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

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:  \( \text{on}(c,a), \text{on}(b,\text{table}), \text{on}(a,\text{table}) \)
• Actions:  \text{move}(c,\text{table}) \) with
  • preconditions:
    executable \( \text{move}(B, L) \) if block(B).
    nonexecutable \( \text{move}(B, B) \).
    nonexecutable \( \text{move}(B, L) \) if blocked(B).
    nonexecutable \( \text{move}(B, L) \) if blocked(L).
  • effects:
    caused \( \text{on}(B, L) \) after \( \text{move}(B, L) \).
    caused \( -\text{on}(B, L) \) after \( \text{move}(B, L1), \text{on}(B, L), L1<>L \).
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), 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**:
    
    \[
    \text{move}(B, L,T) \lor \neg \text{move}(B, L,T) :- \text{block}(B), \text{location}(L), \text{time}(T).
    \]
    
    \[
    :- \text{move}(B,B,T).
    \]
    
    \[
    :- \text{move}(B, L,T), \text{blocked}(B,T). \quad \text{blocked}(B,T) :- \text{on}(\_ ,B,T) .
    \]
    
    \[
    :- \text{move}(B, L), \text{blocked}(L,T) .
    \]
  - **effects**:
    
    \[
    \text{on}(B,L, T+1) :- \text{move}(B,L,T).
    \]
    
    \[
    \neg \text{on}(B,L,T+1) :- \text{move}(B,L1,T), \text{on}(B,L,T), L1<>L.
    \]
  - **implicit background knowledge**:
    
    \[
    \text{blocked}(B,T) :- \text{on}(B1,B,T), \text{block}(B).
    \]
    
    \[
    \text{on}(B,L,T+1) :- \text{on}(B,L,T), \text{not} \neg \text{on}(B,L,T).
    \]
- **Goal**:
  
  \[
  \text{goal}(T) :- \text{on}(a,b), \text{on}(b,c), \text{on}(c,\text{table}).
  \]
  
  \[
  :- \neg \text{goal}(\text{maxTime}).
  \]
  
  \[
  \text{time}(0..\text{maxTime}).
  \]
AI Planning... ease of extensions:

• Nondeterministic actions:
  
  • E.g., “clumsy” move
    
    \[
    \text{on}(B, L, T+1) \lor \text{on}(B, \text{table}, T+1) :\!-\! \text{cmove}(B, L, T), \text{on}(B, L, T), \text{block}(L). \\
    \text{on}(B, L, T+1) :\!-\! \text{cmove}(B, L, T), \text{on}(B, \text{table}, T).
    \]

  • Action costs /cost optimal planning:
    
    \[
    \text{cost}(T, 1) :\!-\! \text{cmove}(B, L, T). \\
    \text{cost}(T, 2) :\!-\! \text{move}(B, L, T). \\
    \sim \text{cost}(T, \text{Cost}). \ [\text{Cost}:]
    \]
Nondeterministic actions:

- E.g., "clumsy" move on(B,L,T+1) v on(B,table,T+1):
  - cmove(B,L,T), on(B,table,T).

- Action costs /cost optimal planning:
  - cost(T,1) :- cmove(B,L,T).
  - cost(T,2) :- move(B,L,T).

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

- Action costs /cost optimal planning:
  - cost(T,1) :- cmove(B,L,T).
  - cost(T,2) :- move(B,L,T).

A logic programming approach to knowledge-state planning:

- Answer set planning under action costs

Authors: Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer, Axel Polleres
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
Semantic Web ... RDF Triples

• “Typed” links on the Web ...


• ... 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!

