Modeling and Solving Diagnosis of Discrete Event Systems via Satisfiability

Alban Grastien, Anbulagan, Jussi Rintanen, Elena Kelareva

DX 07 – 30 May 2007
Outline

1. Diagnosis
2. Reduction to a SAT Problem
3. Experiments
4. Discussion
Example of system to diagnose

Topology of the system

Model of a component
(Alarms: IAmBack and IReboot)
Formal Definition of Diagnosis (1/2)

Definition (Symbolic model of the system)

\[ \langle A, \Sigma_u, \Sigma_o, \delta, s_0 \rangle \]
- \( A \): set of Boolean variables
- \( \Sigma_u \) and \( \Sigma_o \): sets of (un)observable events
- \( \delta (e) \): set of rules (pairs precondition – effect)
- \( s_0 \): initial valuation of the variables

Definition (Observations)

Set of timed observable events
Example: \( e_1 \) occurred at the 5th transition
Definition (Faults)
Set of faulty events: $\Sigma_f \subseteq \Sigma_u$

Definition (Diagnosis)
Answer the following question:
Does there exist a trajectory on the model consistent with the observations that contain no faulty event?
Outline

1 Diagnosis
2 Reduction to a SAT Problem
3 Experiments
4 Discussion
### Definition (SAT)

**Given**
- a set of propositional variables $V$
- a formula $\varphi$ built using $V$ and the Boolean operators $\neg$, $\lor$

Does there exist a valuation $V \rightarrow \{0, 1\}$ so that $\varphi$ is true?

### Complexity

NP-complete but the structure makes real-world problems simpler

### Applications

Bounded Model-Checking, Diagnosis of Static Systems, Planning, Scheduling, etc.
Propositional variables

Hypothesis
The number of transitions $n$ is known

Propositional variables
- $a^t$ for $a \in A$ and $t \in \{0, \ldots, n\}$
- $e^t$ for $e \in \Sigma_u \cup \Sigma_o$ and $t \in \{0, \ldots, n-1\}$
- $\omega^t$ for $\omega \in \delta(e)$ and $t \in \{0, \ldots, n-1\}$
### Translation Rules

#### Encoding the Model $\Phi_{SD}$

- $\omega^t \rightarrow \phi^t \quad \text{(where } \omega = (\phi, c))$
- $\omega^t \rightarrow I^{t+1} \quad \text{(where } \omega = (\phi, c) \text{ and } I \in c)$
- $(\nu^t \land \neg \nu^{t+1}) \rightarrow (\omega_1^t \lor \cdots \lor \omega_k^t)$
- $\neg (\omega_1^t \land \omega_2^t) \quad \text{(if } \omega_1 \text{ and } \omega_2 \text{ interfere)}$
- $(\lor_{\omega \in \delta(e)} \omega^t) \leftrightarrow e^t$
- $\nu^0 \quad \text{(if } s_0(\nu) \text{ is } \text{true}) \quad \neg \quad \neg \nu^0 \quad \text{(if } s_0(\nu) \text{ is } \text{false})$

#### Encoding the Observations $\Phi_{OBS}$

- $e^t \quad \text{(if } e \in \Sigma_o \text{ and } (e, t) \in \text{OBS})$
- $\neg e^t \quad \text{(if } e \in \Sigma_o \text{ and } (e, t) \notin \text{OBS})$
The solutions of the SAT problem $\Phi_{SD} \land \Phi_{OBS}$ are the trajectories of length $n$ consistent with the observations.
Definition (Diagnosis)

Does there exist a trajectory on the model consistent with the observations that contain no faulty event?

Encoding the diagnosis problem $\Phi_\Delta$

- $\Phi_{SD}$
- $\Phi_{OBS}$
- $\neg e^t$ (if $e \in \Sigma_f$)

Diagnosis by SAT

Iff $\Phi_\Delta$ is satisfiable, a normal behaviour exists
Extensions

Observations

- Most of observation uncertainties can be easily encoded (total order, partial order, noise, etc.)
- Issue: determining the number of transitions $n$
  - an upper bound is sufficient
  - the number of transitions can be reduced (concurrency)

Faults

- When the CNF is satisfiable, a trajectory is computed
  - new clauses can be added to get another trajectory
- The number of faults can be computed
  - create clauses that are satisfiable if $k$ faults occurred, where $i \leq k \leq j$
- More complex definitions [Jéron et al. 06] can be encoded
Outline

1. Diagnosis
2. Reduction to a SAT Problem
3. Experiments
4. Discussion
The system to diagnose

Topology of the system

Model of a component
(Alarms: IAmBack and IReboot)

