

Methods for the specification and verification of business processes

MPB (6 cfu, 295AA)

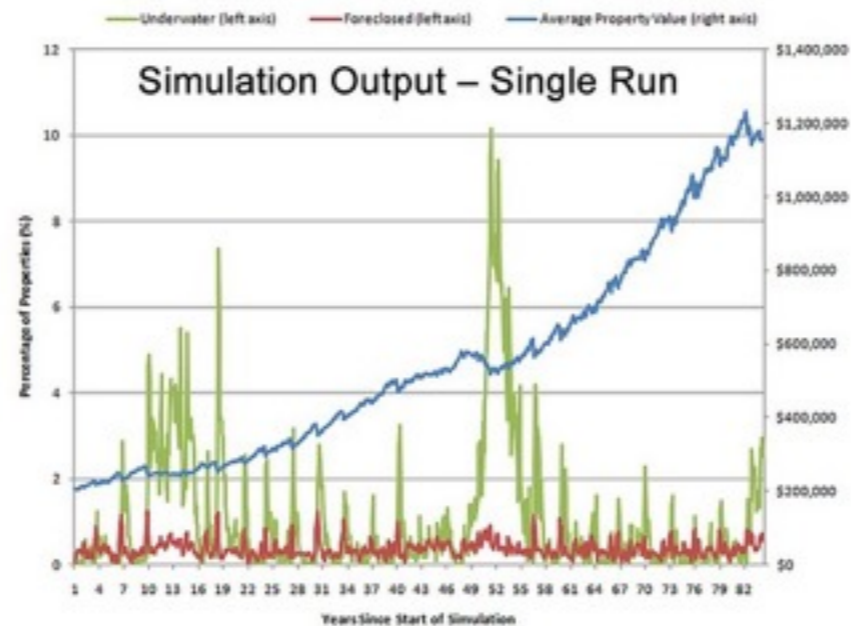
Roberto Bruni

<http://www.di.unipi.it/~bruni>

25 - Simulation



Object



We overview some principles for the quantitative simulation of business processes

Ch.7 of Fundamental of Business Process Management. M. Dumas et al.
(inspired by slides available at <https://courses.cs.ut.ee/2014/bpm/>)

Simulation

Process simulation is the most popular and widely supported technique for quantitative analysis of process models.

It is a very flexible analysis technique.

It is applicable to almost any workflow.

It is accessible to people without mathematical background.

It boils down to **computer aided repeated execution** of paths in the reachability graph:

A large number of hypothetical instances of the process are generated and executed step-by-step;

the produced output can include logs as well as statistics about (average) cycle/waiting times and resource utilization

Resource allocation

Resource management

In a process we can indicate:
which tasks need to be performed,
the order in which they must be carried out,
who should do it

The way in which work items are allocated to resources (people, machines) is very important to the efficiency and effectiveness of the workflow

Resources

The basic characteristic of a resource is that it is able to carry out particular tasks

We assume each resource is uniquely identifiable and has capacity one, i.e., **each resource can work on no more than one activity at any given time**

Resource classification

A resource is permitted to carry out a number of tasks (e.g., in a bank, a teller is not allowed to grant a mortgage)

A task can be performed only by certain resources

Which resources are able to carry out which task?

It is impracticable to indicate them one by one in the process
(staff can change)

We classify them using **resource classes**

- 1) classification based on functional properties (**roles**)
- 2) classification based on the position in the organization (**groups or organizational units**)

Examples

Roles (skill, competence, qualification):
counter-staff, travel-agent, assessor, printer, administrator,
chief-executive, senior-doctor

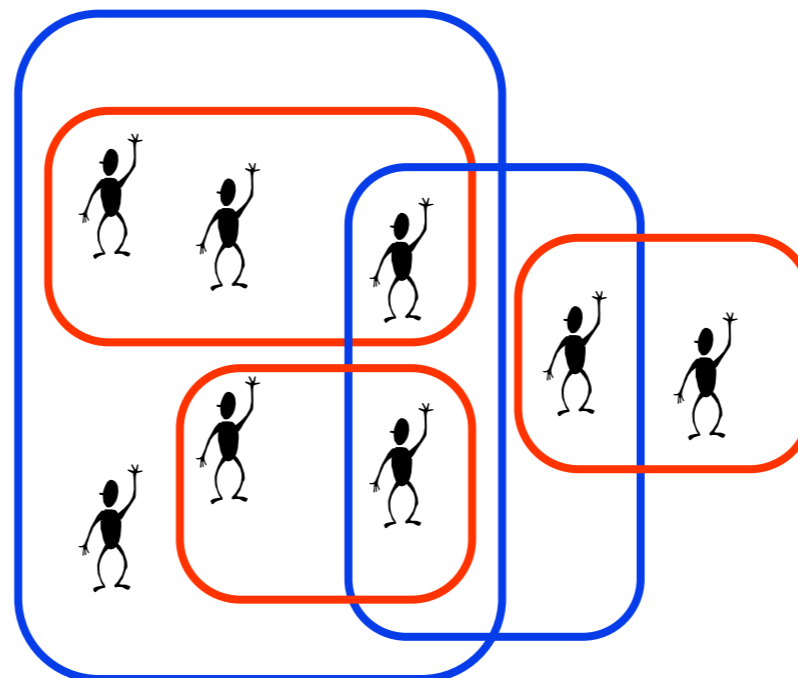
Groups (department, office, team):
sales-department, purchasing-department,
development-team, Avana-branch

**The same resource can belong to
more roles and groups**

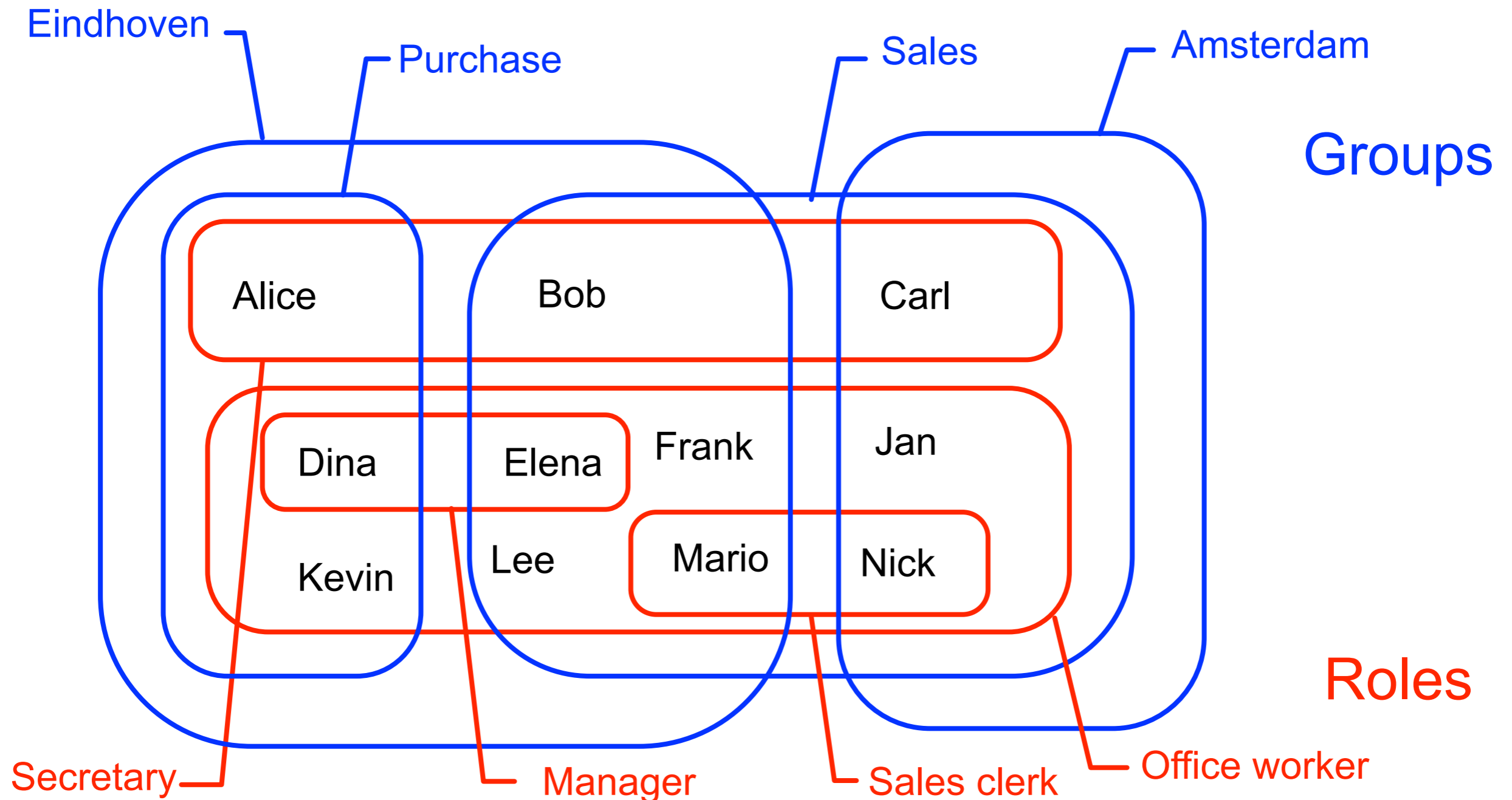
Classification diagram

A **classification diagram** shows which resources belong to each class and group

