MODELLING, ANALYSIS, AND OPTIMIZATION OF BPMN PROCESSES

GWEN SALAÜN

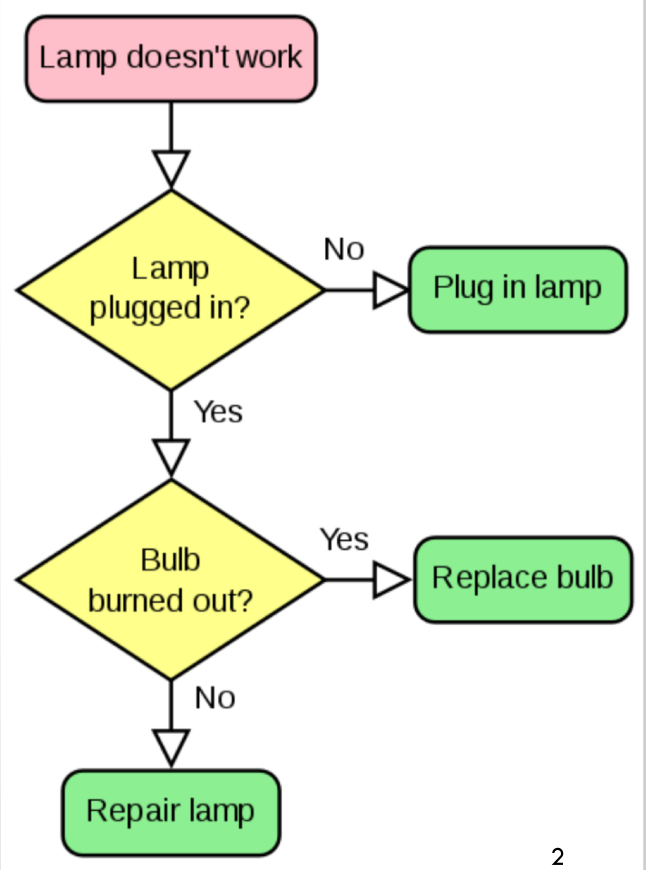
UNIVERSITÉ Grenoble Alpes

JOINT WORK WITH

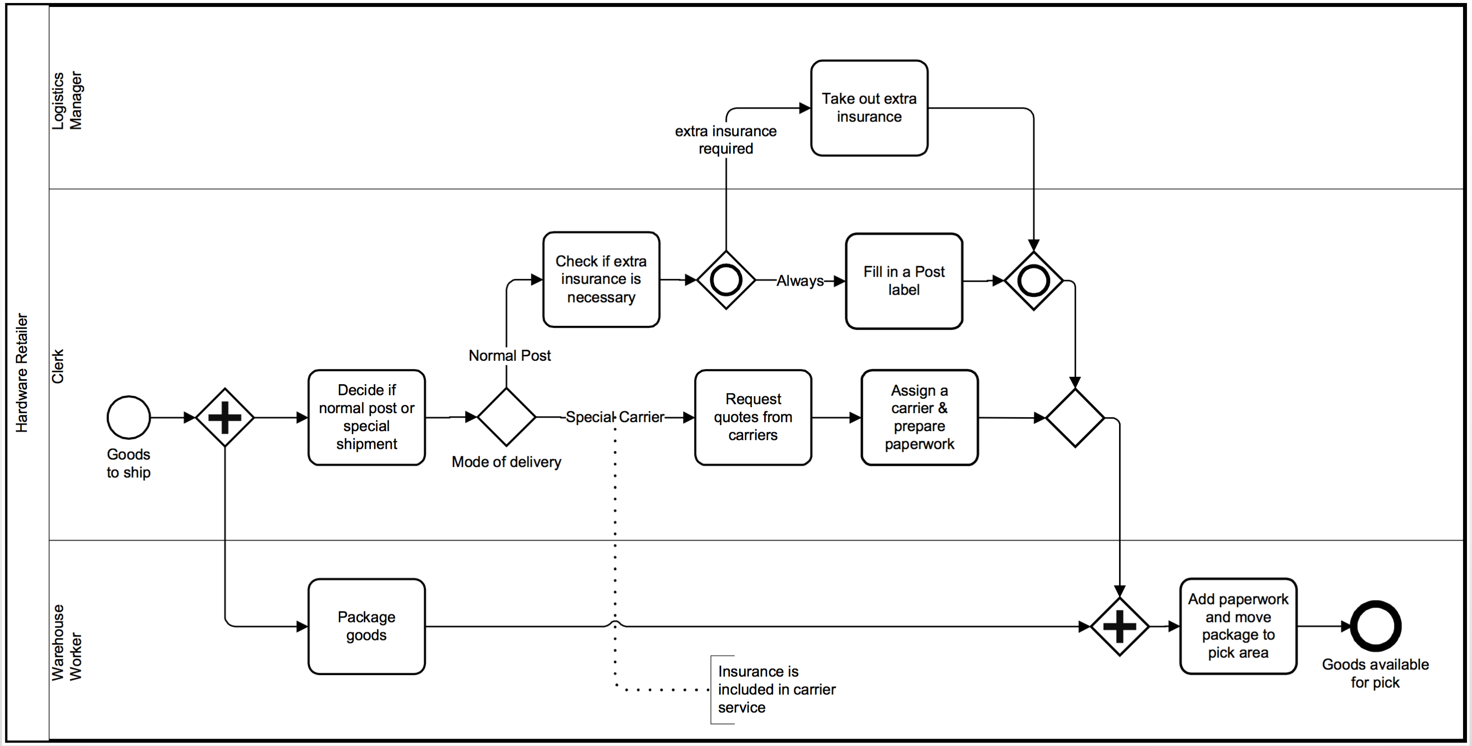
FRANCISCO DURÁN, YLIÈS FALCONE, CAMILO ROCHA AND AHANG ZUO

BUSINESS PROCESSES

"A BUSINESS PROCESS IS A COLLECTION OF RELATED, STRUCTURED ACTIVITIES OR TASKS BY PEOPLE OR EQUIPMENT IN WHICH A SPECIFIC SEQUENCE PRODUCES A SERVICE OR PRODUCT FOR A PARTICULAR CUSTOMER" (WIKIPEDIA)



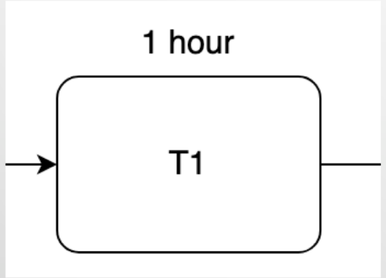
MODELLING PROCESSES WITH BPMN

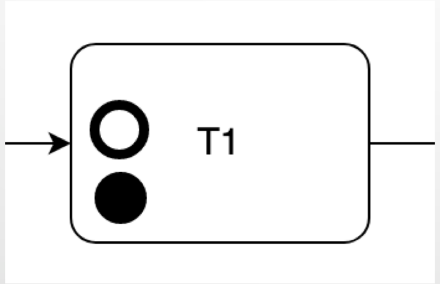


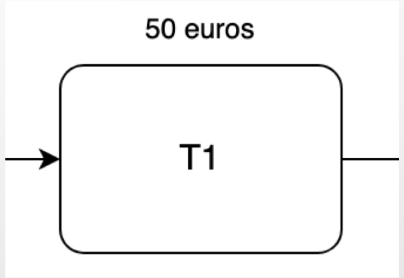
BPMN 2.0 (BUSINESS PROCESS MODELLING NOTATION) WAS PUBLISHED AS AN ISO STANDARD IN 2013

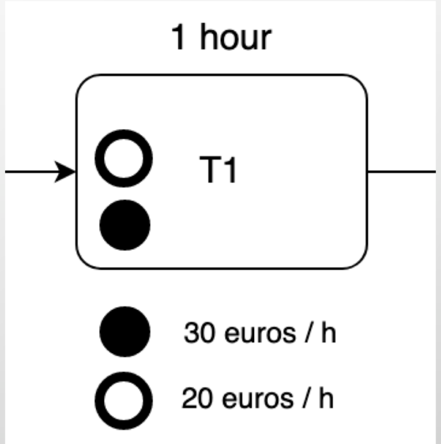
EXTENDED BPMN

BPMN EXTENDED WITH QUANTITATIVE INFORMATION: DURATIONS, RESOURCES, COSTS, POWER CONSUMPTION, CO_2 EMISSION, ETC. (NOT MANDATORILY ALL AT THE SAME TIME)

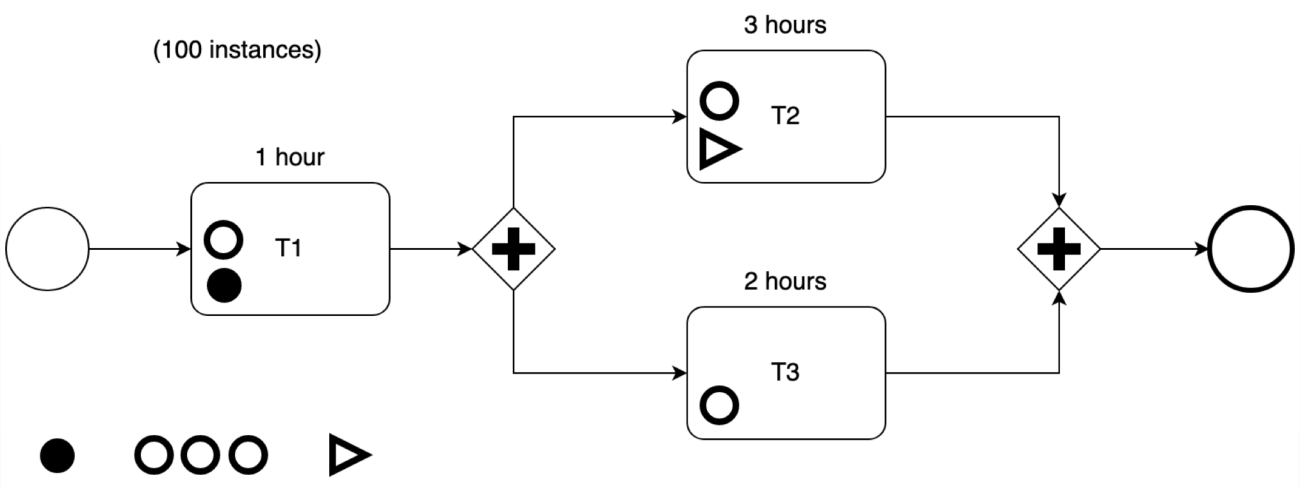








MOTIVATING EXAMPLE

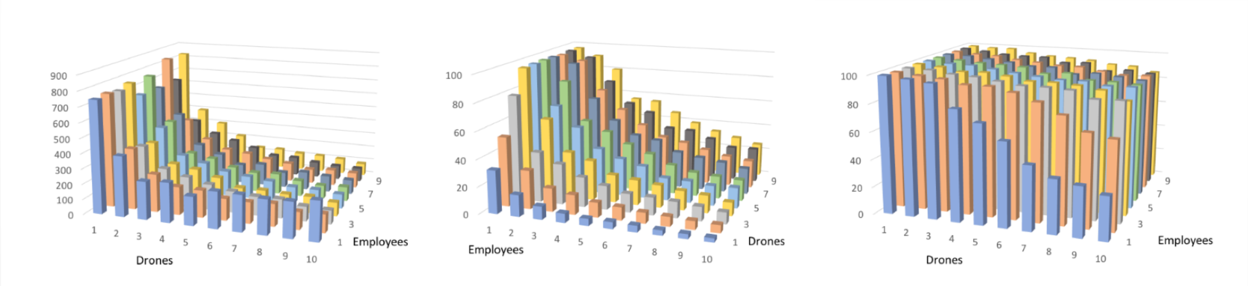


BPMN WITH TIME AND RESOURCES

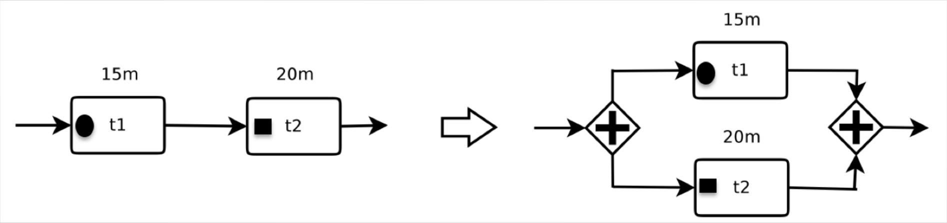
- HOW LONG DOES IT TAKE TO ONE INSTANCE TO COMPLETE IN AVERAGE ?
- HOW MUCH ARE EACH (TYPE OF) RESOURCE BUSY ?
- WHAT IS THE COST OF THIS EXECUTION ?
- WHAT IS THE OPTIMAL NUMBER OF RESOURCES FOR OPTIMIZING EXECUTION TIME AND/OR COST ?

