

Considerations for Online Deviation Detection in Medical Processes

Stefan C. Christov George S. Avrunin Lori A. Clarke



SEHC 2013

Motivation—Medical Errors

- Medical errors are still a major concern
 - 1998 Institute of Medicine report
 - 2009 National Research Council report
- Our research focuses on helping to detect errors during the delivery of care and before harm is done

Motivation—Medical Errors

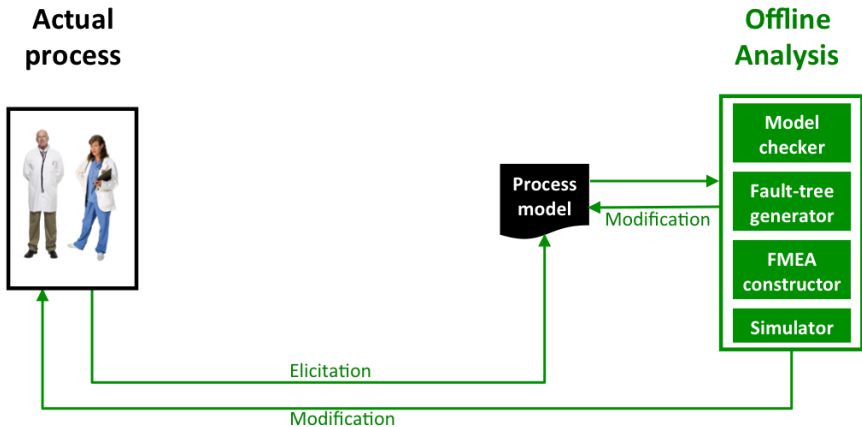
- Medical errors are still a major concern
 - 1998 Institute of Medicine report
 - 2009 National Research Council report
- Our research focuses on helping to detect errors during the delivery of care and before harm is done

Proposed approach

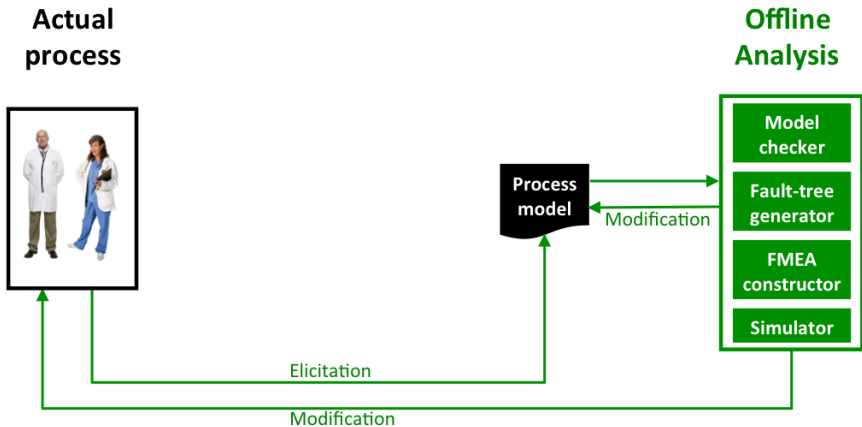
Use detailed, validated process models to provide guidance during ongoing medical processes

- Builds on our previous work on process modeling and analysis

Process Improvement Framework—Prior Work



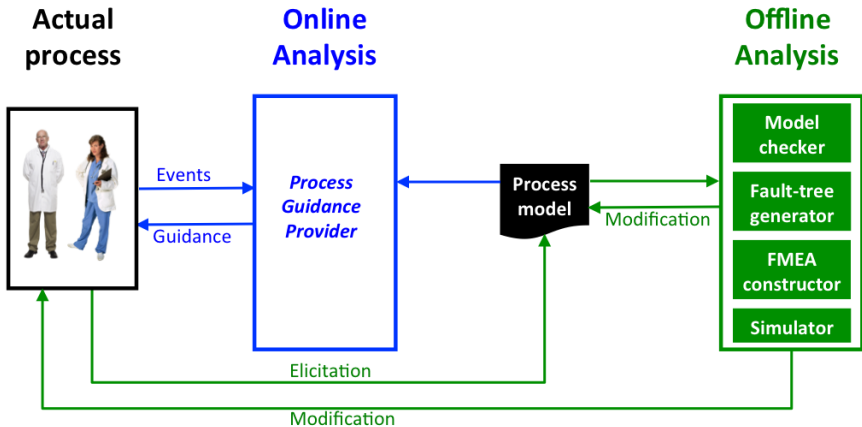
Process Improvement Framework—Prior Work



70% reduction of chemotherapy errors reaching patient

[Mertens 2012, Joint Commission Journal on Quality and Patient Safety]