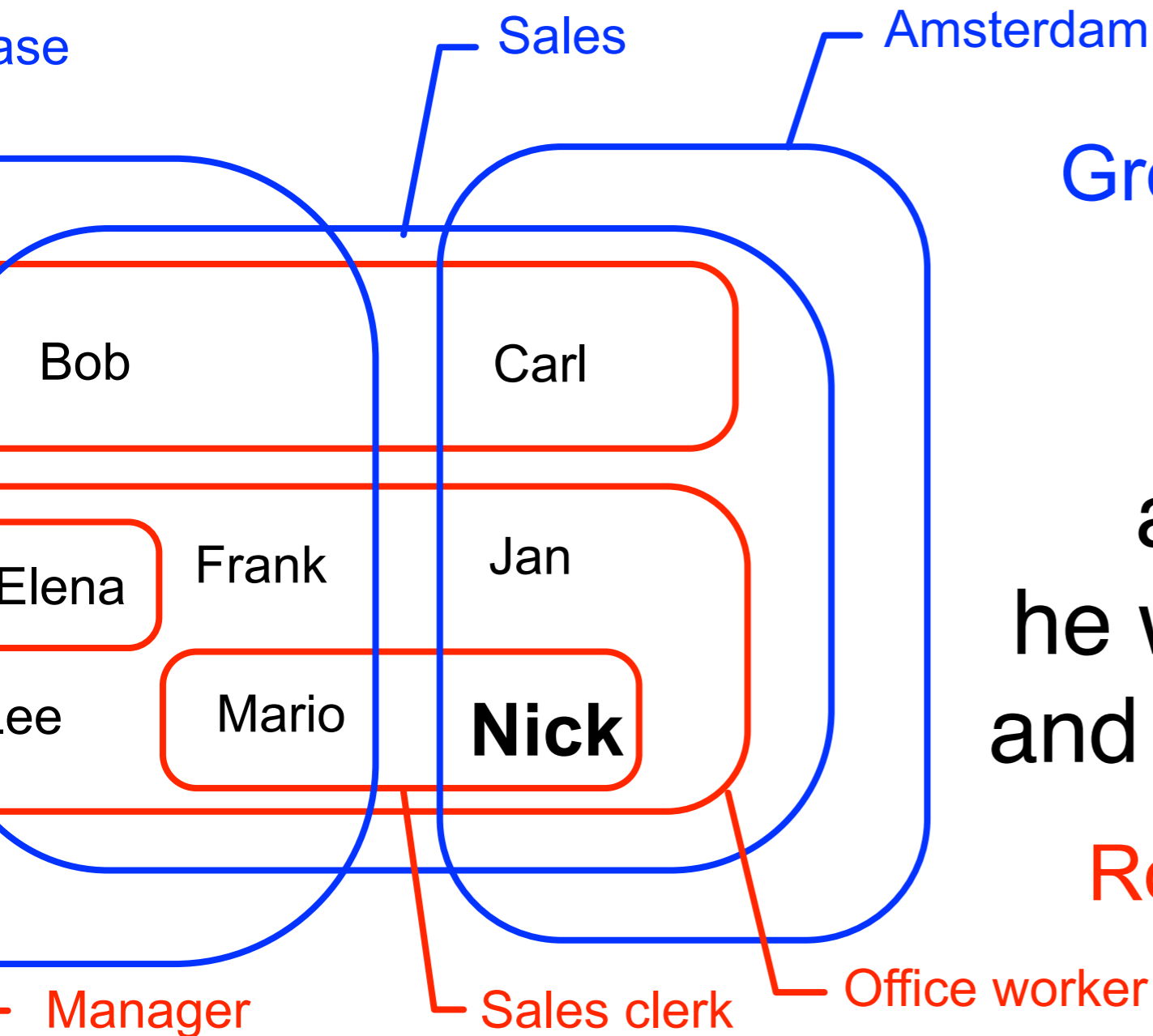
We can list resource ids and enclose them in vertical **blue boxes** that represent **groups** and in horizontal **red boxes** that represent **roles**



Classification diagram: example



Classification diagram: example



Groups

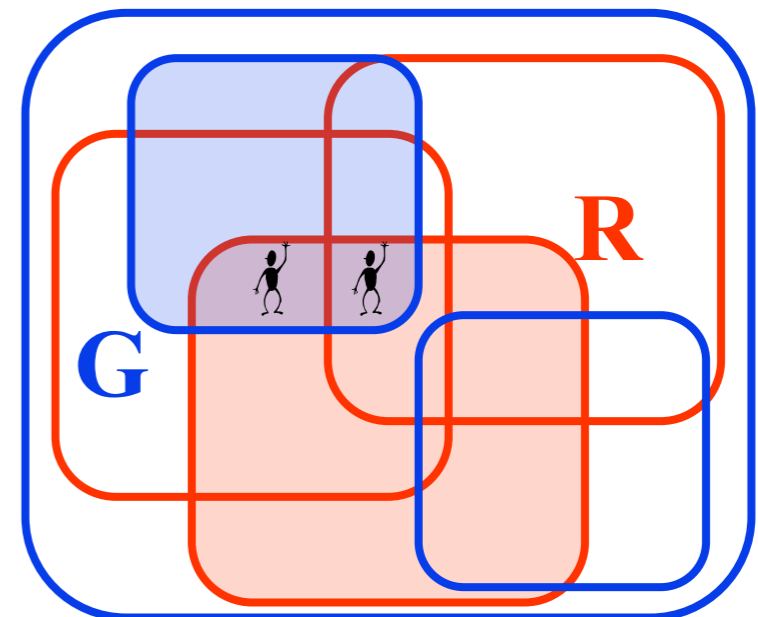
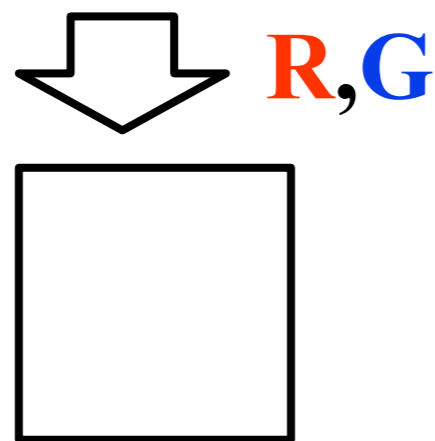
Nick is
a **Sales clerk**
and an **Office worker**
he works in the **Sales** group
and in the **Amsterdam** group

Roles

Resource management rules

Resource management rules:
tell how to map work onto resources

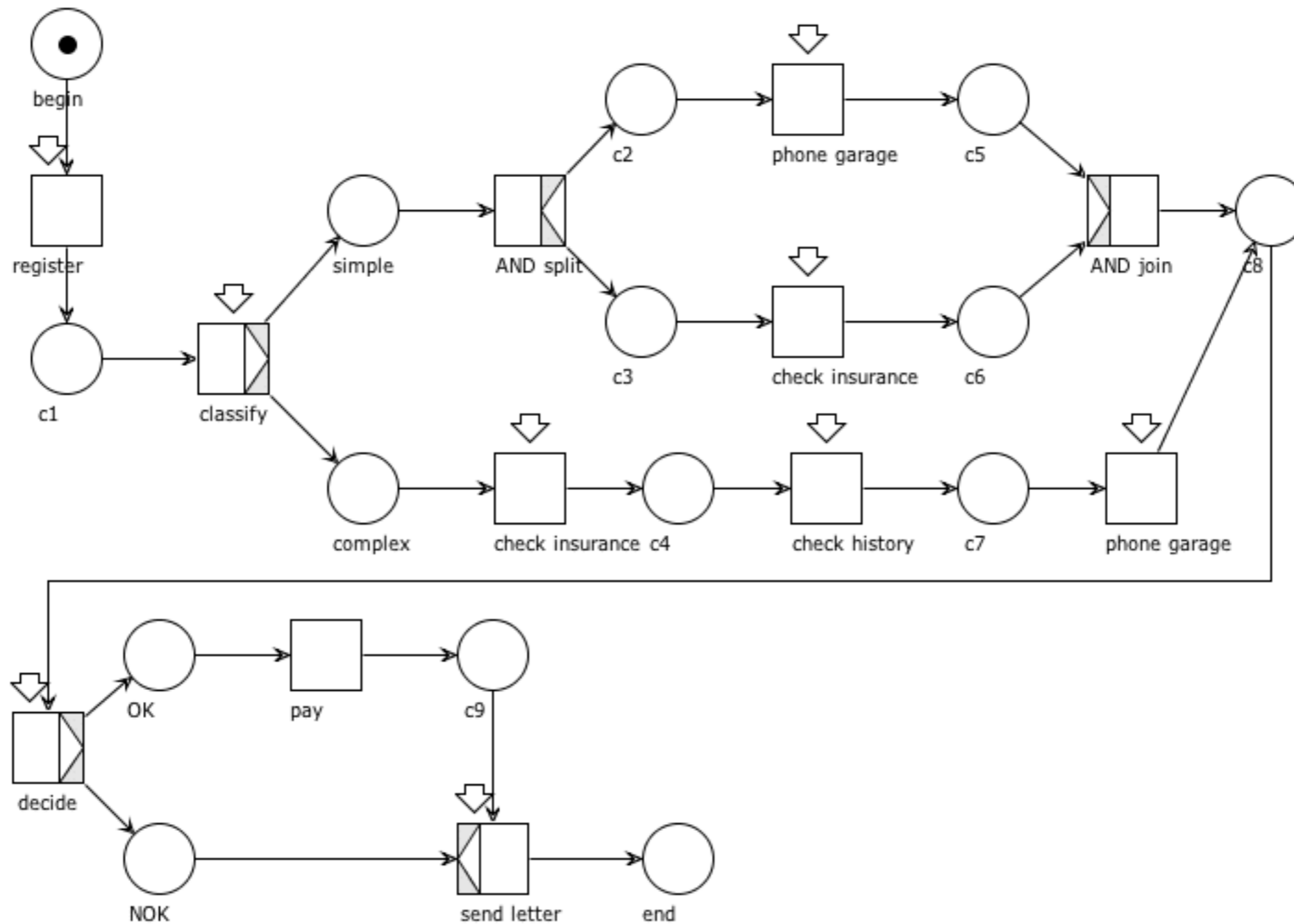
Each task executed by a resource is labelled
with one **role** and one **group**
(allowed resources must lie in the intersection)



Example: Car Damage

- An insurance company uses the following procedure for the processing of the claims
- Every claim, reported by a customer, is **registered**
- After the registration, the claim is **classified**
- There are two categories: **simple** and **complex** claims.
 - For simple claims two tasks need to be executed: **check insurance** and **phone garage**.
These tasks are *independent* of each other.
 - The complex claims require three tasks: **check insurance**, **check damage history** and **phone garage**.
These tasks need to be *executed sequentially* in the order specified.
- After executing the two/three tasks a **decision** is taken with two possible outcomes: **OK** (positive) or **NOK** (negative).
- If the decision is positive, then insurance company will **pay**.
- In any event, the insurance company **sends a letter** to the customer.

Example: Car Damage



Example: Car Damage

- An insurance company uses the following procedure for the processing of the claims
- Every claim, reported by a customer, is **registered** by an employee of department CD
- After the registration, the claim is **classified** by a claim handler (CH) of rank A or B within CD.
- There are two categories: **simple** and **complex** claims.
 - For simple claims two tasks need to be executed: **check insurance** and **phone garage**.
These tasks are *independent* of each other.
 - The complex claims require three tasks: **check insurance**, **check damage history** and **phone garage**.
These tasks need to be *executed sequentially* in the order specified.
- Both for the simple and complex claims, the tasks are done by employees of department CD. After executing the two/three tasks a **decision** is made by a claim handler of rank A and has two possible outcomes: **OK** (positive) or **NOK** (negative).
- If the decision is positive, then insurance company will **pay**.
An employee of the finance department handles the payment.
- In any event, the insurance company **sends a letter** to the customer.
An employee of the department CD writes this letter.

