

# Build a better life by translating problems to Datalog and Answer Set Programming(ASP)

**WU**  
WIRTSCHAFTS  
UNIVERSITÄT  
WIEN VIENNA  
UNIVERSITY OF  
ECONOMICS  
AND BUSINESS

Axel Polleres

Institute for Data, Process and Knowledge Management

28/11/2022 Invited Talk - TAASP 2022, Vienna, Austria

Social media: @AxelPolleres@wien.rocks

<http://polleres.net/>



# Why I love ASP (since over 20 years)...

- ***Intuitive***, understandable Problem ***encodings...***`
- ... easily ***extensible***
- the beauty of **Guess and Check** to solve complex problems on top

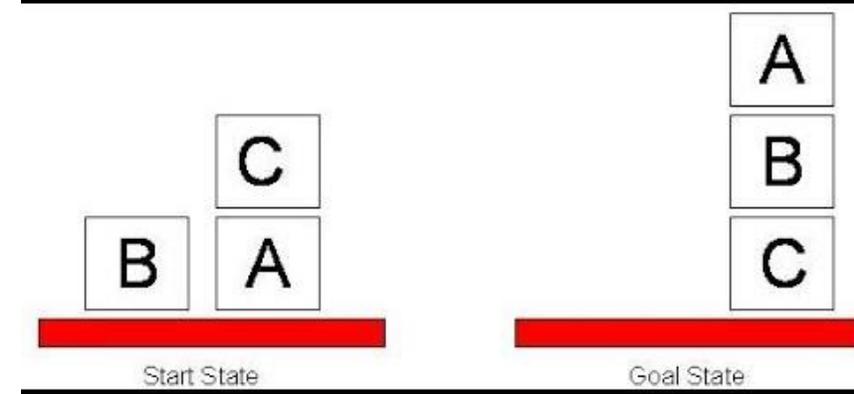
***Station 1:***

1999-2003



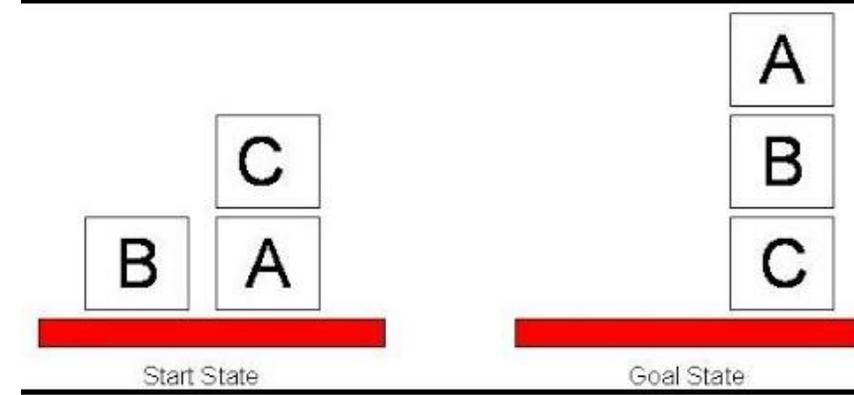
AI Planning

# AI Planning...



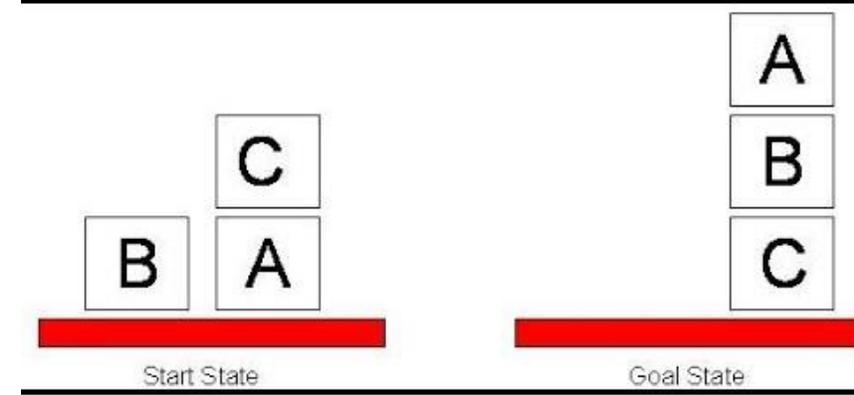
- fluents: `on(c, a)` , `on(b, table)` , `on(a, table)`
- Actions: `move(c, table)` with
  - preconditions:
    - `executable move(B, L) if block(B).`
    - `nonexecutable move(B, B).`
    - `nonexecutable move(B, L) if blocked(B).`
    - `nonexecutable move(B, L) if blocked(L).`
  - effects:
    - `caused on(B, L) after move(B, L).`
    - `caused -on(B, L) after move(B, L1) , on(B, L) , L1<>L.`

# AI Planning... intuitive encoding



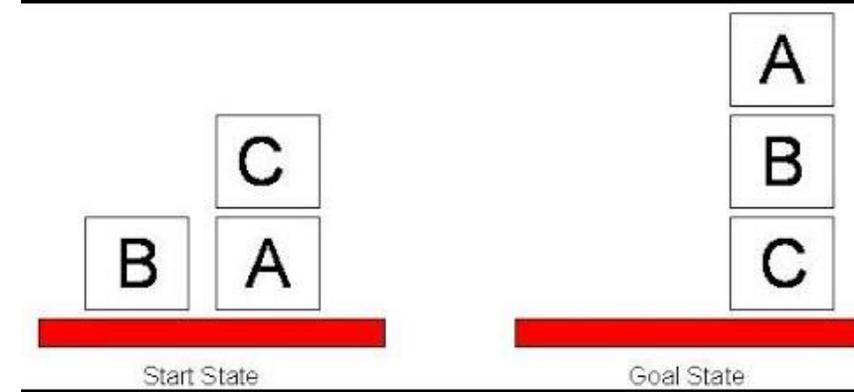
- fluents: `on(c, a, 0) . on(b, table, 0) . on(a, table, 0) .`
- Actions: `move(c, table, T) .`
  - preconditions:
    - `:- move(B, B, T) .`
    - `:- move(B, L, T) , blocked(B, T) .`
    - `:- move(B, L, T) , blocked(L, T) .`
  - effects:
    - `on(B, L, T+1) :- move(B, L, T).`
    - `-on(B, L, T+1) :- move(B, L1, T) , on(B, L, T) , L1<>L.`
  - implicit background knowledge:
    - `blocked(B, T) :- on(_, B, T) , block(B) .`
    - `on(B, L, T+1) :- on(B, L, T) , not -on(B, L, T) .`

# AI Planning... intuitive encoding



- fluents: `on(c, a, 0) . on(b, table, 0) . on(a, table, 0) .`
- Actions: `move(c, table, T) .`
  - preconditions:
    - `:- move(B, B, T) .`
    - `:- move(B, L, T) , blocked(B, T) .`
    - `:- move(B, L) , blocked(L, T) .`
  - effects:
    - `on(B, L, T+1) :- move(B, L, T).`
    - `-on(B, L, T+1) :- move(B, L1, T) , on(B, L, T) , L1<>L.`
  - implicit background knowledge:
    - `blocked(B, T) :- on(_, B, T) , block(B) .`
    - `on(B, L, T+1) :- on(B, L, T) , not -on(B, L, T) .`

# AI Planning... guess and check:



- fluents: `on(c, a, 0) . on(b, table, 0) . on(a, table, 0) .`
- Actions: `move(c, table, T) .`
  - preconditions:

```
move(B, L, T) v -move(B, L, T) :- block(B), location(L), time(T) .  
:- move(B, B, T) .  
:- move(B, L, T) , blocked(B, T) .      blocked(B, T) :- on(_, B, T) .  
:- move(B, L, T) , blocked(L, T) .
```

- effects:

```
on(B, L, T+1) :- move(B, L, T) .  
-on(B, L, T+1) :- move(B, L1, T) , on(B, L, T) , L1<>L .
```

- implicit background knowledge:

```
blocked(B, T) :- on(B1, B, T) , block(B) .  
on(B, L, T+1) :- on(B, L, T) , not -on(B, L, T) .
```

## Goal:

```
goal(T) :- on(a, b) ,  
           on(b, c) ,  
           on(c, table) .  
:- not goal(maxTime) .  
time(0..maxTime) .
```

# AI Planning... ease of extensions:

- Nondeterministic actions:

- E.g., “clumsy” move

- $\text{on}(B, L, T+1) \vee \text{on}(B, \text{table}, T+1) :- \text{cmove}(B, L, T), \text{on}(B, L, T), \text{block}(L).$
    - $\text{on}(B, L, T+1) :- \text{cmove}(B, L, T), \text{on}(B, \text{table}, T).$

- Action costs /cost optimal planning:

- $\text{cost}(T, 1) :- \text{cmove}(B, L, T).$

- $\text{cost}(T, 2) :- \text{move}(B, L, T).$

- $:\sim \text{cost}(T, \text{Cost}). [\text{Cost}:]$

# AI Planning

ARTICLE

## Answer set planning under action costs

Authors:  Thomas Eiter,  Wolfgang Faber,  Nicola Leone,  Gerald Pfeifer,  Axel Polleres [Authors Info & Claims](#)

Journal of Artificial Intelligence Research, Volume 19, Issue 1 • July 2003 •

“crumsy” move  
 $\text{on}(B, L, T+1) \vee \text{on}(B,$   
 $\text{on}(B, L, T+1) :-$

## Action costs / cost optimization

ARTICLE

## Towards automated integration of guess and check programs in answer set programming: a meta-interpreter and applications

Authors:  Thomas Eiter,  Axel Polleres [Authors Info & Claims](#)

Theory and Practice of Logic Programming, Volume 6, Issue 1-2 • January 2006 • pp 23-60 • <https://doi.org/10.1017/S1471068405002577>

ARTICLE

## A logic programming approach to knowledge-state planning: Semantics and complexity

Authors:  Thomas Eiter,  Wolfgang Faber,  Nicola Leone,  Gerald Pfeifer,  Axel Polleres [Authors Info & Claims](#)

ACM Transactions on Computational Logic, Volume 5, Issue 2 • April 2004 • pp 206-263 • <https://doi.org/10.1145/976706.976708>

Knowledge-state planning:

# Why I love ASP (since over 20 years)...

- ***Intuitive***, understandable Problem ***encodings...***
- ... easily ***extensible***
- the beauty of **Guess and Check** to solve complex problems on top

## **Part 1:**

1999-2003



AI Planning

## **Part 2:**

2003 – to date...



Semantic Web



Universidad  
Rey Juan Carlos



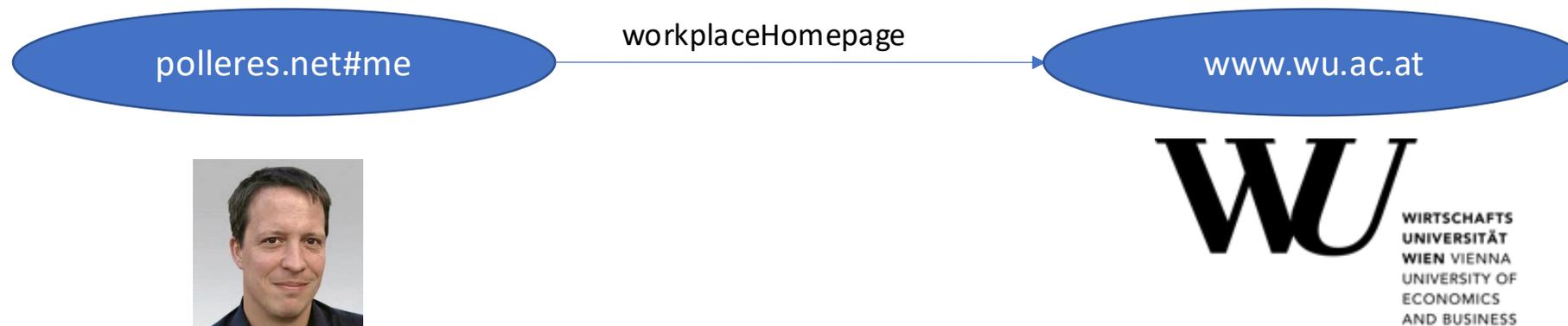
NUI Galway  
OÉ Gaillimh

# Semantic Web ... RDF Triples

- “Typed” links on the Web ...

`<http://www.polleres.net#me>` `<http://xmlns.com/foaf/0.1/workplaceHomepage>` `<http://www.wu.ac.at>` .

- ... can be seen as subject-predicate-object edges in a Graph:



# Semantic Web ... Standards like RDF have lead to (really) big Open “Knowledge Graphs” ...

- ... available on the Web
- ... *queryable via a query language called SPARQL!*



[1,101,215,718](#) triples/edges



[13,602,048,837](#) triples/edges

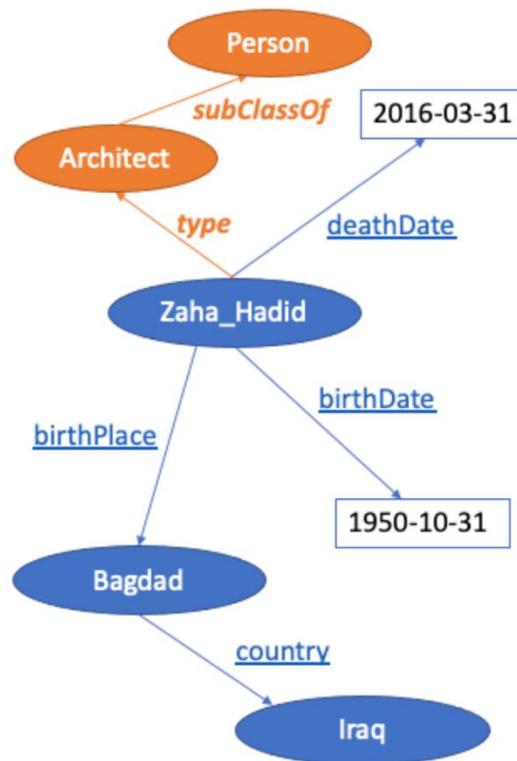
# Semantic Web ... Standards like RDF have lead to (really) big Open “Knowledge Graphs” ...

- ... available on the Web
- ... *queryable via a query language called SPARQL!*

[http://wikipedia.org/wiki/Zaha\\_Hadid](http://wikipedia.org/wiki/Zaha_Hadid)

<b>Born</b>	Zaha Mohammad Hadid 31 October 1950 Baghdad, Kingdom of Iraq
<b>Died</b>	31 March 2016 (aged 65) Miami, Florida, U.S.
<b>Nationality</b>	Iraq, United Kingdom
<b>Alma mater</b>	American University of Beirut Architectural Association School of Architecture
<b>Occupation</b>	Architect
<b>Parent(s)</b>	Mohammed Hadid Wajeeha Sabonji
<b>Practice</b>	Zaha Hadid Architects
<b>Buildings</b>	Vitra Fire Station, MAXXI, Bridge Pavilion, Contemporary Arts Center, Heydar Aliyev Center, Riverside Museum
<b>Website</b>	<a href="http://www.zaha-hadid.com">www.zaha-hadid.com</a>

[http://dbpedia.org/resource/Zaha\\_Hadid](http://dbpedia.org/resource/Zaha_Hadid)



PREFIX dbr: <<http://dbpedia.org/resource/>>  
 PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>  
 PREFIX dbo: <<http://dbpedia.org/ontology/>>  
 PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

```
SELECT ?N WHERE {
  ?X rdf:type dbo:Architect ;
  rdfs:label ?N .
  {{?X dbo:birthPlace dbr:Baghdad}
  UNION
  {?X dbo:birthPlace dbr:Vienna}}
```

```
FILTER (lang(?N) = "en")
```

```
}
```

# Semantic Web ...Status ~2006

- Semantics of SPARQL in parts undefined
- Various extensions being discussed...



SPARQL Query Language for RDF

W3C Recommendation 15 January 2008

# SPARQL... intuitive encoding



SPARQL Query Language for RDF

W3C Recommendation 15 January 2008

- Why I liked SPARQL?

```
SELECT ?X
WHERE {
  ?X rdf:type dbpedia:Architect.
  ?X dbpedia:birthPlace dbpedia:Baghdad .
}
```

for the same reason I love ASP! *Obvious similarities to Datalog...*

```
answer(X) :-
  triple( X, birthPlace , baghdad ) ,
  triple( X, type , architect) .
```

# SPARQL... intuitive encoding some not entirely trivial, e.g. OPTIONAL:

*Give me people who know somebody and OPTIONALLY their email address:*

```
triple( :tim, knows, :jim ) .   triple(:tim, email, timbl@w3.org ) .
triple( :jim, knows, :tim ) .
```

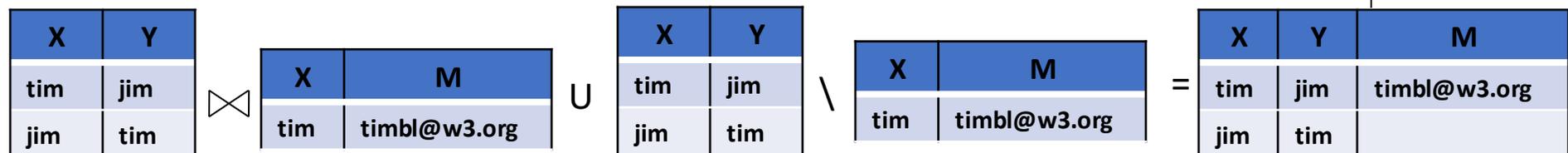
## Example Query:

```
answer(X,M) :- evalP(X,Y,M) .
```

```
evalP(X,Y,M) :- triple( X, knows, Y ) , triple( X, email, M ) .
```

```
evalP(X,Y,null) :- triple( X, knows, Y ) , not evalP1(X) .
```

```
evalP1(X) :- triple( X, email, M ) .
```



# SPARQL... intuitive encoding

[Polleres, 2007] shows that all of SPARQL 1.0 can be translated to (safe) nonrecursive Datalog<sup>not</sup>.

ARTICLE     

---

## From SPARQL to rules (and back)

---

Author:  [Axel Polleres](#) [Authors Info & Claims](#)

---

WWW '07: Proceedings of the 16th international conference on World Wide Web • May 2007 • Pages 787–796 • <https://doi.org/10.1145/1242572.1242679>

[Angles&Gutierrez 2008] vice versa show that (safe) nonrecursive Datalog<sup>not</sup> likewise be encoded into SPARQL.

ARTICLE     

---

## The Expressive Power of SPARQL

---

Authors:  [Renzo Angles](#),  [Claudio Gutierrez](#) [Authors Info & Claims](#)

---

ISWC '08: Proceedings of the 7th International Conference on The Semantic Web • October 2008 • Pages 114–129 • [https://doi.org/10.1007/978-3-540-88564-1\\_8](https://doi.org/10.1007/978-3-540-88564-1_8)

# SPARQL... ASP's ease of extensions:

1) We could show that additional features of SPARQL 1.1 were also easily encodable in Datalog...

Articles

## On the relation between SPARQL1.1 and Answer Set Programming

Axel Polleres  & Johannes Peter Wallner

Pages 159-212 | Published online: 24 Jun 2013

 Download citation  <https://doi.org/10.1080/11663081.2013.798992>

2) ... and (already before) proposed a semantics to using SPARQL as a rules language to define the semantics of RDF sources potentially mutually referring to each other...

ARTICLE     

## SPARQL++ for mapping between RDF vocabularies

Authors:  Axel Polleres,  François Scharffe,  Roman Schindlauer [Authors Info & Claims](#)

OTM'07: Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I • November 2007 • Pages 878–896

# SPARQL... ASP's ease of extensions:

ARTICLE



## SPARQL++ for mapping between RDF vocabularies

Authors: Axel Polleres, François Scharffe, Roman Schindlauer [Authors Info & Claims](#)

OTM'07: Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I • November 2007 • Pages 878–896

Unfortunately didn't make it to the standard ;-), but here's the idea:

Web source P1:

```
:tim :knows :jim .  
:tim :email <mailto:timbl@w3.org> .  
:jim :knows :tim .
```

```
CONSTRUCT {?s :email ?m }  
FROM P2  
WHERE {?s :email ?m }
```

Web source P2:

```
:jim <mailto:jim@rpi.edu> .
```

```
CONSTRUCT {jim :knows ?m }  
FROM P1  
WHERE {:jim :knows ?m }
```

# SPARQL... ASP's guess and check :

ARTICLE



## SPARQL++ for mapping between RDF vocabularies

Authors: Axel Polleres, François Scharffe, Roman Schindlauer [Authors Info & Claims](#)

OTM'07: Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I • November 2007 • Pages 878–896

Unfortunately didn't make it to the standard ;-), but here's the idea:

Web source P1:

```
:tim :knows :jim .
:tim :email <mailto:timbl@w3.org> .
:jim :knows :tim .
```

```
CONSTRUCT {?s :email ?m }
FROM P2
WHERE {?s ... OPTIONAL ... }
```

Web source P2:

```
:jim <mailto:jim@rpi.edu> .

CONSTRUCT {jim :knows ?m }
FROM P1
WHERE { :jim ... NOT EXISTS ... }
```

Could involve recursion and cycles over negation, elegantly solvable with ASP semantics!

# Semantic Web ... more recent ASP applications:

- SHACL: a language to formulate constraints over RDF graphs

## Shapes Constraint Language (SHACL)

W3C Recommendation 20 July 2017



- E.g.:

Shape:

```
:StudentShape a sh:NodeShape ;
  sh:targetNode :Ben ;
  sh : property [
    sh:path :enrolledIn ;
    sh : qualifiedMinCount 1 ;
    sh : qualifiedValueShape [
      sh:class :Course ] ] .
```

Each defined target node needs to fulfill the constraints, in this case:

There needs to be at least one Course that the student is enrolled in.

Data Graph:

```
: Ben :enrolledIn :C1 .
:C1 :type Course.
```

# SHACL ... computing repairs by Guess and check:

- Idea: Without going into details of the encoding: we encode repairs inspired by

**Database Repairing and Consistent Query Answering**  
**Synthesis Lectures on Data Management**

August 2011, 121 pages, (<https://doi.org/10.2200/S00379ED1V01Y201108DTM020>)

Leopoldo Bertossi  
Carleton University, Ottawa, Canada

# SHACL ... computing repairs by Guess and check:

- Idea: Without going into details of the encoding: we encode repairs

ARTICLE



## Repairing SHACL Constraint Violations Using Answer Set Programming

Authors: Shqiponja Ahmetaj, Robert David, Axel Polleres, Mantas Šimkus [Authors Info & Claims](#)

The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings • Oct 2022 • Pages 375–391 • [https://doi.org/10.1007/978-3-031-19433-7\\_22](https://doi.org/10.1007/978-3-031-19433-7_22)



### Data Graph:

```
: Ben :enrolledIn :C1 .  
:C1 :type Course.
```

vs.

```
:Ben :enrolledIn :C1.  
:Ben :enrolledIn :new1.  
:new1 :type Course.
```

### Guess a pair (A,D) of additions and deletions that repair all target nodes

- cardinality minimality
- strategy for introducing new nodes (for minimality constraints)
- relaxed encoding to repair a maximal number of target nodes
  - in the case not all target nodes can be repaired
- Implementation using Java and Clingo

# SHACL ... computing repairs by Guess and check:

Somewhat work in progress, since it is not 100% clear what a “good” repair semantics should look like, but – we hope – encoding this as a repair problem helps to *clarify the semantics of the standard*:

- encode more complex repair policies
  - *e.g. fix a part of the vocabulary/signature*
- similar issues as in SPARQL++ arise when you allow recursion (cf.
- raise a discussion about intuitive repairs, may need extension of the SHACL standard!

Active work in this space, we can built upon:

RESEARCH-ARTICLE



## Stable Model Semantics for Recursive SHACL

**Authors:** Medina Andresel, Julien Corman, Magdalena Ortiz, Juan L. Reutter, Ognjen Savkovic, Mantas Simkus [Authors Info & Claims](#)

WWW '20: Proceedings of The Web Conference 2020 • April 2020 • Pages 1570–1580 • <https://doi.org/10.1145/3366423.3380229>

KR Proceedings ▶ 2021 ▶ Full Papers ▶ Pages 12–21

## Reasoning about Explanations for Non-validation in SHACL

Shqiponja Ahmetaj (Vienna University of Economics and Business, Austria)  
 Robert David (Semantic Web Company, Austria)  
 Magdalena Ortiz (Technical University of Vienna, Austria)  
 Axel Polleres (Vienna University of Economics and Business, Austria, Complexity Science Hub Vienna, Austria)  
 Bojken Shehu (Polytechnic University of Tirana, Albania)  
 Mantas Šimkus (Technical University of Vienna, Austria)

# Why I love ASP (since over 20 years)...

- ***Intuitive***, understandable Problem ***encodings...***
- ... easily ***extensible***
- the beauty of **Guess and Check** to solve complex problems on top

**Part 1:**  
1999-2003



AI Planning

**Part 2:**  
2003 – to date...



Semantic Web



Universidad  
Rey Juan Carlos



NUI Galway  
OÉ Gaillimh

**Part 3:**  
ca. 2014 – 2022...



**SIEMENS**

Business Process  
Management

# Resource allocation in BPM

- Recently concluded PhD thesis:



Giray Havur



**SHAPE**

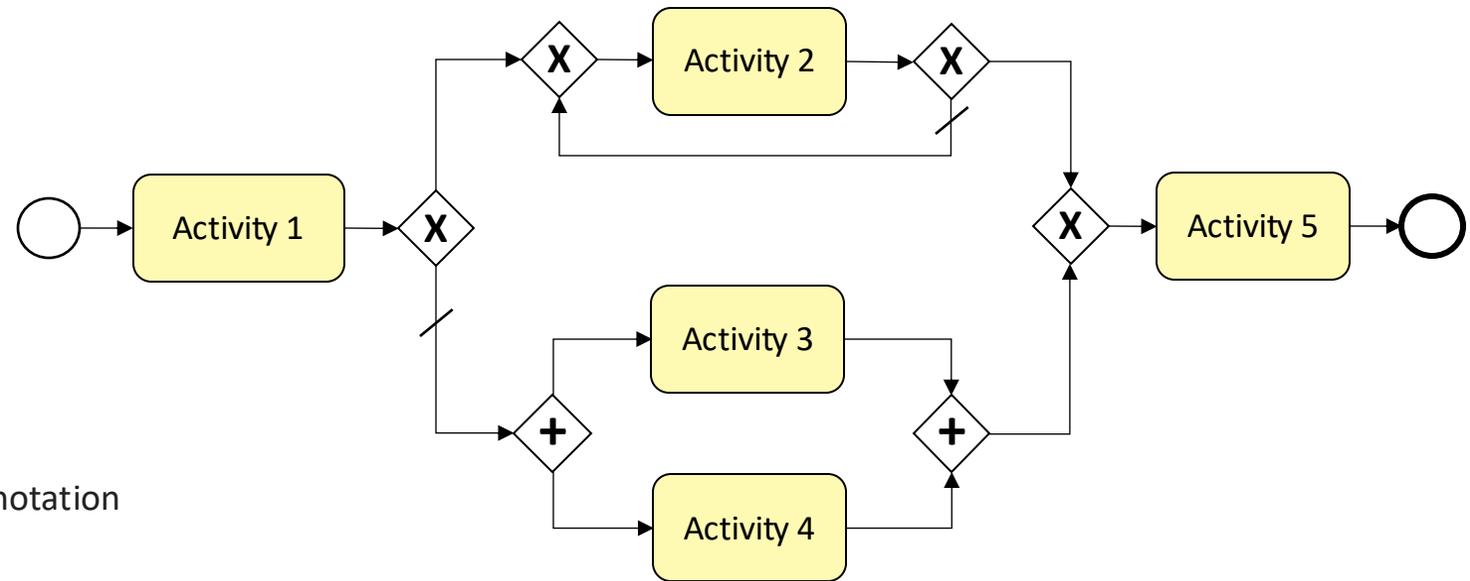
Safety-critical Human- and dAta-centric Process management in Engineering projects



Co-supervisor/Co-PI

# Organizing Work

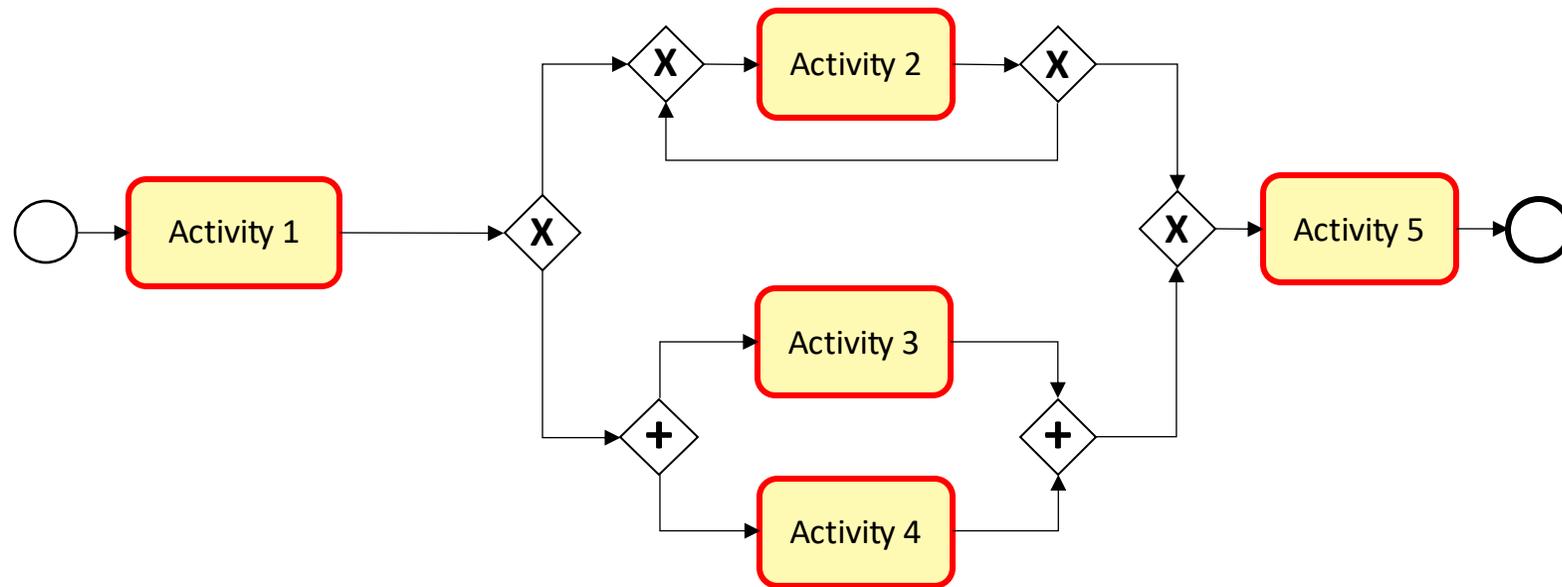
## Business Process Modelling Notation



- A standardized and popular graphical notation for specifying business processes

