Automatic Fault Tree Derivation from Little-JIL Process Definitions

Bin Chen, George S. Avrunin, Lori A. Clarke, and Leon J. Osterweil

Laboratory for Advanced Software Engineering Research (LASER)
University of Massachusetts Amherst

April, 2006
Accident and Hazard

• **Accident**
  - Undesired or unplanned event that results in harm to people or damage to property

• **Hazard**
  - A state of a system that, together with certain conditions in the environment, will lead inevitably to an accident

  e.g.

  Hazard: Incorrect type of blood is transfused

  \( \land \) **Condition**: Patient’s blood type is incompatible with the blood transfused

  \( \rightarrow \) **Accident**: Patient has transfusion reaction
Hazard Analysis

• Types of hazard analysis
  - Identify hazards and evaluate their effects (e.g. Failure Mode and Effects Analysis)
  - Identify and evaluate the causal factors of hazards (e.g. Fault Tree Analysis)
Hazard Analysis

• Types of hazard analysis
  - Identify hazards and evaluate their effects (e.g. Failure Mode and Effects Analysis)
  - Identify and evaluate the causal factors of hazards (e.g. Fault Tree Analysis)
Fault Tree Analysis (FTA)

• A well accepted and widely practiced hazard analysis technique

• Systematically identifies and reasons about all possible events that could lead to a given hazard by
  - Developing fault trees
  - Analyzing fault trees

• Analysis results can be used to improve the system
Elements of Fault Trees

**Basic Event:** basic initiating faults or conditions

**Undeveloped Event:** events that are not developed any further, either because necessary information for deriving the fault tree leading to these events is unavailable or because these events are considered to have insignificant consequence

**Intermediate Event:** events that need to be developed

**AND Gate:** the output event occurs if all of the input events occur

**OR Gate:** the output event occurs if any of the input events occur
A fault tree captures all the parallel and sequential combinations of events that could lead to a given hazard.
FTA for Process Improvement

- Apply FTA to improve medical safety processes
- Case study: The real-world blood transfusion process used at Baystate Medical Center, Massachusetts
The Problem

• Difficult to manually develop correct fault trees for large processes
  - Requires deep understanding of processes
  - Time-consuming
  - Error-prone
    • Errors in fault trees affect the validity of decisions made to improve processes
Our Approach

• Developed an algorithm that automatically develops fault trees from process definitions
  - The algorithm requires the process to be formal defined
  - The idea can be applied to the other process definition languages

• Extended existing FTA with visualization

• Evaluated our approach by applying it to the real-world blood transfusion process
Roadmap of Presentation

• Running example
• Automatically deriving fault trees from process definitions
• Fault Tree Analysis for process improvement
• Case Study
• Related work
• Conclusions
• Future work
Simplified Blood Transfusion Process

- Obtain Patient's Blood Type
- Contact Lab for Patient's Blood Type
- Test Patient's Blood Type
- Pick up Blood from Blood Bank
  - Handle Exception: Patient's Blood Type Unavailable
- Perform Transfusion
Simplified Blood Transfusion Process

Steps

- Obtain Patient's Blood Type
- Pick up Blood from Blood Bank
- Perform Transfusion
- Contact Lab for Patient's Blood Type
- Test Patient's Blood Type

Handle Exception:
Patient's Blood Type Unavailable

Step: represents some specified task to be done
Simplified Blood Transfusion Process

Sub-step sequencing: defines the order in which a step's sub-steps execute
Simplified Blood Transfusion Process

**Sequential Step**

- Obtain Patient's Blood Type
- Contact Lab for Patient's Blood Type
- Test Patient's Blood Type
- Pick up Blood from Blood Bank
- Handle Exception: Patient's Blood Type Unavailable
- Perform Transfusion
Simplified Blood Transfusion Process

Try Step

- Obtain Patient's Blood Type
- Contact Lab for Patient's Blood Type
- Test Patient's Blood Type