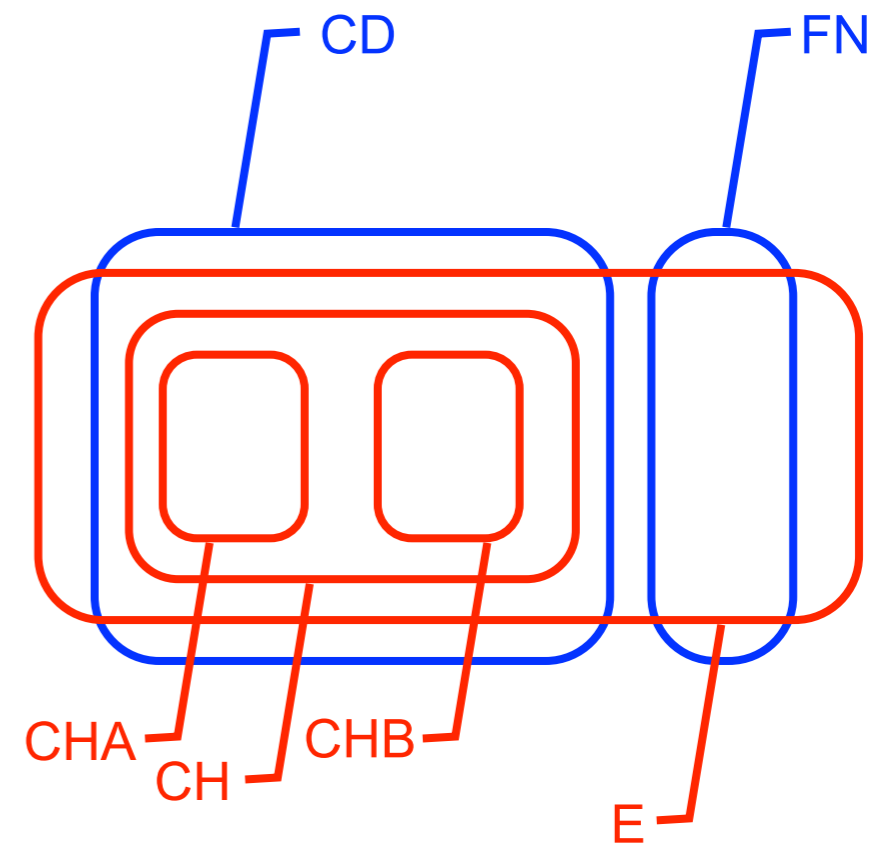
Example: Car Damage

The following roles are identified:

- *Employee* (E)
- *Claim handler* (CH)
- *Claim handler A* (CHA)
- *Claim handler B* (CHB)

The following groups are identified:

- *Car Damages Department* (CD)
- *Finance Department* (FN)



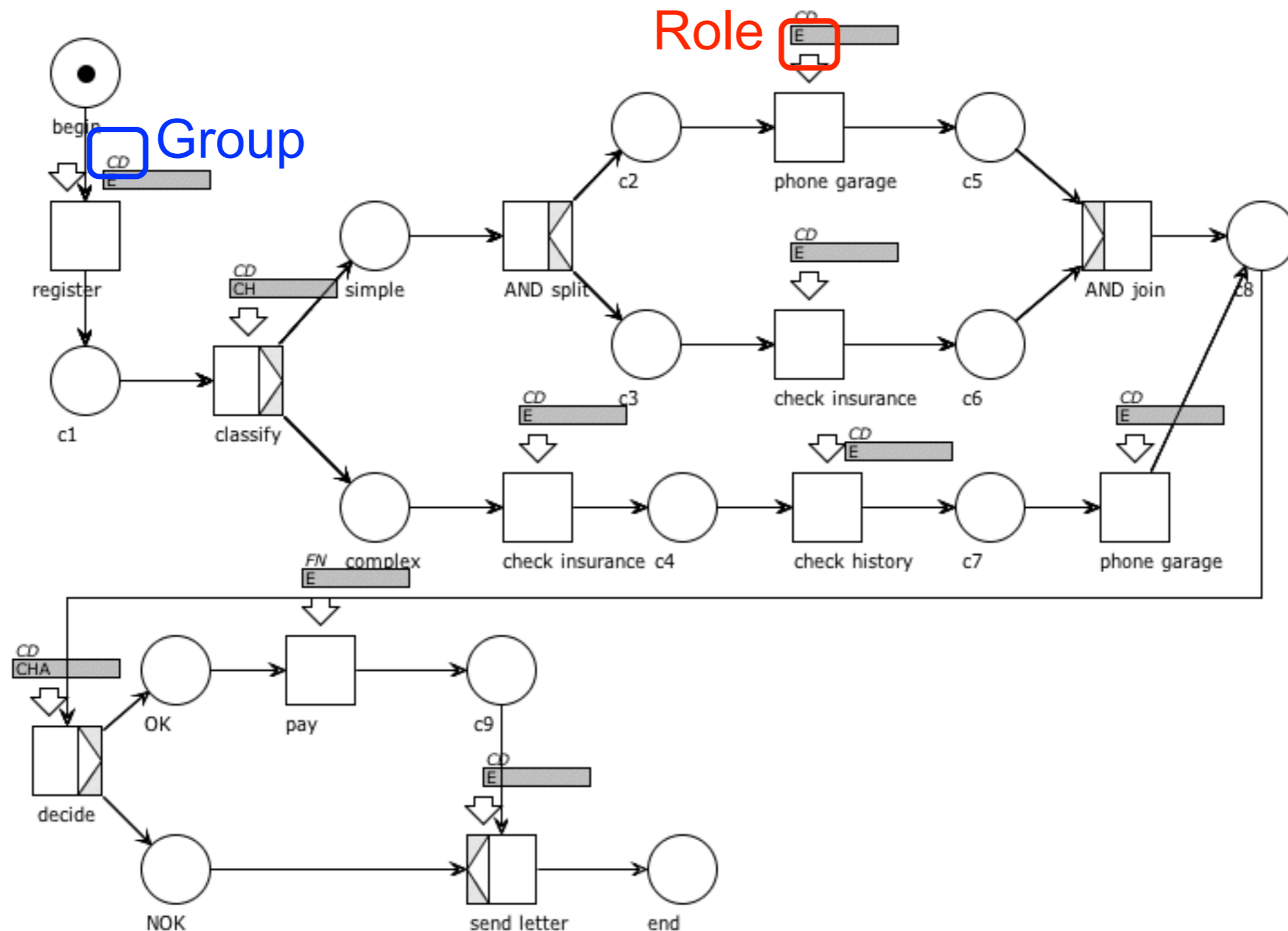
Example: Car Damage

The screenshot shows a software interface titled "Untitled - 0" with a "Resources" tab selected. The interface is divided into several panels:

- Objects:** Contains "Unassigned" and "Assigned" categories.
- Roles:** Contains roles "E", "CHA", and "CHB". This panel is highlighted with a blue box.
- Groups:** Contains groups "CD" and "FN". This panel is highlighted with a blue box.
- Compound roles:** Contains a compound role "CH" which is expanded to show sub-roles "CHA" and "CHB". This panel is highlighted with a blue box.
- Compound groups:** Currently empty.

A mouse cursor with a green arrow is positioned over the central workspace area.

Example: Car Damage



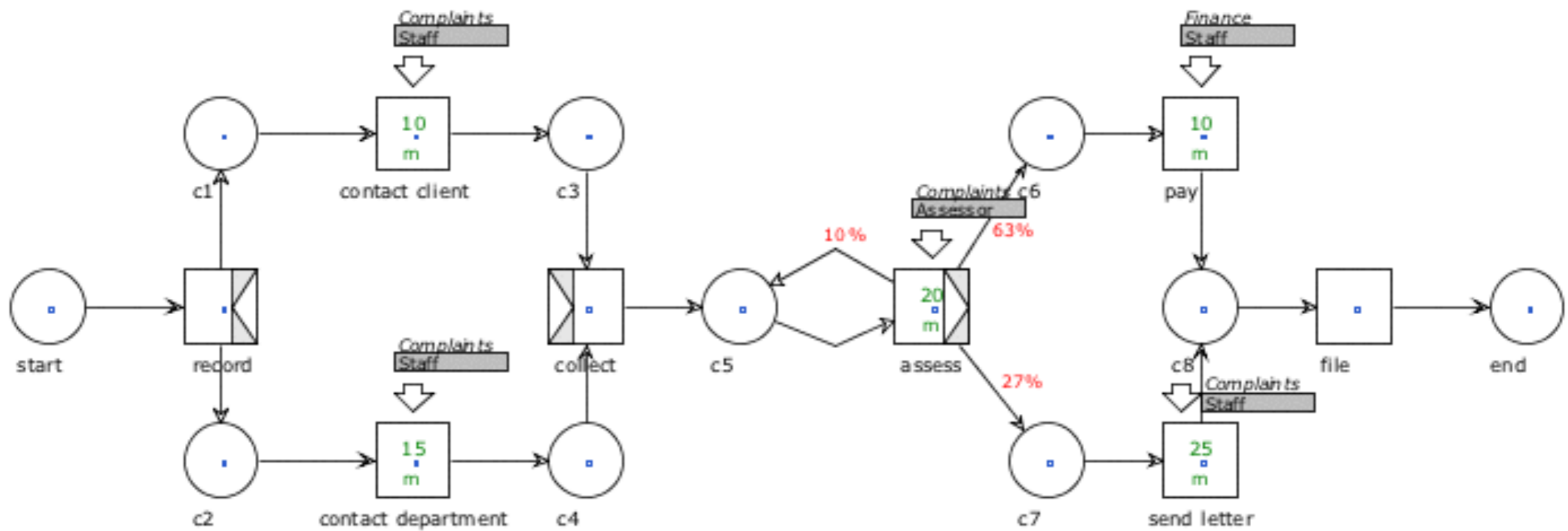
Capacity plan

The staff may vary according to seasonal factors
(vacation, illness, leaving the company)

The capacity plan shows what resources are needed for a
period of time

If we have a forecast of the supply of new cases,
then the capacity requirement can be estimated

Example



Example

The screenshot displays a software window titled "wfnet-capacity-planning.pnml" with three tabs: "Process", "Resources", and "BPEL preview". The "Resources" tab is active. In the top-left corner, there is a "Choose view" section with "List view" selected (indicated by a blue dot) and "Graphical view" unselected. Below this is an "Objects" panel with a toolbar containing icons for search, edit, delete, and move, and a list containing "Unassigned" and "Assigned". To the right are two panels: "Roles" and "Groups". The "Roles" panel has a toolbar with search, edit, delete, and a blue downward arrow, and lists "Assessor" and "Employee". The "Groups" panel has a toolbar with search, edit, delete, a blue downward arrow, and an upward arrow, and lists "Complaints" and "Finance". A mouse cursor with a green arrow points from the "Objects" panel towards the "Roles" panel. At the bottom, there are two more panels: "Compound roles" with a toolbar and a list containing "Staff", and "Compound groups" with a toolbar. The bottom status bar shows "Horizontal", "Zoom: 85%" with a slider, and "Not saved".