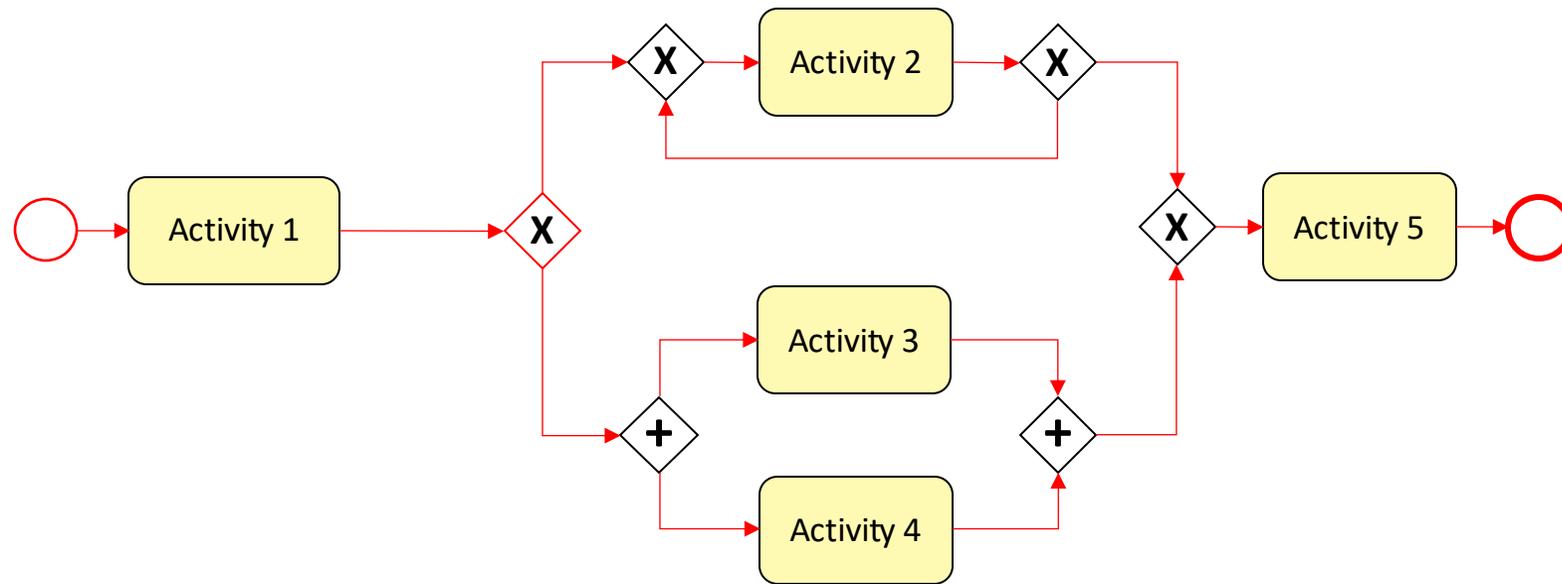
# Organizing Work

## Business Process Modelling Notation



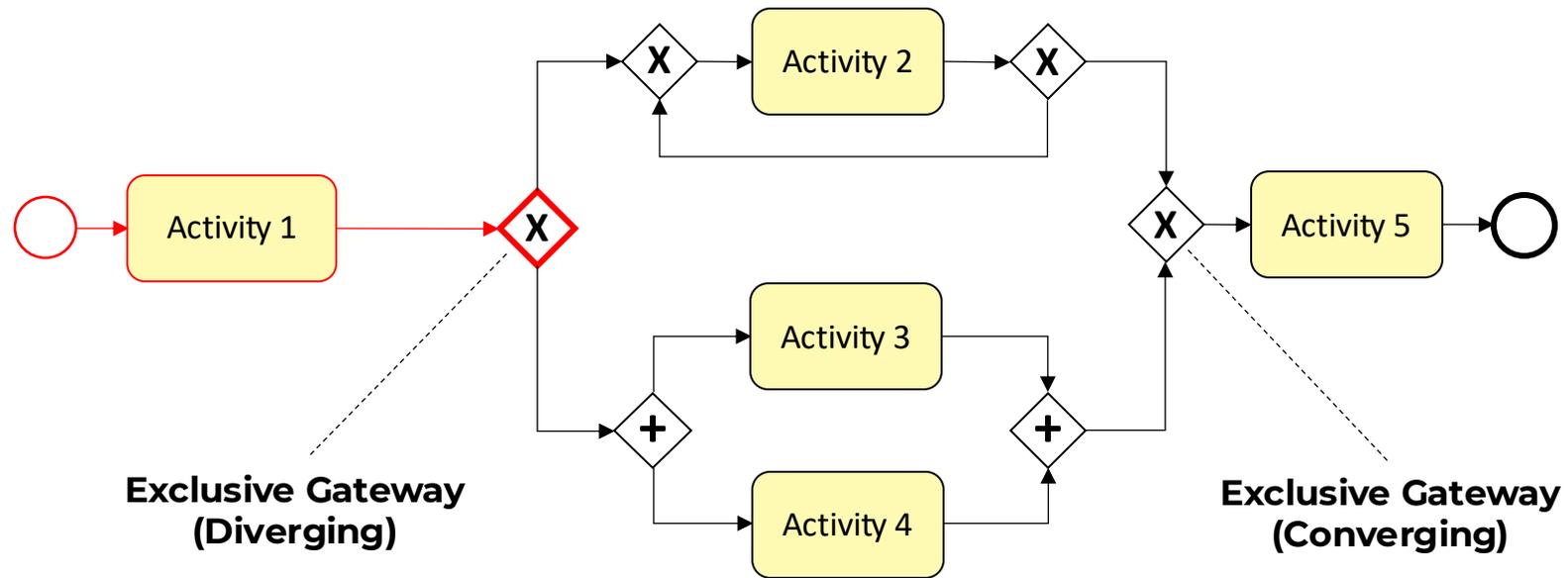
# Organizing Work

## Business Process Modelling Notation



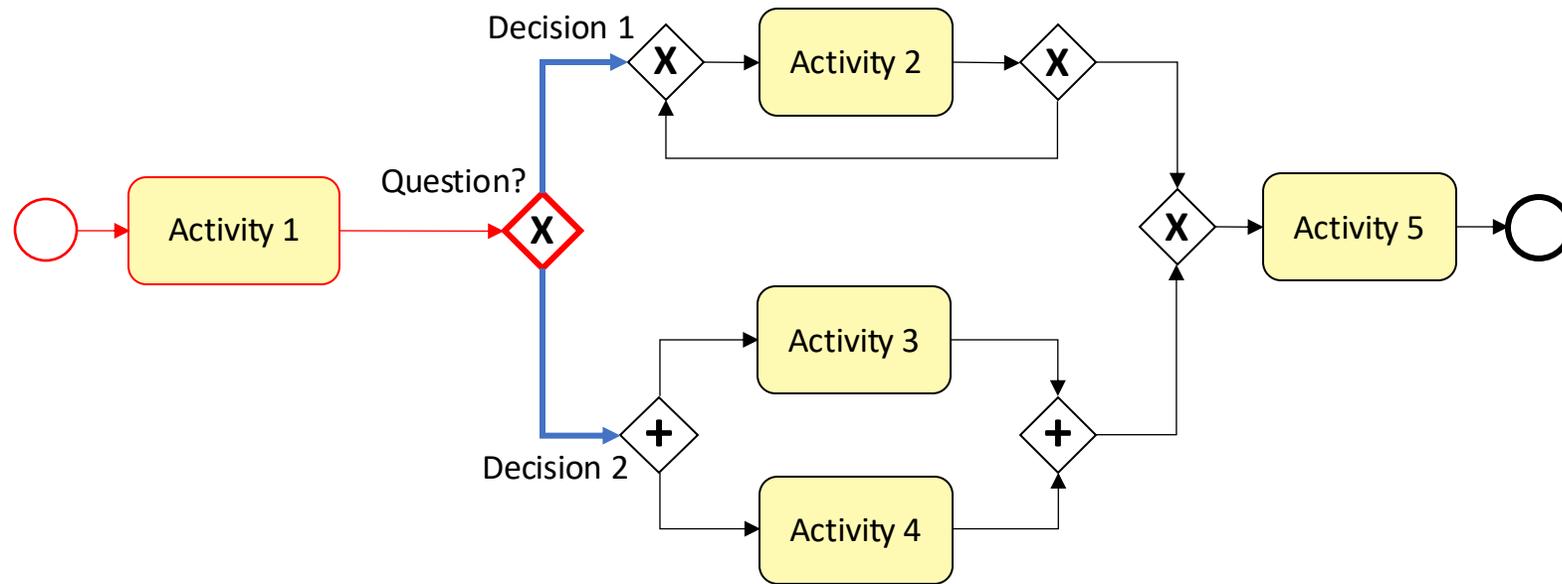
# Organizing Work

## Business Process Modelling Notation



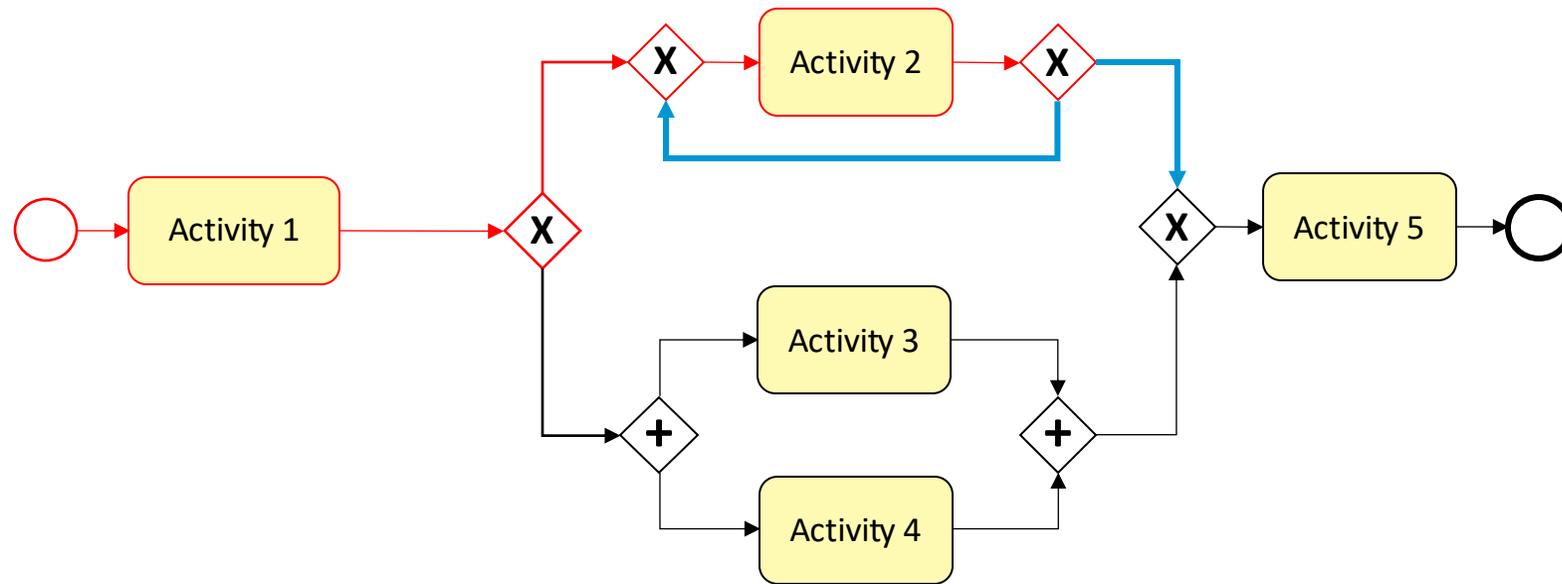
# Organizing Work

## Business Process Modelling Notation



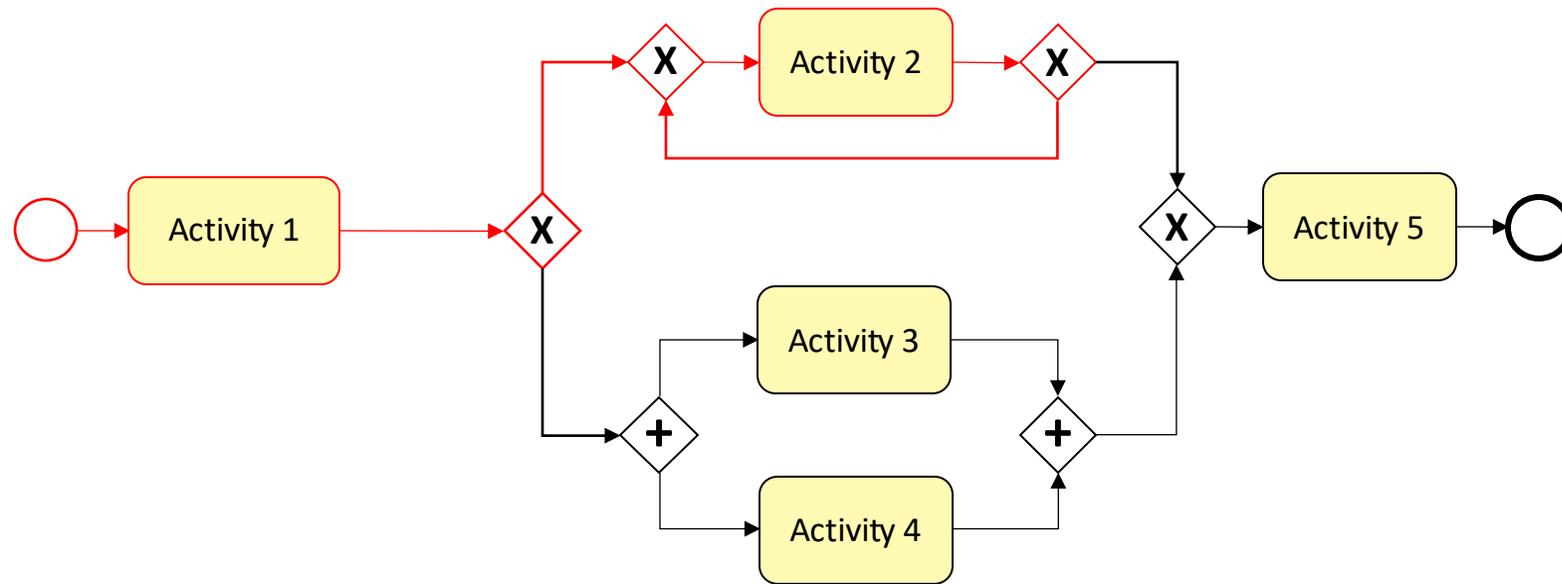
# Organizing Work

## Business Process Modelling Notation



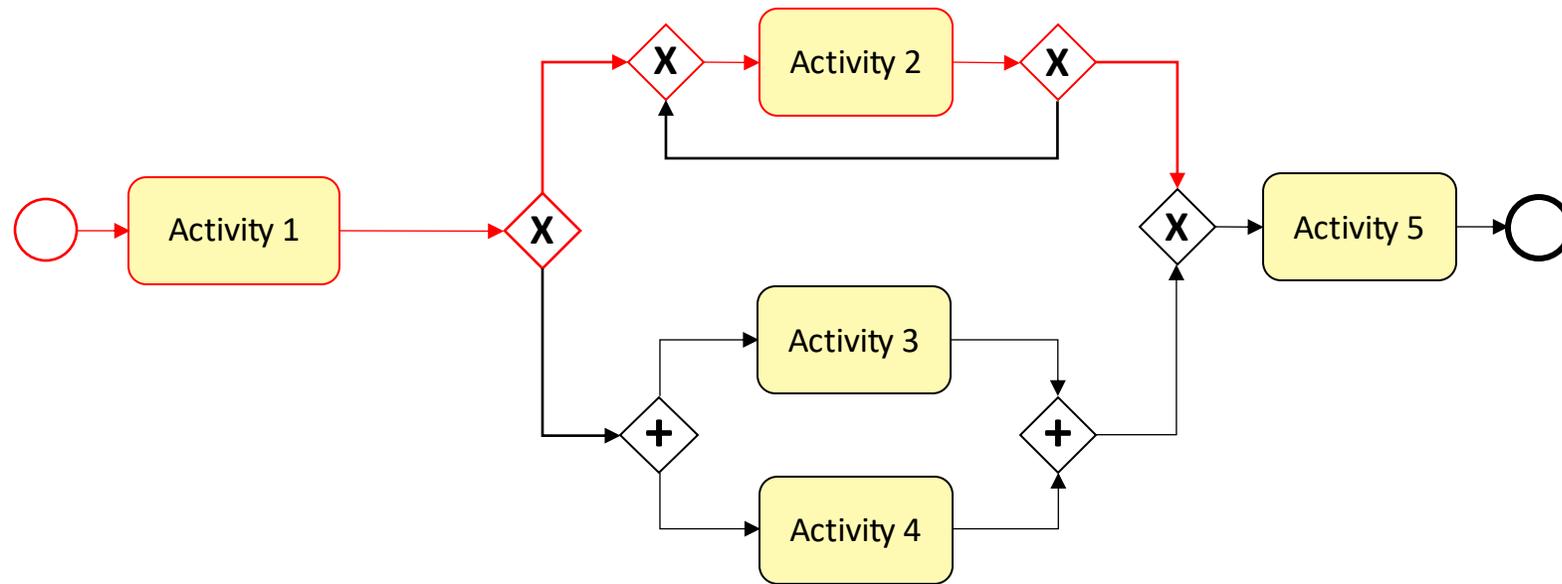
# Organizing Work

## Business Process Modelling Notation



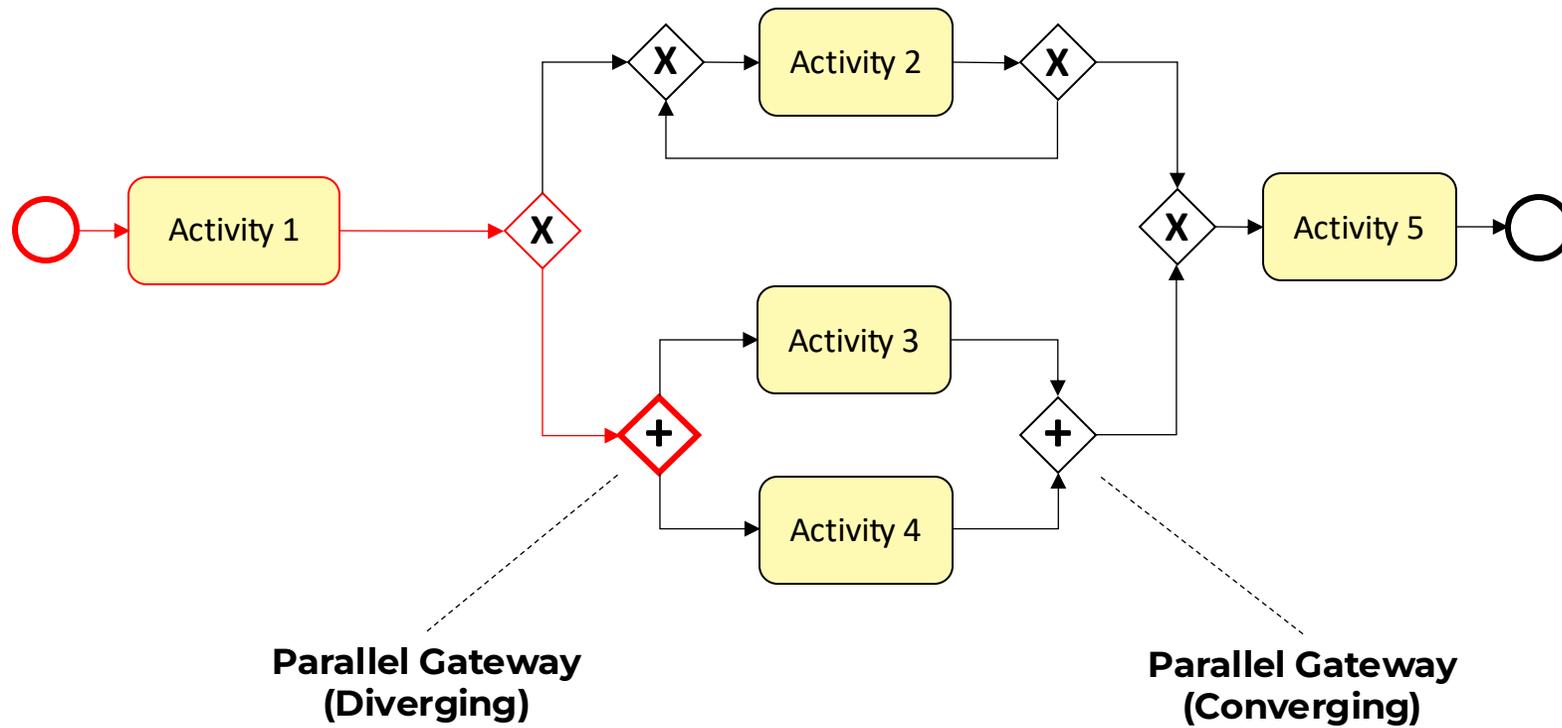
# Organizing Work

## Business Process Modelling Notation



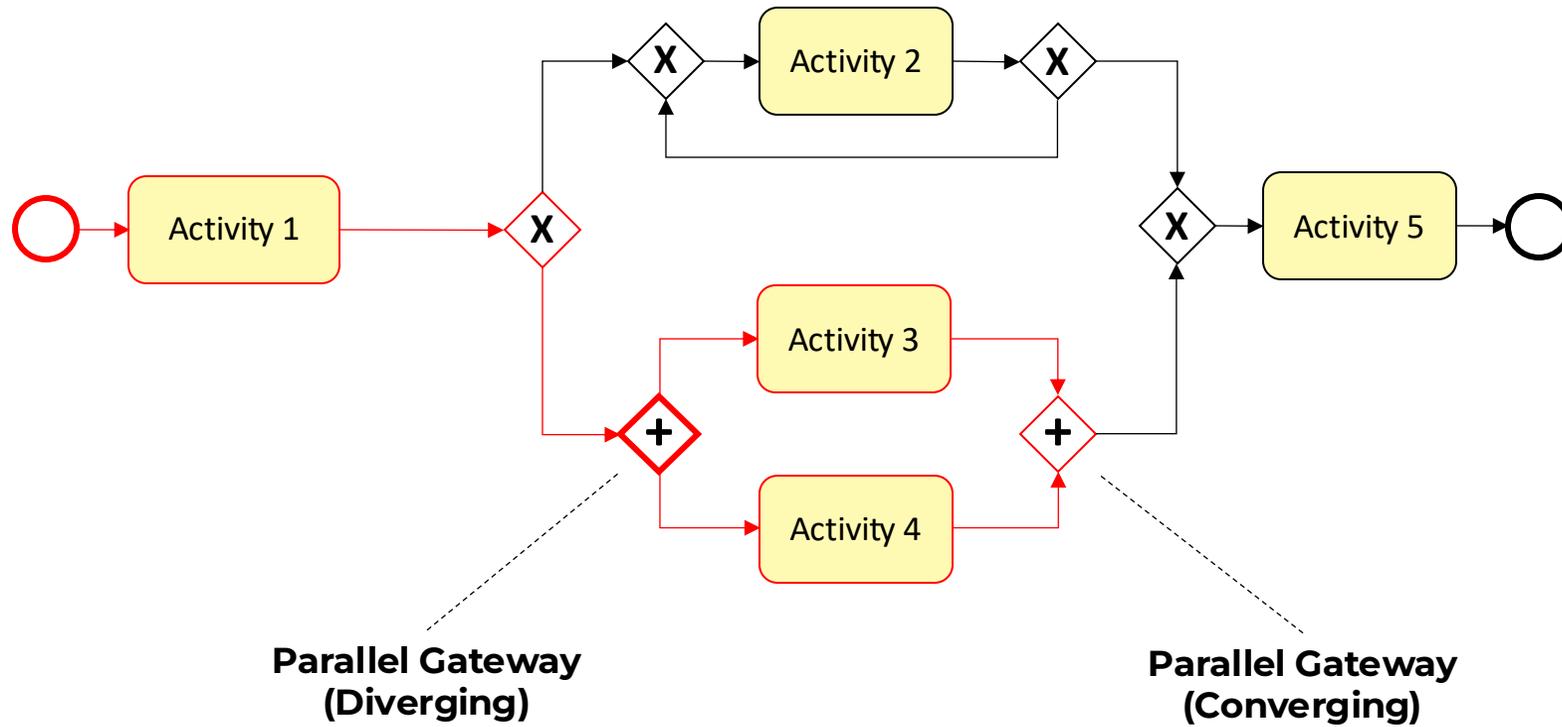
# Organizing Work

## Business Process Modelling Notation



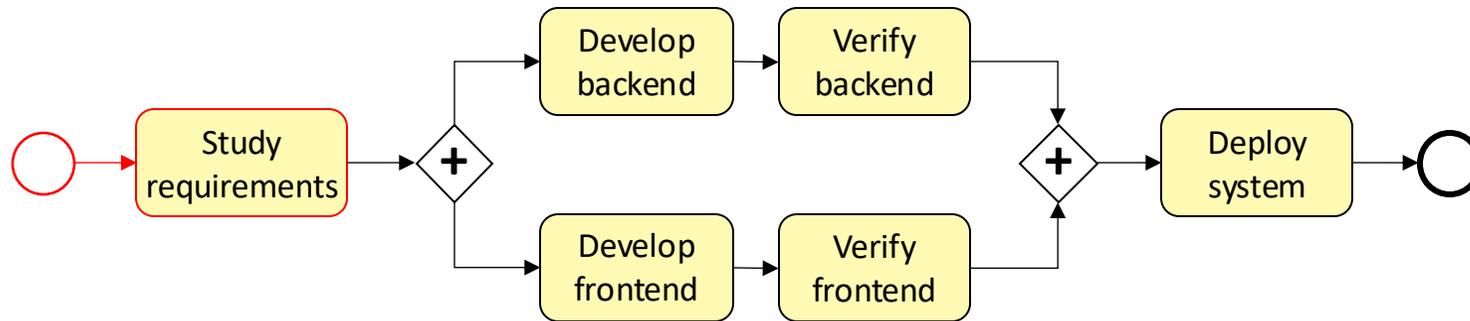
# Organizing Work

## Business Process Modelling Notation



# Organizing Work

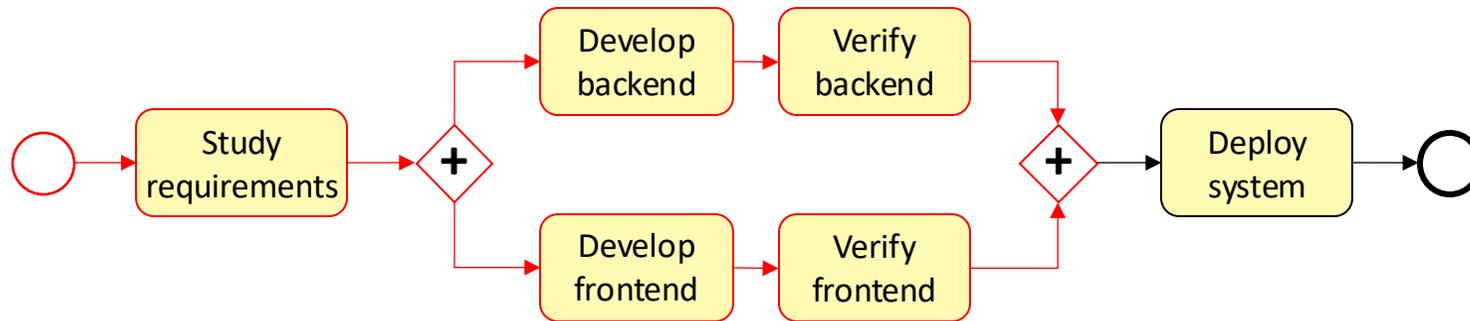
## BPMN Example



Process Model: A BPMN Model for Software Development

# Organizing Work

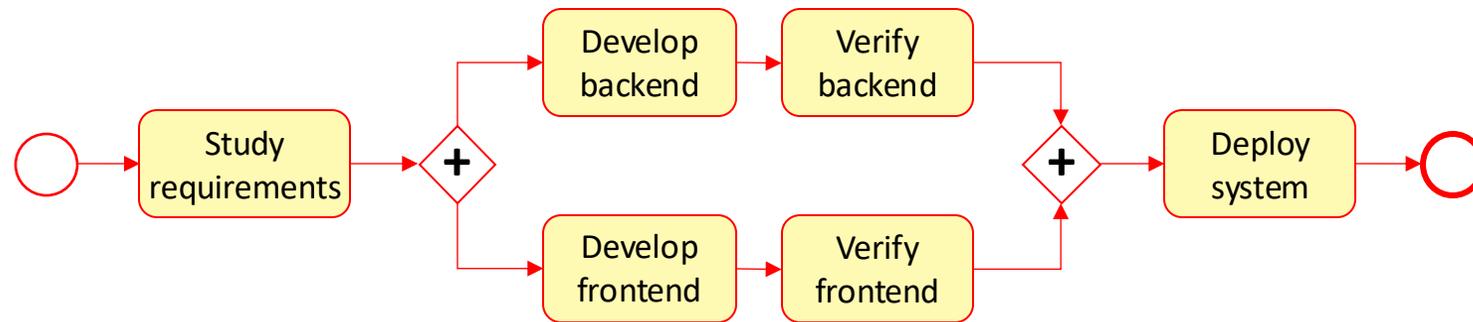
## BPMN Example



Process Model: A BPMN Model for Software Development

# Organizing Work

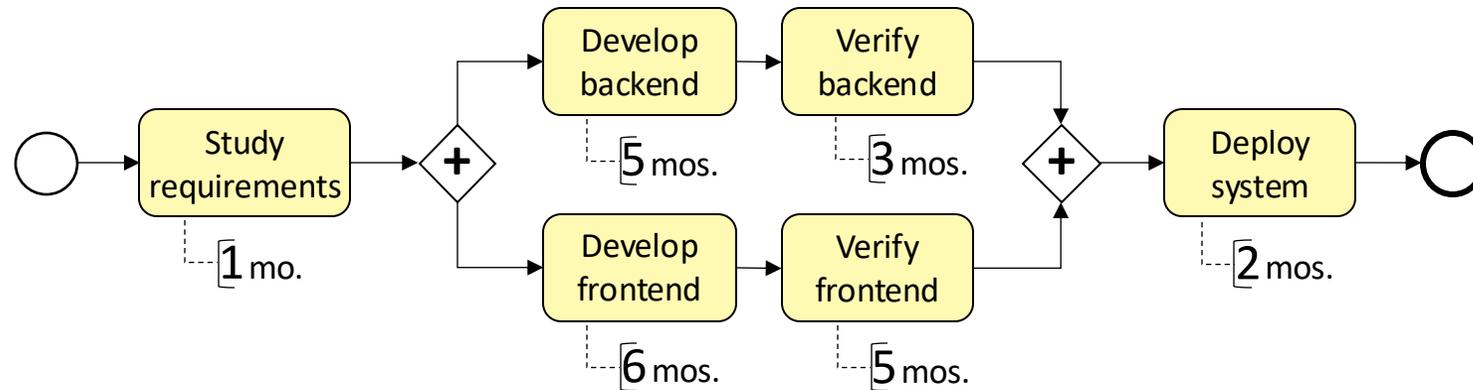
## BPMN Example



Process Model: A BPMN Model for Software Development

# Organizing Work

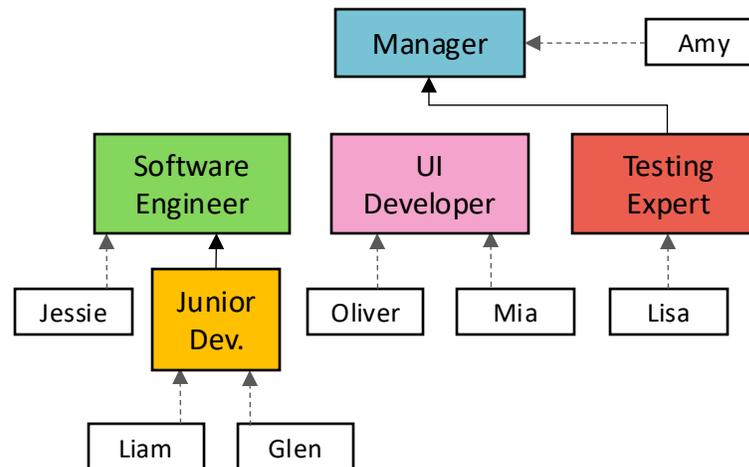
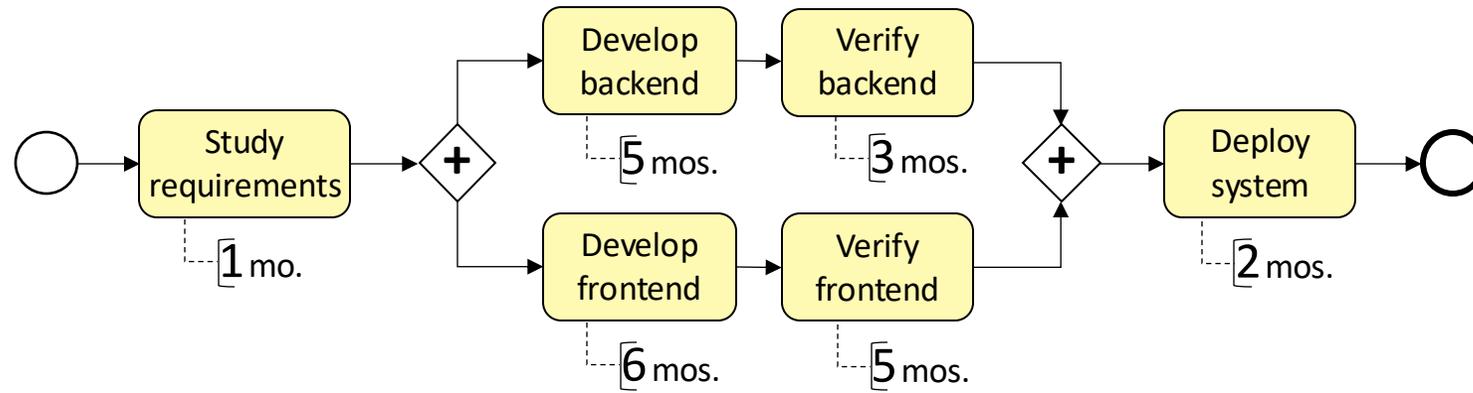
## BPMN Example



Process Model: A BPMN Model for Software Development

# Organizing Resources

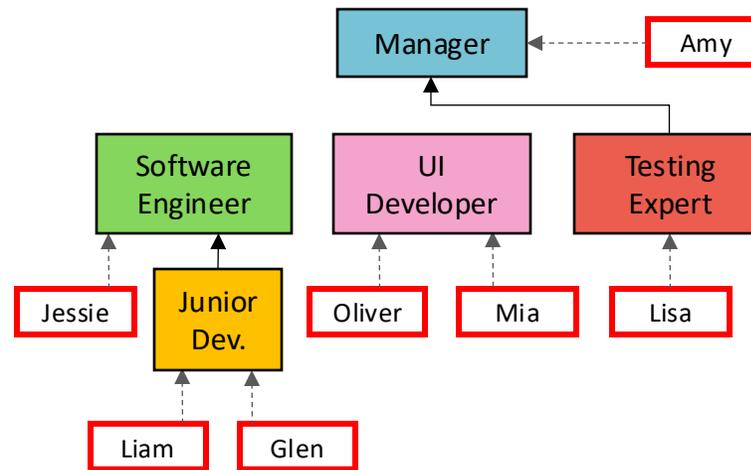
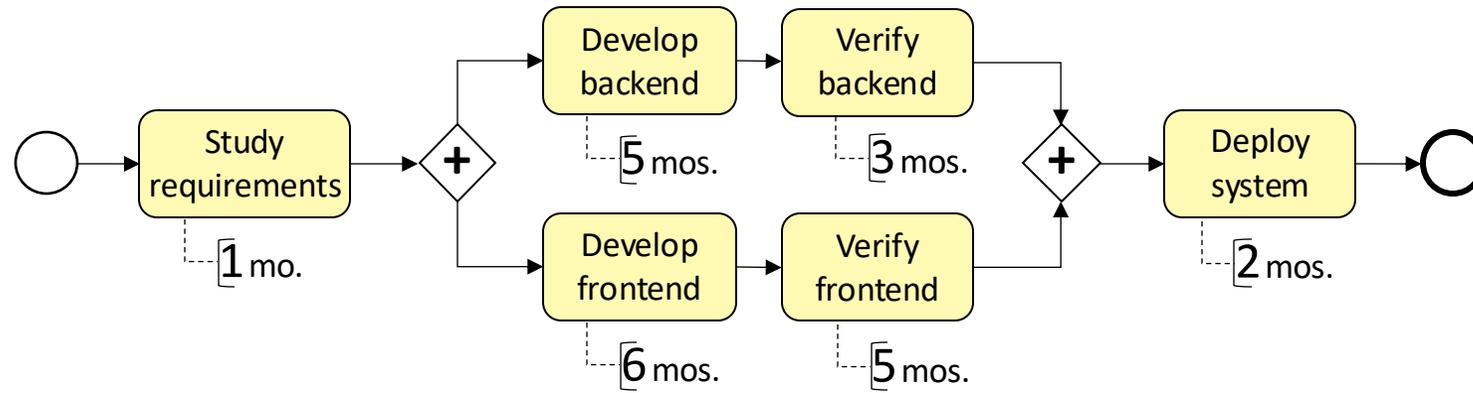
## Role-based Access Control Model



Organizational Model: An RBAC Model of the Software Development Company

# Organizing Resources

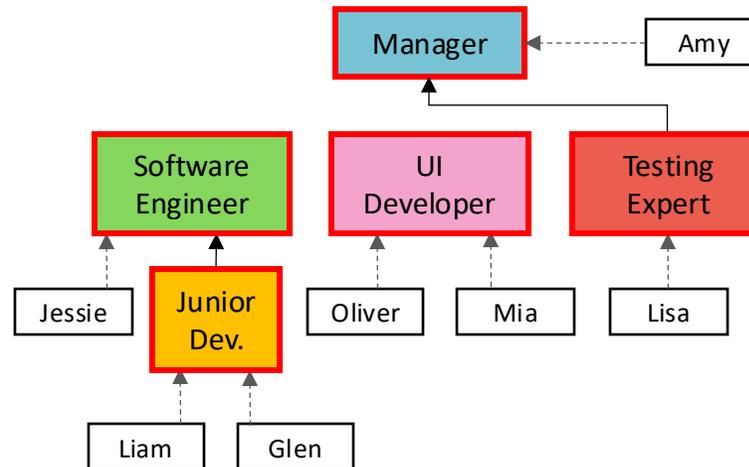
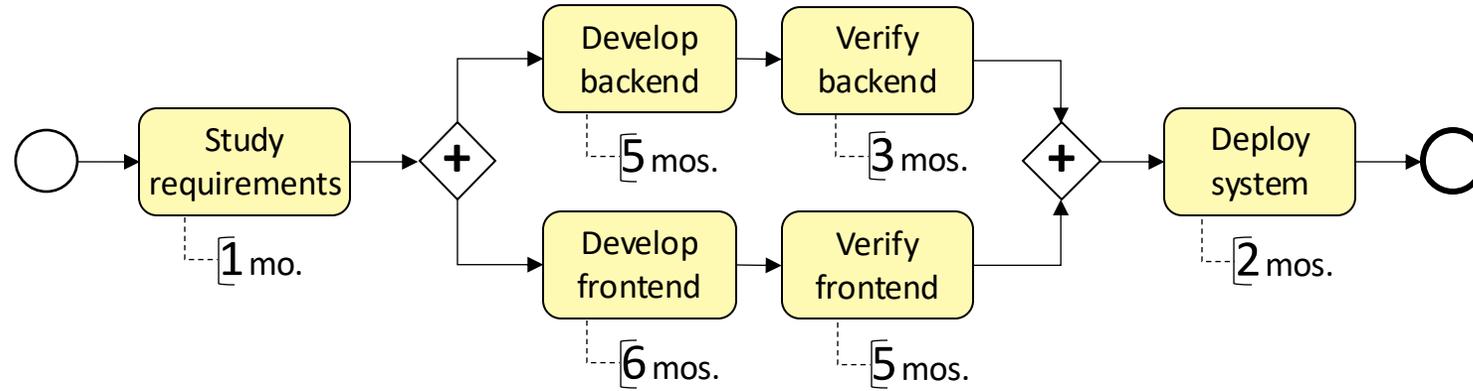
## Role-based Access Control Model



- Resources

# Organizing Resources

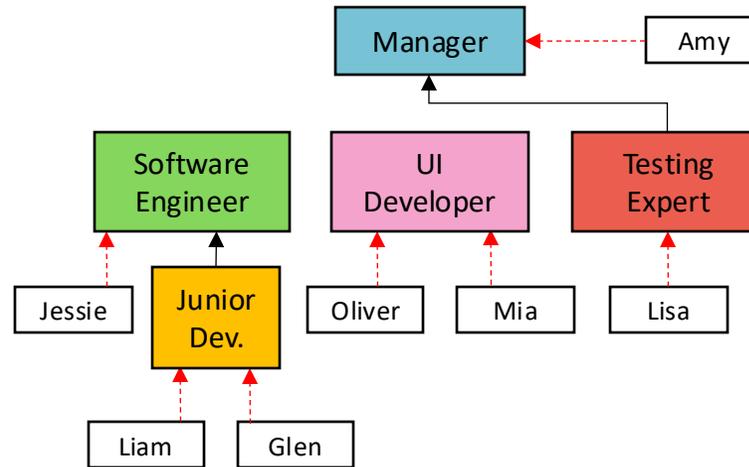
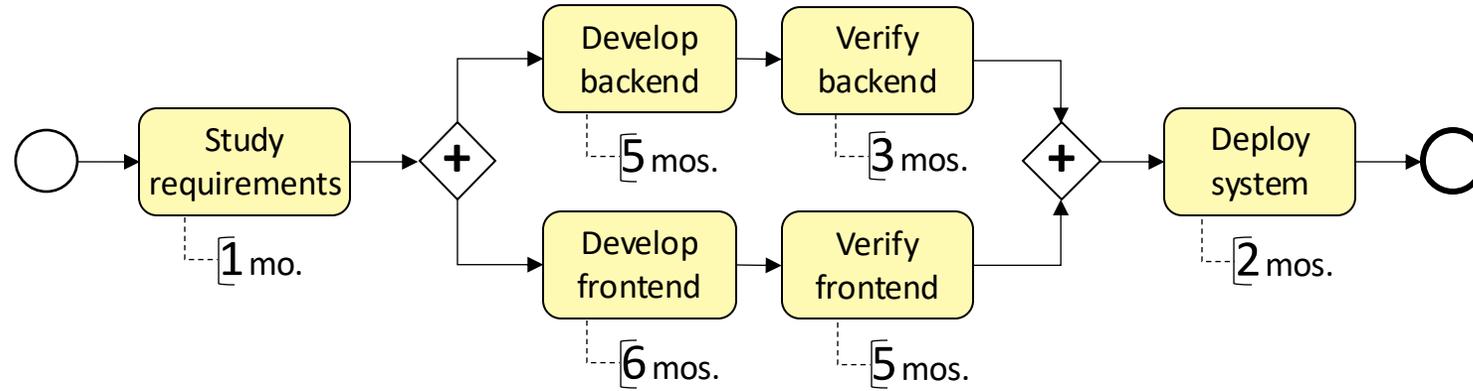
## Role-based Access Control Model



