

# Evaluating Finite State Machine-Based Testing Methods on RBAC systems

Carlos Diego Nascimento Damasceno, PhD Candidate – [damascenodiego@usp.br](mailto:damascenodiego@usp.br)

Advisor: Prof. Dr. Adenildo da Silva Simão

Laboratory of Software Engineering – LabES

Institute of Mathematics and Computer Science – ICMC

University of Sao Paulo – USP

Sao Carlos – SP – Brazil



# Agenda

1. *Context, Motivation and Objective*
2. *FSM-Based Testing and Role-Based Access Control (RBAC)*
3. *Experimental Framework and Analysis of Results*
4. *Conclusion and Future work*



# Context

- Software security is a *major requirement* of industrial-scale IT systems
- Access control systems
  - Mediate users access to resources
  - Role-Based Access Control (**RBAC**)
    - **Grouping privileges**
    - Users  $\leftrightarrow$  Roles  $\leftrightarrow$  Permissions



# Motivation

- RBAC and Software testing
- FSM-Based Testing of RBAC <sup>[1]</sup>
  - **Effective** but **large** test suites
- ***Recent FSM testing methods tend to rely on fewer test cases*** <sup>[2]</sup>



[1] MASOOD, A.; BHATTI, R.; GHAFOR, A.; MATHUR, A. P. Scalable and effective test generation for role-based access control systems. IEEE Transactions on Software Engineering, IEEE Press, Piscataway, NJ, USA, v. 35, n. 5, p. 654–668, Sep. 2009.

[2] ENDO, A. T.; SIMAO, A. Evaluating test suite characteristics, cost, and effectiveness of fsm based testing methods. Information and Software Technology, v. 55, n. 6, p. 1045 – 1062, 2013.

# Motivation

- RBAC and Software testing
- FSM-Based Testing of RBAC <sup>[1]</sup>
  - **Effective** but **large** test suites
- *Recent FSM testing methods tend to rely on fewer test cases* <sup>[2]</sup>
  - ***Random FSM models ≠ RBAC policies as FSMs***



[1] MASOOD, A.; BHATTI, R.; GHAFOR, A.; MATHUR, A. P. Scalable and effective test generation for role-based access control systems. IEEE Transactions on Software Engineering, IEEE Press, Piscataway, NJ, USA, v. 35, n. 5, p. 654–668, Sep. 2009.

[2] ENDO, A. T.; SIMAO, A. Evaluating test suite characteristics, cost, and effectiveness of fsm based testing methods. Information and Software Technology, v. 55, n. 6, p. 1045 – 1062, 2013.

# Objective

Compare ***recent*** and ***traditional*** FSM-based ***testing methods*** on ***RBAC domain***

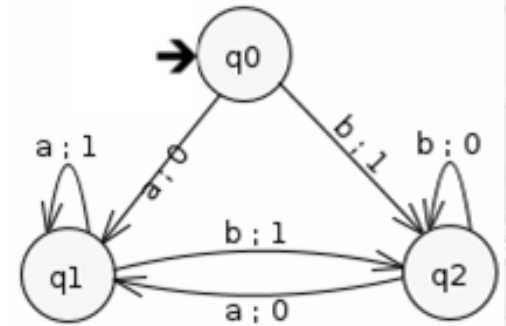
a. Test ***characteristics*** and ***Effectiveness***

i. ***number of resets, avg. test case length*** and ***test suite length***

ii. ***RBAC fault domain***

# Finite State Machine Based Testing

- Finite state machines (**FSM**) are widely used for modeling **reactive systems** <sup>[5]</sup>
  - **FSM-Based Testing** *checks if an SUT **conforms** to a given specification*
  - Mealy Machines (states + transitions labeled with I/O)
- FSM-Based Testing Methods <sup>[5]</sup>
  - **Traditional** methods (W and HSI)
  - **Recent** methods (SPY) <sup>[4]</sup>
    - Reduces tests (e.g. **~40% shorter** than HSI)



