Model-based diagnosis

- Goal: detect possible faults to explain observations of the system
- System knowledge: system model including normal behavior and fault behavior
- Reasoning: deduce the exact fault that is consistent with system knowledge and observations

Diagnosability

- Perform on system model at the design stage
- Determine the effectiveness of diagnosis algorithm
- Help in designing diagnosis algorithm
Diagnosis and diagnosability

Model-based diagnosis

- Goal: detect possible faults to explain observations of the system
- System knowledge: system model including normal behavior and fault behavior
- Reasoning: deduce the exact fault that is consistent with system knowledge and observations

Diagnosability

- Perform on system model at the design stage
- Determine the effectiveness of diagnosis algorithm
- Help in designing diagnosis algorithm
System model

- System model: a Finite State Machine (FSM) $G = (Q, \Sigma, \delta, q^0)$
- $\Sigma = \Sigma_o \cup \Sigma_u$
- $L(G) = \{s \in \Sigma^* | \exists q \in Q, (q^0, s, q) \in \delta\}$
- $L(G)/s = \{t \in \Sigma^* | s.t \in L(G)\}$
- $P(s)$: the projection of the trajectory $s$ to observable events
- Assumption: there is no unobservable cycle in the system and language is live.

- A pattern is a deterministic complete FSM with stable final states set $F_\Omega$.
- Marked language: $L_m(\Omega) = \{s \in L(\Omega) | \exists q \in F_\Omega, (q^0_\Omega, s, q) \in \delta_\Omega\}$
- A trajectory $s$ of the system recognizes the pattern if $s \in L_m(\Omega)$
System model

- System model: a Finite State Machine (FSM) $G = (Q, \Sigma, \delta, q^0)$
- $\Sigma = \Sigma_o \cup \Sigma_u$
- $L(G) = \{ s \in \Sigma^* | \exists q \in Q \text{, } (q^0, s, q) \in \delta \}$
- $L(G)/s = \{ t \in \Sigma^* | s.t \in L(G) \}$
- $P(s)$: the projection of the trajectory $s$ to observable events
- Assumption: there is no unobservable cycle in the system and language is live.
- A pattern is a deterministic complete FSM with stable final states set $F_\Omega$.
- Marked language: $L_m(\Omega) = \{ s \in L(\Omega) \text{ | } \exists q \in F_\Omega \text{, } (q_0^\Omega, s, q) \in \delta_\Omega \}$
- A trajectory $s$ of the system recognizes the pattern if $s \in L_m(\Omega)$
Operations

Synchronization

Given two FSMs $G_1 = (Q_1, \Sigma_1, \delta_1, q_1^0)$ and $G_2 = (Q_2, \Sigma_2, \delta_2, q_2^0)$, their synchronization is

$$G_1 \parallel \Sigma G_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \parallel_2, (q_1^0, q_2^0)),$$

where $\Sigma_s = \Sigma_1 \cap \Sigma_2$ is the set of shared events, which can be omitted when there is no ambiguity in the context, and $\delta_1 \parallel_2$ is defined as follows:

- $((q_1, q_2), \sigma, (q_1', q_2')) \in \delta_1 \parallel_2$, if $\sigma \in \Sigma_s$, $(q_1, \sigma, q_1') \in \delta_1$ and $(q_2, \sigma, q_2') \in \delta_2$;
- $((q_1, q_2), \sigma, (q_1', q_2)) \in \delta_1 \parallel_2$, if $\sigma \in \Sigma_1 \setminus \Sigma_s$ and $(q_1, \sigma, q_1') \in \delta_1$;
- $((q_1, q_2), \sigma, (q_1, q_2')) \in \delta_1 \parallel_2$, if $\sigma \in \Sigma_2 \setminus \Sigma_s$ and $(q_2, \sigma, q_2') \in \delta_2$.