- Resources
- Roles

# Organizing Resources

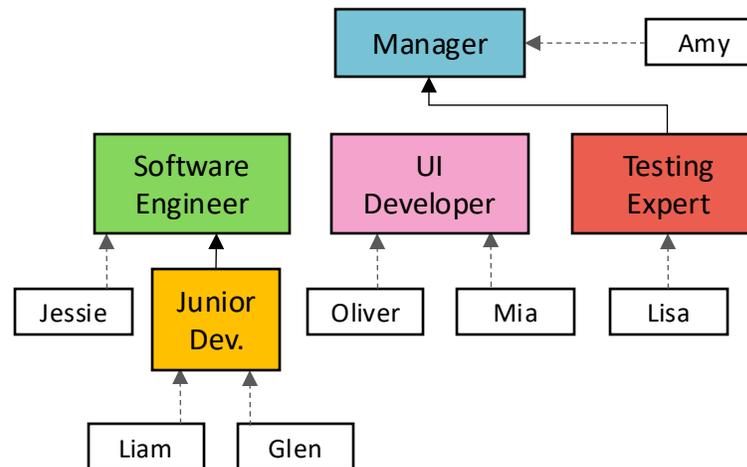
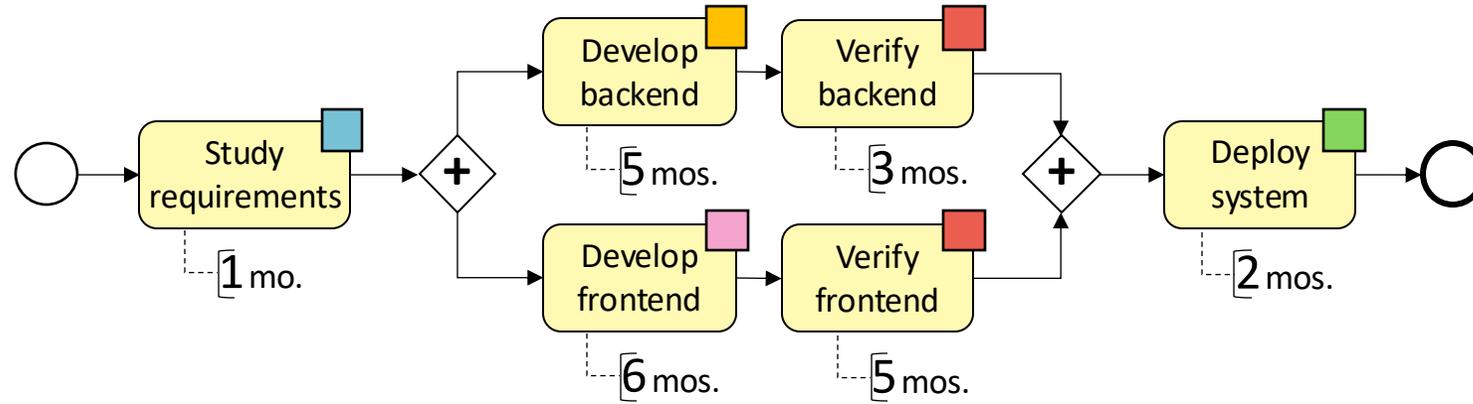
## Role-based Access Control Model



- Resources
- Roles
- **Resource-to-Role assignments**

# Organizing Resources

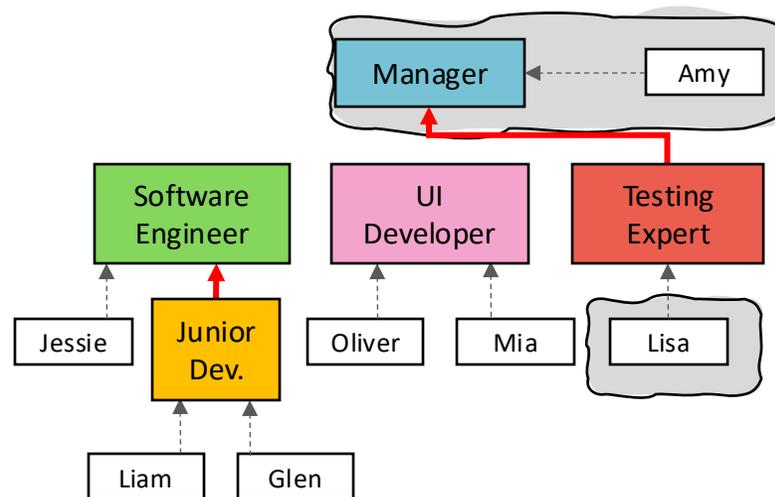
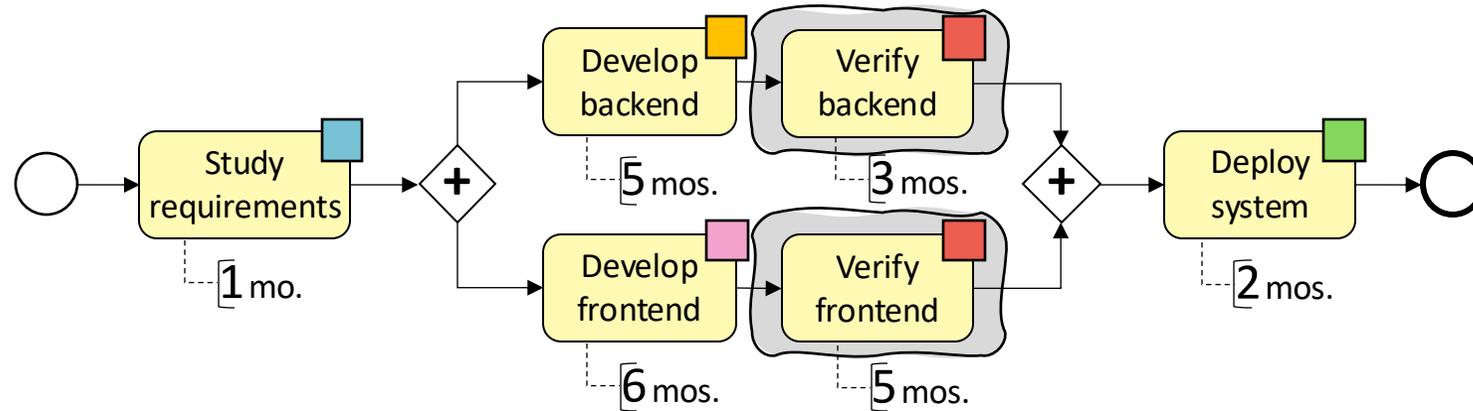
## Role-based Access Control Model



- Resources
- Roles
- Resource-to-Role assignments
- **Activity-to-Role assignments**

# Organizing Resources

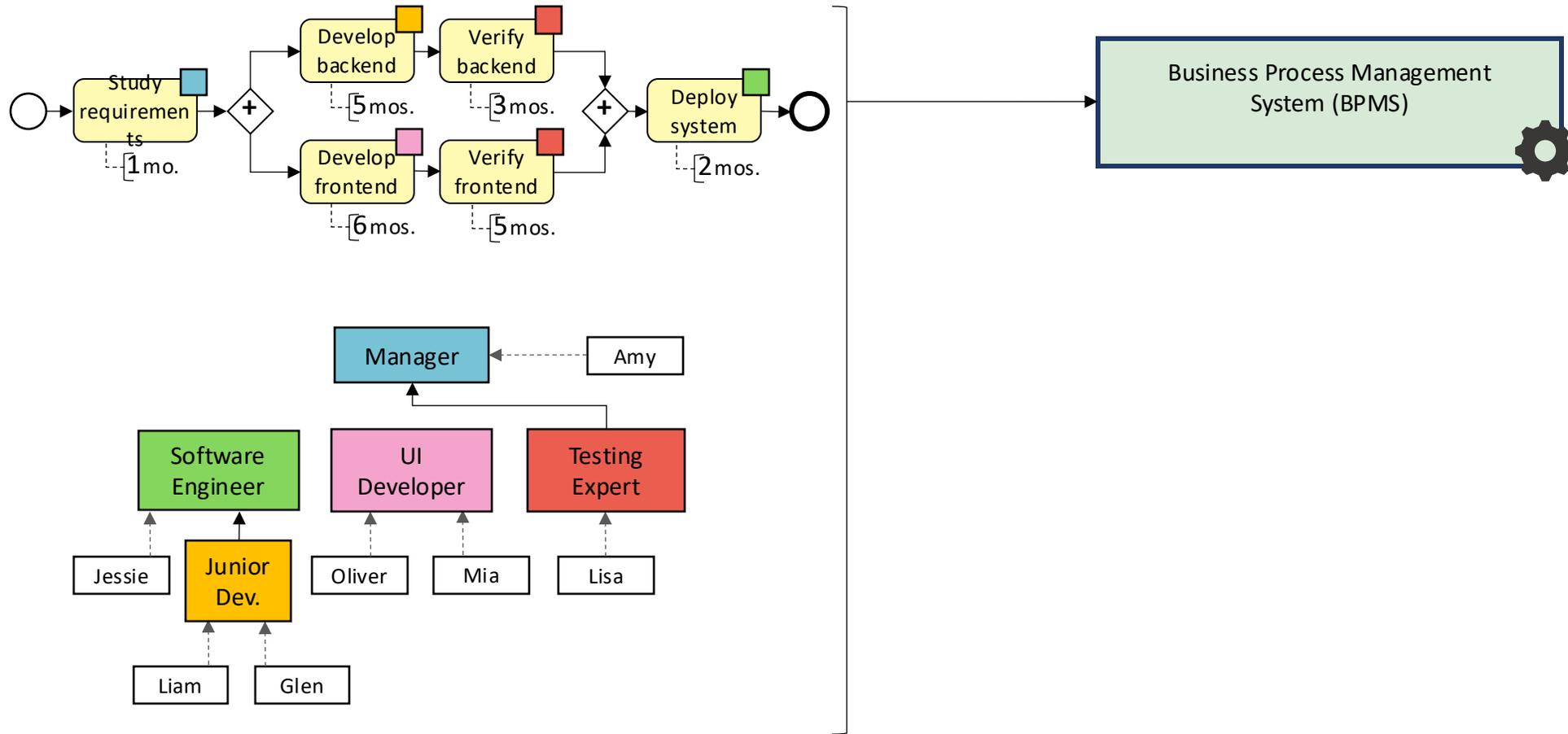
## Role-based Access Control Model



- Resources
- Roles
- Resource-to-Role assignments
- Activity-to-Role assignments
- **Role-to-Role assignments**

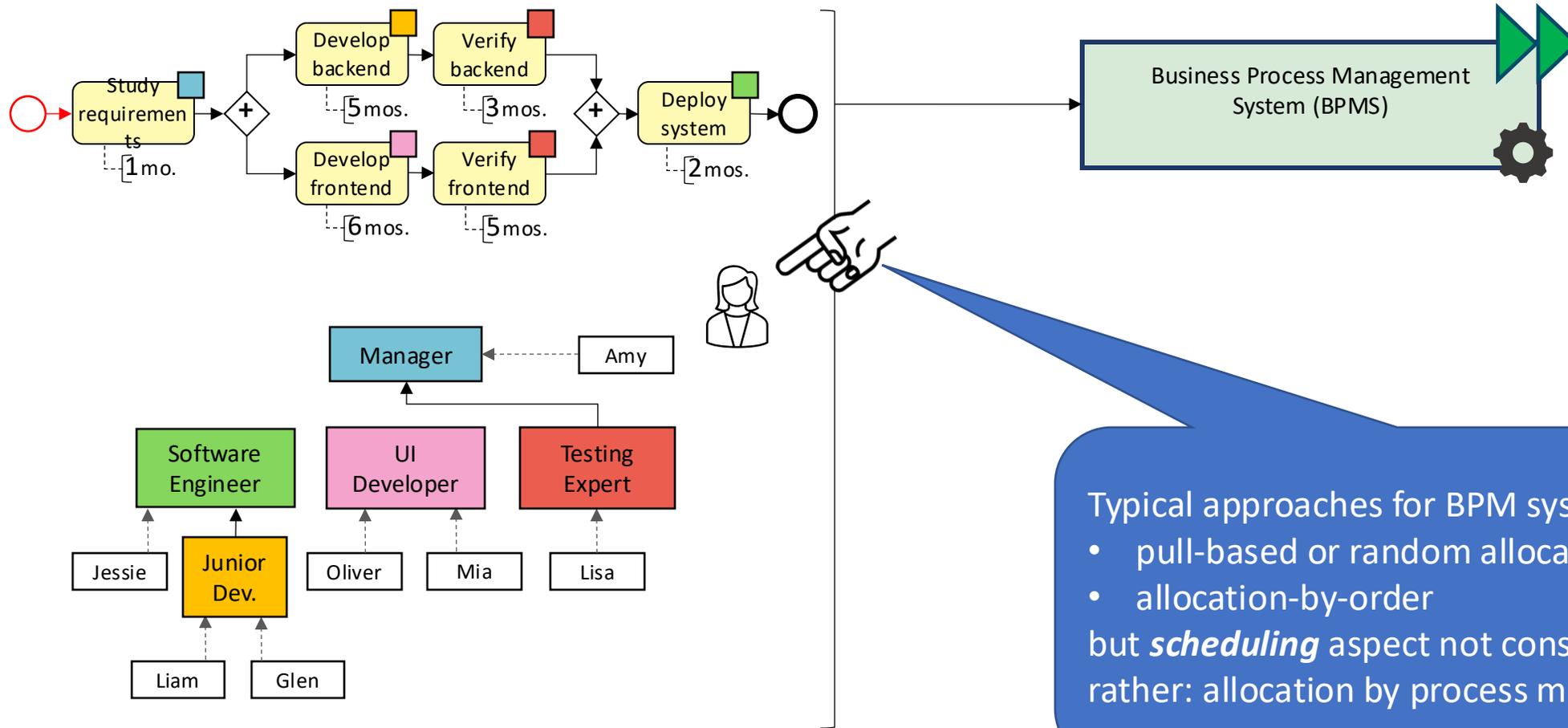
# Running Business Processes

## Business Process Management System



# Running Business Processes

## Business Process Management System



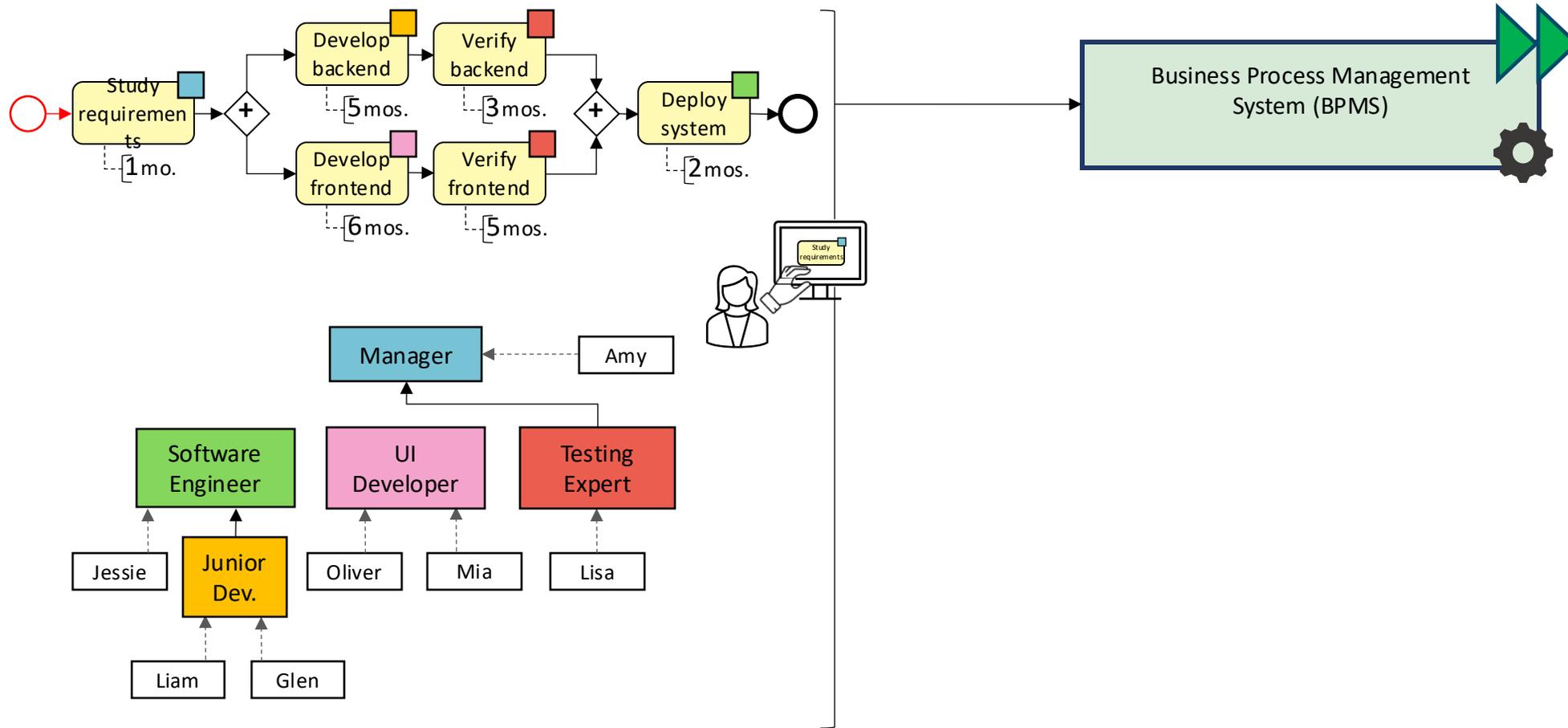
Typical approaches for BPM systems:

- pull-based or random allocation,
- allocation-by-order

but **scheduling** aspect not considered so far, rather: allocation by process managers.

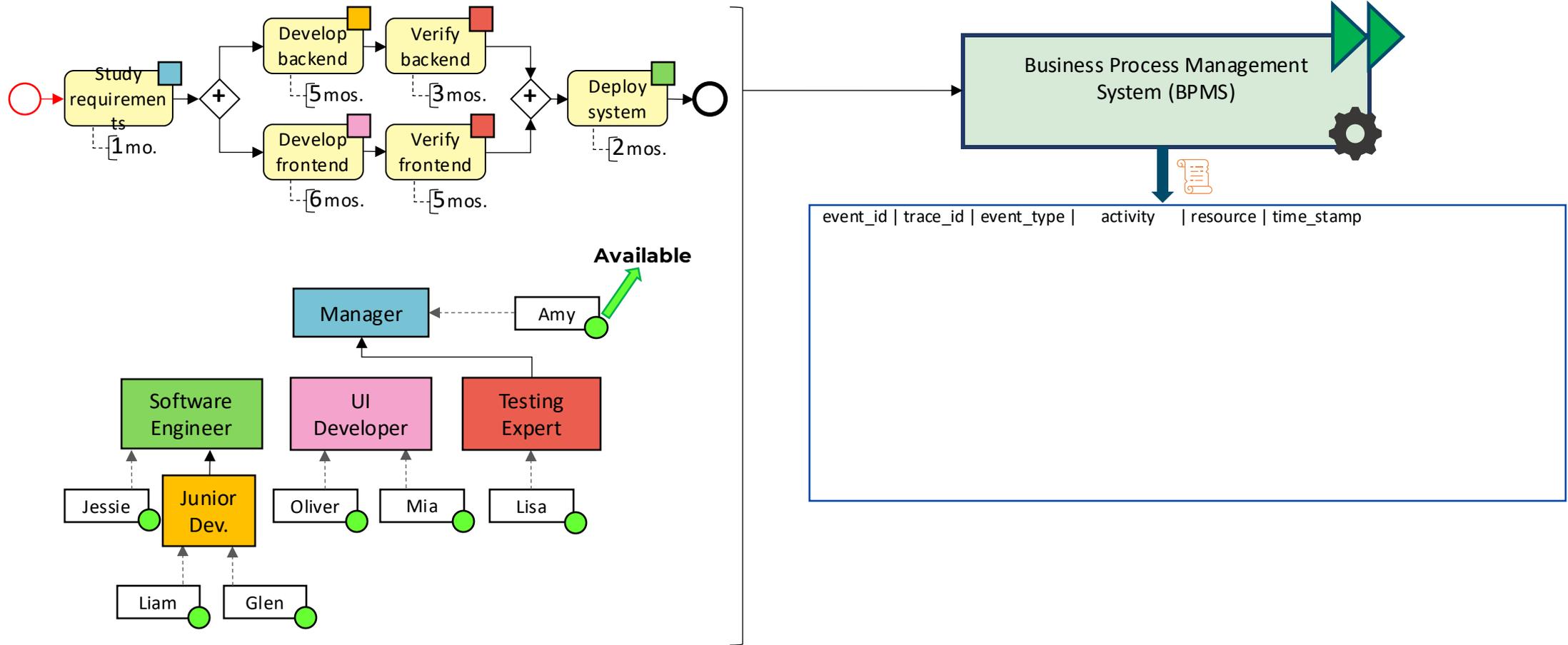
# Running Business Processes

## Business Process Management System



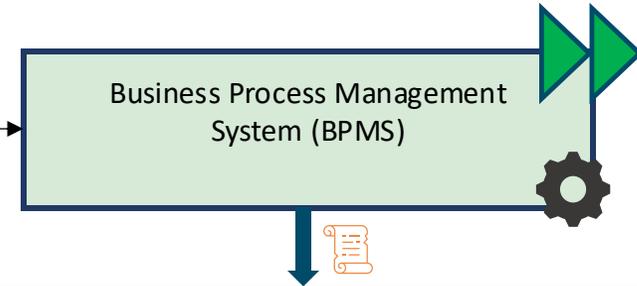
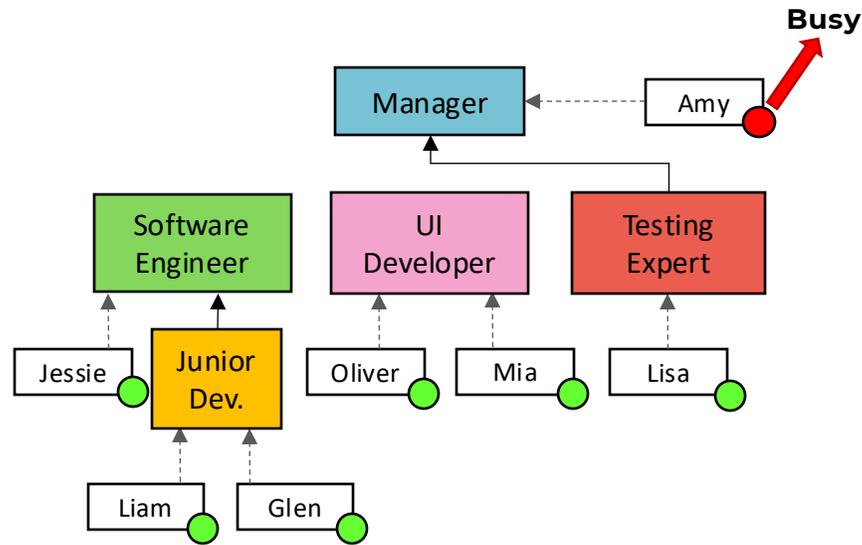
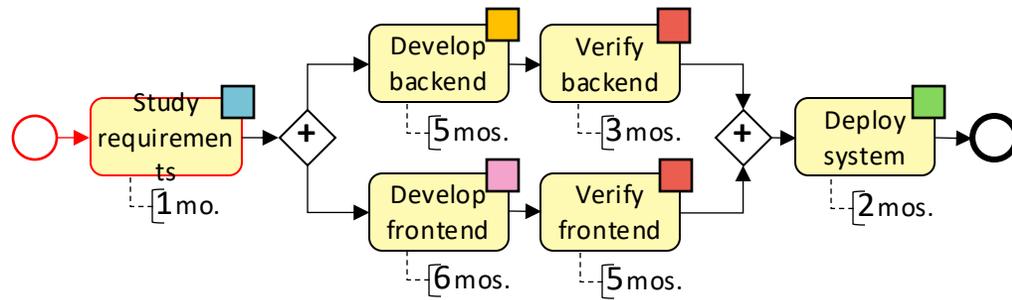
# Running Business Processes

## Business Process Management System



# Running Business Processes

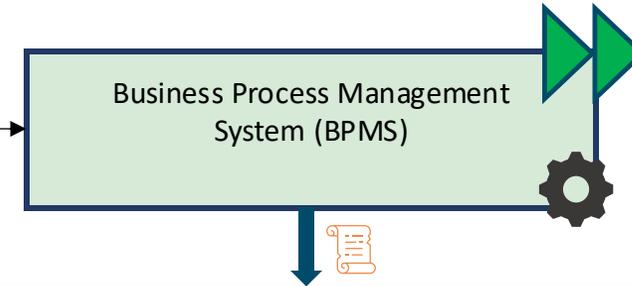
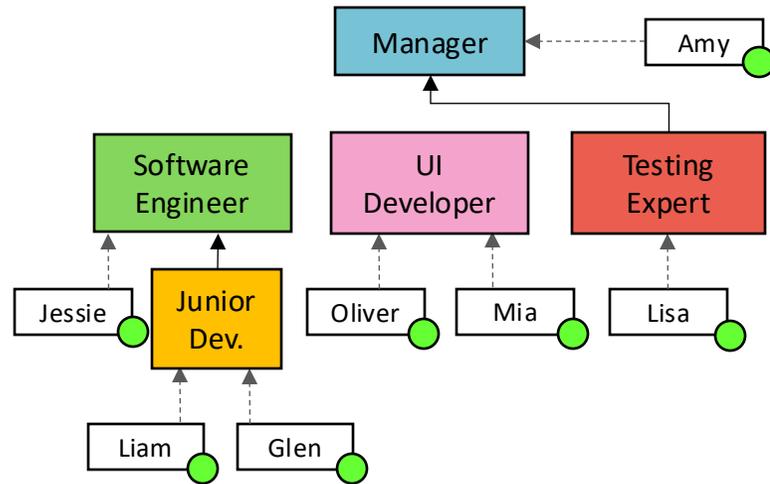
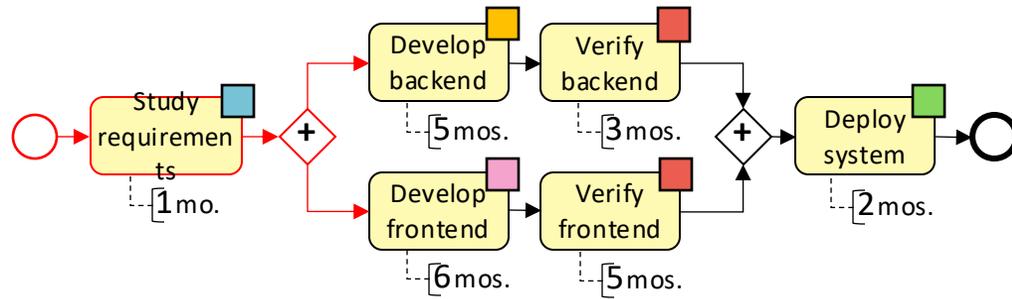
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020

# Running Business Processes

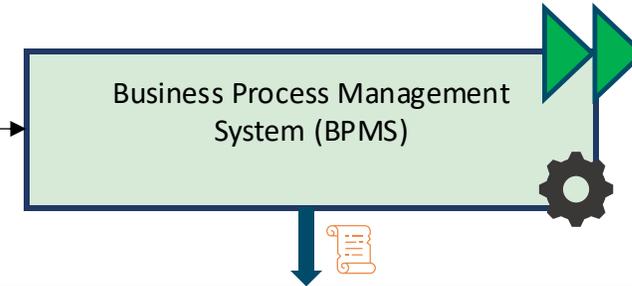
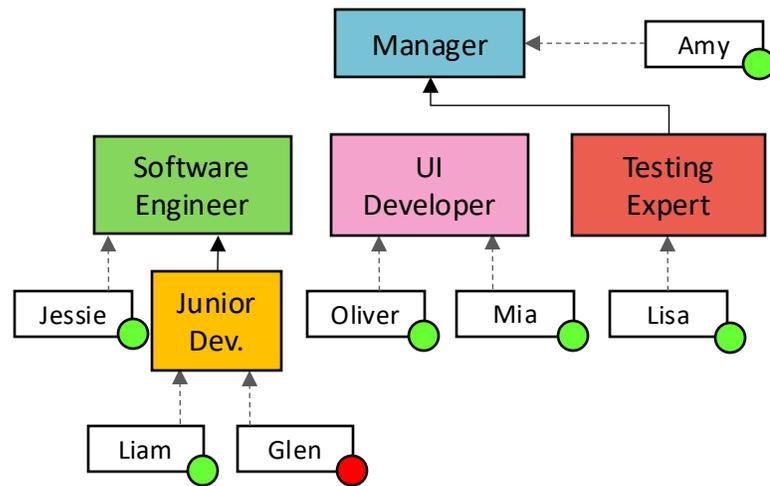
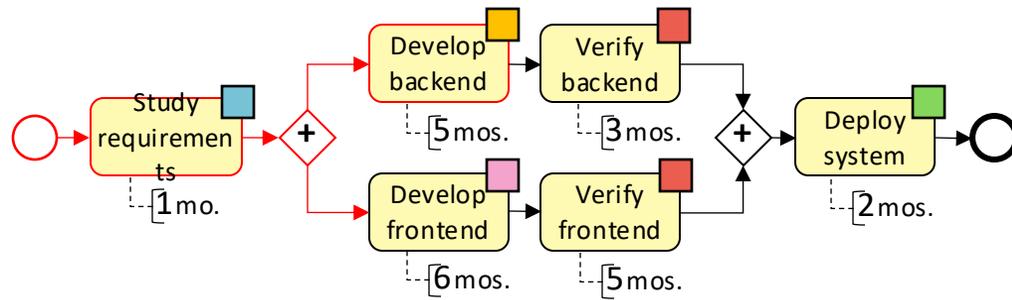
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020

# Running Business Processes

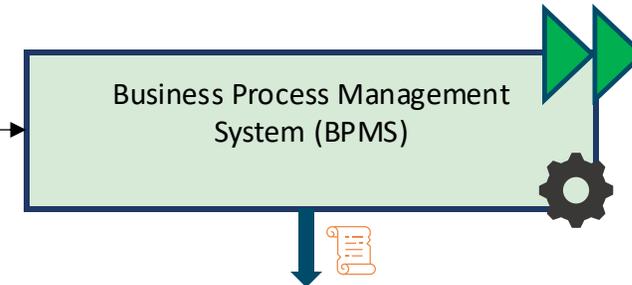
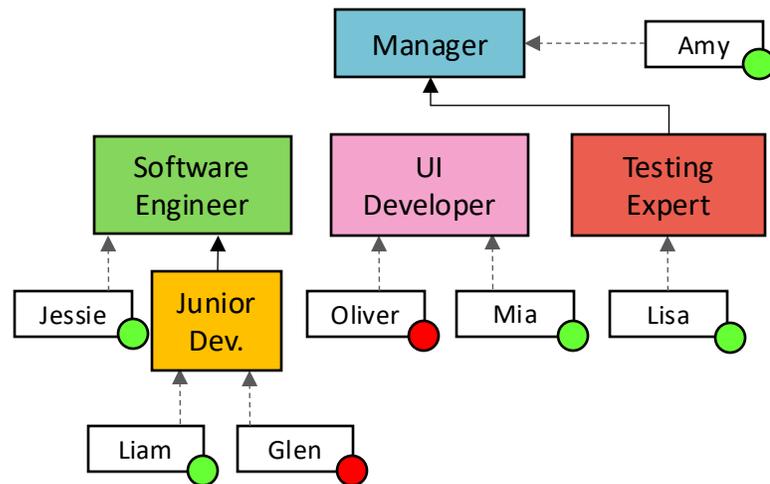
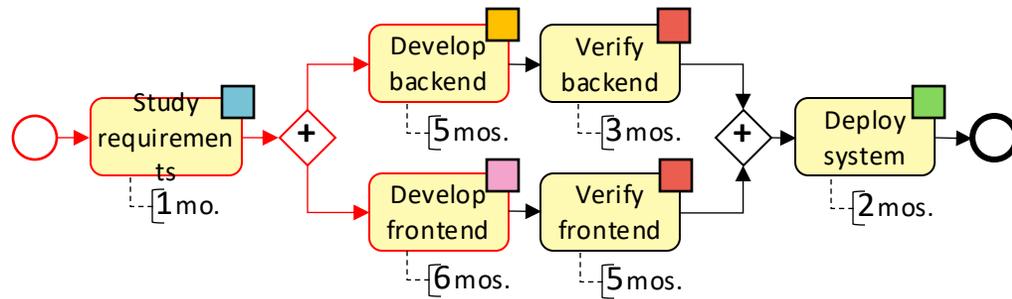
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020

# Running Business Processes

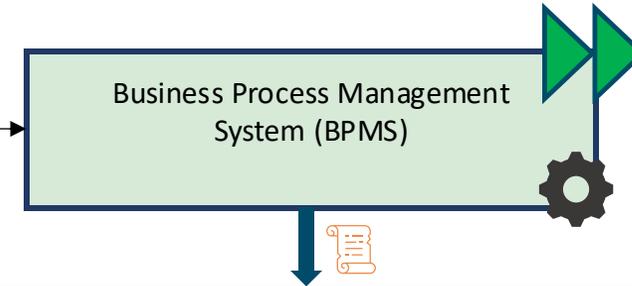
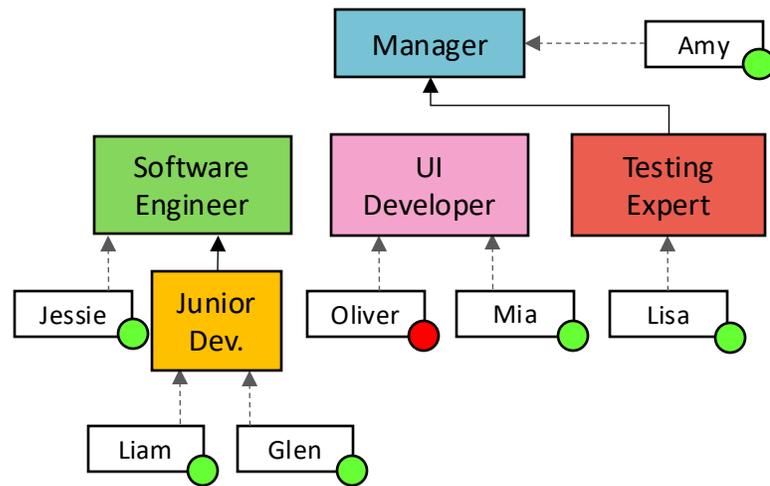
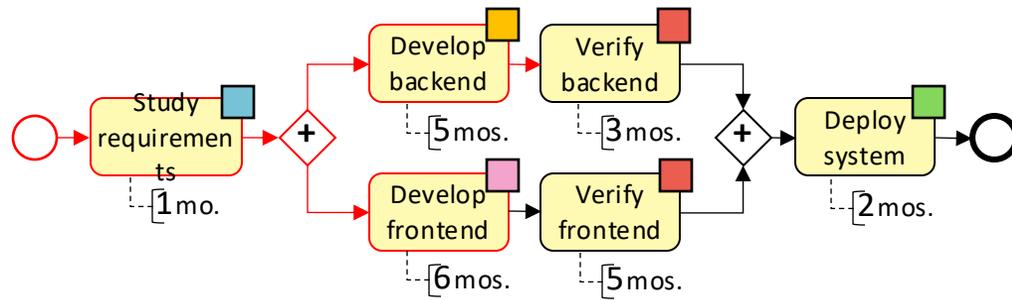
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020

# Running Business Processes

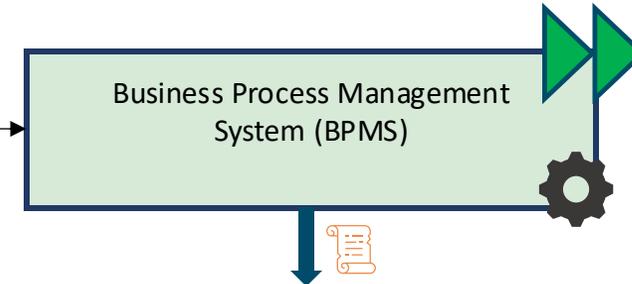
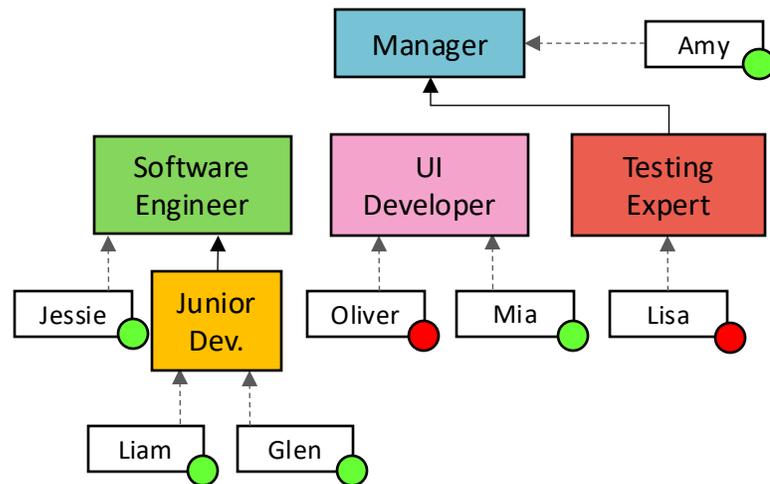
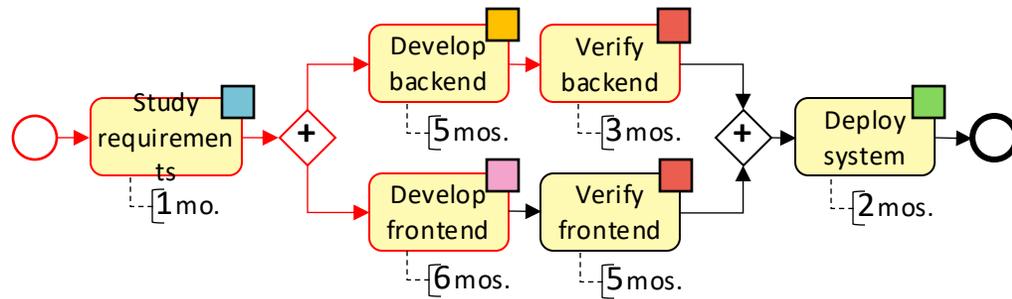
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020
e5	1	end	Develop_backend	Glen	07/2020

