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

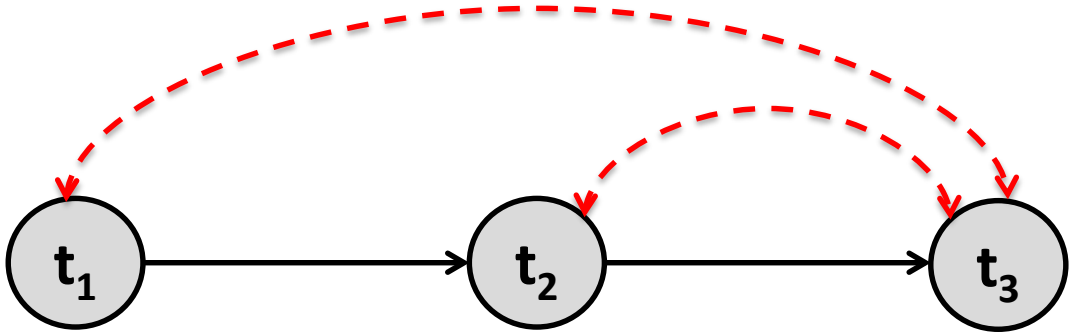
The Science of Security initiative is funded by the National Security Agency.

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 = $\{u_1, u_2, u_3\}$



u_1 or u_2?

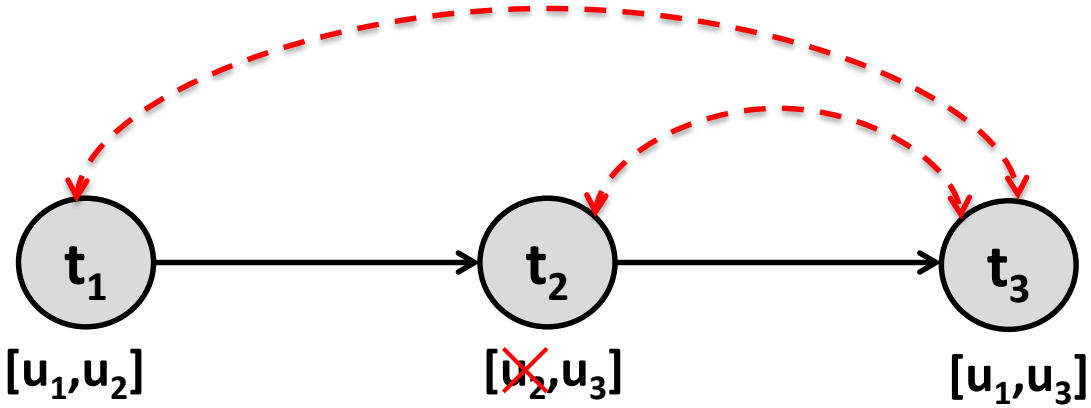
u_2 or u_3?

u_1 or u_3?

- Tasks
- Ordering
- Users

- Permissions
- Constraints
- Assignment?

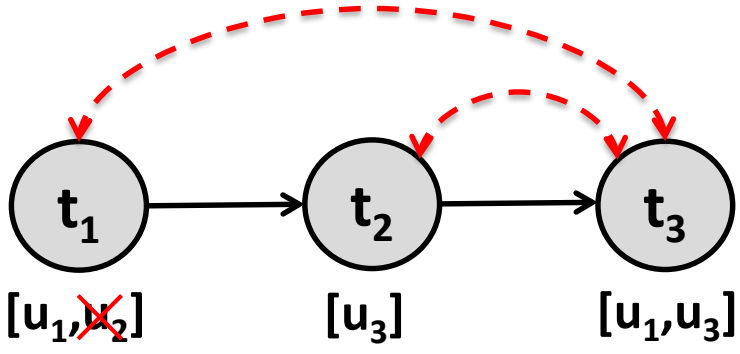
Workflow Satisfiability Problem



a_1 :	u_1	u_3	✗	Design time
a_2 :	u_1	u_2	u_3	
a_2 :	u_1	u_2 unavailable -> u_3	✗	Run time
a_3 :	u_2	u_3	u_1	