Process Improvement Framework—Current Research



Process Improvement Framework—Current Research


Home | John Doe | Jane Ray | George Cooper | May 13, 2013 2:22:28 PM EDT

▼  **John Doe** Birthdate: 01-01-1960 Medical Record Number: 442321

Blood Transfusion | Chemotherapy

Blood Transfusion Process

▼ Obtain Patient's Blood Type

- ✓ 14:21 Contact Lab for Patient's Blood Type 
- Test Patient's Blood Type 

✓ 14:22 Pick up Blood from Blood Bank 

▼ Perform Bedside Check

- ✓ Identify Patient   
- ▶ Check Blood Product

Infuse Blood 

Analysis

el er

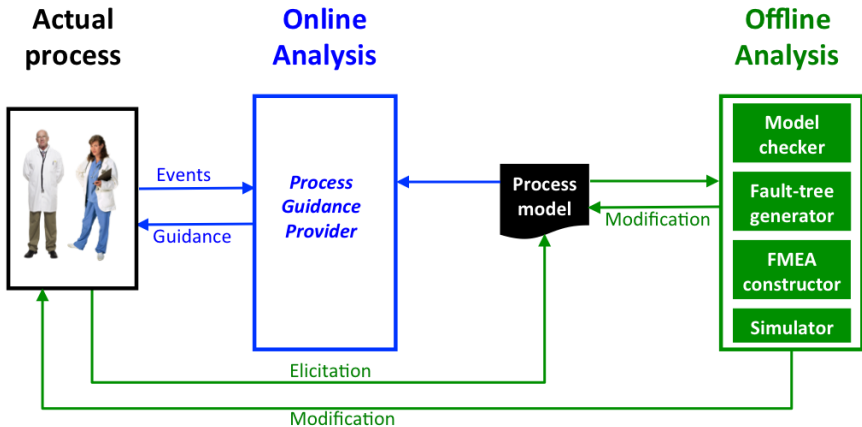
ree tor

A ctor

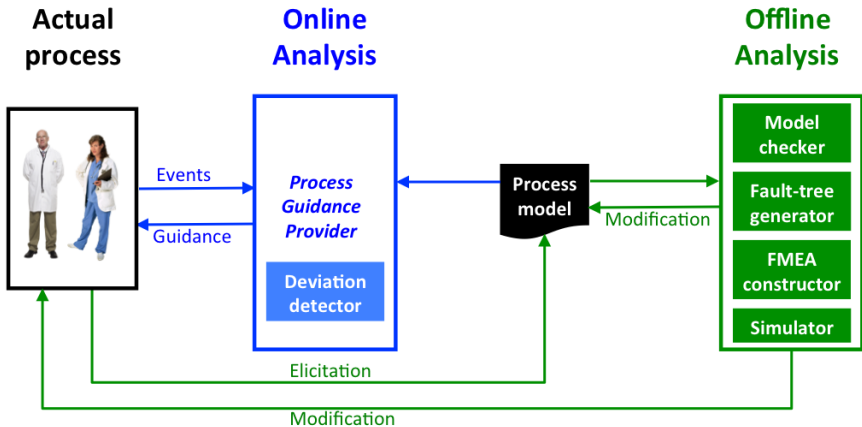
tor

3 / 13

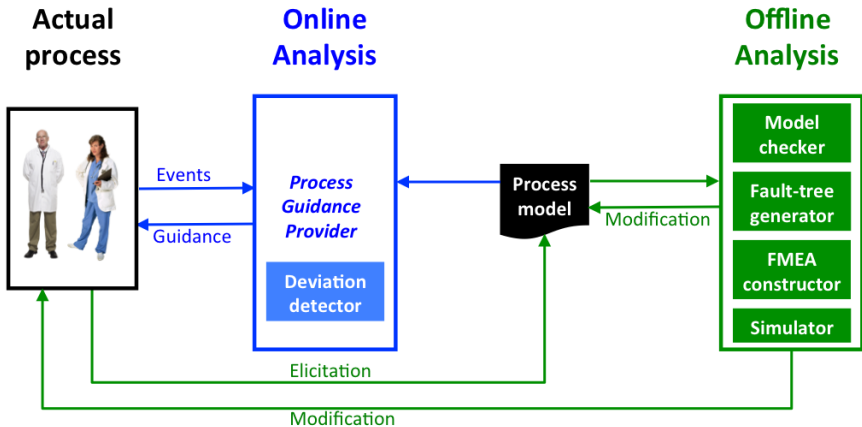
Process Improvement Framework—Current Research