# Running Business Processes

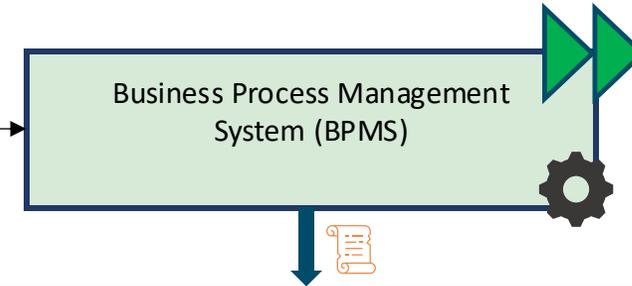
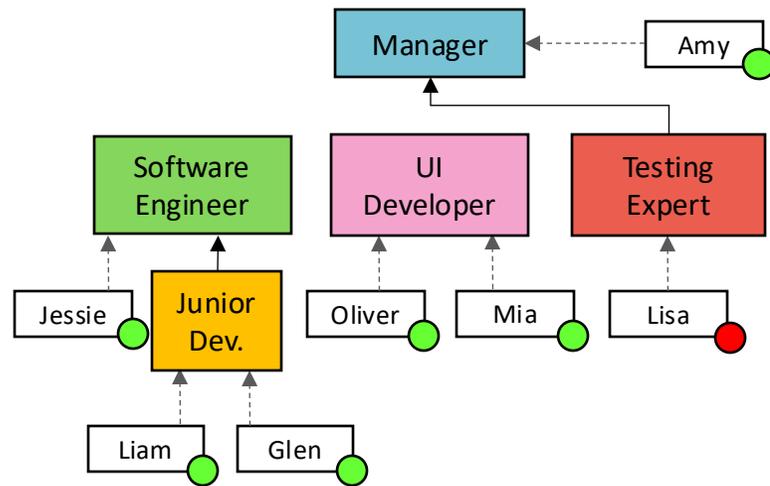
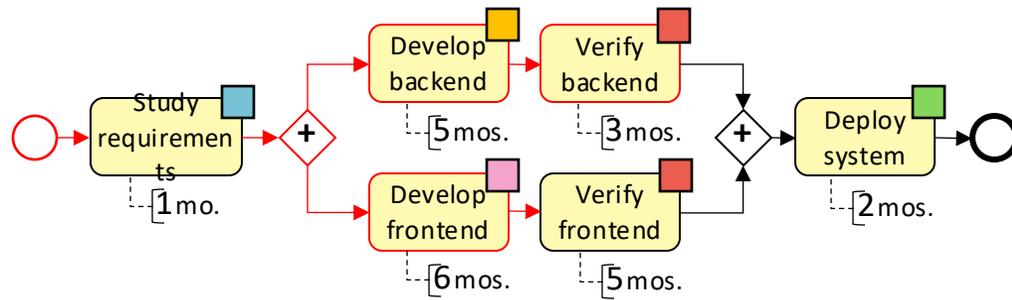
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020
e5	1	end	Develop_backend	Glen	07/2020
e6	1	start	Verify_backend	Lisa	07/2020

# Running Business Processes

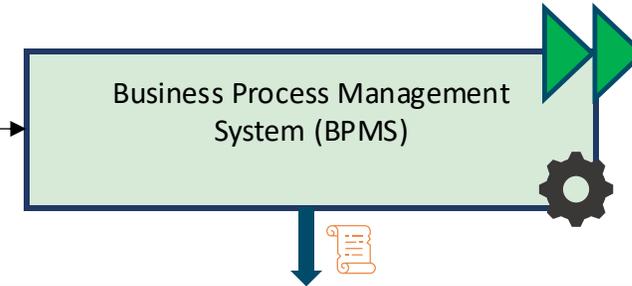
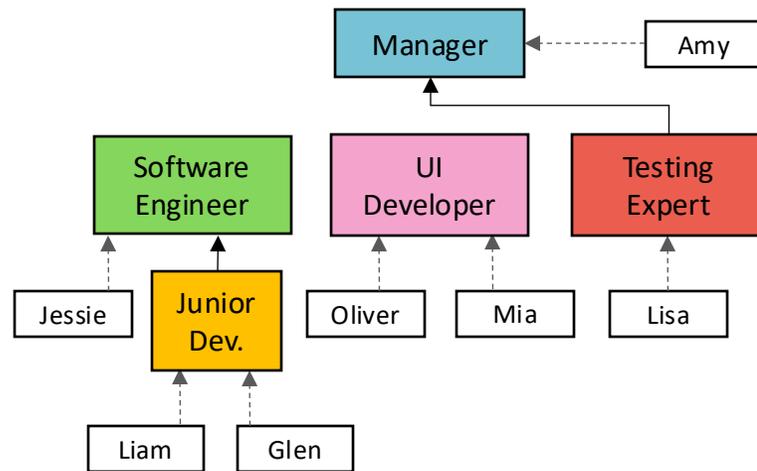
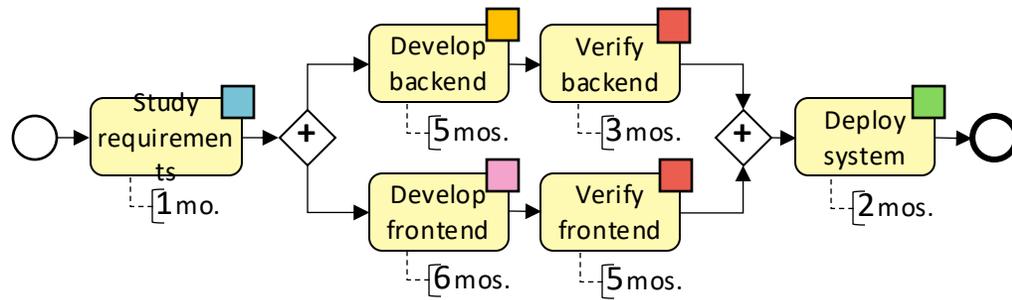
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020
e5	1	end	Develop_backend	Glen	07/2020
e6	1	start	Verify_backend	Lisa	07/2020
e7	1	end	Develop_frontend	Oliver	10/2020

# Running Business Processes

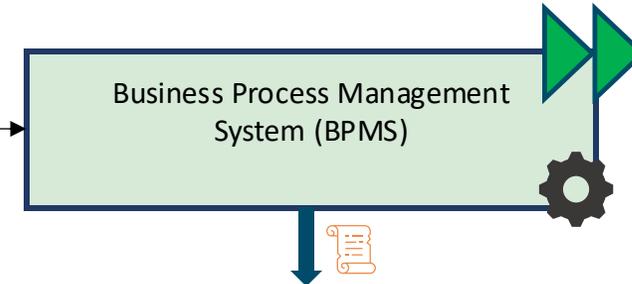
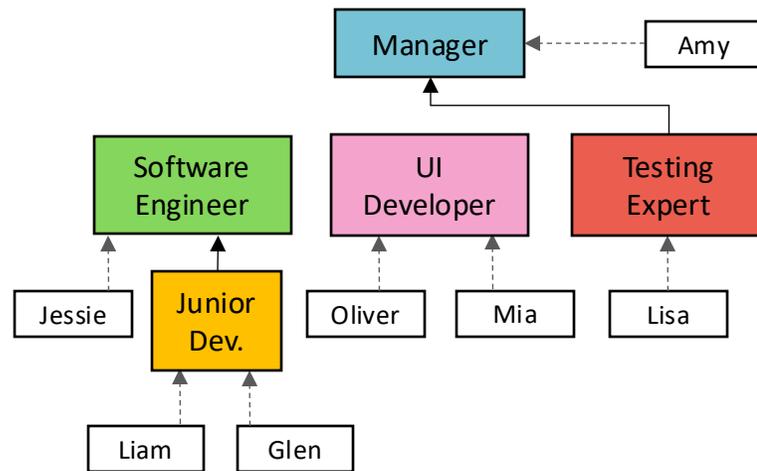
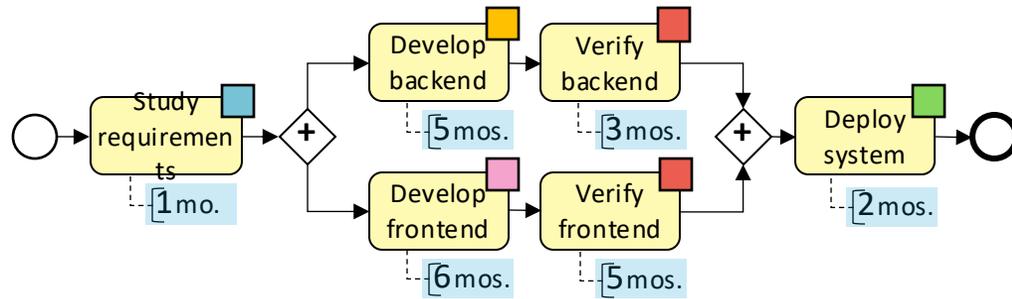
## Business Process Management System



event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020
e5	1	end	Develop_backend	Glen	07/2020
e6	1	start	Verify_backend	Lisa	07/2020
e7	1	end	Develop_frontend	Oliver	10/2020
e8	1	end	Verify_backend	Lisa	10/2020
e9	1	start	Verify_frontend	Lisa	10/2020
e10	1	end	Verify_frontend	Lisa	11/2020
e11	1	start	Deploy_system	Jessie	11/2020
e12	1	end	Deploy_system	Jessie	01/2021

# Running Business Processes

## Mining a Temporal Model from the Event Log

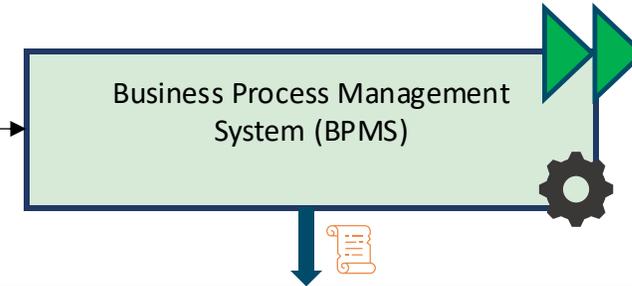
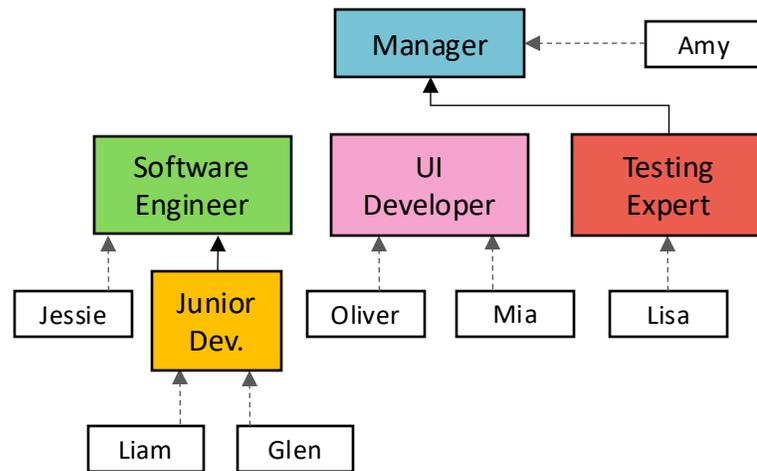
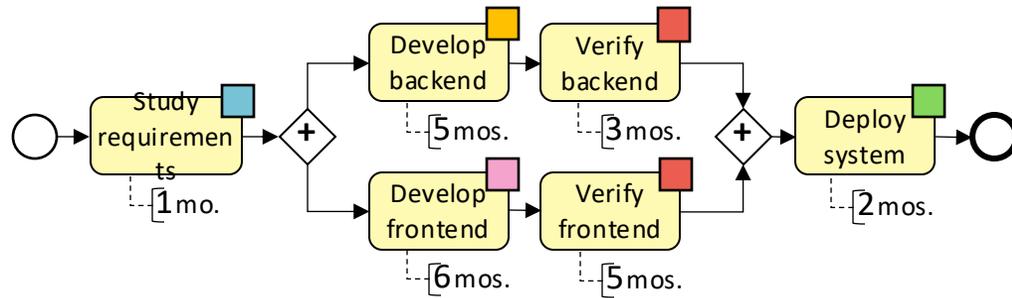


event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020
e5	1	end	Develop_backend	Glen	07/2020
e6	1	start	Verify_backend	Lisa	07/2020
e7	1	end	Develop_frontend	Oliver	10/2020
e8	1	end	Verify_backend	Lisa	10/2020
e9	1	start	Verify_frontend	Lisa	10/2020
e10	1	end	Verify_frontend	Lisa	11/2020
e11	1	start	Deploy_system	Jessie	11/2020
e12	1	end	Deploy_system	Jessie	01/2021

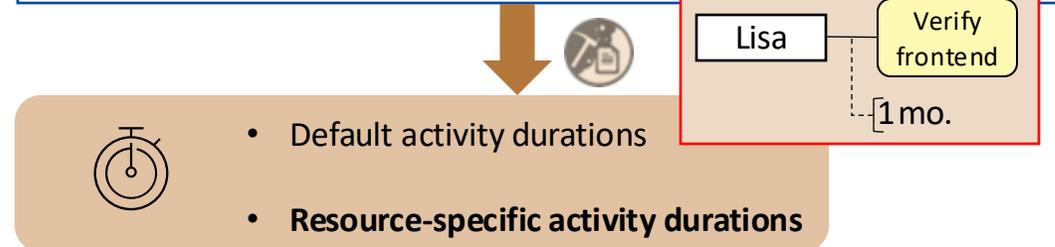

**Default activity durations**

# Running Business Processes

## Mining a Temporal Model from the Event Log

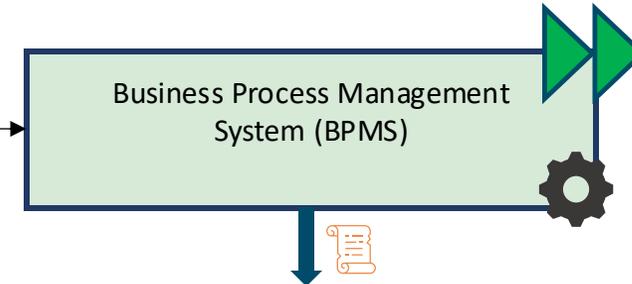
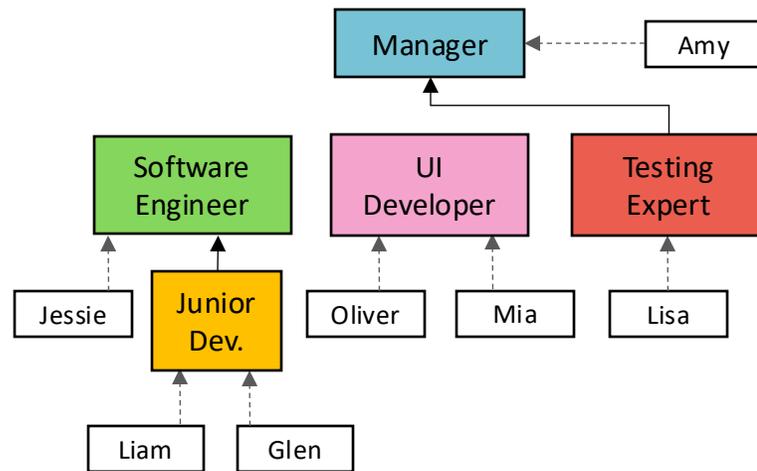
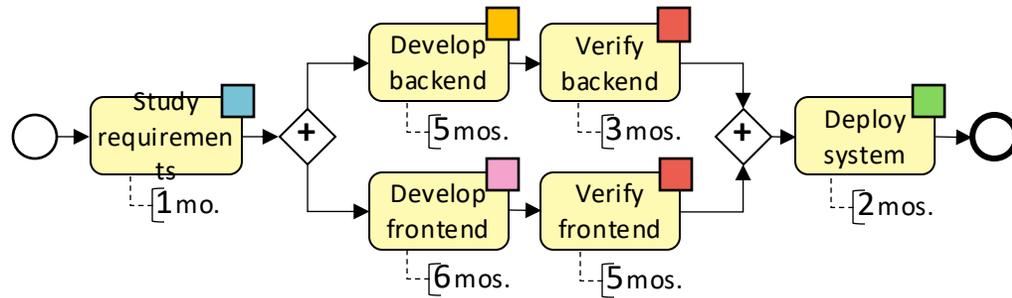


event_id	trace_id	event_type	activity	resource	time_stamp
e1	1	start	Study_requirements	Amy	01/2020
e2	1	end	Study_requirements	Amy	02/2020
e3	1	start	Develop_backend	Glen	02/2020
e4	1	start	Develop_frontend	Oliver	04/2020
e5	1	end	Develop_backend	Glen	07/2020
e6	1	start	Verify_backend	Lisa	07/2020
e7	1	end	Develop_frontend	Oliver	10/2020
e8	1	end	Verify_backend	Lisa	10/2020
e9	1	start	Verify_frontend	Lisa	10/2020
e10	1	end	Verify_frontend	Lisa	11/2020
e11	1	start	Deploy_system	Jessie	11/2020
e12	1	end	Deploy_system	Jessie	01/2021

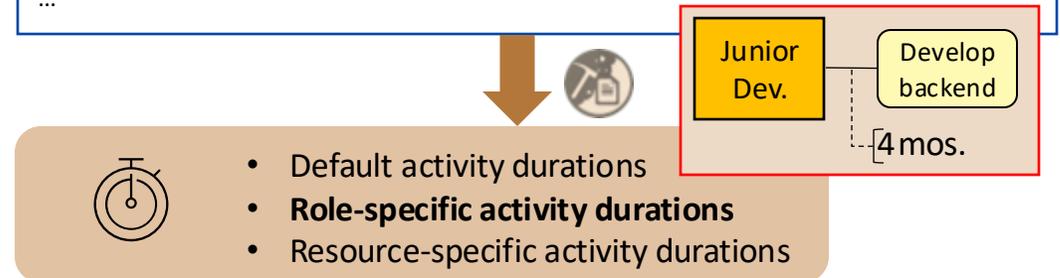


# Running Business Processes

## Mining a Temporal Model from the Event Log

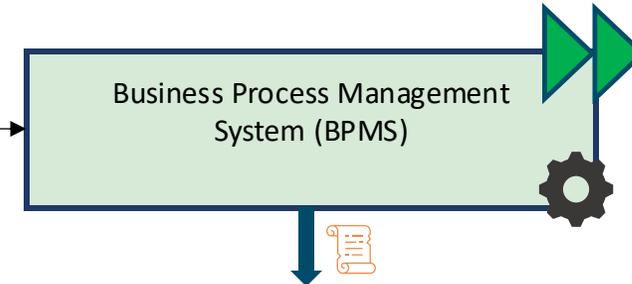
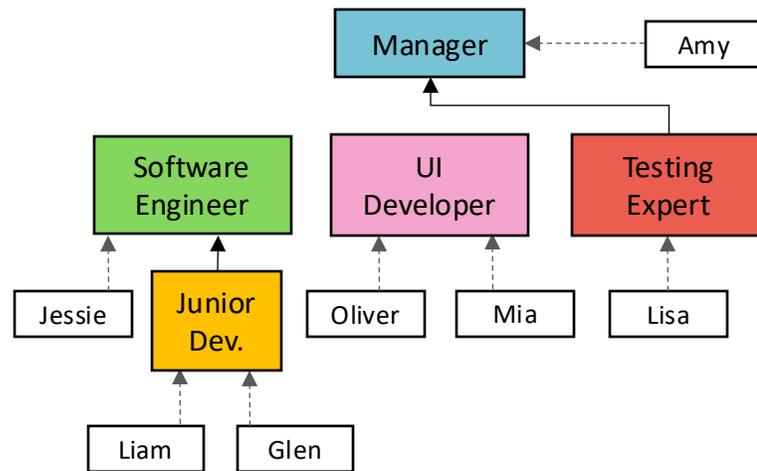
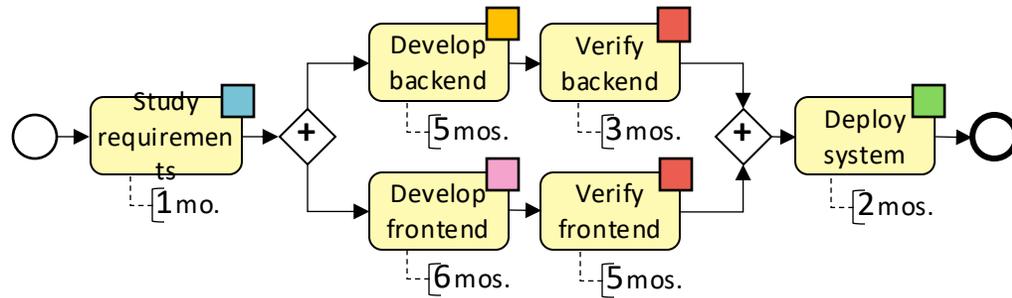


event_id	trace_id	event_type	activity	resource	time_stamp
...					
e3	1	start	Develop_backend	Glen	02/2020
e5	1	end	Develop_backend	Glen	07/2020
...					
e12	2	start	Develop_backend	Liam	10/2020
e15	2	end	Develop_backend	Liam	01/2021
...					

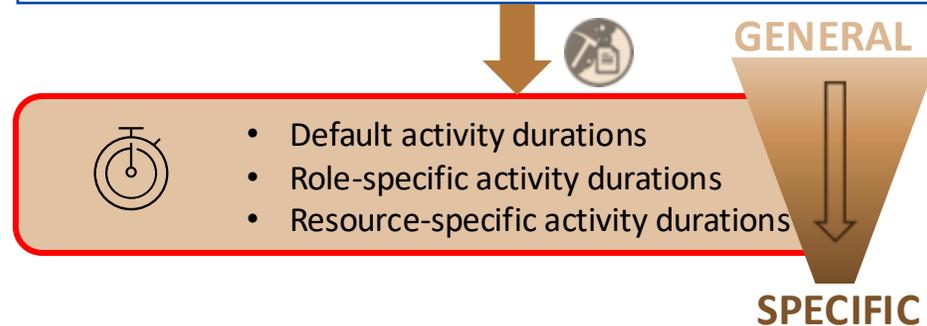


# Running Business Processes

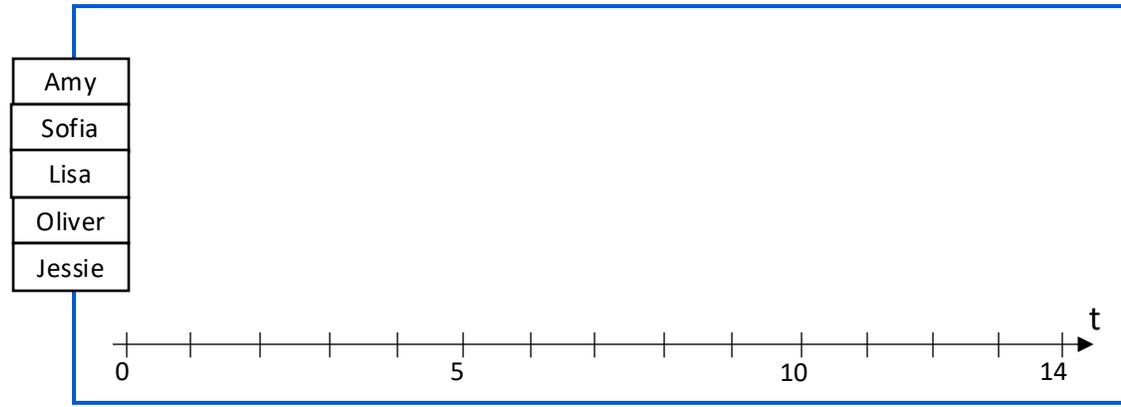
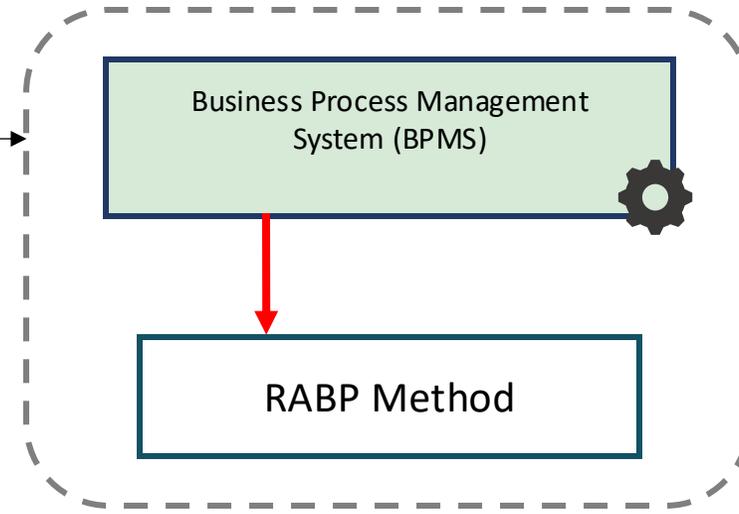
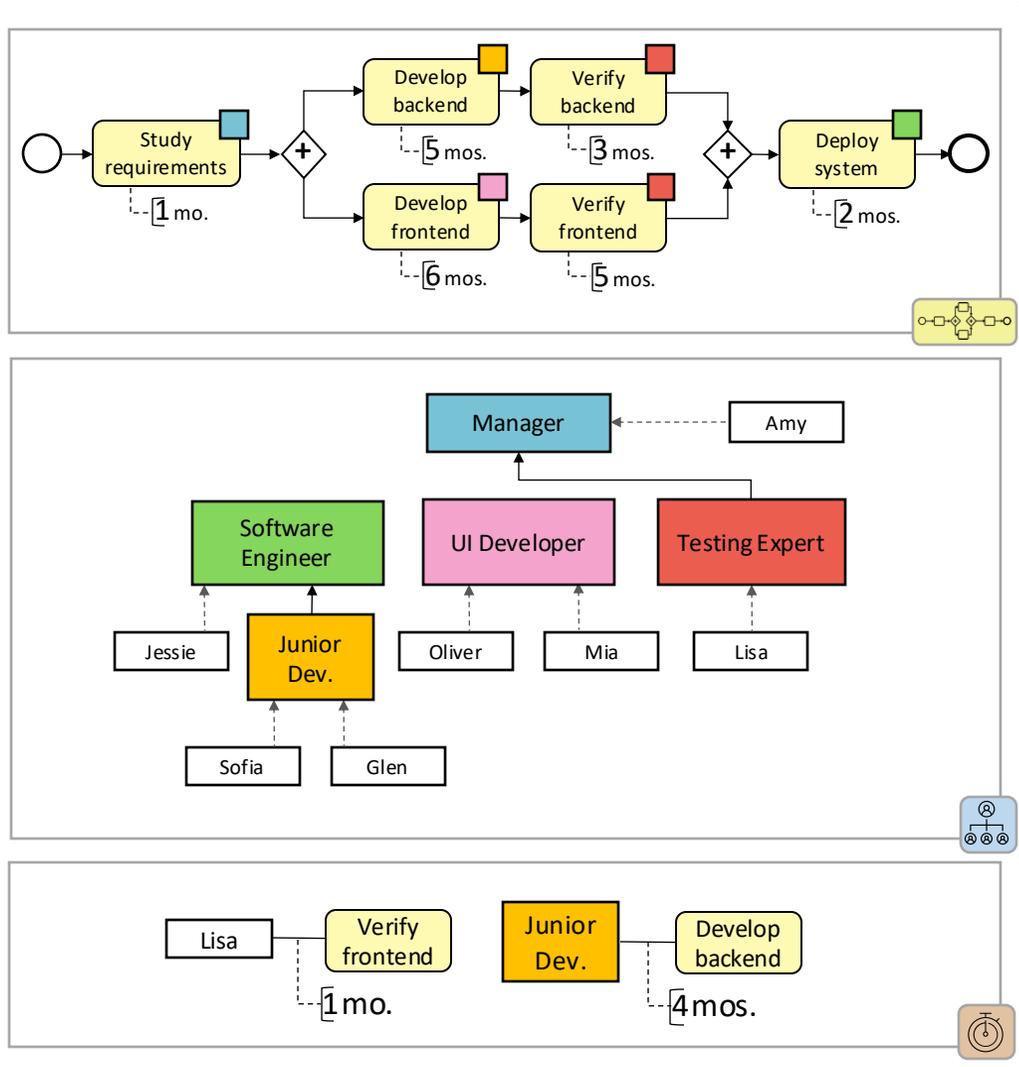
## Mining a Temporal Model from the Event Log



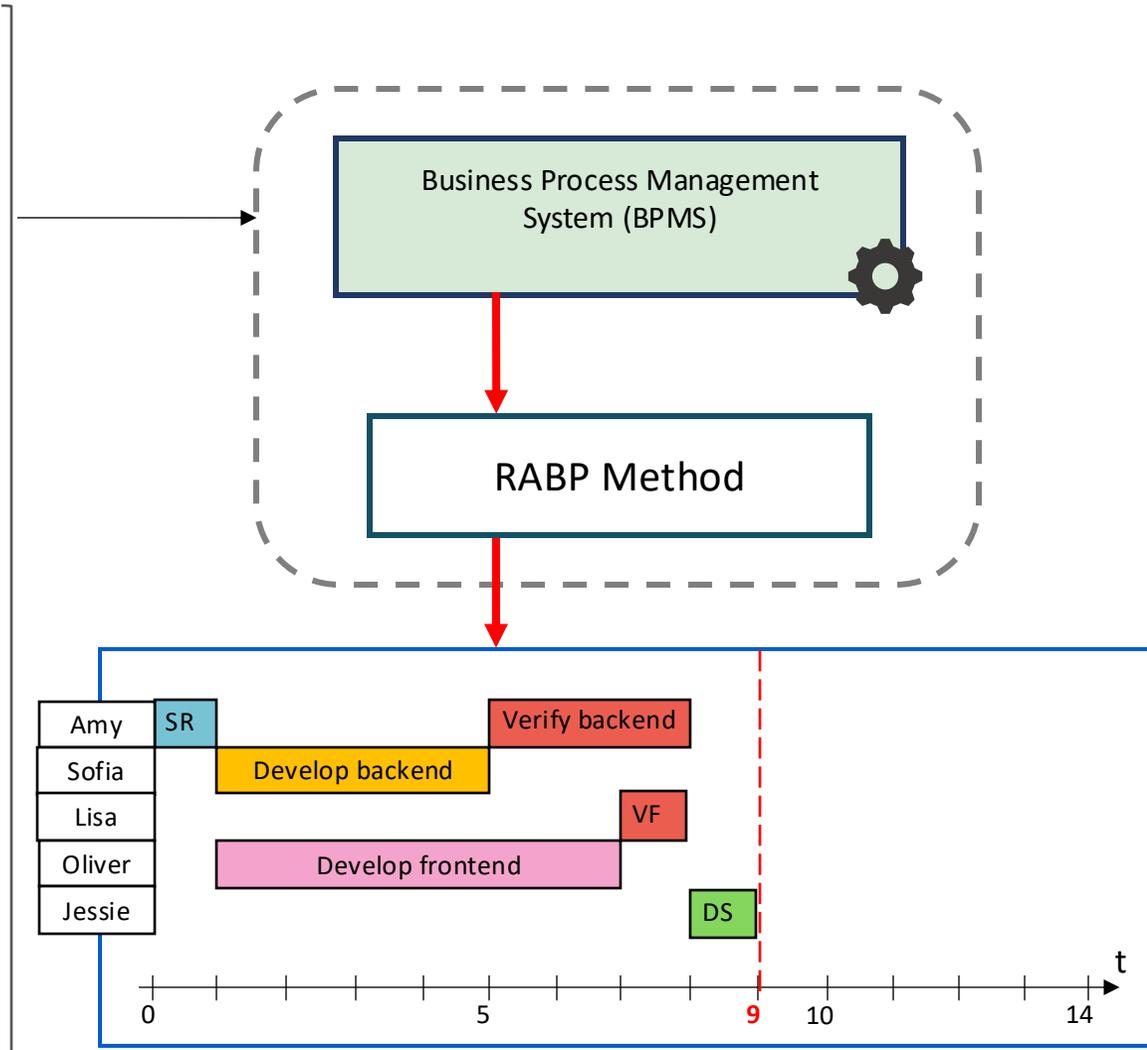
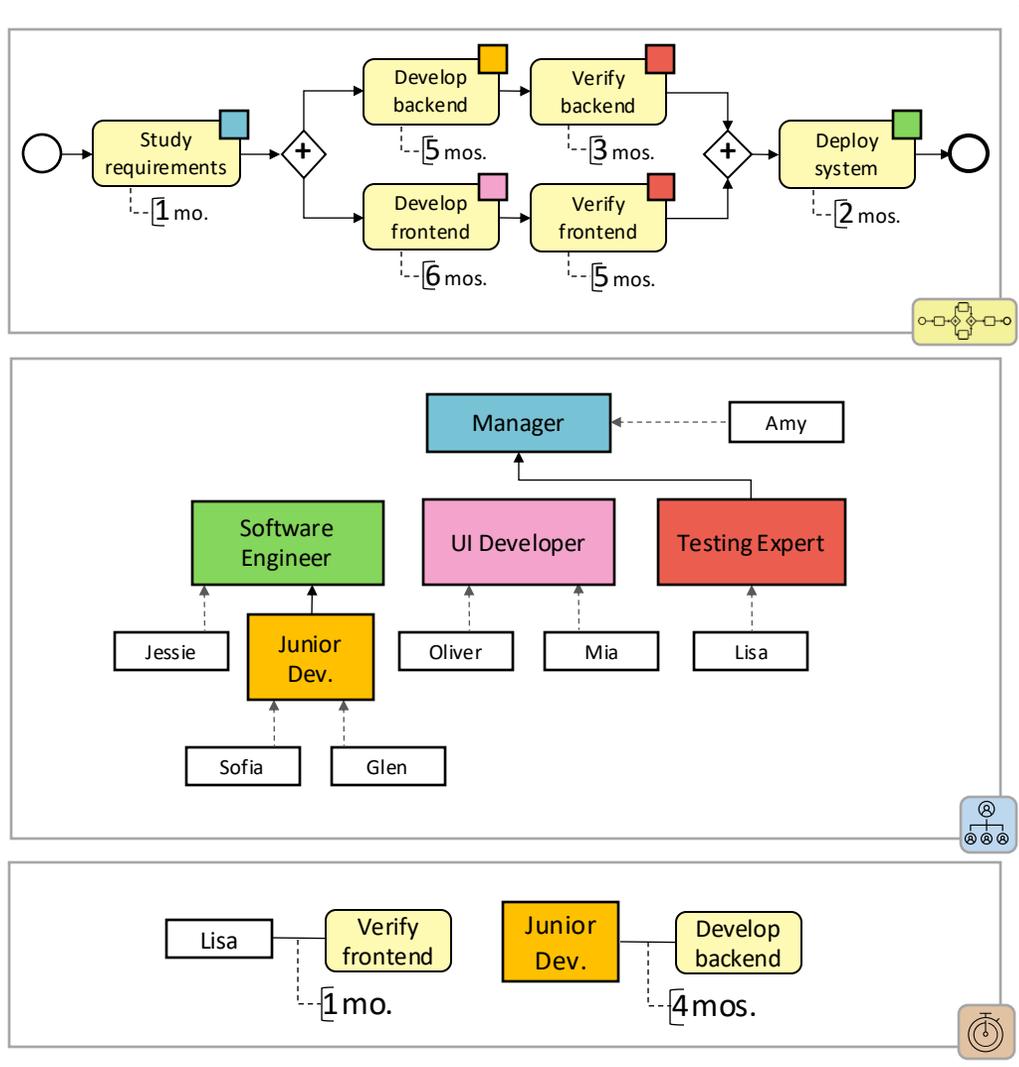
event_id	trace_id	event_type	activity	resource	time_stamp
...					
e3	1	start	Develop_backend	Glen	02/2020
e5	1	end	Develop_backend	Glen	07/2020
...					
e12	2	start	Develop_backend	Liam	10/2020
e15	2	end	Develop_backend	Liam	01/2021
...					