Product

Given two FSMs $G_1$ and $G_2$, their product is $G_1 \times G_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \times_2, (q_1^0, q_2^0))$, where

$$\delta_1 \times_2 ((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$$

if both $\delta_1(q_1, \sigma)$ and $\delta_2(q_2, \sigma)$ are defined in $G_1, G_2$ respectively. Otherwise, $\delta_1 \times_2 ((q_1, q_2), \sigma)$ is undefined in $G_1 \times G_2$. 
Operations

Examples

Synchronization

Product
Given a FSM $G (Q, \Sigma, \delta, q^0)$, its delay closure with respect to $\Sigma_d$, where $\Sigma_d \subseteq \Sigma$, is $\mathcal{C}_{\Sigma_d}(G) = (Q, \Sigma_d, \delta_d, q^0)$, where $\delta_d(q, \sigma) = q'$ with $\sigma \in \Sigma_d$ if $\exists s \in (\Sigma \setminus \Sigma_d)^*$, $\delta(q, s\sigma) = q'$ in $G$.

Example

Delay closure with respect to $\Sigma_d = \{O_1, O_2, O_3\}$

Delay closure with respect to $\Sigma_d = \{O_1, O_2\}$
Operations

Delay Closure

Given a FSM $G (Q, \Sigma, \delta, q^0)$, its delay closure with respect to $\Sigma_d$, where $\Sigma_d \subseteq \Sigma$, is $\mathcal{C}_{\Sigma_d}(G) = (Q, \Sigma_d, \delta_d, q^0)$, where $\delta_d(q, \sigma) = q'$ with $\sigma \in \Sigma_d$ if $\exists s \in (\Sigma \setminus \Sigma_d)^*$, $\delta(q, s\sigma) = q'$ in $G$.

Example

Delay closure with respect to $\Sigma_d = \{O_1, O_2, O_3\}$

Delay closure with respect to $\Sigma_d = \{O_1, O_2\}$
Operations

**Delay Closure**

Given a FSM $G (Q, \Sigma, \delta, q^0)$, its delay closure with respect to $\Sigma_d$, where $\Sigma_d \subseteq \Sigma$, is $\mathcal{C}_{\Sigma_d}(G) = (Q, \Sigma_d, \delta_d, q^0)$, where $\delta_d(q, \sigma) = q'$ with $\sigma \in \Sigma_d$ if $\exists s \in (\Sigma \setminus \Sigma_d)^*$, $\delta(q, s\sigma) = q'$ in $G$.

**Example**

$\Sigma_d = \{O1, O2, O3\}$

$\Sigma_d = \{O1, O2\}$
Definitions

Pattern diagnosability

A pattern $\Omega$ is diagnosable in a system $G$, iff

$\exists n \in N, \forall s \in L(G) \cap L_m(\Omega), \forall t \in L(G)/s, \text{if } |t| \geq n, \text{ then } P^{-1}P(s.t) \subseteq L_m(\Omega)$.

Critical pair

A critical pair $p, p'$ of system $G$ with respect to the pattern $\Omega$ should satisfy three conditions:

- $p \in L_m(\Omega)$ and $p' \notin L_m(\Omega)$;
- $p$ is of arbitrarily long length after pattern recognition;
- $P(p) = P(p')$.

Theorem

A pattern $\Omega$ is diagnosable in a system $G$ iff there is no critical pair in $G$. 
# Definitions

## Pattern diagnosability

A pattern $\Omega$ is diagnosable in a system $G$, iff
\[ \exists n \in N, \forall s \in L(G) \cap L_m(\Omega), \forall t \in L(G)/s, \text{ if } |t| \geq n, \text{ then } P^{-1} P(s,t) \subseteq L_m(\Omega). \]

## Critical pair

A critical pair $p, p'$ of system $G$ with respect to the pattern $\Omega$ should satisfy three conditions:
- $p \in L_m(\Omega)$ and $p' \notin L_m(\Omega)$;
- $p$ is of arbitrarily long length after pattern recognition;
- $P(p) = P(p')$.