Process Improvement Framework—Current Research



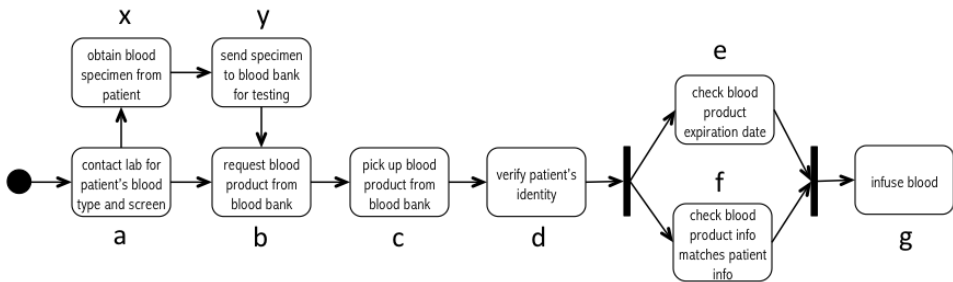
Process Improvement Framework—Current Research



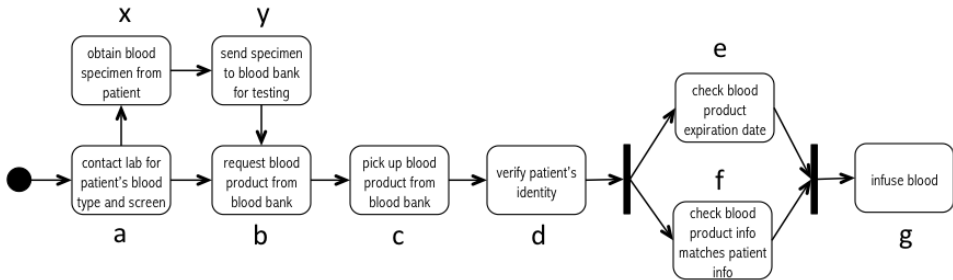
Assumptions

- Process execution events are recorded in real time
- All events of interest are recorded
- Events not of interest are filtered

Example: Simplified Blood Transfusion Process



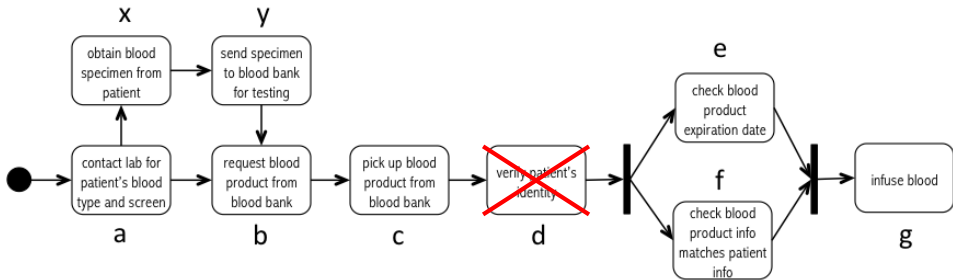
Example: Simplified Blood Transfusion Process



- Scenario:

- The patient is wearing an incorrect ID band—another patient's ID band

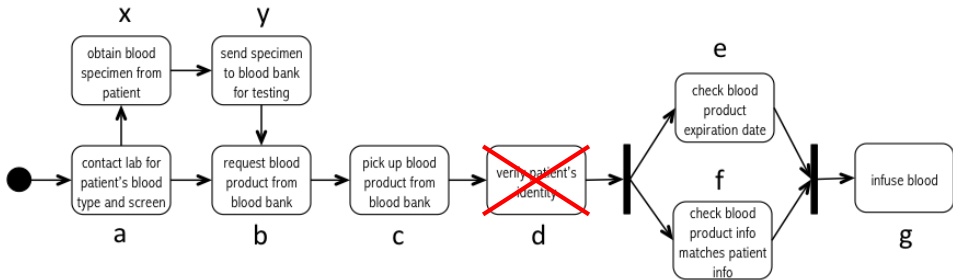
Example: Simplified Blood Transfusion Process



• Scenario:

- The patient is wearing an incorrect ID band—another patient's ID band
- Nurse forgets to verify patient's identity

Example: Simplified Blood Transfusion Process



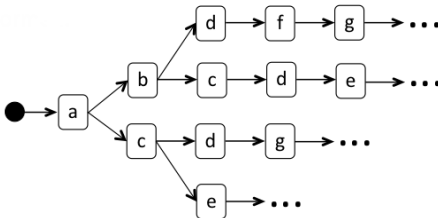
• Scenario:

- The patient is wearing an incorrect ID band—another patient's ID band
- Nurse forgets to verify patient's identity

The patient might receive the wrong blood!