- Diagnoser approach fails
- Decentralised diagnosis fails
Parameters

- Randomly generated scenarios with $k$ faults from $k = 1$ to $k = 20$
- Different observabilities: timed observations, totally ordered observations, partially ordered observations
- Different answers: satisfiable or not
- Best SAT solvers based on DPLL procedure with CL enhancement (Siege_v4, zChaff, MINISAT1.13, MINISAT1.14, MINISAT2.0)
Trying to find a trajectory with $k$ faults
Trying to find a trajectory with $k - 1$ faults

<table>
<thead>
<tr>
<th>$k$</th>
<th>CNF properties</th>
<th>Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$#\text{var}$</td>
<td>$#\text{cls}$</td>
</tr>
<tr>
<td>1</td>
<td>10 440</td>
<td>43 300</td>
</tr>
<tr>
<td>2</td>
<td>22 999</td>
<td>94 977</td>
</tr>
<tr>
<td>3</td>
<td>36 118</td>
<td>149 931</td>
</tr>
<tr>
<td>4</td>
<td>51 077</td>
<td>213 683</td>
</tr>
<tr>
<td>5</td>
<td>64 516</td>
<td>271 913</td>
</tr>
<tr>
<td>6</td>
<td>79 665</td>
<td>338 711</td>
</tr>
<tr>
<td>7</td>
<td>94 974</td>
<td>407 267</td>
</tr>
<tr>
<td>8</td>
<td>108 573</td>
<td>469 491</td>
</tr>
<tr>
<td>9</td>
<td>126 072</td>
<td>549 653</td>
</tr>
<tr>
<td>10</td>
<td>143 371</td>
<td>630 653</td>
</tr>
<tr>
<td>11</td>
<td>155 090</td>
<td>688 191</td>
</tr>
<tr>
<td>12</td>
<td>170 659</td>
<td>763 837</td>
</tr>
<tr>
<td>13+</td>
<td>186 308</td>
<td>841 001</td>
</tr>
</tbody>
</table>
## Diagnosis with different observabilities

<table>
<thead>
<tr>
<th>$k$</th>
<th>Timed observations</th>
<th>Total order</th>
<th>Partial order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>2</td>
<td>0.07</td>
<td>0.08</td>
<td>0.09</td>
</tr>
<tr>
<td>3</td>
<td>0.16</td>
<td>0.26</td>
<td>0.14</td>
</tr>
<tr>
<td>4</td>
<td>0.36</td>
<td>0.3</td>
<td>3.6</td>
</tr>
<tr>
<td>5</td>
<td>2.6</td>
<td>1.9</td>
<td>9.7</td>
</tr>
<tr>
<td>6</td>
<td>3.5</td>
<td>16</td>
<td>5.6</td>
</tr>
<tr>
<td>7</td>
<td>4.8</td>
<td>9.1</td>
<td>350</td>
</tr>
<tr>
<td>8</td>
<td>4.1</td>
<td>24</td>
<td>290</td>
</tr>
<tr>
<td>9</td>
<td>4.3</td>
<td>61</td>
<td>63</td>
</tr>
<tr>
<td>10</td>
<td>4.2</td>
<td>25</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>11</td>
<td>2.1</td>
<td>3.9</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>12</td>
<td>2.7</td>
<td>6.3</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>13</td>
<td>4.8</td>
<td>8.7</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>14</td>
<td>4.3</td>
<td>16</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>15</td>
<td>3.7</td>
<td>18.7</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>16</td>
<td>11.8</td>
<td>9.6</td>
<td>&gt; 600</td>
</tr>
<tr>
<td>17</td>
<td>7.7</td>
<td>30</td>
<td>&gt; 600</td>
</tr>
</tbody>
</table>
Outline

1. Diagnosis
2. Reduction to a SAT Problem
3. Experiments
4. Discussion
Why SAT is Efficient

- SAT is good (and simple)
  - Problem simple to define (⇒ benchmarks, competitions, etc.)
  - Domain independent algorithm: used for planning or model-checking

- SAT for diagnosis
  - Look for only ONE trajectory
  - Dynamically choose the *best* variable (unit propagation, etc.)
  - Automatically decompose the problem into subproblems
  - Learn
When (Not) to Use SAT

Do not use SAT if:
- You can precompile your model
- You have many diagnosis questions
- You want a state-estimation / all the answers
- You have too many observations (→ incremental diagnosis)
- You cannot estimate the number of time steps

Use SAT if:
- You have no choice (large system, uncertainties)
- The answer you expect is yes or no
- You need justification
- You expect new improvements from SAT community
# Conclusion & Prospects

## Conclusion
- Diagnosis of Discrete Event Systems by SAT is an interesting option
- **Main issues**
  - Requires the number of time steps
  - Computes only one trajectory

## Prospects
- Build diagnosis algorithms based on SAT algorithms
- Incremental diagnosis by SAT
- Diagnosis specific SAT heuristics