## Theorem

A pattern $\Omega$ is diagnosable in a system $G$ iff there is no critical pair in $G$. 
Definitions

Pattern diagnosability

A pattern $\Omega$ is diagnosable in a system $G$, iff

$$\exists n \in N, \forall s \in L(G) \cap L_m(\Omega), \forall t \in L(G)/s, \text{ if } |t| \geq n, \text{ then } P^{-1}P(s.t) \subseteq L_m(\Omega).$$

Critical pair

A critical pair $p, p'$ of system $G$ with respect to the pattern $\Omega$ should satisfy three conditions:

- $p \in L_m(\Omega)$ and $p' \notin L_m(\Omega)$;
- $p$ is of arbitrarily long length after pattern recognition;
- $P(p) = P(p')$.

Theorem

A pattern $\Omega$ is diagnosable in a system $G$ iff there is no critical pair in $G$. 
Global pattern recognizer

Definition

Given a system model $G$ and a pattern $\Omega$, then the global pattern recognizer is $R_G = G \times \Omega$, where the final states of the recognizer are those containing a final state of the pattern, i.e., $F_{R_G} = (Q \times F_\Omega) \cap Q_{R_G}$.

Pattern

Global pattern recognizer

- The same observations from the blue and red trajectories: $\text{obs}=O_1.(O_2.O_3)^*$

The existence of critical pair violates pattern diagnosability property
**Global pattern recognizer**

**Definition**

Given a system model $G$ and a pattern $\Omega$, then the global pattern recognizer is

$$R_G = G \times \Omega,$$

where the final states of the recognizer are those containing a final state of the pattern, i.e.,

$$F_{R_G} = (Q \times F_\Omega) \cap Q_{R_G}.$$

- The same observations from the blue and red trajectories: $\text{obs} = O_1.(O_2,O_3)^*$

The existence of critical pair violates pattern diagnosability property
Global pattern verifier

**Definition**

The global pattern verifier of the system $G$, denoted by $V$, is the FSM:

$$V = \mathcal{C}_{\Sigma_o}(R_G) \parallel \mathcal{C}_{\Sigma_o}(R_G),$$

where $R_G$ is the global pattern recognizer of the system $G$.

- ambiguous state: only one final state of recognizer in the pattern verifier
- critical path: ambiguous state cycle with at least one observable event
- critical path $\iff$ critical pair
- pattern diagnosable $\iff$ non existence of critical path

**Refined pattern recognizer**

**Part of global pattern verifier**
**Global pattern verifier**

**Definition**

The global pattern verifier of the system $G$, denoted by $V$, is the FSM:

$$V = \left(\mathcal{C}_{\Sigma_o}(R_G) \parallel \Sigma_o \mathcal{C}_{\Sigma_o}(R_G)\right)$$

where $R_G$ is the global pattern recognizer of the system $G$.

- ambiguous state: only one final state of recognizer in the pattern verifier
- critical path: ambiguous state cycle with at least one observable event
- critical path $\iff$ critical pair
- pattern diagnosable $\iff$ non existence of critical path
Outline

1 Introduction

2 Recognition

3 Diagnosability

4 Implementation

5 Conclusion
Distributed system model

Notation

- System model: a set of Finite State Machines (FSM) $G_i = (Q_i, \Sigma_i, \delta_i, q_0^i)$, $\Sigma_i = \Sigma_{i_o} \cup \Sigma_{i_u} \cup \Sigma_{i_c}$
- Assumption: for each component, local language is live and observable live.
- Significant event: events changing pattern state.

Components $G_1, G_2, G_3$

Pattern
Distributed system model

Notation

- System model: a set of Finite State Machines (FSM) \( G_i = (Q_i, \Sigma_i, \delta_i, q^0_i) \), \( \Sigma_i = \Sigma_{io} \cup \Sigma_{iu} \cup \Sigma_{ic} \)
- Assumption: for each component, local language is live and observable live.
- Significant event: events changing pattern state.

