02291 System Integration

Goal-oriented Requirements Engineering: Solutions to exercises

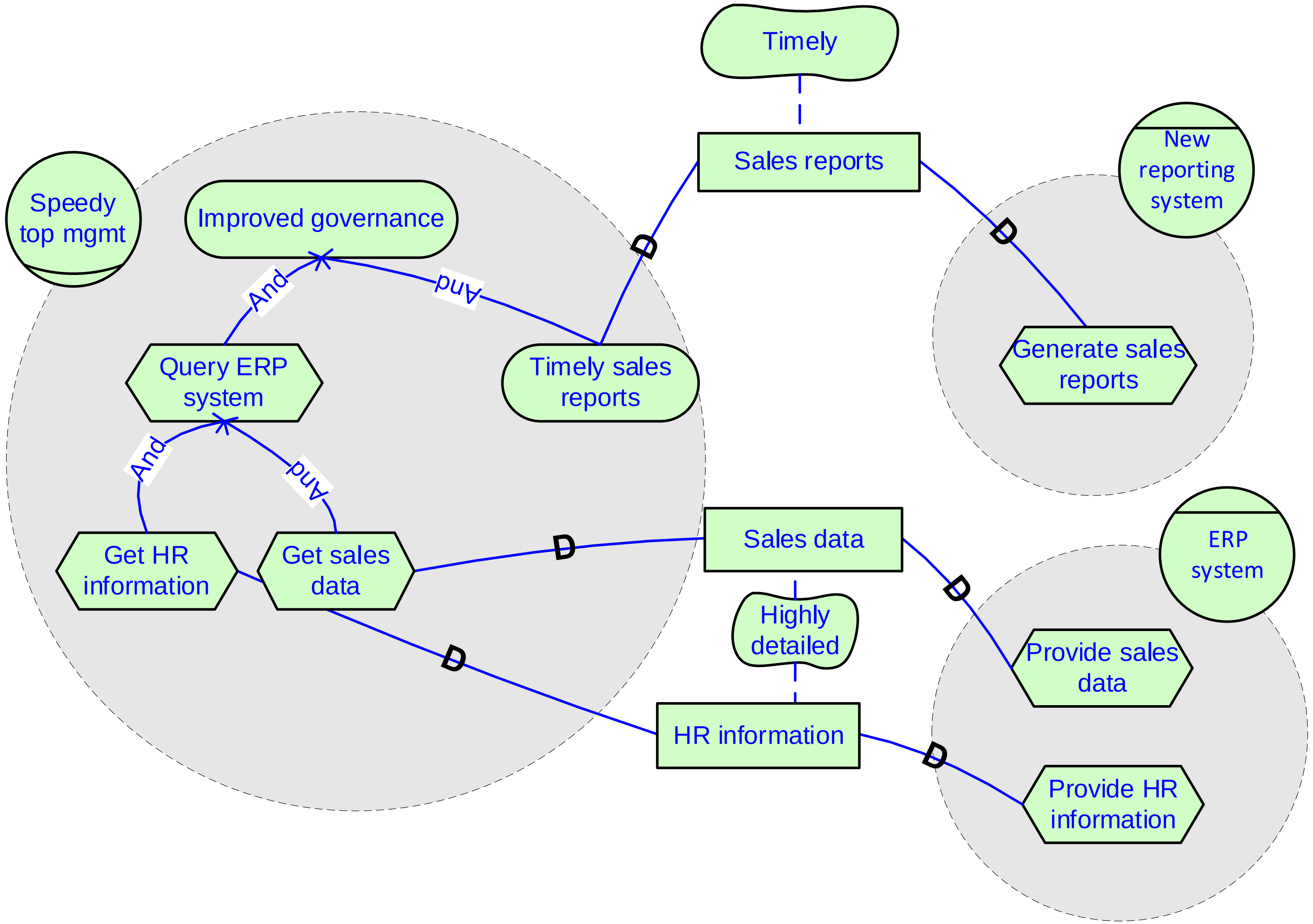
Exercise 1 – Speedy

Speedy is an international delivery company that needs to refocus on which markets it should operate. Indeed, Speedy lacks a clear understanding on which markets are the most profitable. Thus, to address this issue, Speedy’s top management aims at improving their governance. To achieve this, the top management requires sales reports containing timely information. Thus, the top management decided to build a new reporting system to automatically produce such reports. After inspecting the sales reports, the top management may also need to query the existing ERP system to get detailed sales and HR information.

Prepare an ArchiMate and an I-star model that captures the formalizes Speedy’s new architecture.

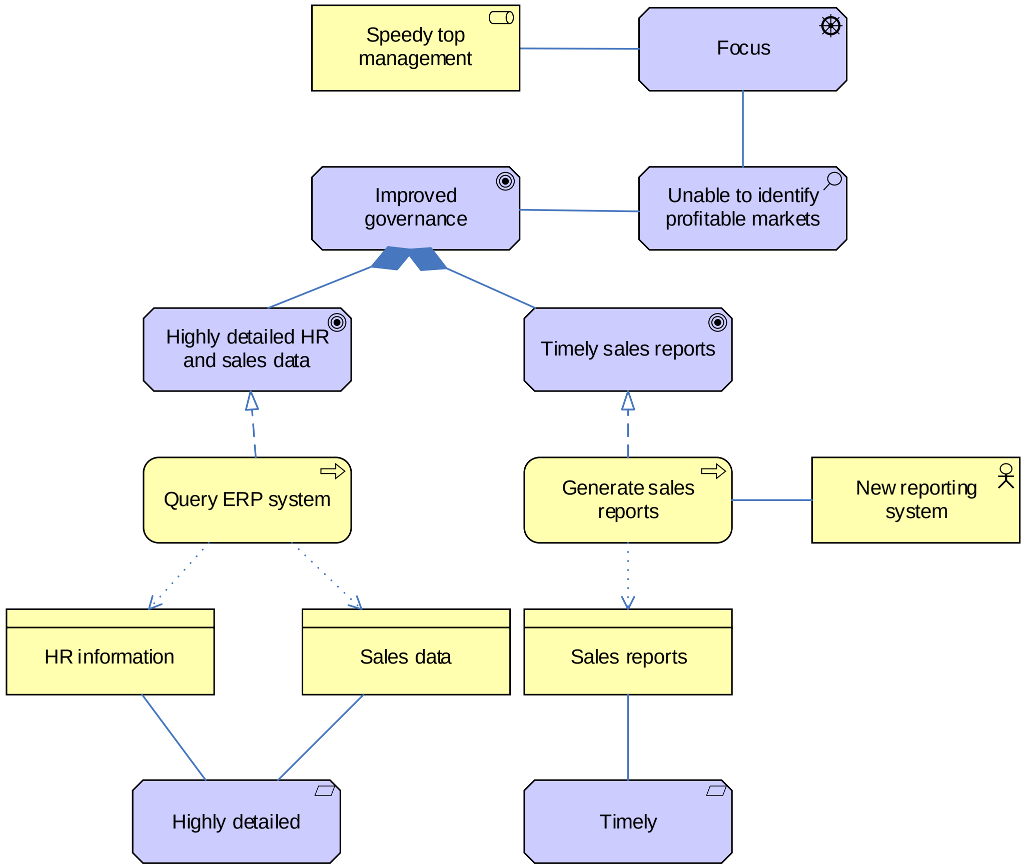
- Can you model all the elements and relations described in this exercise with both languages? If not, which elements can be captured only in ArchiMate? Which ones in I-star?

Exercise 1 – I-star model



We cannot model the driver and assessment concepts

Exercise 1 – ArchiMate model



We cannot explicitly model the social dependency between Speedy’s top management and the new reporting systems

Exercise 2 – IC

IC is an insurance company which wants to offer a new insurance service for small assets (<2000$) managed completely online for reliable customers.

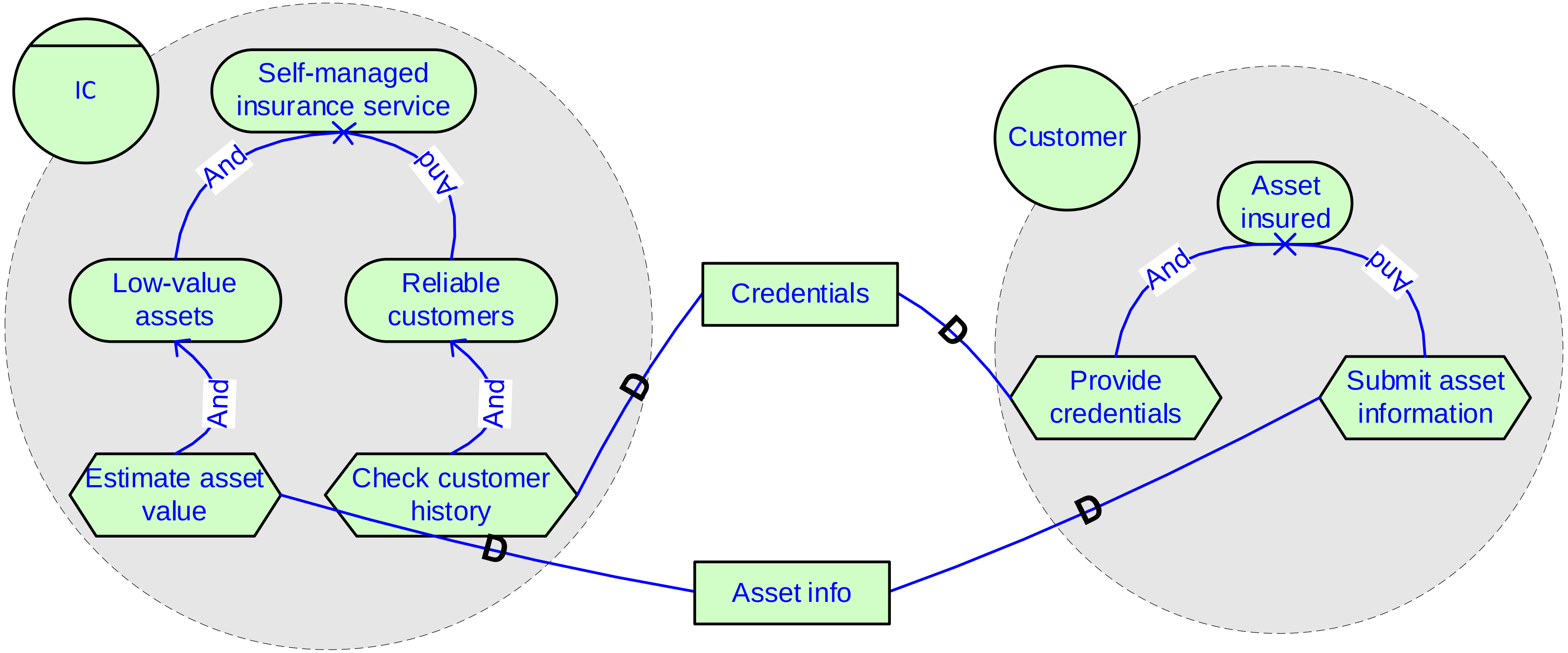
To achieve this, a customer who wants his assets to be insured has to provide its credentials and photo of the asset and its details (serial number, purchase date) to IC. To ensure that the customer is reliable and the asset inexpensive, IC will then check the customers credentials and past history and estimate the asset’s price.

Prepare an ArchiMate and an I-star model that captures the formalizes IC’s requirements.

- Can you model all the elements and relations described in this exercise with both languages? If not, which elements can be captured only in ArchiMate? Which ones in I-star?