Example

The image shows a screenshot of a BPMN editor window titled "wfnet-capacity". The main canvas displays a process diagram with several activities and transitions. A transition named "assess" is highlighted with a blue box. The "Transition properties" dialog box is open, showing the configuration for this transition.

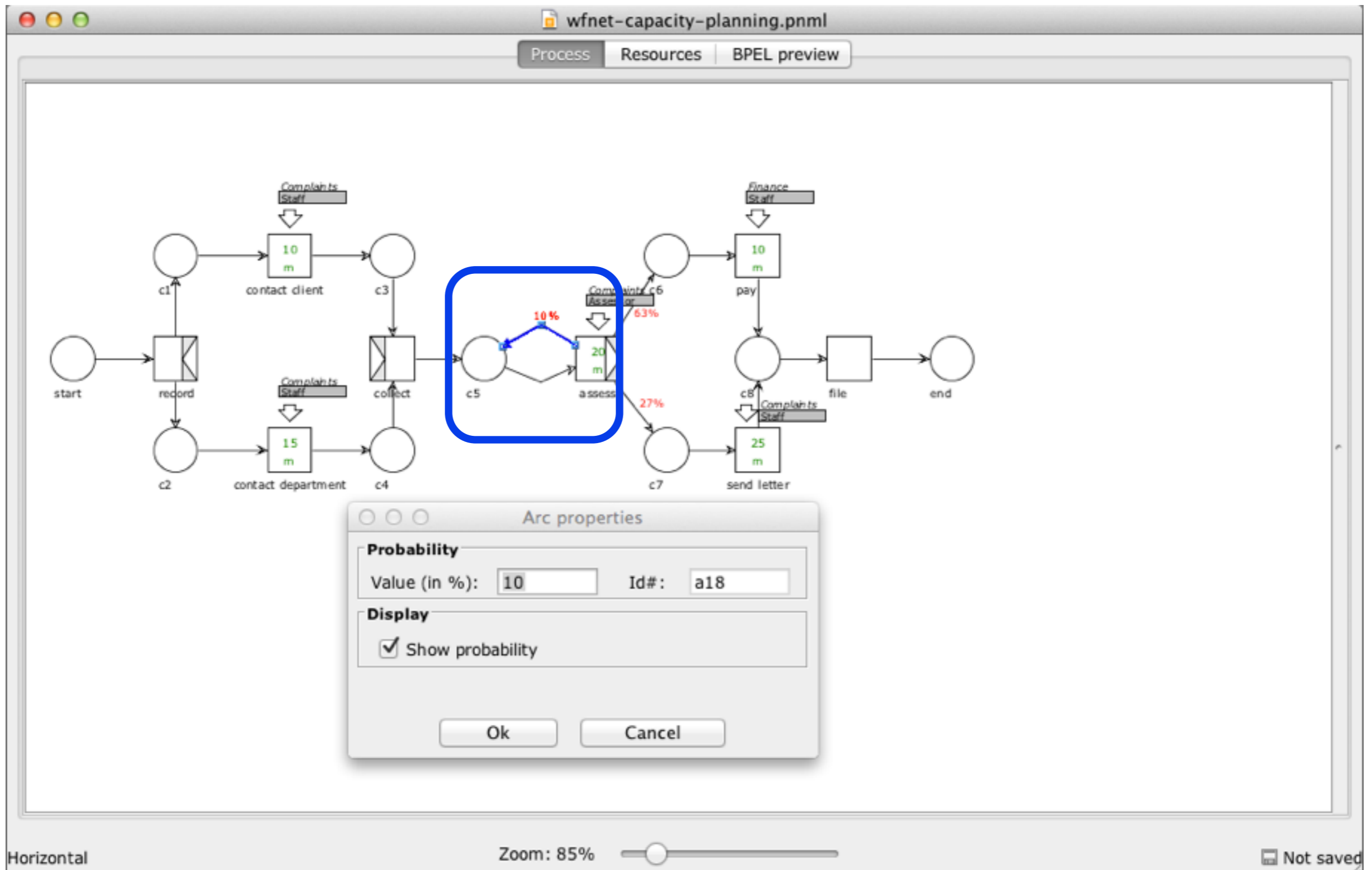
Transition properties dialog:

- Identification:** Name: Id#:
- Branching:**
 - None
 - XOR-split
 - AND-split
 - XOR-join-split
 - AND-join-XOR-split
 - AND-join
 - XOR-join
 - AND-join-split
 - XOR-join-AND-split
- Trigger:**
 - Automatic
 - Resource
 - Message
 - Time
- Orientation:**
 - North
 - West
 - East
 - South
- Service time:** Average:
- Resource mapping:** Role: Group: No. of assigned resource objects:

Buttons:

Horizontal Zoom: 85% Not saved

Example



Example

Capacity planning

Parameters

Observation period: hour(s)

Arrivals per period (λ):

Loop termination threshold (ϵ):

Capacity requirements per task

Time unit: 1.0 minute(s) Floating point precision:

Task	Service time	Items/case	Time/case	Items/period	Time/period	Group / Role
contact client (t2)	10.00	1.00	10.00	50.00	500.00	Complaints/Staff
contact departmen...	15.00	1.00	15.00	50.00	750.00	Complaints/Staff
assess (t5)	20.00	1.11	22.22	55.56	1111.10	Complaints/Assessor
pay (t6)	10.00	0.70	7.00	35.00	350.00	Finance/Staff
send letter (t7)	25.00	0.30	7.50	15.00	375.00	Complaints/Staff
Whole process			61.72		3086.09	

Capacity requirements per resource class

Average resource utilization:

Resource class	Aggregate time	Min. number of resource objects
Employee	0.00	0.00
Finance	350.00	0.91
Staff	1974.99	5.14
Complaints	2736.10	7.13
Assessor	1111.10	2.89

The unfolded net has 69 nodes. The relative deviation is (estimated): +0.0%

Process simulation

System: parameters

For each **task**:

the probability distribution for the processing time
other performance attributes (e.g., cost, value)
the set of resources able to perform the task

For each **(XOR) split**:

branching probability of every outgoing arc

For each **resource class/group**:

the number of resources in the pool,
other performance attributes (e.g., hourly cost)

Task duration: fixed

Fixed:

the processing time is relatively constant

rare when humans are involved,
common for automated tasks
(e.g., automatic report generation)

Task duration

(Negative) Exponential distribution:

applicable when the processing time is most often around a **mean value**, but sometimes considerably longer (e.g., assess insurance claim)