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 ;
  {{?X dbo:birthPlace dbr:Baghdad} UNION
   {?X dbo:birthPlace dbr:Vienna}}
  FILTER (lang(?N) = "en")

  2016-03-31
  type
  deathDate

  1950-10-31
  birthDate

  Iraq
  country

  Baghdad
  birthPlace

  Zaha_Hadid
  Person

  Architect
  subClassOf
}
Semantic Web ...Status ~2006

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

• Why I liked SPARQL?

```sparql
SELECT ?X
WHERE {
}
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:

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

Example Query:

```sparql
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) .
```

```
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>tim</td>
<td>jim</td>
<td><a href="mailto:timbl@w3.org">timbl@w3.org</a></td>
</tr>
<tr>
<td>jim</td>
<td>tim</td>
<td></td>
</tr>
</tbody>
</table>
```

```
\[] X, Y \rightarrow \pi X, M
```

```
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>tim</td>
<td>jim</td>
<td><a href="mailto:timbl@w3.org">timbl@w3.org</a></td>
</tr>
<tr>
<td>jim</td>
<td>tim</td>
<td></td>
</tr>
</tbody>
</table>
```

Give me people who know somebody and OPTIONALLY their email address:
SPARQL... intuitive encoding

[Polleres, 2007] shows that all of SPARQL 1.0 can be translated to (safe) nonrecursive Datalog\textsuperscript{not}.

[Angles&Gutierrez 2008] vice versa show that (safe) nonrecursive Datalog\textsuperscript{not} likewise be encoded into SPARQL.
SPARQL... ASP’s ease of extensions:

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

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…

On the relation between SPARQL1.1 and Answer Set Programming
Axel Polleres & Johannes Peter Wallner
Pages 159-212 | Published online: 24 Jun 2013

SPARQL++ for mapping between RDF vocabularies

Authors: Axel Polleres, François Scharffe, Roman Schindlauer

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:

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

### Web source P1:

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

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

### Web source P2:

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

CONSTRUCT {jim :knows ?m }
FROM P1
WHERE {jim :knows ?m }
```
SPARQL... ASP’s guess and check:

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

Web source P1:

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

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 ]].
  ```

  Data Graph:
  
  ```
  : Ben :enrolledIn :C1 .
  :C1 :type 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.
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/S00379ED1V01Y201108DM020)

Leopoldo Bertossi
Carleton University, Ottawa, Canada
SHACL ... computing repairs by Guess and check:

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

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

**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 Simkus (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**

**Part 3:**
ca. 2014 – 2022...
- **Business Process Management**
Resource allocation in BPM

• Recently concluded PhD thesis:

Giray Havur

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

Activity 1 → Activity 2 → Activity 3 → Activity 4 → Activity 5
Organizing Work
Business Process Modelling Notation
Organizing Work
Business Process Modelling Notation

Activity 1

Activity 2

Activity 3

Activity 4

Activity 5
Organizing Work
Business Process Modelling Notation
Organizing Work
Business Process Modelling Notation
Organizing Work

Business Process Modelling Notation

Activity 1 → Parallel Gateway (Diverging) → Activity 2
                         ↓                           ↓
                            +                                +
                      Activity 3 → Parallel Gateway (Converging) → Activity 4

Activity 2 → Activity 3 → Activity 4 → Activity 5 → End
Organizing Work
Business Process Modelling Notation

Activity 1 -> Activity 2

Activity 2 -> Activity 3

Activity 3 -> Activity 4

Activity 4 -> Activity 5

Parallel Gateway (Diverging)

Parallel Gateway (Converging)
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

Study requirements
- Develop backend
- Develop frontend
- Verify backend
- Verify frontend
- Deploy system

1 mo.
5 mos.
3 mos.
6 mos.
5 mos.
2 mos.

Manager: Amy
Software Engineer: Jessie
UI Developer: Oliver
Testing Expert: Mia
Junior Dev.: Liam
UI Developer: Glen

Organizational Model: An RBAC Model of the Software Development Company
Organizing Resources
Role-based Access Control Model

1. Study requirements
2. Develop backend
   - 5 mos.
3. Verify backend
   - 3 mos.
4. Develop frontend
   - 6 mos.
5. Verify frontend
   - 5 mos.
6. Deploy system
   - 2 mos.

Manager: Amy

Software Engineer: Jessie
UI Developer: Oliver
Testing Expert: Mia
Junior Dev.: Liam
Glen

- Resources
Organizing Resources
Role-based Access Control Model

Study requirements
+ Develop backend
  - 5 mos.
  - 5 mos.
Develop frontend
  - 6 mos.
Verify backend
  - 3 mos.
  - 5 mos.
+ Deploy system
  - 2 mos.