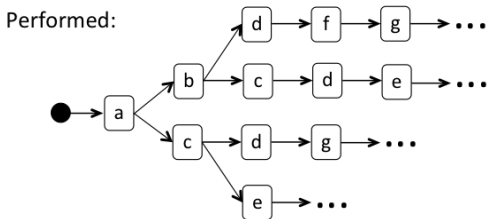
Deviation detection

- Appears fairly straightforward
 - Every time a step is performed, check if the sequence of performed steps is a *trace*
 - Trace: a prefix of a step sequence from the process model
 - Explore process model in a breadth-first manner
 - A deviation is detected when no trace matches the sequence of performed steps



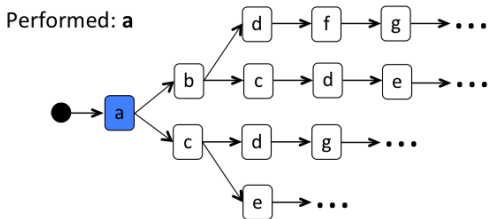
Deviation detection

- Appears fairly straightforward
 - Every time a step is performed, check if the sequence of performed steps is a *trace*
 - Trace: a prefix of a step sequence from the process model
 - Explore process model in a breadth-first manner
 - A deviation is detected when no trace matches the sequence of performed steps



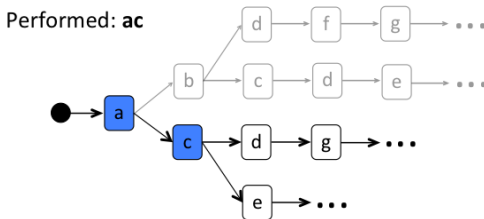
Deviation detection

- Appears fairly straightforward
 - Every time a step is performed, check if the sequence of performed steps is a *trace*
 - Trace: a prefix of a step sequence from the process model
 - Explore process model in a breadth-first manner
 - A deviation is detected when no trace matches the sequence of performed steps



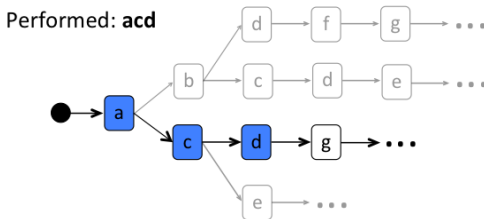
Deviation detection

- Appears fairly straightforward
 - Every time a step is performed, check if the sequence of performed steps is a *trace*
 - Trace: a prefix of a step sequence from the process model
 - Explore process model in a breadth-first manner
 - A deviation is detected when no trace matches the sequence of performed steps



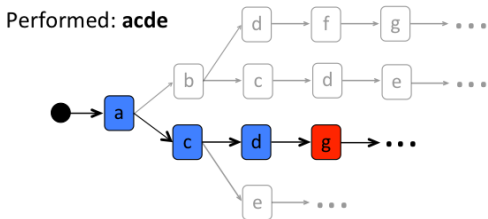
Deviation detection

- Appears fairly straightforward
 - Every time a step is performed, check if the sequence of performed steps is a *trace*
 - Trace: a prefix of a step sequence from the process model
 - Explore process model in a breadth-first manner
 - A deviation is detected when no trace matches the sequence of performed steps



Deviation detection

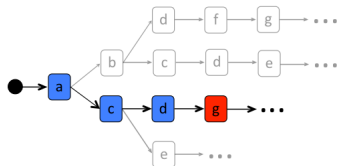
- Appears fairly straightforward
 - Every time a step is performed, check if the sequence of performed steps is a *trace*
 - Trace: a prefix of a step sequence from the process model
 - Explore process model in a breadth-first manner
 - A deviation is detected when no trace matches the sequence of performed steps



Deviation detected!

Issues

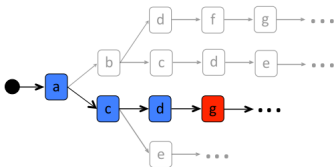
Performed: **acde**



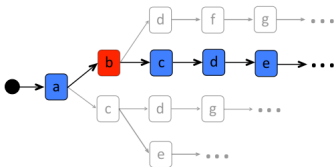
- Do not know the intended trace, so need to consider multiple traces