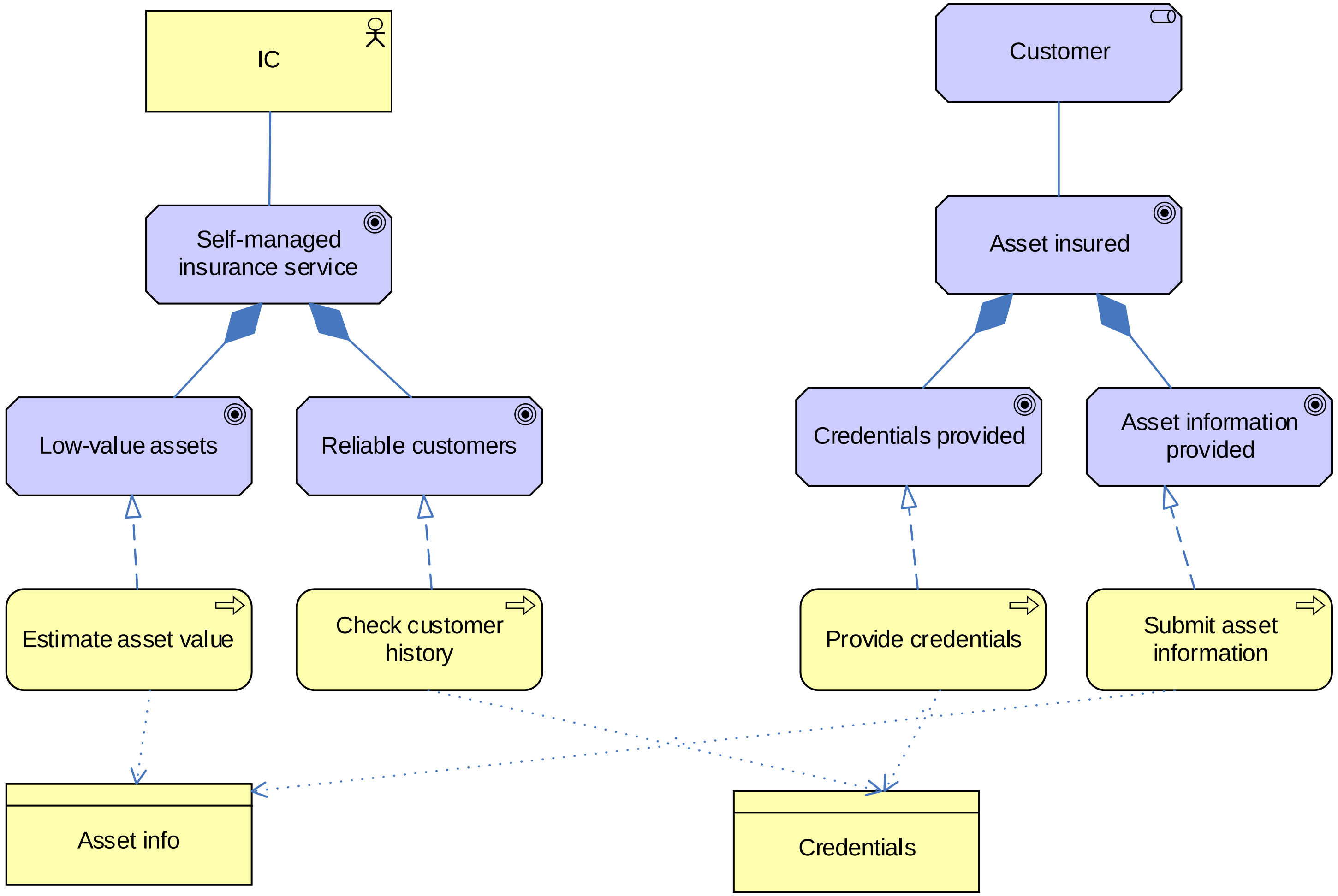
Exercise 2 – I-star model

The model captures all concepts and dependencies

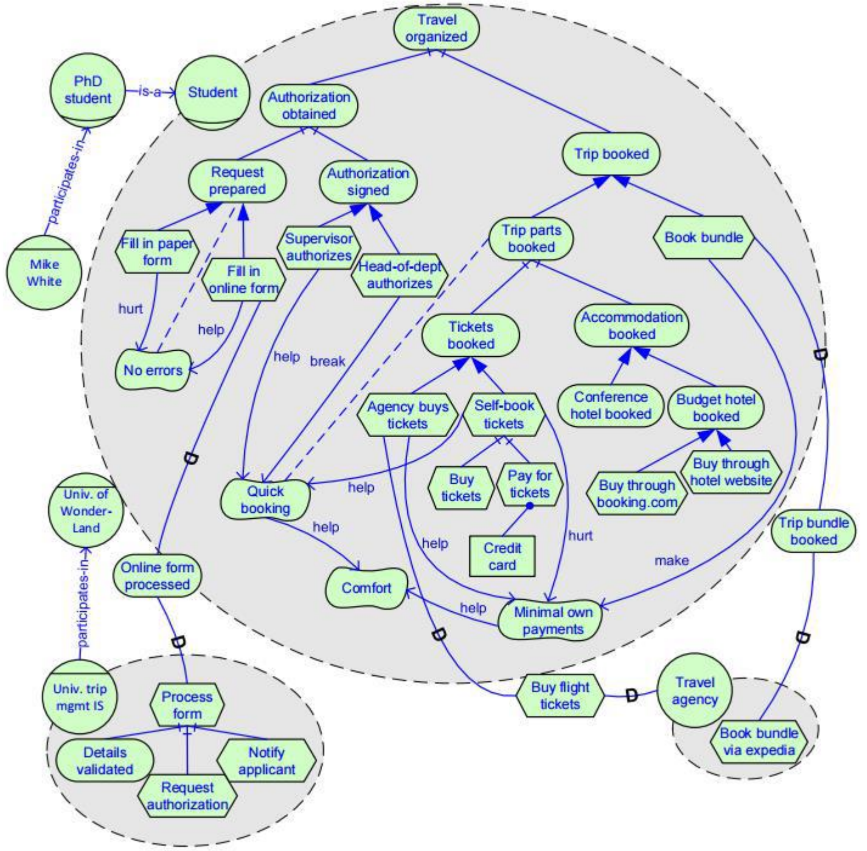


Exercise 2 – ArchiMate model

We cannot explicitly model the social dependencies between IC and the customer



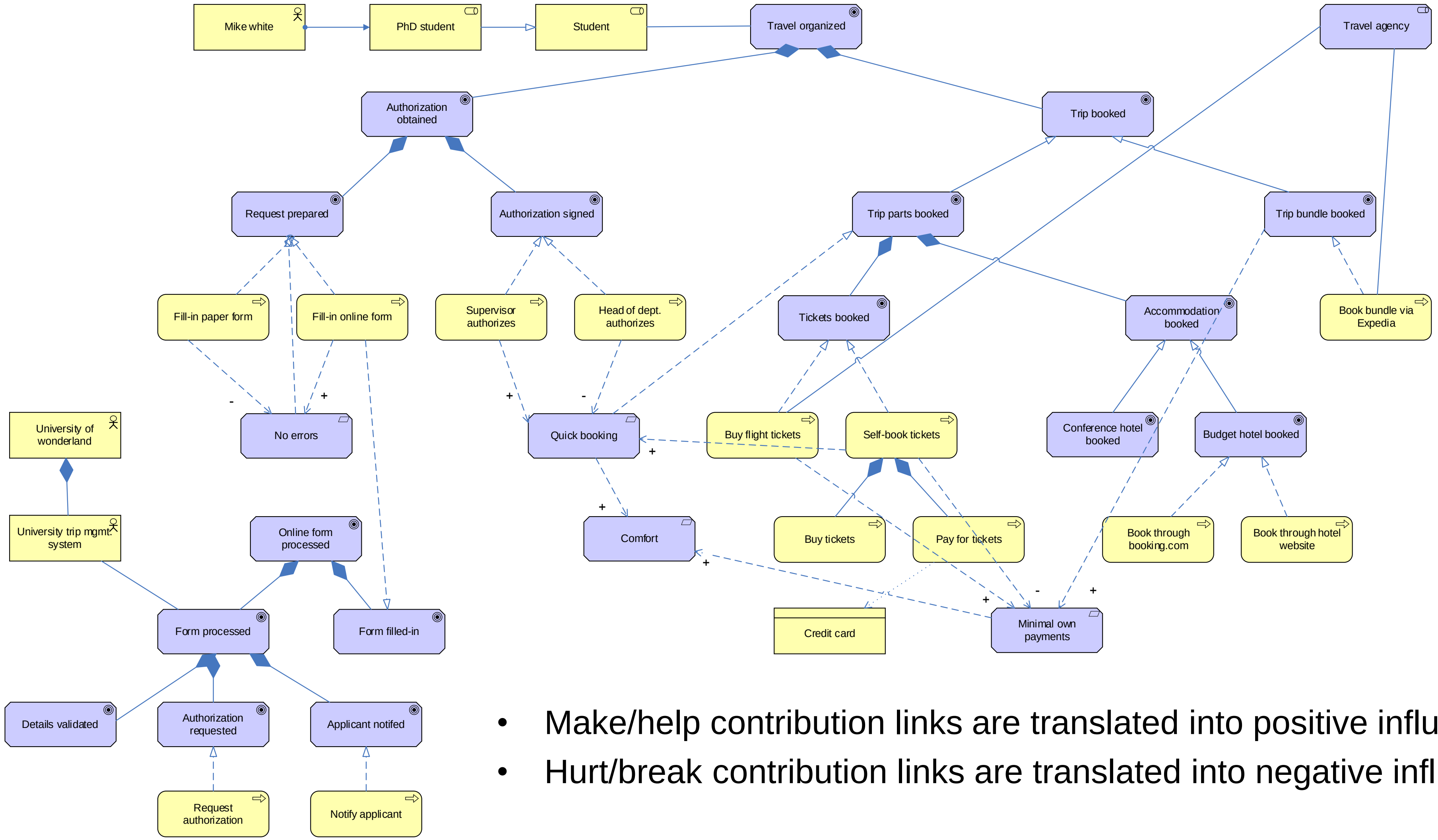
Exercise 3 – University travel reimbursement



- This I-star model represents a university travel reimbursement system
- Prepare an ArchiMate model representing the same system
  - – Can you model all the elements and relations in this I-star model?
  - – If not, which elements and relations cannot be modeled?

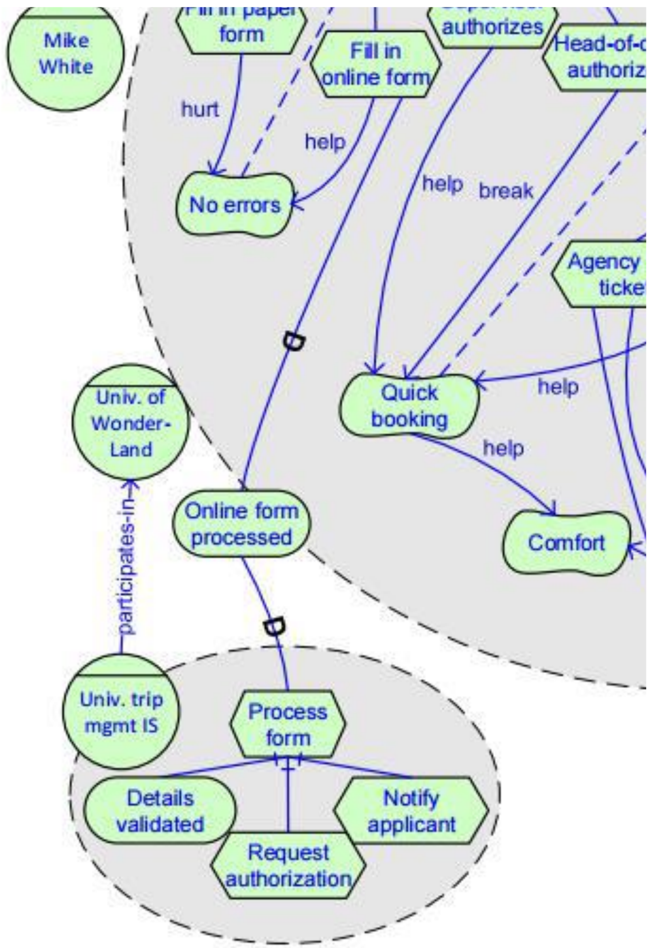
Model from: F. Dalpiaz, X. Franch, and J. Horko – iStar 2.0 Language Guide
https://arxiv.org/abs/1605.07767

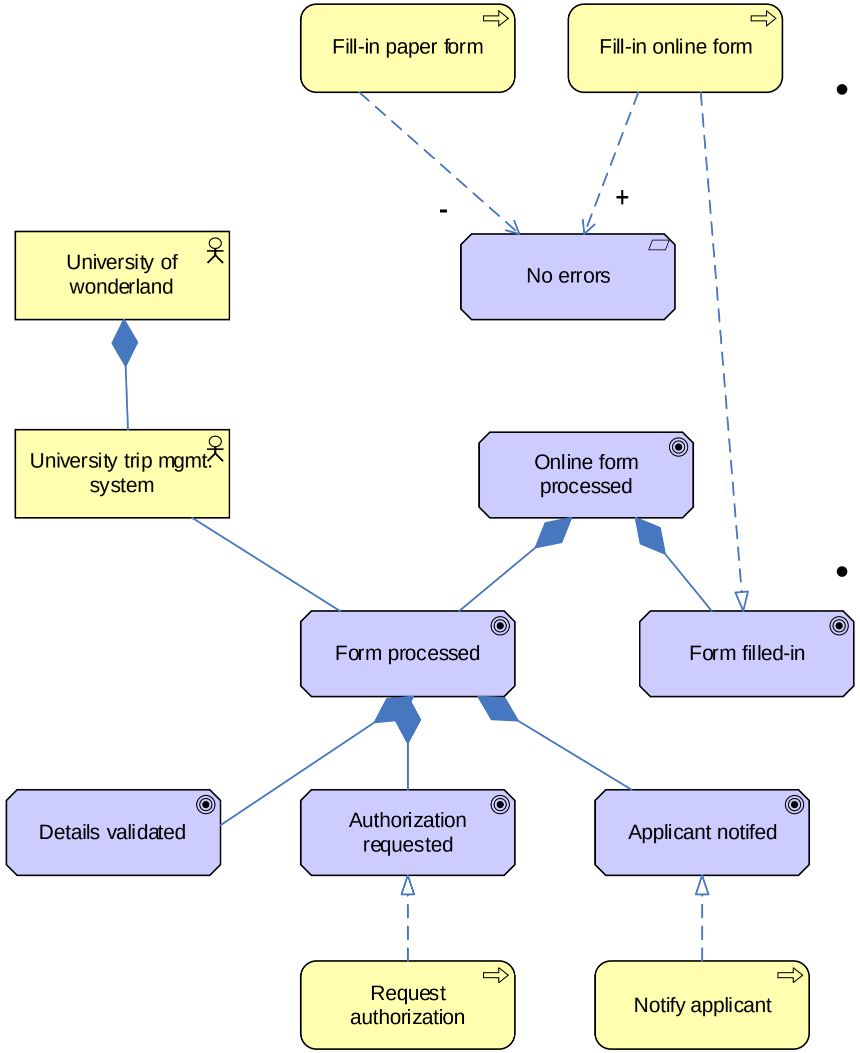
Exercise 3 – ArchiMate model



- Make/help contribution links are translated into positive influence relations
- Hurt/break contribution links are translated into negative influence relations

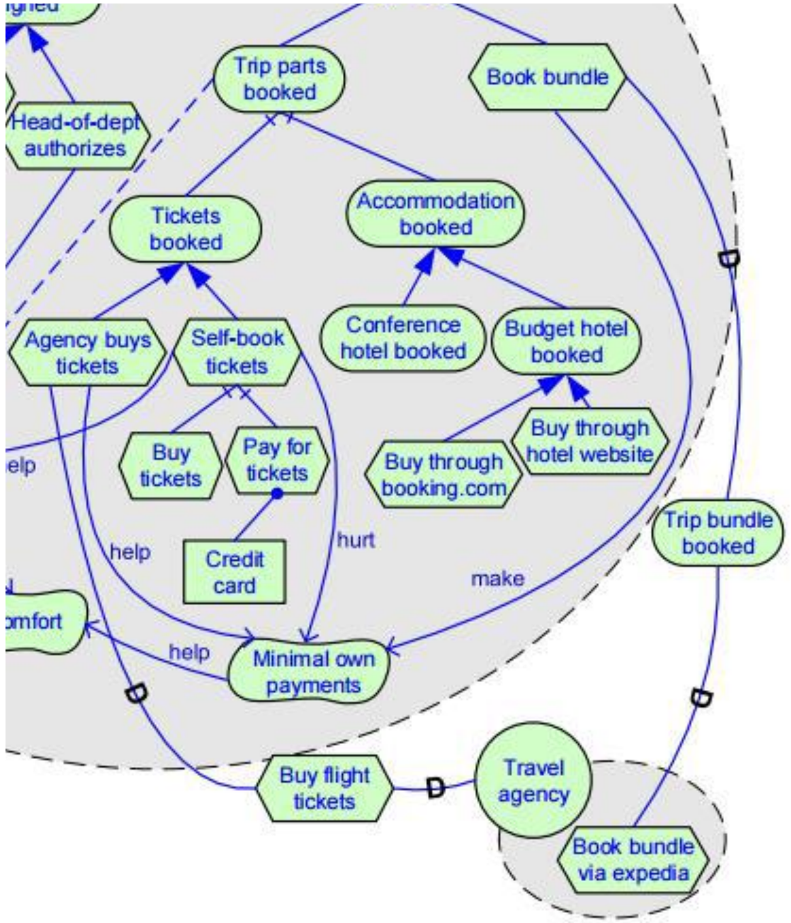
Exercise 3 – Comparison

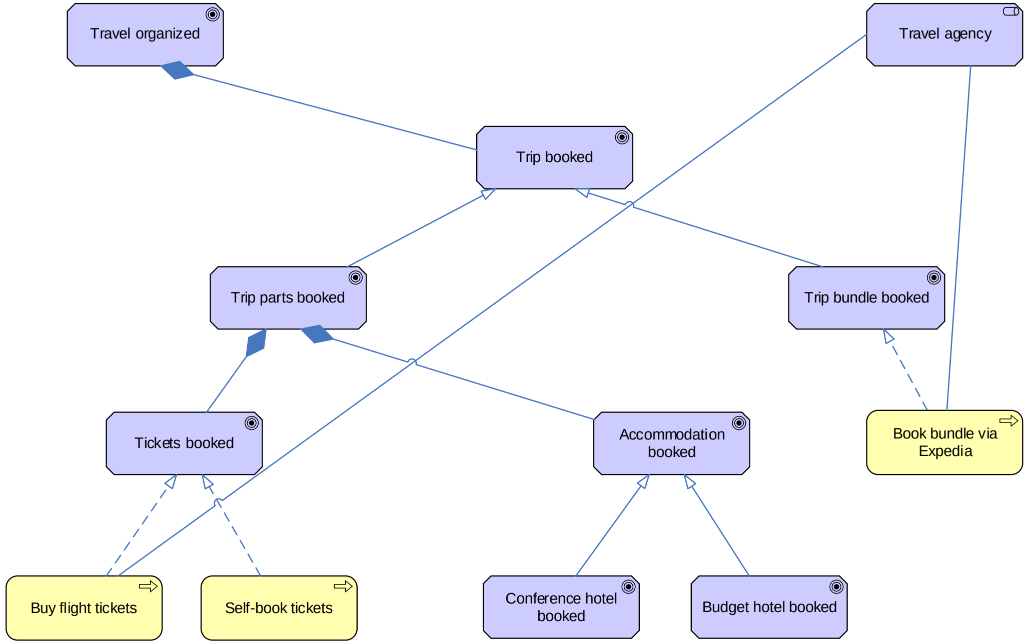




- We cannot model the social dependency between IS and student
  - We can approximate it by breaking the dependum (Online form processed) into subgoals, which are inclusively realized by subgoals belonging to the two entities
- We cannot model that goals are refined from tasks:
  - We either turn Process form into a goal (as depicted), or turn Details validated into a task

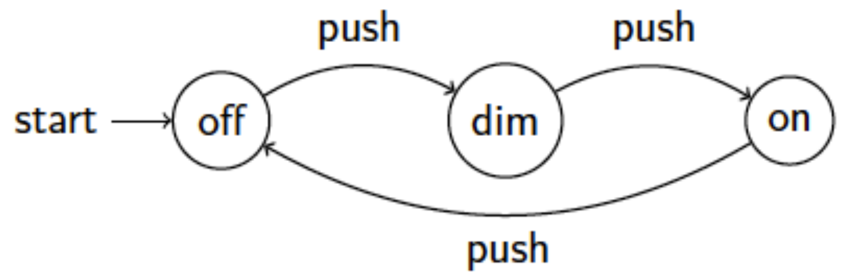
Exercise 3 – Comparison





- We cannot model the social dependencies between travel agency and student
- We can approximate them:
  - We replace task Agency buys tickets with dependum Buy flight ticketst, and we assign it to the travel agency
  - We replace task Book bundle with dependum Trip bundle booked, which is realized by task Book bundle with Expedia

Alphabets, Strings and Languages



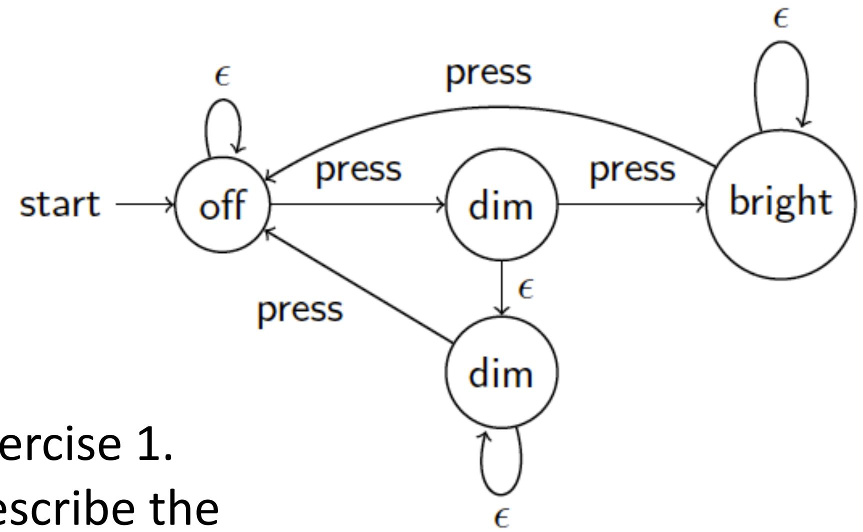
- An *alphabet* Σ is a set of symbols (or letters)

Σ={push}

- A *string* *w*=$a_1, a_2, \ldots, a_k$ is a sequence of symbols from Σ

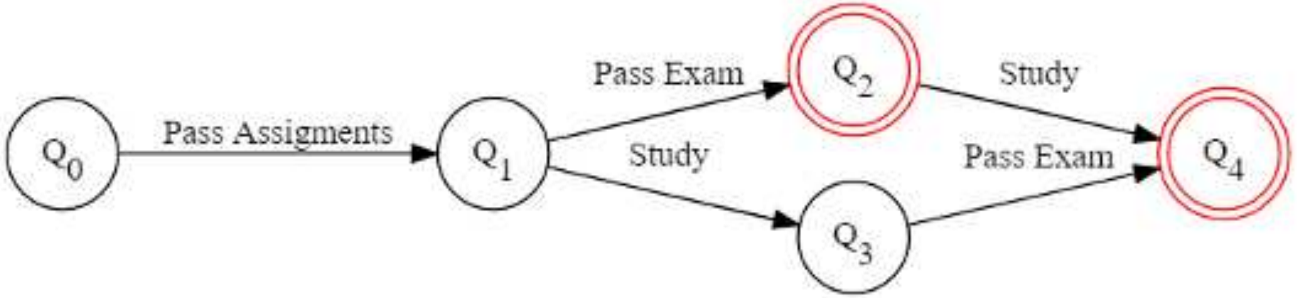
w_1= push,push;
w_2= push,push,push;
w_3=
push,push,pushpush,push,push;
…

- The empty string is written ε
- A language *L* is a set of strings over Σ, that is, a subset of Σ*
  - It includes the empty language: Ø
  - It includes the language of the empty string: {ε}



Exercise 1.
Describe the
language of this
automaton

How does a DFA define a language?