Resources
- Manager
- Amy
- Software Engineer
- Jessie
- UI Developer
- Oliver
- Testing Expert
- Mia
- Junior Dev.
- Lisa
- Liam
- Glen
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

1. Study requirements
   - 1 mo.

2. Develop backend
   - 5 mos.

3. Verify backend
   - 3 mos.

4. Develop frontend
   - 6 mos.

5. Verify frontend
   - 5 mos.

6. Deploy system
   - 2 mos.

Resources:
- Manager: Amy
- Software Engineer: Jessie
- UI Developer: Oliver
- Testing Expert: Mia
- Junior Dev.: Lisa
- Development: Liam
- UI: Glen

Business Process Management System (BPMS)
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

**Business Process Management System (BPMS)**

- **Study requirements**: 1 mo.
- **Develop backend**: 5 mos.
- **Verify backend**: 3 mos.
- **Deploy system**: 2 mos.
- **Develop frontend**: 6 mos.
- **Verify frontend**: 5 mos.
- **Deploy system**: 2 mos.

**Resource Allocation in Business Processes**

- **Manager**: Amy
- **Software Engineer**: Jessie
- **UI Developer**: Oliver
- **Testing Expert**: Mia
- **Junior Dev.**: Lisa
- **Glen**:
Running Business Processes

Business Process Management System

Study requirements

- 1 mo.

Develop backend

- 5 mos.

Verify backend

- 3 mos.

Develop frontend

- 6 mos.

Verify frontend

- 5 mos.

Deploy system

- 2 mos.

Business Process Management System (BPMS)

event_id | trace_id | event_type | activity | resource | time_stamp

Available

Manager

Amy

Software Engineer

UI Developer

Testing Expert

Junior Dev.

Jessie

Oliver

Mia

Lisa

Liam

Glen
Running Business Processes
Business Process Management System

Manager

Study requirements

- 1 mo.

Develop backend

5 mos.

Verify backend

3 mos.

Deploy system

2 mos.

Develop frontend

6 mos.

Verify frontend

5 mos.

Deploy system

2 mos.

Software Engineer

UI Developer

Testing Expert

Junior Dev.

Manager

Amy

Busy

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

Business Process Management System (BPMS)
Running Business Processes
Business Process Management System

Manager

Software Engineer

UI Developer

Testing Expert

Junior Dev.

Liam

Glen

Jessie

Oliver

Mia

Lisa

Resource Allocation in Business Processes

Study requirements

- 1 mo.

Develop backend

- 5 mos.

Verify backend

- 3 mos.

Deploy system

- 2 mos.

Develop frontend

- 6 mos.

Verify frontend

- 5 mos.

Business Process Management System (BPMS)

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
</tbody>
</table>
Running Business Processes
Business Process Management System

Resource Allocation in Business Processes

<table>
<thead>
<tr>
<th>Study requirements</th>
<th>1 mo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop backend</td>
<td>5 mos.</td>
</tr>
<tr>
<td>Verify backend</td>
<td>3 mos.</td>
</tr>
<tr>
<td>Deploy system</td>
<td>2 mos.</td>
</tr>
<tr>
<td>Develop frontend</td>
<td>6 mos.</td>
</tr>
<tr>
<td>Verify frontend</td>
<td>5 mos.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
</tbody>
</table>
Running Business Processes
Business Process Management System

Event

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e4</td>
<td>1</td>
<td>start</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>04/2020</td>
</tr>
</tbody>
</table>
Running Business Processes
Business Process Management System

Resource Allocation in Business Processes

<table>
<thead>
<tr>
<th>Study requirement</th>
<th>Develop backend</th>
<th>Verify backend</th>
<th>Develop frontend</th>
<th>Verify frontend</th>
<th>Deploy system</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mo.</td>
<td>5 mos.</td>
<td>3 mos.</td>
<td>6 mos.</td>
<td>5 mos.</td>
<td>2 mos.</td>
</tr>
</tbody>
</table>

Business Process Management System (BPMS)

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e4</td>
<td>1</td>
<td>start</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>04/2020</td>
</tr>
<tr>
<td>e5</td>
<td>1</td>
<td>end</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>07/2020</td>
</tr>
</tbody>
</table>

Running Business Processes
Business Process Management System
Running Business Processes
Business Process Management System

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e4</td>
<td>1</td>
<td>start</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>04/2020</td>
</tr>
<tr>
<td>e5</td>
<td>1</td>
<td>end</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>07/2020</td>
</tr>
<tr>
<td>e6</td>
<td>1</td>
<td>start</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>07/2020</td>
</tr>
</tbody>
</table>
Running Business Processes
Business Process Management System

Manager

Software Engineer

UI Developer

Testing Expert

Junior Dev.