- Pick up Blood from Blood Bank
- Perform Transfusion
- Handle Exception: Patient's Blood Type Unavailable
Simplified Blood Transfusion Process

Exception: Patient's Blood Type Unavailable

Exception: indicates that execution of a step fails
Simplified Blood Transfusion Process

Exception handler: handles exception
Simplified Blood Transfusion Process

Control-flow badge: indicates how the step catching the exception executes after the handler finishes
Simplified Blood Transfusion Process

Artifacts: entities that are used or produced in processes
Example Hazard in Simplified Blood Transfusion Process

Hazard: Blood Unit passed to Perform Transfusion is wrong
Automatically Deriving FTs from Little-JIL Process Definitions

• 17 templates are defined based on Little-JIL semantics
  - Each can be used to develop certain kind of intermediate events

• A template
  - specifies events related to the intermediate event
  - defines the cause-consequence relationships between those events and the intermediate event
Artifact o from S is wrong

- S is a leaf step
- o is an OUT parameter of S
- i₁, ..., i_j are IN parameter of S
IN: Blood Type ↓  OUT: Blood Unit

Pick up Blood from Blood Bank

Blood Unit from “Pick up Blood from Blood Bank” is wrong

Blood Type to “Pick up Blood from Blood Bank” is wrong

Input Blood Type is correct, but “Pick up Blood from Blood Bank” produces wrong Blood Unit
Automatic Fault Tree Derivation Algorithm

• Define the given hazard as the TOP event and put it to the worklist

• While the worklist is not empty do
  – Remove an event e from the worklist
  – Apply the proper template according to e
  – For each new intermediate event e’, add e’ to the worklist
Example Hazard in Simplified Blood Transfusion Process

Hazard: Blood Unit passed to Perform Transfusion is wrong
Fault Tree for Simplified Blood Transfusion Process

E1: Blood Unit to “Perform Transfusion” is wrong

E2: Blood Unit from “Pick up Blood from Blood Bank” is wrong
Simplified Blood Transfusion Process

Blood unit from “Pick up Blood from Blood Bank” is wrong
Fault Tree for Simplified Blood Transfusion Process

E1: Blood Unit to “Perform Transfusion” is wrong
E2: Blood Unit from “Pick up Blood from Blood Bank” is wrong
E3: Blood Type to “Pick up Blood from Blood Bank” is wrong
E4: Input Blood Type is correct, but “Pick up Blood from Blood Bank” produces wrong Blood Unit
Simplified Blood Transfusion Process

Blood Type to "Pick up Blood from Blood Bank" is wrong
Fault Tree for Simplified Blood Transfusion Process

- **E1**: Blood Unit to “Perform Transfusion” is wrong
- **E2**: Blood Unit from “Pick up Blood from Blood Bank” is wrong
- **E3**: Blood Type to “Pick up Blood from Blood Bank” is wrong
- **E4**: Input Blood Type is correct, but “Pick up Blood from Blood Bank” produces wrong Blood Unit
- **E5**: Blood Type from “Contact Lab for Patient’s Blood Type” is wrong
- **E6**: Blood Type from “Test Patient’s Blood Type” is wrong

Blood Unit from “Pick up Blood from Blood Bank” is wrong.
Blood Type to “Pick up Blood from Blood Bank” is wrong.

Blood Type from “Contact Lab for Patient’s Blood Type” is wrong.
Blood Type from “Test Patient’s Blood Type” is wrong.
Simplified Blood Transfusion Process

- Contact Lab for Patient’s Blood Type
- Test Patient’s Blood Type
- Obtain Patient’s Blood Type
- Pick up Blood from Blood Bank
- Perform Transfusion

Artifact Flow

Blood Transfusion Process

Handle Exception: Patient’s Blood Type Unavailable

Blood Type from “Contact Lab for Patient’s Blood Type” is wrong
Fault Tree for Simplified Blood Transfusion Process