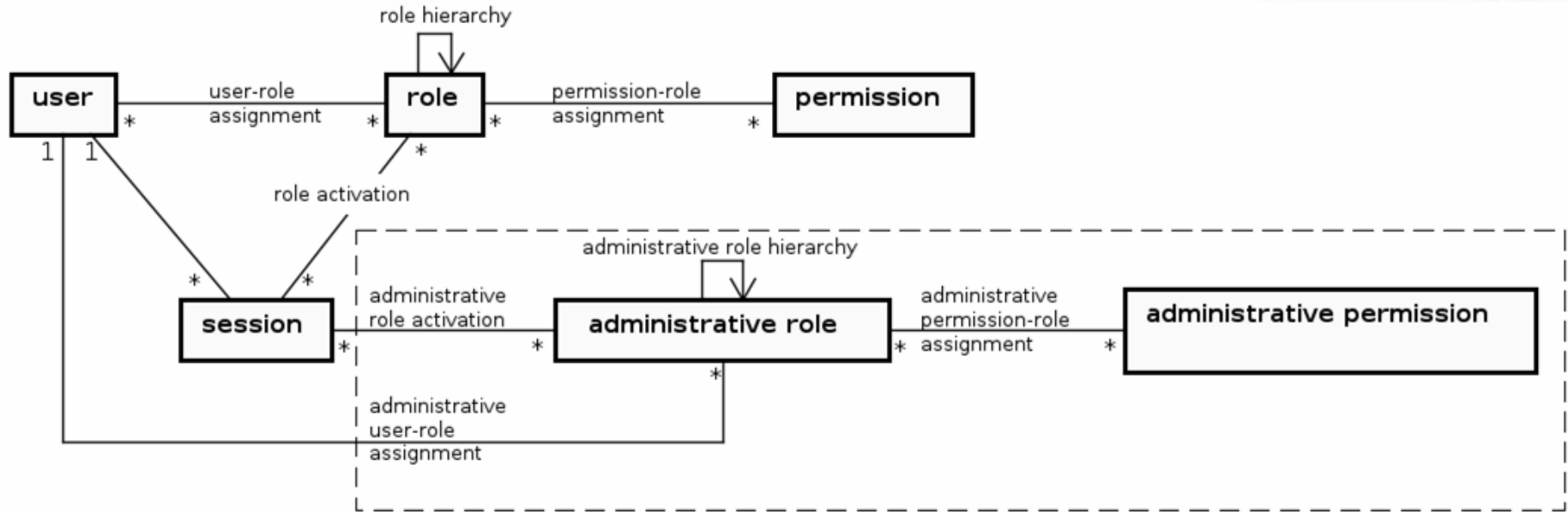
[4] ENDO, A. T.; SIMAO, A. Evaluating test suite characteristics, cost, and effectiveness of fsm based testing methods. Information and Software Technology, v. 55, n. 6, p. 1045 – 1062, 2013.

[5] BROU, M.; JONSSON, B.; KATOEN, J.-P.; LEUCKER, M.; PRETSCHNER, A. Model-Based Testing of Reactive Systems: Advanced Lectures (Lecture Notes in Computer Science).

# Role-Based Access Control

(RBAC model)

- **RBAC: *Users* receive *privileges* through *role assignments***



ANSI RBAC and Administrative RBAC models [6]



# Role-Based Access Control

(RBAC constraints)

- RBAC constraints <sup>[6]</sup>
  - **Cardinality constraints**
    - “There are **at most two** users **assigned to Admin role**”
  - **Separation of duty (SoD) constraints**
    - “Users cannot be **author and reviewer simultaneously**”

# Role-Based Access Control

(FSM-Based Testing of RBAC systems)<sup>[1]</sup>

---

```
1 U = {u1,u2}
2 R = {r1}
3 Pr = {pr1,pr2}
4 UR = {(u1,r1)}
5 PR = {(r1,pr1), (r1,pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

---



# Role-Based Access Control

(FSM-Based Testing of RBAC systems)<sup>[1]</sup>

---