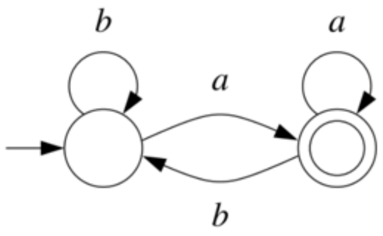


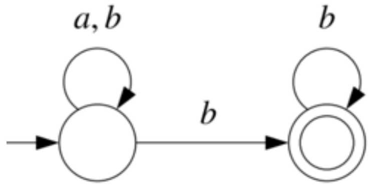
- Consider a string *w*
- Start in the initial state *Q_0*
- Read the first symbol, say *Pass Assignments*, of w
- Determine the new state $Q_1 = \delta(Q_0,$ *Pass Assignments*$)$
- Read the second symbol, say *Pass Exam*, of w
- Determine the new state $Q_2 = \delta(Q_1,$ *Pass Exam*$)$
- ...
- Let *Q_k* be the state obtained after having read the last symbol of *w*
- If *Q_k* is in F then accept w; otherwise reject w

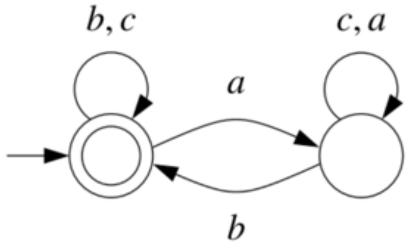
Exercise 2:
How many words exist in the language above?

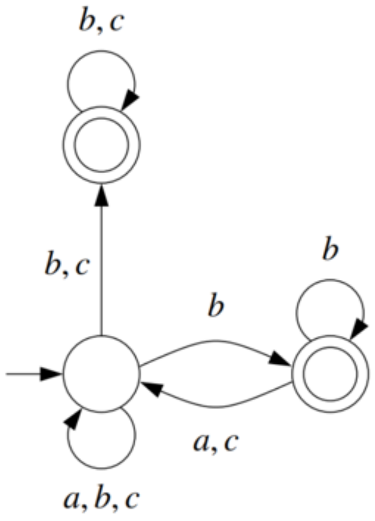
Exercise 3

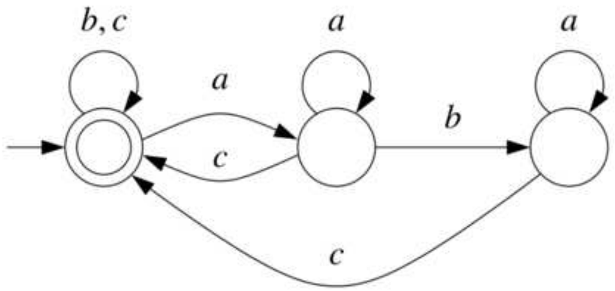
- Describe in natural language the following automata





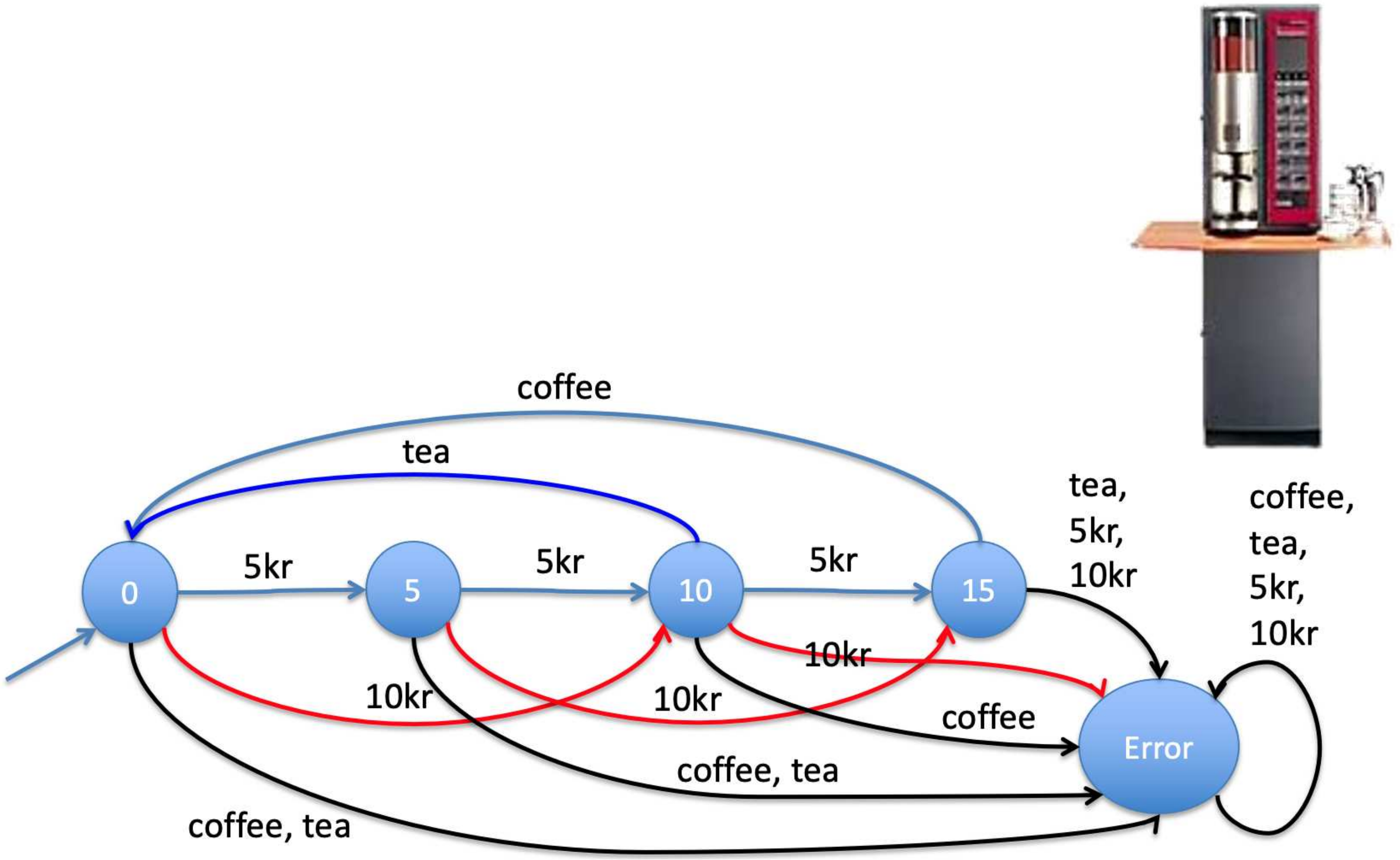






Exercise 4

- In the programming language of your choice, implement the following vending machine



Exercise 5

- Extend the vending machine:
    - To offer chocolate for 20kr.
    - To accept coins of 20kr.

Exercise 6 2x2 puzzle

- Model the 2 × 2 puzzle, the simplest form of the N × N puzzle (https://en.wikipedia.org/wiki/15_puzzle) with a DFA. Which sequences of transitions help you solve the puzzle?

02291 System Integration

Introduction to ArchiMate: Solutions to exercises

Exercise 1 – Speedy

Speedy is a delivery company that wants to create a new reporting system for the top management. After inspecting the sales reports, the top management may also need to query the existing ERP system, based on Oracle Fusion, to get detailed sales and HR information.

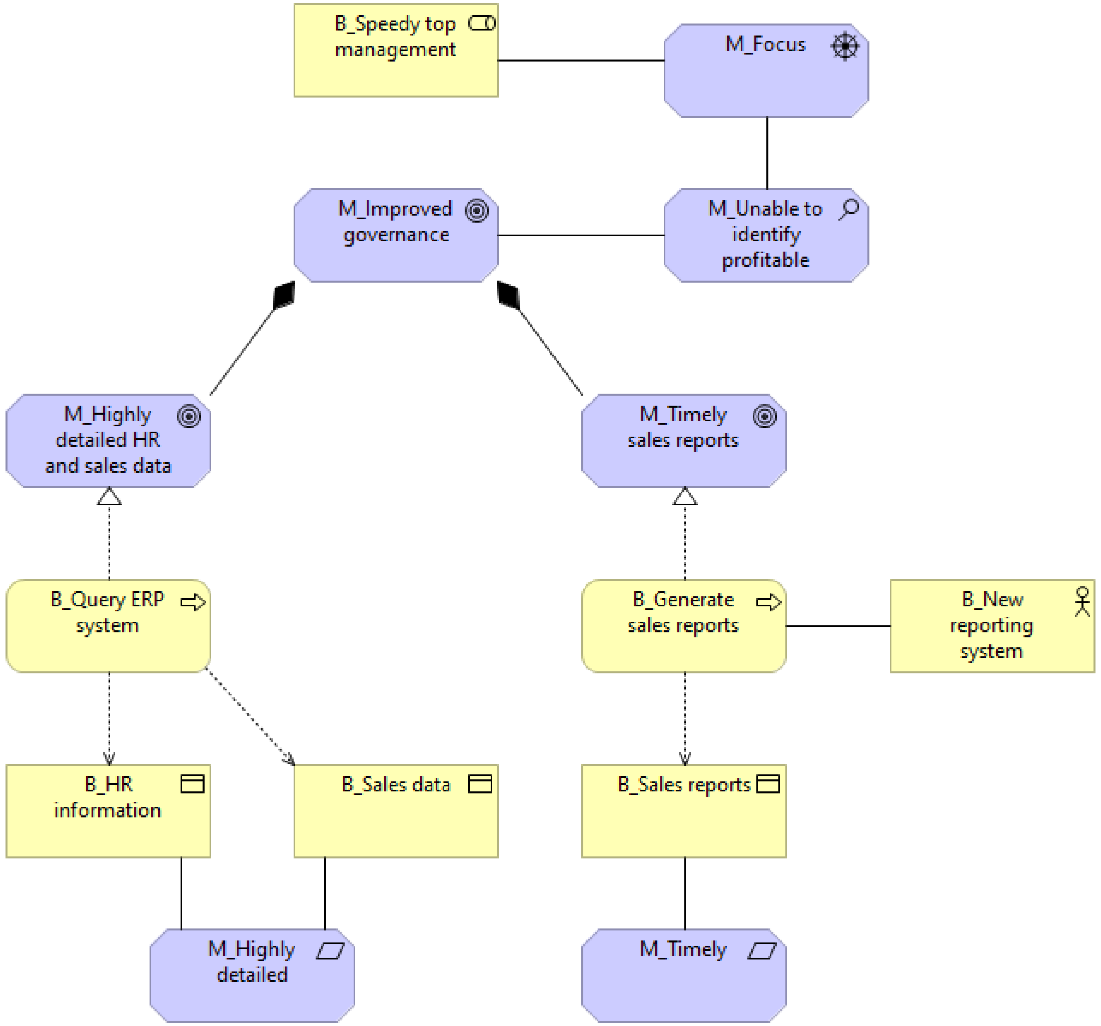
To implement the reporting system, a data warehouse (DW) based on Microsoft SQL Server 2022 will be used. The DW will run three components responsible for extracting sales data from the ERP database, managing analysis data, and generating sales reports.

The DW will run on-premise on a different node than the one running the ERP. Both nodes will share the same LAN.

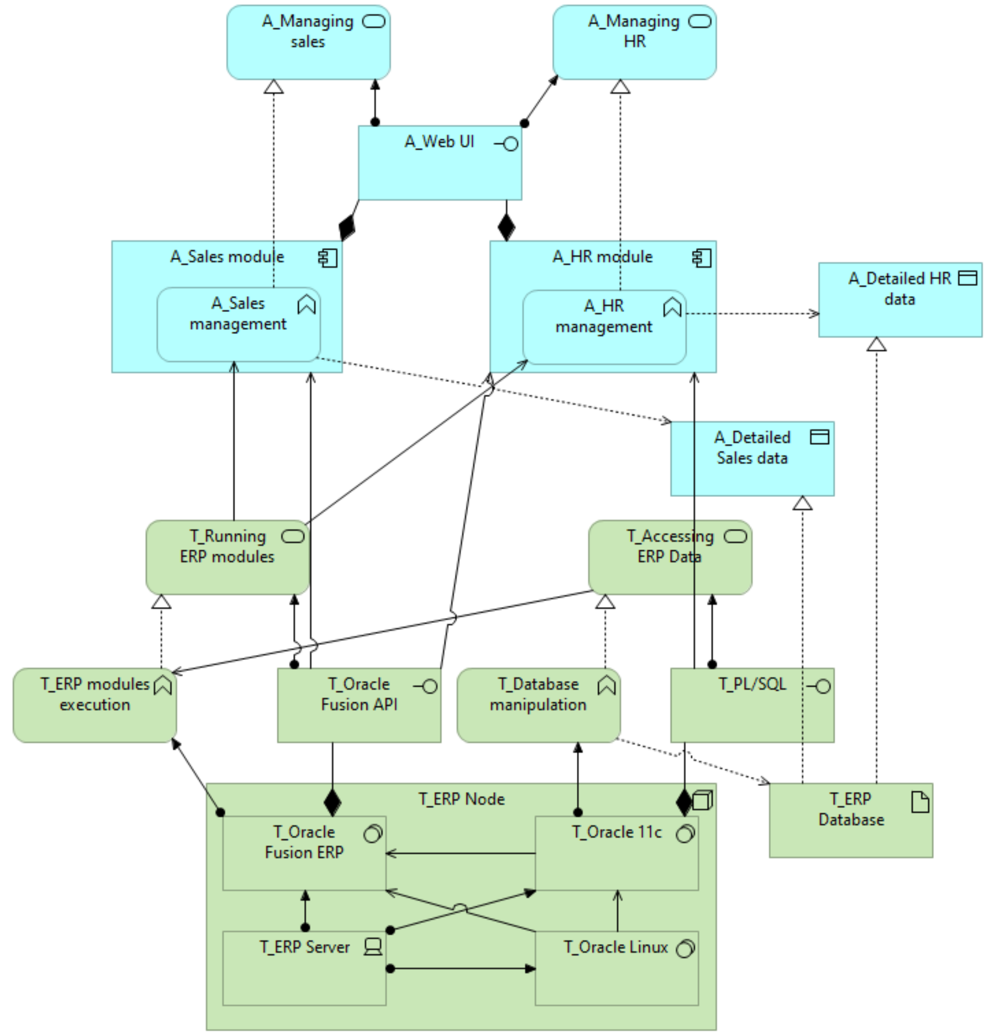
ArchiMate models representing the business goals and the existing ERP system are enclosed in the next slides.

Starting from these models, create a complete ArchiMate model that: 1) fully represents the business layer and links it to the application layer, 2) extends the application and technology layers by adding the elements required for the new reporting service.