E1: Blood Unit to “Perform Transfusion” is wrong

E2: Blood Unit from “Pick up Blood from Blood Bank” is wrong

E3: Blood Type to “Pick up Blood from Blood Bank” is wrong

E5: Blood Type from “Contact Lab for Patient’s Blood Type” is wrong

E6: Blood Type from “Test Patient’s Blood Type” is wrong

E7: “Contact Lab for Patient’s Blood Type” produces wrong Blood Type

E8: Exception is not thrown by “Contact Lab for Patient’s Blood Type”

E4: Input Blood Type is correct, but “Pick up Blood from Blood Bank” produces wrong Blood Unit
Simplified Blood Transfusion Process

“Contact Lab for Patient’s Blood Type” produces wrong Blood Type
Fault Tree for Simplified Blood Transfusion Process

- E1: Blood Unit to “Perform Transfusion” is wrong
- E2: Blood Unit from “Pick up Blood from Blood Bank” is wrong
- E3: Blood Type to “Pick up Blood from Blood Bank” is wrong
- E4: Input Blood Type is correct, but “Pick up Blood from Blood Bank” produces wrong Blood Unit
- E5: Blood Type from “Contact Lab for Patient’s Blood Type” is wrong
- E6: Blood Type from “Test Patient’s Blood Type” is wrong
- E7: “Contact Lab for Patient’s Blood Type” produces wrong Blood Type
- E8: Exception is not thrown by “Contact Lab for Patient’s Blood Type”
- E11: Patient ID to “Blood Transfusion Process” is wrong
- E12: Input patient ID is correct, but “Contact Lab for Patient’s Blood Type” produces wrong Blood Type
Fault Tree for Simplified Blood Transfusion Process

E1
Blood Unit to "Perform Transfusion" is wrong

E2
Blood Unit from "Pick up Blood from Blood Bank" is wrong

E3
Blood Type to "Pick up Blood from Blood Bank" is wrong

E5
Blood Type from "Contact Lab for Patient 's Blood Type" is wrong

E7
"Contact Lab for Patient 's Blood Type" produces wrong Blood Type

E11
Patient ID to "Blood Transfusion Process" is wrong

E12
Input Patient ID is correct, but "Contact Lab for Patient 's Blood Type" produces wrong Blood Type

E8
Exception is not thrown by "Contact Lab for Patient 's Blood Type"

E9
"Test Patient 's Blood Type" produces wrong Blood Type

E10
Input Patient ID is correct, but "Test Patient 's Blood Type" produces wrong Blood Type

E13
Exception is thrown by "Contact Lab for Patient 's Blood Type"
Analyze Fault Trees - Calculate Minimal Cut Sets (MCSs)

- **Cut set** - a set of basic events and/or undeveloped events whose occurrence ensures that the TOP event occurs
- **MCS** - a cut set that cannot be further reduced
- **MCSs** can be automatically calculated from a fault tree using Boolean algebra
Calculate MCSs

Each gate corresponds to an equation
Each gate corresponds to an equation

1: \( E1 = E2 \)
Calculate MCSs

Each gate corresponds to an equation

1: \( E_1 = E_2 \)
2: \( E_2 = E_3 + E_4 \)
Each gate corresponds to an equation
1: E1 = E2  2: E2 = E3 + E4  3: E3 = E5 + E6  4: E5 = E7 \cdot E8  
5: E6 = E9 \cdot E13  6: E7 = E11 + E12  7: E9 = E11 + E10
Derive an equation for E1 by eliminating and substituting the other intermediate events:

\[
E1 = (E4) + (E11) + (E12 \cdot E8) + (E10 \cdot E13)
\]
Analyze FTs - Calculate MCSs (cont’d)

\[ E_1 = ( E_4 ) + ( E_{11} ) + ( E_{12} \cdot E_8 ) + ( E_{10} \cdot E_{13} ) \]