parameter: rate λ

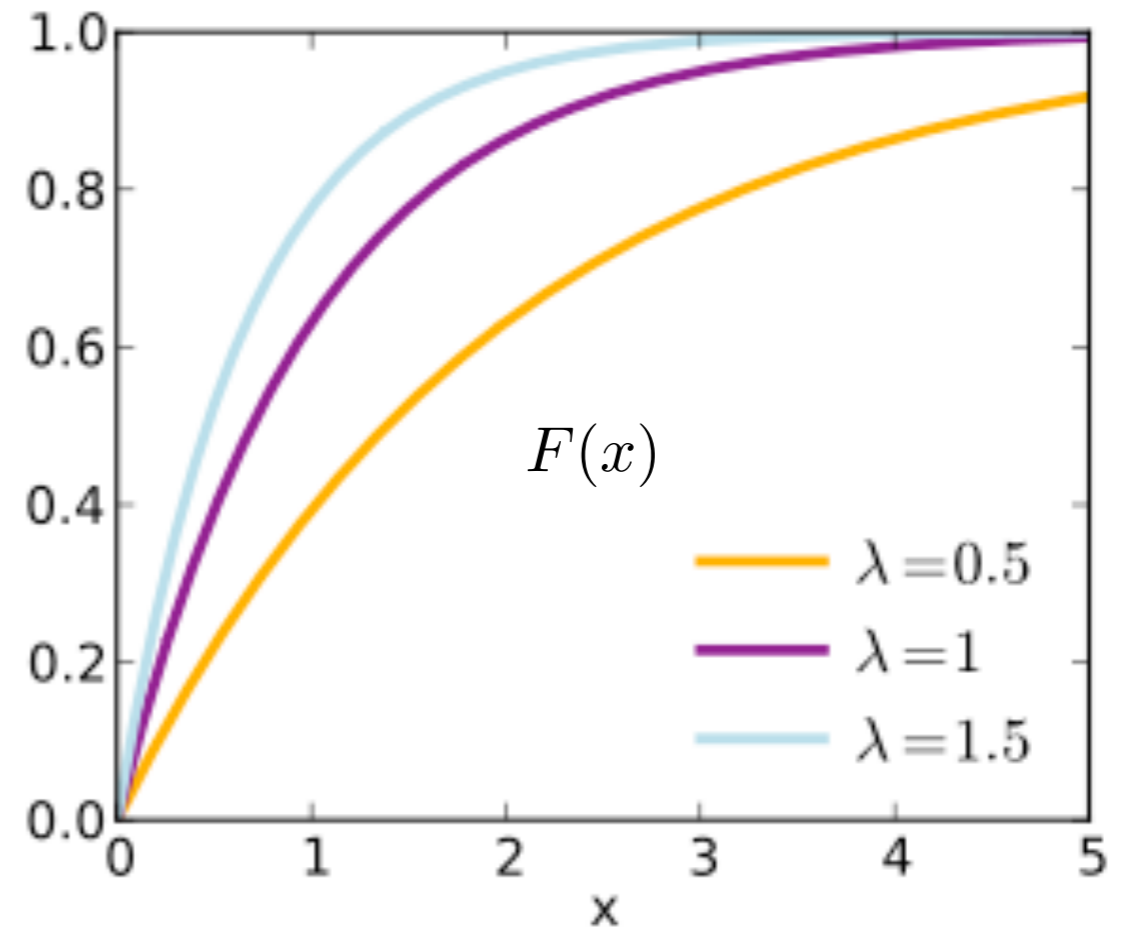
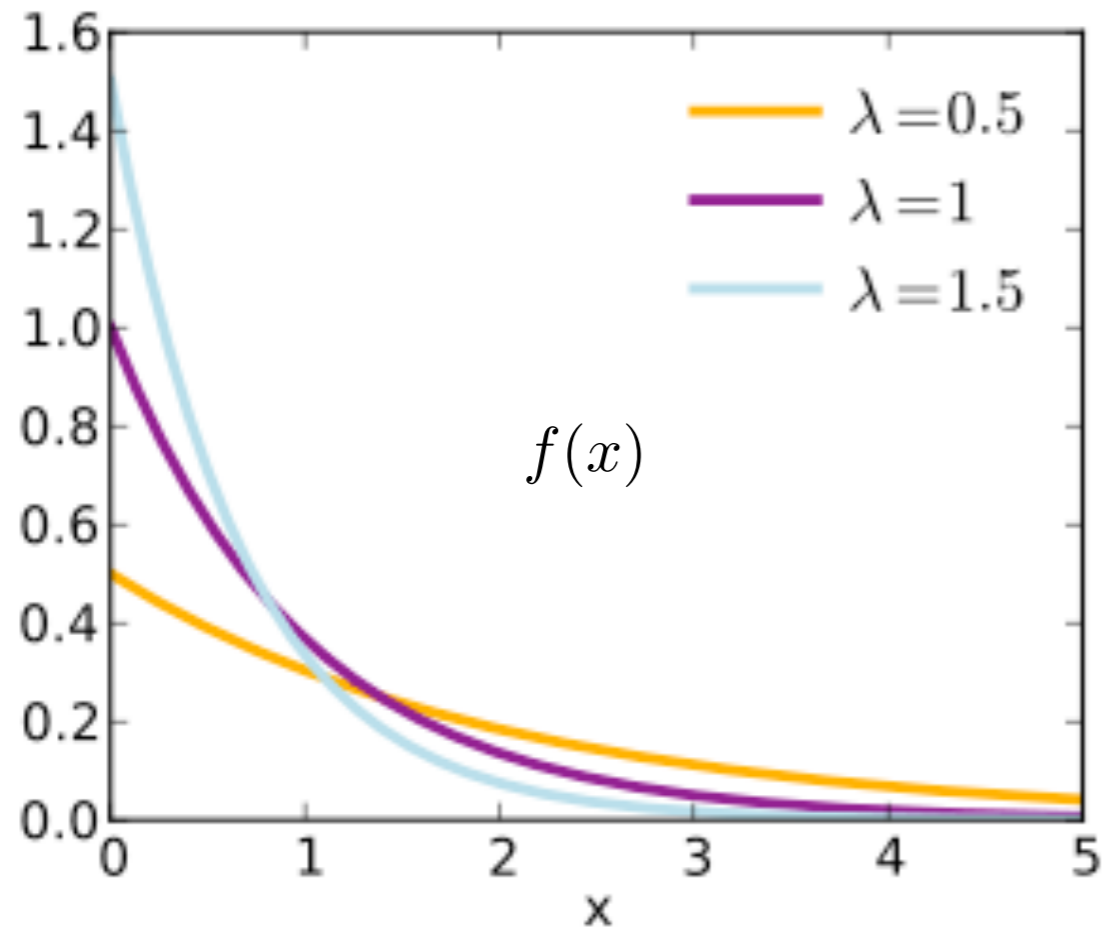
probability density function: $f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$

cumulative distribution function: $F(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$

mean value: $\frac{1}{\lambda}$

Task duration

Exponential distribution:



Task duration

Normal distribution:

applicable when the processing time is around a given **average** and the **deviation** around this value is symmetric (e.g., paper-form check)

parameters: mean value μ , standard deviation σ

probability density function: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

cumulative distribution function: $\Phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$

Task duration

Normal distribution:

applicable when the processing time is around a given **average** and the **deviation** around this value is symmetric (e.g., paper-form check)

parameters: mean value $\mu=0$, standard deviation $\sigma=1$

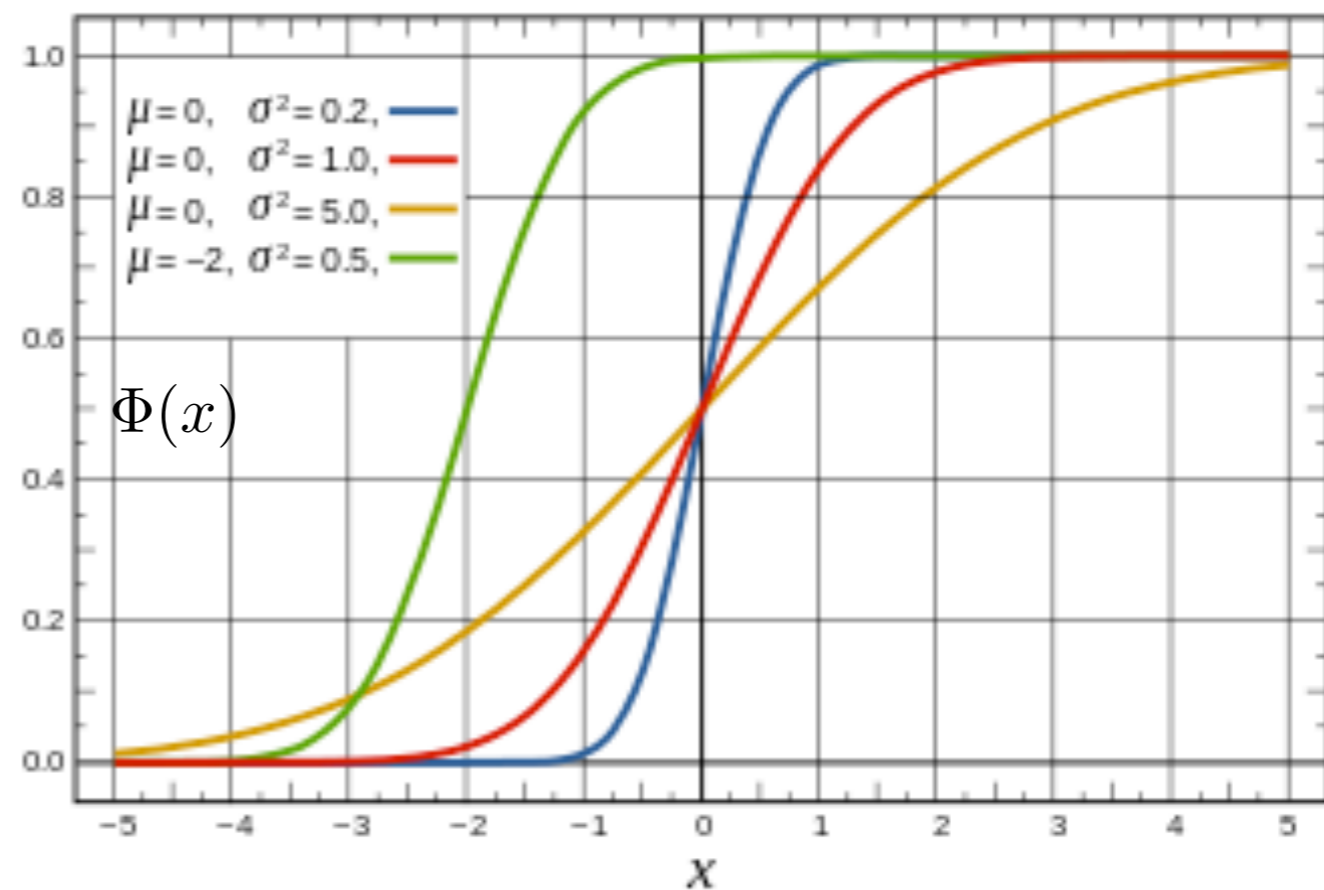
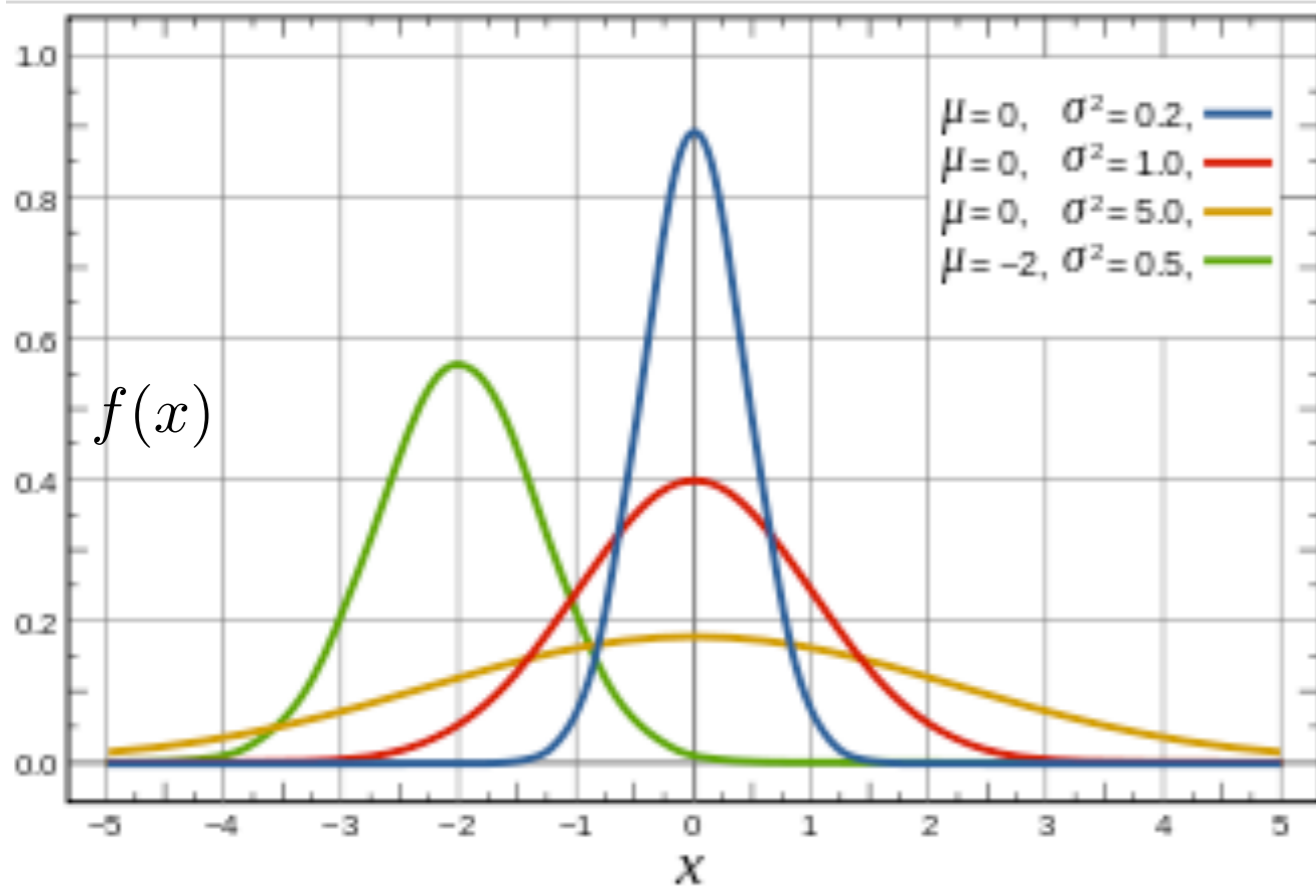
probability density function: $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

cumulative distribution function: $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt$

(standard normal distribution $\mu=0$, $\sigma=1$)

Task duration

Normal distribution:



Task duration: estimate

How to estimate the parameters?