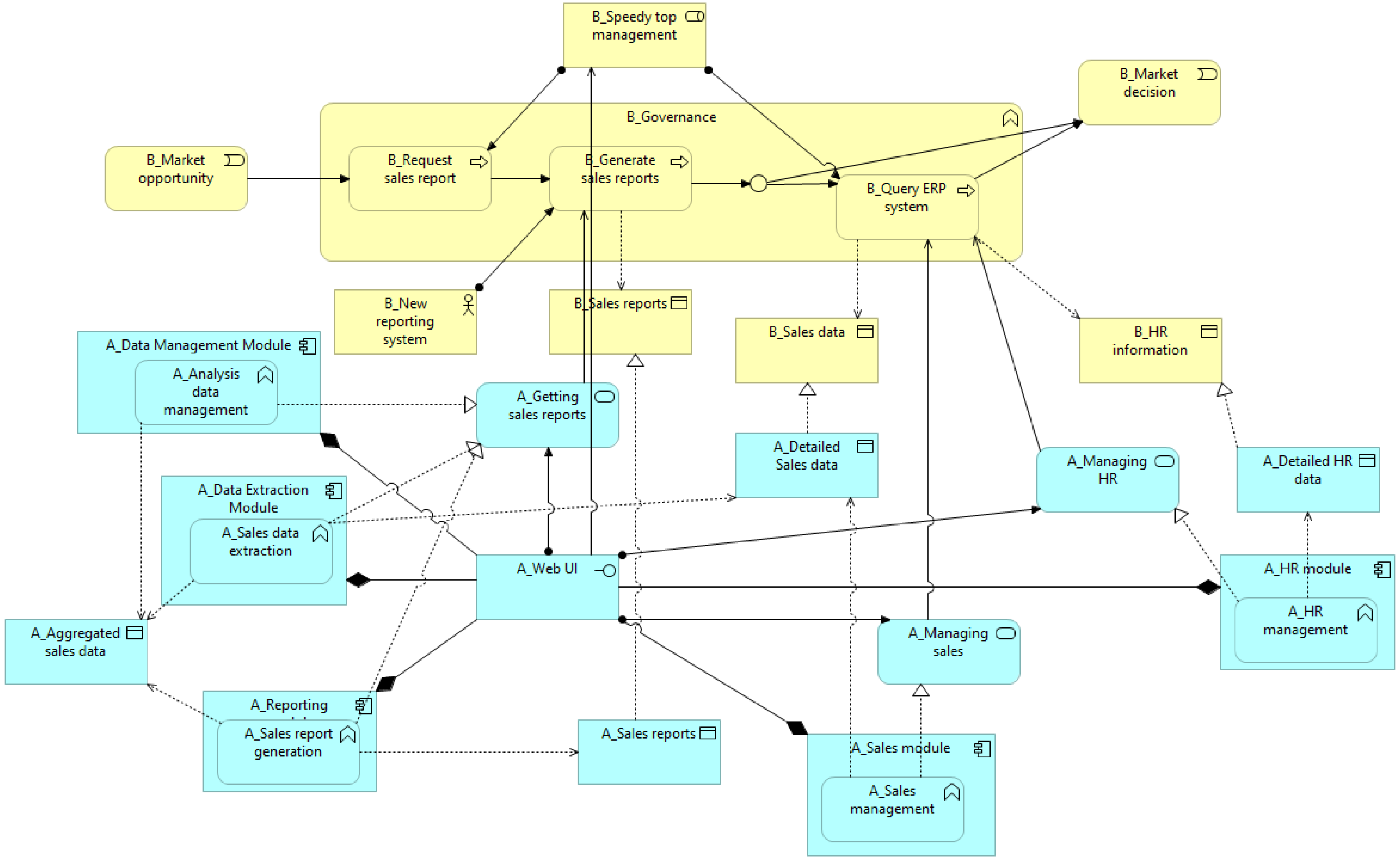
Exercise 1 – Motivation



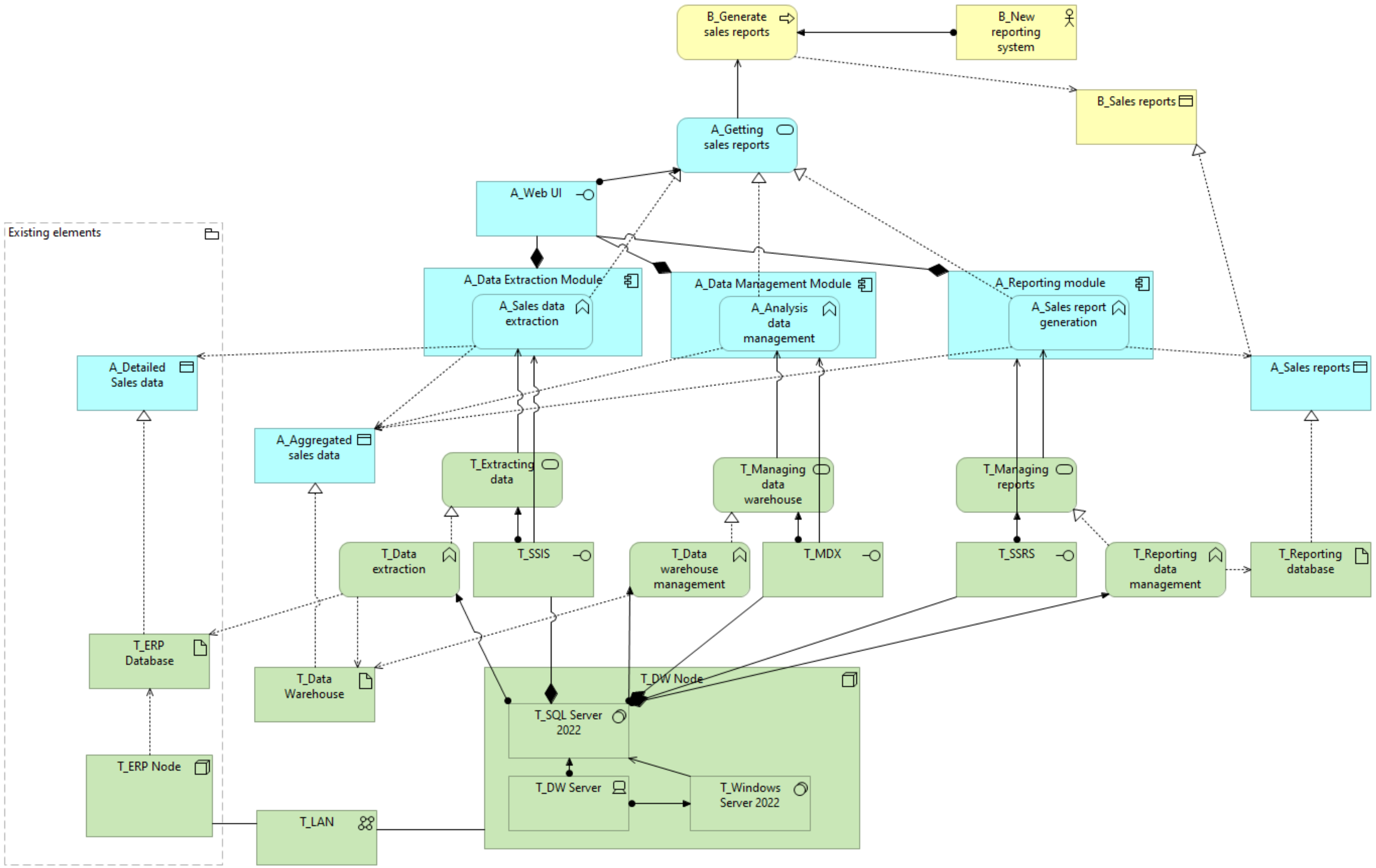
Exercise 1 – Existing ERP system



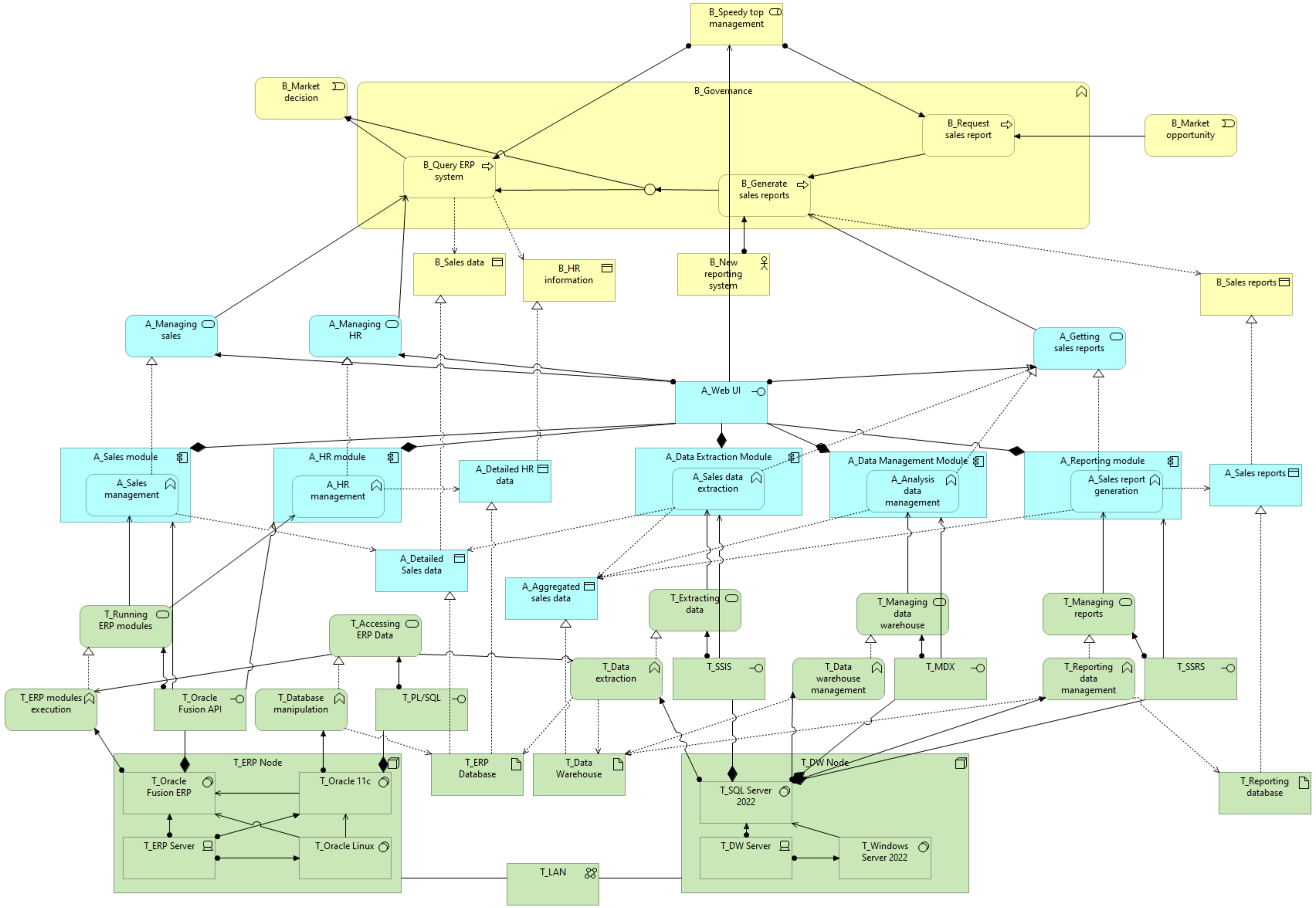
Exercise 1 – Business and Application layers



Exercise 1 – New infrastructure



Exercise 1 – Complete model



Exercise 2 – IC

IC is an insurance company which wants to offer a new insurance service for small objects (<2000$) managed completely online for reliable customers.

To achieve this, a customer who wants his assets to be insured has to provide its credentials and photo of the asset and its details (serial number, purchase date) to IC. To ensure that the customer is reliable and the asset inexpensive, IC will then check the customers credentials and past history and estimate the asset's price. If the checks succeed, a proposal will be generated. Otherwise, the request will be rejected.
To support the new service, IC needs to develop a new application with the following functionalities: proposal generation, price estimation, customer reliability check.

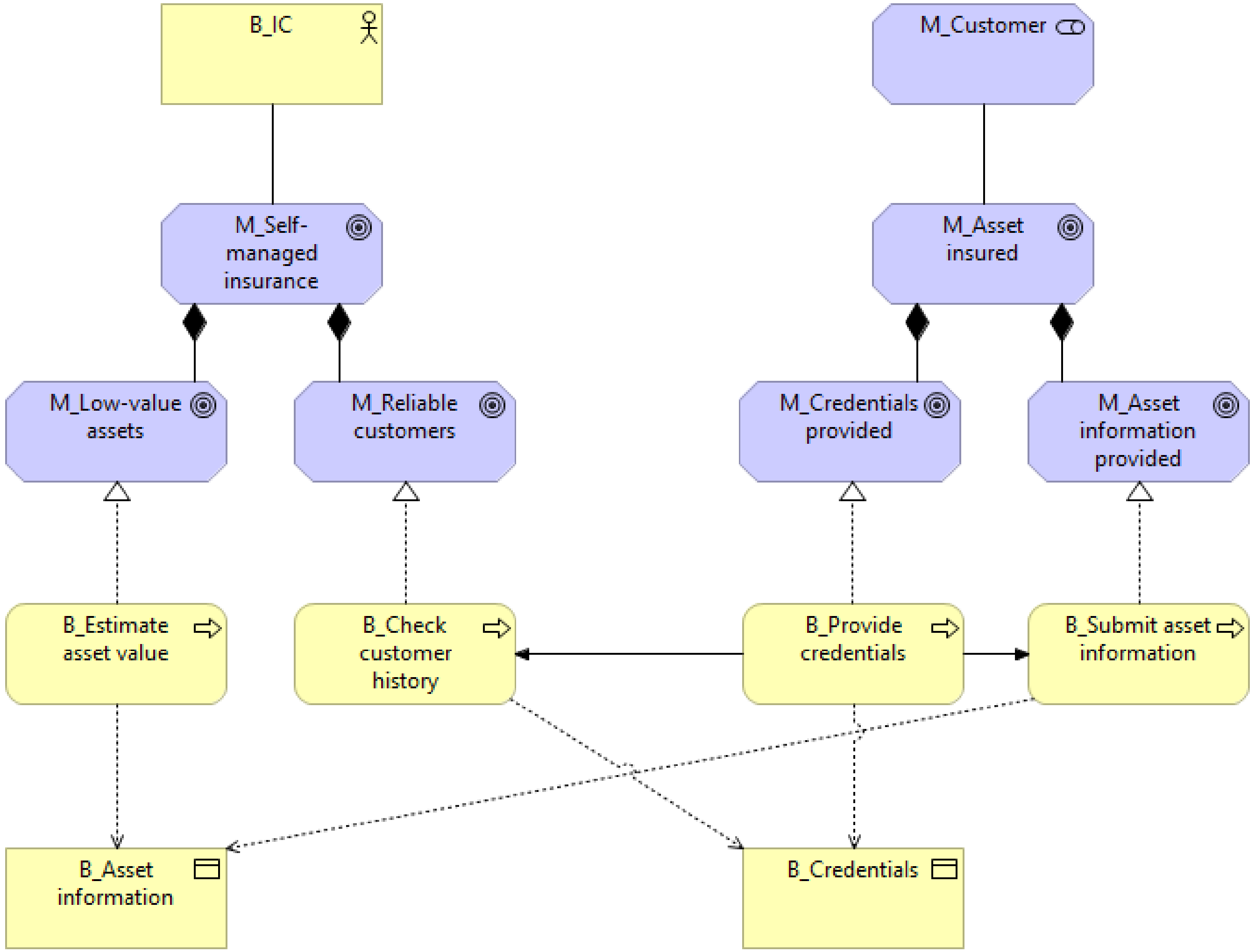
The new application will be implemented as a Java EE component running on Apache Tomcat. It will also rely on an operational database deployed on Oracle MySQL. Both the database and the application will reside on a dedicated server, which is connected to the corporate LAN.

To estimate an item price, an external service will be used.

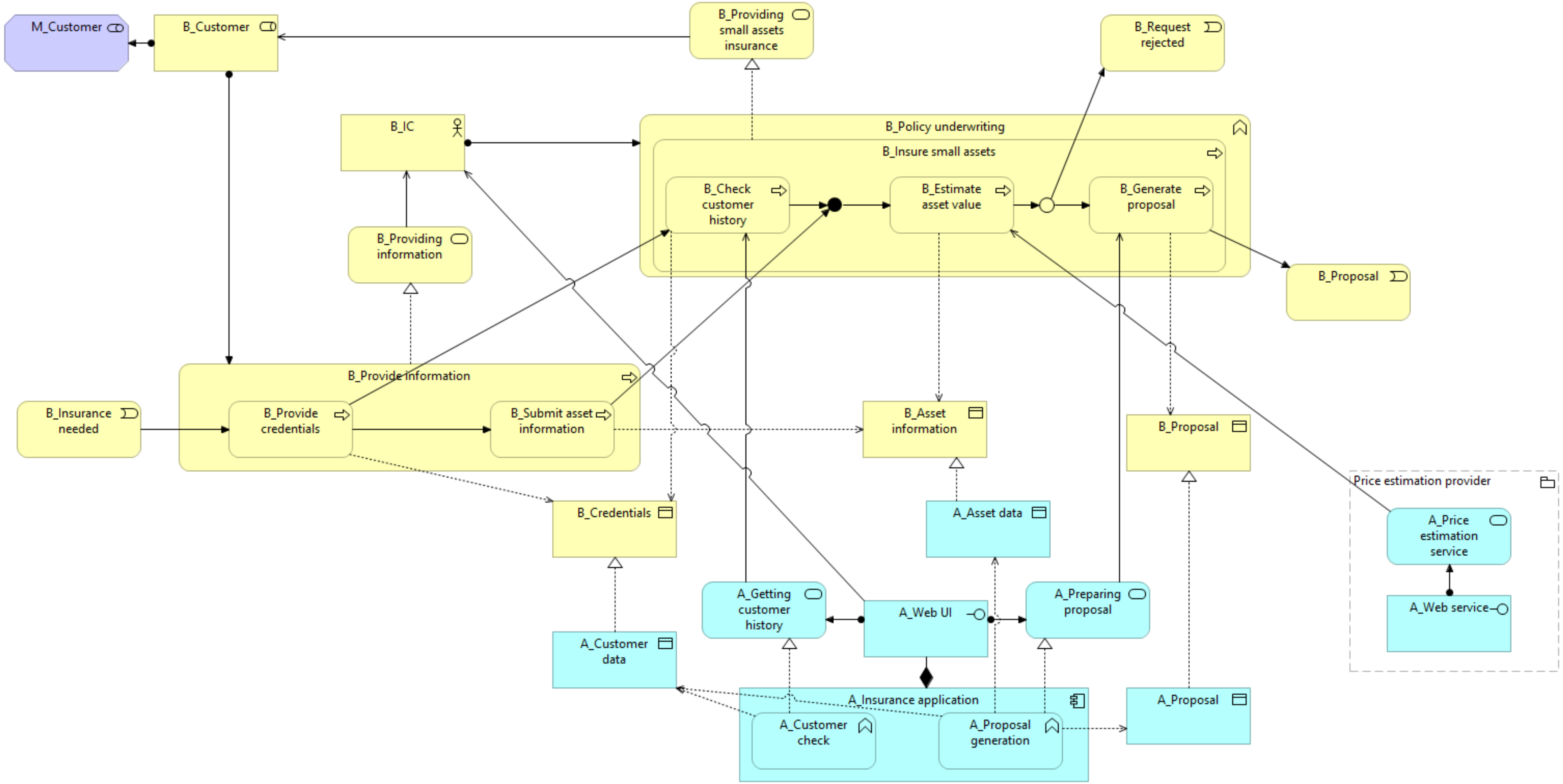
To check the customer reliability, the application needs data coming from the company's CRM, exposed by the CRM as a REST service and accessed through the corporate LAN.

The corporate LAN is separated from the public Internet by a firewall.
Model in ArchiMate the service provided by the company and its infrastructure.

