Web Services and Diagnosis (1)

- **Web Service:**
  - web-based application that provides a service to a user
  - the user can be a human, a client application, or another Web Service
  - complex WSs provide a service by integrating and composing the activities of other (basic) WSs
  - Some examples:
    - plane ticket reservation → online travel planner
    - computer monitor seller → computer configurator
  - Web Services also used to model internal business processes of a company
Web Services and Diagnosis (2)

- We do **not** want to do debugging.
- **Runtime tracking** of an error source
  - failures mostly due to mishandled exceptions, lack of robustness, unpredicted behavior of one of the involved entities
  - quality of service failures (not tackled in this work)
- Final goal: **recovery**
  - find a way to provide the service in spite of the error
  - try to keep as low as possible the overhead for the user
- Current practice:
  - direct symptom handling (only for some error types)
  - no attempt at identifying causes
  - mostly: “unable to provide service, try again”

What do we diagnose?

- We diagnose a conversation.
  - A complex service results from the interaction between multiple basic service \(\rightarrow\) conversation
  - A conversation is a set of partially ordered activities carried out by different (basic) services.
    - internal activities
    - communications between services
- Component-oriented qualitative models:
  - component \(\leftrightarrow\) activity
  - system structure \(\leftrightarrow\) data i/o between activities
  - fault \(\leftrightarrow\) activity error
Motivating example: online book sales
Model-Based Diagnosis of WSs

- **Model** of an activity expresses:
  - how errors on input data affect the correctness of output data (ok mode)
  - how an error in activity execution can affect the correctness of output data (abnormal mode)
- pure deviation models:
  - a variable for each i/o data piece with domain \{ok, ab\}

- **Observations**:
  - alarms raised by a service
  - diagnosers receive and log messages between services
  - the model maps alarms and checks on logged messages to hypotheses on data correctness

A complete static model does not exist
A complete static model does not exist

1. private models
2. runtime composition

Decentralized Diagnosis
Decentralized Diagnosis

Web Service sends messages to local diagnoser

Decentralized Diagnosis

Local Diagnoser
local model + alarms + checkpoints

Web Service sends messages to local diagnoser
Decentralized Diagnosis

Global Diagnoser
no initial info

Local Diagnoser
local model + alarms + checkpoints

Web Service
sends messages to
local diagnoser

Our Approach

• **Consistency-based diagnosis** (with fault modes).
  
  • We do provide:
    • a **specification** of local diagnoser operations
    • a **formal characterization** of local diagnoser operations
    • an **algorithm** for the Global Diagnoser
      • starts with no information on local services
      • the algorithm only assumes that local diagnosers meet the specifications of their operations
      • the algorithm **merges** information from local diagnosers and **decides** which local diagnosers to contact.

  • We do not provide:
    • **algorithms** for local diagnosers.
**Starting Diagnosis Upon Alarms**

Something’s wrong corresponding local diagnoser reacts to a fault message.

**Initial info:**
- local **observations** (alarms + checkpoints) **OBS**

**Compute:**
- a set of **candidate diagnoses** → hypotheses of misbehaviour that explain **OBS**
  - **internal misbehaviour**: errors occurred inside the WS
  - **external misbehaviour**: errors in inputs received from other WSs (**blame on other services**)
- **consequences** of each hypothesis on service outputs
  - can be used to validate/discard a candidate diagnosis
**Local Candidate Diagnosis**

A local candidate diagnosis contains three elements:

- hypotheses on local behaviour
- blames on other (input) services
- consequences of hypotheses on other (output) services

**The Role of the Global Diagnoser**

**COLLECT**
local candidate diagnoses
The Role of the Global Diagnoser

QUESTION
ask for blame
explanation

VALIDATE
ask for consequence
validation
**Local Diagnosers - Explanation**

- Local diagnoser receives **blames**
- It produces local candidate diagnoses that explain observations **AND** blames.
  - additional hypotheses of internal misbehaviour
  - additional blames
  - additional consequences
- New local candidate diagnoses:
  - merged with the ones that originated the blame **by the global diagnoser**
- If no explanation:
  - the candidate diagnosis that originated the blame is **rejected by the global diagnoser**