```
1 U = {u1, u2}
```

← Users

```
2 R = {r1}
```

```
3 Pr = {pr1, pr2}
```

```
4 UR = {(u1, r1)}
```

```
5 PR = {(r1, pr1), (r1, pr2)}
```

```
6 Su(u1) = Su(u2) = 1
```

```
7 Du(u1) = Du(u2) = 1
```

```
8 Sr(r1) = 2
```

```
9 Dr(r1) = 1
```

---


# Role-Based Access Control

(FSM-Based Testing of RBAC systems)<sup>[1]</sup>

---

```
1 U = {u1,u2}
2 R = {r1}
3 Pr = {pr1,pr2}
4 UR = {(u1,r1)}
5 PR = {(r1,pr1), (r1,pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

---



# Role-Based Access Control

(FSM-Based Testing of RBAC systems) <sup>[1]</sup>

---

```
1 U = {u1,u2}
2 R = {r1}
3 Pr = {pr1,pr2}
4 UR = {(u1,r1)}
5 PR = {(r1,pr1), (r1,pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

---

Permissions

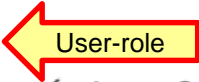
# Role-Based Access Control

(FSM-Based Testing of RBAC systems) <sup>[1]</sup>

---

```
1 U = {u1, u2}
2 R = {r1}
3 Pr = {pr1, pr2}
4 UR = {(u1, r1)}
5 PR = {(r1, pr1), (r1, pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

---



# Role-Based Access Control

(FSM-Based Testing of RBAC systems)<sup>[1]</sup>

---

```
1 U = {u1, u2}
2 R = {r1}
3 Pr = {pr1, pr2}
4 UR = {(u1, r1)}
5 PR = {(r1, pr1), (r1, pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

---

← Permission-role



# Role-Based Access Control

(FSM-Based Testing of RBAC systems) <sup>[1]</sup>

---

```
1 U = {u1, u2}
2 R = {r1}
3 Pr = {pr1, pr2}
4 UR = {(u1, r1)}
5 PR = {(r1, pr1), (r1, pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

---

**User  
Cardinality**



# Role-Based Access Control

(FSM-Based Testing of RBAC systems)<sup>[1]</sup>

---

```
1 U = {u1, u2}
2 R = {r1}
3 Pr = {pr1, pr2}
4 UR = {(u1, r1)}
5 PR = {(r1, pr1), (r1, pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

**Role Cardinality**

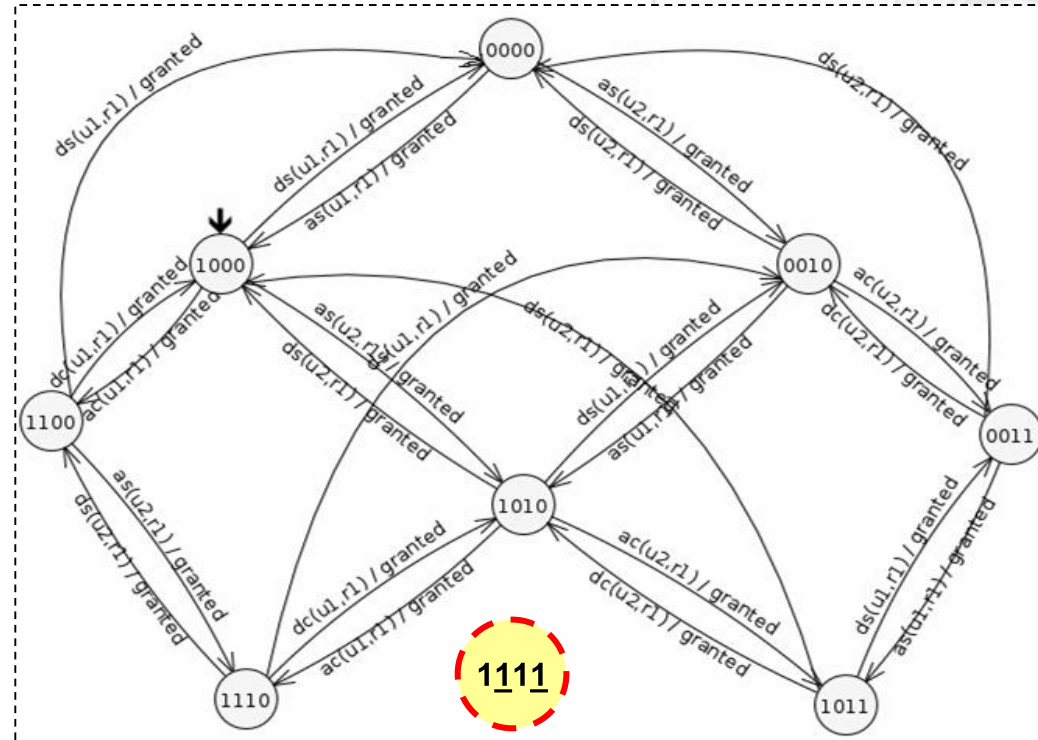
---

# Role-Based Access Control

(FSM-Based Testing of RBAC systems)<sup>[1]</sup>