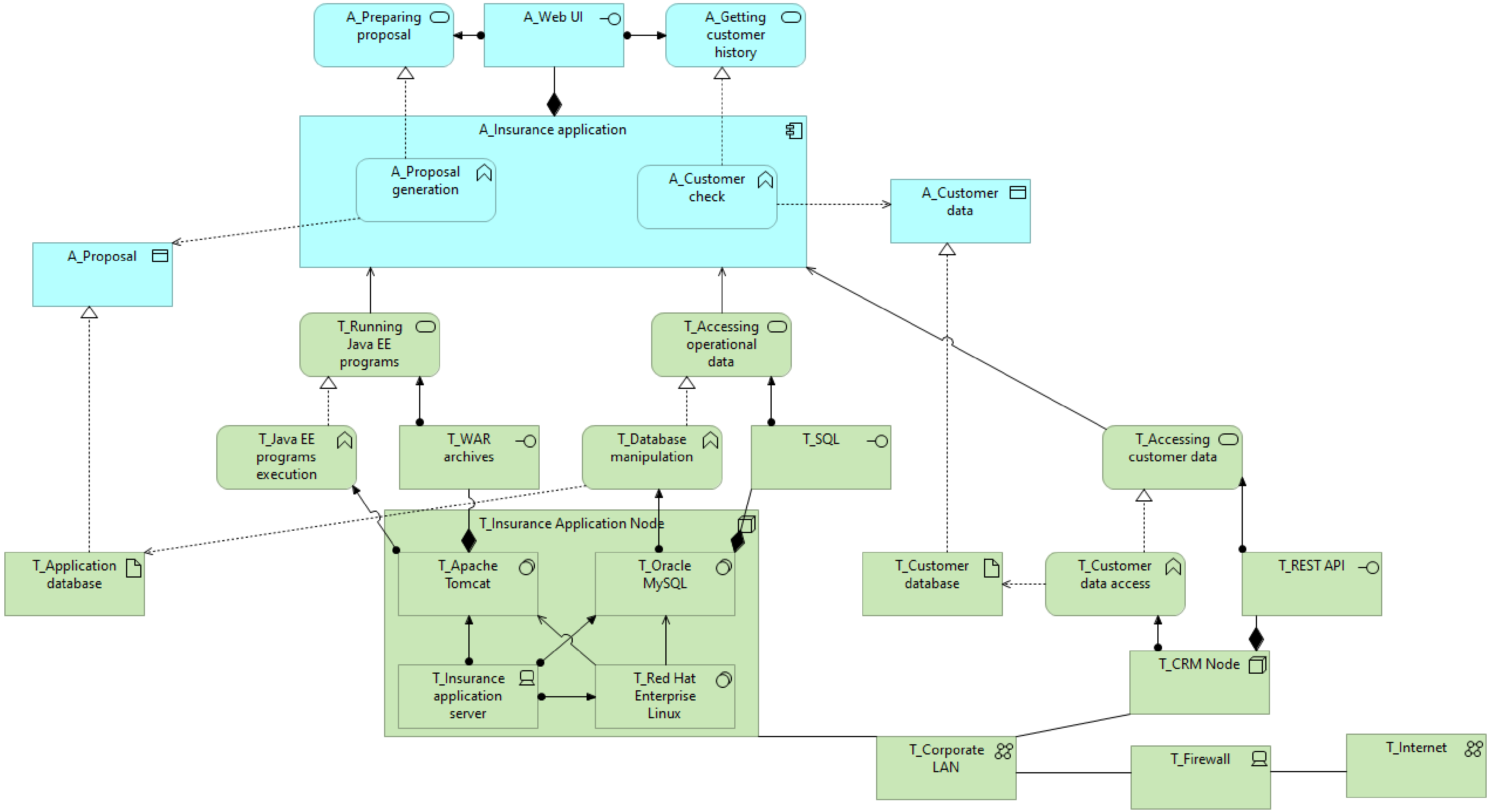
Exercise 2 – Motivation



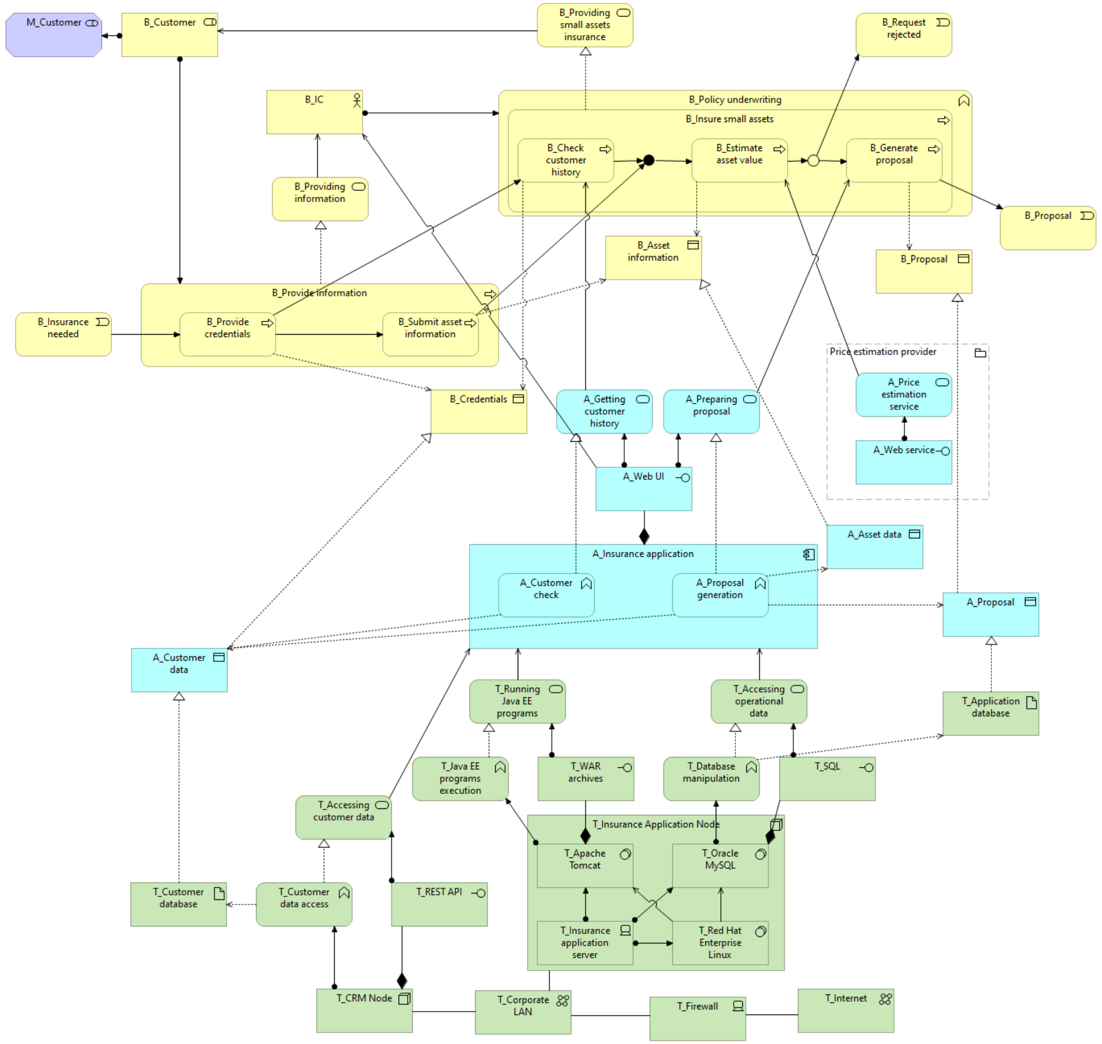
Exercise 2 – Business and Application layers



Exercise 2 – New infrastructure



Exercise 2 – Complete model



Exercise 3 – TEL

TEL is a telephony company interested in improving its customer experience. Currently, billing details are only accessible internally by TEL staff.

Data about telephony usage is stored on a Linux server running IBM DB2 UDB database, and it can only be accessed through SQL queries. Conversely, billing information is stored and handled by a legacy transactional CICS application running on an IBM mainframe.

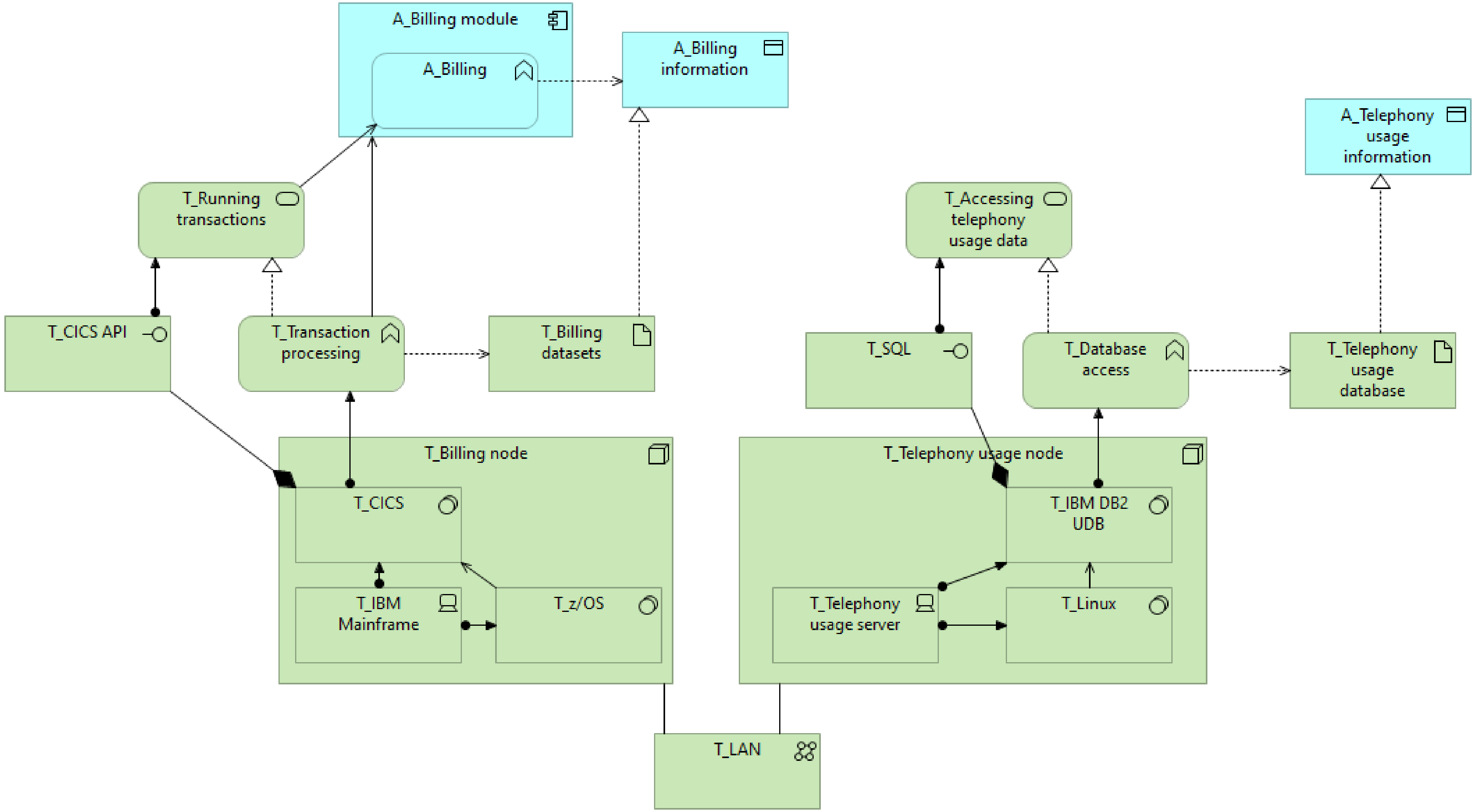
To improve its customer experience, TEL wants to develop a new web portal that can provide real-time billing details to users. In particular, the new portal will provide two main functionalities: inspect billing information and inspect usage information. The process enabled by the new portal will be organized as follows: the user logs in the portal, then he can select an item to inspect from the menu, and finally he can view the billing details in a dedicated page.

The new portal will be built for Microsoft SharePoint Online PaaS cloud service.

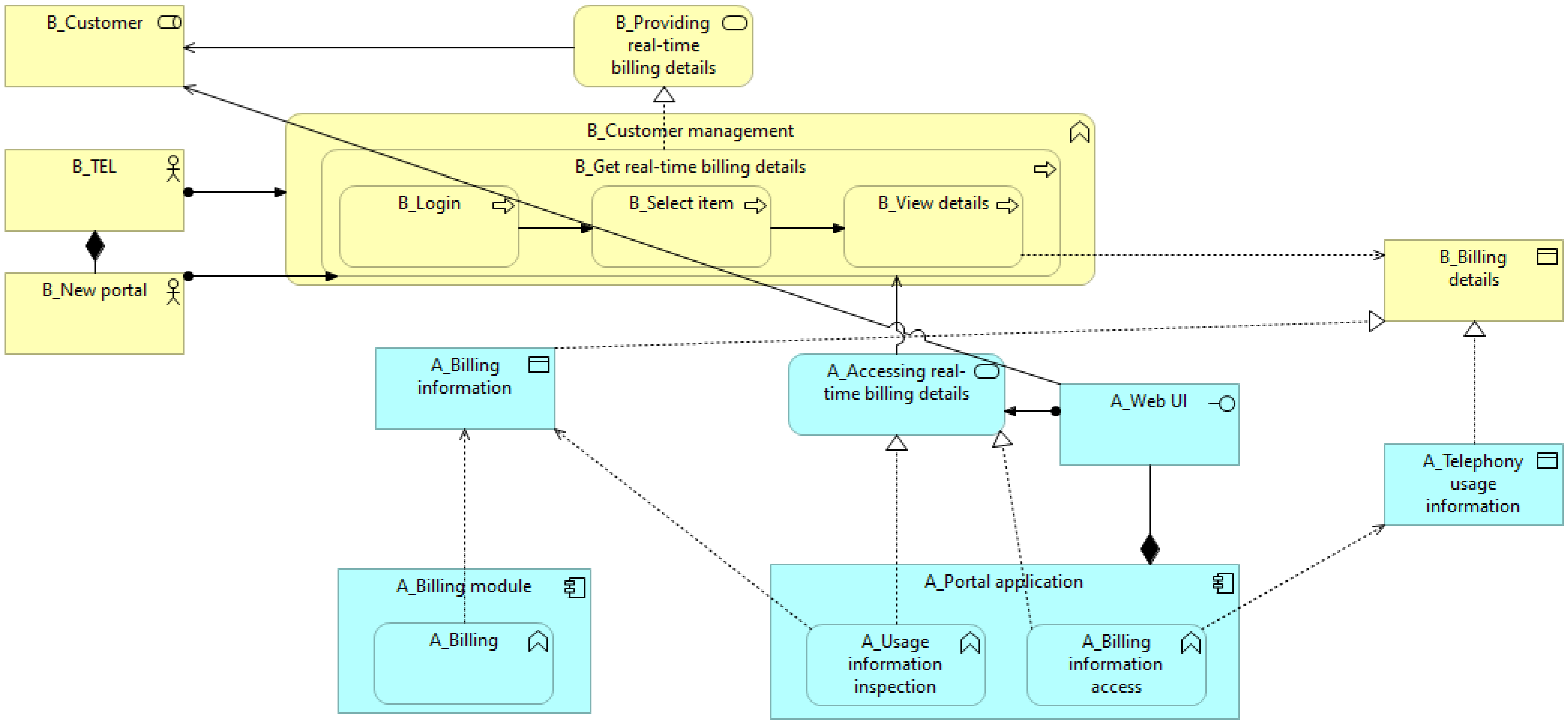
To access data from the existing infrastructure, a new node running Microsoft BizTalk Server 2020 middleware will also be introduced. BizTalk Server will offer a gateway service, making CICS applications and relational databases accessible through a standard REST interface.

An ArchiMate model of the existing system is enclosed in the next slide. Extend the model by adding the elements required for the new service.

