Peer Assessment: Business Process modeling and analysis

Task #1 – Happy Path Modeling

a) Modell events, gateways, and activities

A process instance is created when an order is received from a customer. => (message) start event „order received“.

Due to the fact that customers can choose different ways to place an order, a received order has to be checked before any further step is performed. => activity „check order“.

***If** the order contains medicine that requires a prescription,* => gateway XOR split „prescription required/not required“ *the customer has to send it in soon. We assume that the customer sends the prescription always in time.*

After the prescription is received => activity „receive prescription“, *it has to be verified.* => activity „verify prescription“ ***

*Assuming the prescription can always be verified, **or** in case no prescription is required,* => gateway XOR join *the order is checked for possible drug-to-drug interactions.* => activity „check for drug-to-drug interactions“

Afterwards the ordered medicines are obtained from the warehouse => activity „obtain medicines from warehouse“ *and get boxed for shipment.* => activity „package medicines for shipment“

***While** obtaining the medicines and boxing them,* => gateway AND split *an invoice is created* => activity „create invoice“ *and sent to the customer* => activity „send invoice“.

***When** the box is packaged **and** the payment is received,* => activity „receive payment“, gateway AND join *the mail order pharmacy ships the parcel to the customer.* => activity „issue shipment“

Process ends. => end event „order fulfilled“

*** Remarks: The activities „receive prescription“ and „verify prescription“ could potentially be modelled as a single activity. However, I decided to make things more explicit by using two activities here.

b) Create business process model

Guidelines/best practices (from the slides):

- Use a consistent vocabulary. √
- Label events in the “noun verb (participle)” style. √
- Start each business process model with exactly one start event. √
- Label activities in the “verb noun” style. √
- Use gateways for splits and joins. √

- Always use default flow with an exclusive gateway. √
- Avoid mixed gateways. √
- Use a distinct end event for each possible outcome of the process. √

Please focus only on the process of the mail order-pharmacy during modeling. Please use only BPMN basic elements as start events, end events, gateways and simple activities. It is not requried to assign task types to the BPMN activities. √

=> Take a look at the attached PDF „Mail-order-pharmacy-task1.pdf“ with the process model.

Task #2 – Exception Modeling

a) Modell exception flow

If the order is incomplete or any other problem is detected with it, the order check will be ***interrupted****, the order will be cancelled, and the customer will be informed.* => intermediate error event, message send task „cancel order“, end event „order cancelled“

Due to legal requirements, the verification process of the prescription can only be done ***manually*** *by employees of the pharmacy.* => user task „verify prescription“

If the prescription is invalid or does not fit the order, the verification will ***interrupted****, the customer will be informed about the issue, and the order will be cancelled.* => intermediate error event, message send task „cancel order“, end event „order cancelled“

Based on the check regarding the drug-to-drug interaction, it is decided whether to continue with normal process execution or with order cancelation if an interaction is detected whereby the customer is informed about the issue. => gateway XOR split, message send task „cancel order“, end event „order cancelled“ ***

The order check and the drug-to-drug interaction check are done ***manually****.* => user task „check order“, user task „check for drug-to-drug interactions“

Obtaining and boxing the medicine ***is done by an employee*** *of the pharmacy whereas the invoice creation* ***is done automatically*** *by an IT system.* => user task „obtain medicines from warehouse“, user task „package medicines for shipment“, service task „create invoice“

*** Remarks: The drug-to-drug interaction description does not explicitly mention that the drug-to-drug interaction check should be interrupted in case of issues. Thus I decided to use a XOR split here – otherwise I would have choosen an intermediate error event.

b) Extend business process model

Please assign tasks types to the activities of the process model. To ease the grading procedure, please use only tasks types discussed in the course (2-18): Message task (receive and send), user task and service task. √

Remarks: The description does not include a requriment to separate between different

„order cancelled“ end events – thus I decided to use the same „order cancel“ path for the various error conditions (order incomplete, prescription invalid, drug-to-drug interaction detected).

The „cancel order“ and „inform customer“ activities could potentially be modelled as separate tasks. However, I decided to use a single task and bundle everything into a single message send event „cancel order“.

=> Take a look at the attached PDF „Mail-order-pharmacy-task2.pdf“ with the process model.

Task #3 – Process Analysis

3.1) Soundness Check

Please copy your model of task #2 including the exceptions and create an equivalent structural sound model.