MCSs: \{ E_4 \} \{ E_{11} \} \{ E_{12}, E_8 \} \{ E_{10}, E_{13} \}
Analyze FTs - Calculate MCSs (cont’d)

$$E_1 = (E_4) + (E_{11}) + (E_{12} \cdot E_8) + (E_{10} \cdot E_{13})$$

MCSs: \{E_4\} \{E_{11}\} \{E_{12}, E_8\} \{E_{10}, E_{13}\}

Single points of failure !!!
Analyze FTs – Calculate MCSs (cont’d)

E1 = (E4) + (E11) + (E12 • E8) + (E10 • E13)

MCSs: {E4} {E11} {E12, E8} {E10, E13}

The hazard will definitely occur if

- “Pick up Blood from Blood Bank” produces wrong Blood Unit (E4)
- Patient ID to “Blood Transfusion Process” is wrong (E11)
Improve Simplified Blood Transfusion Process
On Alternative to Improve Simplified Blood Transfusion Process

Add a consistency check step
On Alternative to Improve Simplified Blood Transfusion Process

Add a consistency check step
New Fault Tree
New MCSs

{ E14, E4 }
{ E14, E11 }
{ E14, E12, E8 }
{ E14, E10, E13 }

No single point of failure now
New MCSs

\[
\begin{align*}
\{ & E_{14}, E_4 \} \\
\{ & E_{14}, E_{11} \} \\
\{ & E_{14}, E_{12}, E_8 \} \\
\{ & E_{14}, E_{10}, E_{13} \}
\end{align*}
\]

No single point of failure now

E14 appears in every MCS, which means “Blood Product Verification” is important for the new process. Thus further improvement could be focused on “Blood Product Verification”
Case Study: Real-World Blood Transfusion Process

• The real-world blood transfusion process used at Baystate Medical Center, Massachusetts
• The Little-JIL definition of this blood transfusion process contains 112 steps
Blood Transfusion Process

- Perform Transfusion
- Single-Unit Transfusion Process
- Deal With Failed Product
- Prepare Documentation for Blood Pick-up
- Single-Unit Transfusion Process
- Obtain Replacement Product
- Pick up Blood from Blood Bank
- Perform Transfusion
- Notify Blood Bank
- Return Blood

- Handle Failed Product Verification
Blood Transfusion Process

Single-Unit Transfusion Process

Bedside Checks
Verify Patient Identification
Product Verification
Gather Infusion Equipment
Administer Unit of Blood Product
Post Transfusion Work
Discard Transfusion Materials
Record Infusion Information

Suspected Transfusion Reaction
The Hazard

Blood Unit is Wrong!!!

Administer Unit of Blood Product is wrong
The Fault Tree Generated
Minimal Cut Sets

{ "Prepare Documentation for Blood Pick up" produces wrong Document, "Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit }

{ "Release Blood from Blood Bank" produces wrong Blood Unit, "Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit }

{ Patient’s Armband to “In Patient Blood Transfusion Process” is wrong, "Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit }

......
Display How a MCS Leads to the Hazard

• Support two ways to show how events in a MCS cause the hazard to occur
  - Display the sub-tree
  - Display an execution trace of the process
Display How a MCS Leads to the Hazard

- Support two ways to show how events in a MCS cause the hazard to occur
  - Display the sub-tree
  - Display an execution trace of the process
Example:
{
    "Prepare Documentation for Blood Pick up" produces wrong Document,
    "Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit
}
Trace for a MCS

In-Patient Blood Transfusion Process is started

Artifacts:

- Physician Order is right
- Patient Record is right
- Patient Armband is right
- Patient is right

In-Patient Blood Transfusion Process

Exception: No Patient Consent

Parameters

- Physician Order
- Patient Record
- Patient Armband
- Patient

Prepare Documentation for Blood Pick-up produces wrong Document

"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit
"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

<table>
<thead>
<tr>
<th>Artistats:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physician Order is right</td>
</tr>
<tr>
<td>Patient Record is right</td>
</tr>
<tr>
<td>Patient Armband is right</td>
</tr>
<tr>
<td>Patient is right</td>
</tr>
</tbody>
</table>