Informed guess: interviews with stakeholders

sampling: collect sample data from real executions

logs import: allowed by some tools

(this functionality is called **simulation input analysis**)

Simulation: other inputs

Inter-arrival times and mean arrival rate
(e.g., exponential distribution, normal distribution)

Lapse of time of the simulation

Either:

end date and time of the simulation; or
real-time duration of the simulation; or
number of process instances to be simulated.

Simulation: execution

During a process simulation, the tasks in the process are not actually executed

When a task is ready to be executed a so-called work item is created and the simulator first tries to find a resource to which it can assign this work item

If no resource is available the work item is put in waiting mode until a resource is freed up

Once a resource is assigned to a work item, the simulator determines the duration of the activity by drawing a random number according to a probability distribution

Simulation: execution

Once the duration has been determined, the work item is put in sleeping mode for that duration (to simulate the execution of the task)

Once the time interval has passed (according to the simulation's clock), the work is completed and the resource that was assigned to it is available again

Simulators exploit smart algorithms to complete the simulation as fast as possible: they do not effectively wait for tasks to come back from their sleeping mode and do not wait for the whole activity duration either (thousands of process instance can be simulated in matter of seconds)

Simulation: execution

For each work item created during a simulation, the simulator records the identifier of the resource that was assigned to this instance as well as three time-stamps:

the time when the task was **ready** to be executed

the time when the it was (assigned to a resource and) **started**

the time when the task was **completed**

Thanks to the collected data, the simulator can compute the

average waiting time for each task

(important to identify bottlenecks in the process)

It can also compute the **total amount of time** during which a

resource is **busy** handling work items and its **resource utilization** (the average percentage of time that it is busy)

BIMP

<http://bimp.cs.ut.ee>

BIMP

BIMP is a free, simple online simulator of BPMN models.

1. Upload BPMN2.0 models created with BPMN-compliant tools
2. Create a simulation scenario including parameters such as:
 - the number of process instances to be simulated;
 - their arrival rates;
 - the number, types and timetables of resources;
 - branching probabilities;
 - the duration and fixed cost of each task (uniform, normal and exponential distributions for durations are supported).The (parameters of the) simulation scenario can be saved.
3. Run the simulation

BIMP: output

A dashboard is eventually displayed that includes:

Cost information: total cost of the scenario, min, average and max costs of individual process instance and a diagram of the process instance cost distribution.

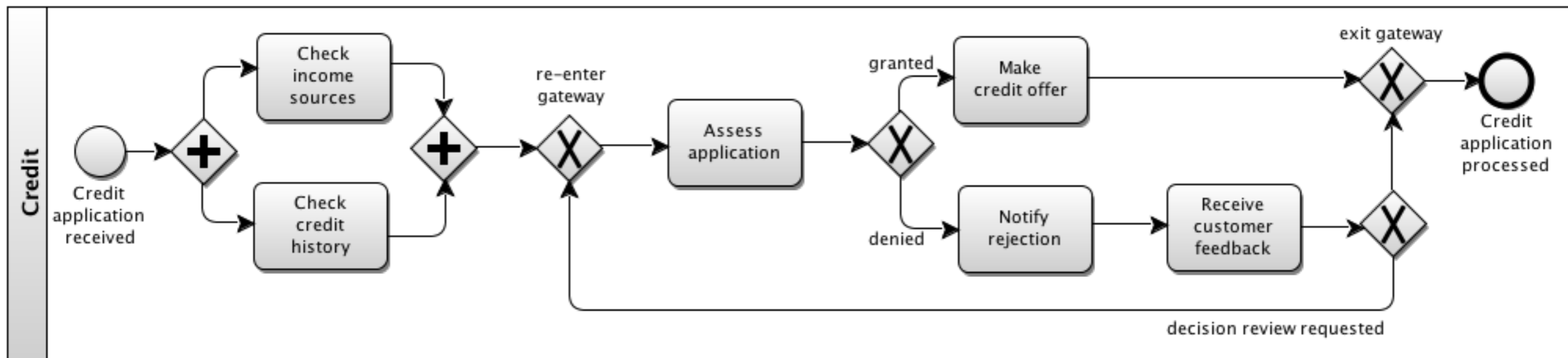
Bottlenecks: If resources are not sufficient to handle the scenario, then tasks will queue, causing high waiting times and cycle times.
A diagram shows the distribution of waiting times.

Resource utilization: Average utilization percentage of each resource.

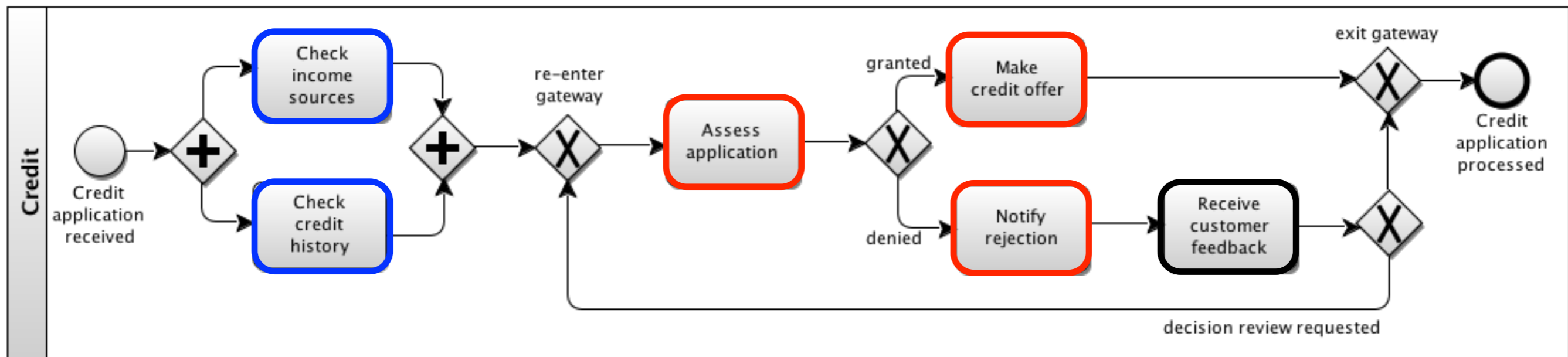
Cycle times (process duration): Total cycle time of the scenario, and diagrams about duration and cycle time distribution.

Simulation logs can be exported in MXML format and then imported in the ProM toolset for more detailed analysis.

Example: Credit application



Example: Credit application



#3 Clerks
(cost 25€/h)

#3 Credit officers
(cost 25€/h)