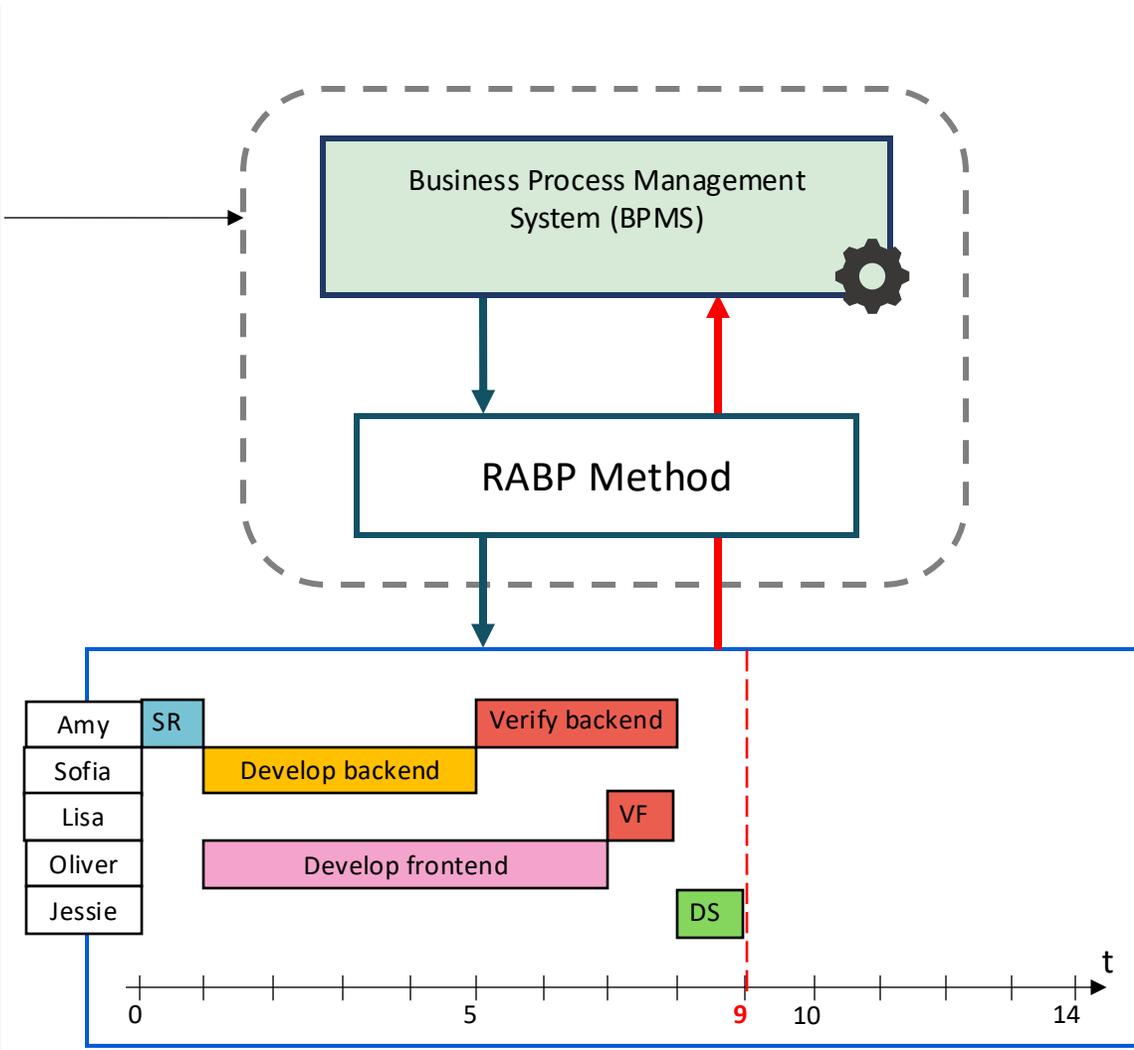
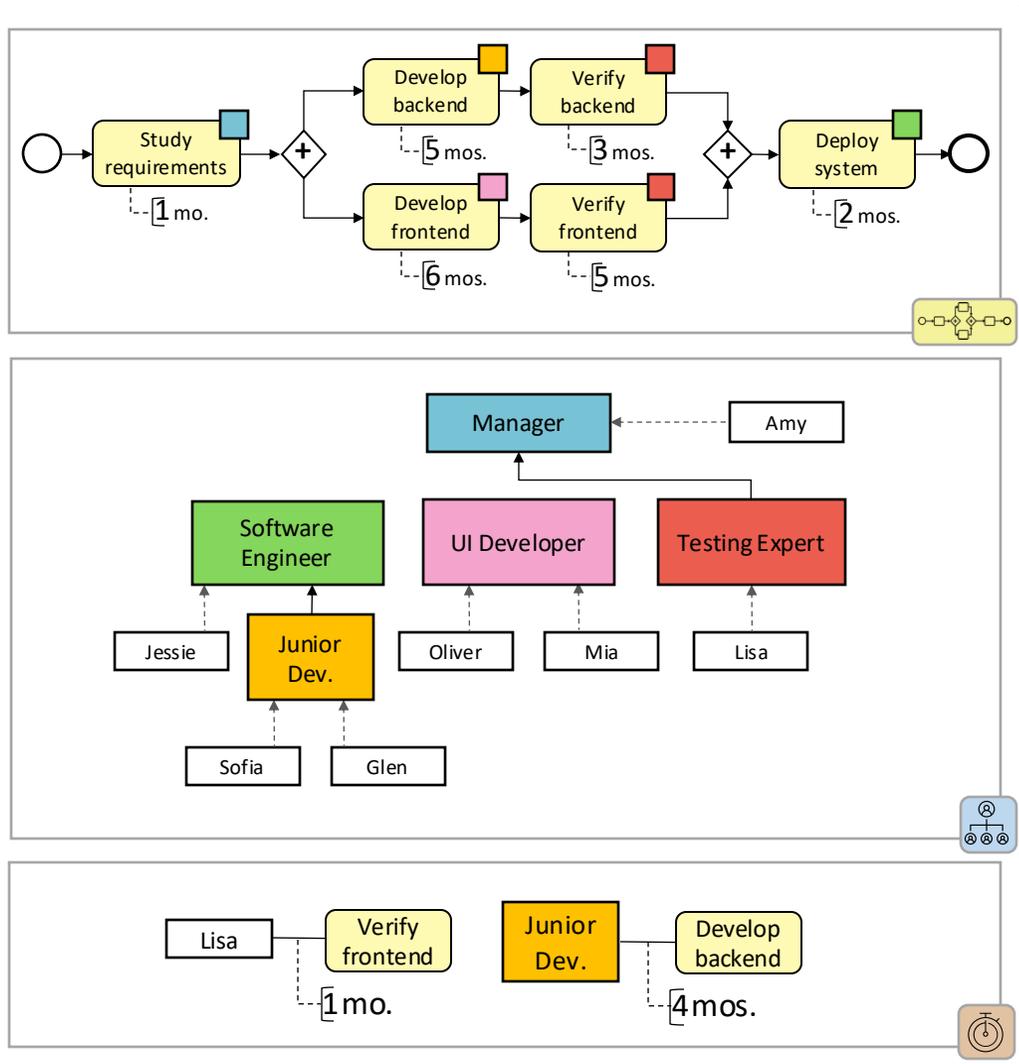
# BPMS with RABP Support



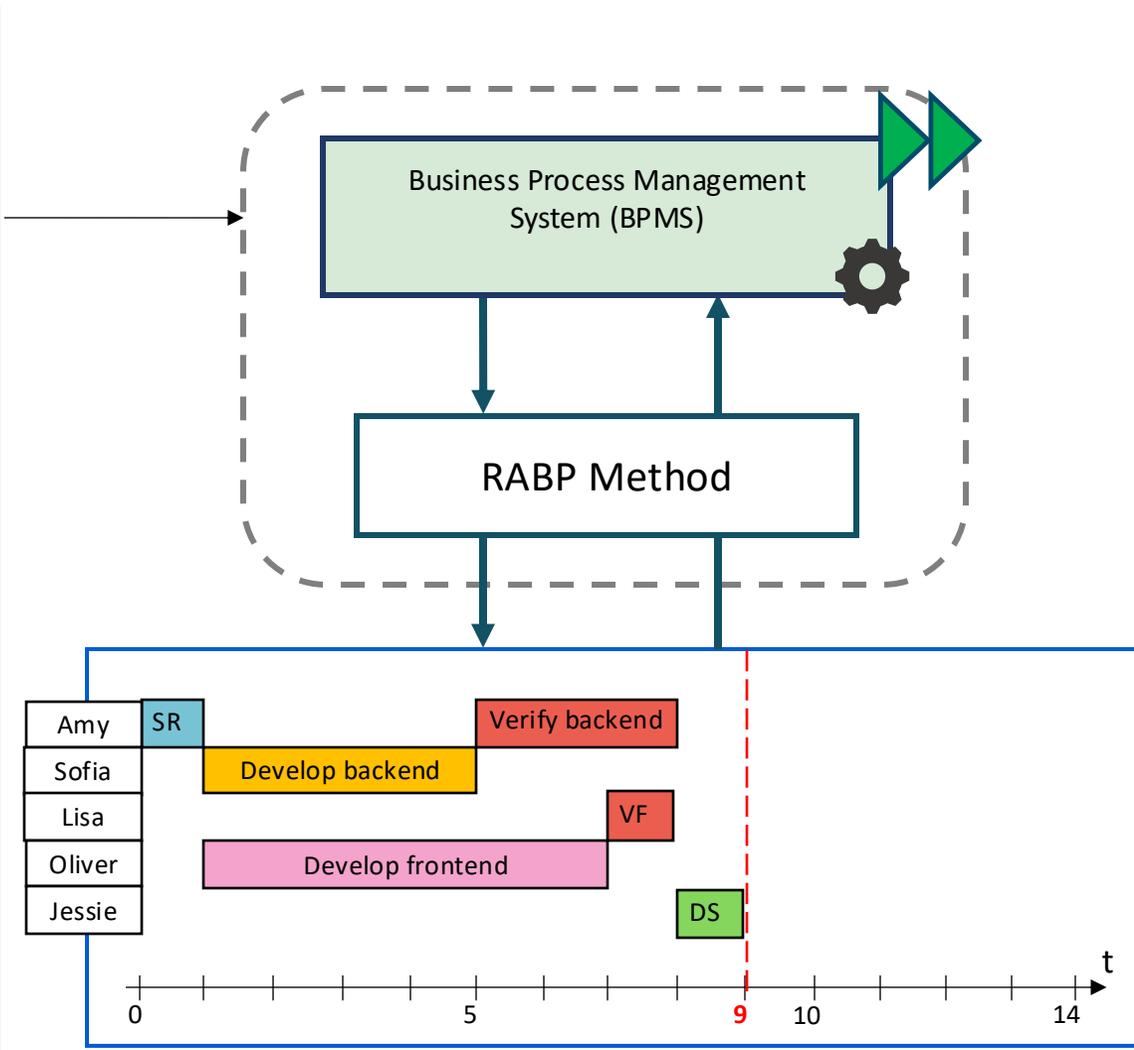
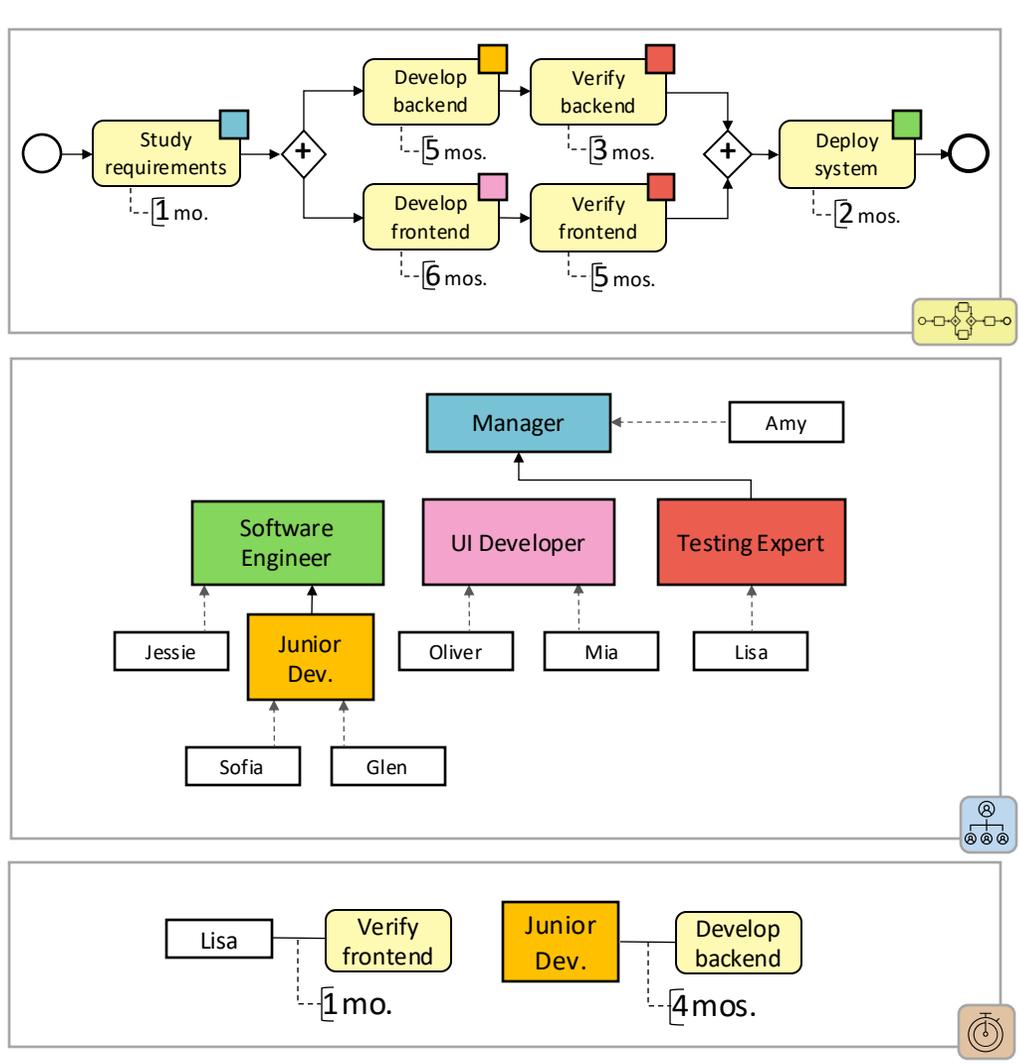
# BPMS with RABP Support



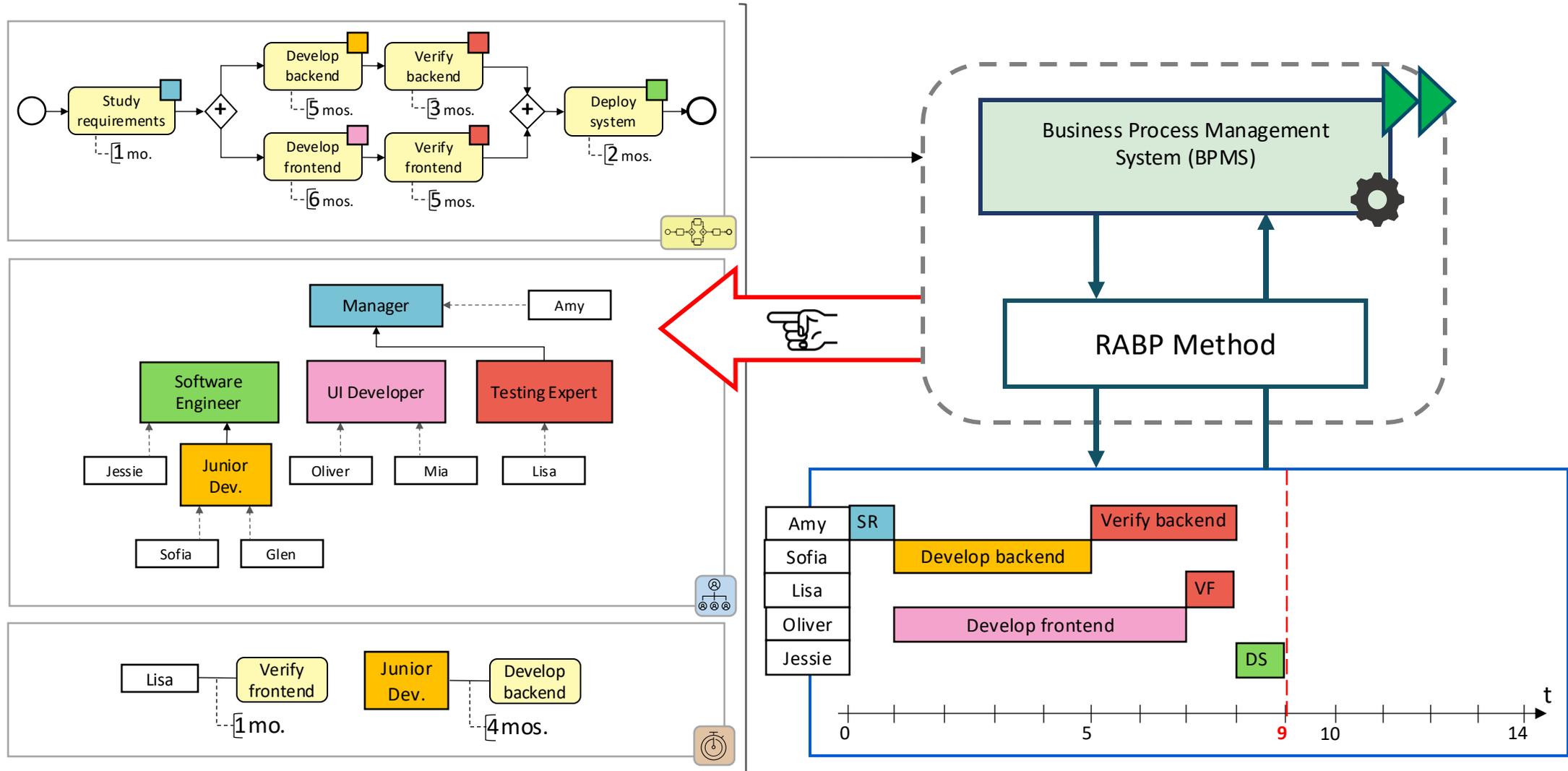
# BPMS with RABP Support



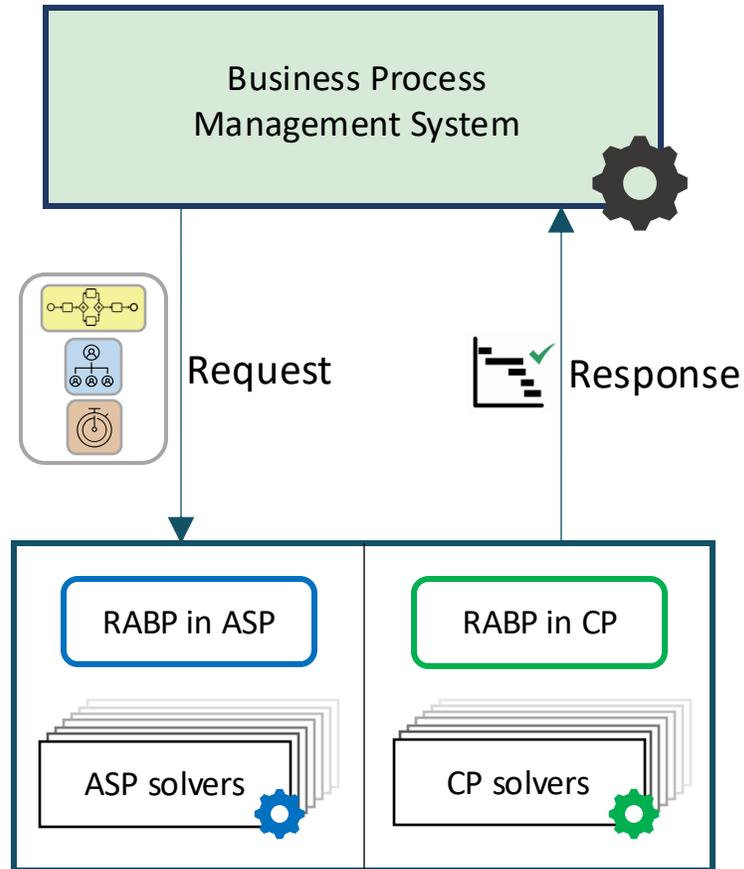
# BPMS with RABP Support



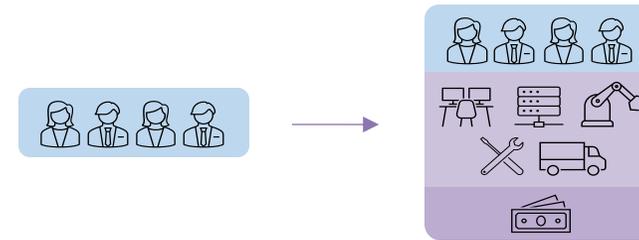
# BPMS with RABP Support



# Goals:



## 1. Representing a wide variety of resources in RABP

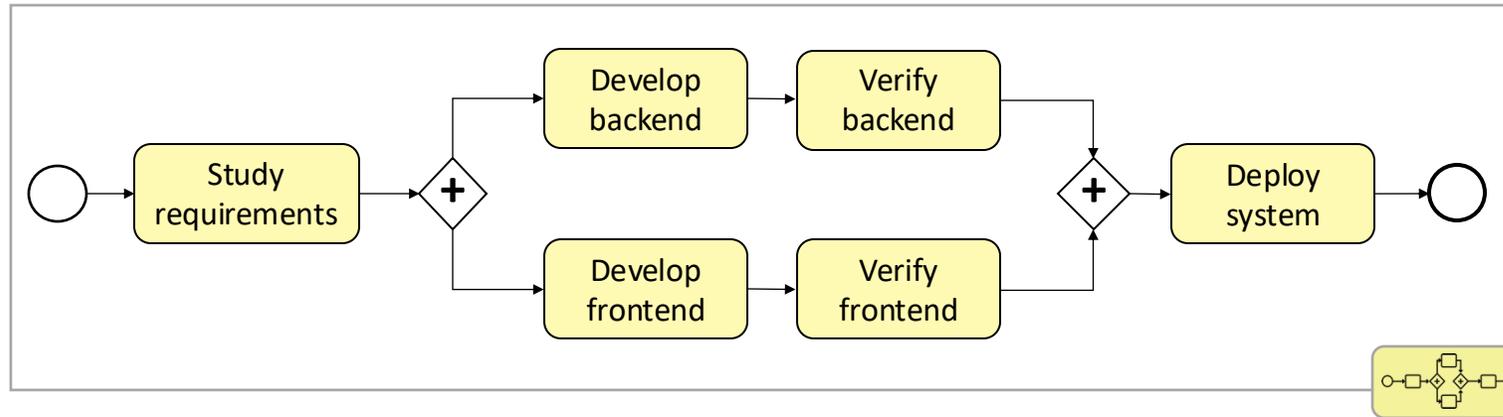


## 2. Selecting suitable KRR formalisms for implementing RABP

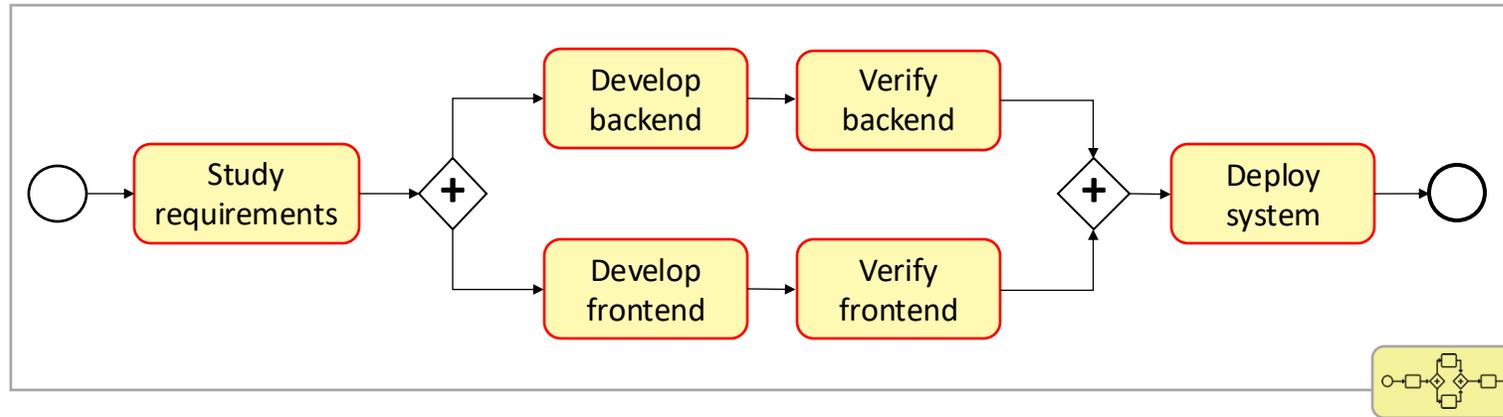
- Declarative formalisms
  - Answer Set Programming (ASP)
  - Constraint Programming (CP)

## 3. Devising a realistic benchmark for testing RABP methods

# BPM... Ease of encoding in ASP

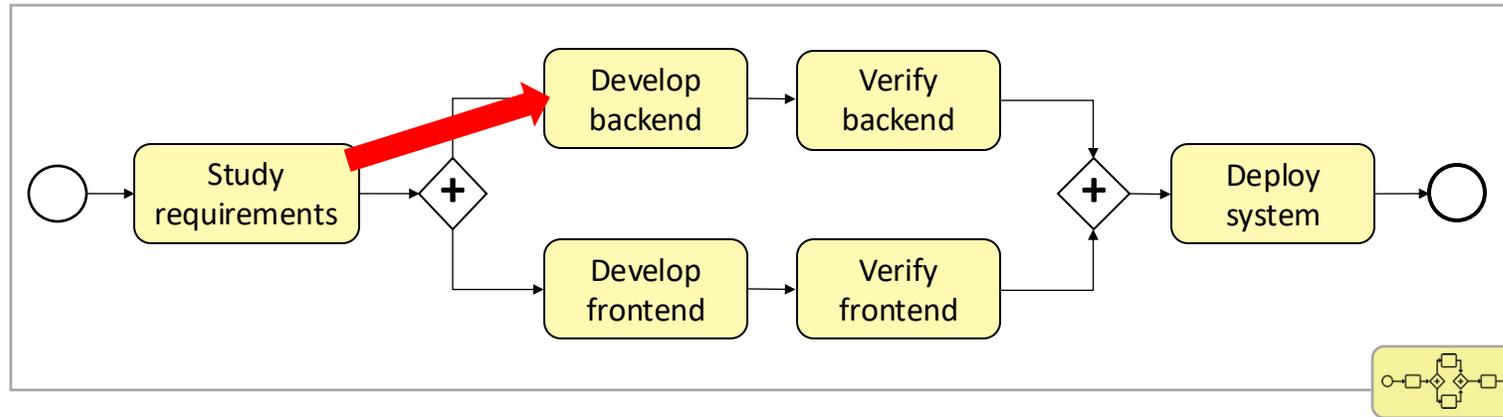


# BPM... Ease of encoding in ASP



```
activity("StudyRequirements").  
activity("DevelopBackend").  
activity("VerifyBackend").  
activity("DevelopFrontend").  
activity("VerifyFrontend").  
activity("DeploySystem").
```

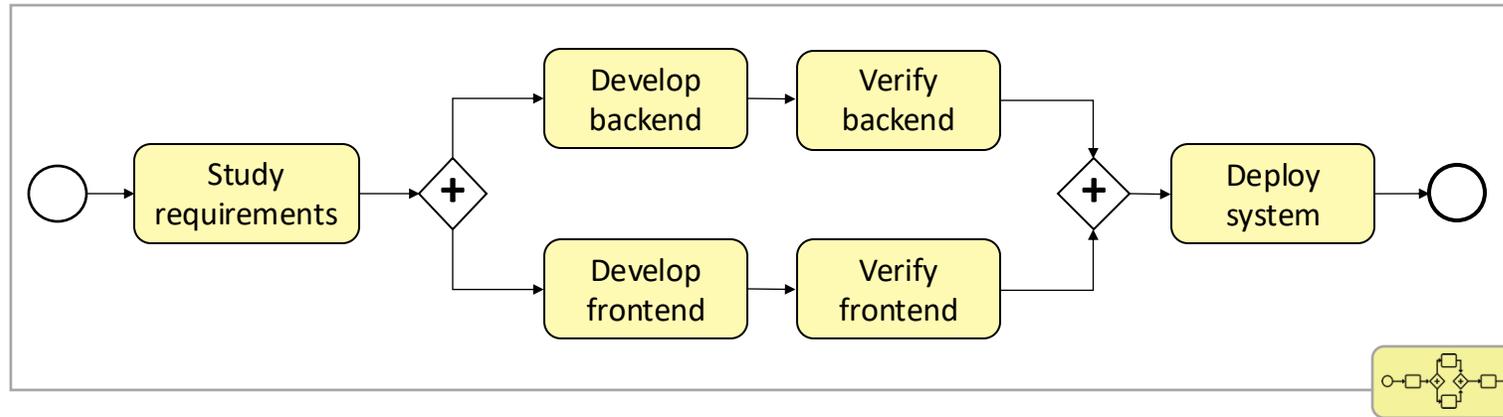
# BPM... Ease of encoding in ASP



```
activity("StudyRequirements").  
activity("DevelopBackend").  
activity("VerifyBackend").  
activity("DevelopFrontend").  
activity("VerifyFrontend").  
activity("DeploySystem").
```

```
dprec("StudyRequirements", "DevelopBackend").
```

# BPM... Ease of encoding in ASP

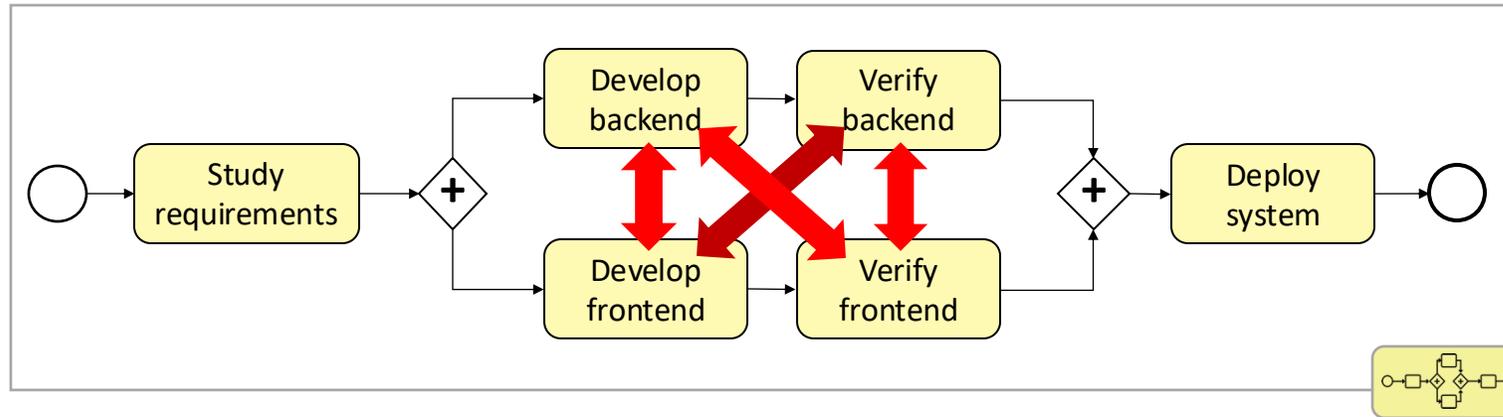


activity("StudyRequirements").  
activity("DevelopBackend").  
activity("VerifyBackend").  
activity("DevelopFrontend").  
activity("VerifyFrontend").  
activity("DeploySystem").

dprec("StudyRequirements", "DevelopBackend").  
dprec("StudyRequirements", "DevelopFrontend").  
dprec("DevelopBackend", "VerifyBackend").  
dprec("DevelopFrontend", "VerifyFrontend").  
dprec("VerifyBackend", "DeploySystem").  
dprec("VerifyFrontend", "DeploySystem").

# Contributions to RQ1

## Representation of Processes in ASP

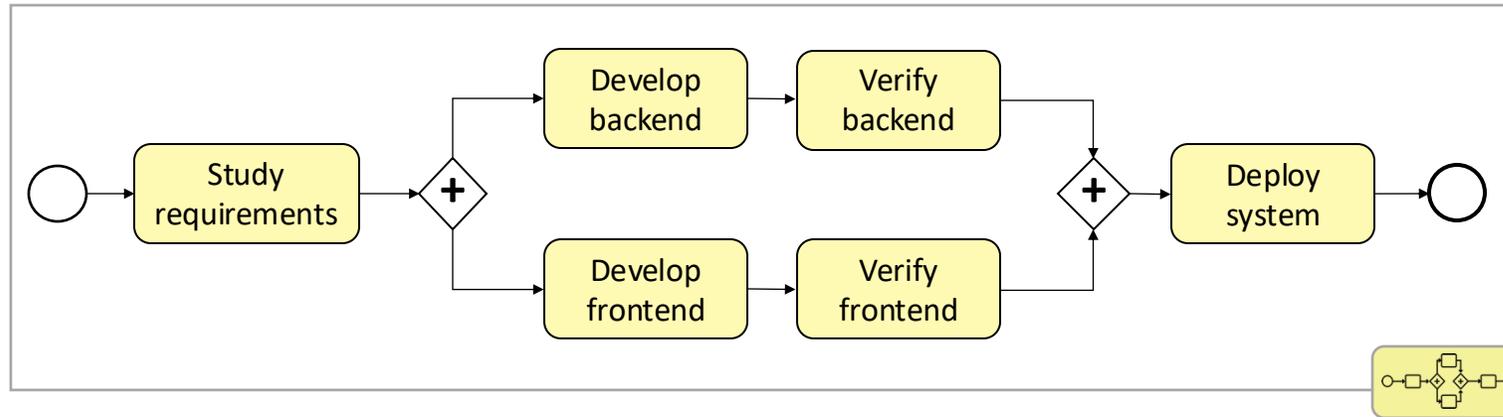


activity("StudyRequirements").  
 activity("DevelopBackend").  
 activity("VerifyBackend").  
 activity("DevelopFrontend").  
 activity("VerifyFrontend").  
 activity("DeploySystem").

dprec("StudyRequirements", "DevelopBackend").  
 dprec("StudyRequirements", "DevelopFrontend").  
 dprec("DevelopBackend", "VerifyBackend").  
 dprec("DevelopFrontend", "VerifyFrontend").  
 dprec("VerifyBackend", "DeploySystem").  
 dprec("VerifyFrontend", "DeploySystem").

conc("DevelopBackend", "DevelopFrontend").  
 conc("DevelopBackend", "VerifyFrontend").  
 conc("DevelopFrontend", "DevelopBackend").  
 conc("DevelopFrontend", "VerifyBackend").