Components \( G_1, G_2, G_3 \)

Pattern
Local pattern recognizer

Given a subsystem $G_S$ and a pattern $\Omega$, then the pattern recognizer of $G_S$ is $R_{G_S} = G_S \times \Omega$, where the final states of the recognizer are those whose pattern state is a final state of the pattern.
Pattern recognition

Local pattern recognizer

Given a subsystem $G_S$ and a pattern $\Omega$, then the pattern recognizer of $G_S$ is $R_{G_S} = G_S \times \Omega$, where the final states of the recognizer are those whose pattern state is a final state of the pattern.
Incremental recognition

**Diagnosability relative paths**

- In a pattern recognizer of $G_S$, if $\exists q_r = (q, q_\Omega)$ such that $q_\Omega$ is either a final state or in the pattern, it is a source state of a significant event $\sigma$ that is outside of $G_S$, then the corresponding paths are recognition relative paths and such a significant event is next recognizable event.

- All recognition relative paths and all paths with the same observations as a recognition relative path are called diagnosability relative paths.

- Subsystem extension: synchronize reduced recognizer, i.e., only diagnosability relative paths of recognizer, with next component, which contains at least one next recognizable event.

**Recognizer reduction and next component**

```plaintext
X0P0 ----> X3P0 ----> X4P0 ----> O1
^       |
|       v
C2 ----> X3P0 ----> X4P0 ----> O1

X1P1 ----> X3P1 ----> X4P0 ----> O1
^       |
|       v
C1 ----> X3P1 ----> X4P0 ----> O1

^       v
U1 ----> O1

^       v
U2 ----> O1

^       v
U2 ----> O1
```

Part of extended subsystem
Incremental recognition

Diagnosability relative paths

- In a pattern recognizer of $G_S$, if $\exists q_r = (q, q_{\Omega})$ such that $q_{\Omega}$ is either a final state or in the pattern, it is a source state of a significant event $\sigma$ that is outside of $G_S$, then the corresponding paths are recognition relative paths and such a significant event is next recognizable event.

- All recognition relative paths and all paths with the same observations as a recognition relative path are called diagnosability relative paths.

- Subsystem extension: synchronize reduced recognizer, i.e., only diagnosability relative paths of recognizer, with next component, which contains at least one next recognizable event.

Recognizer reduction and next component

Part of extended subsystem
Incremental recognition

Diagnosability relative paths

- In a pattern recognizer of $G_S$, if $\exists q_r = (q, q_{\Omega})$ such that $q_{\Omega}$ is either a final state or in the pattern, it is a source state of a significant event $\sigma$ that is outside of $G_S$, then the corresponding paths are recognition relative paths and such a significant event is next recognizable event.

- All recognition relative paths and all paths with the same observations as a recognition relative path are called diagnosability relative paths.

- Subsystem extension: synchronize reduced recognizer, i.e., only diagnosability relative paths of recognizer, with next component, which contains at least one next recognizable event.

Recognizer reduction and next component

Part of extended subsystem
Incremental recognition

Diagnosability relative paths

- In a pattern recognizer of $G_S$, if $\exists q_r = (q, q_\Omega)$ such that $q_\Omega$ is either a final state or in the pattern, it is a source state of a significant event $\sigma$ that is outside of $G_S$, then the corresponding paths are recognition relative paths and such a significant event is next recognizable event.
- All recognition relative paths and all paths with the same observations as a recognition relative path are called diagnosability relative paths.
- Subsystem extension: synchronize reduced recognizer, i.e., only diagnosability relative paths of recognizer, with next component, which contains at least one next recognizable event.

Recognizer reduction and next component

Part of extended subsystem
Incremental recognition

Diagnosability relative paths

- In a pattern recognizer of $G_S$, if $\exists q_r = (q, q_\Omega)$ such that $q_\Omega$ is either a final state or in the pattern, it is a source state of a significant event $\sigma$ that is outside of $G_S$, then the corresponding paths are recognition relative paths and such a significant event is next recognizable event.

- All recognition relative paths and all paths with the same observations as a recognition relative path are called diagnosability relative paths.