Issues

Performed: **acde**

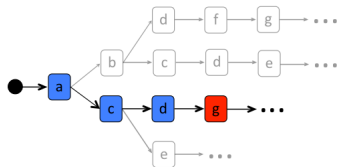


- Do not know the intended trace, so need to consider multiple traces

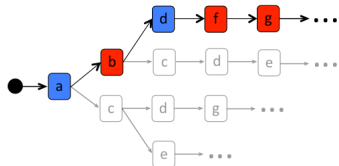
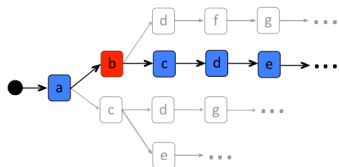


Issues

Performed: **acde**

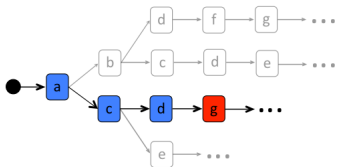


- Do not know the intended trace, so need to consider multiple traces

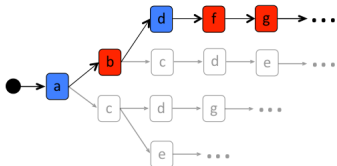
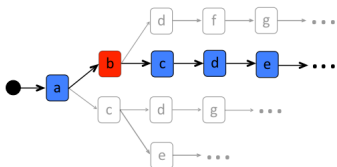


Issues

Performed: **acde**

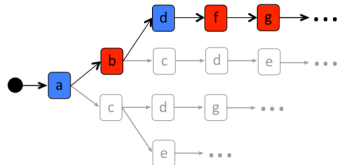
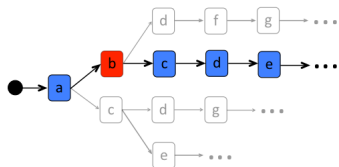
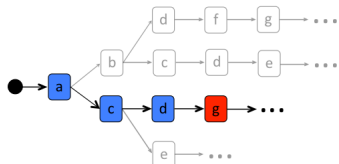


- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected



Issues

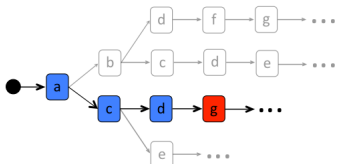
Performed: **acde**



- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*

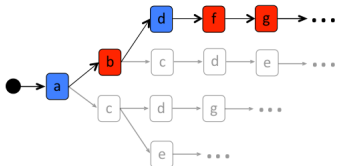
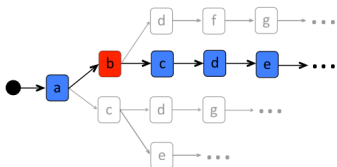
Issues

Performed: **acde**



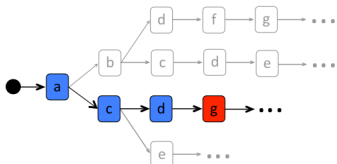
- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*

- Multiple potential errors

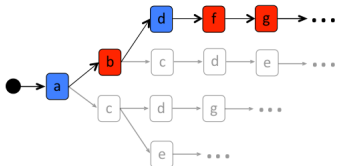
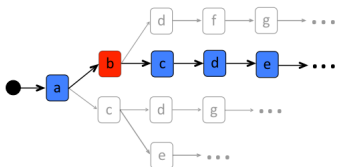


Issues

Performed: **acde**

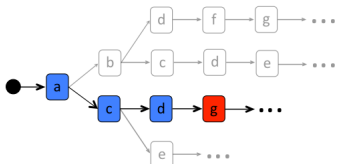


- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*
- Multiple potential errors
 - *Appropriate recovery may depend on the error*

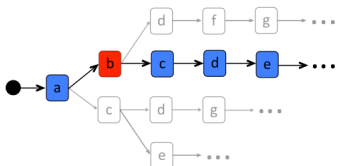


Issues

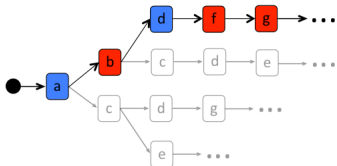
Performed: **acde**



- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*

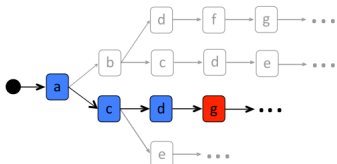


- Multiple potential errors
 - *Appropriate recovery may depend on the error*
 - Provide help to identify most likely and/or most harmful errors

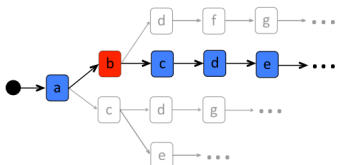


Issues

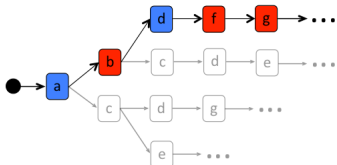
Performed: **acde**



- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*



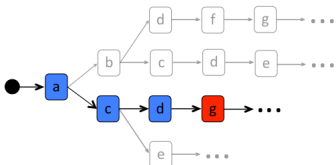
- Multiple potential errors
 - *Appropriate recovery may depend on the error*
 - Provide help to identify most likely and/or most harmful errors