# BPM... Ease of encoding in ASP

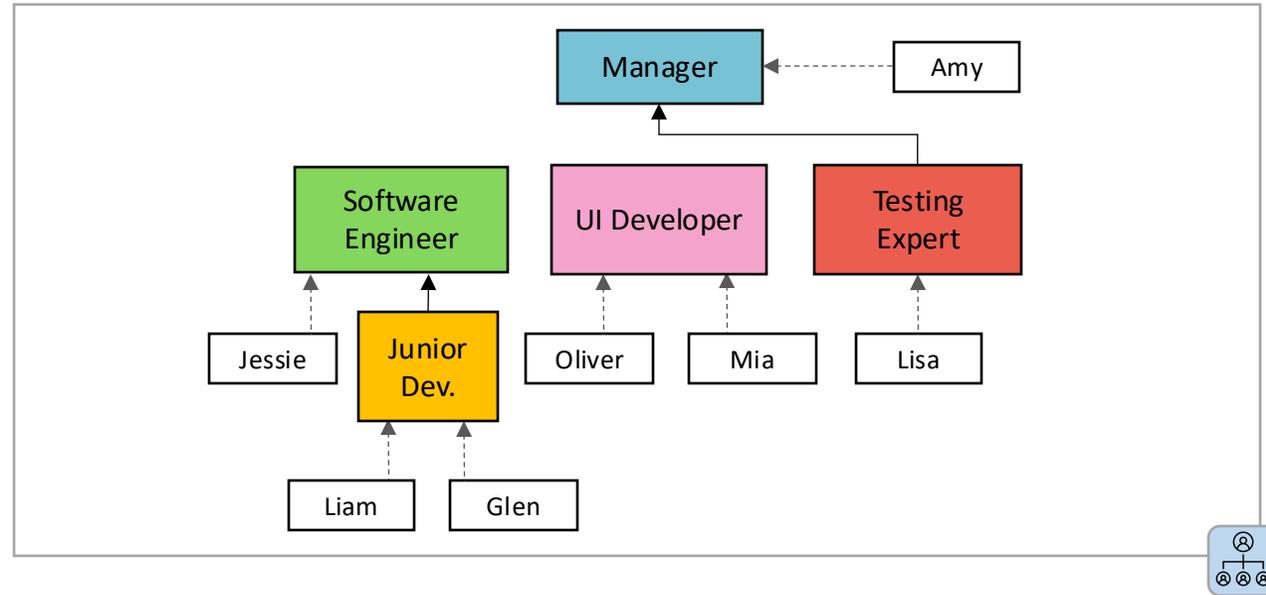


```
activity("StudyRequirements").  
activity("DevelopBackend").  
activity("VerifyBackend").  
activity("DevelopFrontend").  
activity("VerifyFrontend").  
activity("DeploySystem").
```

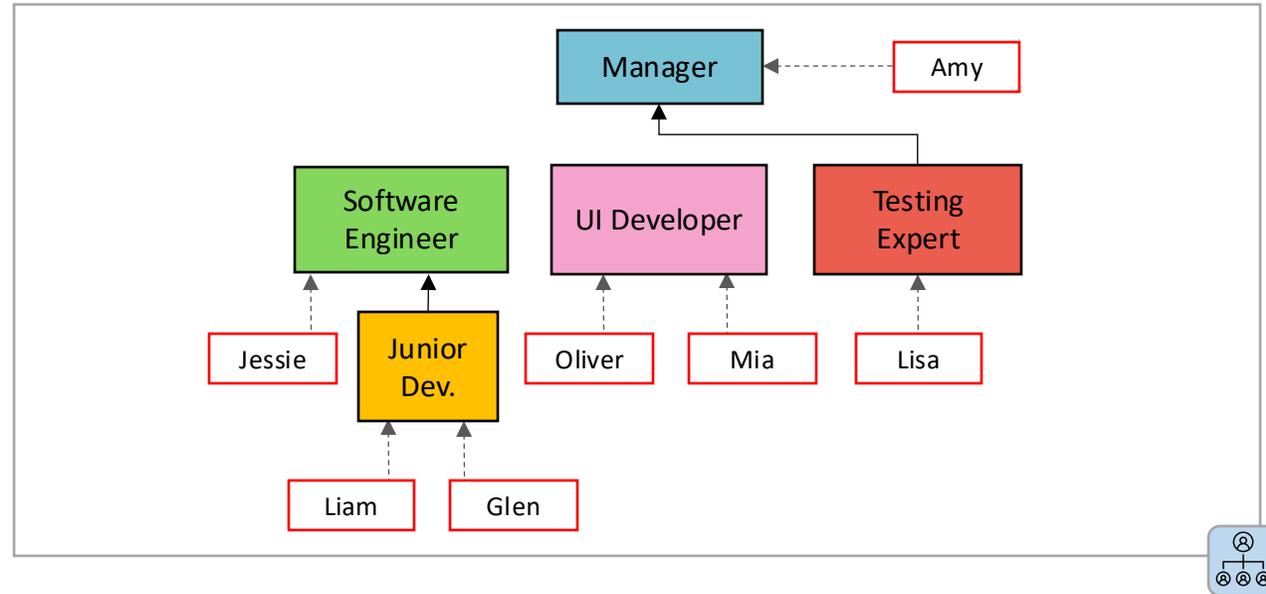
```
dprec("StudyRequirements", "DevelopBackend").  
dprec("StudyRequirements", "DevelopFrontend").  
dprec("DevelopBackend", "VerifyBackend").  
dprec("DevelopFrontend", "VerifyFrontend").  
dprec("VerifyBackend", "DeploySystem").  
dprec("VerifyFrontend", "DeploySystem").
```

```
conc("DevelopBackend", "DevelopFrontend").  
conc("DevelopBackend", "VerifyFrontend").  
conc("DevelopFrontend", "DevelopBackend").  
conc("DevelopFrontend", "VerifyBackend").
```

# BPM... Ease of encoding in ASP

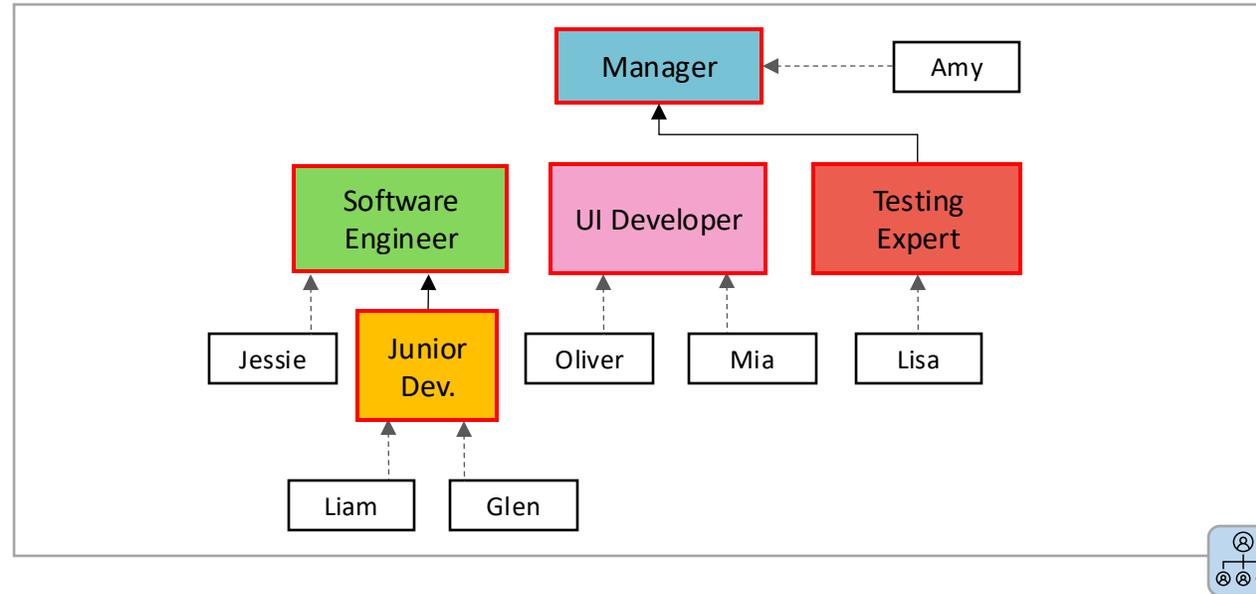


# BPM... Ease of encoding in ASP



```
resource("Amy").  
resource("Jessie").  
resource("Liam").  
resource("Glen").  
resource("Oliver").  
resource("Mia").  
resource("Lisa").
```

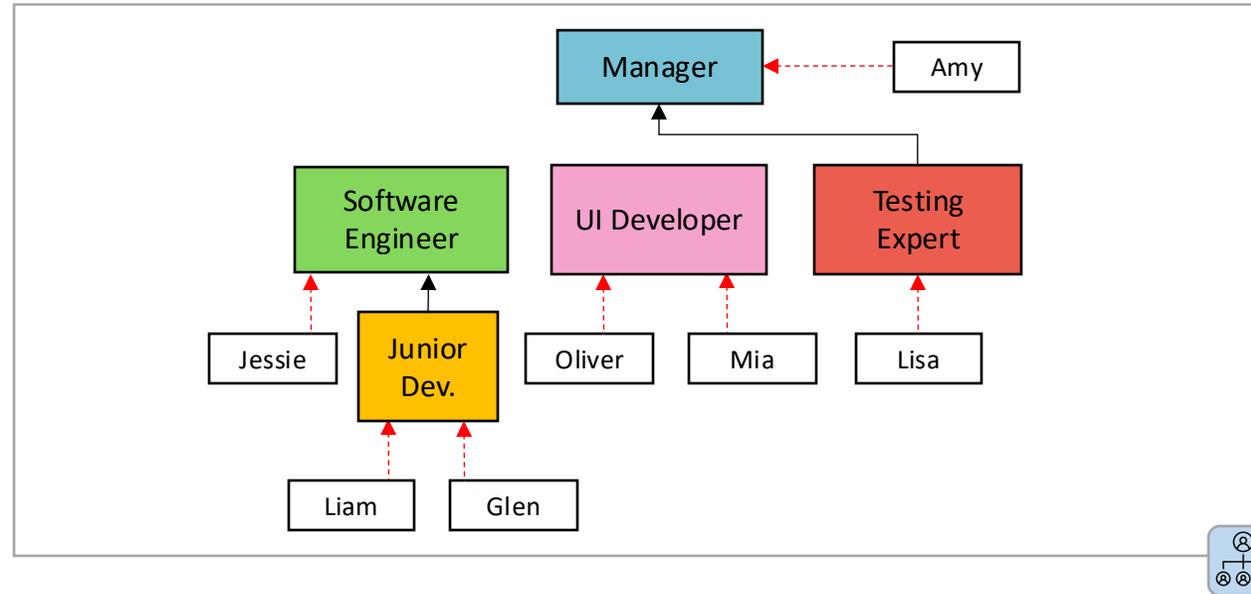
# BPM... Ease of encoding in ASP



```
resource("Amy").  
resource("Jessie").  
resource("Liam").  
resource("Glen").  
resource("Oliver").  
resource("Mia").  
resource("Lisa").
```

```
role("Manager").  
role("SoftwareEngineer").  
role("JuniorDeveloper").  
role("UIDeveloper").  
role("TestingExpert").
```

# BPM... Ease of encoding in ASP

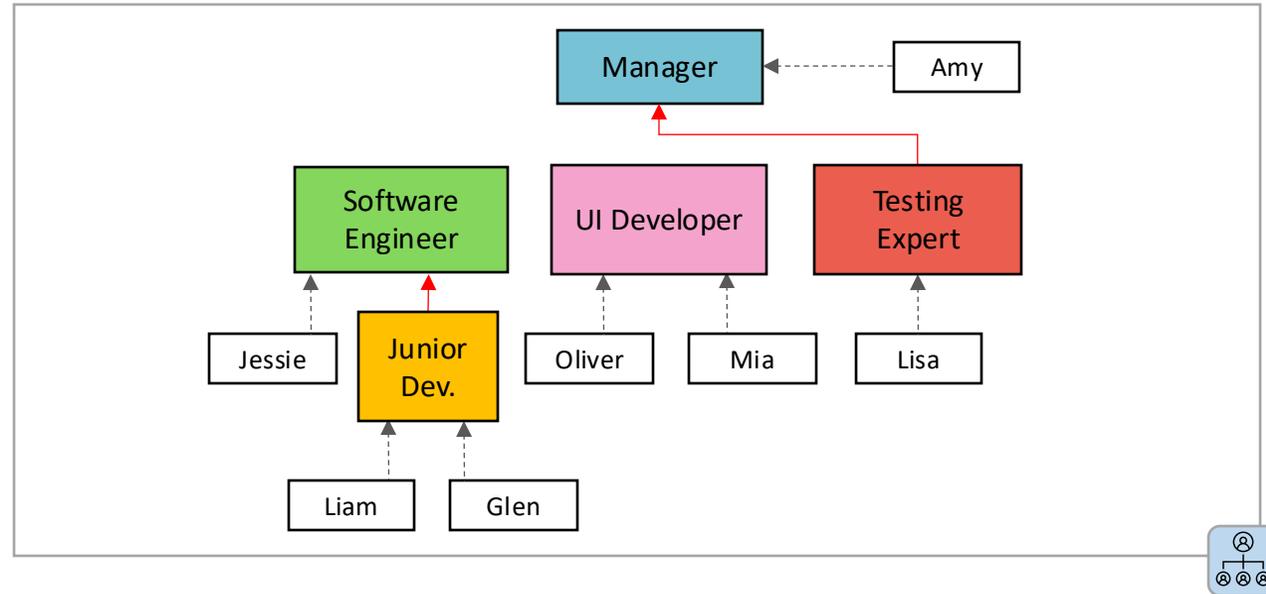


```
resource("Amy").  
resource("Jessie").  
resource("Liam").  
resource("Glen").  
resource("Oliver").  
resource("Mia").  
resource("Lisa").
```

```
role("Manager").  
role("SoftwareEngineer").  
role("JuniorDeveloper").  
role("UIDeveloper").  
role("TestingExpert").
```

```
rIAC("Amy", "Manager").  
rIAC("Jessie", "SoftwareEngineer").  
rIAC("Liam", "JuniorDeveloper").  
rIAC("Glen", "JuniorDeveloper").  
rIAC("Oliver", "UIDeveloper").  
rIAC("Mia", "UIDeveloper").  
rIAC("Lisa", "TestingExpert").
```

# BPM... Ease of encoding in ASP



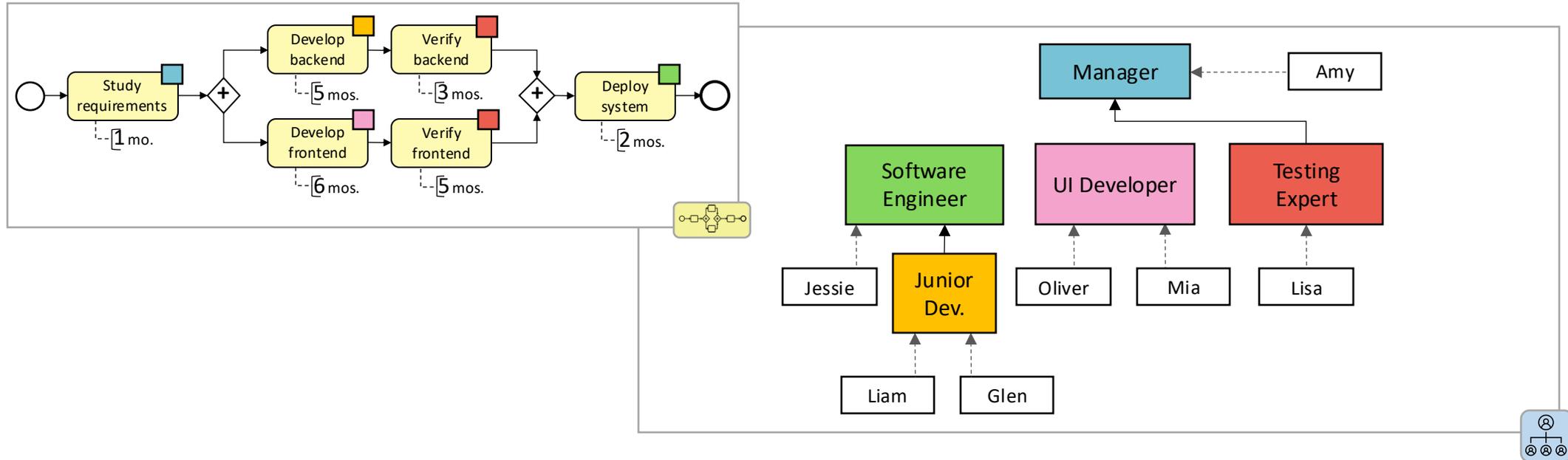
```
resource("Amy").  
resource("Jessie").  
resource("Liam").  
resource("Glen").  
resource("Oliver").  
resource("Mia").  
resource("Lisa").
```

```
role("Manager").  
role("SoftwareEngineer").  
role("JuniorDeveloper").  
role("UIDeveloper").  
role("TestingExpert").
```

```
rIAC("Amy", "Manager").  
rIAC("Jessie", "SoftwareEngineer").  
rIAC("Liam", "JuniorDeveloper").  
rIAC("Glen", "JuniorDeveloper").  
rIAC("Oliver", "UIDeveloper").  
rIAC("Mia", "UIDeveloper").  
rIAC("Lisa", "TestingExpert").
```

```
IIAC("Manager", "TestingExpert").  
IIAC("SoftwareEngineer", "JuniorDeveloper").
```

# BPM... Ease of encoding in ASP



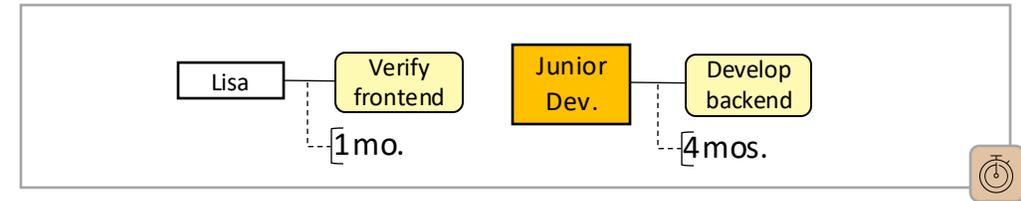
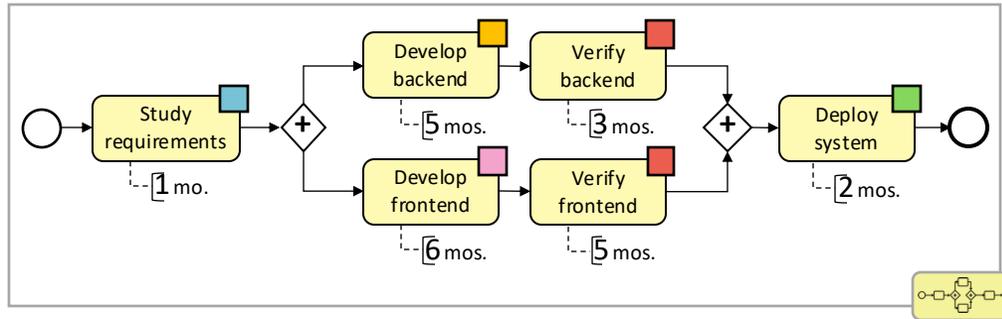
resource("Amy").  
 resource("Jessie").  
 resource("Liam").  
 resource("Glen").  
 resource("Oliver").  
 resource("Mia").  
 resource("Lisa").

role("Manager").  
 role("SoftwareEngineer").  
 role("JuniorDeveloper").  
 role("UIDeveloper").  
 role("TestingExpert").

rIAC("Amy", "Manager").  
 rIAC("Jessie", "SoftwareEngineer").  
 rIAC("Liam", "JuniorDeveloper").  
 rIAC("Glen", "JuniorDeveloper").  
 rIAC("Oliver", "UIDeveloper").  
 rIAC("Mia", "UIDeveloper").  
 rIAC("Lisa", "TestingExpert").

IIAC("Manager", "TestingExpert").  
 IIAC("SoftwareEngineer", "JuniorDeveloper").  
 aIAC("StudyRequirements", "Manager").  
 aIAC("DevelopBackend", "JuniorDeveloper").  
 aIAC("VerifyBackend", "TestingExpert").  
 aIAC("DevelopFrontend", "UIDeveloper").  
 aIAC("VerifyFrontend", "TestingExpert").  
 aIAC("DeploySystem", "SoftwareEngineer").

# BPM... Ease of encoding in ASP

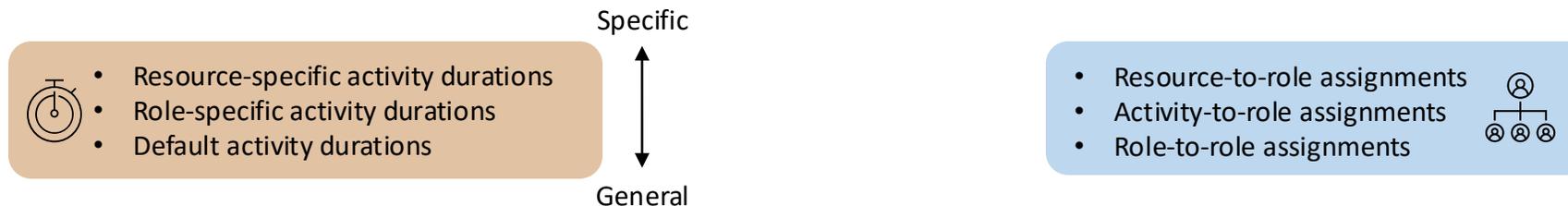


```

defaultDuration("StudyRequirements",1).
defaultDuration("DevelopBackend",5).
defaultDuration("VerifyBackend",3).
defaultDuration("DevelopFrontend",6).
defaultDuration("VerifyFrontend",5).
defaultDuration("DeploySystem",2).
  
```

```

rsaDuration("Lisa","VerifyFrontend",1).
lsaDuration("JuniorDeveloper","DevelopBackend",4).
  
```



```

allowedRAD(R,A,D) :- rsaDuration(R,A,D), rIAC(R,L), aIAC(A,L).
allowedRAD(R,A,D) :- lsaDuration(L,A,D), not rsaDuration(R,A,_), rIAC(R,L), aIAC(A,L).
allowedRAD(R,A,D) :- defaultDuration(A,D), not rsaDuration(R,A,_), not lsaDuration(L,A,_), rIAC(R,L), aIAC(A,L).
  
```

# BPM... Guess and Check

**% generate allocations**

$1 \leq \{ \text{allocation}(R,A,S,C) : \text{time}(S), \text{time}(C), \text{allowedRAD}(R,A,D), C=S+D \} \leq 1 \text{ :- activity}(A).$

**% check for scheduling of the preceding activities**

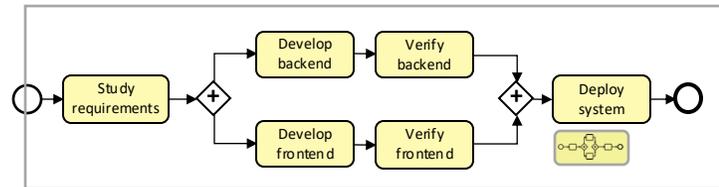
$\text{:- dPrec}(A1,A2), \text{allocation}(\_,A1,\_,C1), \text{allocation}(\_,A2,S2,\_), C1 > S2.$

**% check for scheduling of the concurrent activities**

$\text{:- conc}(A1,A2), \text{allocation}(R,A1,S1,\_), \text{allocation}(R,A2,S2,C2), S2 \leq S1, C2 > S1, A1 < A2.$

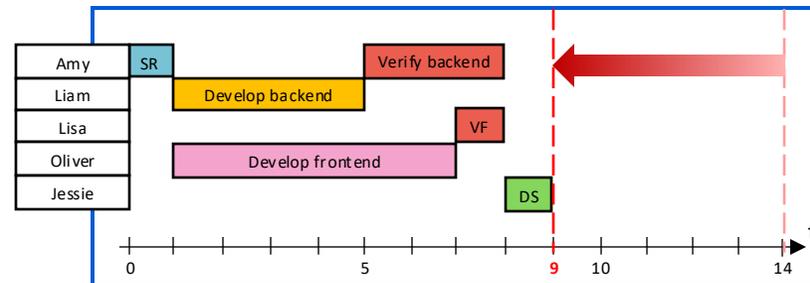
$\text{:- conc}(A1,A2), \text{allocation}(R,A1,\_,C1), \text{allocation}(R,A2,S2,C2), S2 < C1, C2 \geq C1, A1 < A2.$

$\text{:- conc}(A1,A2), \text{allocation}(R,A1,S1,C1), \text{allocation}(R,A2,S2,C2), S2 > S1, C2 < C1, A1 < A2.$



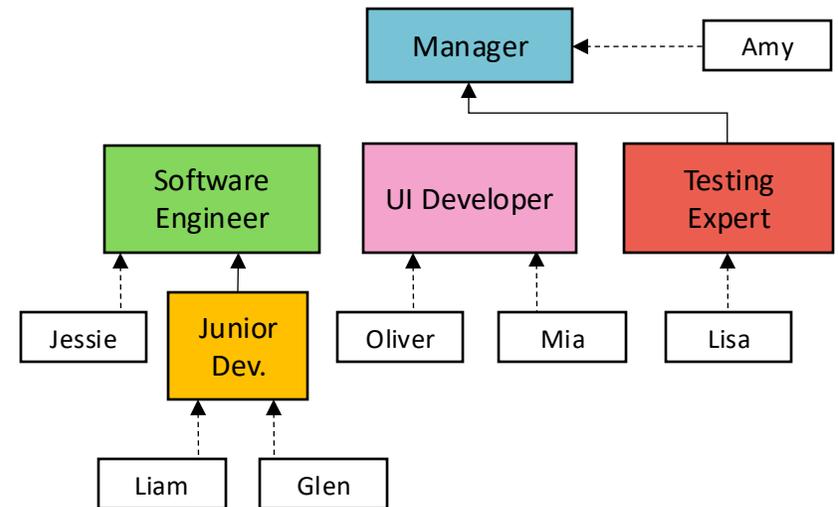
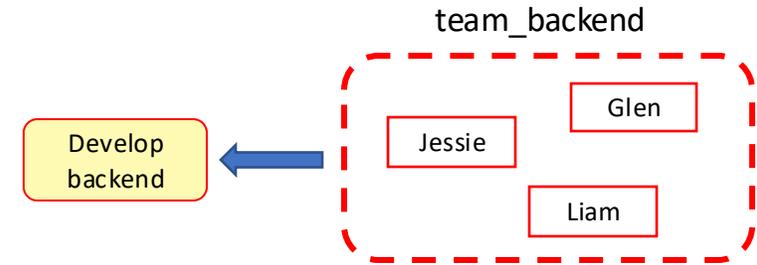
**% minimize makespan**

$\text{:- } \sim \text{makespan}(U). [U]$



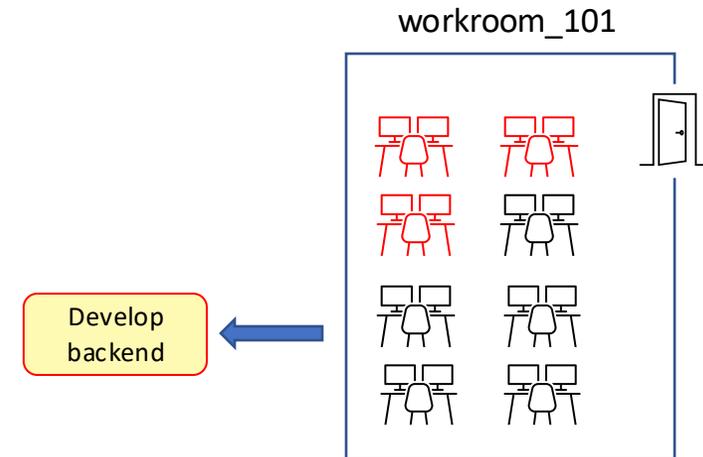
# BPM ... Easily Extensible encoding

- Teams
  - E.g., `team("team_backend",R) :- rIAC(R,"JuniorDeveloper")`.  
`team("team_backend",R) :- rIAC(R,"SoftwareEngineer")`.  
`tRequirement("DevelopBackend","team_backend")`.



# BPM ... Easily Extensible encoding

- Teams
  - **E.g.**, `team("team_backend",R) :- rIAC(R,"JuniorDeveloper").`  
`team("team_backend",R) :- rIAC(R,"SoftwareEngineer").`  
`tRequirement("DevelopBackend","team_backend").`
- Partially-renewable resources
  - **E.g.**, `pResource("workroom_101",8).`  
`pRequirement("DevelopBackend","workroom_101", N) :-`  
`N = #count{R: team("team_backend",R)}.`

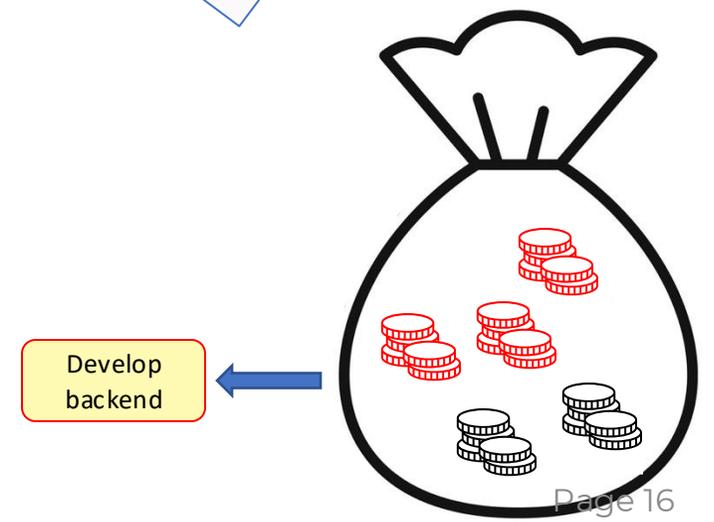


# BPM ... Easily Extensible encoding

**Resource Allocation with Dependencies in Business Process Management Systems**  
Giray Havur , Cristina Cabanillas, Jan Mendling & Axel Polleres  
Conference paper | First Online: 04 September 2016  
905 Accesses | 17 Citations  
Part of the Lecture Notes in Business Information Processing book series (LNBI, volume 260)

- Teams
  - **E.g.**, team("team\_backend",R) :- rIAC(R,"JuniorDeveloper").  
team("team\_backend",R) :- rIAC(R,"SoftwareEngineer").  
tRequirement("DevelopBackend","team\_backend").
- Partially-renewable resources
  - **E.g.**, pResource("workroom\_101",8).  
pRequirement("DevelopBackend","workroom\_101", N) :-  
N = #count{R: team("team\_backend",R)}.
- Non-renewable resources
  - **E.g.**, nResource("budget\_software\_dev\_1",50000).  
nRequirement("DevelopBackend","budget\_software\_dev\_1",N) :-  
TPM = #sum{PM,R: allocate(R,"DevelopBackend",S,C), PM=C-S},  
costPersonMonth(CPM), N = TPM \* CPM.

budget\_software\_dev\_1



# RABP in Practice: Camunda BPMS Integration

**Role Assignment Complete**

**RAL Information**

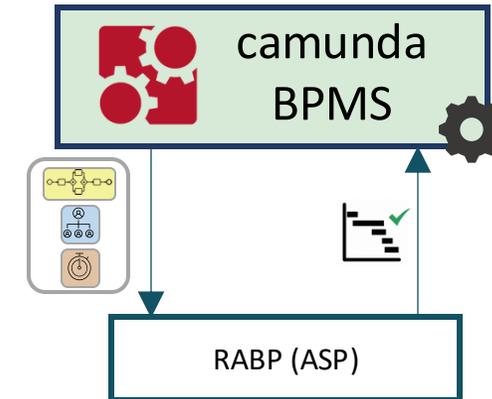
ASSESSMENT OF INPUT DOCUMENTS: IS A PHASE265  
 REVIEW: IS A PHASE265  
 CREATION OF DOCUMENT OVERVIEW: IS A PHASE265  
 CREATE DOCUMENT AND PLAN OVERVIEW: IS A PHASE265  
 INPUT DOCUMENTS AVAILABLE AND APPROVED BY CUSTOMER: IS A PHASE265

**Assigned Resources**

Solution 0

Lock	Activity	User	Start	
<input type="checkbox"/>	Assessment of Input Documents	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input type="checkbox"/>	Review	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input type="checkbox"/>	Creation of Document Overview	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input type="checkbox"/>	Create Document and Plan Overview	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input checked="" type="checkbox"/>	Input Documents Available and Approved by Customer	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input checked="" type="checkbox"/>	Creation of Checklist	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input checked="" type="checkbox"/>	Create README	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input checked="" type="checkbox"/>	Create Concept of Ordering	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input type="checkbox"/>	Coordination Development, SPP, Decision Circe-Upgrade, Establishing Technique/System Architecture	Demo Demo	08.10.2015 09:17:46	09.10.2015
<input type="checkbox"/>	Projecting SIP	Demo Demo	08.10.2015 09:17:46	09.10.2015

**Validate** **Confirm Assignment and Start Process**



**camunda cockpit** demo

Home » Phase 265

PROCESS DEFINITION: Phase 265

Version 1

**Running Instances**

- of the selected version: 2
- of all versions: 2

Filter

- by variable
- by business key
- by start date

**Process Flow:**

```

    graph TD
      Start((ProjectStart)) --> AssignProjectant[Assign Projectant]
      AssignProjectant --> AssignPhase265[Assign Phase 265]
      AssignPhase265 --> CreateProjectFolder[Create Project Folder]
      CreateProjectFolder --> CreateDocPlan[Create Document and Plan Overview]
      CreateDocPlan --> Decision{Created?}
      Decision -- Yes --> End(( ))
      Decision -- No --> AssignPhase265
  
```

**Process Instances** | **Called Process Definitions** | **Job Definitions**

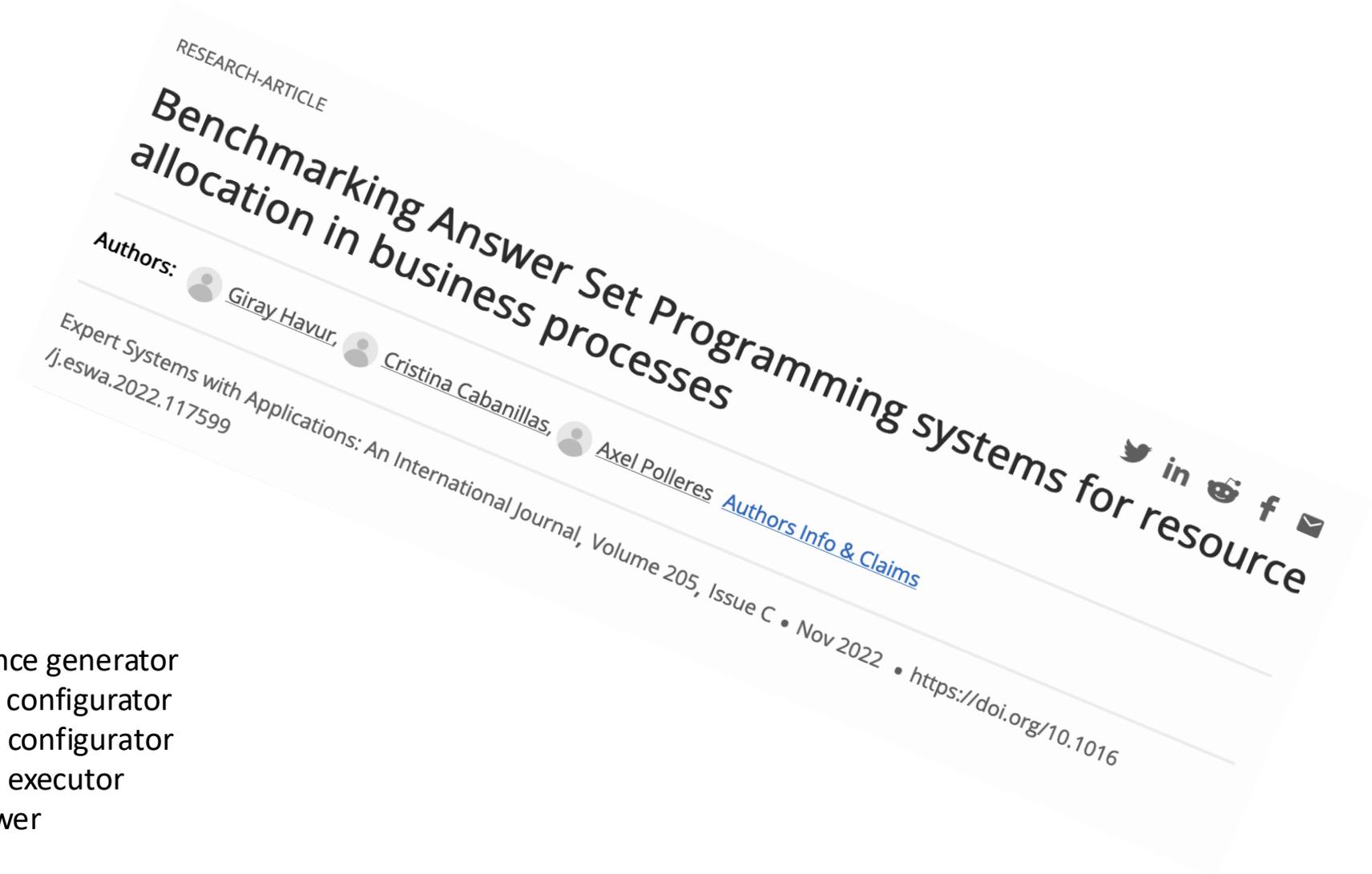
Instance ID	Start Date	End Date
24bbbcdf-4cab-11e5-a813-0...	2015-08-27T07:02:00	

Powered by camunda BPM / v7.3.0

Saimir Bala, Giray Havur, Simon Sperl, Simon Steyskal, Alois Haselbock, Jan Mendling, and Axel Polleres. Shapeworks: A BPMS extension for complex process management. In the BPM Demo Track 2016 co-located with BPM 2016, volume 1789 of CEUR Workshop Proceedings, pages 50–55, 2016.