- Subsystem extension: synchronize reduced recognizer, i.e., only diagnosability relative paths of recognizer, with next component, which contains at least one next recognizable event.

Recognizer reduction and next component

Part of extended subsystem
Outline

1 Introduction
2 Recognition
3 Diagnosability
4 Implementation
5 Conclusion
From recognition to diagnosability

Regional pattern verifier

The regional pattern verifier is constructed from the complete recognizer as follows:
- keep the set of observable events and communication events
- modify the left instance by prefixing communication events by $L$
- modify the right instance by prefixing communication events by $R$
- synchronize the left one and the right one based on observable events
- partial critical path: paths in verifier containing ambiguous states

Part of the complete recognizer

Part of verifier (partial critical path)
Regional pattern verifier

The regional pattern verifier is constructed from the complete recognizer as follows:

- keep the set of observable events and communication events
- modify the left instance by prefixing communication events by $L$
- modify the right instance by prefixing communication events by $R$
- synchronize the left one and the right one based on observable events
- partial critical path: paths in verifier containing ambiguous states
From recognition to diagnosability

Regional pattern verifier

The regional pattern verifier is constructed from the complete recognizer as follows:
- keep the set of observable events and communication events
- modify the left instance by prefixing communication events by $L$
- modify the right instance by prefixing communication events by $R$
- synchronize the left one and the right one based on observable events
- partial critical path: paths in verifier containing ambiguous states

Part of the complete recognizer

Part of verifier (partial critical path)
From recognition to diagnosability

Regional pattern verifier

The regional pattern verifier is constructed from the complete recognizer as follows:

- keep the set of observable events and communication events
- modify the left instance by prefixing communication events by $L$
- modify the right instance by prefixing communication events by $R$
- synchronize the left one and the right one based on observable events
- partial critical path: paths in verifier containing ambiguous states

Part of the complete recognizer

Part of verifier (partial critical path)
From recognition to diagnosability

Regional pattern verifier

The regional pattern verifier is constructed from the complete recognizer as follows:

- keep the set of observable events and communication events
- modify the left instance by prefixing communication events by $L$
- modify the right instance by prefixing communication events by $R$
- synchronize the left one and the right one based on observable events
- partial critical path: paths in verifier containing ambiguous states

Part of the complete recognizer

Part of verifier (partial critical path)
Abstracted pattern verifier

The abstracted pattern verifier (APV) from a pattern verifier is obtained by the following steps:

1. Keep only the communication information.
2. Recuperate qualitatively ambiguous cycles with only observable events (lost in step 1).
3. Recuperate qualitatively the ambiguous states before or after communication events lost in step 1.

From pattern verifier to abstracted pattern verifier
Local twin checker

**Definition**

The local twin checker is obtained from the local model by the following steps:

1. refine local model by keeping observable and communication information;
2. modify left and right instances with prefix $L$ and $R$;
3. synchronize the two instances based on the observable events.