CONTRIBUTIONS

- AN EXTENSION OF BPMN FOR MAKING EXPLICIT DESCRIPTION OF TIME & RESOURCES
- AUTOMATED ANALYSIS TECHNIQUES FOR COMPUTING METRICS ON RESOURCE USAGE, EXECUTION TIME AND COST



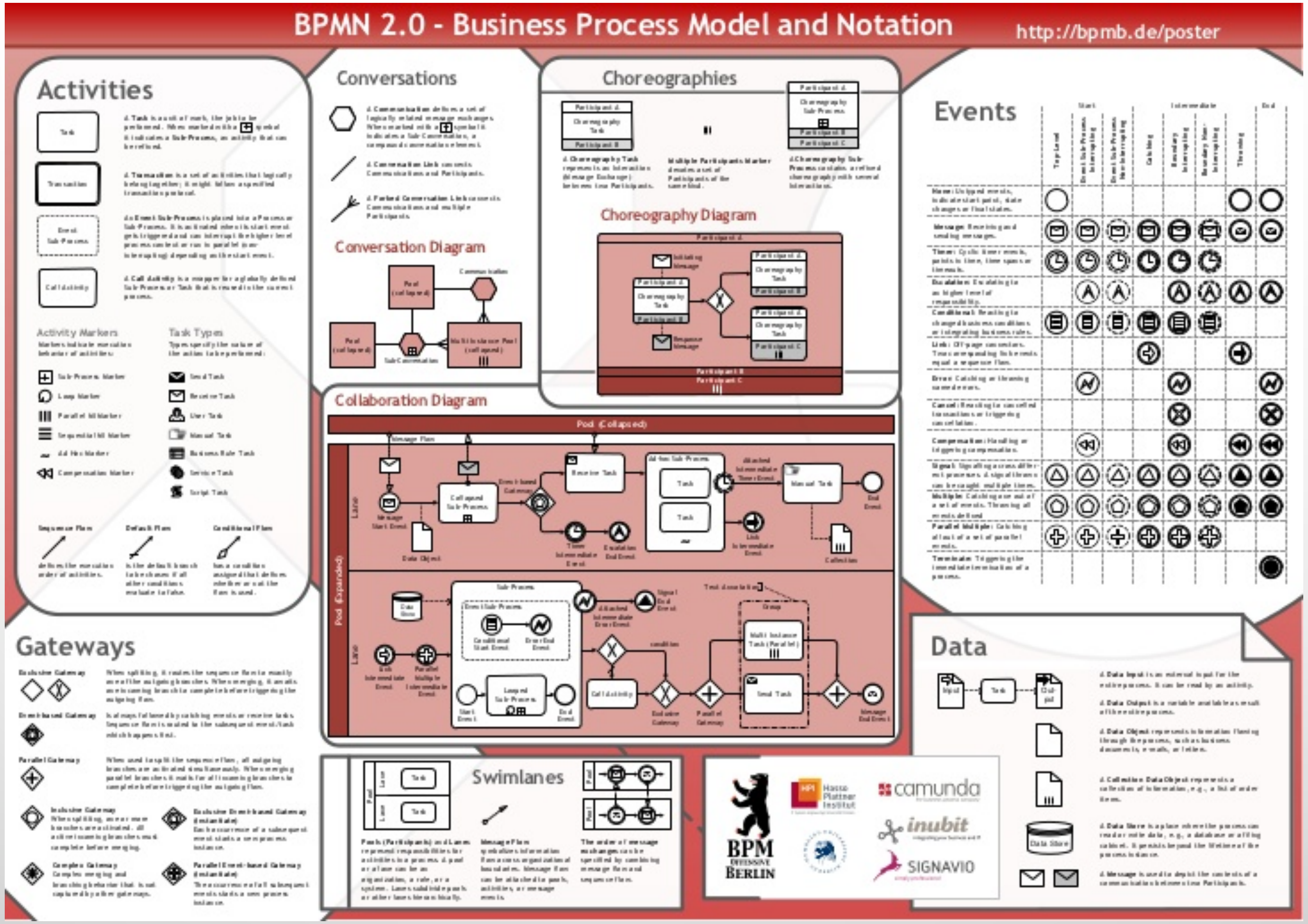
- PRESENTATION OF TWO POSSIBLE APPROACHES: DESIGN TIME *VS.* RUNTIME ANALYSIS
- ALTERNATIVE APPROACH FOR OPTIMIZING PROCESSES: REFACTORING OF BPMN PROCESSES



OUTLINE

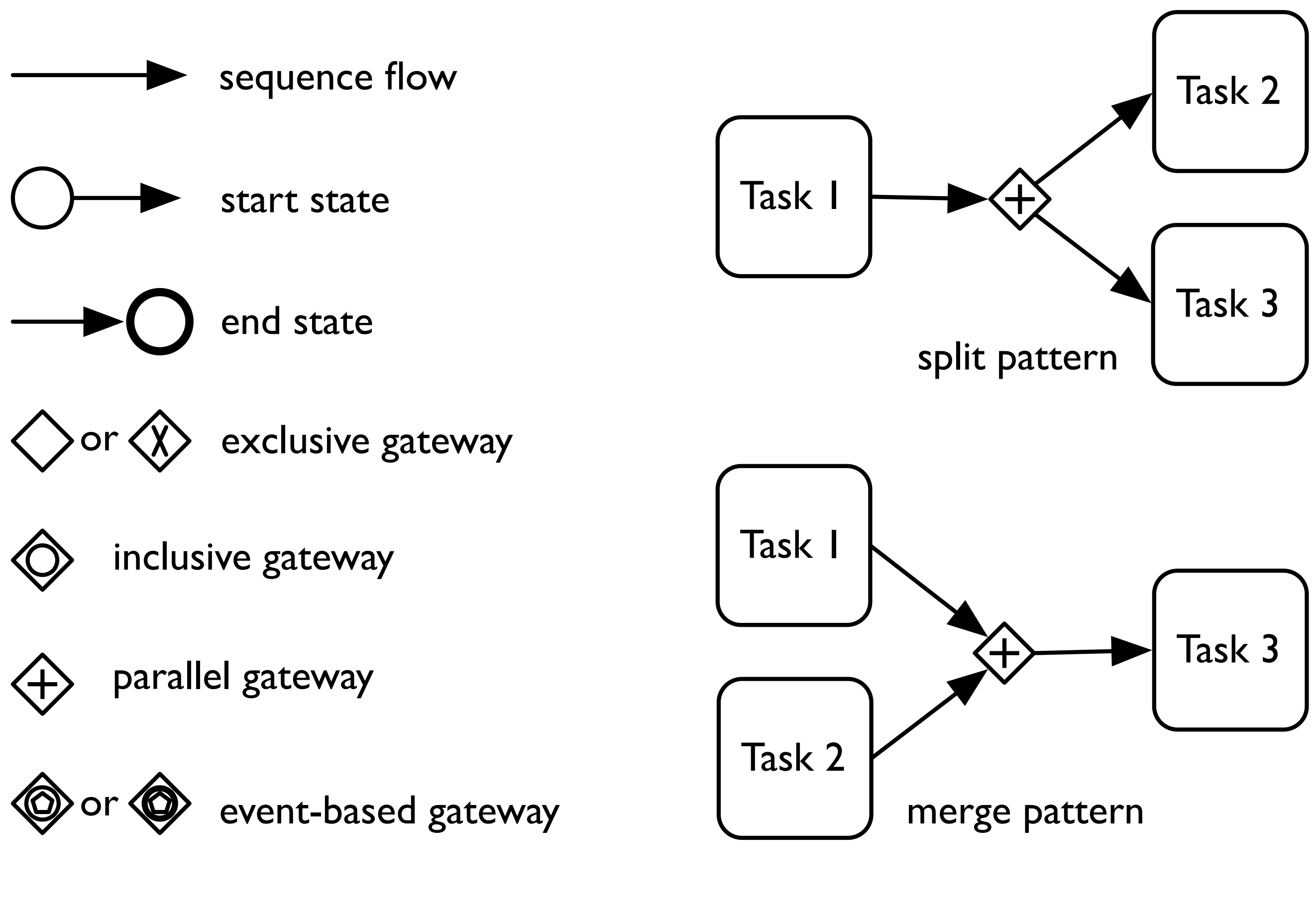
1. BPMN WITH TIME AND RESOURCES
2. DESIGN TIME ANALYSIS
3. RUNTIME ANALYSIS
4. AUTOMATED REFACTORING
5. TOOL SUPPORT
6. CONCLUDING REMARKS

BPMN

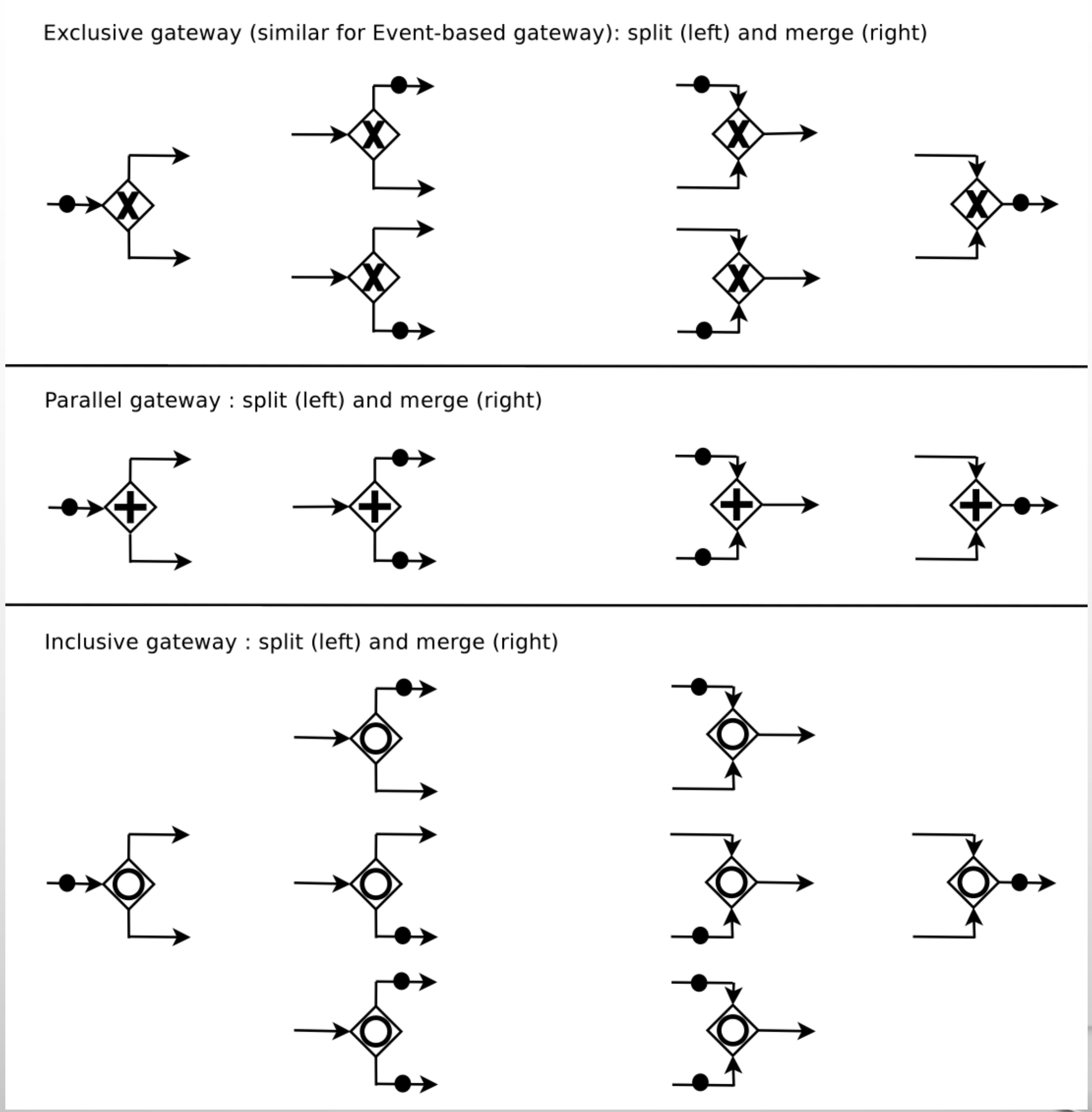


SEVERAL MODELLING AND DEVELOPMENT FRAMEWORKS: ACTIVITI, BONITA, CAMUNDA, SIGNAVIO, …

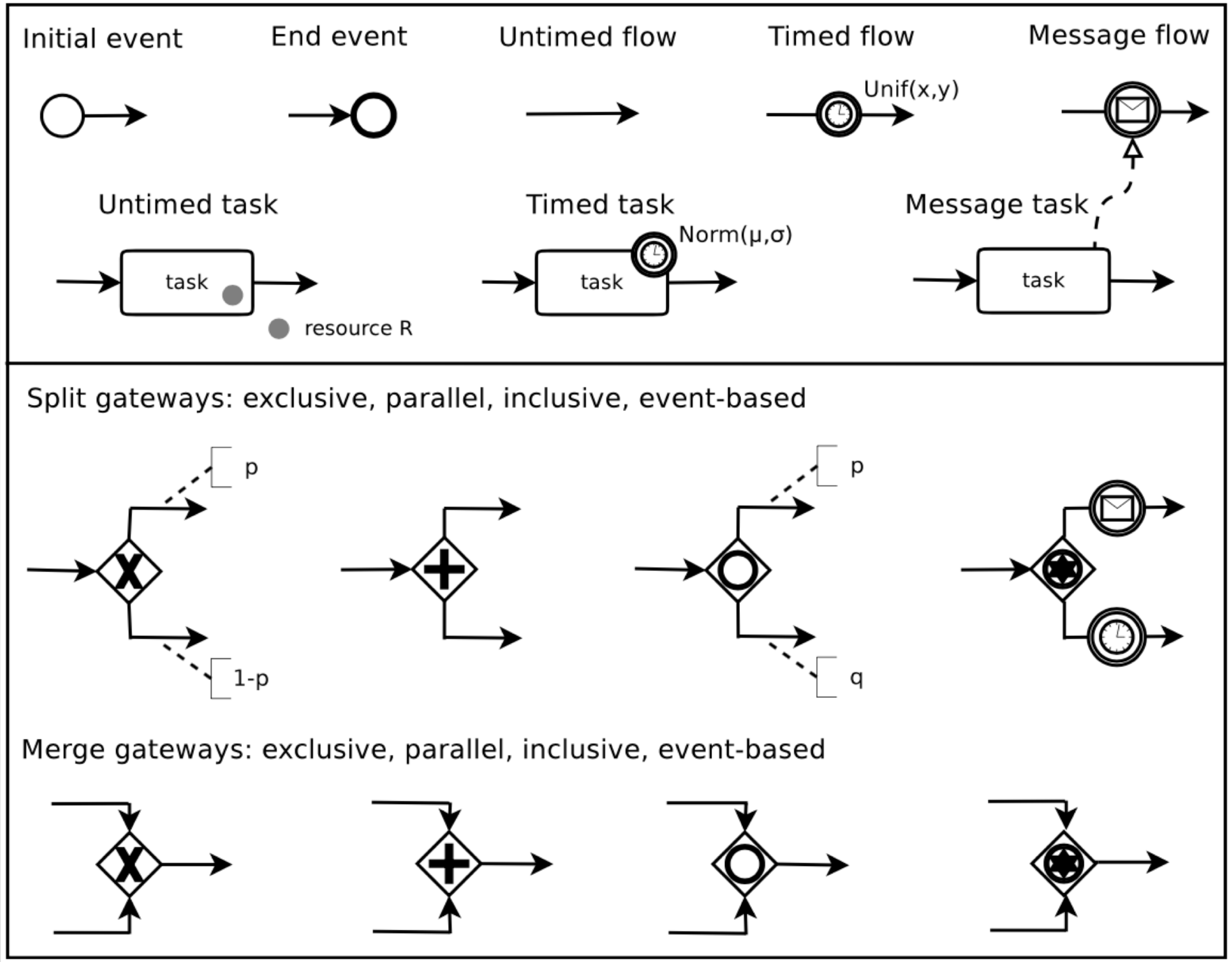
CONTROL FLOWS AND GATEWAYS



SEMANTICS


Exclusive gateway (similar for Event-based gateway): split (left) and merge (right)
Parallel gateway : split (left) and merge (right)
Inclusive gateway : split (left) and merge (right)