Liam

Glen

Jessie

Oliver

Mia

Lisa

Study requirements

Develop backend

Verify backend

Develop frontend

Verify frontend

Deploy system

Business Process Management System (BPMS)

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e4</td>
<td>1</td>
<td>start</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>04/2020</td>
</tr>
<tr>
<td>e5</td>
<td>1</td>
<td>end</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>07/2020</td>
</tr>
<tr>
<td>e6</td>
<td>1</td>
<td>start</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>07/2020</td>
</tr>
<tr>
<td>e7</td>
<td>1</td>
<td>end</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>10/2020</td>
</tr>
</tbody>
</table>
### Resource Allocation in Business Processes

**Business Process Management System (BPMS)**

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e4</td>
<td>1</td>
<td>start</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>04/2020</td>
</tr>
<tr>
<td>e5</td>
<td>1</td>
<td>end</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>07/2020</td>
</tr>
<tr>
<td>e6</td>
<td>1</td>
<td>start</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>07/2020</td>
</tr>
<tr>
<td>e7</td>
<td>1</td>
<td>end</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>10/2020</td>
</tr>
<tr>
<td>e8</td>
<td>1</td>
<td>end</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>10/2020</td>
</tr>
<tr>
<td>e9</td>
<td>1</td>
<td>start</td>
<td>Verify_frontend</td>
<td>Lisa</td>
<td>11/2020</td>
</tr>
<tr>
<td>e10</td>
<td>1</td>
<td>end</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>11/2020</td>
</tr>
<tr>
<td>e11</td>
<td>1</td>
<td>start</td>
<td>Deploy_system</td>
<td>Jessie</td>
<td>11/2020</td>
</tr>
<tr>
<td>e12</td>
<td>1</td>
<td>end</td>
<td>Deploy_system</td>
<td>Jessie</td>
<td>01/2021</td>
</tr>
</tbody>
</table>
Running Business Processes
Mining a Temporal Model from the Event Log

Manager

Amy

Software Engineer

Junior Dev.

Jessie

UI Developer

Oliver

Testing Expert

Mia

Lisa

Junior Dev.

Liam

Glen

Business Process Management System (BPMS)

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 | 11/2020
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

- Default activity durations
- Resource-specific activity durations

Manager
- Amy
- Junior Dev.
- Jessie
- Oliver
- Mia
- Lisa

Software Engineer
- Junior Dev.

UI Developer
- Oliver
- Mia
- Lisa

Testing Expert
- Lisa

Business Process Management System (BPMS)

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>1</td>
<td>start</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>01/2020</td>
</tr>
<tr>
<td>e2</td>
<td>1</td>
<td>end</td>
<td>Study_requirements</td>
<td>Amy</td>
<td>02/2020</td>
</tr>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e4</td>
<td>1</td>
<td>start</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>04/2020</td>
</tr>
<tr>
<td>e5</td>
<td>1</td>
<td>end</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>07/2020</td>
</tr>
<tr>
<td>e6</td>
<td>1</td>
<td>start</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>07/2020</td>
</tr>
<tr>
<td>e7</td>
<td>1</td>
<td>end</td>
<td>Develop_frontend</td>
<td>Oliver</td>
<td>10/2020</td>
</tr>
<tr>
<td>e8</td>
<td>1</td>
<td>end</td>
<td>Verify_backend</td>
<td>Lisa</td>
<td>10/2020</td>
</tr>
<tr>
<td>e9</td>
<td>1</td>
<td>start</td>
<td>Verify_frontend</td>
<td>Lisa</td>
<td>10/2020</td>
</tr>
<tr>
<td>e10</td>
<td>1</td>
<td>end</td>
<td>Verify_frontend</td>
<td>Lisa</td>
<td>11/2020</td>
</tr>
<tr>
<td>e11</td>
<td>1</td>
<td>start</td>
<td>Deploy_system</td>
<td>Jessie</td>
<td>11/2020</td>
</tr>
<tr>
<td>e12</td>
<td>1</td>
<td>end</td>
<td>Deploy_system</td>
<td>Jessie</td>
<td>01/2021</td>
</tr>
</tbody>
</table>
Running Business Processes
Mining a Temporal Model from the Event Log