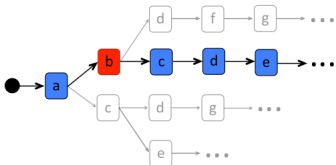
- Impractical to consider all traces

Issues

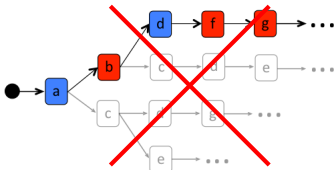
Performed: **acde**



- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*



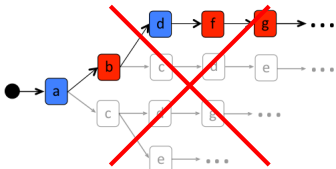
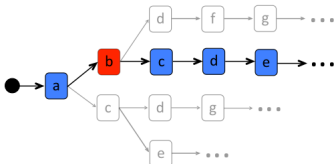
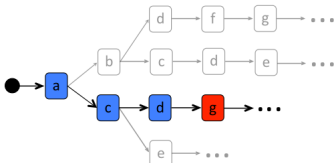
- Multiple potential errors
 - *Appropriate recovery may depend on the error*
 - Provide help to identify most likely and/or most harmful errors



- Impractical to consider all traces

Issues

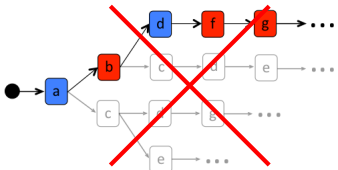
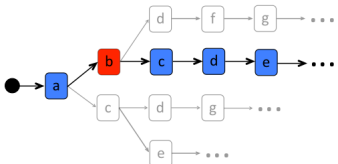
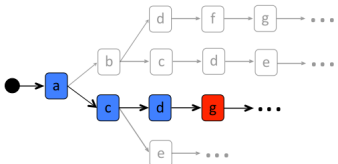
Performed: **acde**



- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*
- Multiple potential errors
 - *Appropriate recovery may depend on the error*
 - Provide help to identify most likely and/or most harmful errors
- Impractical to consider all traces
 - Can use edit distance and sequence comparison techniques to select most likely intended traces

Issues

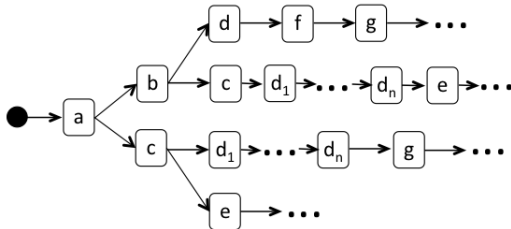
Performed: **acde**



- Do not know the intended trace, so need to consider multiple traces
- Errors might occur before deviation is detected
 - *Harm can be done before a deviation is even detected*
- Multiple potential errors
 - *Appropriate recovery may depend on the error*
 - Provide help to identify most likely and/or most harmful errors
- Impractical to consider all traces
 - Can use edit distance and sequence comparison techniques to select most likely intended traces

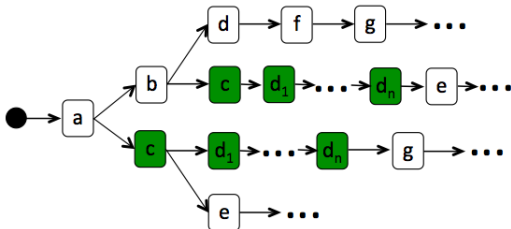
Investigating Techniques for Dealing with Delayed Error Detection

- Statically analyze the process model to discover parts of the model *vulnerable to delayed error detection*



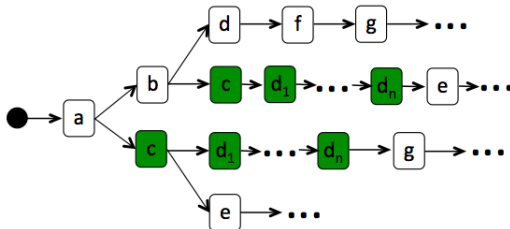
Investigating Techniques for Dealing with Delayed Error Detection

- Statically analyze the process model to discover parts of the model *vulnerable to delayed error detection*