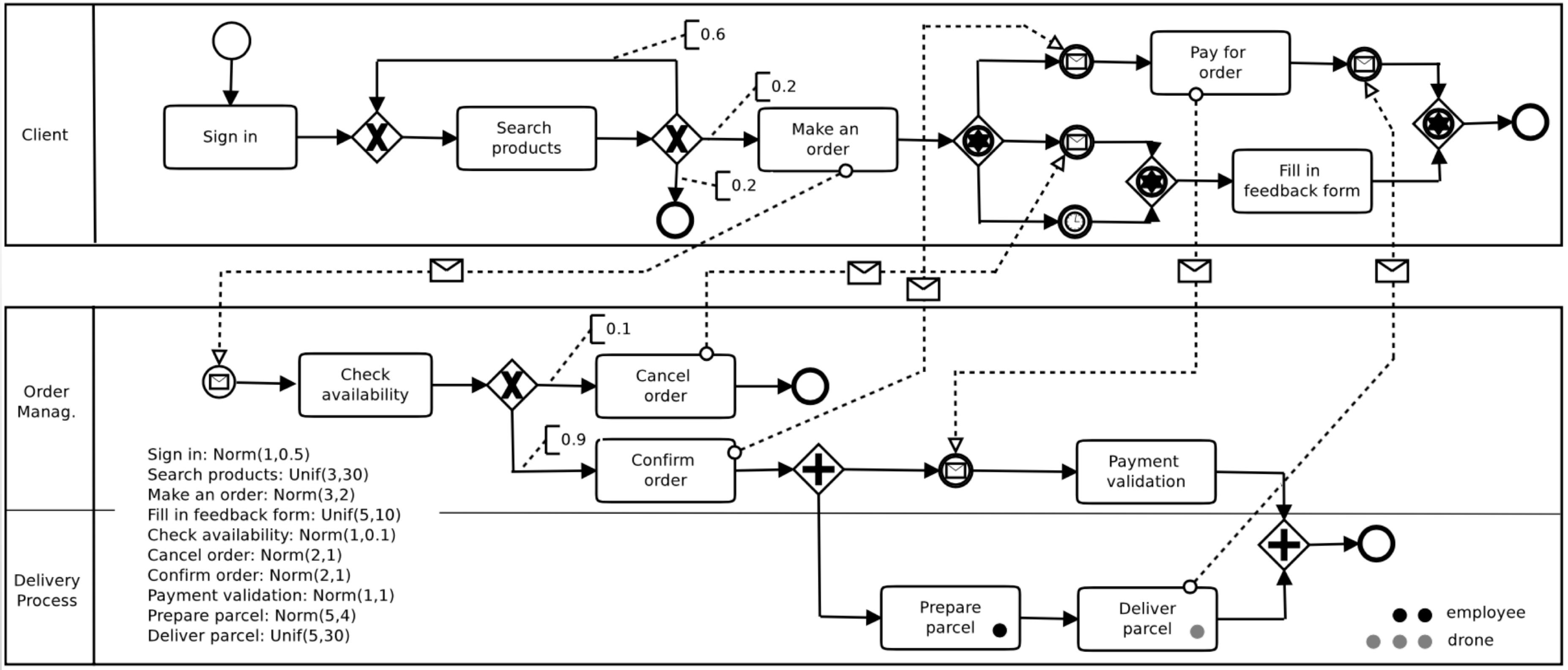
EXTENDED BPMN



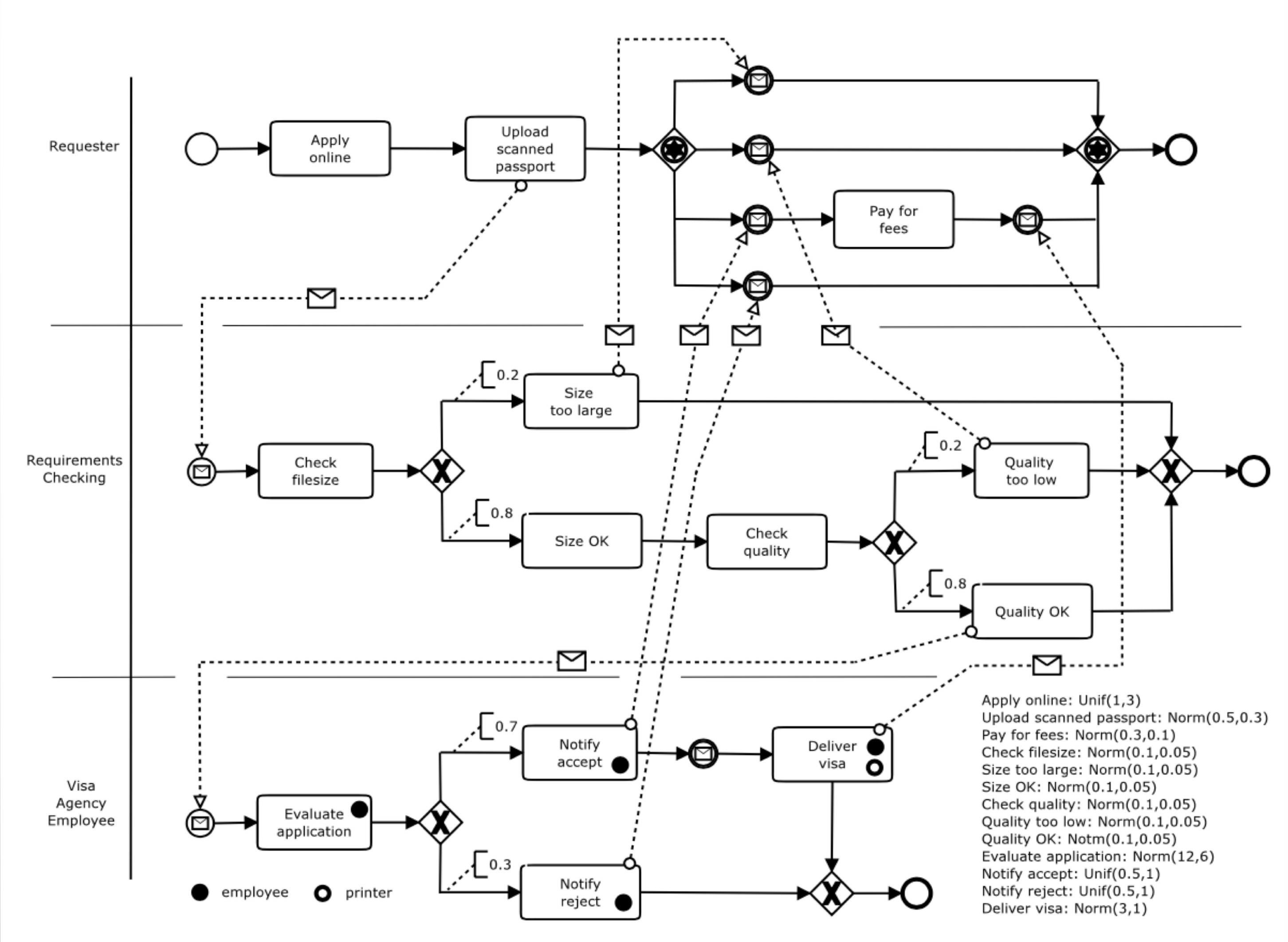
IN THIS WORK, WE SUPPORT

- ACTIVITY DIAGRAMS
- COLLABORATION DIAGRAMS

EXAMPLE: PRODUCT ORDER AND DELIVERY



EXAMPLE: VISA APPLICATION PROCESS

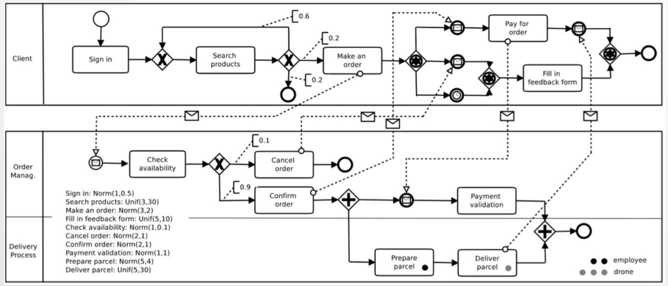


OUTLINE

1. BPMN WITH TIME AND RESOURCES
2. DESIGN TIME ANALYSIS
3. RUNTIME ANALYSIS
4. AUTOMATED REFACTORING
5. TOOL SUPPORT
6. CONCLUDING REMARKS

FORMAL MODEL

FORMAL SEMANTICS IS GIVEN TO THIS EXTENDED SUBSET OF BPMN BY ENCODING INTO REWRITING LOGIC



1.

```
< pid : Process |
    nodes : (start(initial, cf1),
             merge(g1, exclusive, (cf2, cf5), cf3),
             split(g2, exclusive, cf4, ((cf5, 0.6) (cf6, 0.2) (cf7, 0.2))),
             split(g3, eventbased, cf8, (cf9, cf10, cf11)),
             task(t10, "Prepare parcel", mf7, df1, Norm(5.0, 4.0), employee, empty),
             task(t11, "Deliver parcel", df1, df2, Unif(5.0, 30.0), drone, parceldelivered),
             ...),
    flows : (flow(cf1,  0),
             flow(cf9,  0, message(orderconfirmed, "Order confirmed")),
             flow(cf10, 0, message(ordercanceled, "Order canceled")),
             flow(cf11, 0, timer(timeout, 60)),
             ...) >
```

2.

```
< s : Simulation | tokens : ...,             ---- scheduler
                   gtime : ...,              ---- global time
                   resources : ...,          ---- resource set
                   events : ...,             ---- event set
                   process-execs : ....,     ---- execution times
                   sync-times : ...,         ---- synchronization times
                   task-times : ...,         ---- task execution times
                   ... >
```

3.

```
rl [initTask] :
   < PId : Process |
     nodes : (task(NId, TaskName, FId1, FId2, SE, RIds, SEI), Nodes), Atts >
   < SId : Simulation |
     tokens : (token(TId, FId1, 0) Tks),
     task-tstamps : TTSs, gtime : T, resources : Rs, Atts1 >
   < CId : Counter | counter : N >
=> if allResourcesAvailable(RIds, Rs)
   then < PId : Process |
          nodes : (task(NId, TaskName, FId1, FId2, SE, RIds, SEI), Nodes), Atts >
        < SId : Simulation |
          tokens : insert(Tks, token(TId, NId, time(eval(SE, N)))),
          task-tstamps : if TTSs[TId][NId] == undefined
                         then insert(TId, insert(NId, T, TTSs[TId]), TTSs)
                         else TTSs
                         fi,            ---- for loops, stamps get overwritten
          gtime : T,
          resources : grabResources&updateTime(RIds, Rs, time(eval(SE, N)), T), Atts1 >
        < CId : Counter | counter : int(eval(SE, N)) >
   else ...                             ---- if necessary, the scheduler is updated
   fi .
```

THE MAUDE SPECIFICATION CONSISTS OF THREE PARTS:

1. PROCESS SYNTAX IS REPRESENTED AS AN OBJECT WITH A SET OF FLOWS AND A SET OF NODES AS ATTRIBUTE
2. THE SIMULATION OBJECT KEEPS INFORMATION ON THE EXECUTION OF THE PROCESS
3. REWRITING RULES REPRESENT HOW TOKENS EVOLVE THROUGH THE PROCESS AND EVENTS ARE FIRED

AUTOMATED ANALYSIS OF PROPERTIES

- SIMULATION TECHNIQUES TAKE AS INPUT: A PROCESS DESCRIPTION, A SPECIFICATION OF RESOURCES AND A WORKLOAD (NUMBER OF INSTANCES)

- TIMING PROPERTIES:
  - AVERAGE EXECUTION TIMES (AET) OF A PROCESS EXECUTION + SHORTEST / LONGEST EXECUTION
  - AVERAGE SYNCHRONIZATION TIME FOR MERGE GATEWAYS

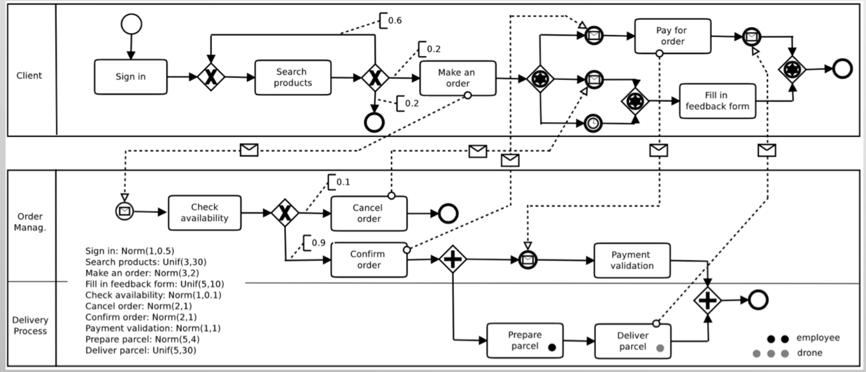
- RESOURCE-BASED PROPERTIES:
  - THE GLOBAL TIME USAGE (GTU_R) OF ALL REPLICAS OF EACH RESOURCE *R*
  - THE GTU PER REPLICA OF RESOURCE *R* (GTU^1_R)
  - THE AVERAGE USAGE PERCENTAGE FOR EACH RESOURCE *R* OVER THE GLOBAL EXECUTION TIME (UP^1_R)

