Sensitivity Analysis of Probabilistic Workflow Models with Security Constraints

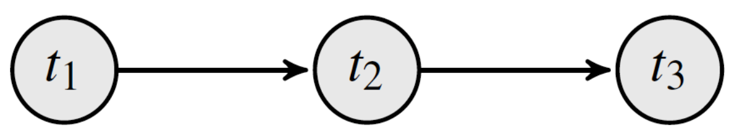
John C. Mace, Nipun Thekkummal, Aad van Moorsel
School of Computing Science, Newcastle University, UK
john.mace@ncl.ac.uk, nipun.thekkummal@ncl.ac.uk, aad.vanmoorsel@ncl.ac.uk

BACKGROUND

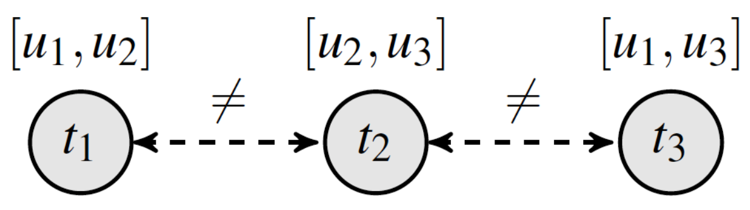
- Workflow security constraints restrict which tasks a user can perform each time a workflow is executed
- Completing a workflow consists of assigning each task to an available user whilst respecting all security constraints
- Security constraints can make the availability of some users more critical than others for workflow completion
- A user with a junior role may be more critical to workflow completion and arguably have more 'power' than a user with a senior role
- Malicious users could use their power to obstruct workflow completion by restricting their availability. This may necessitate security constraint overrides to complete a workflow
- The maximum probability of workflow completion by users who may become unavailable is known as ***workflow resiliency***[1]
- We want to identify the power of users by measuring how changes in user availability impact the resiliency of a workflow

SECURITY CONSTRAINED WORKFLOWS

- We consider workflows with:
  - ***Authorization constraints*** **~** which individual tasks can be assigned to which users
  - ***Separation of duty constraints*** **~** which tasks cannot be assigned to the same user in a single execution
  - ***Binding of duty constraints*** **~** which tasks which must be assigned to the same user in a single execution