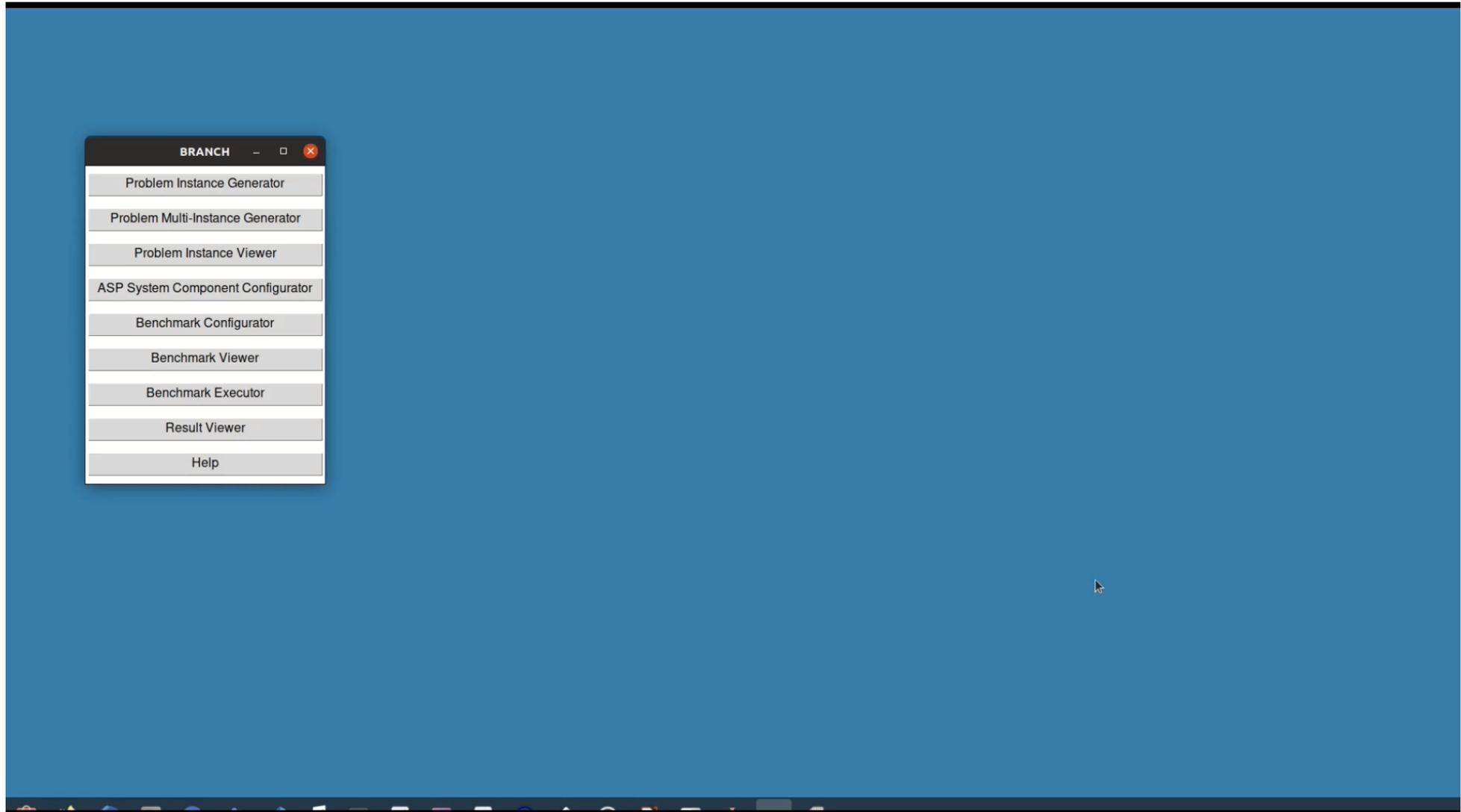
<https://www.youtube.com/watch?v=3IPxXoQR9n4>

# BRANCH: An ASP Systems Benchmark for RABP “hot from the press”

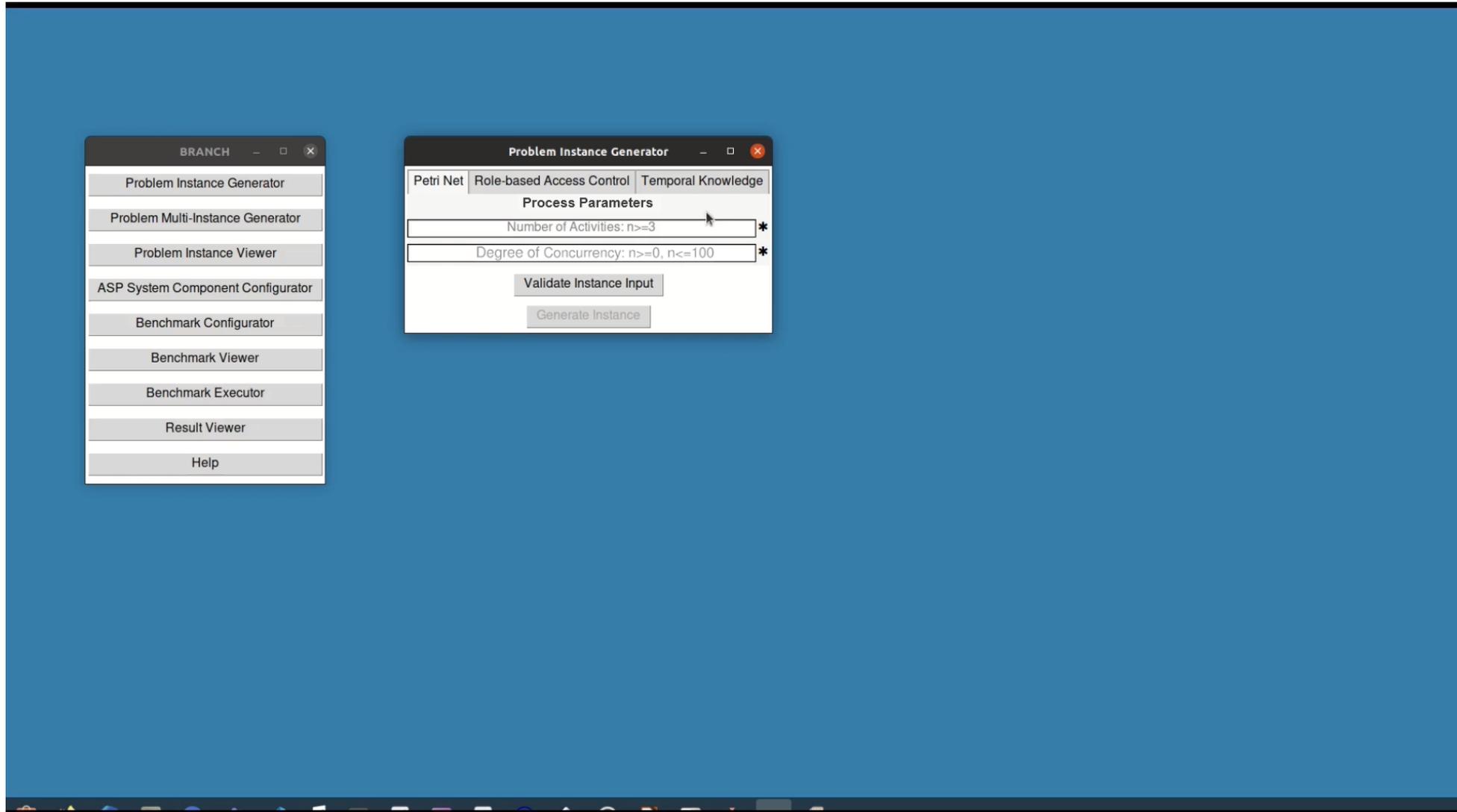


- **BRANCH**
  - RABP instance generator
  - ASP system configurator
  - Benchmark configurator
  - Benchmark executor
  - Results viewer