**Component $G_3$**

```
Component $G_3$

$Z0$ --- $Z1$ --- $Z2$

$C1$ --- $O5$ --- $O6$
```

**Part of local twin checker of $G_3$**

```
Part of local twin checker of $G_3$

$Z0$ --- $Z1$ --- $Z2$

$L:C1$ --- $O5$ --- $O6$
```

```
Local twin checker

Definition

The local twin checker is obtained from the local model by the following steps:

1. refine local model by keeping observable and communication information;
2. modify left and right instances with prefix $L$ and $R$;
3. synchronize the two instances based on the observable events.
Abstracted local twin checker - ALTC

**Definition**

The abstracted local twin checker from a given local twin checker is obtained by the following steps:

1. only keep the communication information;
2. recuperate qualitatively all cycles lost in the previous step.
Abstracted local twin checker - ALTC

Definition

The abstracted local twin checker from a given local twin checker is obtained by the following steps:

1. only keep the communication information;
2. recuperate qualitatively all cycles lost in the previous step.
Global consistency checking

Global consistency

A partial critical path, i.e., a path containing an ambiguous state, is globally consistent if after synchronizing with the ALTCs of all connected components, it satisfies one of the following conditions:

- it contains an ambiguous state cycle;
- it contains an ambiguous state and there exists at least one non connected component in the system;

Lemma

A globally consistent partial critical path corresponds to a global critical path.

Theorem

A pattern is diagnosable in the system iff there is no globally consistent partial critical path.
Global consistency checking

Global consistency

A partial critical path, i.e., a path containing an ambiguous state, is globally consistent if after synchronizing with the ALTCs of all connected components, it satisfies one of the following conditions:

- it contains an ambiguous state cycle;
- it contains an ambiguous state and there exists at least one non-connected component in the system;

Lemma

A globally consistent partial critical path corresponds to a global critical path.

Theorem

A pattern is diagnosable in the system iff there is no globally consistent partial critical path.
Global consistency checking

**Global consistency**

A partial critical path, i.e., a path containing an ambiguous state, is globally consistent if after synchronizing with the ALTCs of all connected components, it satisfies one of the following conditions:

- it contains an ambiguous state cycle;
- it contains an ambiguous state and there exists at least one non connected component in the system;

**Lemma**

A globally consistent partial critical path corresponds to a global critical path.

**Theorem**

A pattern is diagnosable in the system iff there is no globally consistent partial critical path.
Search space reduction

- Incremental pattern recognition normally smaller than global recognition (global model)
- Abstracted pattern verifier smaller than global pattern verifier in most of the cases
- During global consistency checking, ALTC is used instead of the local twin checker
Test case

System

Pattern
Test case

Search space results

- **Number of states**
  - Pattern recognition
  - Pattern diagnosability
  - Algorithm procedure
  - Distributed algorithm
  - Centralized algorithm

- **Number of transitions**
  - Pattern recognition
  - Pattern diagnosability
  - Algorithm procedure
  - Distributed algorithm
  - Centralized algorithm
Test case

Search space results

- Global PR
- Global PV
- Distributed PR
- Distributed PV

Graphs showing:
- States number vs. Components number
- Transitions number vs. Components number
<table>
<thead>
<tr>
<th></th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Recognition</td>
</tr>
<tr>
<td>3</td>
<td>Diagnosability</td>
</tr>
<tr>
<td>4</td>
<td>Implementation</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Related works

- Sampath et al. (1995): deterministic diagnoser method, exponential complexity
- Jiang et al. (2001), Yoo et al. (2002), etc.: twin plant method (Verifier), polynomial complexity but in a centralized way
- Pencolé (2004), Constant et al. (2006), Schumann et al. (2007): distributed way (synchronization of local objects)
- Jéron et al. (2006): extension from fault event case to pattern case but in a centralized way

For event fault

- Event fault case is a special one of pattern fault case.
- Compared to Pencolé (2004), our abstracted distributed approach can save up to 60 percentage on our examples, depending on what kind of systems.
## Conclusion

- Propose an optimized algorithm for distributed pattern diagnosability
- Incrementally recognize pattern by extending subsystem but avoid global model
- Construct abstracted pattern verifier to obtain original diagnosability information
- Check global consistency through synchronizing with connected ALTCs
- Applicable to single, multiple even infinite sequences of events.

## Future Work

- For a given pattern diagnosable distributed system, investigate how to reduce the observations but still make the system diagnosable.
- Extend pattern diagnosability to pattern predictability in a distributed way.
Conclusion

- Propose an optimized algorithm for distributed pattern diagnosability
- Incrementally recognize pattern by extending subsystem but avoid global model
- Construct abstracted pattern verifier to obtain original diagnosability information
- Check global consistency through synchronizing with connected ALTCs
- Applicable to single, multiple even infinite sequences of events.

Future work

- For a given pattern diagnosable distributed system, investigate how to reduce the observations but still make the system diagnosable.
- Extend pattern diagnosability to pattern predictability in a distributed way.
Thanks!