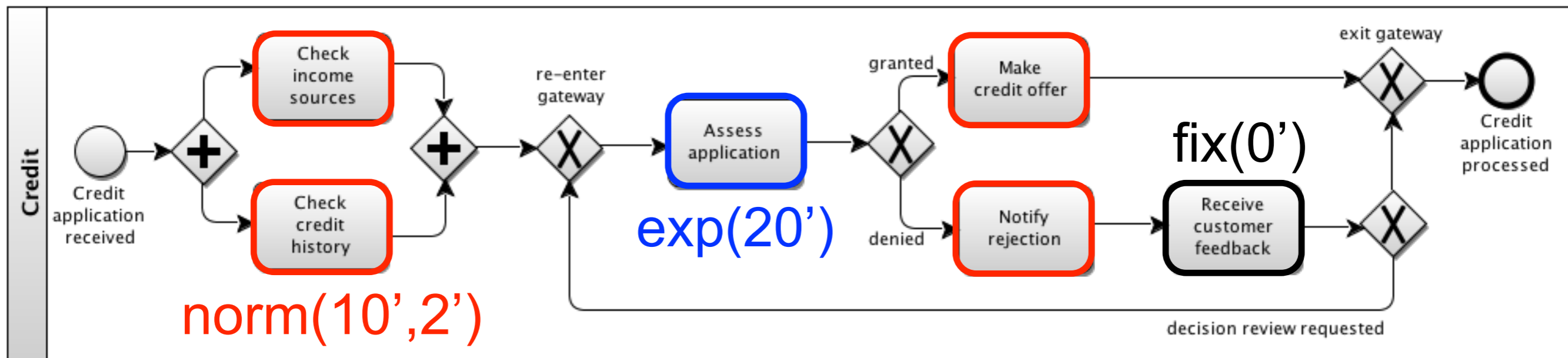
System

Working hours: monday/friday, 9:00/17:00

Example: Credit application

$\text{norm}(20', 4')$

$\text{norm}(10', 2')$



$\text{norm}(10', 2')$

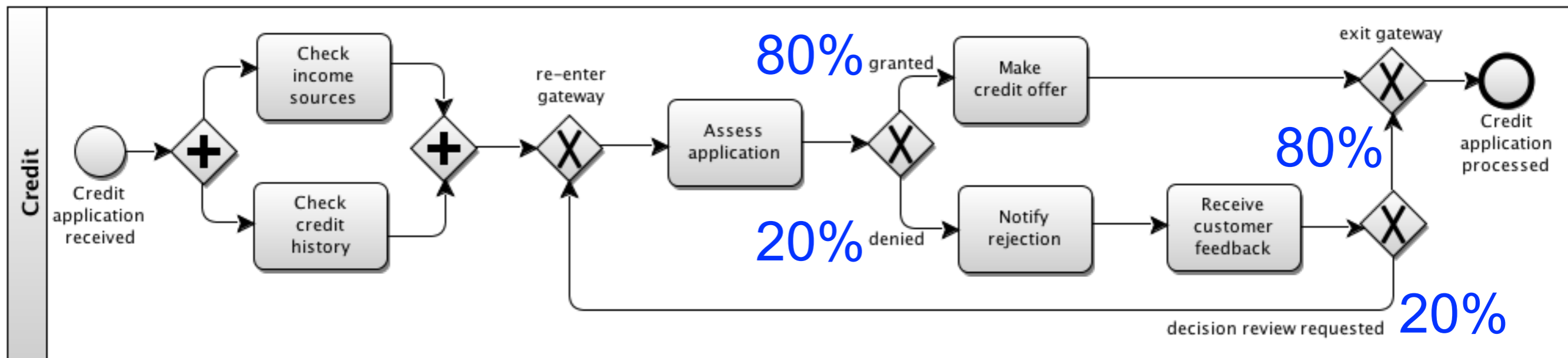
$\text{fix}(0')$

decision review requested

$\text{norm}(10', 2')$

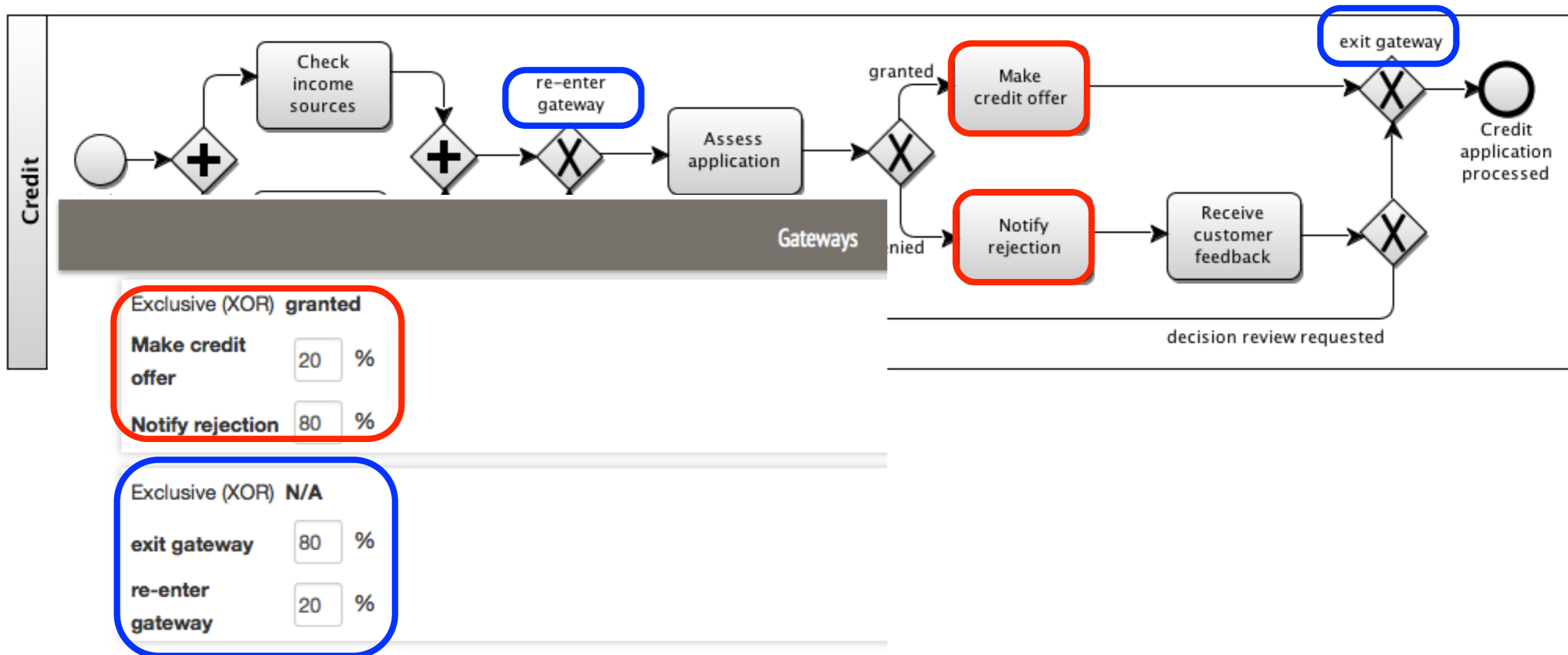
Inter-arrival time:
 $\text{exp}(30')$

Example: Credit application



Example: Credit application

Always assign names to items that follow any decision gateway:
BIMP use those names to indicate branching probabilities



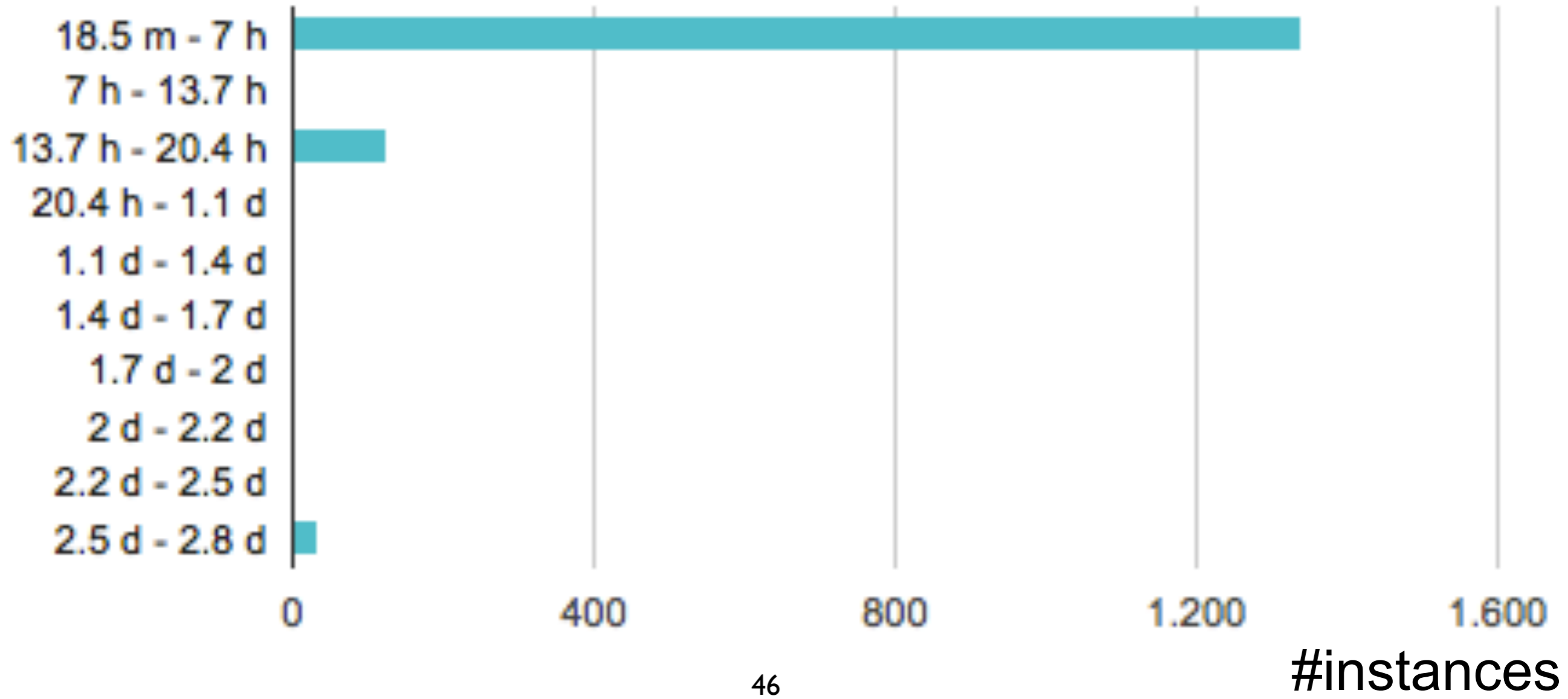
Simulation Results

General information

Completed process instances 1500
Total cost 62174.5 EUR
Total simulation time 19 weeks

Example: Credit application

Process cycle times (including off-timetable hours)



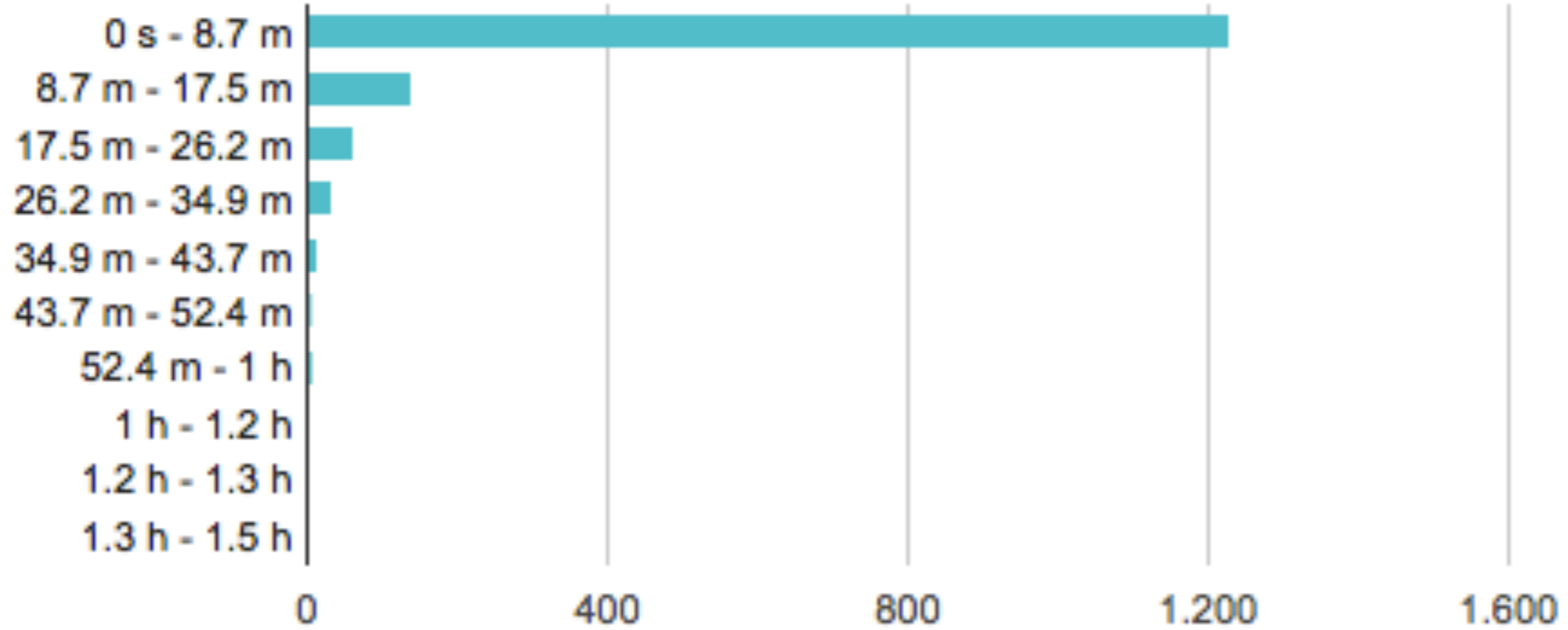
Simulation Results

General information

Completed process instances 1500
Total cost 62174.5 EUR
Total simulation time 19 weeks