- TOTAL COST

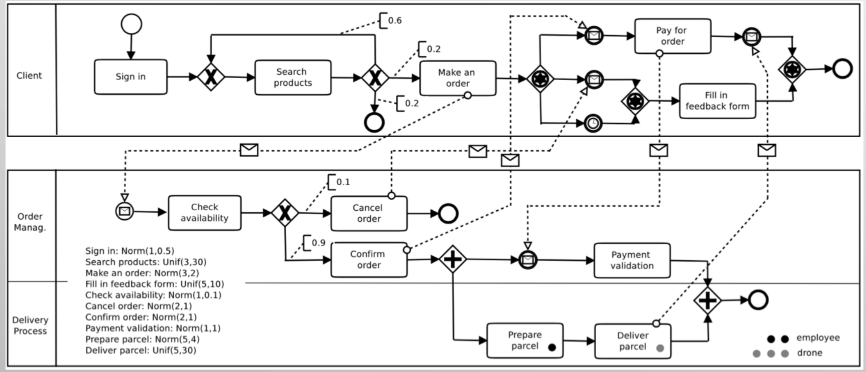
EXAMPLE: SIMULATION RESULTS

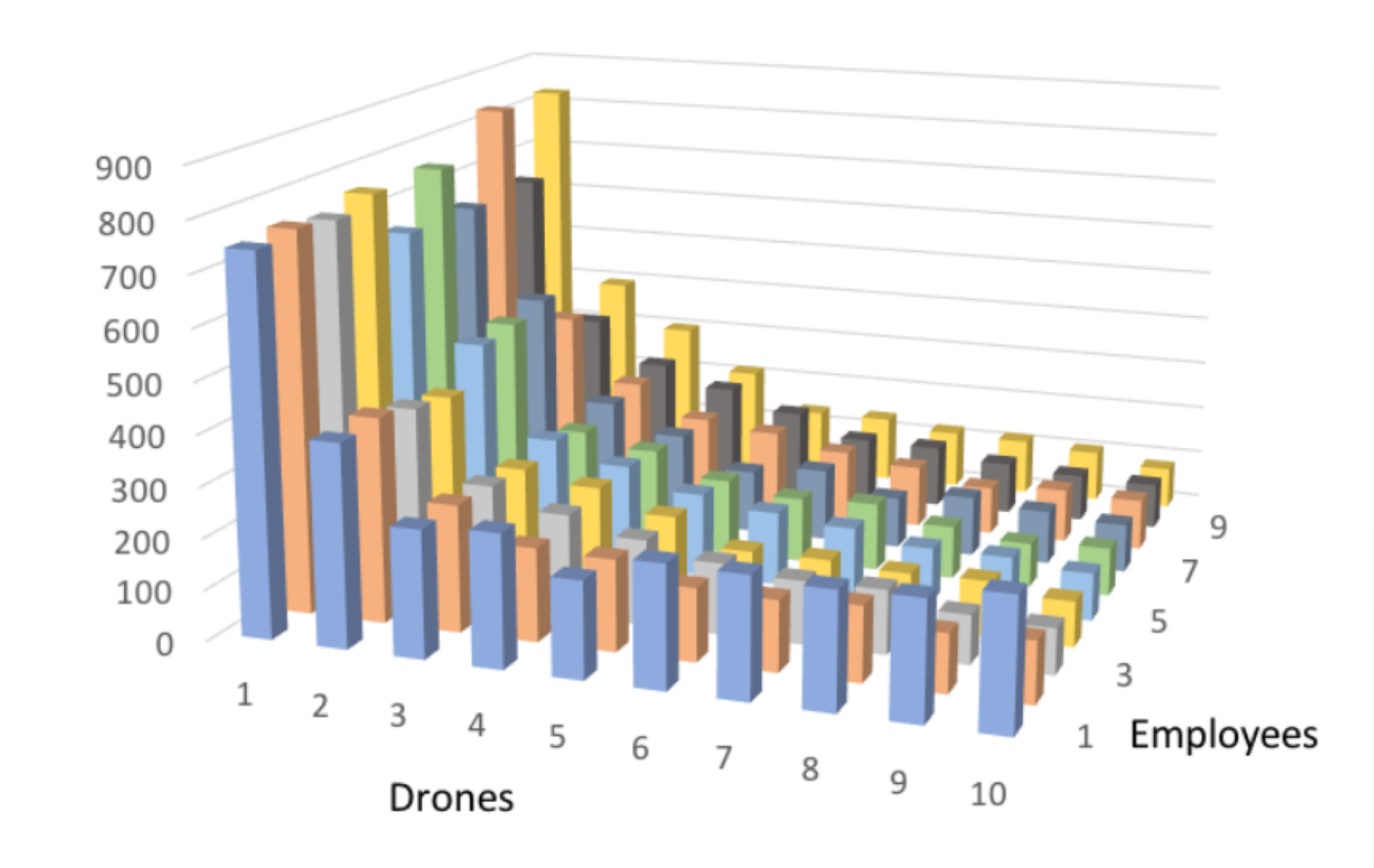
Numb. inst.	AET	Var	AST_{g8}	AST_{ee}	Total time	Resources						Analysis time
						GTU_e	GTU_e^1	UP_e^1	GTU_d	GTU_d^1	UP_d^1	
100	106	72	58	58	326	271	135	41	853	284	87	5s
200	185	134	71	139	670	514	257	38	1892	630	94	26s
400	284	173	98	237	1132	994	497	43	3270	1089	96	189s
800	506	294	145	459	2217	1867	933.6	42	6525	2171	98	1233s
1600	891	473	240	844	4187	3714	1857	44	12428	4142	98	7909s

Table 1. Experimental results for the running example (2 employees, 3 drones)

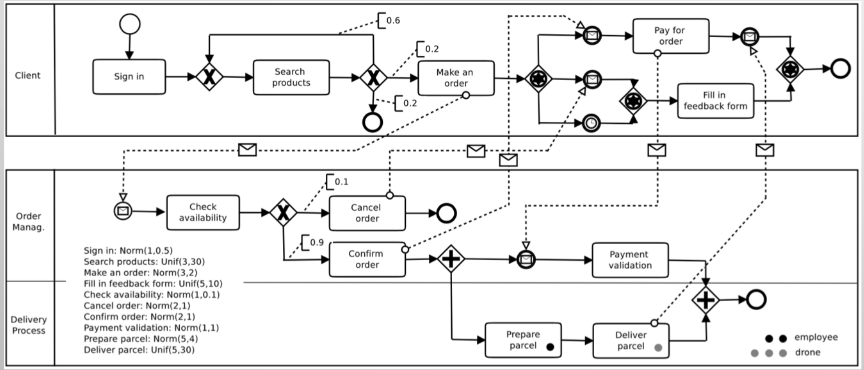


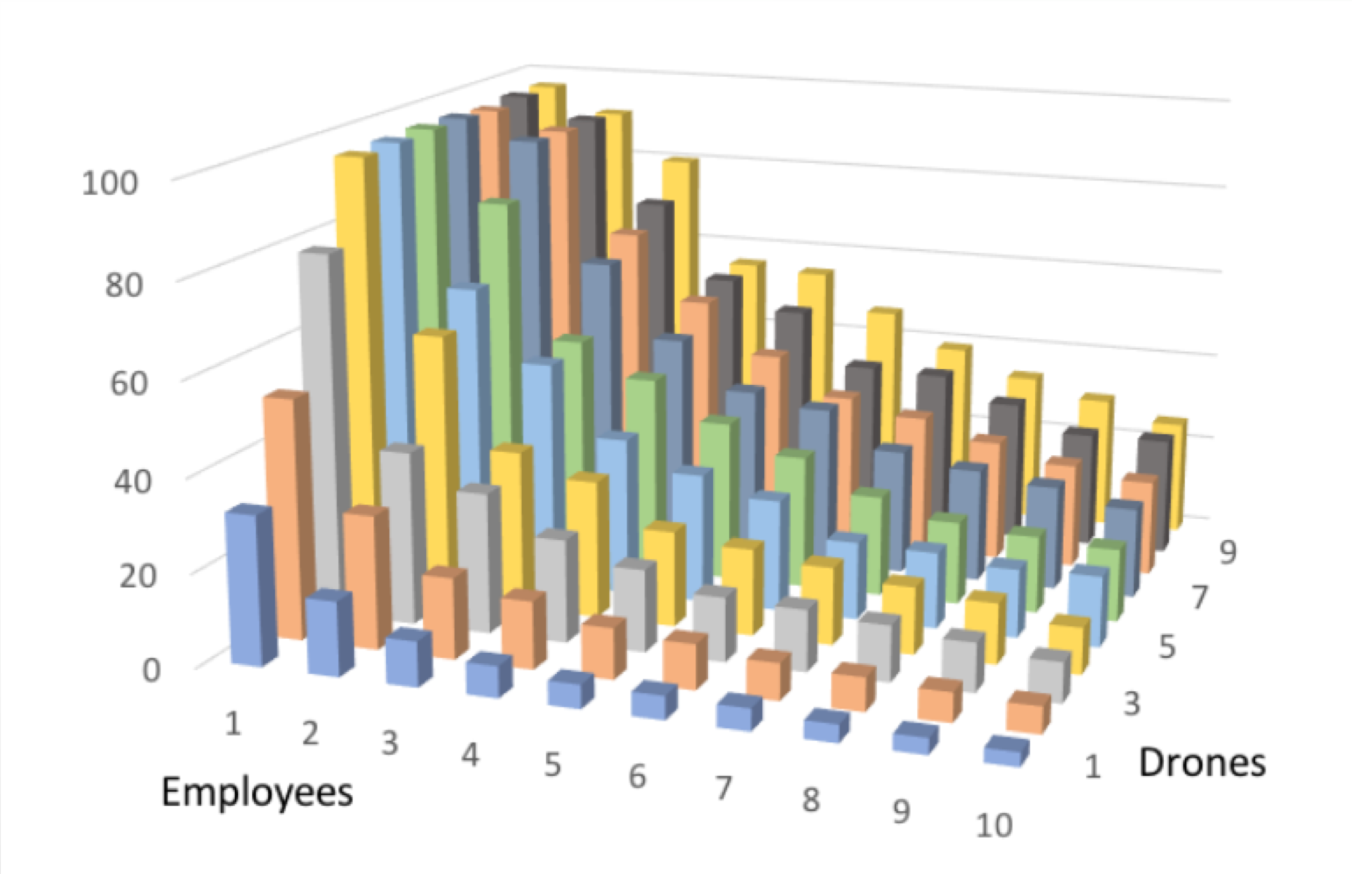
EXAMPLE: AVERAGE EXECUTION TIME



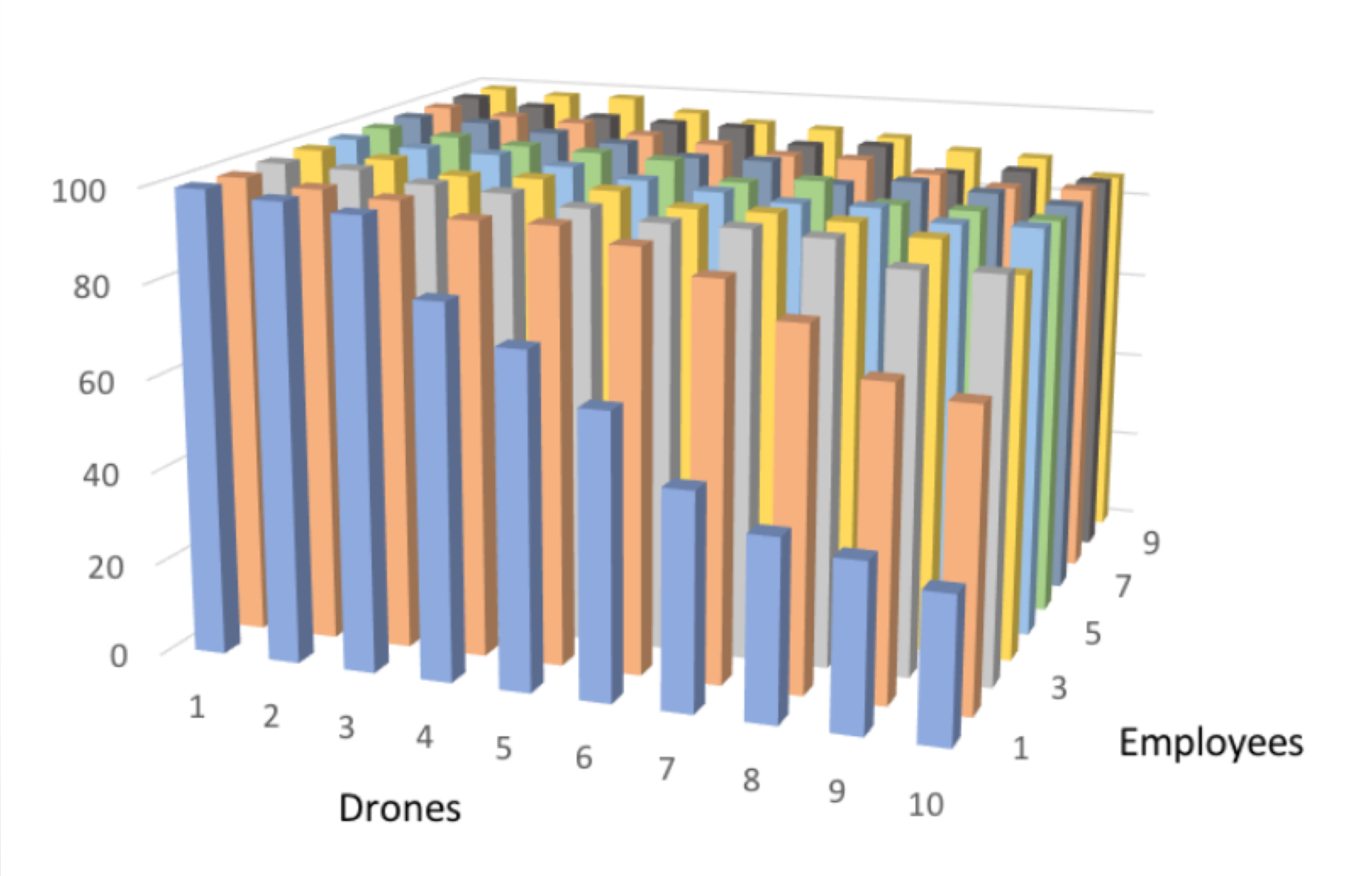


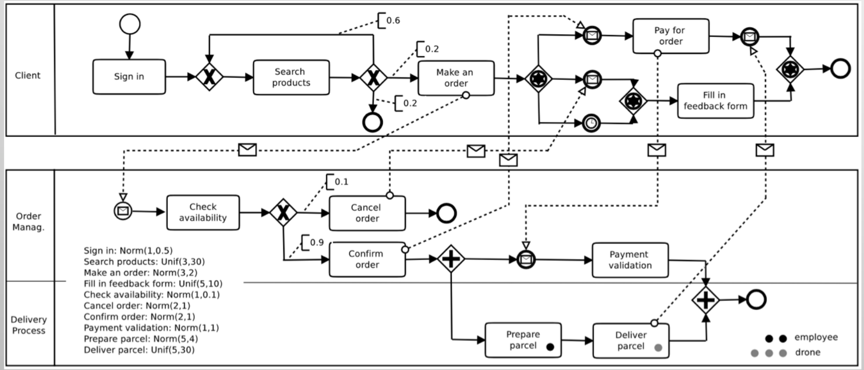
EXAMPLE: AVERAGE USAGE (EMPLOYEE)





EXAMPLE: AVERAGE USAGE (DRONE)