Workflow example

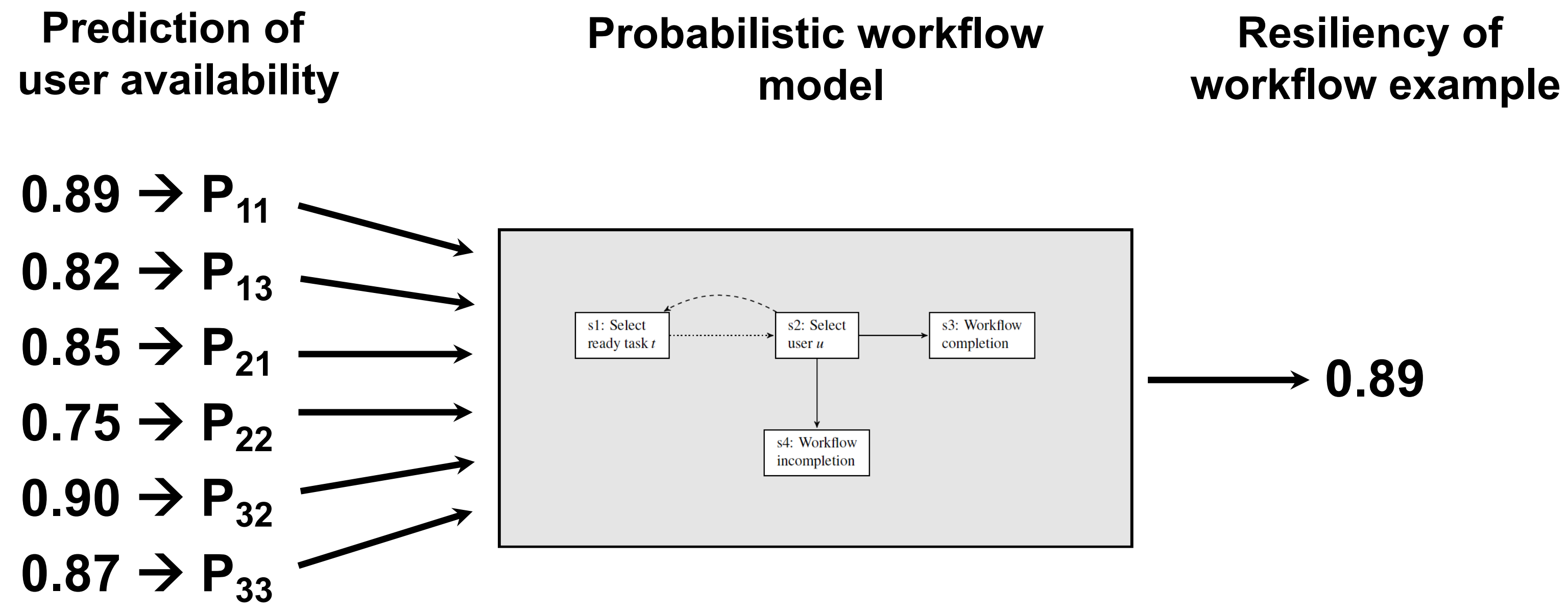


Workflow security constraints

- Each user is authorized to perform two tasks, e.g. u_2 can be assigned to tasks t_1 and t_2
- Two separation of duty constraints between t_1 and t_2, and t_2 and t_3
- User u_2 cannot be assigned to t_1 and t_2 in the same workflow execution

WORKFLOW RESILIENCY

- Workflow resiliency can be computed by modelling an abstracted workflow task assignment process[2]
- The probability of user u_i being available for authorized task t_j is an input parameter P_{ij} for a workflow model[3]
- Model properties are verified using the probabilistic model checker PRISM[4]
- We ask PRISM to verify the maximum probability of reaching a model state which indicates workflow completion



Prediction of user availability | **Probabilistic workflow model** | **Resiliency of workflow example**

0.89 → P_{11}
0.82 → P_{13}
0.85 → P_{21}
0.75 → P_{22}
0.90 → P_{32}
0.87 → P_{33}

→ 0.89

SENSITIVITY ANALYSIS USING DIFFERENTIAL METHOD

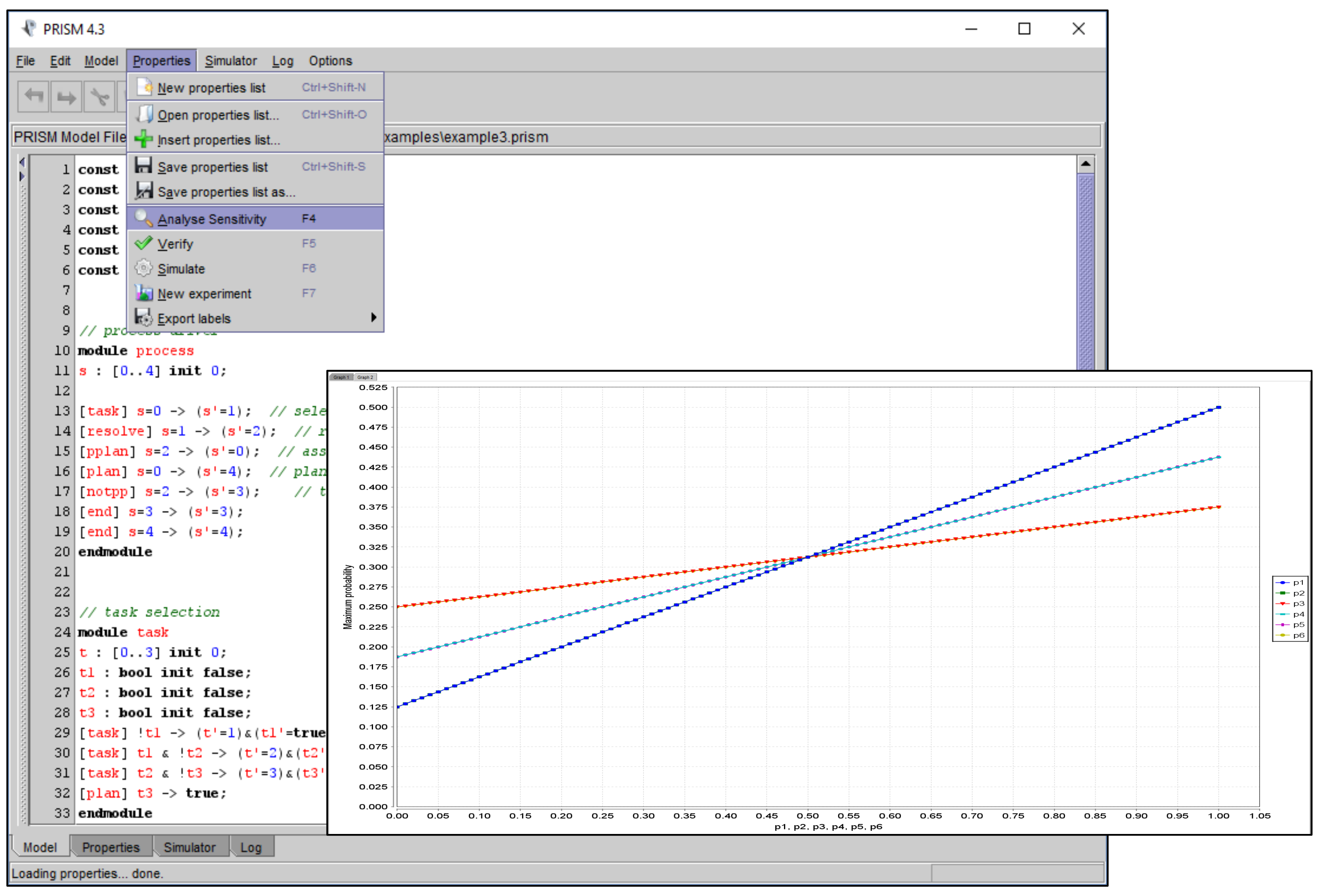
- Sensitivity analysis determines how different values of a model's input parameters impact the model's output value
- Differential analysis is conducted by changing one parameter at a time whilst all other parameters are assigned their mean value
- The rate of change of output to input values is calculated for the entire range of inputs and summed to get the sensitivity coefficient