- Study requirements
- Develop backend: 5 mos.
- Verify backend: 3 mos.
- Develop frontend: 6 mos.
- Verify frontend: 5 mos.
- Deploy system: 2 mos.

Business Process Management System (BPMS)

<table>
<thead>
<tr>
<th>event_id</th>
<th>trace_id</th>
<th>event_type</th>
<th>activity</th>
<th>resource</th>
<th>time_stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>e3</td>
<td>1</td>
<td>start</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>02/2020</td>
</tr>
<tr>
<td>e5</td>
<td>1</td>
<td>end</td>
<td>Develop_backend</td>
<td>Glen</td>
<td>07/2020</td>
</tr>
<tr>
<td>e12</td>
<td>2</td>
<td>start</td>
<td>Develop_backend</td>
<td>Liam</td>
<td>10/2020</td>
</tr>
<tr>
<td>e15</td>
<td>2</td>
<td>end</td>
<td>Develop_backend</td>
<td>Liam</td>
<td>01/2021</td>
</tr>
</tbody>
</table>

- Default activity durations
- Role-specific activity durations
- Resource-specific activity durations
Running Business Processes
Mining a Temporal Model from the Event Log

- Default activity durations
- Role-specific activity durations
- Resource-specific activity durations
BPMS with RABP Support

Business Process Management System (BPMS)

RABP Method

Study requirements (1 mo.)

Develop backend (5 mos.)

Verify backend (3 mos.)

Develop frontend (6 mos.)

Verify frontend (5 mos.)

Deploy system (2 mos.)

1 mo.

5 mos.

3 mos.

6 mos.

5 mos.

2 mos.

4 mos.

Manager

Amy

Software Engineer

UI Developer

Testing Expert

Junior Dev.

Jessie

Oliver

Mia

Lisa

Sofia

Glen

Lisa

Verify frontend

Junior Dev.

Develop backend

1 mo.

4 mos.

Amy

Sofia

Lisa

Oliver

Jessie

0 5 10 14

t
BPMS with RABP Support

Business Process Management System (BPMS)

RABP Method

Study requirements → Develop backend → Verify backend → Develop frontend → Verify frontend → Deploy system

Manager

Software Engineer

UI Developer

Testing Expert

Junior Dev.

Lisa

Sofia

Glen

Source: Glen

Develop backend

Verify backend

Develop frontend

Verify frontend

Deploy system

Amy, Sofia, Lisa, Oliver, Jessie

Development Timeline:

- Develop backend: Amy, Sofia, Lisa
- Develop frontend: Lisa, Jessie
- Verify backend: Amy, Sofia
- Verify frontend: Sofia, Lisa
- Deploy system: Lisa

Timeline:

0 5 10 14

1 mo. 5 mos. 3 mos. 6 mos. 5 mos. 2 mos. 4 mos.
BPMS with RABP Support

Business Process Management System (BPMS)

RABP Method

Study requirements
- Develop backend
- Develop frontend
- Verify backend
- Verify frontend
- Deploy system

1 mo.
5 mos.
6 mos.
3 mos.
5 mos.
2 mos.
4 mos.