OPTIMAL SOLUTIONS

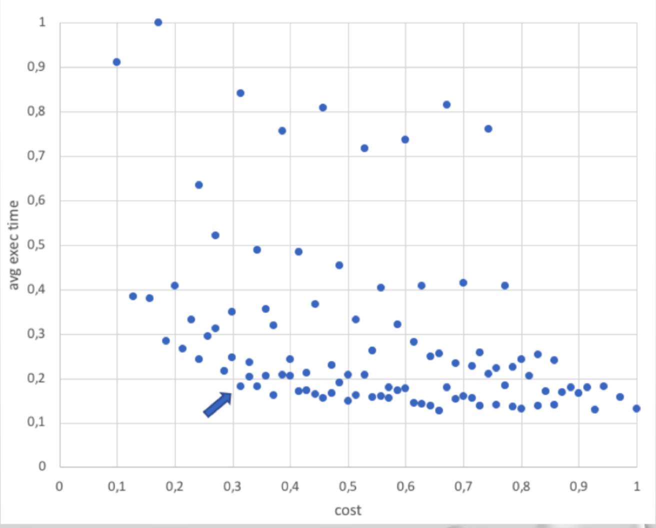
COMPUTATION OF THE OPTIMAL ALLOCATION USING MULTI-OBJECTIVE OPTIMIZATION TECHNIQUES

$$\min_{x \in X} \sum_{i \in \{c,t\}} w_i f_i(x)$$

WHERE F_C AND F_T REPRESENT DRONE/EMPLOYEE COST AND AVERAGE EXECUTION TIME

DRONE COST = 20€
EMPLOYEE COST = 50€
$w_c = 0.4$
$w_t = 0.6$

GOAL: MINIMIZE DELIVERY TIME

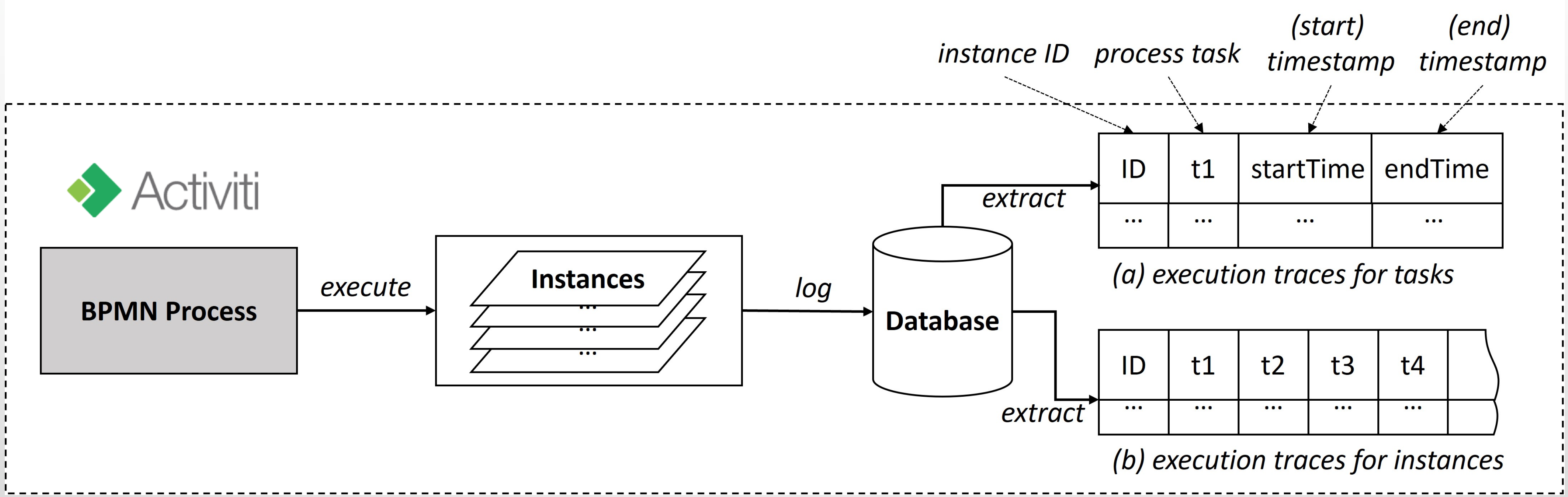


OPTIMAL SOLUTION:
6 DRONES AND 2 EMPLOYEES

OUTLINE

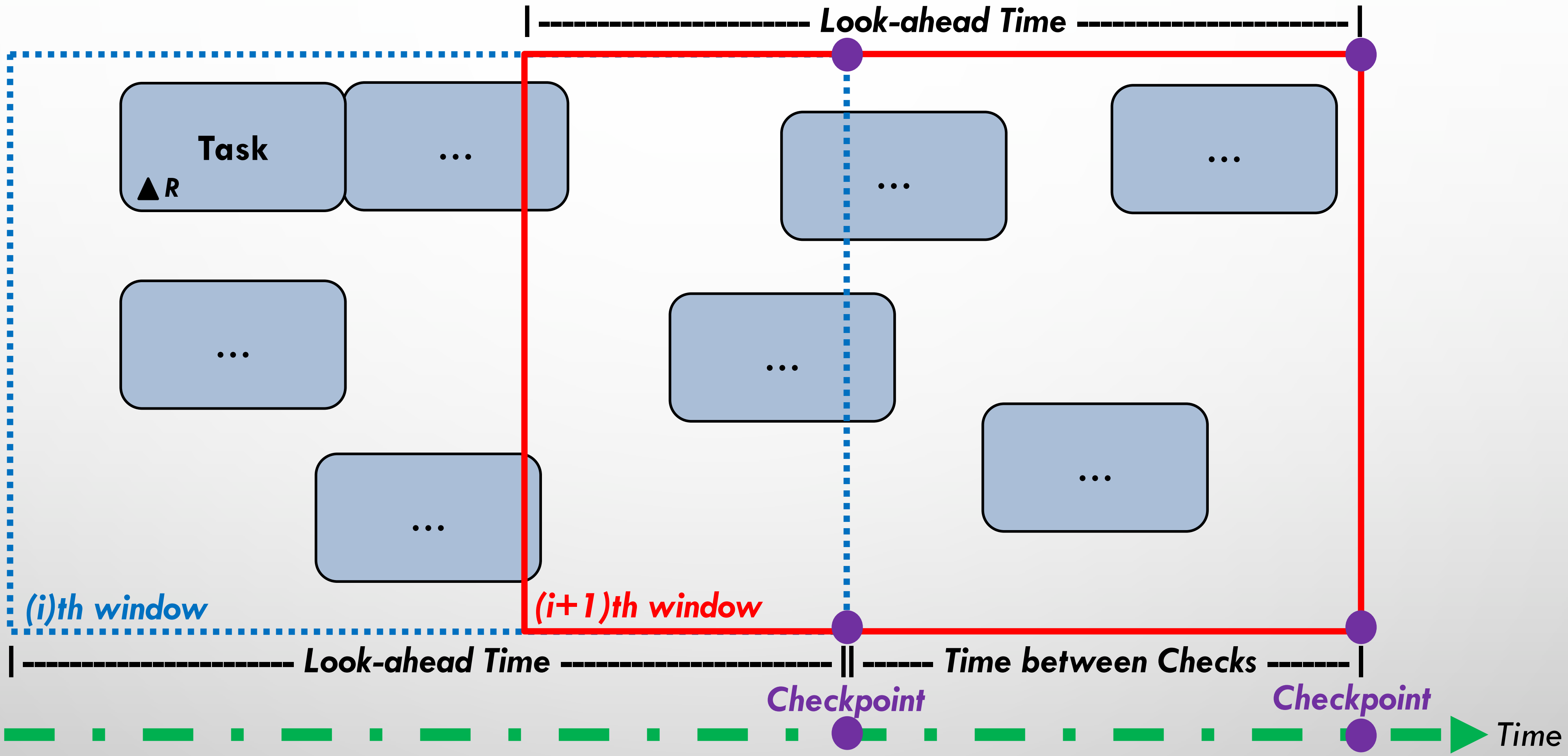
1. BPMN WITH TIME AND RESOURCES
2. DESIGN TIME ANALYSIS
3. RUNTIME ANALYSIS
4. AUTOMATED REFACTORING
5. TOOL SUPPORT
6. CONCLUDING REMARKS

INSTRUMENTATION



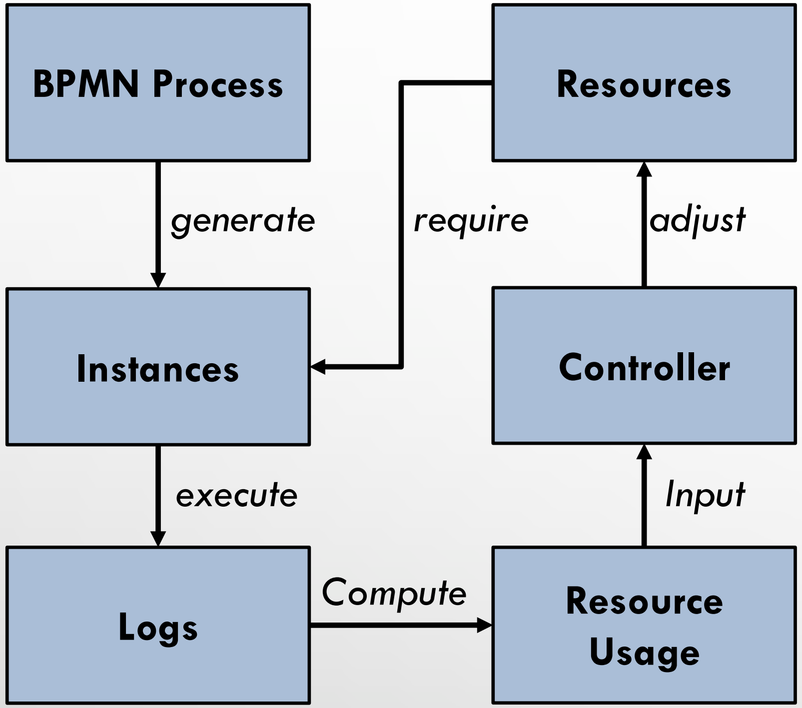
NO RESTRICTION ON THE BPMN SYNTAX

COMPUTATION OF PROPERTIES



COMPUTE THE PROPERTIES FOR EACH WINDOW (E.G., RESOURCE USAGE OR AET)

DYNAMIC RESOURCE ALLOCATION



OVERVIEW OF THE APPROACH

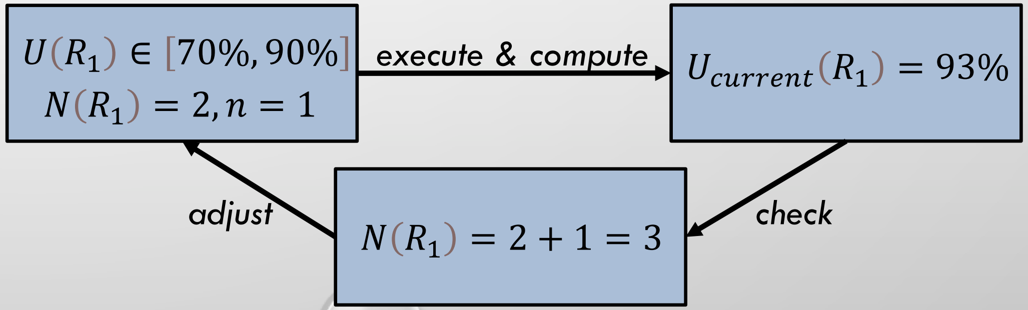
$U(R)$: USAGE OF RESOURCE ***R***.
$U_{current}(R)$: *CURRENT USAGE OF RESOURCE* ***R***.
$N(R)$: *NUMBER OF REPLICAS OF RESOURCE* ***R, N(R)>0.***

EACH RESOURCE ***R*** IS DEFINED WITH A MINIMUM / MAXIMUM USAGE:

$$U(R) \in [minValue, maxValue]$$

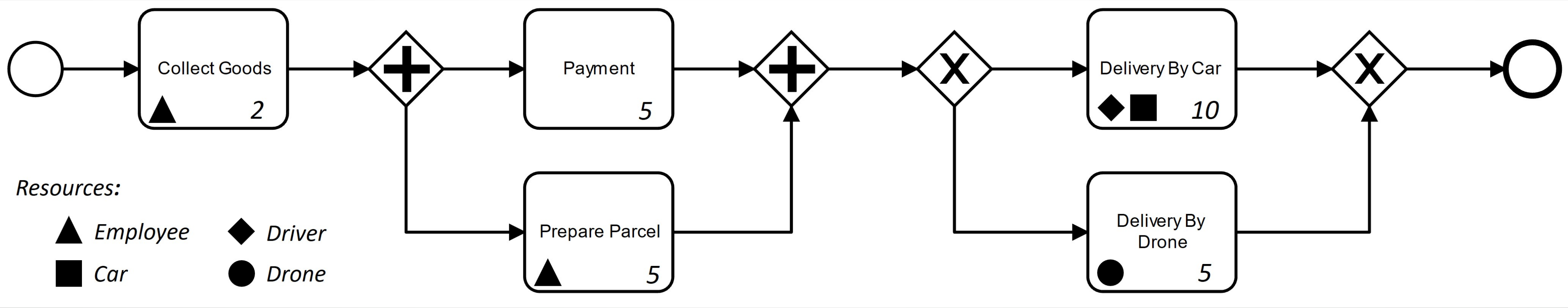
FOR EACH NEW WINDOW, WE CHECK:

- $If\ U_{current}(R) < minValue, then\ N(R) = N(R) - n.$
- $If\ Ucurr_{ent}(R) > maxValue, then\ N(R) = N(R) + n.$
- OTHERWISE, NO OPERATION IS PERFORMED.