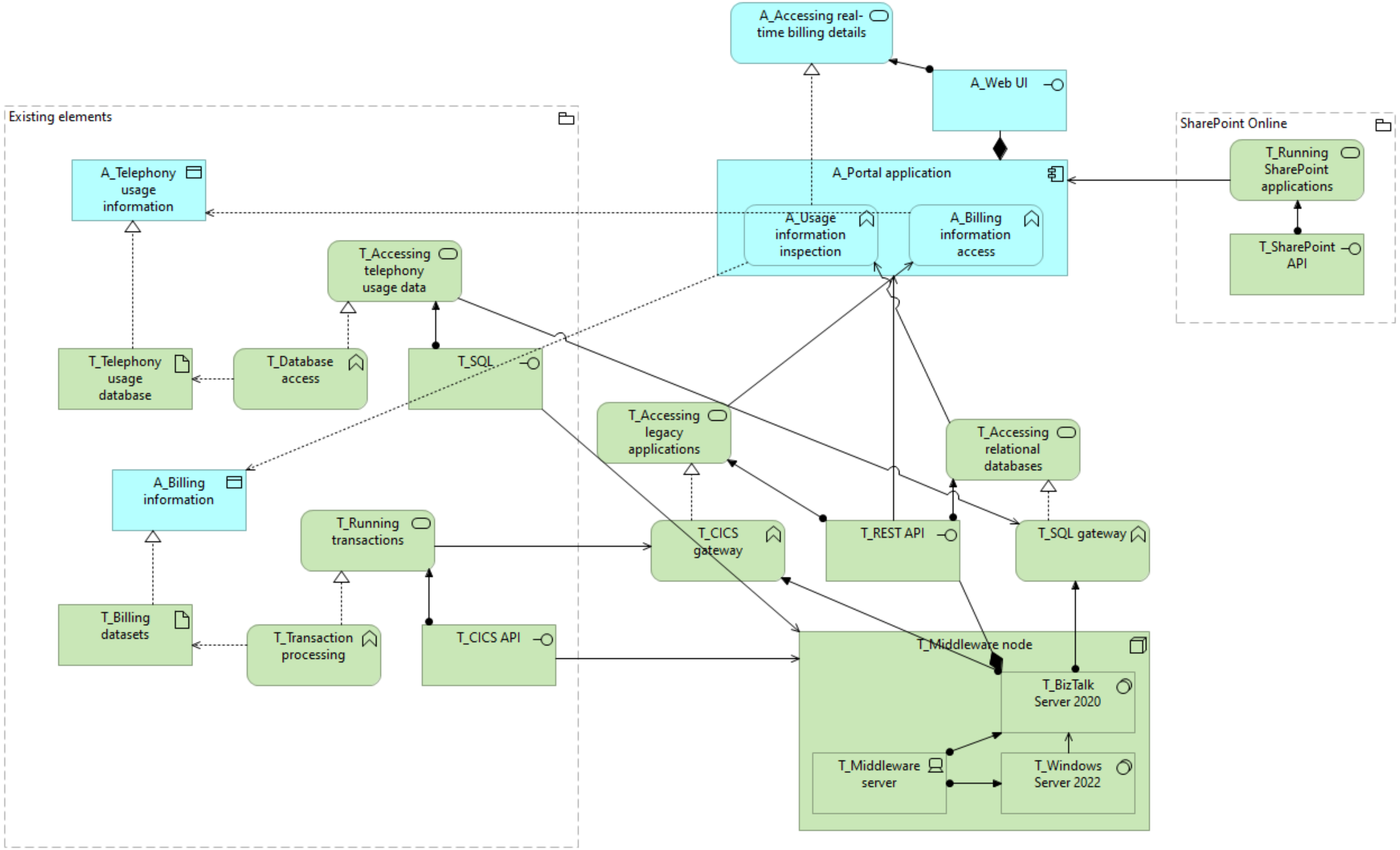
Exercise 3 – Existing infrastructure



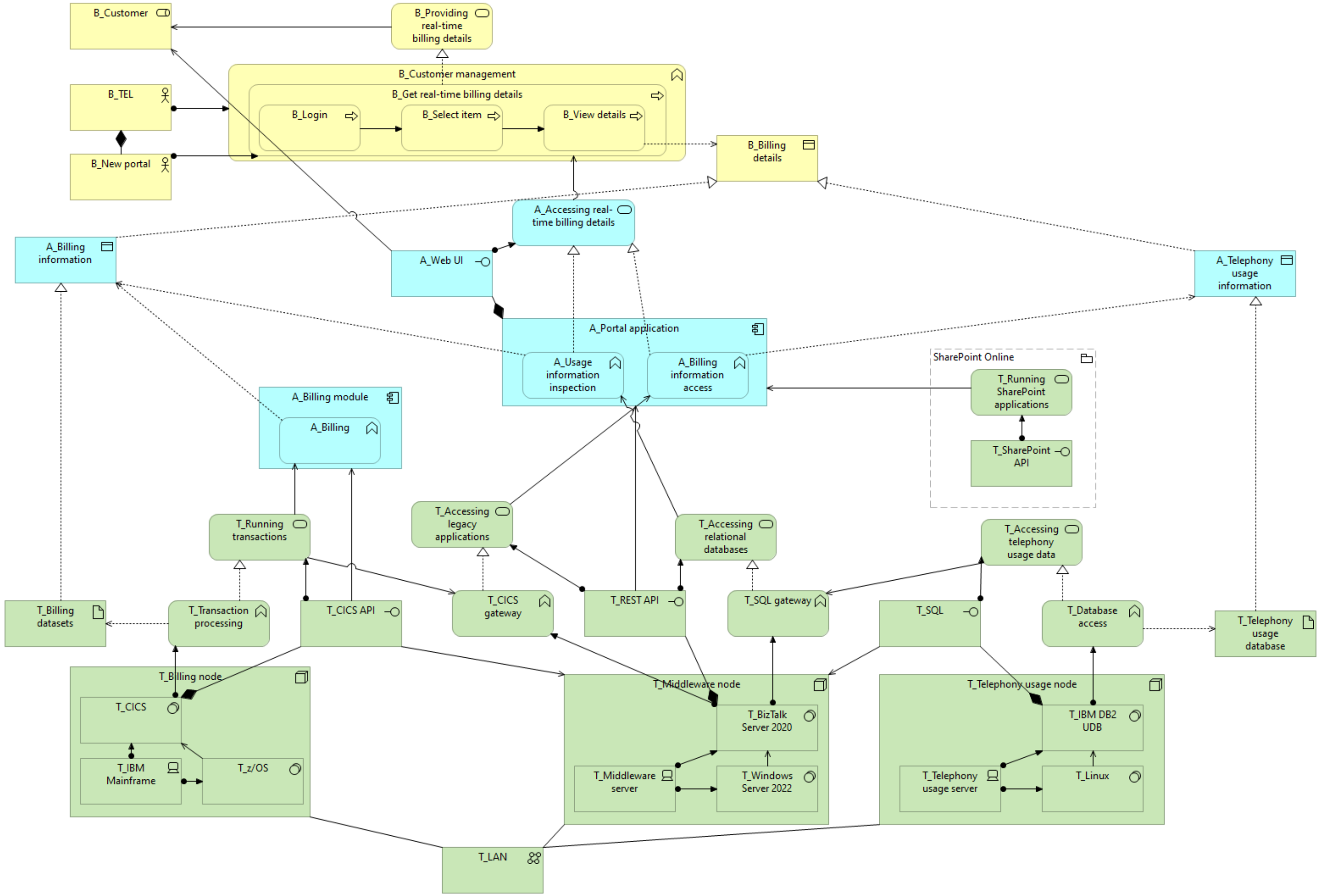
Exercise 3 – Business and Application layers


B_Customer
B_Providing real-time billing details
B_Customer management
B_TEL
B_Get real-time billing details
B_Login
B_Select item
B_View details
B_Billing details
B_New portal
A_Billing information
A_Accessing real-time billing details
A_Web UI
A_Telephony usage information
A_Billing module
A_Billing
A_Portal application
A_Usage information inspection
A_Billing information access

Exercise 3 – New infrastructure



Exercise 3 – Complete model



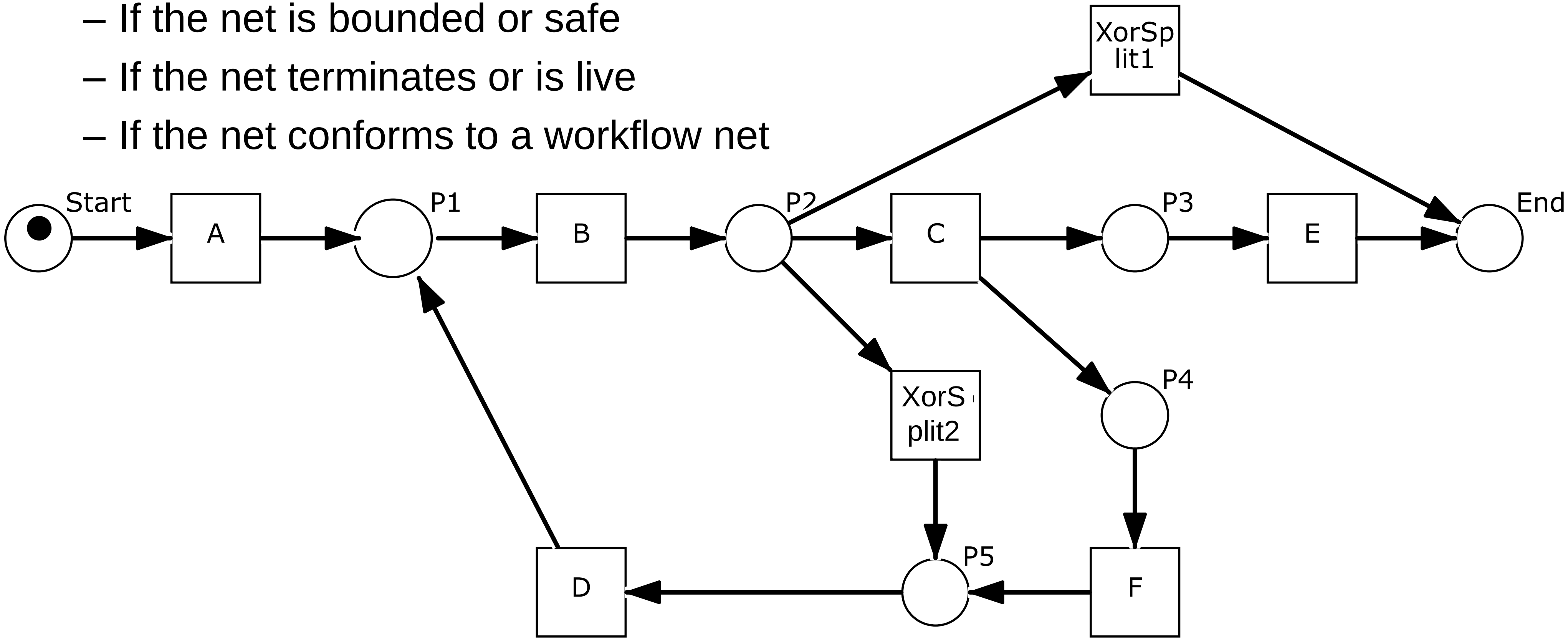
02291 System Integration

Behavioral Models with Petri Nets: Solutions to exercises

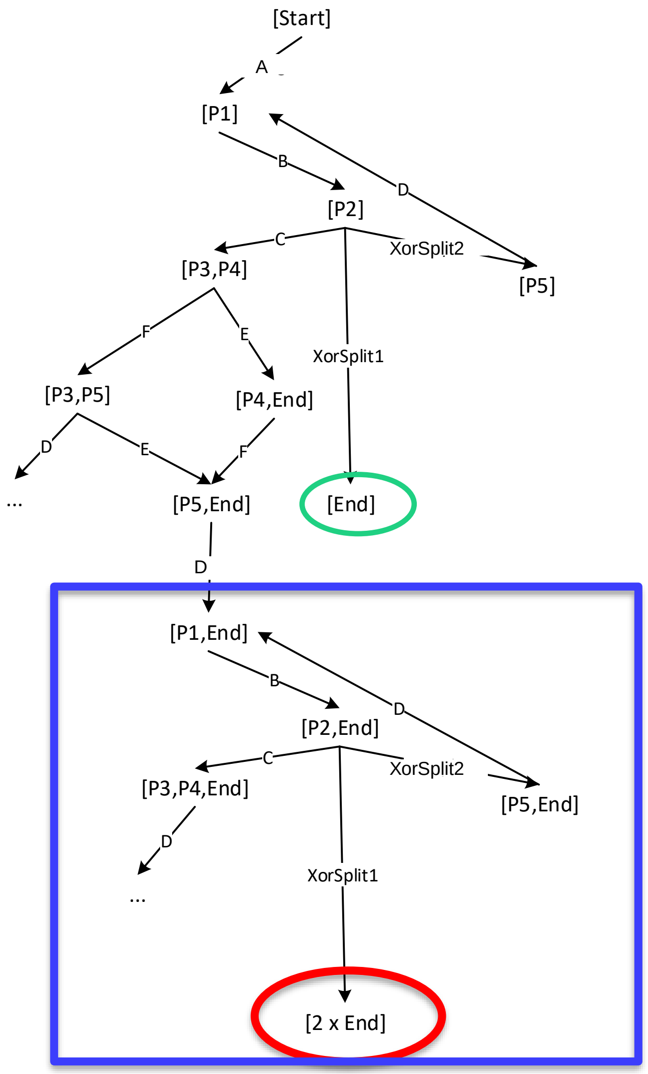
© Giovanni Meroni

Exercise 1

- Given the following Petri Net, compute:
  - – P, T, F sets
  - – The reachability graph
  - – If the net is bounded or safe
  - – If the net terminates or is live
  - – If the net conforms to a workflow net



Exercise 1 – Reachability graph



- P = {Start, P1, P2, P3, P4, P5, End}
- T = {A, B, C, D, E, F, XorSplit1, XorSplit2}
- F = {(Start,A), (A,P1), (P1,B), (B,P2), (P2,XorSplit1), (P2,C), (P2,XorSplit2), (XorSplit1,End), (C,P3), (C,P4), (P3,E), (E,End), (P4,F), (F,P5), (P5,D), (D,P1), (XorSplit2,P5)}

Infinite reachability graph (tokens always grow)

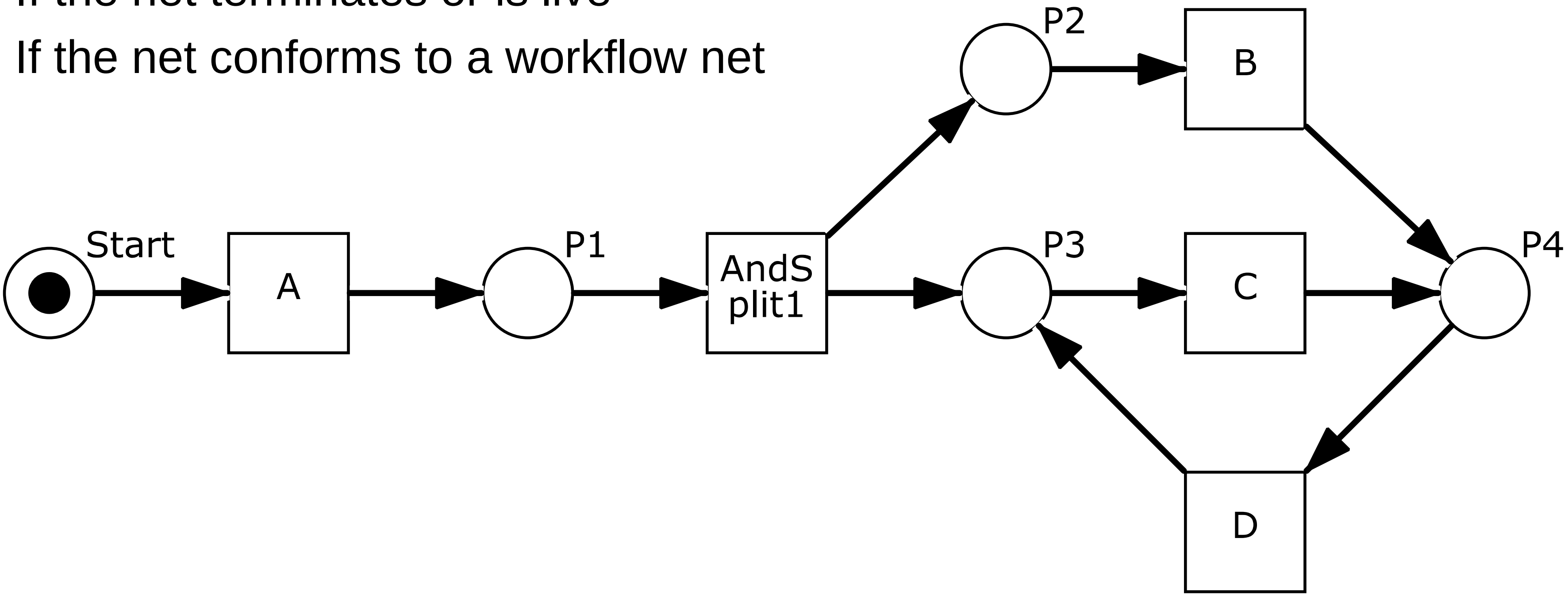
Not safe (not 1-bounded)

Not live (there are final markings)

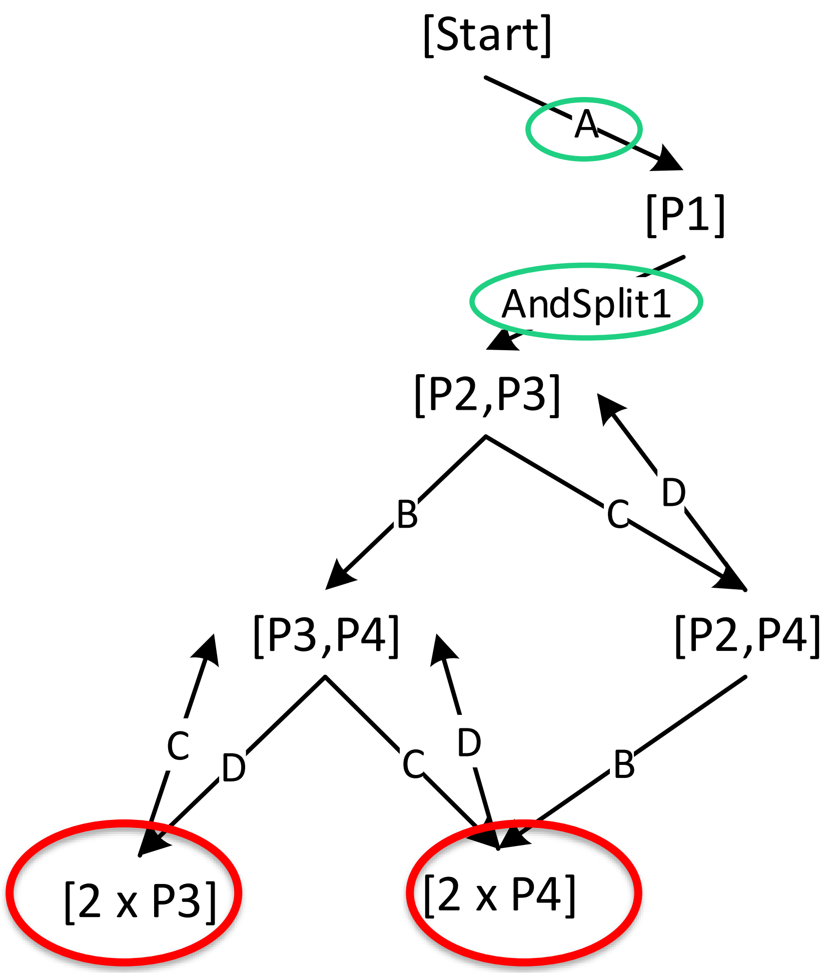
Net conforms to a workflow net: only one input (Start) and one output (End) place, every node is in a path from Start to End

Exercise 2

- Given the following Petri Net, compute:
  - P, T, F sets
  - The reachability graph
  - If the net is bounded or safe
  - If the net terminates or is live
  - If the net conforms to a workflow net



Exercise 2 – Reachability graph



- P = {Start, P1, P2, P3, P4}
- T = {A, AndSplit1, B, C, D}
- F = {(Start,A), (A,P1), (P1,AndSplit1), (AndSplit1,P2), (AndSplit1,P3), (P2,B) (B,P4), (P3,C), (C,P4), (P4,D), (D, P3)}

Not safe (2-bounded), cannot become sound

Not live (we cannot fire A or AndSplit1 more than once)

No final place, hence not terminating (and not conforming to a workflow net)!