Manager
- Amy

Software Engineer
- Jessie

UI Developer
- Glen

Testing Expert
- Lisa

Junior Dev.
- Sofia
- Oliver

Develop backend
- Lisa
- Sofia

Verify backend
- Amy

Develop frontend
- Jessie
- Oliver

Deployment
- DS

Develop frontend
- VF

Verify backend
- SR

Amy
- SR

Sofia
- Develop backend

Lisa
- VF

Oliver
- Develop frontend

Jessie
- DS
BPMS with RABP Support

Business Process Management System (BPMS)

RABP Method

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration (mo.)</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop backend</td>
<td>5</td>
<td>Sofia, Lisa, Oliver, Jessie</td>
</tr>
<tr>
<td>Verify backend</td>
<td>3</td>
<td>SR, Amy</td>
</tr>
<tr>
<td>Develop frontend</td>
<td>6</td>
<td>Lisa, Glen, Sofia</td>
</tr>
<tr>
<td>Verify frontend</td>
<td>5</td>
<td>VF, Amy</td>
</tr>
<tr>
<td>Deploy system</td>
<td>2</td>
<td>DS</td>
</tr>
</tbody>
</table>

Manager: Amy
Software Engineer: Sofia
UI Developer: Lisa
Testing Expert: Glen
Junior Dev.: Jessie

Develop backend: Sofia, Lisa, Oliver, Jessie
Verify backend: SR, Amy
Develop frontend: Lisa, Glen, Sofia
Verify frontend: VF, Amy
Deploy system: DS
BPMS with RABP Support

Study requirements
1 mo.

Develop backend
5 mos.

Verify backend
3 mos.

Develop frontend
6 mos.

Verify frontend
5 mos.

Deploy system
2 mos.

1 mo.
5 mos.
3 mos.
6 mos.
5 mos.
2 mos.
4 mos.

Software Engineer
Jessie
Manager
Amy
UI Developer
Oliver
Testing Expert
Mia
Junior Dev.
Lisa

RABP Method

Business Process Management System (BPMS)

Amy
SR
Verify backend

Sofia
Develop backend

Lisa
VF

Oliver
Develop frontend

Jessie

VF

Design

Mia

Glen

Junior Dev.

Lisa

Verify frontend

1 mo.
4 mos.
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").
dprec("StudyRequirements", "DevelopBackend").
activity("DevelopBackend").
activity("VerifyBackend").
activity("DevelopFrontend").
activity("VerifyFrontend").
activity("DeploySystem").
BPM.... Ease of encoding in ASP

activity("StudyRequirements").
activity("DevelopBackend").
dprec("StudyRequirements", "DevelopBackend").
activity("VerifyBackend").
dprec("StudyRequirements", "DevelopFrontend").
activity("DevelopFrontend").
dprec("DevelopBackend", "VerifyBackend").
activity("VerifyFrontend").
dprec("DevelopFrontend", "VerifyFrontend").
activity("DeploySystem").
dprec("VerifyBackend", "DeploySystem").
Contributions to RQ 1
Representation of Processes in ASP

activity("StudyRequirements").
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”).
dprec(“StudyRequirements”, “DevelopBackend”).
dprec(“StudyRequirements”, “DevelopFrontend”).
dprec(“DevelopBackend”, “VerifyBackend”).
dprec(“DevelopBackend”, “DeploySystem”).
dprec(“VerifyBackend”, “DeploySystem”).
con(“DevelopBackend”, “DevelopFrontend”).
con(“DevelopBackend”, “VerifyFrontend”).
con(“DevelopFrontend”, “DevelopBackend”).
con(“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").
role("Manager").
rlAC("Amy", "Manager").

resource("Jessie").
role("SoftwareEngineer").
rlAC("Jessie", "SoftwareEngineer").

resource("Liam").
role("JuniorDeveloper").
rlAC("Liam", "JuniorDeveloper").

resource("Glen").
role("UIDeveloper").
rlAC("Glen", "UIDeveloper").

resource("Oliver").
role("UIDeveloper").
rlAC("Oliver", "UIDeveloper").

resource("Mia").
role("UIDeveloper").
rlAC("Mia", "UIDeveloper").

resource("Lisa").
role("TestingExpert").
rlAC("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").

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

llAC("Manager", "TestingExpert").
llAC("SoftwareEngineer", "JuniorDeveloper").
BPM.... Ease of encoding in ASP


BPM.... Ease of encoding in ASP

Study requirements
Develop backend
Verify backend
Deploy system
Develop frontend
Verify frontend
Deploy system

Manager
Software Engineer
UI Developer
Testing Expert

Junior Dev.

Jessie
Liam
Glen
Oliver
Mia
Lisa
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).

