Modelling User Availability in Workflow Resiliency Analysis

John C. Mace, Charles Morisset & Aad van Moorsel

School of Computing Science
Newcastle University, UK

21 April 2015
Take Home Message

• Automatically calculate the resiliency of a workflow
  – Resiliency is a measure of success rate for a workflow executed by users who may become unavailable at runtime
  – Resiliency indicates risk of: workflow failure, security policy violation
  – Resiliency informs: mitigation strategy, redesign, recruitment, etc.

• Runtime user availability can be modelled in several ways when calculating resiliency

• Availability model choice can impact the resiliency calculated for the same workflow
  – Large resiliency variance
  – Also impacts on complexity, e.g., computation time
Workflow

Users = \{u_1,u_2,u_3\}

- Tasks
- Ordering
- Users

- Permissions
- Constraints
- Assignment?
Workflow Satisfiability Problem

Design time

\[ a_1 : \quad u_1 \quad u_3 \quad \times \quad \text{Design time} \]
\[ a_2 : \quad u_1 \quad u_2 \quad u_3 \]

Run time

\[ a_2 : \quad u_1 \quad \text{u}_2 \quad \text{unavailable} \rightarrow u_3 \quad \times \quad \text{Run time} \]
\[ a_3 : \quad u_2 \quad u_3 \quad u_1 \]
Workflow Resiliency

- $k = 1$, 10 possible cases of up to 1 unavailable user
- 1 example case - $u_2$ unavailable at $t_1$

**Workflow $w_1$**
- 0 resiliency $\rightarrow$ current
- $w_1$ : assign 4 of 10 cases $\rightarrow$ new

**Workflow $w_2$**
- 0 resiliency $\rightarrow$ current
- $w_2$ : assign 9 of 10 cases $\rightarrow$ new
Assignment Process

- Maximise $v$ returned by value function of a Markov Decision Process (MDP)

- **WSP** → full user availability
  - can Success be reached?
  - $v = 0$ or $1$

- **Resiliency** → probabilistic user availability
  - maximum probability of reaching Success?
  - $0 \geq v \leq 1$
Non-deterministic Availability

- **Static model** – make choice before start of workflow
- **Decremental model** - make choice for each task while $u_i$ is available
- **Dynamic model** - make choice for each task $[m_1]$
Bounded Availability

- Up to $k$ users can become unavailable across entire workflow
- For $k = 1$, consider all possible cases
  - Assume decremental availability
  - Assume cases are equiprobable

<table>
<thead>
<tr>
<th>All users available</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1,u_2,u_3$</td>
<td>$u_1,u_2,u_3$</td>
<td>$u_1,u_2,u_3$</td>
<td>$u_1,u_2,u_3$</td>
</tr>
<tr>
<td>$u_1$ unavailable at $t_3$</td>
<td>$u_1,u_2,u_3$</td>
<td>$u_1,u_2,u_3$</td>
<td>$u_2,u_3$</td>
</tr>
<tr>
<td>$u_2$ unavailable at $t_2$</td>
<td>$u_1,u_2,u_3$</td>
<td>$u_1,u_3$</td>
<td>$u_1,u_3$</td>
</tr>
<tr>
<td>$u_3$ unavailable at $t_1$</td>
<td>$u_1,u_2$</td>
<td>$u_1,u_2$</td>
<td>$u_1,u_2$</td>
</tr>
</tbody>
</table>

... and so on for every possible case $[m_2]$
Probabilistic Availability

\( u_i \) has same probability for each task \( t_i \) \[m_3\]

\[
\begin{align*}
u_1 & \quad 0.75 \quad \text{Available for} \quad t_i \\
0.25 & \quad \text{Unavailable for} \quad t_i
\end{align*}
\]