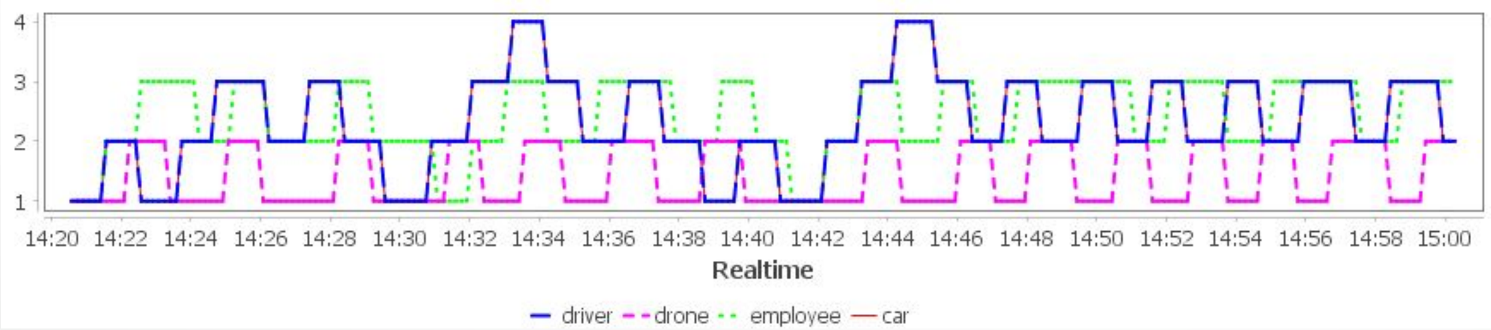


EXAMPLE OF THE APPROACH

EXAMPLE: GOODS DELIVERY PROCESS

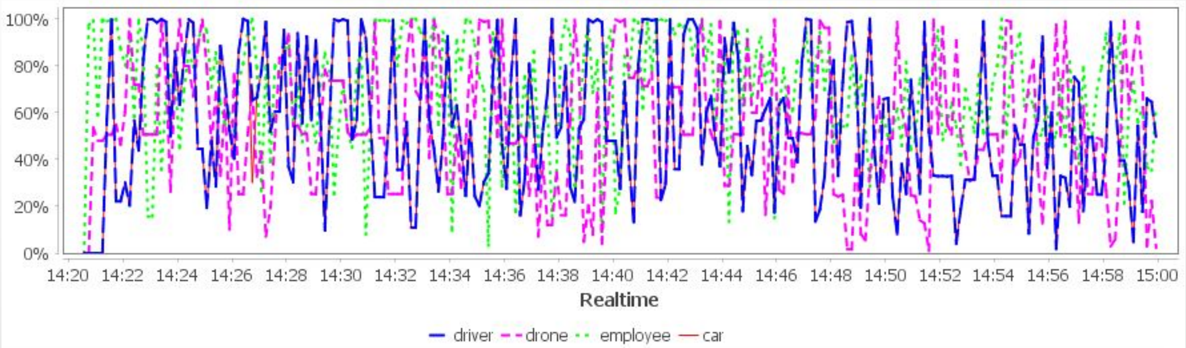


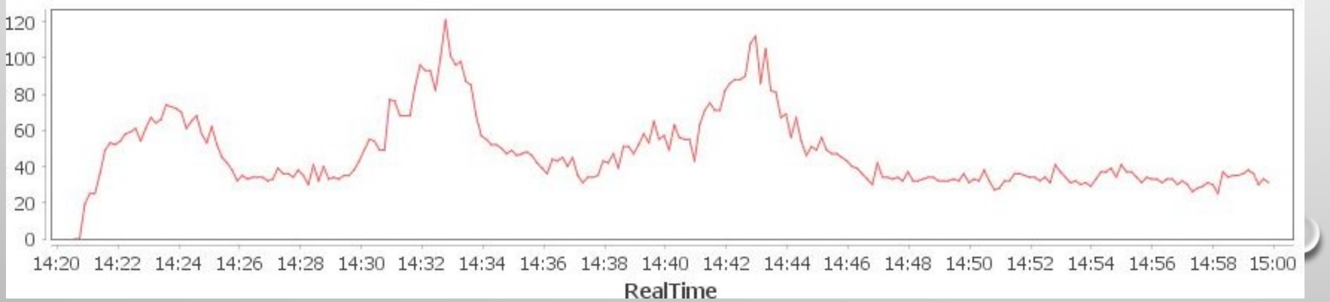
EXAMPLE: EXPERIMENTAL RESULTS



NUMBER OF REPLICAS

RESOURCE USAGE





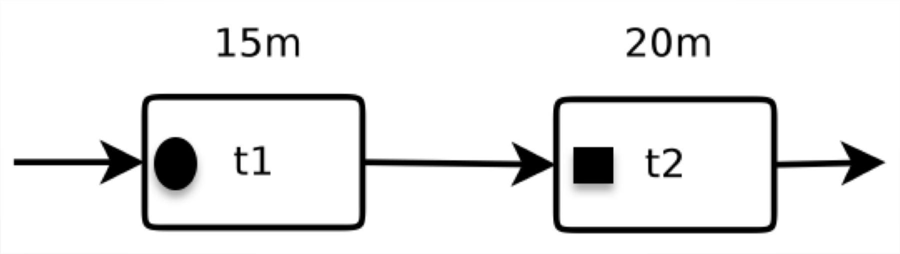
AVERAGE EXECUTION TIME

OUTLINE

1. BPMN WITH TIME AND RESOURCES
2. DESIGN TIME ANALYSIS
3. RUNTIME ANALYSIS
4. AUTOMATED REFACTORING
5. TOOL SUPPORT
6. CONCLUDING REMARKS

MOTIVATIONS

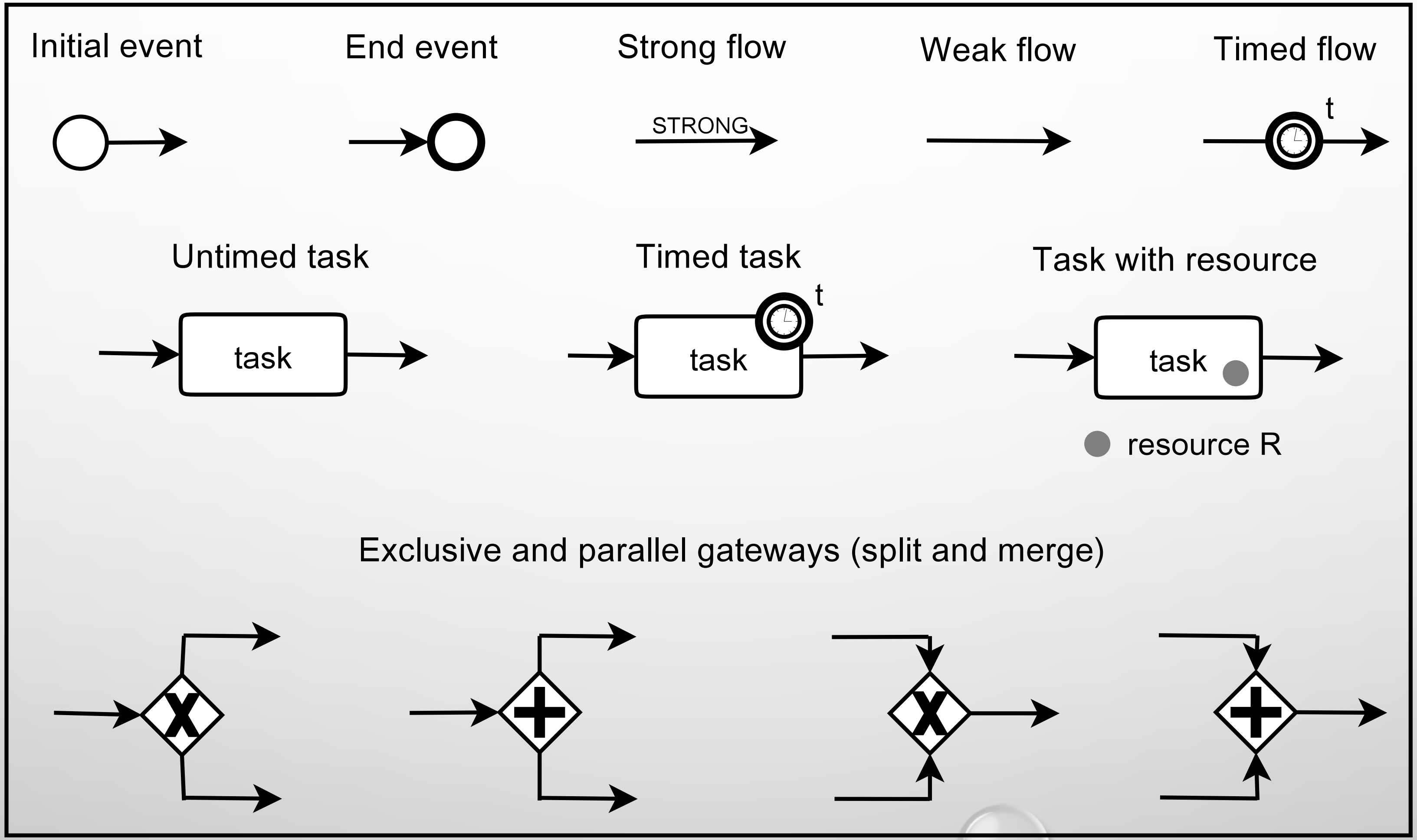
- ADJUSTING THE NUMBER OF RESOURCES DOES NOT ALWAYS INDUCE PROCESS OPTIMIZATION



- SOLUTION: WE PROPOSE REFACTORING TECHNIQUES THAT AUTOMATICALLY CHANGE THE STRUCTURE OF THE GIVEN INPUT PROCESS WITH THE FINAL GOAL OF REDUCING THE TOTAL EXECUTION TIME OF THE PROCESS

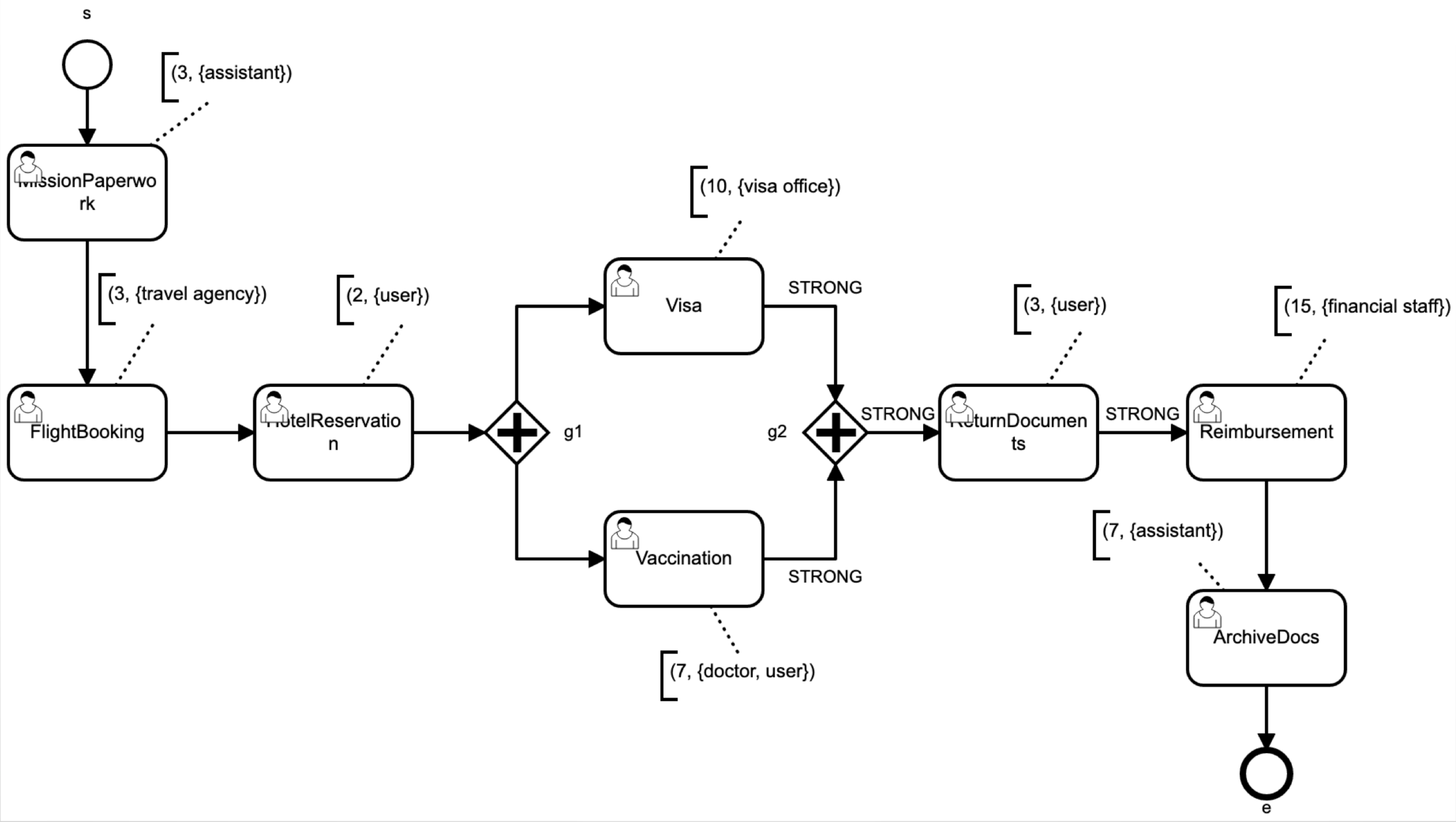
- MAIN IDEA: ADD PARALLELISM AS OFTEN AS POSSIBLE

BPMN SYNTAX

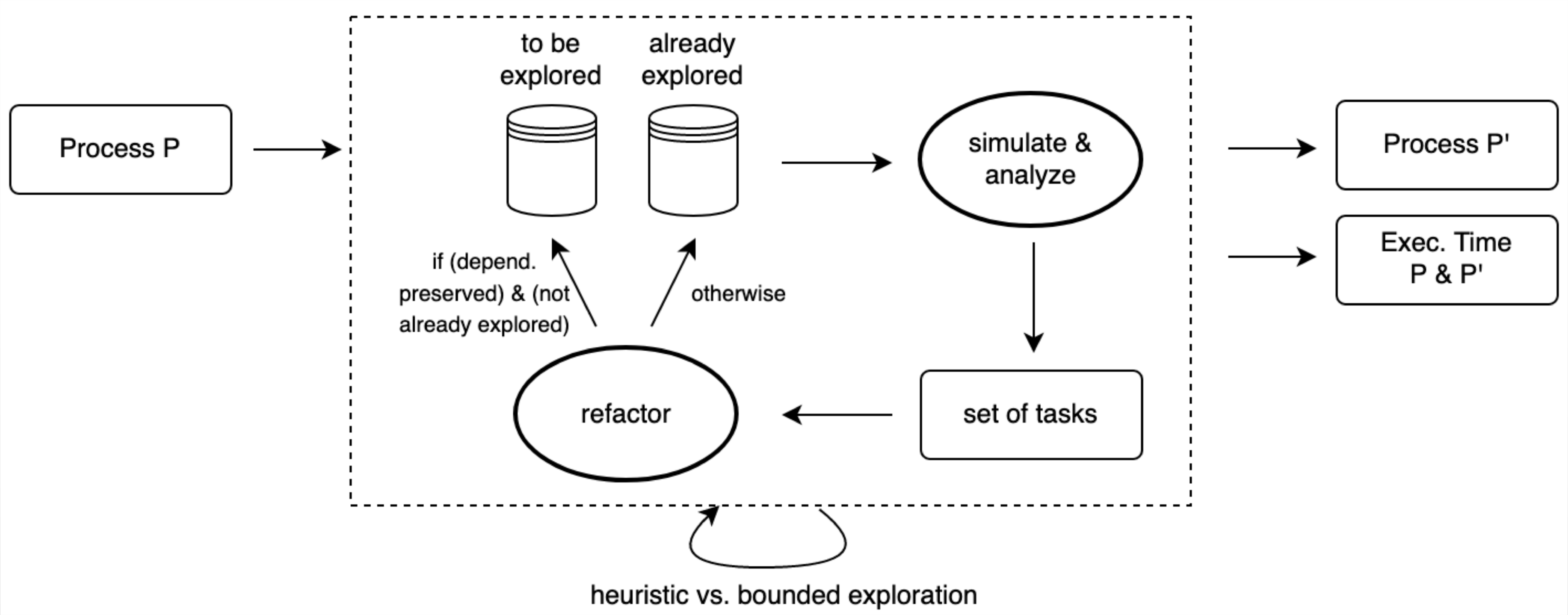


Initial event
End event
Strong flow
STRONG
Weak flow
Timed flow
t
Untimed task
task
Timed task
task
t
Task with resource
task
resource R
Exclusive and parallel gateways (split and merge)

TRIP ORGANIZATION PROCESS



OVERVIEW

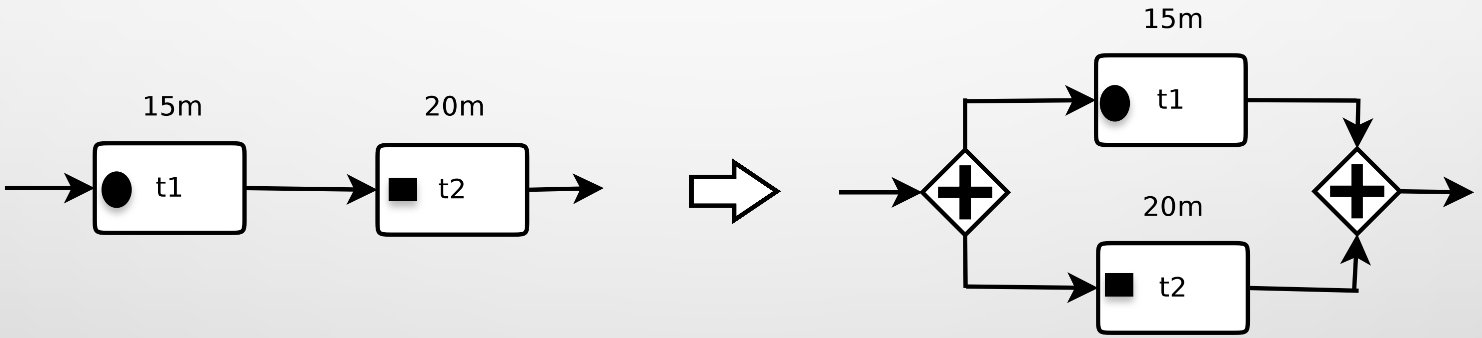


Process P
to be explored
already explored
simulate & analyze
Process P'
Exec. Time P & P'
if (depend. preserved) & (not already explored)
otherwise
refactor
set of tasks
heuristic vs. bounded exploration