```
1 U = {u1, u2}
2 R = {r1}
3 Pr = {pr1, pr2}
4 UR = {(u1, r1)}
5 PR = {(r1, pr1), (r1, pr2)}
6 Su(u1) = Su(u2) = 1
7 Du(u1) = Du(u2) = 1
8 Sr(r1) = 2
9 Dr(r1) = 1
```

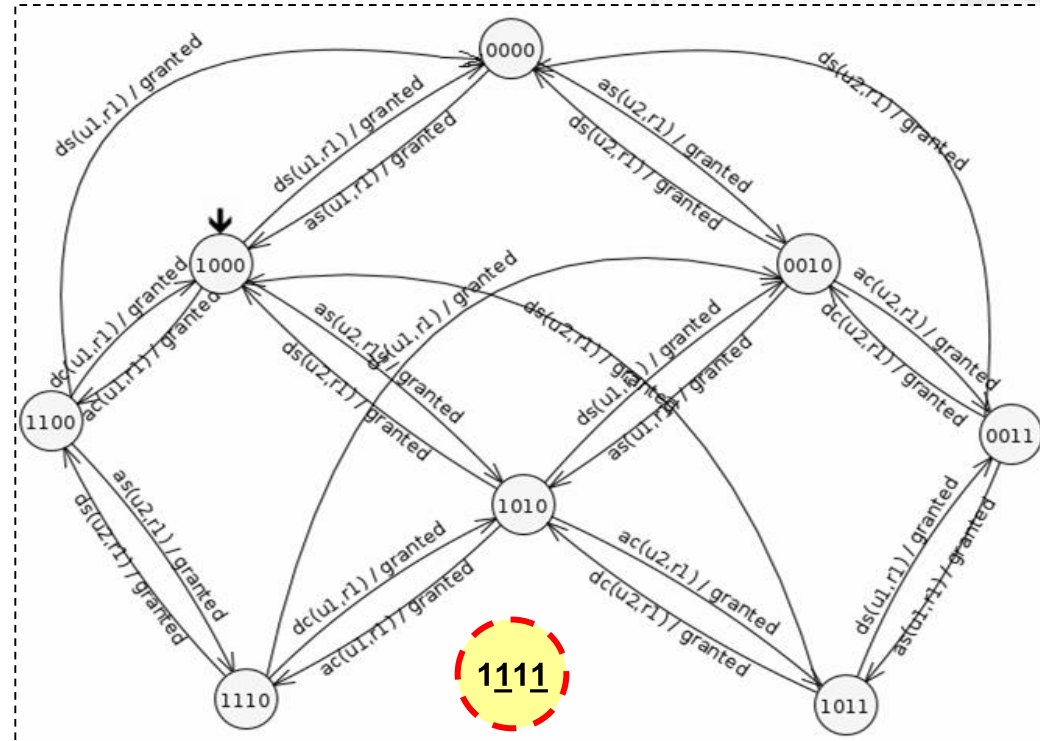
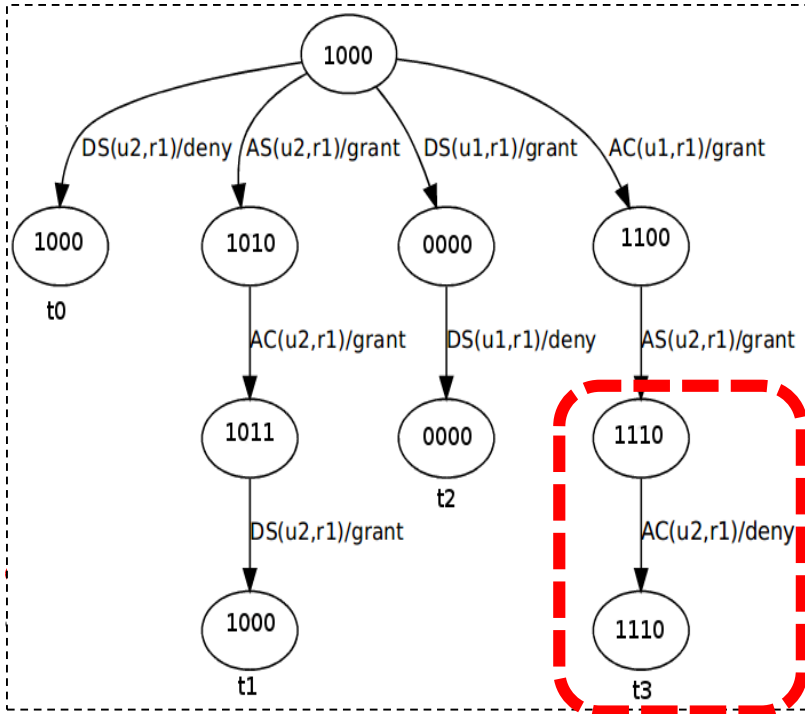
1000



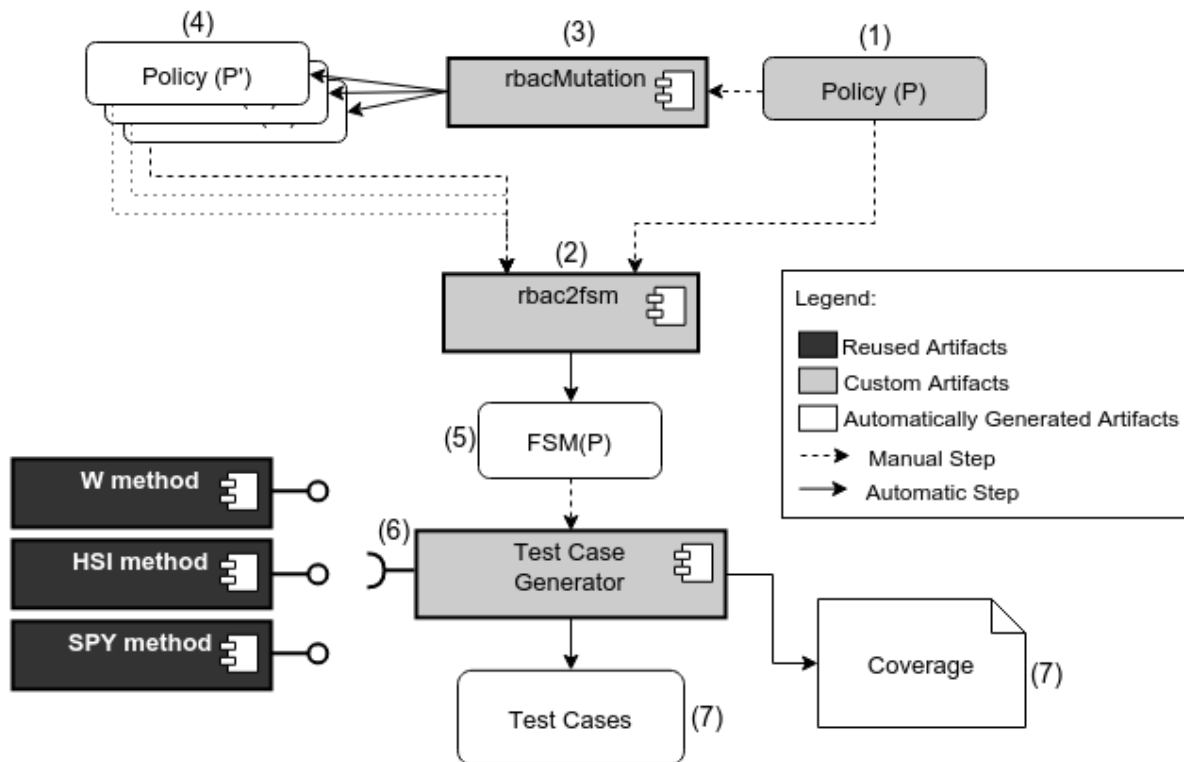
1111

# Role-Based Access Control

(FSM-Based Testing of RBAC systems) <sup>[1]</sup>



# Experimental Framework



# Experimental Study

(Selection of RBAC policies)

Table 2: FSMs and mutants generated from policies

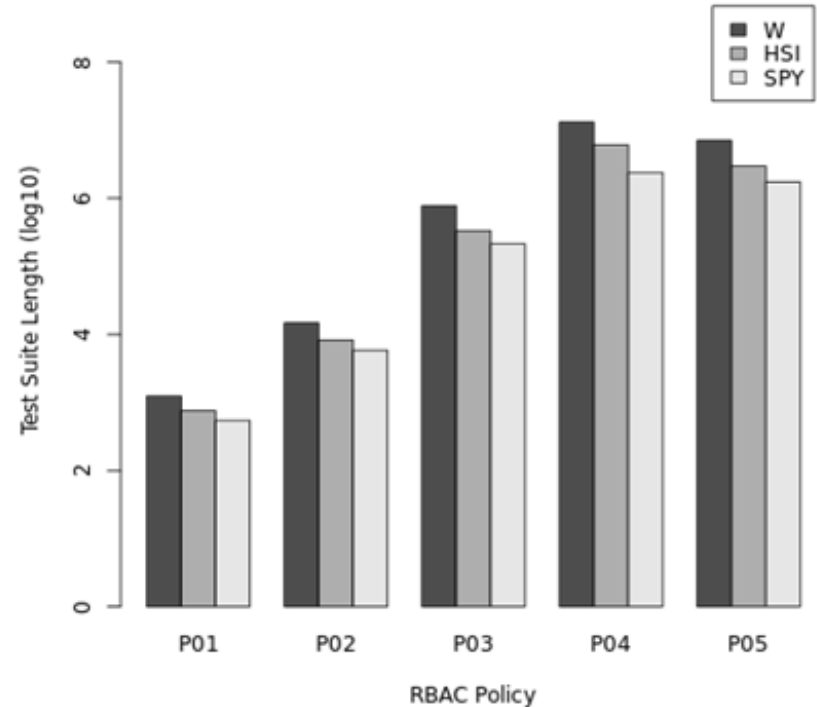
Alias	Policy name	States	Transitions	Mutants
P01	01_Masood2010Example1	8	64	9
P02	02_SeniorTraineeDoctor	21	336	17
P03	03_ExperiencePointsv2	203	6496	11