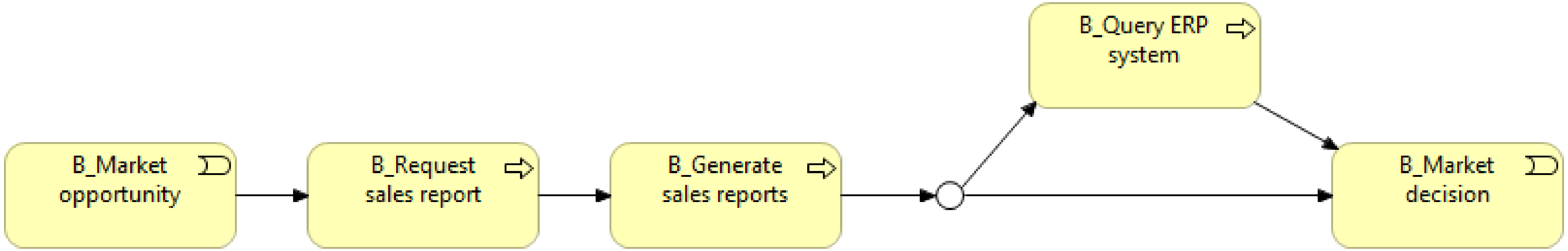
Exercise 3

Speedy is a delivery company that wants to create a new reporting system for the top management. After inspecting the sales reports, the top management may also need to query the existing ERP system, based on Oracle Fusion, to get detailed sales and HR information.

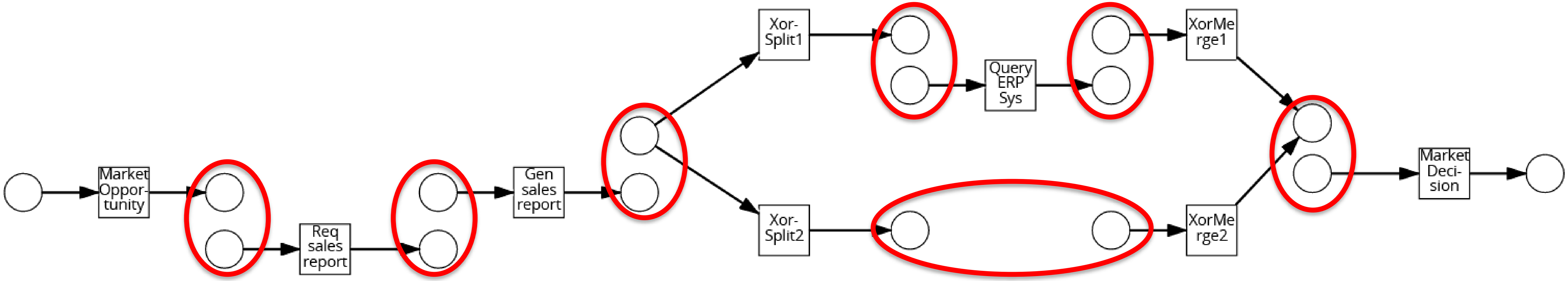
An ArchiMate model representing the process is enclosed below.

Starting from this model, create an equivalent Petri Net and try to answer the following questions:

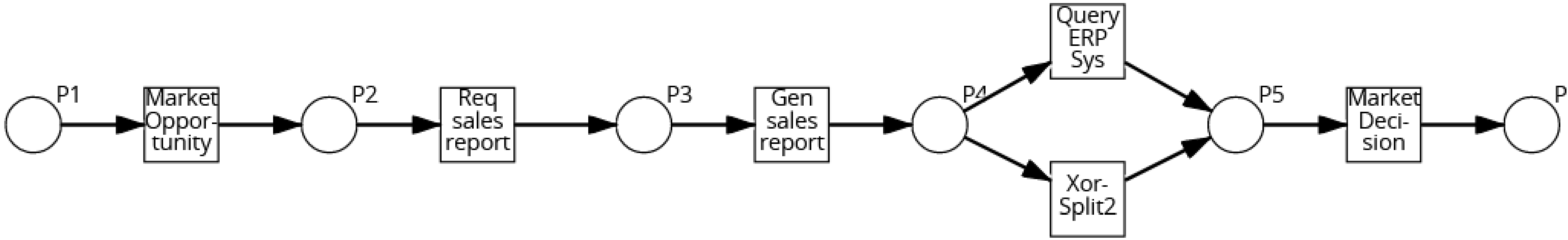
- What is the reachability graph of the net?
- Is the net sound?



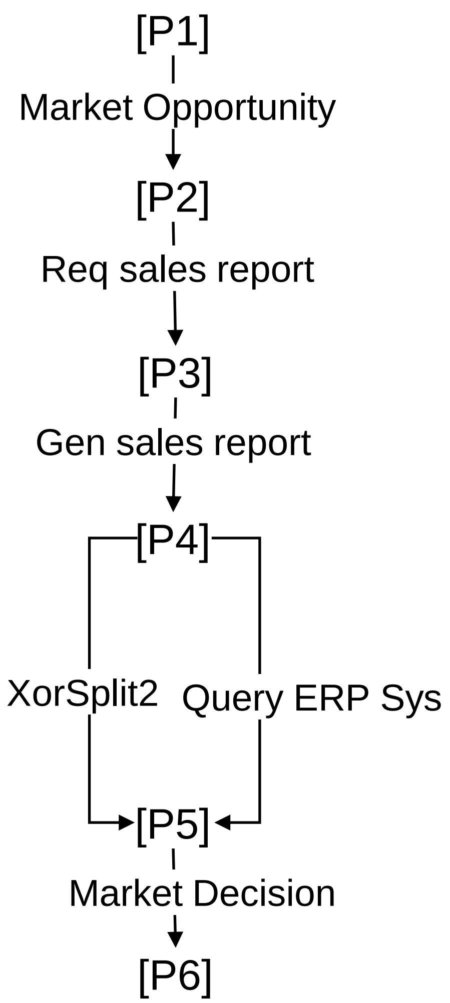
Exercise 3 – Petri Net



Exercise 3 – Optimized Petri Net



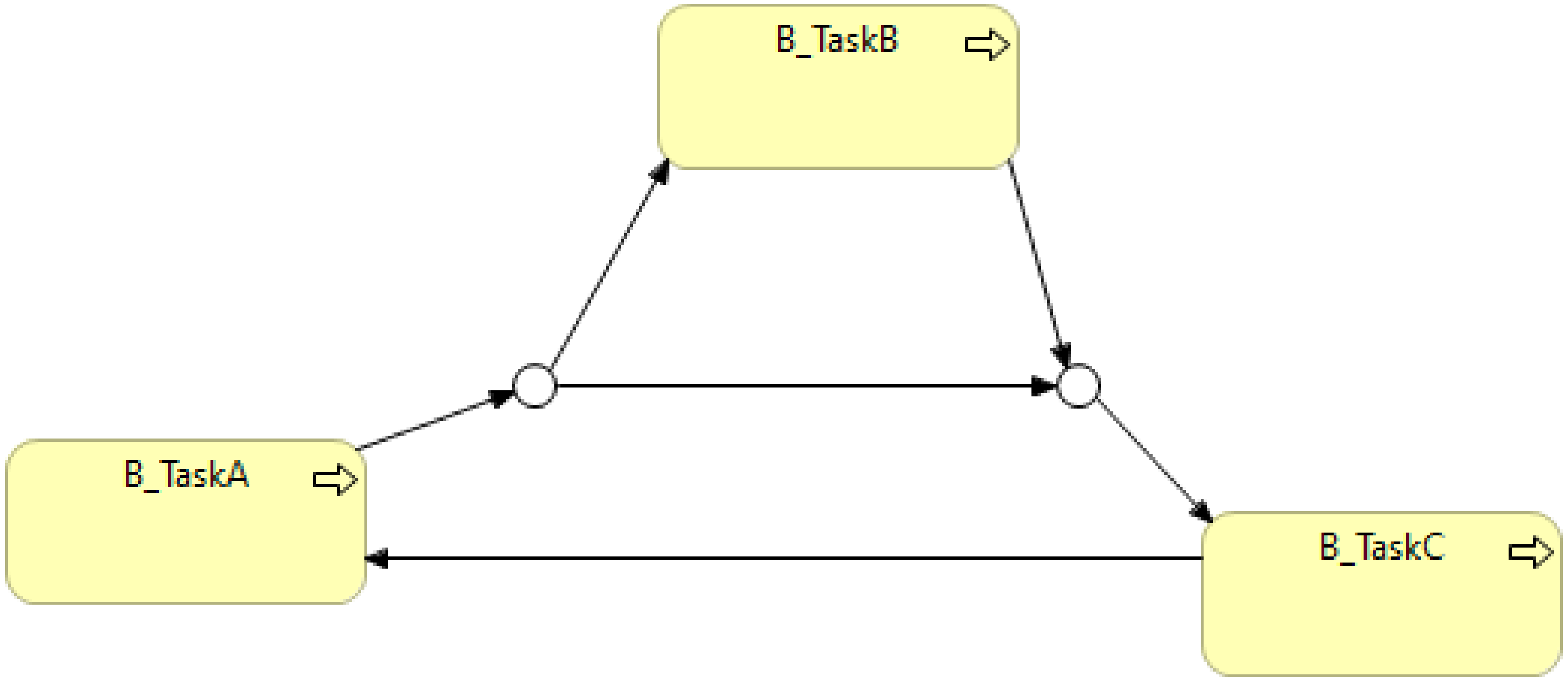
Exercise 3 – Reachability graph



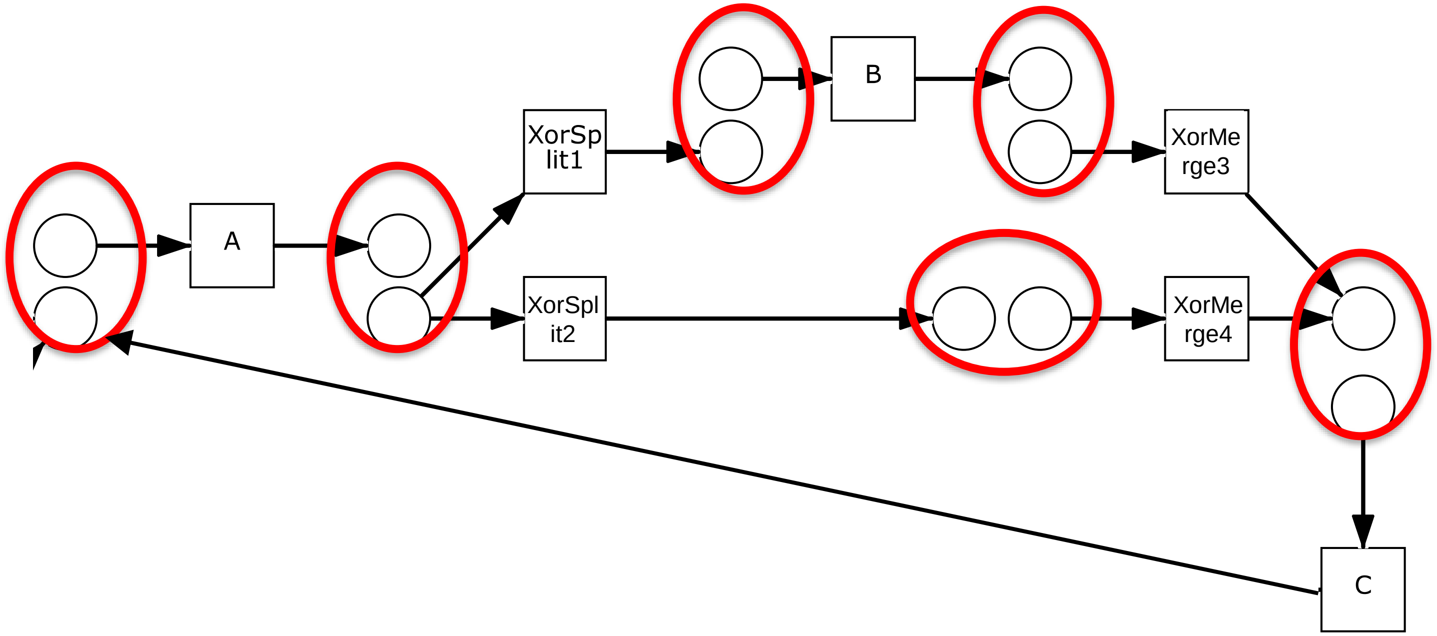
Net conforms to a workflow net: only one input (P1) and one output (P6) place, every node is in a path from P1 to P6

Exercise 4

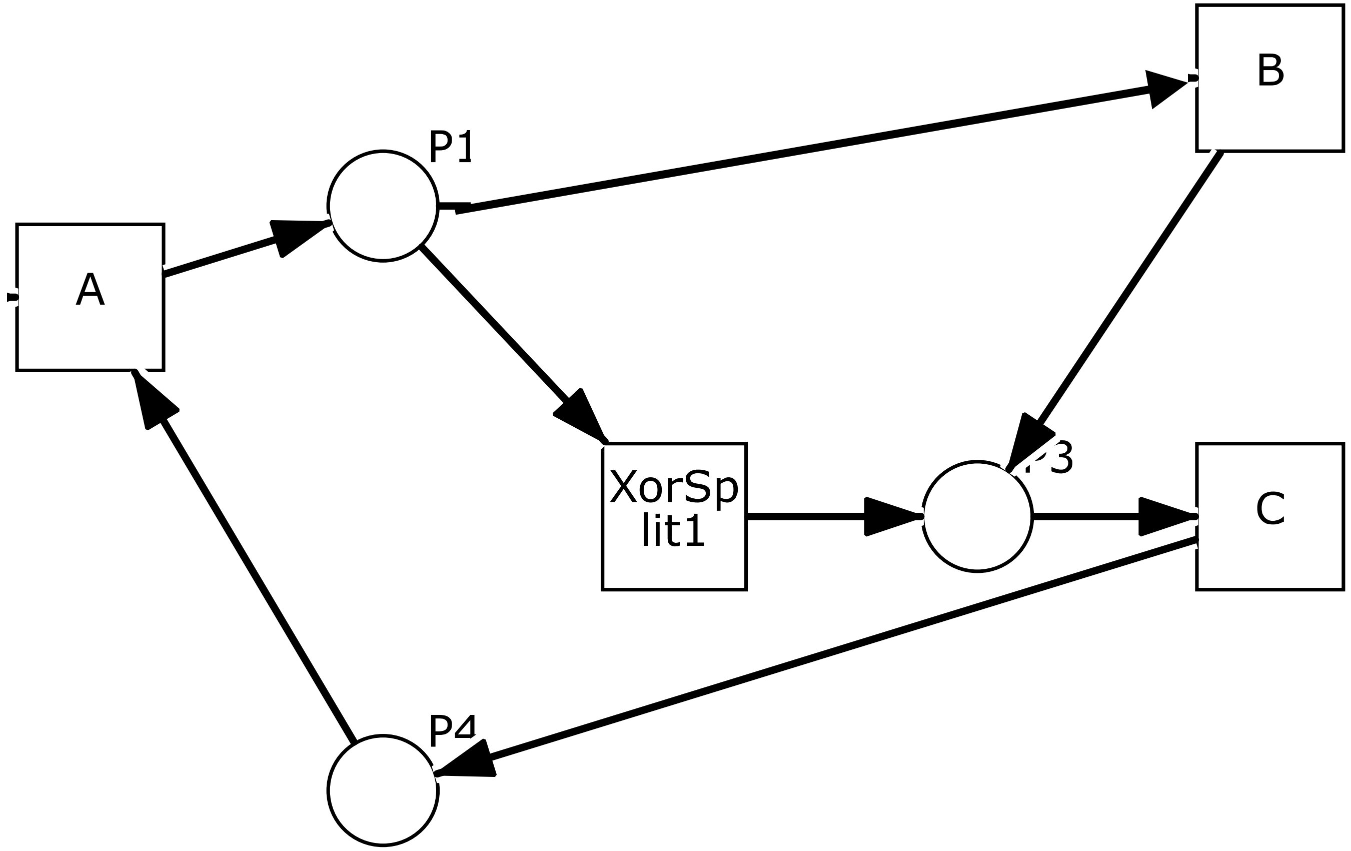
- Translate the following process model into a Petri Net.
- Check if the resulting Petri Net is sound. If not, propose an action to repair the process, to make it sound.