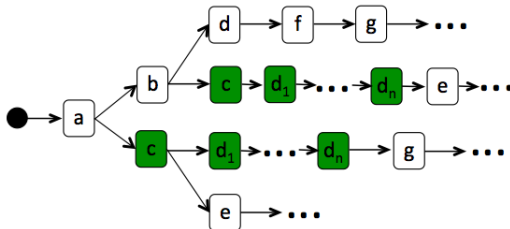
Investigating Techniques for Dealing with Delayed Error Detection

- Statically analyze the process model to discover parts of the model *vulnerable to delayed error detection*
- Record and utilize additional information



Investigating Techniques for Dealing with Delayed Error Detection

- Statically analyze the process model to discover parts of the model *vulnerable to delayed error detection*
- Record and utilize additional information
- Modify the process to make it less vulnerable to delayed error detection

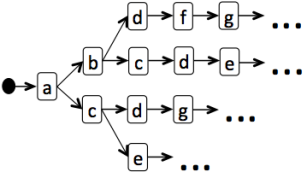
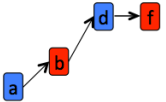
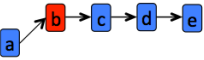



Investigating Techniques to Help Locate Potential Errors

- Identify potential error indexes (PEIs) by *aligning* the performed sequence to the potentially intended traces
 - PEI: index in the performed sequence where an error might have occurred

Investigating Techniques to Help Locate Potential Errors

- Identify potential error indexes (PEIs) by *aligning* the performed sequence to the potentially intended traces
 - PEI: index in the performed sequence where an error might have occurred

Performed: a c d e Process model: 	Potentially intended traces	Dist.	Alignments	PEIs
		2	a b d f a c d e	2,4
	1	a b c d e a _ c d e	2	
	1	a c d g a c d e	4	

Investigating Techniques to Help Locate Potential Errors

- Identify potential error indexes (PEIs) by *aligning* the performed sequence to the potentially intended traces
 - PEI: index in the performed sequence where an error might have occurred
- Rank PEIs based on edit distance

Performed: a c d e	Potentially intended traces	Dist.	Alignments	PEIs
Process model: 		2	<pre> a b d f a c d e </pre>	2,4
		1	<pre> a b c d e a _ c d e </pre>	2
		1	<pre> a c d g a c d e </pre>	4

Investigating Techniques to Help Locate Potential Errors

- Identify potential error indexes (PEIs) by *aligning* the performed sequence to the potentially intended traces
 - PEI: index in the performed sequence where an error might have occurred
- Rank PEIs based on edit distance

Performed: a c d e	Potentially intended traces	Dist.	Alignments	PEIs
Process model: 		2	$\begin{matrix} a & b & d & f \\ a & c & d & e \end{matrix}$	2,4
		1	<i>top-ranked PEI</i> $\begin{matrix} a & b & c & d & e \\ a & _ & c & d & e \end{matrix}$	2
		1	<i>top-ranked PEI</i> $\begin{matrix} a & c & d & g \\ a & c & d & e \end{matrix}$	4

Preliminary Experiment

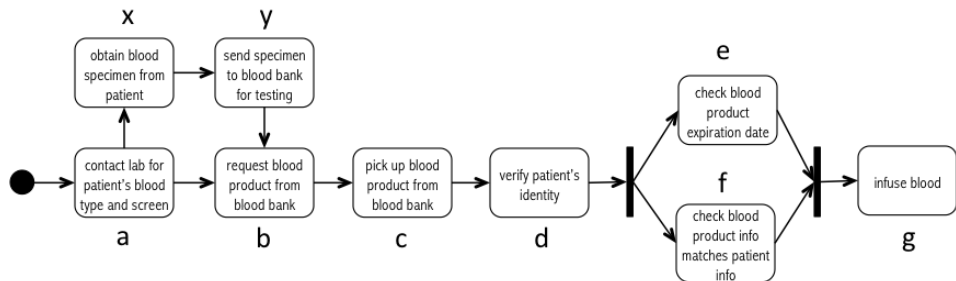
- Used part of Little-JIL process model from the blood transfusion benchmark [*A Benchmark for Evaluating SE Techniques for Improving Medical Processes, SEHC 2010*]
 - 18 exception handling situations
- Generated all traces of up to 15 steps (164 traces total)
- Mutated each trace to represent errors
 - Deletion, insertion, substitution of a single step at each index of each generated trace
- Applied deviation detection and error localization to each mutant

Results

- How often does delayed error detection occur?
 - 0.9% for insertion and substitution mutants;
 - 11.5% for deletion mutants
 - Delayed error detection mostly due to *shuffle regions*
 - Shuffle region: subprocess where steps can occur in any order
 - Delays due to shuffle regions might not be considered delays in practice