**Local Diagnosers - Validation**

- Local diagnoser receives **consequences**
- It verifies through local observations whether the predicted consequences hold.
- Produces:
  - additional consequences on other services
- If initial consequences **hold**:
  - the global diagnoser **adds new consequences** to the local candidate diagnosis that originated them.
- If initial consequences **do not hold**:
  - the candidate diagnosis that originated them blame is **rejected by the global diagnoser**.
Characterization of Local Diagnosers (I)

- Candidate diagnoses are represented by **partial assignments** to model variables
  - assignment of *ok* or *ab* value to variables representing internal activities
  - assignment of *ok* or *ab* value to model variables
- For both **explanation/validation**:
  - local diagnosers receive the parts of the assignments that concerns them
  - work by **extending** partial assignments
- Both can be characterized in the same way
  - **EXTEND** operation **explains** and **validates** at the same time.

---

The EXTEND operation (I)

**Def.** An assignment $\alpha$ is **admissible** in a model $M_i$ if
i. $\alpha$ is **consistent** with $M_i$
ii. the **restriction** of $M_i \cup \alpha$ to variable **not assigned** in $\alpha$ is **equivalent** to the restriction of $M_i$ alone to the same variables.
The EXTEND operation (I)

**Def.** An assignment $\alpha$ is **admissible** in a model $M_i$ if

i. $\alpha$ is **consistent** with $M_i$

ii. the restriction of $M_i \cup \alpha$ to variable **not assigned** in $\alpha$ is **equivalent** to the restriction of $M_i$ alone to the same variables.

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>ok</td>
<td>ok</td>
<td>ok</td>
</tr>
<tr>
<td>ok</td>
<td>ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
</tr>
<tr>
<td>ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
</tr>
</tbody>
</table>

---

The EXTEND operation (I)

**Def.** An assignment $\alpha$ is **admissible** in a model $M_i$ if

i. $\alpha$ is **consistent** with $M_i$

ii. the restriction of $M_i \cup \alpha$ to variable **not assigned** in $\alpha$ is **equivalent** to the restriction of $M_i$ alone to the same variables.

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>ok</td>
<td>ok</td>
<td>ok</td>
</tr>
<tr>
<td>ok</td>
<td>ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
</tr>
<tr>
<td>ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>*</td>
<td>*</td>
<td>ab</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>ok</td>
<td>*</td>
<td>ab</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>ab</td>
<td>*</td>
<td>ab</td>
</tr>
</tbody>
</table>
The EXTEND operation (I)

Def. An assignment \( \alpha \) is admissible in a model \( M_i \) if

i. \( \alpha \) is consistent with \( M_i \)

ii. the restriction of \( M_i \cup \alpha \) to variable not assigned in \( \alpha \) is equivalent to the restriction of \( M_i \) alone to the same variables.

\[
\begin{array}{c|c|c|c|c}
    a & x & y1 & y2 \\
    \hline
    & ok & ok & ok & ok \\
    ok & ab & ok,ab & ok,ab & \\
    ab & ok,ab & ok,ab & ok,ab & \\
\end{array}
\]

\( M_i \)

\[
\begin{array}{c|c|c|c|c}
    a & x & y1 & y2 \\
    \hline
    \alpha_1 & & * & * & ab \\
    \alpha_2 & ok & ok & * & ab \\
    \alpha_3 & ab & * & * & ab \\
\end{array}
\]
**The EXTEND operation (I)**

**Def.** An assignment $\alpha$ is **admissible** in a model $M_i$ if

i. $\alpha$ is **consistent** with $M_i$

ii. the **restriction** of $M_i \cup \alpha$ to variable **not assigned** in $\alpha$ is **equivalent** to the restriction of $M_i$ alone to the same variables.