P04	04_users11roles2v2	485	42680	28
P05	05_Masood2009P2v2	857	34280	48

**15 test scenarios:  $\{W, HSI, SPY\} \times \{P01, P02, P03, P04, P05\}$**

# Analysis of Results

(Test Suite Length)

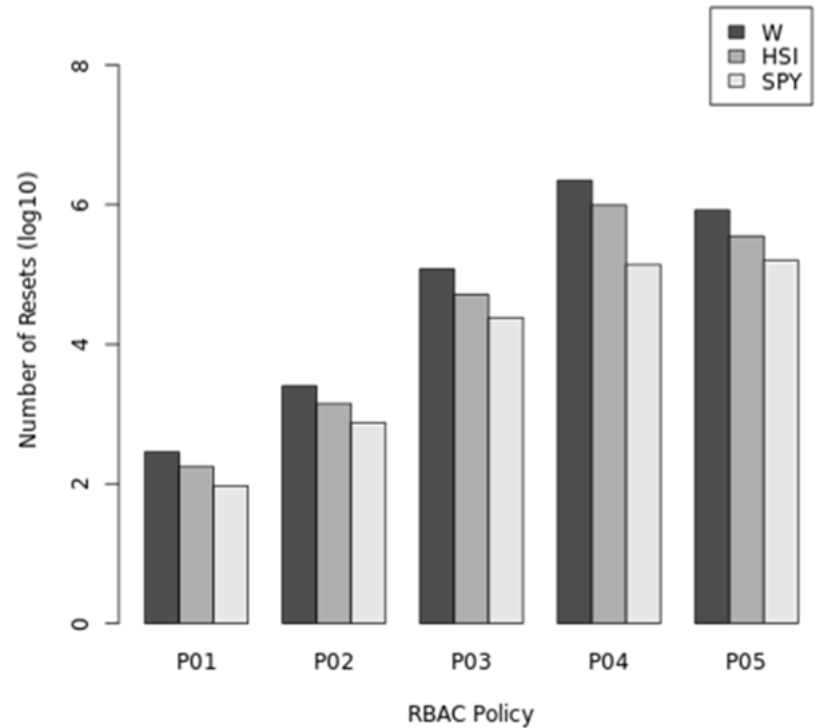
- Test generation duration
  - Total: 63 hours
  - *Min: 5 ms / Max: 24 h*
- **Strong positive correlation** [2]
  - $|Users| \times |Roles|$
- *SPY test suite length (average)*
  - **46%** of the *W* and *HSI* length
  - **18%** of the *W* length



# Analysis of Results

(Number of Resets)

- **Strong positive correlation** [2]
- SPY number of resets (average)
  - **42% of HSI resets**
    - Corroborate SPY's paper [7]
  - **22% of W resets**