SIMULATION AND ANALYSIS

- SIMULATION TECHNIQUES EXECUTE THE PROCESS A CERTAIN NUMBER OF TIMES (PARAMETER) AND LOG DATA ABOUT THIS MULTIPLE EXECUTION

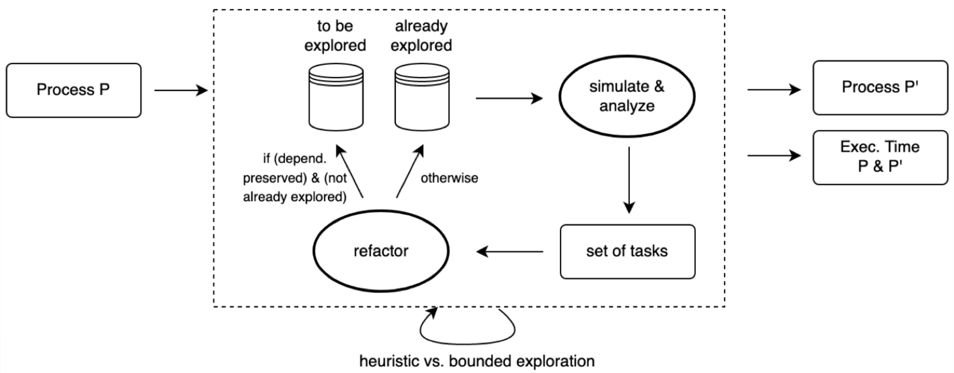
- ANALYSIS LOOKS FOR SPECIFIC MOMENTS DURING THE SIMULATION AT WHICH A TASK IS STILL WAITING TO EXECUTE, AND ALL RESOURCES REQUIRED FOR EXECUTING THIS TASK ARE AVAILABLE



- THIS MEANS THAT THIS SPECIFIC TASK COULD EXECUTE EARLIER IN THE PROCESS, AND THIS INFORMATION IS USED TO CHANGE THE STRUCTURE OF THE PROCESS

EXPLORATION OF RESULTS

- THE WHOLE APPROACH APPLIES BY SUCCESSIVE ITERATIONS, AND STOPS WHEN THE QUEUE OF PROCESSES TO BE EXPLORED IS EMPTY

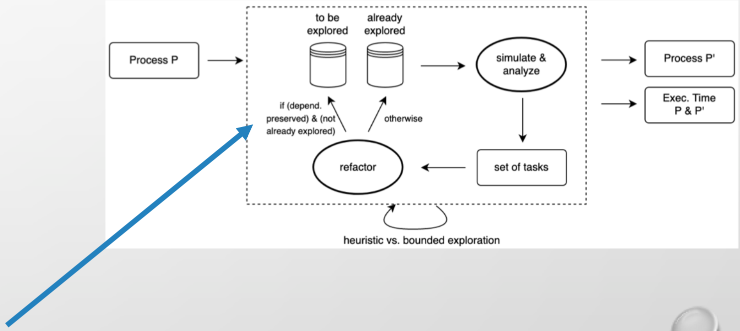


- THIS APPROACH MAY TAKE TIME, BECAUSE THERE MAY BE MANY PROCESSES TO BE EXPLORED, THEREFORE, SOME STRATEGY IS REQUIRED TO GUIDE THE EXPLORATION OF NEW PROCESSES
  - THE *FIRST STRATEGY* APPLIES REFACTORING FOR THE TASK CLOSER TO THE INITIAL EVENT IN THE PROCESS BY TRYING TO MOVE FIRST THE TASK CLOSER TO ITS FINAL POSITION
  - THE *SECOND STRATEGY* IS AN EXHAUSTIVE EXPLORATION OF ALL PROCESSES TO BE EXPLORED UP TO A CERTAIN BOUND

REFACTORING PATTERNS

- GIVEN AS INPUT A PROCESS AND A TASK THAT HAS TO BE MOVED EARLIER IN THE PROCESS, REFACTORING RETURNS A NEW PROCESS AS OUTPUT

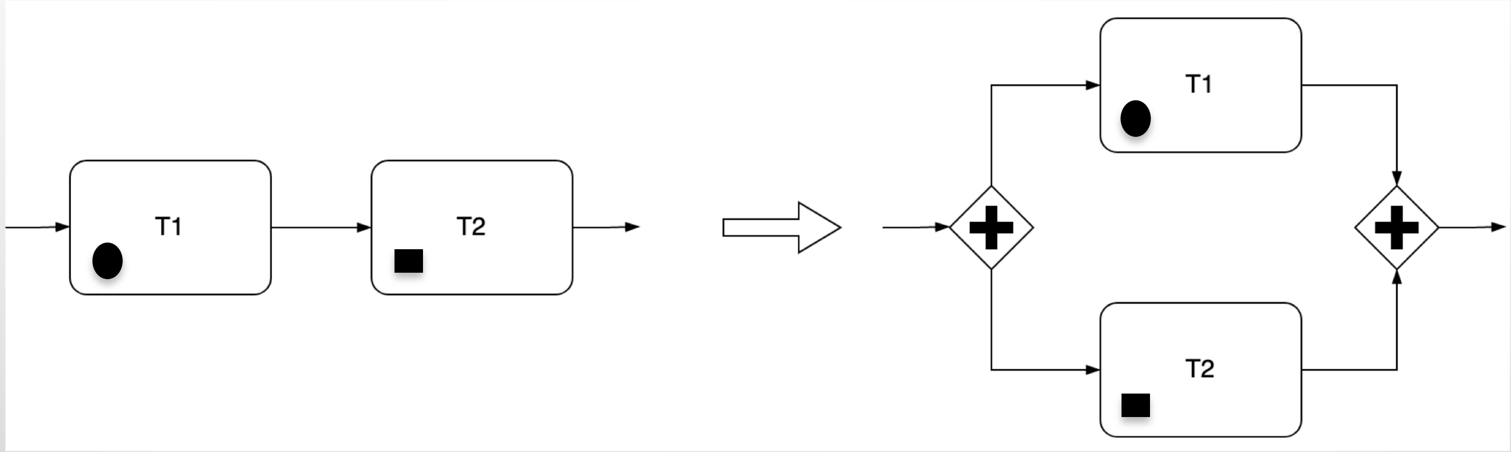
- THE REFACTORING PATTERN TO BE APPLIED DEPENDS ON WHAT TYPE OF NODE PRECEDES THE TASK TO BE MOVED BACKWARDS, WHICH CAN BE:
  - ANOTHER TASK
  - A MERGE GATEWAY (PARALLEL / EXCLUSIVE)
  - A SPLIT GATEWAY (PARALLEL / EXCLUSIVE)



- THIS REFACTORING STEP FOCUSES ON THE PROCESS STRUCTURE AND ON THE USAGE OF RESOURCES BY TASKS, BUT DOES NOT TAKE INTO ACCOUNT STRONG FLOWS

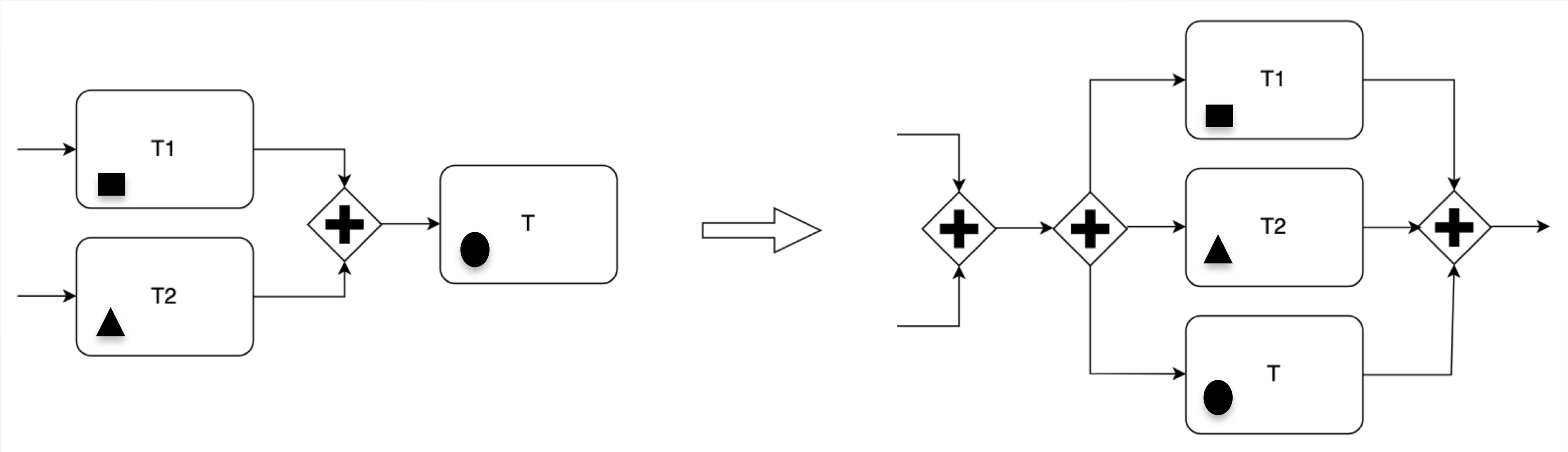
REFACTORING PATTERNS: TASK

IF THE TASK IS PRECEDED BY ANOTHER TASK, AND IF THEY DO NOT SHARE ANY RESOURCES, WE TRANSFORM THE PROCESS TO EXECUTE THESE TWO TASKS WITHIN A COMMON PARALLEL GATEWAY



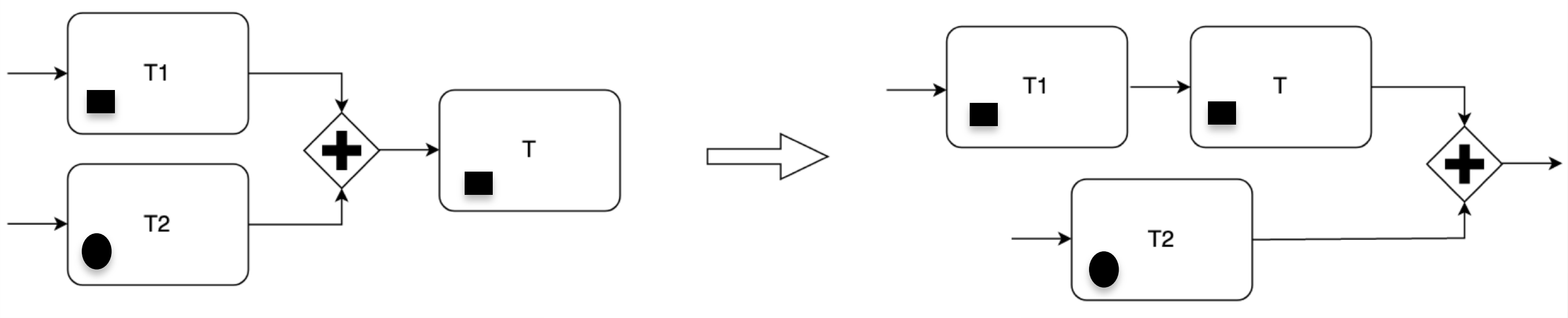
REF. PATTERNS: MERGE PARALLEL GATEWAY (1/2)

IF THE TASK 'T' IS PRECEDED BY A MERGE PARALLEL GATEWAY, IF THAT MERGE IS PRECEDED BY A SET OF TASKS, AND NONE OF THESE TASKS SHARE RESOURCES WITH 'T', THEN ALL TASKS ARE GATHERED IN PARALLEL BEFORE THE MERGE GATEWAY

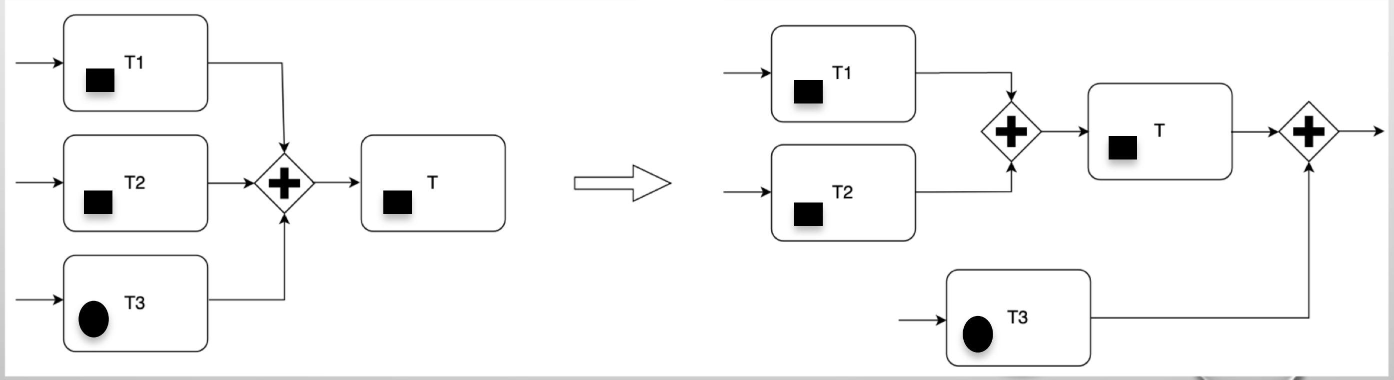


REF. PATTERNS: MERGE PARALLEL GATEWAY (2/2)

- MERGE PARALLEL GATEWAY WITH PRECEDING TASKS (SHARED RESOURCES WITH ONE TASK)