Results

- How often does delayed error detection occur?
 - 0.9% for insertion and substitution mutants;
 - 11.5% for deletion mutants
 - Delayed error detection mostly due to *shuffle regions*
 - Shuffle region: subprocess where steps can occur in any order
 - Delays due to shuffle regions might not be considered delays in practice



Results

- How often does delayed error detection occur?
 - 0.9% for insertion and substitution mutants;
 - 11.5% for deletion mutants
 - Delayed error detection mostly due to *shuffle regions*
 - Shuffle region: subprocess where steps can occur in any order
 - Delays due to shuffle regions might not be considered delays in practice
- How long is the delay?
 - Maximum 3 steps, 1.5 on average

Results

- How often does delayed error detection occur?
 - 0.9% for insertion and substitution mutants;
 - 11.5% for deletion mutants
 - Delayed error detection mostly due to *shuffle regions*
 - Shuffle region: subprocess where steps can occur in any order
 - Delays due to shuffle regions might not be considered delays in practice
- How long is the delay?
 - Maximum 3 steps, 1.5 on average
- How often is the “real” error location the highest-ranked PEI?
 - Mutation index always among the PEIs tied for highest rank

Results

- How often does delayed error detection occur?
 - 0.9% for insertion and substitution mutants;
 - 11.5% for deletion mutants
 - Delayed error detection mostly due to *shuffle regions*
 - Shuffle region: subprocess where steps can occur in any order
 - Delays due to shuffle regions might not be considered delays in practice
- How long is the delay?
 - Maximum 3 steps, 1.5 on average
- How often is the “real” error location the highest-ranked PEI?
 - Mutation index always among the PEIs tied for highest rank
- How often are there multiple PEIs tied for the highest rank?
 - 2.5% of insertion mutants, 1% of substitution mutants, 25.7% of deletion mutants
 - Large percentage for deletion mutants due to shuffle regions

Results

- How often does delayed error detection occur?
 - 0.9% for insertion and substitution mutants;
 - 11.5% for deletion mutants
 - Delayed error detection mostly due to *shuffle regions*
 - Shuffle region: subprocess where steps can occur in any order
 - Delays due to shuffle regions might not be considered delays in practice
- How long is the delay?
 - Maximum 3 steps, 1.5 on average
- How often is the “real” error location the highest-ranked PEI?
 - Mutation index always among the PEIs tied for highest rank
- How often are there multiple PEIs tied for the highest rank?
 - 2.5% of insertion mutants, 1% of substitution mutants, 25.7% of deletion mutants
 - Large percentage for deletion mutants due to shuffle regions
 - How many PEIs tied for the highest rank?
 - Maximum 3, 2.1 on average

Related Work

- Process aids
 - E.g., Checklists, medical guidelines
 - Tend to specify only major steps during normal flow, omitting exceptional flow and concurrent execution
 - Some aids intended only during training, not during process performance
 - If used online, add to the workload and are not an integral part of the process
- Semi-automated online process guidance
 - Fitzgerald et al.–process guidance in trauma center
 - Encode a medical algorithm for trauma resuscitation
 - System checks compliance with encoded algorithm
 - **Reduction in error rate and morbidity**
 - Concurrency and exception handling not supported
 - **We expect to achieve similar results but for more complex processes**

Related Work

- Formal frameworks for dealing with process deviation
 - Theoretical framework for process deviations (Cugola et al.)
 - String comparison-based validation of process definitions (Cook and Wolf)
- Fault localization and anomaly detection
 - Provides high-level interpretation of detected program execution anomalies (Babenko et al.)
 - Locates faults in programs (Jones et al., Renieries and Reiss)
- Theory of human errors
 - Cognitive science-based classification (Reason)
 - Error phenotype classification (Hollnagel)
 - Classification of ED errors (Henneman et al.)

Summary and Future Work

- Summary
 - Proposed an approach for online deviation detection and error localization to catch errors before harm is done
 - Identified some research issues
 - Implemented a prototype and performed a preliminary evaluation
 - Results are promising

Summary and Future Work

● Summary

- Proposed an approach for online deviation detection and error localization to catch errors before harm is done
- Identified some research issues
- Implemented a prototype and performed a preliminary evaluation
 - Results are promising

● Future directions

- Evaluate proposed approach with larger, more realistic medical process models using real sequences of performed steps
- Investigate ways to deal with delayed error detection
- Evaluate effectiveness of error localization and consider other kinds of potentially useful information
- Leverage information from process model to:
 - Select traces to compare against sequence of performed steps
 - Select edit operations based on context
- Continue development of the process guidance framework