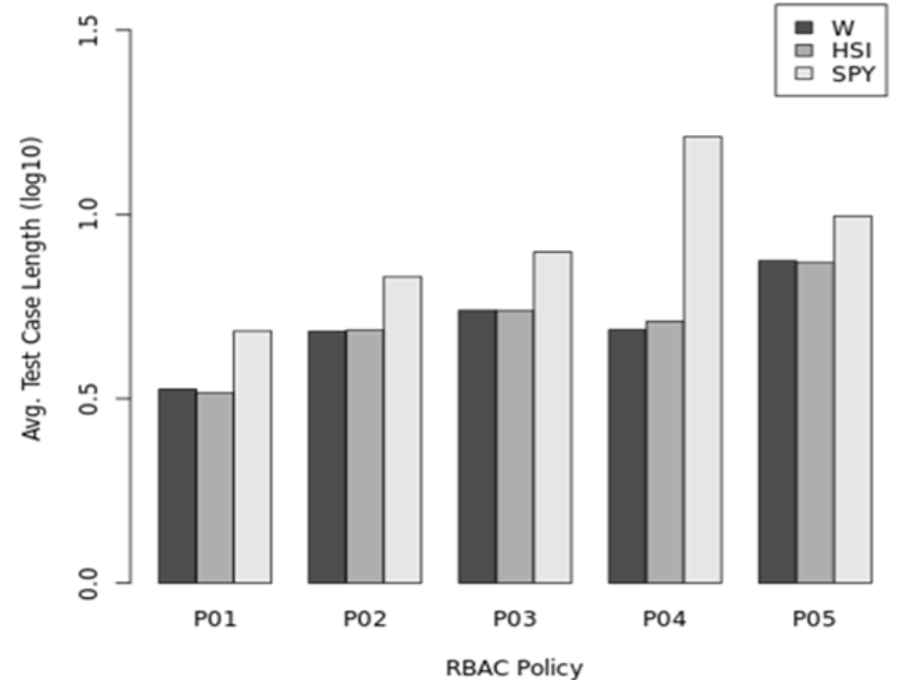
[2] ENDO, A. T.; SIMAO, A. Evaluating test suite characteristics, cost, and effectiveness of fsm based testing methods. Information and Software Technology, v. 55, n. 6, p. 1045 – 1062, 2013.

[7] SIMÃO, A.; PETRENKO, A.; YEVTUSHENKO, N. Generating reduced tests for fsms with extra states. In: NUNEZ, M.; BAKER, P.; MERAYO, M. (Ed.). Testing of Software and Communication Systems. Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5826). p. 129–145.

# Analysis of Results

(Average Test Case Length)

- **No negative correlation** [2]
- Average test case length
  - W and HSI were similar
  - SPY ~**78%** longer than {W, HSI}
- Maximum test case length