---

**Parameters**

- Physician Order
- Patient Record
- Patient Armband
- Patient

---

**State**

*Carry Out Physician Order for Transfusion* is started

- **Artifacts:**
  - Physician Order is right
  - Patient Record is right
  - Patient Armband is right
  - Patient is right
Check for Existence of Type and Screen is started

Artifacts:

- Physician Order is right
- Patient Record is right
- Patient Armband is right
- Patient is right

"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

Parameters

- Patient Record
- Patient Armband
- Patient
- Blood Type & Screen

In-Patient Blood Transfusion Process

Exception: No Patient Consent

Carry Out Physician Order for Transfusion

Prepare Documentation for Blood Pick-up
Pick up Blood from Blood Bank

Follow Through Check

Perform Transfusion
### Parameters

- Patient Record
- Patient Armband
- Patient
- Blood Type & Screen

### State

- **Check for Existence of Type and Screen** produces right **Blood Type & Screen**

**Artifacts:**
- Physician Order is right
- Patient Record is right
- Blood Type & Screen is right
- Patient Armband is right
- Patient is right
Prepare Document for Blood Pick-up is started

Artifacts:
- Physician Order is right
- Patient Record is right
- Blood Type & Screen is right
- Patient Armband is right
- Patient is right

"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

Exception: No Patient Consent

In-Patient Blood Transfusion Process

Carry Out Physician Order for Transfusion

Check for Existence of Type and Screen

Prepare Documentation for Blood Pick-up

Pick up Blood from Blood Bank

Follow Through Check

Perform Transfusion

State

Trace for a MCS
Trace for a MCS

MCS

"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

Parameters

- Patient Record
- Patient Armband
- Physician Order
- Blood Type & Screen
- Document

In-Patient Blood Transfusion Process

- Carry Out Physician Order for Transfusion
- Check for Existence of Type and Screen
- Prepare Documentation for Blood Pick-up
- Pick up Blood from Blood Bank
- Follow Through Check
- Perform Transfusion

Exception: No Patient Consent

State

Prepare Document for Blood Pick-up produces wrong Document

Artifacts:

- Physician Order is right
- Patient Record is right
- Blood Type & Screen is right
- Patient Armband is right
- Patient is right
- Document is wrong
Pick up Blood from Blood Bank is started

Artifacts:
- Physician Order is right
- Patient Record is right
- Document is wrong
- Patient Armband is right
- Patient is right

"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

MCS:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>In-Patient Blood Transfusion Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td></td>
</tr>
<tr>
<td>Blood Unit</td>
<td></td>
</tr>
</tbody>
</table>

Trace for a MCS
Trace for a MCS

"Prepare Documentation for Blood Pick up" produces wrong Document

"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

**Parameters**

- Document
- Blood Unit

**State**

*Pick up Blood from Blood Bank* is completed and produces wrong *Blood Unit*

Artifacts:

- *Physician Order* is right
- *Patient Record* is right
- *Document* is wrong
- *Patient Armband* is right
- *Patient* is right
- *Blood Unit* is wrong
Perform Transfusion is started

Artifacts:

- Physician Order is right
- Patient Record is right
- Blood Unit is wrong
- Patient Armband is right
- Patient is right
Trace for a MCS

MCS:

- "Prepare Documentation for Blood Pick up" produces wrong Document
- "Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

State:

- Single-Unit Transfusion Process is started
- Artifacts:
  - Physician Order is right
  - Patient Record is right
  - Blood Unit is wrong
  - Patient Armband is right
  - Patient is right

Parameters:

- Patient Record
- Patient Armband
- Patient
- Blood Unit
Bedside Checks is started
Artifacts:
- Physician Order is right
- Patient Record is right
- Blood Unit is wrong
- Patient Armband is right
- Patient is right

Trace for a MCS
Verify Patient Identification is started

Artifacts:
- Physician Order is right
- Patient Record is right
- Patient Armband is right
- Patient is right
- Blood Unit is wrong
Verify Patient Identification is completed