Example: Credit application

Process waiting times



#instances

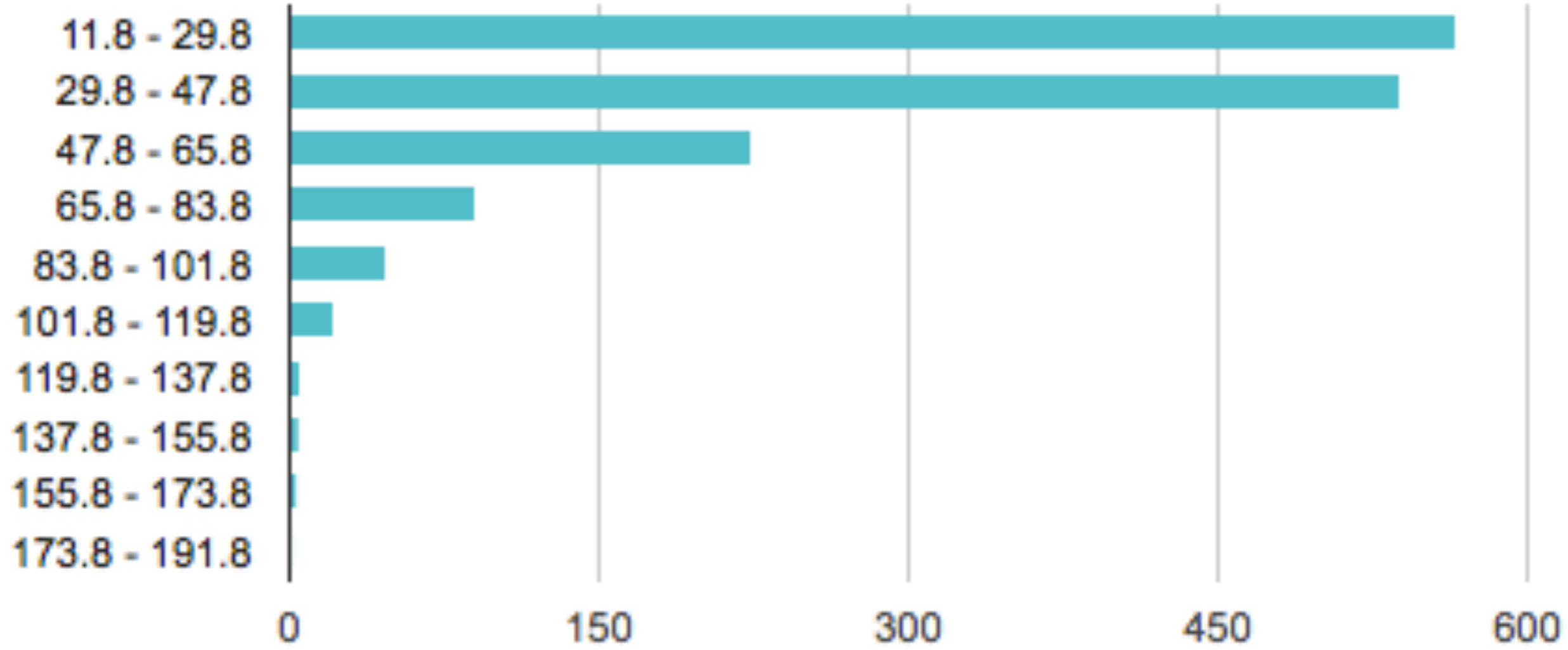
Simulation Results

General information

Completed process instances 1500
Total cost 62174.5 EUR
Total simulation time 19 weeks

Example: Credit application

Process costs (EUR)



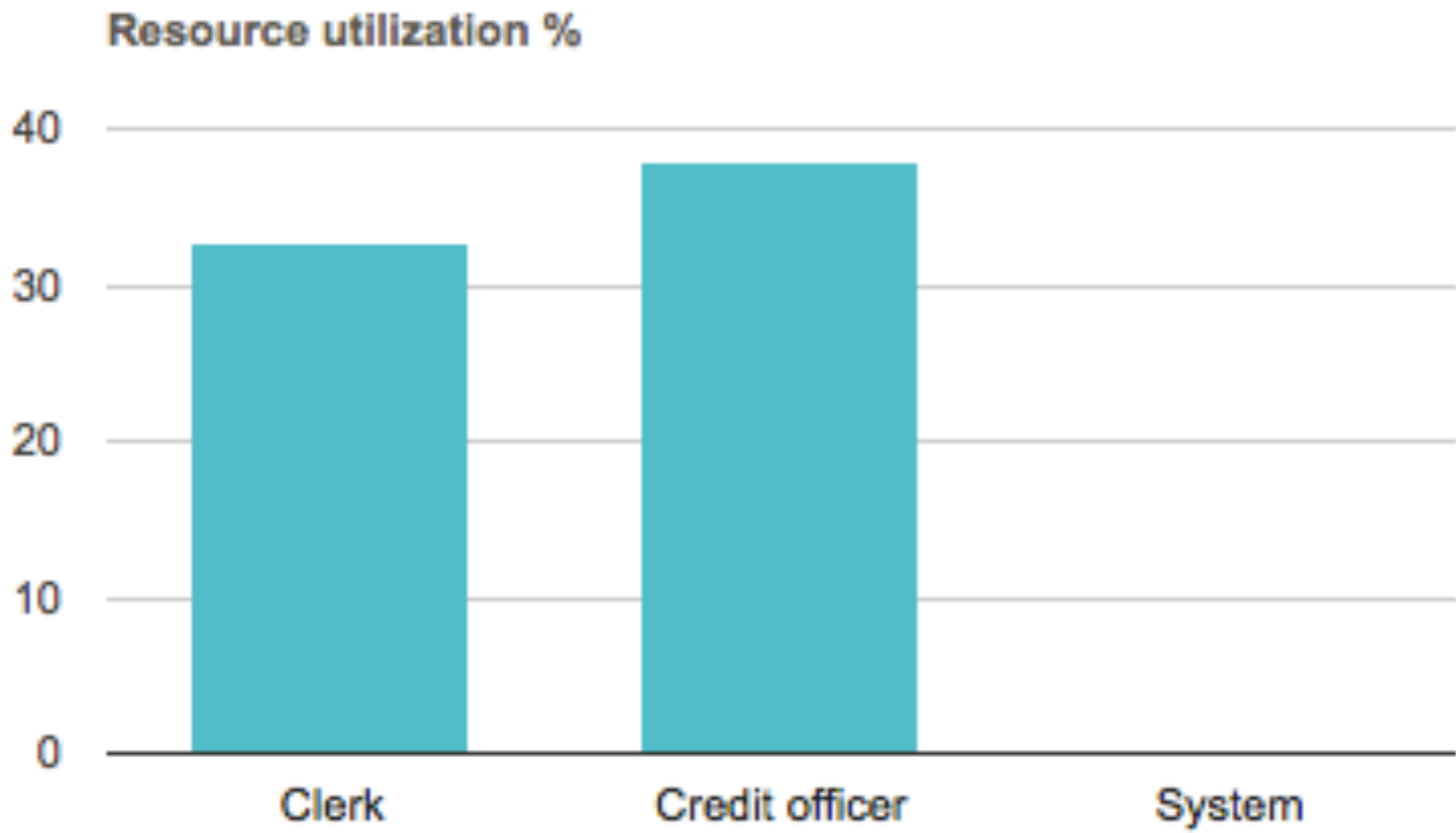
#instances

Simulation Results

General information

Completed process instances 1500
Total cost 62174.5 EUR
Total simulation time 19 weeks

Example: Credit application



Simulation Results

General information
Completed process instances 1500
Total cost 62174.5 EUR
Total simulation time 19 weeks

Example: Credit application

Process instance costs and cycle times (incl. off-timetable hours)

Minimum process cost 11.8 EUR	Maximum process cost 183.9 EUR	Average cost 41.4 EUR
Minimum cycle time 18.5 minutes	Maximum cycle time 2.8 days	Average cycle time 3.8 hours

General information

Completed process instances 1500
Total cost 62174.5 EUR
Total simulation time 19 weeks

Example: Credit application

Task costs and waiting times

Task name	Average cost	Average waiting time
Assess application	16.4 EUR	28.2 seconds
Check credit history	4.2 EUR	59.6 seconds
Check income sources	8.3 EUR	2.6 minutes
Make credit offer	8.4 EUR	21.9 seconds
Notify rejection	8.3 EUR	30.1 seconds

Advices

It is recommended to run the simulation multiple times and then take the average of results

Quantitative analysis in general and simulation in particular are based on simplified models: their reliability depends very much on the quality of inputs (check the sensitivity of the analysis w.r.t. small changes)

Process participants are humans, not robot: they are not all the same, they get distracted, get ill, change the way to handle cases, change job, their performance may vary,....

Cross-check simulation results against reality

Exercises

1. Run the BIMP simulation of the example by yourselves
2. Change the inter-arrival time to 15' and re-run the simulation: observe the changes in the results
3. Change the inter-arrival time to 10' and re-run the simulation: observe the changes in the results
4. Change the number of Clerks and Credit officers to 5 and re-run the simulation: observe the changes in the results
5. Change the branching probabilities to 50% and re-run the simulation: observe the changes in the results