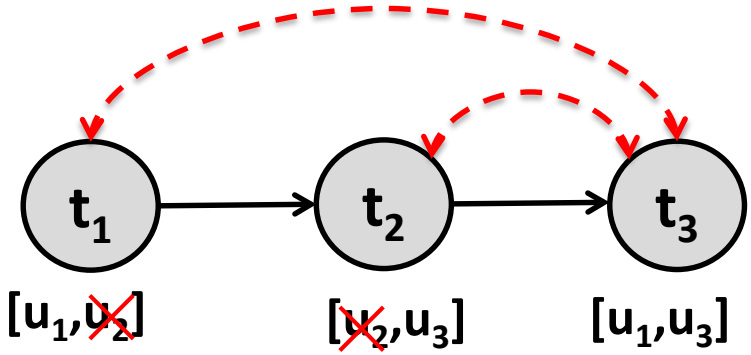
Workflow Resiliency

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



Workflow w_1

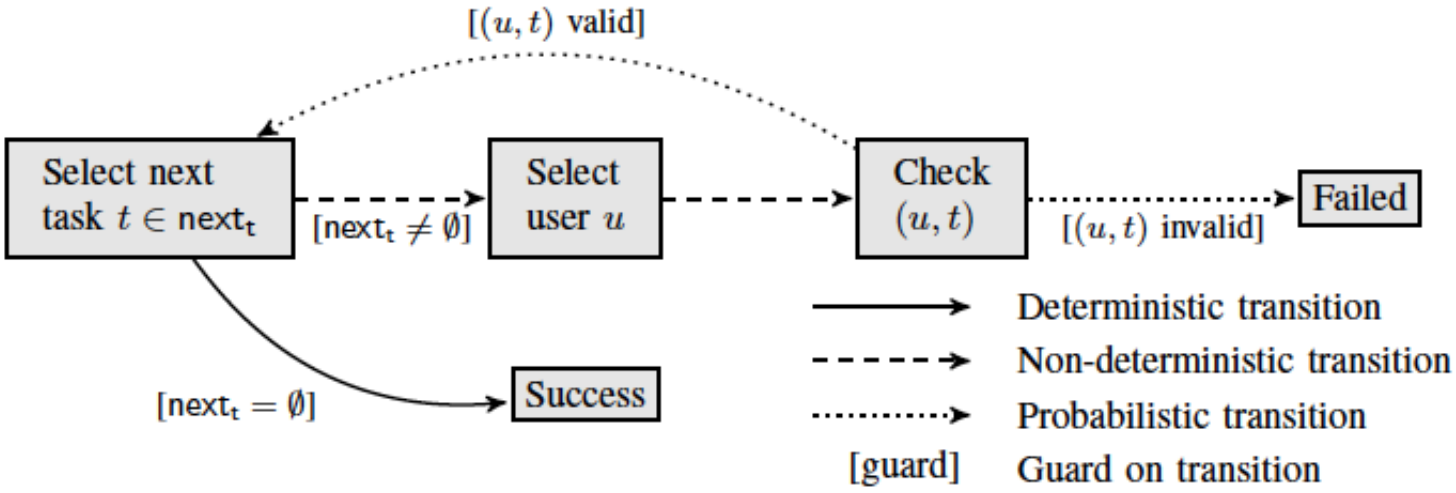
- **0** resiliency –> *current*
- w_1 : assign **4** of **10** cases –> *new*



Workflow $w_2 - u_2$ added to t_2

- **0** resiliency –> *current*
- w_2 : assign **9** of **10** cases –> *new*

Assignment Process

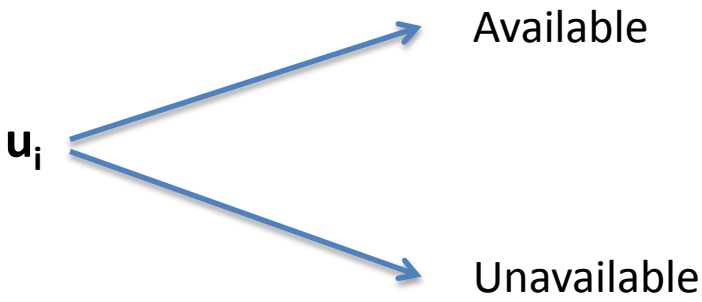


- Maximise ***v*** returned by value function of a Markov Decision Process (**MDP**)

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

- **Resiliency** –> probabilistic user availability
  - maximum probability of reaching **Success**?
  - **0 ≥ *v* ≤ 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

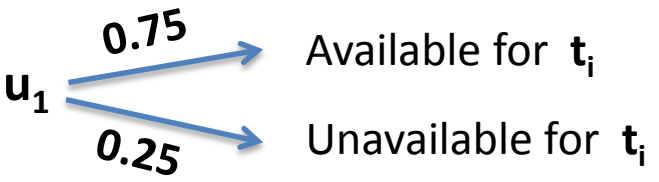
- Up to ***k*** users can become unavailable across entire workflow
- For ***k* = 1**, consider all possible cases
  - Assume decremental availability
  - Assume cases are equiprobable

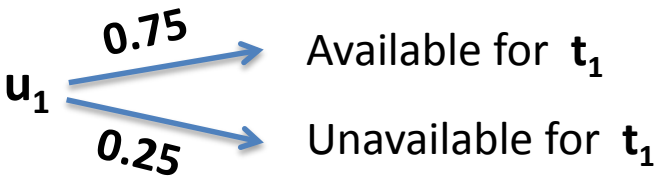
	t_1	t_2	t_3
All users available	u_1,u_2,u_3	u_1,u_2,u_3	u_1,u_2,u_3
u_1 unavailable at t_3	u_1,u_2,u_3	u_1,u_2,u_3	u_2,u_3
u_2 unavailable at t_2	u_1,u_2,u_3	u_1,u_3	u_1,u_3
u_3 unavailable at t_1	u_1,u_2	u_1,u_2	u_1,u_2