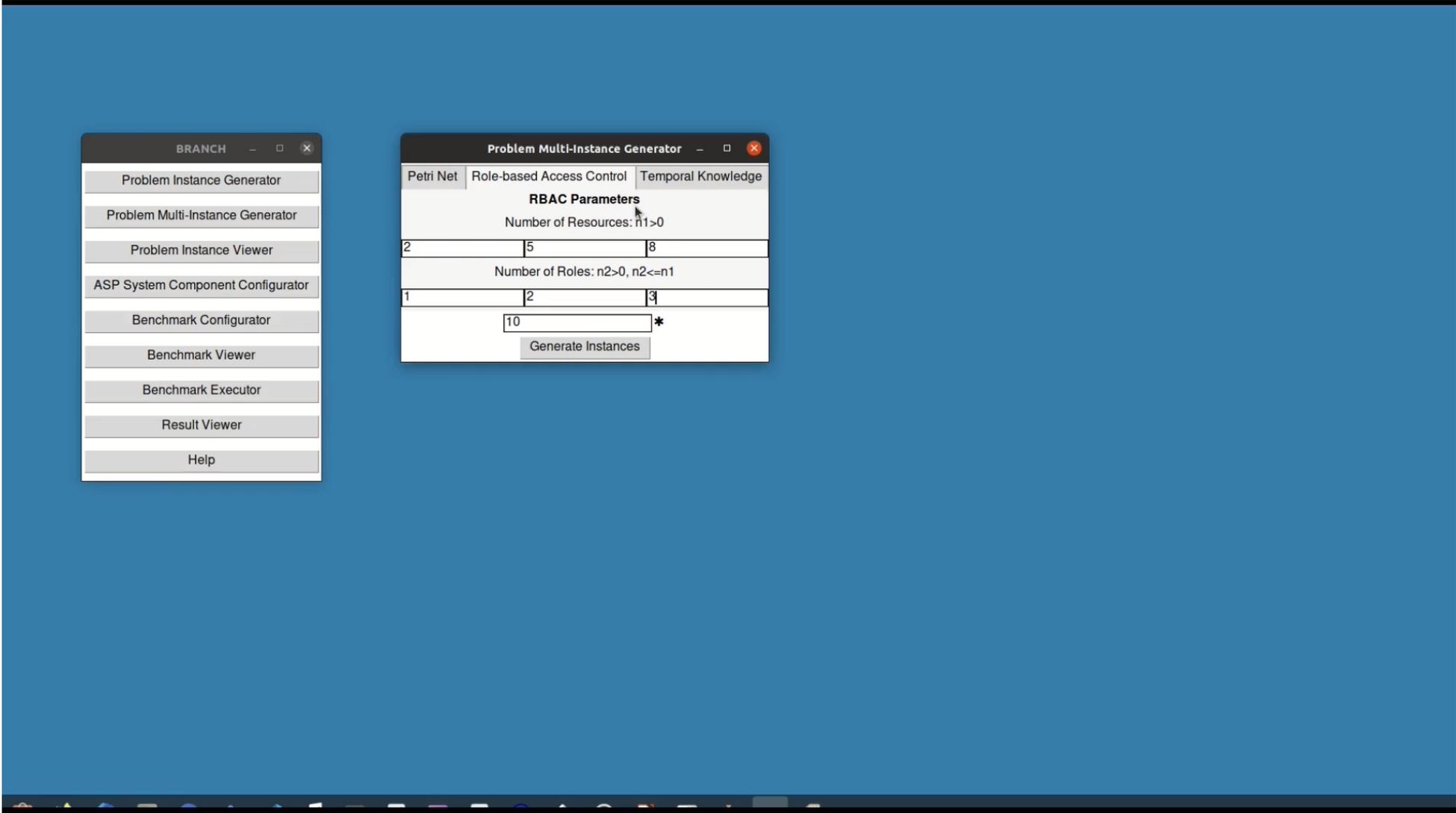
# An ASP Systems Benchmark for RABP



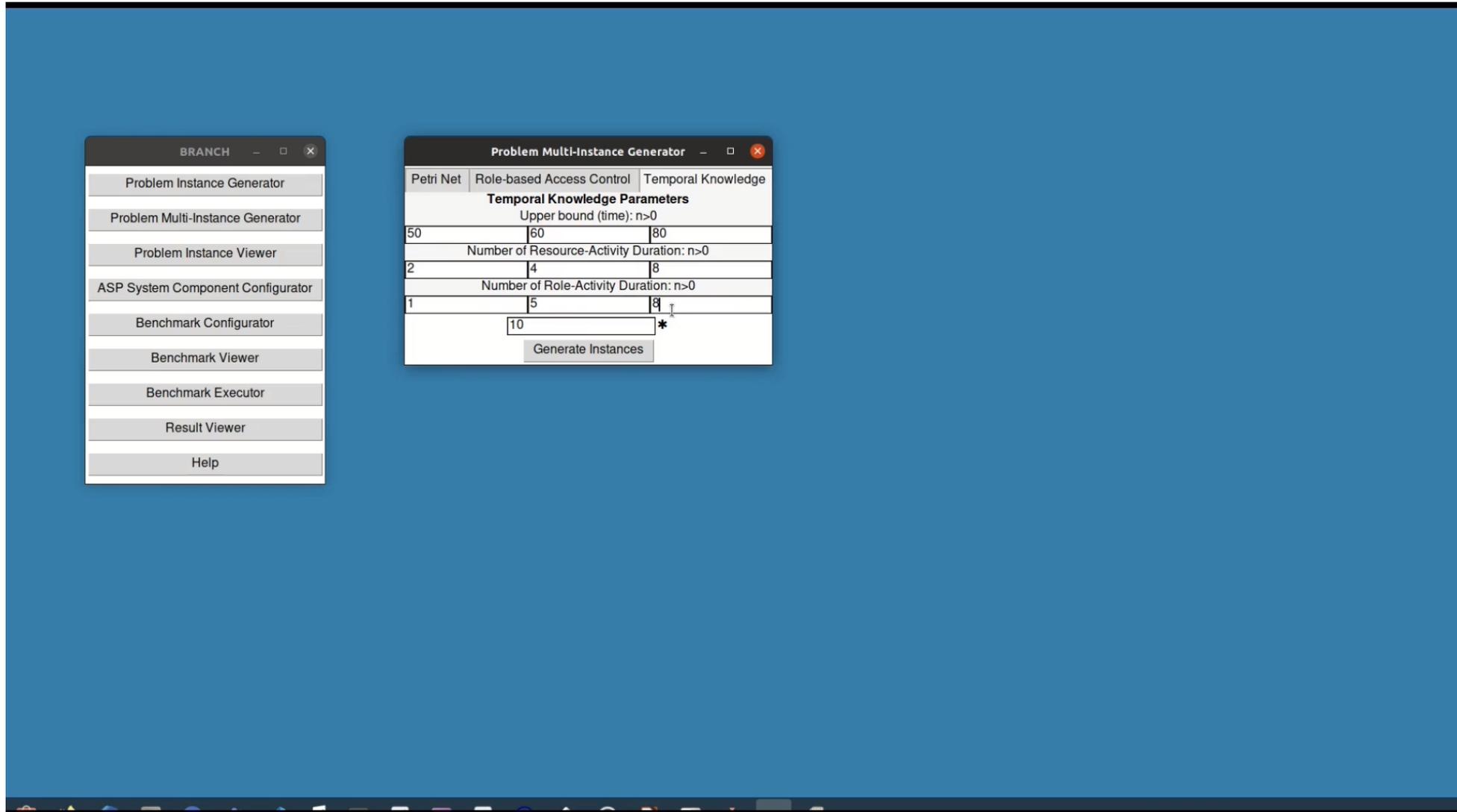
# An ASP Systems Benchmark for RABP



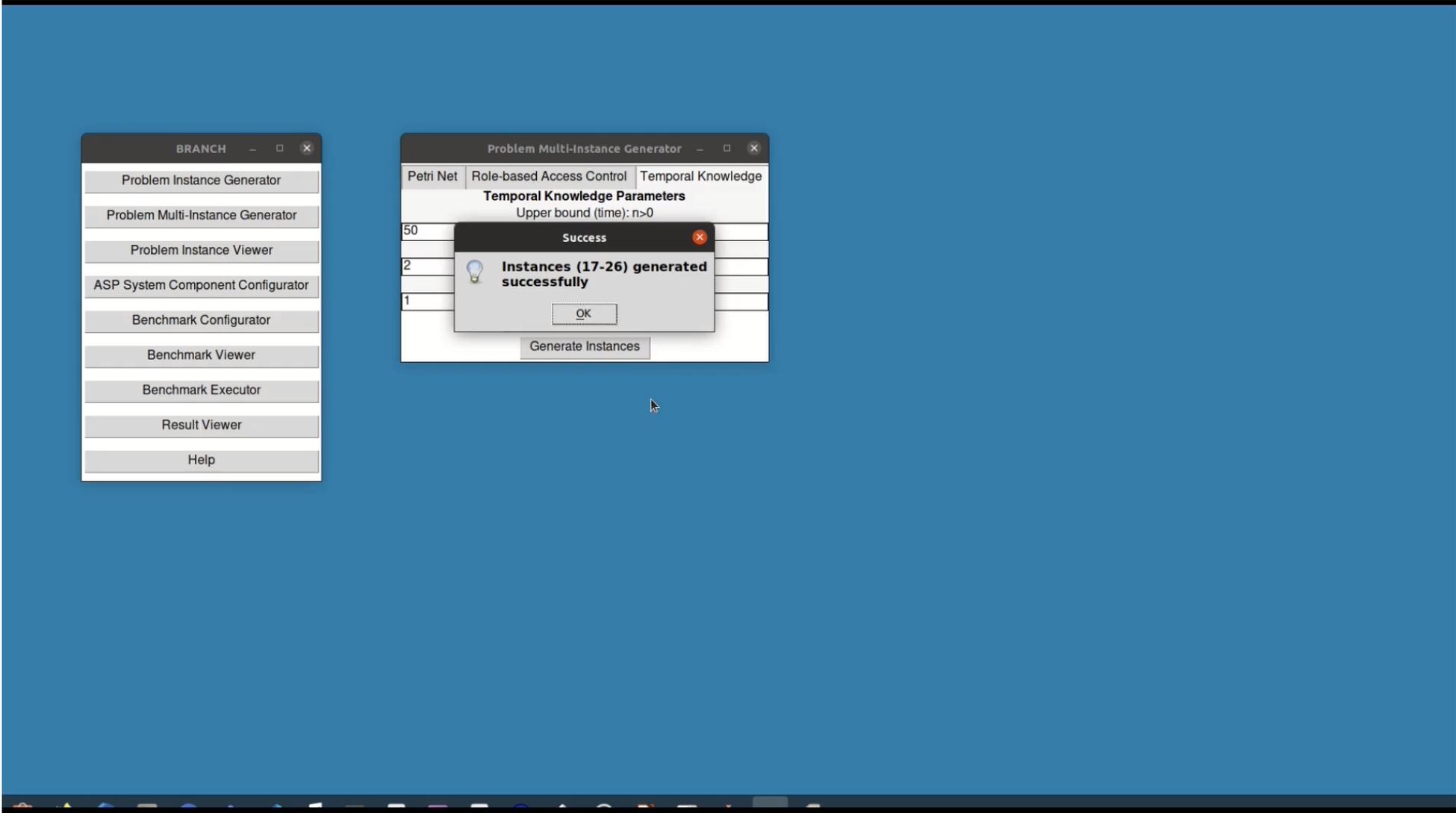
# An ASP Systems Benchmark for RABP



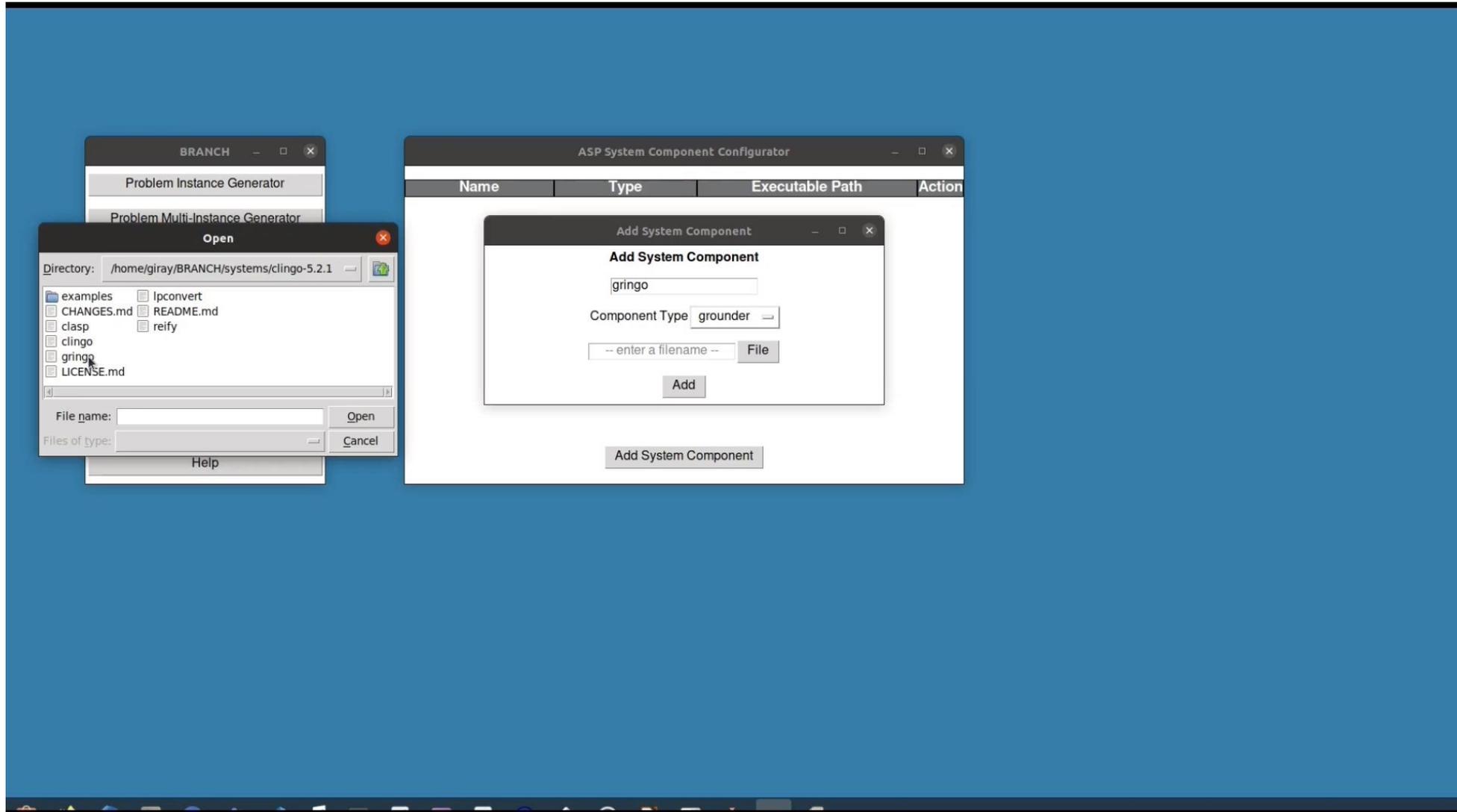
# An ASP Systems Benchmark for RABP



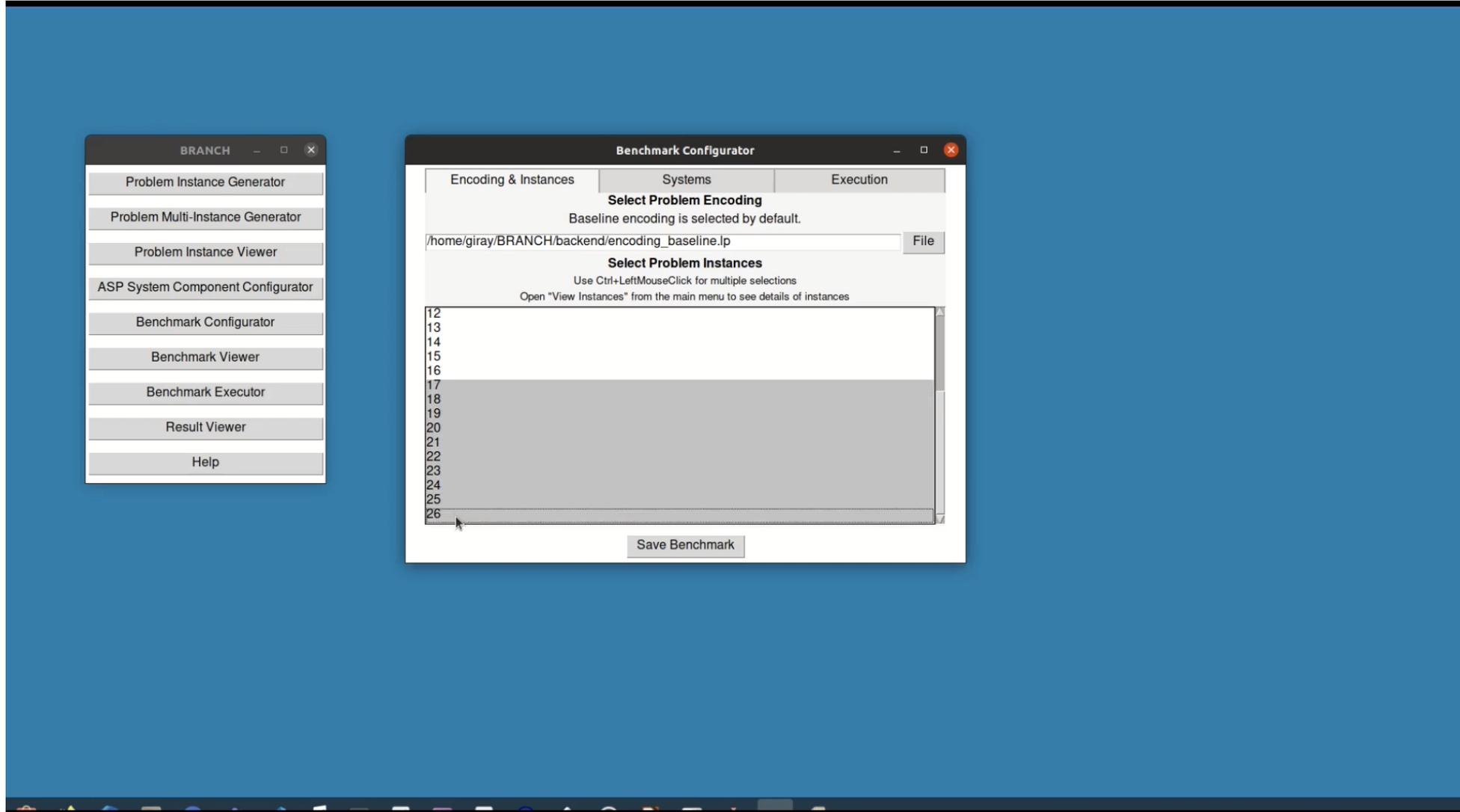
# An ASP Systems Benchmark for RABP



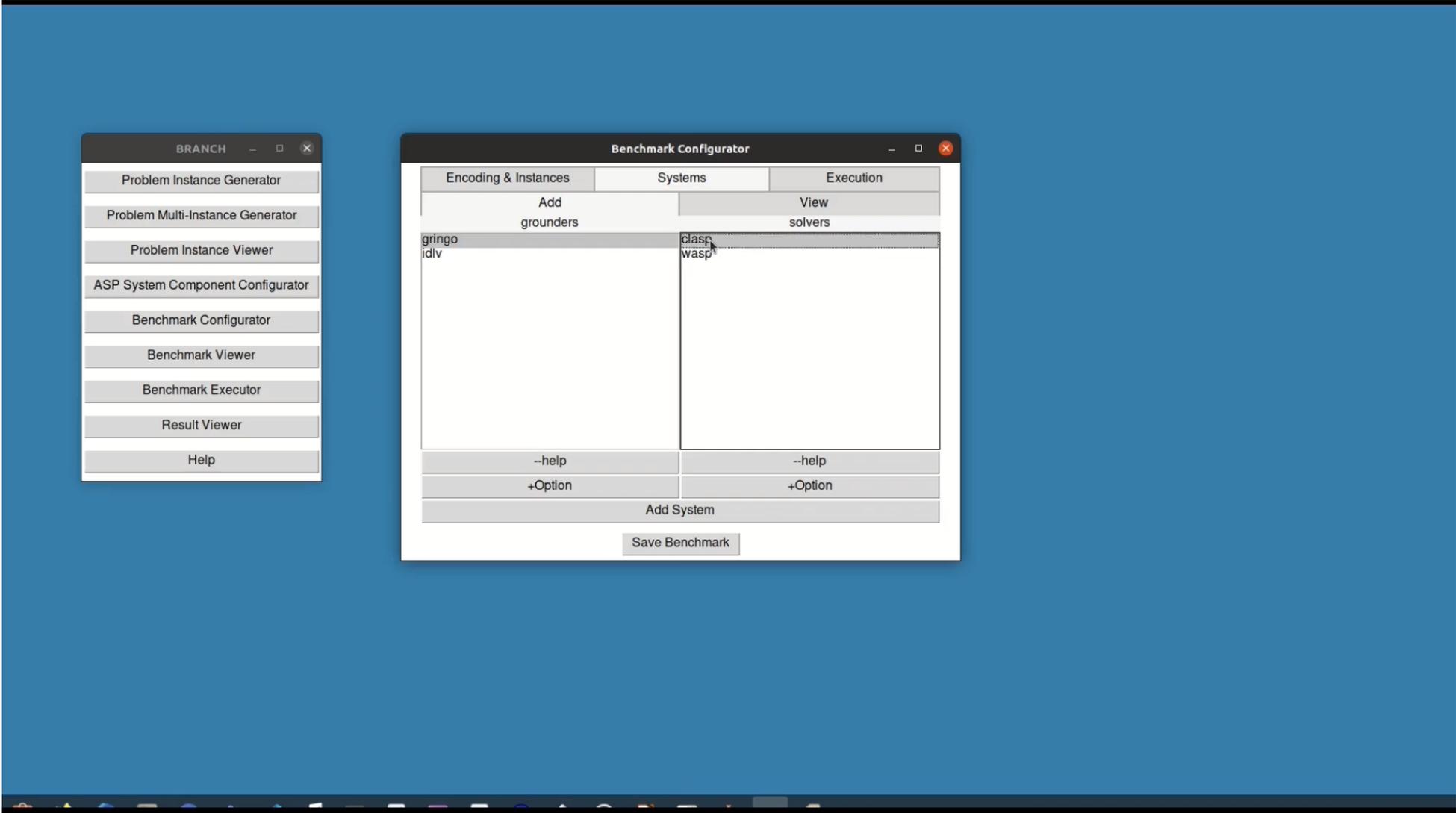
# An ASP Systems Benchmark for RABP



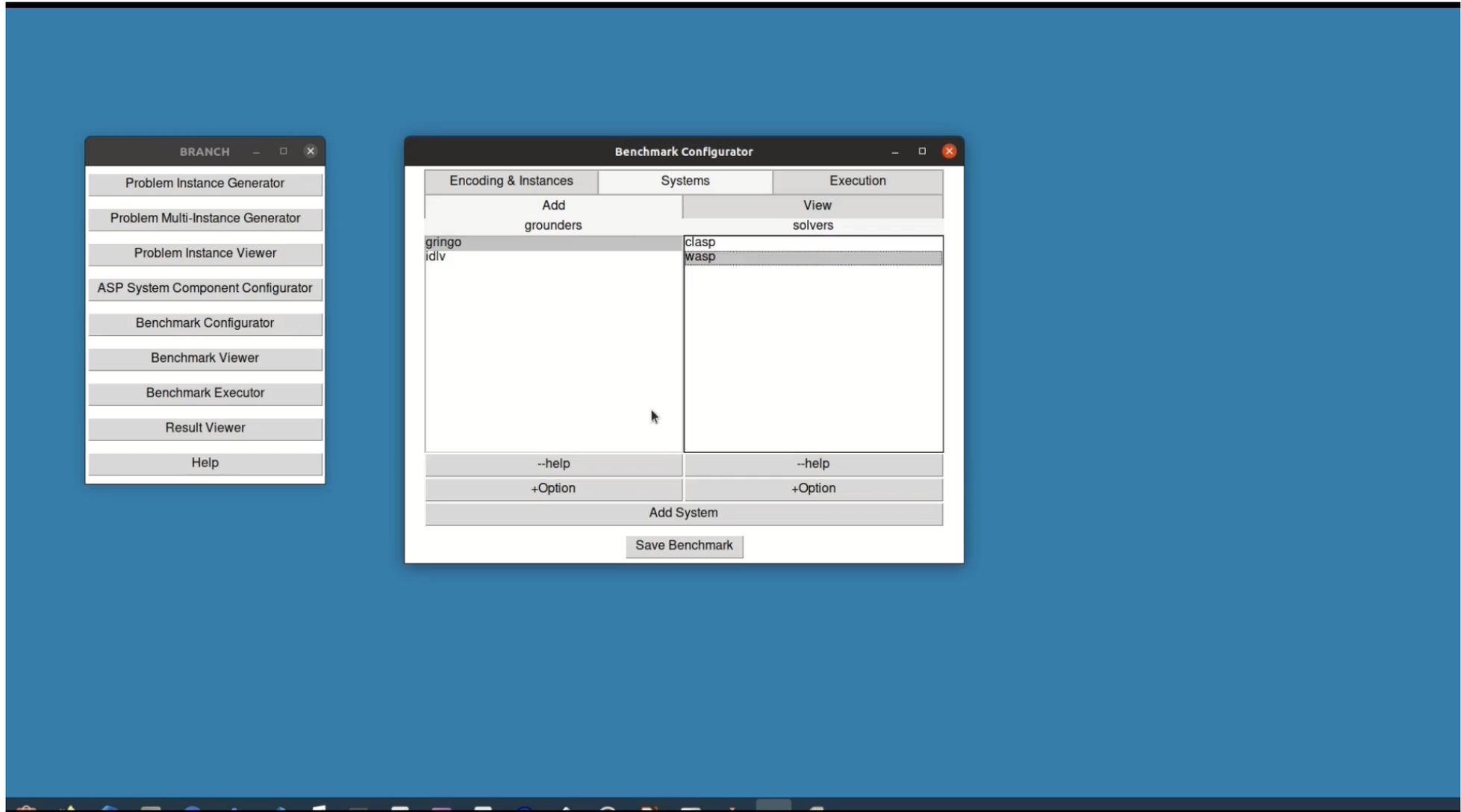
# An ASP Systems Benchmark for RABP



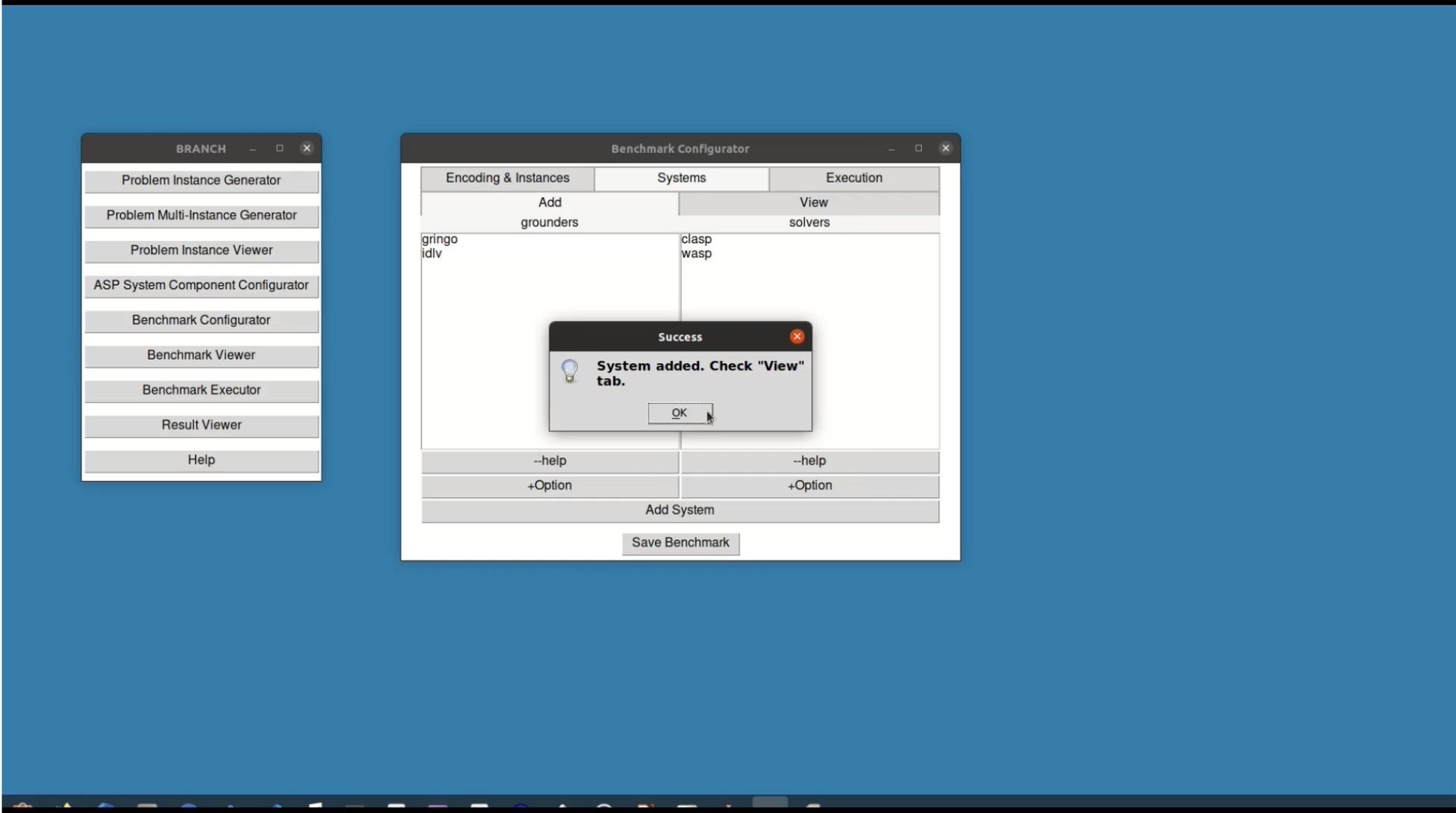
# An ASP Systems Benchmark for RABP



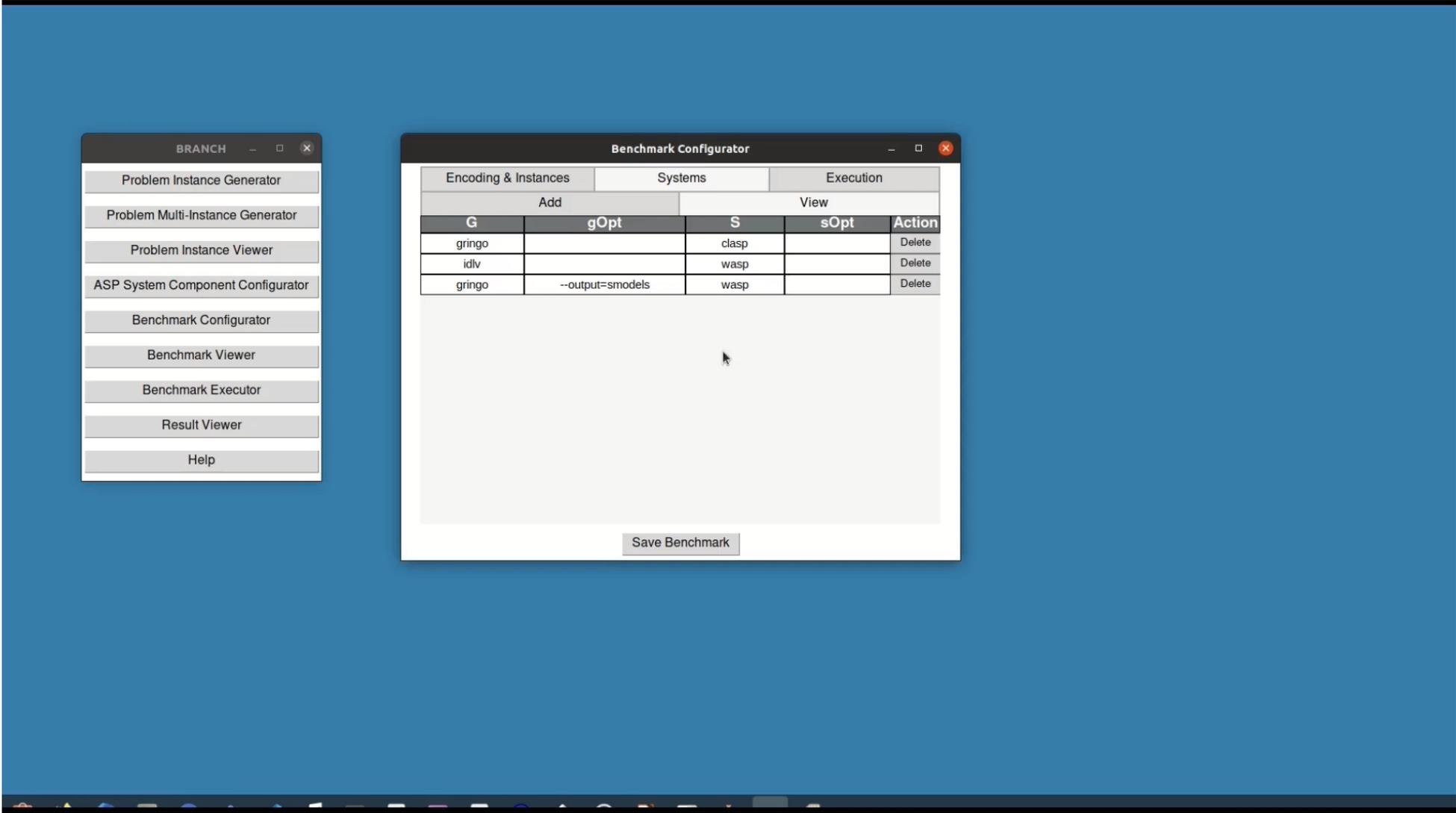
# An ASP Systems Benchmark for RABP



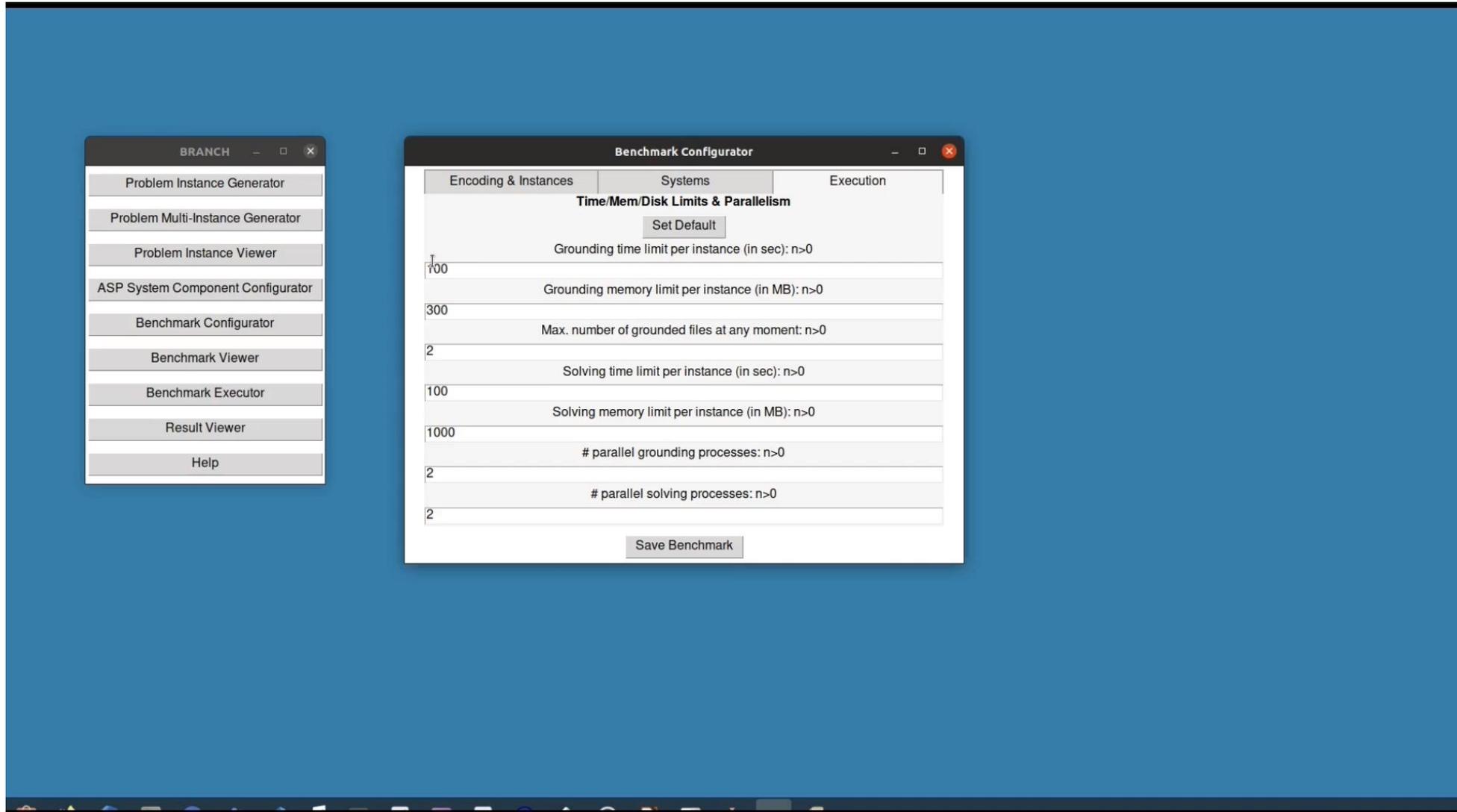
# An ASP Systems Benchmark for RABP



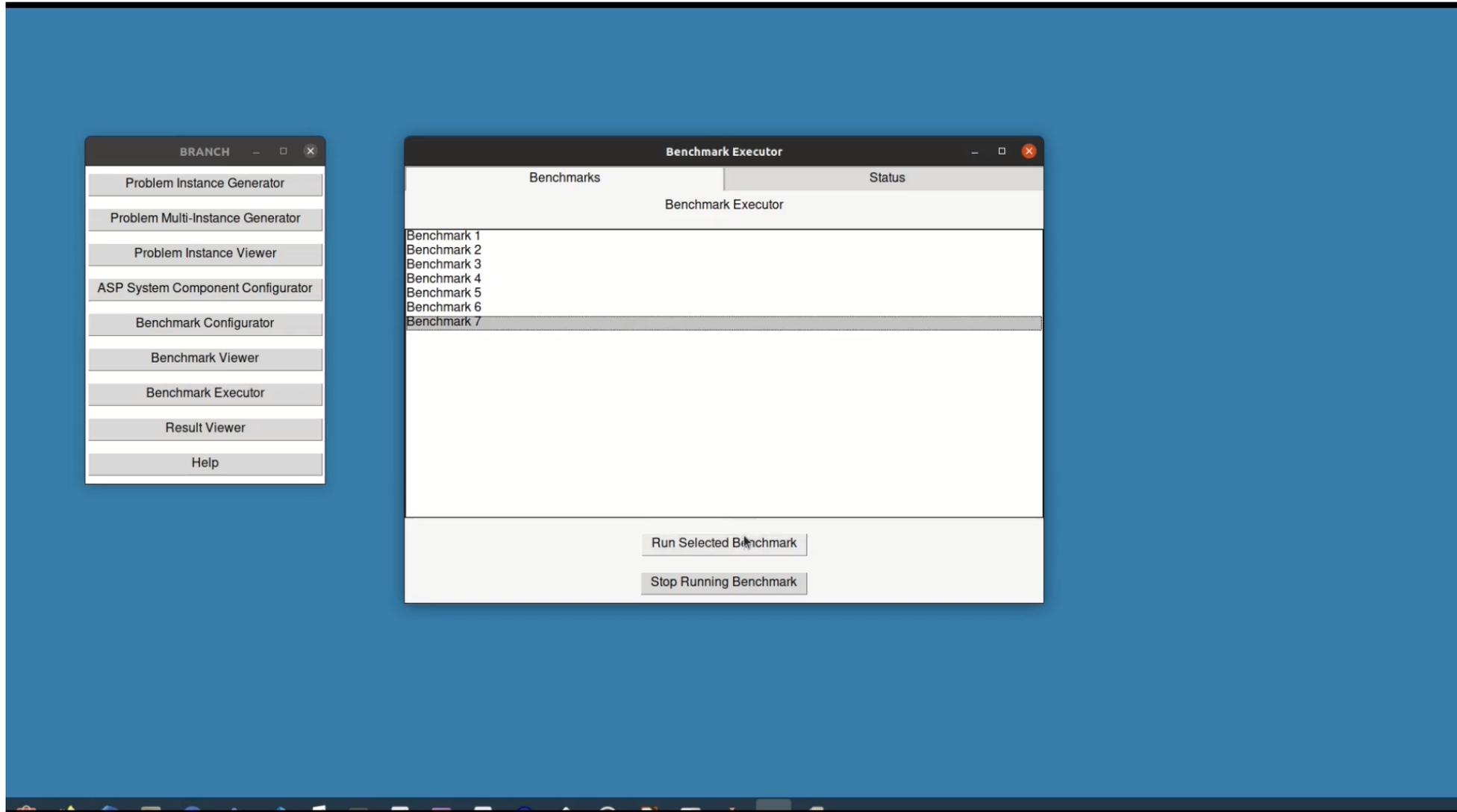
# An ASP Systems Benchmark for RABP



# An ASP Systems Benchmark for RABP



# An ASP Systems Benchmark for RABP



# An ASP Systems Benchmark for RABP

The screenshot displays the BRANCH software interface. On the left is a vertical menu with the following options: Problem Instance Generator, Problem Multi-Instance Generator, Problem Instance Viewer, ASP System Component Configurator, Benchmark Configurator, Benchmark Viewer, Benchmark Executor, Result Viewer, and Help. The 'Benchmark Executor' window is open in the center, showing a table of benchmarks and their execution status.

Benchmarks	Status
Instance 19 -> gringo of System 1	Instance 18 -> gringo of System 1
Instance 20 -> gringo of System 1	Instance 20 -> gringo of System 1
Instance 21 -> gringo of System 1	Instance 21 -> gringo of System 1
Instance 22 -> gringo of System 1	Instance 22 -> gringo of System 1
Instance 23 -> gringo of System 1	Instance 23 -> gringo of System 1
Instance 24 -> gringo of System 1	Instance 24 -> gringo of System 1
Instance 25 -> gringo of System 1	Instance 25 -> gringo of System 1
Instance 26 -> gringo of System 1	Instance 26 -> gringo of System 1
Instance 17 -> idlv of System 2	Instance 17 -> idlv of System 2
Instance 18 -> idlv of System 2	Instance 18 -> idlv of System 2
Instance 19 -> idlv of System 2	Instance 19 -> idlv of System 2
Instance 20 -> idlv of System 2	Instance 20 -> idlv of System 2
Instance 21 -> idlv of System 2	Instance 21 -> idlv of System 2
Instance 22 -> idlv of System 2	Instance 22 -> idlv of System 2
(Grounded Instance 18) -> clasp of System 1	(Grounded Instance 17) -> clasp of System 1
(Grounded Instance 20) -> clasp of System 1	(Grounded Instance 19) -> clasp of System 1
(Grounded Instance 21) -> clasp of System 1	(Grounded Instance 18) -> clasp of System 1
(Grounded Instance 22) -> clasp of System 1	(Grounded Instance 20) -> clasp of System 1
(Grounded Instance 23) -> clasp of System 1	(Grounded Instance 22) -> clasp of System 1
(Grounded Instance 24) -> clasp of System 1	(Grounded Instance 21) -> clasp of System 1
(Grounded Instance 25) -> clasp of System 1	(Grounded Instance 24) -> clasp of System 1
(Grounded Instance 26) -> clasp of System 1	(Grounded Instance 25) -> clasp of System 1
(Grounded Instance 17) -> wasp of System 2	(Grounded Instance 26) -> clasp of System 1
(Grounded Instance 18) -> wasp of System 2	(Grounded Instance 17) -> wasp of System 2
(Grounded Instance 19) -> wasp of System 2	(†) (Grounded Instance 23) -> clasp of System 1
(Grounded Instance 20) -> wasp of System 2	(Grounded Instance 18) -> wasp of System 2
(Grounded Instance 21) -> wasp of System 2	(Grounded Instance 20) -> wasp of System 2
(Grounded Instance 22) -> wasp of System 2	(Grounded Instance 19) -> wasp of System 2

# An ASP Systems Benchmark for RABP

BRANCH

- Problem Instance Generator
- Problem Multi-Instance Generator
- Problem Instance Viewer
- ASP System Component Configurator
- Benchmark Configurator
- Benchmark Viewer
- Benchmark Executor
- Result Viewer
- Help

Result Viewer

encoding	#instances	instances	systems	per instance limits	parallel execution
encoding_opt_projection.lp	2	(9-10)	System 1 (g[s]: gringo   clasp  System 2 (g[s]: idlv   wasp	Grounding time(s): 100 Grounding mem(MB): 300 #Max. Grounding Files: 2 Solving time(s): 100 Solving mem(MB): 1000	#Grounding threads: 2 #Solving threads: 2

Benchmark	Time
Benchmark 2	2021-06-17_23-14-21
Benchmark 3	2021-06-17_11-18-43
Benchmark 4	2021-06-17_11-15-52
Benchmark 5	2021-06-17_11-11-31
Benchmark 6	2021-06-17_11-02-21
Benchmark 7	2021-06-17_10-10-47
	2021-06-17_10-09-35
	2021-06-17_10-06-51
	2021-06-17_10-05-02
	2021-06-17_01-42-42
	2021-06-17_01-37-38
	2021-06-17_01-35-07
	2021-06-17_01-31-13
	2021-06-17_00-49-59
	2021-06-17_00-46-27
	2021-06-17_00-44-50
	2021-06-16_19-04-59
	2021-06-16_19-02-06
	2021-06-16_18-59-53
	2021-06-16_18-51-39
	2021-06-16_18-19-30
	2021-06-16_17-19-38

Grounding comparison

- G. time box plot
- G. time csv
- G. time cactus plot
- G. mem box plot
- G. mem csv
- Ground size box plot
- Ground size csv

Solving comparison

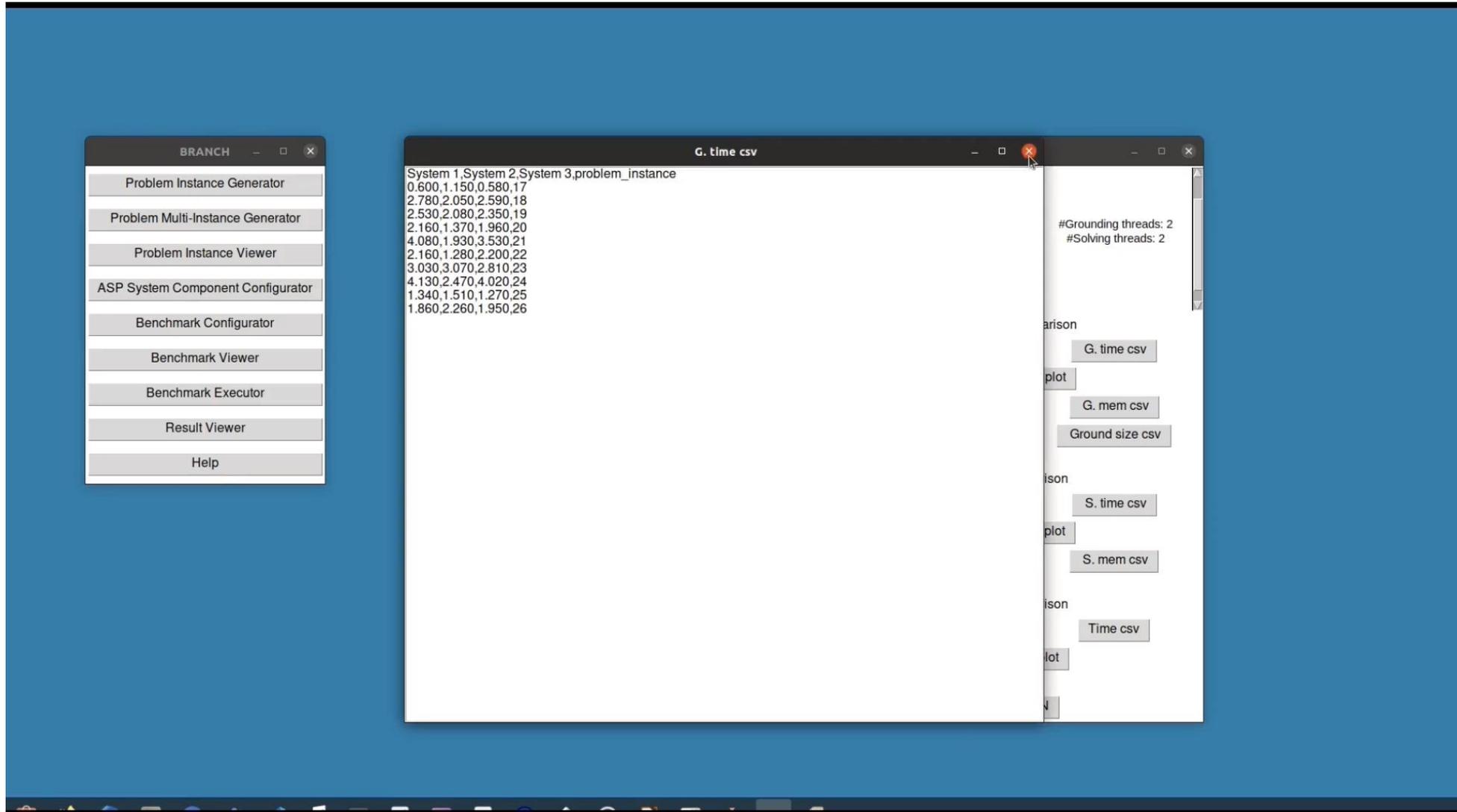
- S. time box plot
- S. time csv
- S. time cactus plot
- S. mem box plot
- S. mem csv

System comparison

- Time box plot
- Time csv
- Time cactus plot
- Result JSON

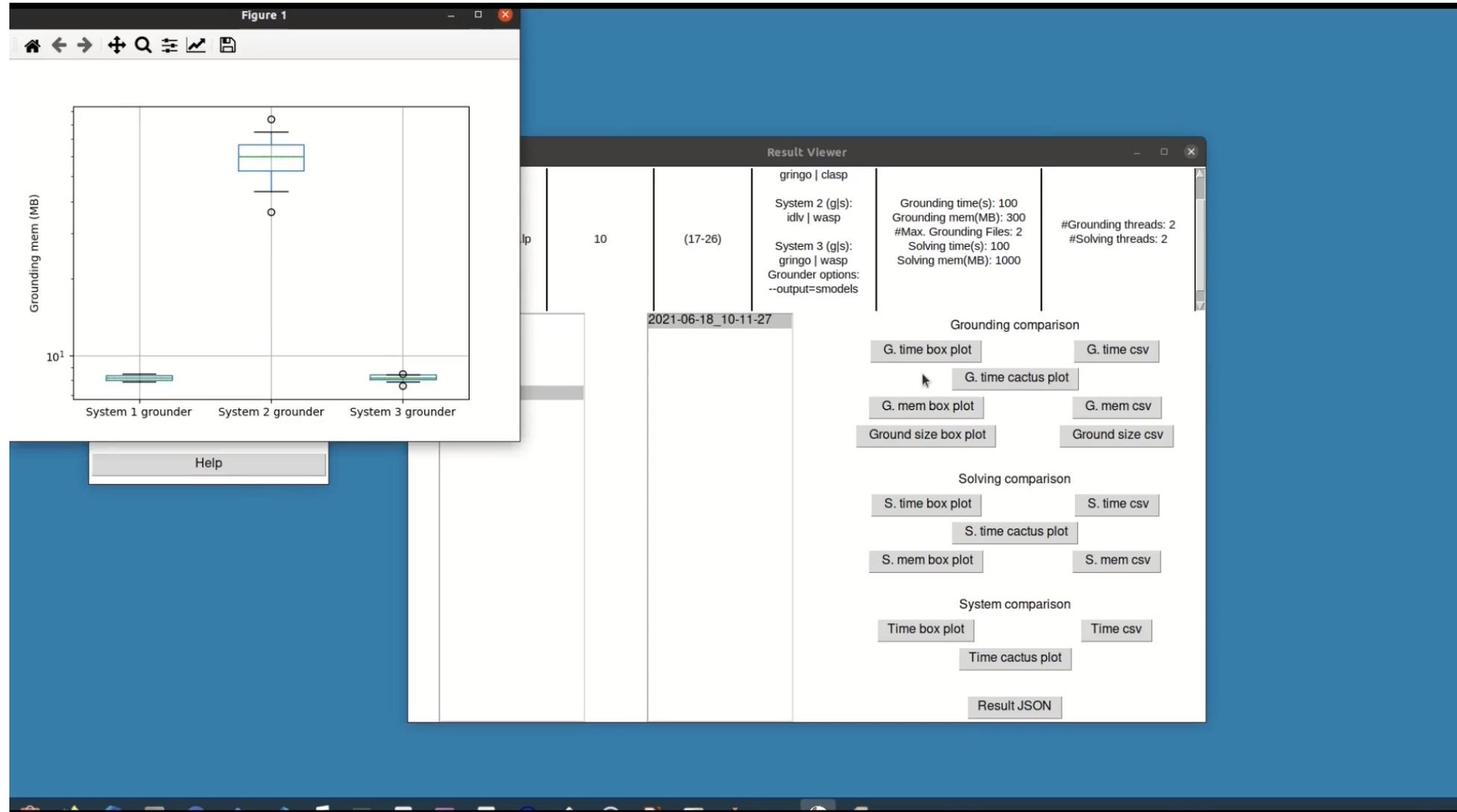
# Contributions to RQ 2

## An ASP Systems Benchmark for RABP



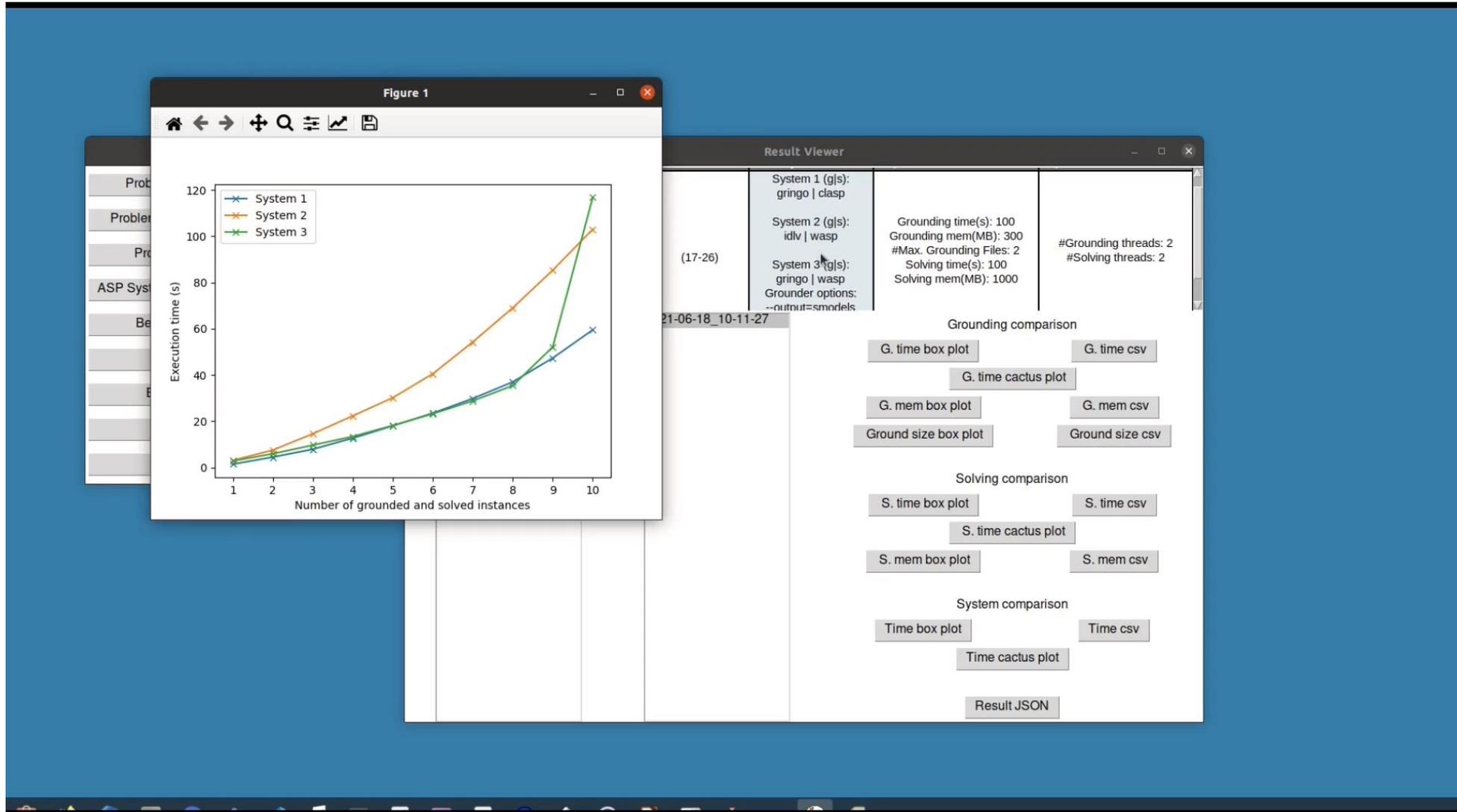
# Contributions to RQ 2

## An ASP Systems Benchmark for RABP



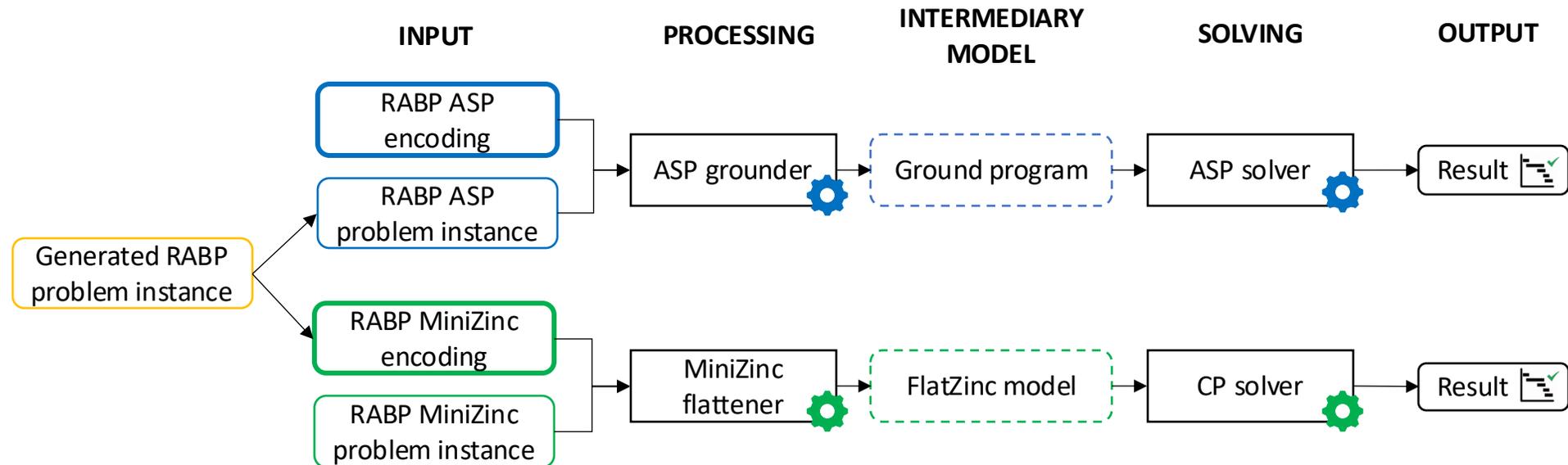
# Contributions to RQ 2

## An ASP Systems Benchmark for RABP



# ASP vs CP: Performance Evaluation

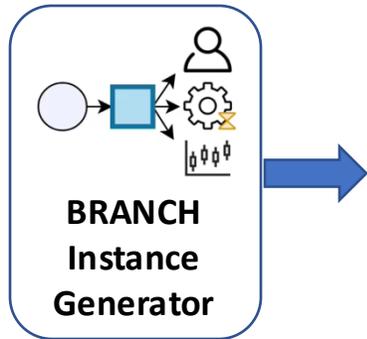
- Running RABP Instances in ASP and CP



- Comparison of four ASP systems against four CP systems

- ASP systems: **gringo+clasp, gringo+wasp, idlv+clasp, idlv+wasp**
- CP systems: **mzn2fzn+gencode, mzn2fzn+chuffed, mzn2fzn+hcsp, mzn2fzn+or-tools**

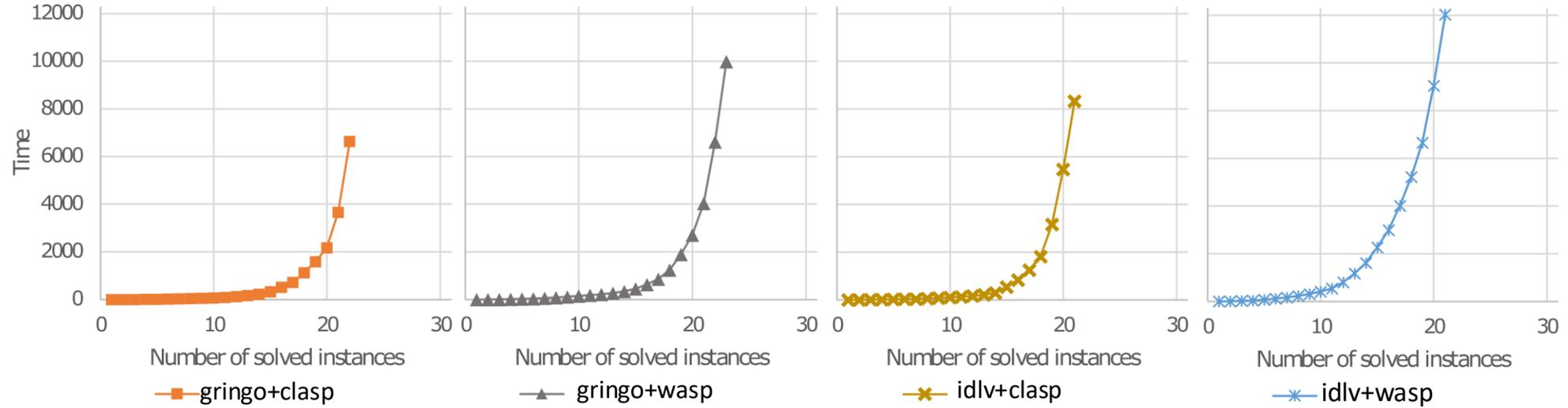
# ASP vs CP: Performance Evaluation



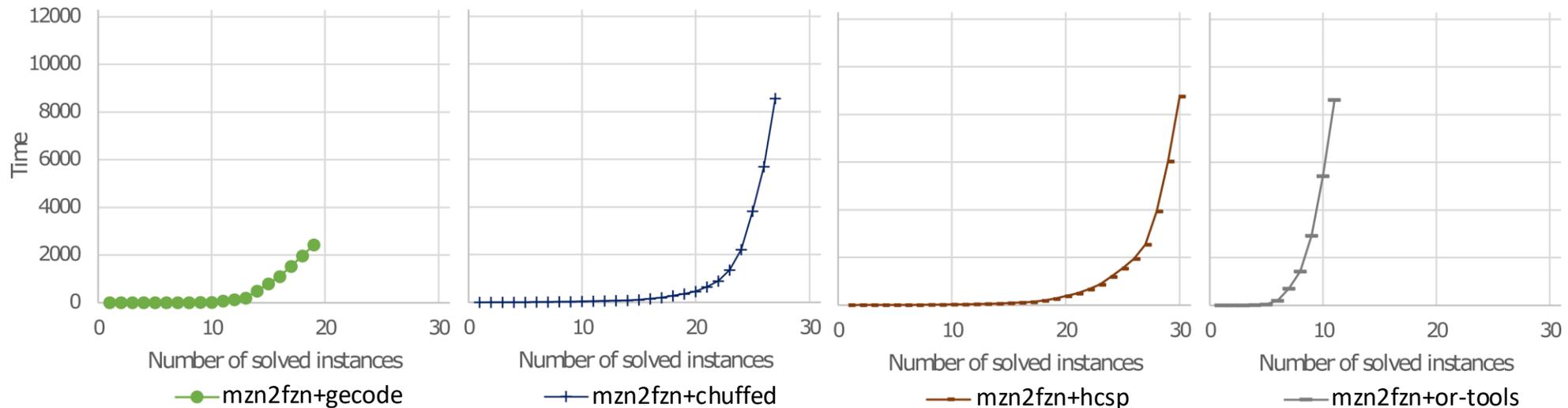
id	#A	%conc	#R	#L	u	id	#A	%conc	#R	#L	u
1	16	10	2	1	32	22	16	50	2	1	48
2	16	50	2	1	32	23	16	50	2	1	64
3	16	90	2	1	32	24	24	50	3	1	72
4	24	10	3	1	48	25	24	50	3	1	96
5	24	50	3	1	48	26	32	50	4	2	96
6	24	90	3	1	48	27	32	50	4	2	128
7	32	10	4	2	64	28	48	50	6	3	144
8	32	50	4	2	64	29	48	50	6	3	192
9	32	90	4	2	64	30	64	50	8	4	192
10	48	10	6	3	96	31	64	50	8	4	256
11	48	50	6	3	96	32	96	50	12	6	288
12	48	90	6	3	96	33	96	50	12	6	384
13	64	10	8	4	128	34	128	50	16	8	384
14	64	50	8	4	128	35	128	50	16	8	512
15	64	90	8	4	128						
16	96	10	12	6	192						
17	96	50	12	6	192						
18	96	90	12	6	192						
19	128	10	16	8	256						
20	128	50	16	8	256						
21	128	90	16	8	256						

# ASP vs CP: Performance Evaluation

## ASP Systems



## CP Systems



## ASP vs CP

	ASP	CP
Performance	++	+++
Readability	+++	+
Encoding RABP Problem	+++	++
Encoding RABP Instances	+++	+

## Representing a wide variety of resources in RABP

- BPI15:** [Giray Havur](#), Cristina Cabanillas, Jan Mendling, and Axel Polleres. Automated resource allocation in business processes with answer set programming. In Business Process Management Workshops: (BPM 2015), Revised Papers, pages 191–203, 2016.
- SIMPDA15:** Saimir Bala, Cristina Cabanillas, Alois Haselbock, [Giray Havur](#), Jan Mendling, Axel Polleres, Simon Sperl, and Simon Steyskal. A framework for safety-critical process management in engineering projects. In SIMPDA 2015, Revised Selected Papers, volume 244 of LNBIP, pages 1–27, Springer, 2015.
- BPMDemo15:** Saimir Bala, [Giray Havur](#), Simon Sperl, Simon Steyskal, Alois Haselbock, Jan Mendling, and Axel Polleres. Shapeworks: A BPMS extension for complex process management. In the BPM Demo Track 2016 co-located with BPM 2016, volume 1789 of CEUR Workshop Proceedings, pages 50–55, 2016.
- BPMForum16:** [Giray Havur](#), Cristina Cabanillas, Jan Mendling, and Axel Polleres. Resource allocation with dependencies in business process management systems. In Business Process Management Forum, pages 3–19, 2016.

RESEARCH-ARTICLE



# Benchmarking Answer Set Programming systems for resource allocation in business processes

**Authors:** [Giray Havur](#), [Cristina Cabanillas](#), [Axel Polleres](#) [Authors Info & Claims](#)

Expert Systems with Applications: An International Journal, Volume 205, Issue C • Nov 2022 • <https://doi.org/10.1016/j.eswa.2022.117599>

More details in Giray's thesis...

*Thank you!*



# Why I love ASP (since over 20 years)...

- **Intuitive**, understandable problem **encodings...**
- ... easily **extensible**
- the beauty of **Guess and Check** to solve complex problems on top
- integrateable in real systems... looking forward to learn more from you!

## Part 1:

1999-2003



AI Planning

## Part 2:

2003 – to date...



Semantic Web



Universidad  
Rey Juan Carlos



NUI Galway  
OÉ Gaillimh

## Part 3:

ca. 2014 – 2022...



**SIEMENS**

Business Process  
Management