  - SPY was 14 times longer
- Test case length tends to increase
  - SPY





# Analysis of Results

(Test analysis)

- SPY method enabled **significant reduction** of the overall **test costs**
  - *Lower: Test Suite Length, Number of Resets*
  - *Greater: Test Case Length*
- 100% of effectiveness in all scenarios
  - *State and transition coverage* <sup>[1]</sup>
- Order of dominance:  $SPY > HSI > W$ 
  - *A large amount of test cases tends to be generated on RBAC domain*

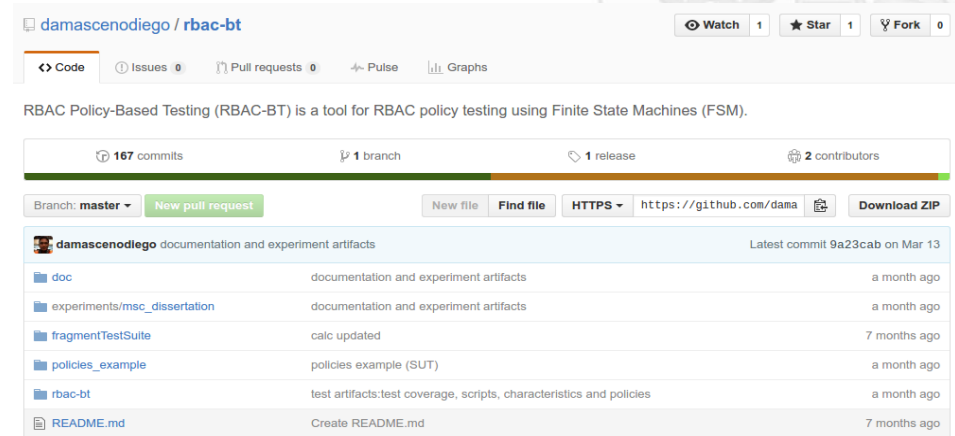
# Conclusion

- **RBAC testing** tends to be **costly**, **regardless** the **testing method**
- **SPY method** can be **more adequate** to RBAC testing
  - *Less resets (test cases)*
  - Shorter test suites
  - *Longer test cases*
- Fault detection **effectiveness** does not change (**100%**)
- Outcomes **corroborate** some previous results



# Future work

- **RBAC-BT:** <https://github.com/damascenodiego/rbac-bt>
- Extending RBAC-BT with *Hierarchical RBAC*
- Investigate RBAC test criteria
  - Test generation
  - Test selection/prioritization



# Thank you!

Carlos Diego Nascimento Damasceno, PhD Candidate – [damascenodiego@usp.br](mailto:damascenodiego@usp.br)

Advisor: Prof. Dr. Adenilso da Silva Simão

Laboratory of Software Engineering – LabES

Institute of Mathematics and Computer Science – ICMC

University of Sao Paulo – USP

Sao Carlos – SP – Brazil