… and so on for every possible case **[m_2]**

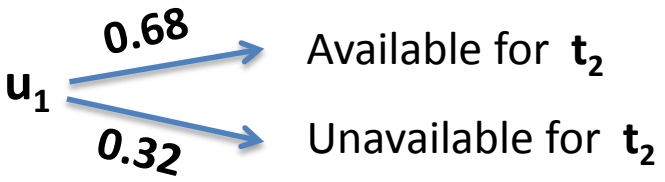
Probabilistic Availability

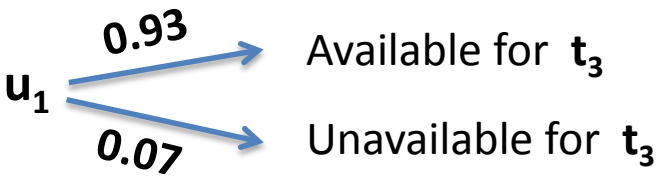
$\mathbf{u_i}$ has same probability for each task $\mathbf{t_i}$ $[m_3]$





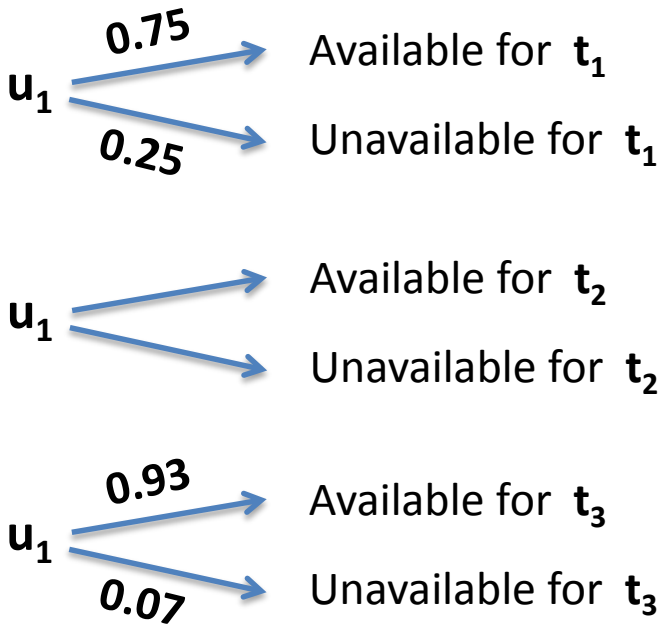
$\mathbf{u_i}$ has different probability for each task $\mathbf{t_i}$ $[m_4]$





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



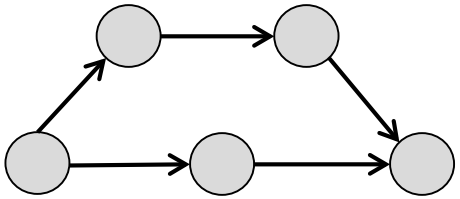
Calculating Resiliency

- Solve **MDP** to find $\boldsymbol{v}$ using model checker **PRISM**[1]
- Model consists of interactive named modules containing:
  - Variables $name$ **:** $type$ **init** $value$
  - Commands $[label]\ guard\ \text{->}\ p_1 : update_1\ \&\ \ldots\ \&\ p_n : update_n$
- Non-deterministic choice
  - $[label_i]\ guard_i\ \text{->}\ update_1$
  - $[label_i]\ guard_i\ \text{->}\ update_2$
- Satisfiability property
  - Pmax =? [F (t=-1) & (!fail)]

[1] http://www.prismmodelchecker.org/

Resiliency Analysis

Model	Res	States	Transitions	Build time (s)	Verify time (s)	File size (KB)	Size on disk (KB)
$[m_1]$	1.00	8530	31321	0.219	0.015	2.51	4.00
$[m_2]$	0.43	50489	64377	0.125	0.172	8.95	12.00
$[m_3]$	0.41	8530	31321	0.172	0.016	2.50	4.00
$[m_4]$	0.79	8530	31321	0.172	0.016	3.21	4.00



$[m_1]$: dynamic, non-deterministic

$[m_2]$: decremental, bounded (k=2), equiprobable

$[m_3]$: dynamic, probabilistic (same per task)

$[m_4]$: 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

- J. Crampton, G. Gutin, and A. Yeo. On the parameterized complexity of the workflow satisfiability problem. In *Proceedings* CCS '12, pages 857-868, New York, NY, USA, 2012. ACM.
- Q. Wang and N. Li. Satisfiability and resiliency in workflow authorization systems. *ACM Trans. Inf. Syst. Secur.*, 13(4):40:1-40:35, Dec. 2010.
- J. Mace, C. Morisset, and A. van Moorsel. Quantitative Workflow Resiliency. In *Computer Security – ESORICS 2014*, volume 8712 of LNCS, pages 344-361. Springer, 2014.
- M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In Proceedings CAV'11, volume 6806 of LNCS, pages 585-591. Springer, 2011.

Contact : **john.mace@ncl.ac.uk**