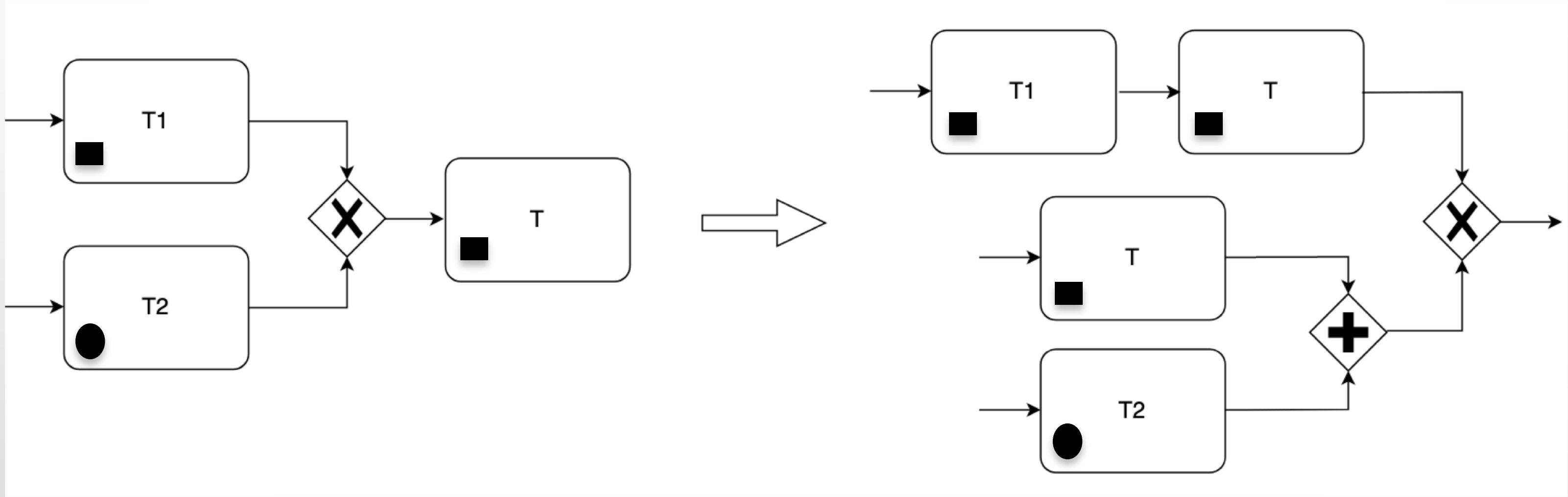


- MERGE PARALLEL GATEWAY WITH PRECEDING TASKS (SHARED RESOURCES WITH SEVERAL TASKS)



REF. PATTERNS: MERGE EXCLUSIVE GATEWAY

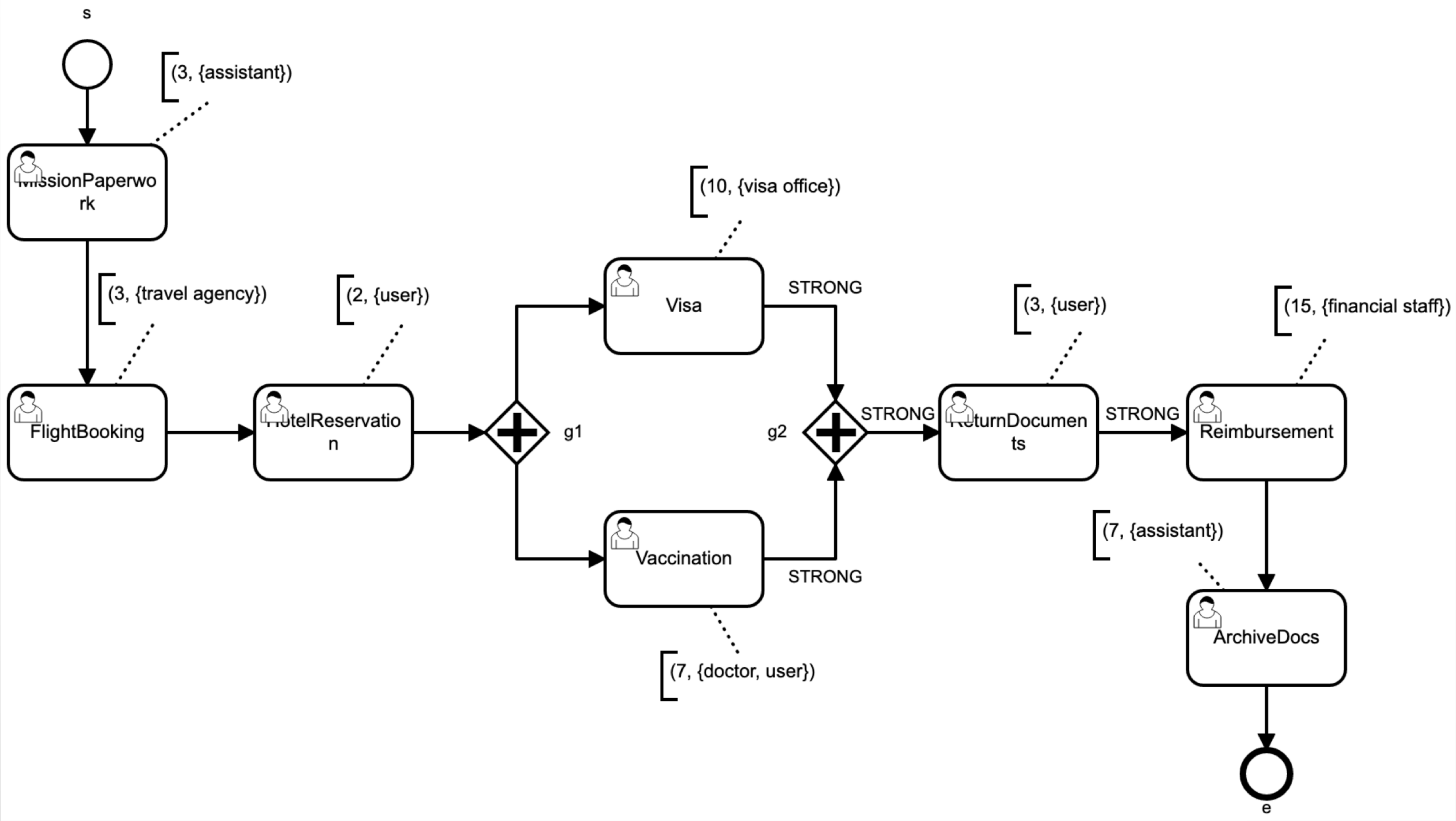
- MERGE EXCLUSIVE GATEWAY WITH PRECEDING TASKS (T1 SHARES RESOURCES WITH T, T2 DOES NOT SHARE RESOURCES WITH T)



- MORE PATTERNS: CASCADING MERGES, SPLIT PARALLEL GATEWAY, SPLIT EXCLUSIVE GATEWAY, ...

EXAMPLE: INITIAL PROCESS

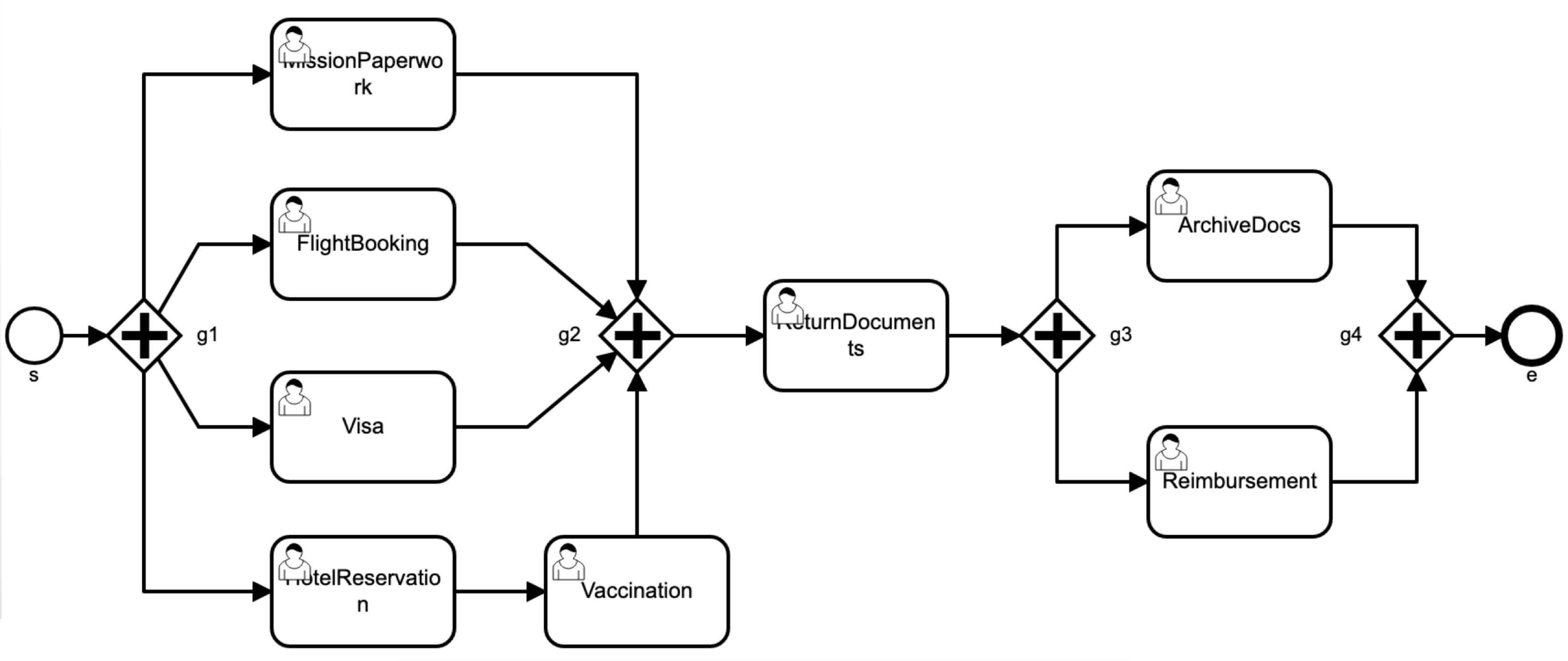
TRIP ORGANIZATION PROCESS

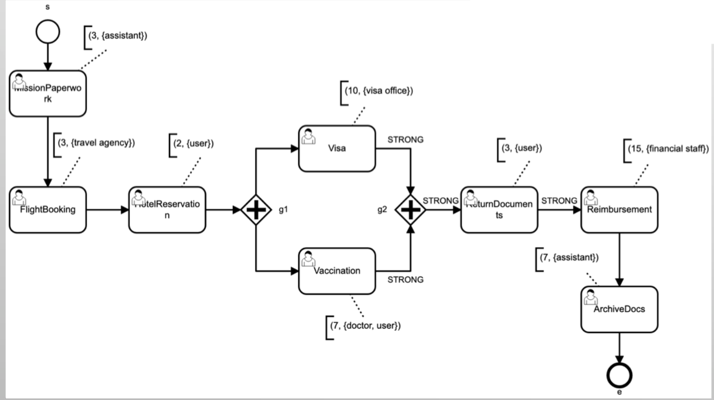


EXECUTION TIME: 43 DAYS

EXAMPLE: REFACTORED PROCESS

TRIP ORGANIZATION PROCESS





EXECUTION TIME: 28 DAYS

EXPERIMENTS

BPMN Proc.	Characteristics						Bounded Explo.		Heuristic	
	Tasks	Flows	Gateways	SF	AET_c	AET_b	AET	Time	AET	Time
1	5	13	4 [+]	0	70	50	50	1s	50	1s
2	5	15	6 [×]	0	40	25	25	2s	25	2s
3	6	13	2 [+] 2 [×]	0	19	6	6	2s	9	1s
4	8	12	2 [+]	4	43	28	28	21s	35	1s
5	8	12	2 [+]	2	21	14	14	23s	16	1s
6	9	10	0	6	90	50	50	31s	90	1s
7	10	20	6 [+]	0	24	18	19	38s	20	1s
8	10	20	2 [+] 4 [×]	3	23	13	13	67s	17	1s
9	12	29	8 [+]	0	200	120	140	50s	120	1s
10	15	36	10 [+]	0	260	180	220	103s	260	3s

OUTLINE

1. BPMN WITH TIME AND RESOURCES
2. DESIGN TIME ANALYSIS
3. RUNTIME ANALYSIS
4. AUTOMATED REFACTORING
5. TOOL SUPPORT
6. CONCLUDING REMARKS

TOOL SUPPORT

- DESIGN TIME ANALYSIS
  - ORIGINAL VERSION BASED ON THE MAUDE FRAMEWORK (~4,000 LOC)
  - RECENT VERSION WRITTEN IN JAVA, SPRING BOOT, POSTGRESQL, JAVASCRIPT, REACTJS, NODE.JS (~10,000 LOC)
- RUNTIME ANALYSIS: EXTENSION OF THE ACTIVITI FRAMEWORK + JAVA CODE (~4,000 LOC)
- REFACTORING TOOL IMPLEMENTED IN PYTHON (~5,000 LOC) + USE OF VBPMN FOR TRANSFORMATION FROM PYTHON TO BPMN

OUTLINE

1. BPMN WITH TIME AND RESOURCES
2. DESIGN TIME ANALYSIS
3. RUNTIME ANALYSIS
4. AUTOMATED REFACTORING
5. TOOL SUPPORT
6. CONCLUDING REMARKS

CONCLUDING REMARKS

- SEVERAL SOLUTIONS HAVE BEEN PRESENTED FOR THE OPTIMIZATION OF BPMN PROCESSES WITH TIME AND RESOURCES

- THE FIRST ONE FOCUSES ON THE PROBLEM OF ALLOCATION OF RESOURCES AND EXPLORED TWO COMPLEMENTARY OPTIONS (DESIGN TIME VS RUNTIME)
  - DESIGN TIME APPROACH IS USEFUL BEFORE DEPLOYMENT BUT STATIC ALLOCATION OF RESOURCES IS NOT OPTIMAL FOR UNSTABLE WORKLOADS
  - RUNTIME APPROACH DYNAMICALLY UPDATES THE NUMBER OF RESOURCES BUT THIS IS NOT ALWAYS POSSIBLE (EMPLOYEES)

- REFACTORING TECHNIQUES CHANGE A PROCESS INTO A NEW ONE, WHOSE EXECUTION TIME IS LOWER THAN THE ORIGINAL ONE

PERSPECTIVES

- DESIGN TIME / RUNTIME APPROACHES: THE MODEL OF RESOURCES NEEDS TO BE MORE ACCURATE (E.G., AN EMPLOYEE NEEDS BREAKS AND HOLIDAYS, A RESOURCE CAN BE EMPTY, ETC.)
- RUNTIME APPROACH: DEVELOP NEW DYNAMIC RESOURCE ALLOCATION STRATEGIES BASED ON AI PREDICTION ANALYTICS
- REFACTORING TECHNIQUES: INVESTIGATE OTHER STRATEGIES TO COMPUTE FASTER THE OPTIMAL PROCESS
- SYNTHESIS TECHNIQUES TO GENERATE AN OPTIMAL VERSION OF A PROCESS

SELECTED PUBLICATIONS

Francisco Durán, Yliès Falcone, Camilo Rocha, Gwen Salaün, Ahang Zuo: *From Static to Dynamic Analysis and Allocation of Resources for BPMN Processes*. WRLA 2022.

Francisco Durán, Camilo Rocha, Gwen Salaün: *Resource provisioning strategies for BPMN processes: Specification and analysis using Maude*. J. Log. Algebraic Methods Program. 123, 2021.

Francisco Durán, Camilo Rocha, Gwen Salaün: *A rewriting logic approach to resource allocation analysis in business process models*. Sci. Comput. Program. 183, 2019.