<table>
<thead>
<tr>
<th>a</th>
<th>x</th>
<th>y1</th>
<th>y2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>ok</td>
<td>ok</td>
<td>ok</td>
</tr>
<tr>
<td>ok</td>
<td>ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
</tr>
<tr>
<td>ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
<td>ok,ab</td>
</tr>
</tbody>
</table>

- $M_i$

**The EXTEND operation (II)**

**Def.** Given an assignment $\alpha$ and observations $\omega$, **EXTEND** computes **all minimal admissible extensions** of $\alpha \cup \omega$
The EXTEND operation (II)

Def. Given an assignment \( \alpha \) and observations \( \omega \), \textit{EXTEND} computes all minimal admissible extensions of \( \alpha \cup \omega \).

\[
\begin{array}{c|c|c|c}
\text{x} & \text{y1} & \text{y2} \\
\hline
\text{ok} & \text{ok} & \text{ok} \\
\text{ok} & \text{ab} & \text{ok},\text{ab} \\
\text{ab} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{a} & \text{x} & \text{y1} & \text{y2} \\
\hline
\text{ok} & \text{ok} & \text{ok} & \text{ok} \\
\text{ok} & \text{ab} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\text{ab} & \text{ok},\text{ab} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{a} & \text{x} & \text{y1} & \text{y2} \\
\hline
\text{ab} & \text{ok},\text{ab} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{a} & \text{x} & \text{y1} & \text{y2} \\
\hline
\text{ab} & \text{ok} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{a} & \text{x} & \text{y1} & \text{y2} \\
\hline
\text{ab} & \text{ok} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\text{a} & \text{x} & \text{y1} & \text{y2} \\
\hline
\text{ab} & \text{ok} & \text{ok},\text{ab} & \text{ok},\text{ab} \\
\end{array}
\]
The EXTEND operation (II)

**Def.** Given an assignment $\alpha$ and observations $\omega$, EXTEND computes all minimal admissible extensions of $\alpha \cup \omega$

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>$\text{ok}$</td>
<td>$\text{ok}$</td>
<td>$\text{ok}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$\text{ab}$</td>
<td>$\text{ok,ab}$</td>
<td>$\text{ok,ab}$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>$\text{ab}$</td>
<td>$\text{ok,ab}$</td>
<td>$\text{ok,ab}$</td>
</tr>
</tbody>
</table>

$\mathbf{M_i}$

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>$*$</td>
<td>$*$</td>
<td>$\text{ab}$</td>
</tr>
</tbody>
</table>

Extensions

The Global Diagnoser

- **EXTEND** defined as set of minimal admissible extensions:
  - allows to avoid unnecessary invocations
  - $*$ represents variables unworthy of further investigations
- The **global diagnoser**:
  - repeatedly invokes EXTEND on local diagnosers
  - A local diagnoser is invoked if:
    - another diagnoser assigned an $\text{ab}$ value to one of its outputs (blame to explain)
    - another diagnoser assigned an $\text{ok}$ or $\text{ab}$ value to one of its inputs (consequence to validate)
  - until there is nothing to explain/validate
  - EXTEND may produce new blames and consequences, but may also reject an assignment
- for a final assignment $\alpha$:
  - diagnosis $D(\alpha) = \{ a \mid a \text{ is an activity and } \alpha(a) = \text{ab} \}$
Conclusions and Future Work

• **Advantages** of the approach:
  • reduction of communication overhead
    • decentralized VS purely distributed
    • does not explore the whole model if not necessary
  • possible to apply it also to other types of systems
    • as long as models are pure deviation models
  • abstract models of correctness propagation
    • could be at least partially derived automatically (to investigate)

• **Future** work:
  • exploit coordination mechanisms and coordination info
  • local diagnosers only characterized
    • propose efficient algorithms for local diagnosers