A process model is **structurally** sound, if

- it has exactly one start event *i* (for „input“ of the process)
- it has exactly one end event *o* (for process „output“), and
- each node in the process model is on a path from *i* to *o*

=> The original process modell was structurally unsound due to the multiple end events. After transformation all of the above statements are fulfilled and the process model is now structurally sound. Take a look at the attached PDF „Mail-order-pharmacy-task3.pdf“ with the process model.

After creating the structural sound model, please check if your model is sound. Explain the result of your analysis.

A process model is **sound**, if it fulfills the soundness properties:

1. Once a process has started, regardless of which decisions are taken by the process, at some point, the process will reach the end event *o*
2. If and when it reaches *o*, the process has completely terminated, i.e., no further activities can be executed by the process anymore
3. Each process activity participates in at least one execution (in one trace)

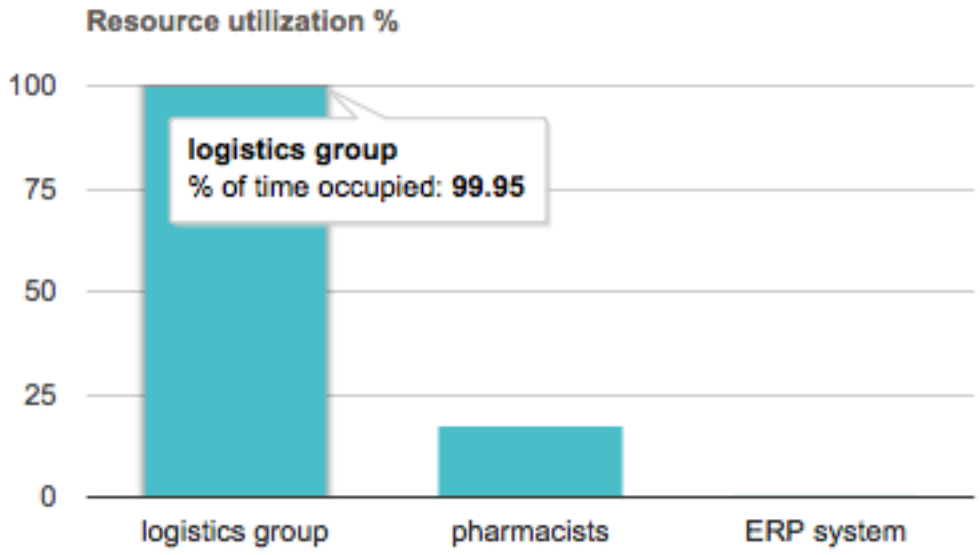
=> The process model is sound as it fulfills the properties (1), (2) and (3). There are no violations of the soundness properties such as deadlocks, livelocks or dead activities.

3.2) Multi-Instance-Analysis

Simulation Results

General information

Completed process instances 1000
Total cost 13022 EUR
Total simulation time 2.7 weeks



Task costs and waiting times

Task name	Average cost	Average waiting time
check for drug-to-drug interactions	2.3 EUR	10.5 seconds
check order	0.8 EUR	5.4 hours
create invoice	0 EUR	0.5 seconds
issue shipment	4.4 EUR	5.8 hours
obtain medicines from warehouse	2.5 EUR	6 hours
package medicines for shipment	2.2 EUR	6.1 hours
receive payment	0 EUR	0 seconds
receive prescription	0 EUR	0.3 seconds
send invoice	0 EUR	0 seconds
verify prescription	2.3 EUR	10.2 seconds

What is the total cost of the analyzed scenario?

=> The total cost of the scenario (with 1000 process instances completed) is around 13022 EUR (obviously it varies a bit in each simulation run).

Is there any bottleneck in this process? Prove your statement using the results of your analysis.

=> The logistics group becomes the bottleneck in the process. The team is occupied around 99.95% of time during the process simulation. As a consequence the average waiting time of acitivities which are performed by the logistics team is in the area of several hours, e.g. check order (5.4 hours), obtain medicines from warehouse (6 hours), package medicines for shipment (6.1 hours), and issue shipment (5.8 hours).

Make at least two recommendations to improve the process.

=> Anything that takes load off from the logistics group would improve the overall process performance:

- Increase the number of employees in the logistics group
- Perform activities automatically. For example, instead of manually performing the „check order“ step, try to automatically perform this step by the ERP system.
- Reduce the duration times of activities, e.g. introduce an information system for the warehouse – thus obtaining the medicine from the warehouse could be done much faster.