• Resource-specific activity durations
• Role-specific activity durations
• Default activity durations

allowedRAD(R,A,D) :- IsaDuration(L,A,D), not rsaDuration(R,A,_), rlAC(R,L), alAC(A,L).
allowedRAD(R,A,D) :- defaultDuration(A,D), not rsaDuration(R,A,_), not IsaDuration(L,A,_), rlAC(R,L), alAC(A,L).
BPM.... Guess and Check

% generate allocations
1<=\{allocation(R,A,S,C): time(S), time(C), allowedRAD(R,A,D), C=S+D}<=1 :- activity(A).

% check for scheduling of the preceding activities
:- dPrec(A1,A2), allocation(_,A1,_,C1), allocation(_,A2,S2,__), C1>S2.

% check for scheduling of the concurrent activities
:- conc(A1,A2), allocation(R,A1,S1,__), allocation(R,A2,S2,__), S2<=S1, C2>S1, A1<A2.
:- conc(A1,A2), allocation(R,A1,_,C1), allocation(R,A2,S2,C2), S2<C1, C2>C1, A1<A2.
:- conc(A1,A2), allocation(R,A1,S1,C1), allocation(R,A2,S2,C2), S2>S1, C2<C1, A1<A2.

% minimize makespan
:\~ makespan(U). [U]
BPM ... Easily Extensible encoding

- Teams
  - E.g., `team("team_backend",R) :- rlAC(R,"JuniorDeveloper").`
    `team("team_backend",R) :- rlAC(R,"SoftwareEngineer").`
    `tRequirement("DevelopBackend","team_backend").`
BPM ... Easily Extensible encoding

• Teams
  • E.g., team("team_backend",R) :- rlAC(R,"JuniorDeveloper").
    team("team_backend",R) :- rlAC(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

- Teams
  - E.g., `team("team_backend",R) :- rAC(R,"JuniorDeveloper").`
  - `team("team_backend",R) :- rAC(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.`
RABP in Practice: Camunda BPMS Integration

Saimir Bala, Giray Havur, Simon Sperl, Simon Steyskal, Alois Haselbock, Jan Mendling, and Axel Polleres.


https://www.youtube.com/watch?v=3IPxXoQ9m4
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
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
Contributions to RQ 2
An ASP Systems Benchmark for RABP
ASP vs CP: Performance Evaluation

- Running RABP Instances in ASP and CP

```
<table>
<thead>
<tr>
<th>INPUT</th>
<th>PROCESSING</th>
<th>INTERMEDIARY MODEL</th>
<th>SOLVING</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>RABP ASP encoding</td>
<td>ASP grounder</td>
<td>Ground program</td>
<td>ASP solver</td>
<td>Result</td>
</tr>
<tr>
<td>RABP ASP problem instance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RABP MiniZinc encoding</td>
<td>MiniZinc flattener</td>
<td>FlatZinc model</td>
<td>CP solver</td>
<td>Result</td>
</tr>
<tr>
<td>RABP MiniZinc problem instance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- Comparison of four ASP systems against four CP systems
  - ASP systems: gringo+clasp, gringo+wasp, idlv+clasp, idlv+wasp
  - CP systems: mzn2fzn+gcode, 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**
- gringo+clasp
- gringo+wasp
- idlv+clasp
- idlv+wasp

**CP Systems**
- mzn2fzn+gecode
- mzn2fzn+chuffed
- mzn2fzn+hcsp
- mzn2fzn+or-tools
## ASP vs CP

<table>
<thead>
<tr>
<th></th>
<th>ASP</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>++</td>
<td>+++</td>
</tr>
<tr>
<td>Readability</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Encoding RABP Problem</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td>Encoding RABP Instances</td>
<td>+++</td>
<td>+</td>
</tr>
</tbody>
</table>
Representing a wide variety of resources in RABP


---

**RESEARCH-ARTICLE**

Benchmarking Answer Set Programming systems for resource allocation in business processes

**Authors:** Giray Havur, Cristina Cabanillas, Axel Polleres **Authors Info & Claims**


---

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
- W3C

**Part 3:**
ca. 2014 – 2022...
- NUI Galway
- OÊ Gaillimh
- WU Wien
- Siemens

Business Process Management