Artifacts:
- Physician Order is right
- Patient Record is right
- Blood Unit is wrong
- Patient Armband is right
- Patient is right
"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient Record</td>
</tr>
<tr>
<td>Patient Armband</td>
</tr>
<tr>
<td>Patient</td>
</tr>
<tr>
<td>Blood Unit</td>
</tr>
</tbody>
</table>

**MCS**

- Bedside Checks
- Gather Infusion Equipment
- Administer Unit of Blood Product
- Post Transfusion Work
- Discard Transfusion Materials
- Record Infusion Information

**Trace for a MCS**

**State**

- Product Verification is started
- Artifacts:
  - Physician Order is right
  - Patient Record is right
  - Patient Armband is right
  - Patient is right
  - Blood Unit is wrong
"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

Product Verification fails to throw exception FailProductCheck given the wrong Blood Unit

Artifacts:
- Physician Order is right
- Patient Record is right
- Blood Unit is wrong
- Patient Armband is right
- Patient is right
"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

State:
- Gather Infusion Equipment is started
- Artifacts:
  - Physician Order is right
  - Patient Record is right
  - Blood Unit is wrong
  - Patient Armband is right
  - Patient is right

Parameters:
- Blood Tube
- Blood Vial
- Catheter
- Labels

Trace for a MCS
Trace for a MCS

MCS

"Prepare Documentation for Blood Pick up" produces wrong Document
"Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

Parameters

| Blood Tube | Blood Vial | Catheter | Labels |

Single-Unit Transfusion Process

Bedside Checks

Gather Infusion Equipment

Administer Unit of Blood Product

Post Transfusion Work

Verify Patient Identification

Product Verification

Discard Transfusion Materials

Record Infusion Information

State

Gather Infusion Equipment is completed

Artifacts:

- Physician Order is right
- Patient Record is right
- Blood Unit is wrong
- Patient Armband is right
- Patient is right

…….
"Prepare Documentation for Blood Pick up" produces wrong Document "Product Verification" fails to throw exception FailedProductCheck given wrong Blood Unit

Parameters

- Patient
- Blood Unit
- Blood Tube
- Blood Vial
- Catheter
- Labels

Single-Unit Transfusion Process

Administer Unit of Blood Product

Suspected Transfusion Reaction

Bedside Checks

Gather Infusion Equipment

Blood Unit is Wrong!!

Verify Patient Identification
Product Verification

Discard Transfusion Materials
Record Infusion Information

State

- Administer Unit of Blood Product is started
- Artifacts:
  - Physician Order is right
  - Patient Record is right
  - Blood Unit is wrong
  - Patient Armband is right
  - Patient is right

......
Limitations

• The completeness and correctness of a derived fault tree depends on the completeness and correctness of the process definition.

• The derived fault tree could contain superfluous subtrees if leaf steps do not satisfy the assumption that any OUT parameter depends on all its IN parameters and resources.
Related Work

- Cha, Leveson, and Shimeall proposed a partially automated technique that derives fault trees from Ada programs based on templates.
- Ratan, Partridge, Reese, and Leveson proposed a fully automatic technique deriving fault trees from the Requirements State Machine Language (RSML) specifications.
- The approach by Pai and Dugan automatically derives fault trees from UML models.
- McKelvin, Eirea, Pinello, Kanajan, and Sangiovanni-Vincentelli designed an algorithm that derives fault trees from Fault Tolerant Data Flow (FTDF) models.
- Liggesmeyer, and Rothfelder use model checking to generate fault trees.
Conclusions

• Demonstrated that FTA can help process improvement
• Developed an automated fault tree derivation algorithm based upon process definitions
• Applied the algorithm to analyze a real-world process
Future Work

• Automatically provide recommendations for process improvement based on analysis results of FTA
• Combine FTA with Finite-State Verification
• Combine FTA with the other Hazard Analysis techniques (e.g. FMEA)
Thank You