\( u_i \) has different probability for each task \( t_i \) \[m_4\]

\[
\begin{align*}
u_1 & \quad 0.75 \quad \text{Available for} \quad t_1 \\
0.25 & \quad \text{Unavailable for} \quad t_1
\end{align*}
\]

\[
\begin{align*}
u_1 & \quad 0.68 \quad \text{Available for} \quad t_2 \\
0.32 & \quad \text{Unavailable for} \quad t_2
\end{align*}
\]

\[
\begin{align*}
u_1 & \quad 0.93 \quad \text{Available for} \quad t_3 \\
0.07 & \quad \text{Unavailable for} \quad t_3
\end{align*}
\]
Combined Models

- Combine both non-deterministic and probabilistic availability
  - Non-deterministic for $t_2$
  - Probabilistic for $t_1$ and $t_3$

- More complex, dependent availability models can be considered, e.g.
  - Current availability
  - Availability for previous tasks
  - Availability of other users

- $u_1$
  - Available for $t_1$
  - Unavailable for $t_1$

- $u_1$
  - Available for $t_2$
  - Unavailable for $t_2$

- $u_1$
  - Available for $t_3$
  - Unavailable for $t_3$
Calculating Resiliency

• Solve **MDP** to find $\nu$ using model checker **PRISM**\(^1\)
• Model consists of interactive named modules containing:
  
  – Variables \hspace{1em} name : type \textbf{init} \textbf{value}
  
  – Commands \hspace{1em} [\textit{label}] \textit{guard} \rightarrow p_1 : \text{update}_1 \& \ldots \& p_n : \text{update}_n

• Non-deterministic choice
  
  – [\textit{label}_i] \textit{guard}_i \rightarrow \text{update}_1
  
  – [\textit{label}_i] \textit{guard}_i \rightarrow \text{update}_2

• Satisfiability property
  
  – Pmax =? [ F (t=-1) \& \neg \text{fail} ]

\(^1\)http://www.prismmodelchecker.org/
## Resiliency Analysis

<table>
<thead>
<tr>
<th>Model</th>
<th>Res</th>
<th>States</th>
<th>Transitions</th>
<th>Build time (s)</th>
<th>Verify time (s)</th>
<th>File size (KB)</th>
<th>Size on disk (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[m₁]</td>
<td>1.00</td>
<td>8530</td>
<td>31321</td>
<td>0.219</td>
<td>0.015</td>
<td>2.51</td>
<td>4.00</td>
</tr>
<tr>
<td>[m₂]</td>
<td>0.43</td>
<td>50489</td>
<td>64377</td>
<td>0.125</td>
<td>0.172</td>
<td>8.95</td>
<td>12.00</td>
</tr>
<tr>
<td>[m₃]</td>
<td>0.41</td>
<td>8530</td>
<td>31321</td>
<td>0.172</td>
<td>0.016</td>
<td>2.50</td>
<td>4.00</td>
</tr>
<tr>
<td>[m₄]</td>
<td>0.79</td>
<td>8530</td>
<td>31321</td>
<td>0.172</td>
<td>0.016</td>
<td>3.21</td>
<td>4.00</td>
</tr>
</tbody>
</table>

[m₁]: dynamic, non-deterministic
[m₂]: decremental, bounded (k=2), equiprobable
[m₃]: dynamic, probabilistic (same per task)
[m₄]: dynamic, probabilistic (different per task)
Conclusion

- We can encode a workflow with a user availability model as a Markov Decision Processes (MDP)
- Used the model checker PRISM to automatically solve an MDP and provide measure of workflow success rate, or resiliency
- Shown user availability in workflows can be modelled in several ways
  - Probabilistic, non-deterministic, bounded, etc.
- Highlighted availability model choice can have an impact on resiliency computations for the same workflow
- We make no assumption on which one is best as this will be context dependent
Future Work

• Analyse different sizes of workflow
  - How does computing resiliency scale?
  - How do complexity metrics change?

• More complex security policies
  - Cardinality constraints

• Development of tools and methodologies for workflow designers
  - Understand what is an appropriate availability model?
  - Automatically calculate appropriate resiliency
References


Contact: john.mace@ncl.ac.uk