$$\text{Sensitivity coefficient } = \sum\left(\frac{\Delta y}{\Delta x} \times \frac{x}{y}\right)$$

- The sensitivity of each user's availability is a good indicator of their power over the resiliency of a workflow
- A large change in a user's availability may have little or no effect on resiliency whilst a small change may have a large effect
- Security constraints can be reconfigured to redistribute and align user power with the seniority of their role

SENSITIVITY ANALYSIS IN PRISM

- We have implemented sensitivity analysis functionality into the probabilistic model checker PRISM



```
const
const
const
const
const
const


// pro
module process
s : [0..4] init 0;

[task] s=0 -> (s'=1);  // sele
[resolve] s=1 -> (s'=2);  // r
[pplan] s=2 -> (s'=0);  // ass
[plan] s=0 -> (s'=4);  // plan
[notpp] s=2 -> (s'=3);   // t
[end] s=3 -> (s'=3);
[end] s=4 -> (s'=4);
endmodule


// task selection
module task
t : [0..3] init 0;
t1 : bool init false;
t2 : bool init false;
t3 : bool init false;
[task] !t1 -> (t'=1)&(t1'=true
[task] t1 & !t2 -> (t'=2)&(t2'
[task] t2 & !t3 -> (t'=3)&(t3'
[plan] t3 -> true;
endmodule
```

Loading properties... done.

- ***GUI mode*** PRISM generates a plot for each parameter where the slope of the plot signifies the parameter's sensitivity
- ***Command Line mode*** PRISM ranks input parameters by their sensitivity using the sensitivity coefficients

Input parameter sensitivity for example workflow model		
1. P_{11}:0.246	2. P_{21}:0.087	3. P_{32}:0.167
1. P_{13}:0.246	2. P_{33}:0.087	3. P_{22}:0.167

- User u_1 has most power over the resiliency of the workflow example, distributed equally across both tasks t_1 and t_3

REFERENCES

1. J. C. Mace, C. Morisset, A. van Moorsel "Quantitative Workflow Resiliency", *ESORICS*, 2014
2. J. C. Mace, C. Morisset, A. van Moorsel "Impact of Policy Design on Workflow Resiliency Computation Time", *QEST*, 2015
3. J. C. Mace, C. Morisset, A. van Moorsel "Modelling User Availability in Workflow Resiliency Analysis", *HotSoS*, 2015
4. M. Kwiatkowska, G. Norman, D. Parker "PRISM 4.0: Verification of Probabilistic Real-time Systems", *CAV*, 2011

This material is based upon work supported by the Maryland Procurement Office under Contract No. H98230-14-C-0141

SCIENCE OF SECURITY
VIRTUAL ORGANIZATION
Funded by the National Security Agency.

INFORMATIONTRUST
INSTITUTE