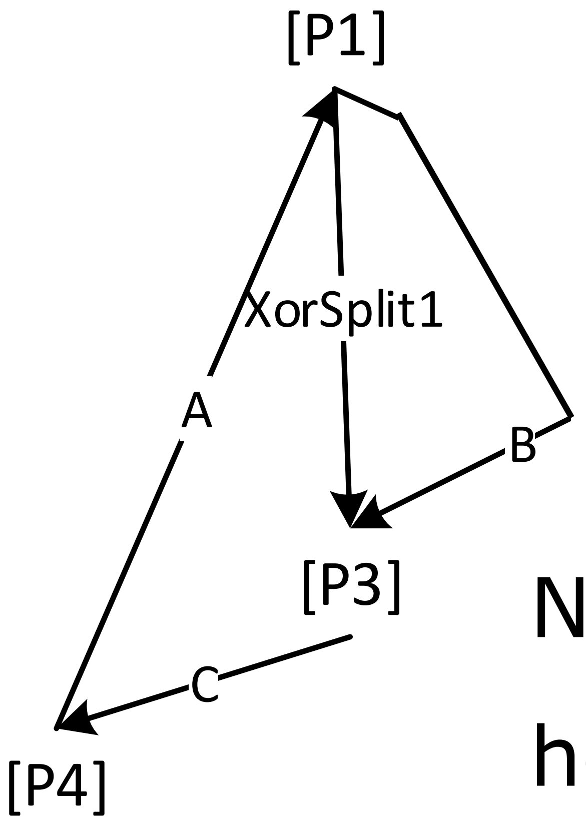
Exercise 4 – Petri Net



Exercise 4 – Optimized Petri Net



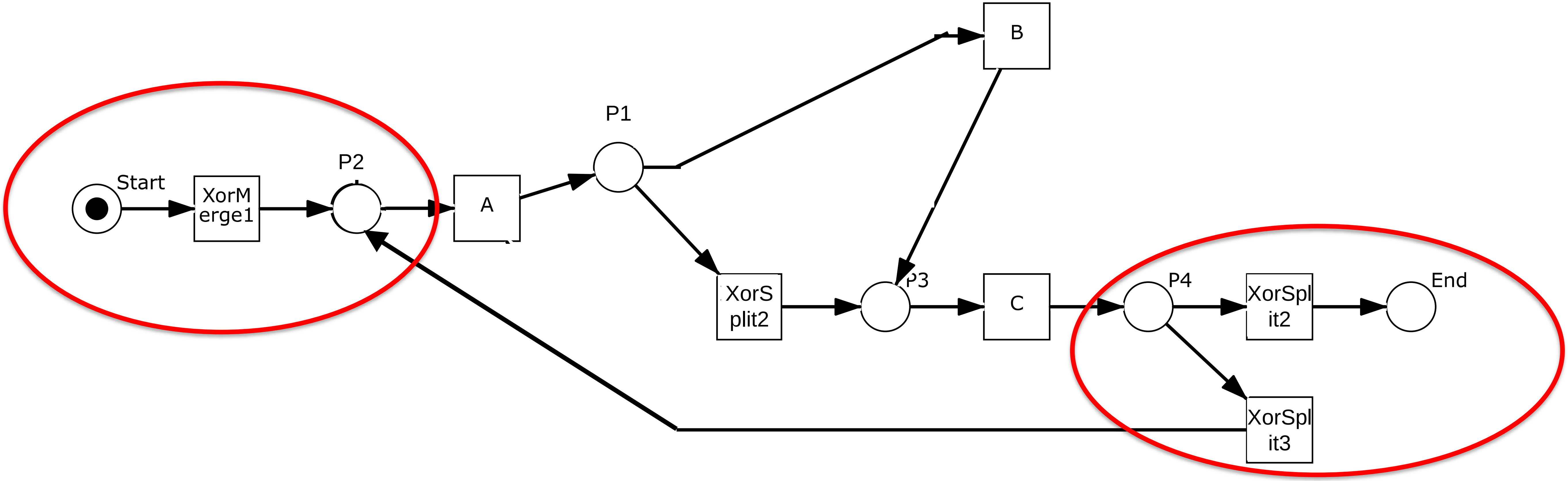
Exercise 4 – Reachability graph



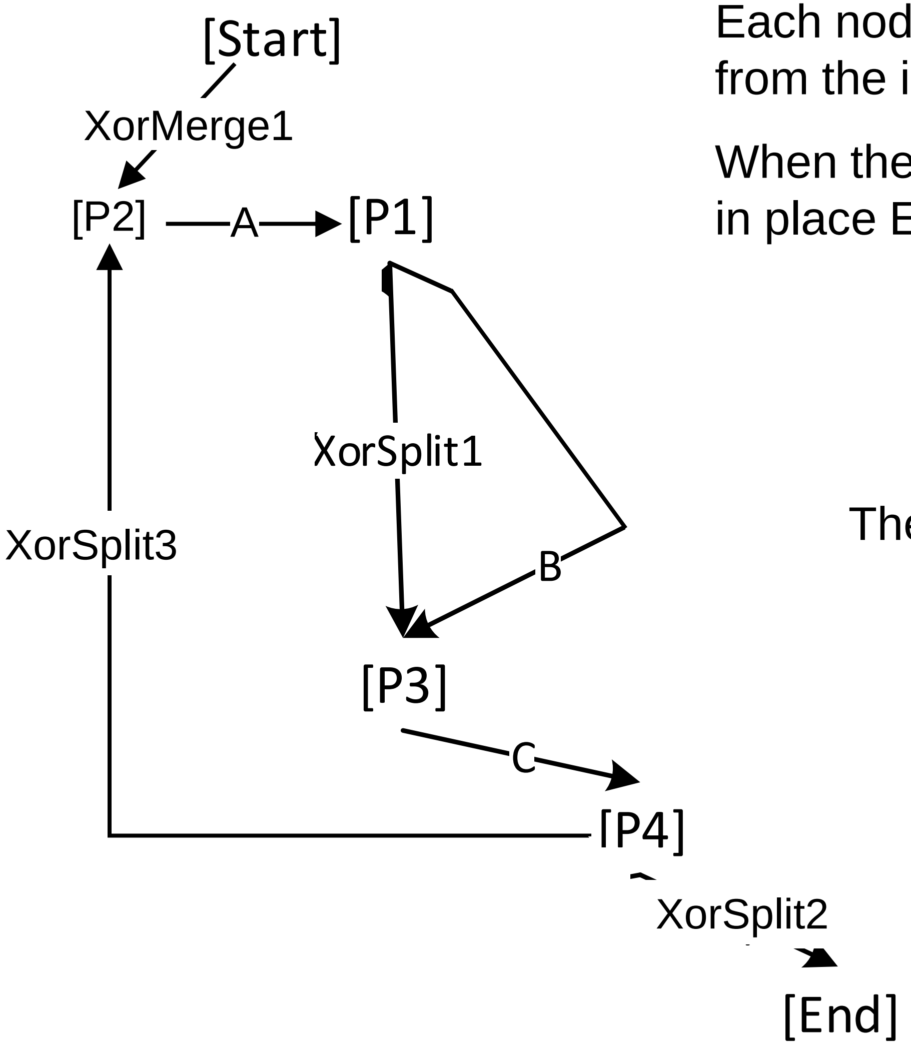
No initial and no final place,

hence not sound!

Exercise 4 – Repaired process



Exercise 4 – Reachability graph of repaired process



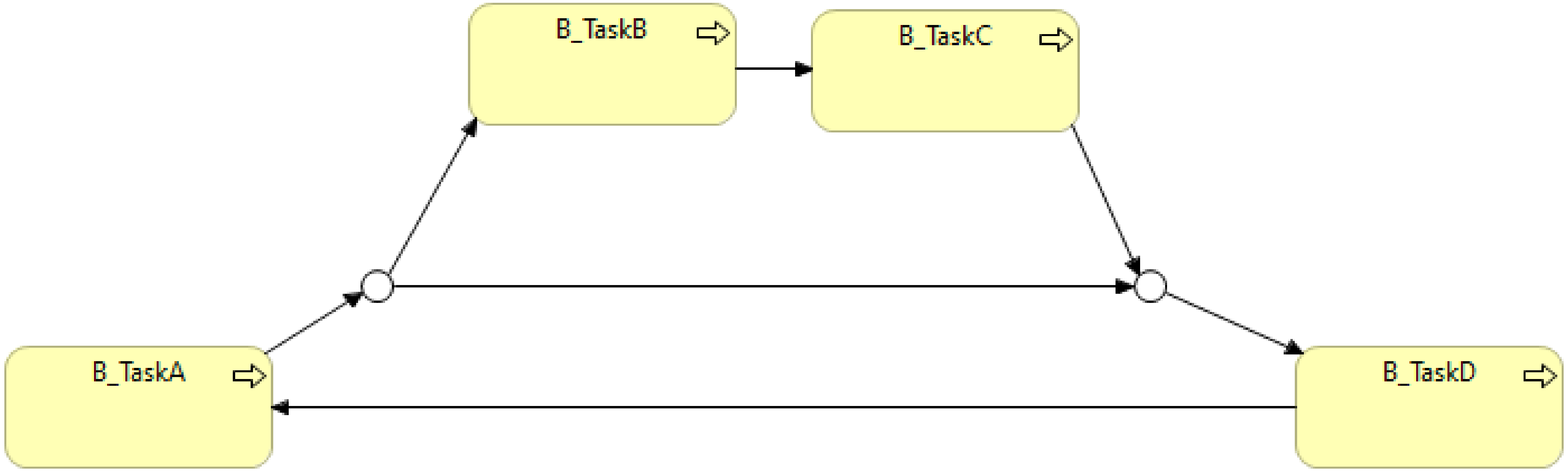
Each node in the process model is on a path from the initial node to the final node

When the procedure terminates, there is a token in place End and all the other places are empty

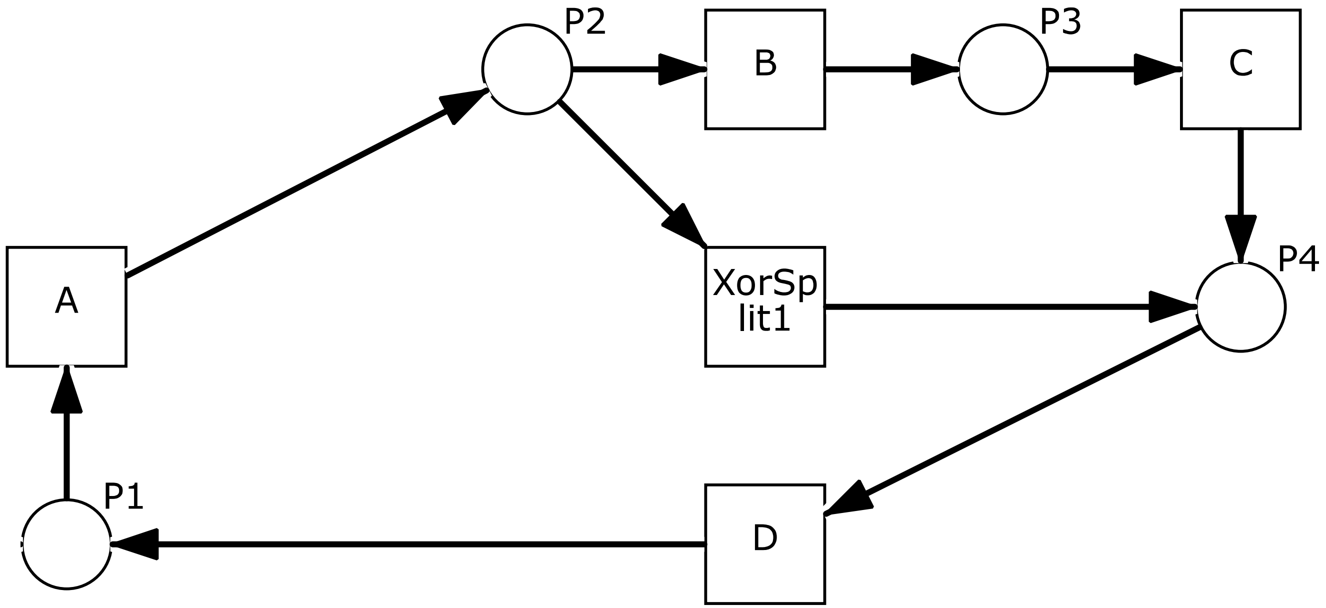
The process is now sound!

Exercise 5

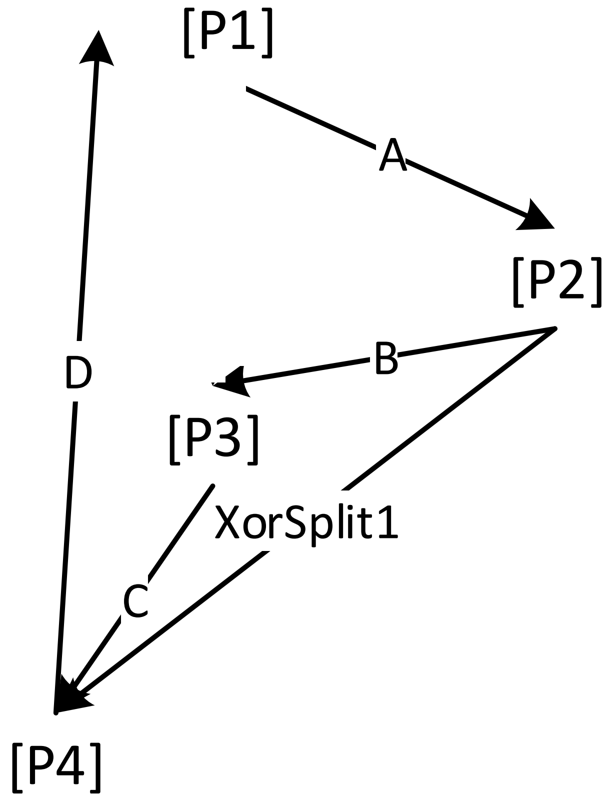
- Translate the following process model into a Petri Net.
- Check if the resulting Petri Net is sound. If not, propose an action to repair the process, to make it sound.



Exercise 5 – Optimized Petri Net



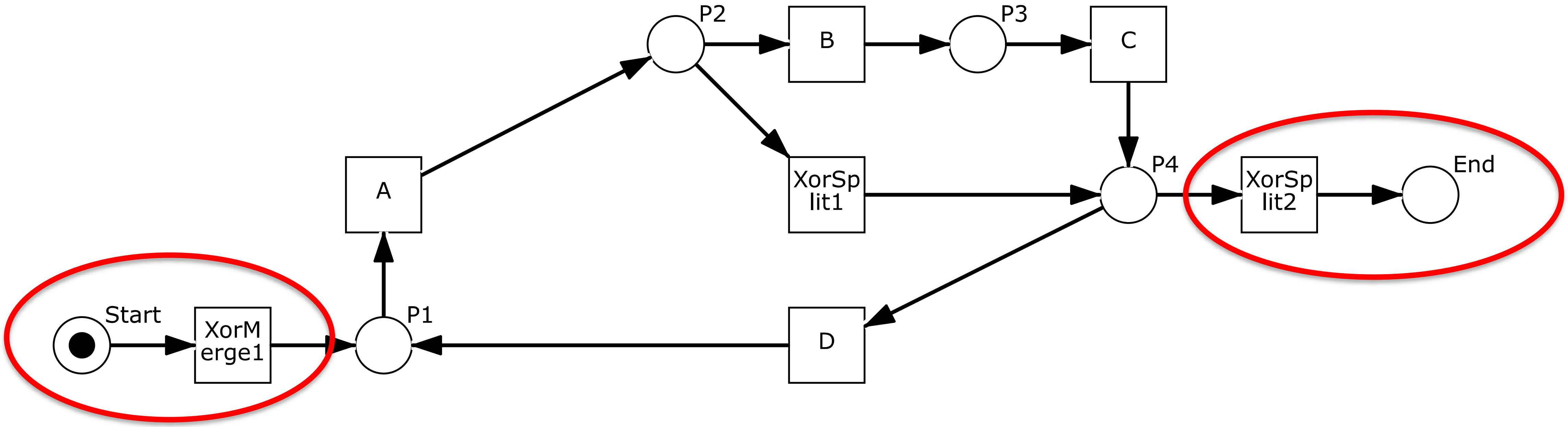
Exercise 5 – Reachability graph



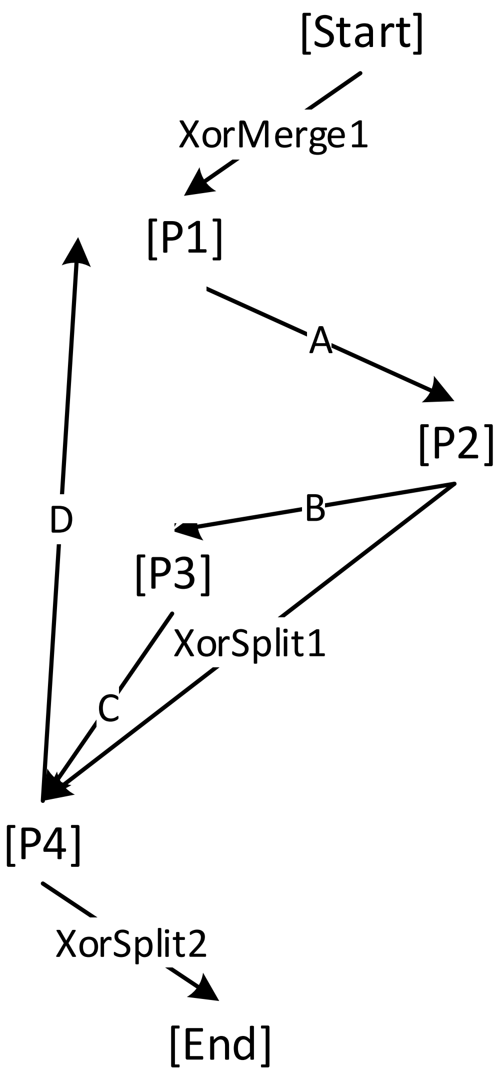
No initial and no final place,

hence not sound!

Exercise 5 – Repaired process



Exercise 5 – Reachability graph of repaired process



Each node in the process model is on a path from the initial node to the final node

When the procedure terminates, there is a token in place End and all the other places are empty

The process is now sound!

Exercise (in groups): vending machine 2.0

Use CCS to model a vending machine selling apples or chocolate bars

The system has 3 processes: a **user**, a **selector**, and a **dispenser**

- The **selector** accepts 2 types of coins from **user**: *coin10* (10 kr) and *coin20* (20 kr)
    - when the **selector** collects 10 kr., it lets the **user** select *apple*, or insert 10 kr. more
        - if the **user** selects *apple*, the **selector** tells the **dispenser** to *giveApple*
    - when the **selector** collects 20 kr., it lets the **user** select *choc*
        - if the **user** selects *choc*, the **selector** tells the **dispenser** to *giveChoc*
- The **dispenser** waits for the **selector** to tell either:
    - *giveApple* — in this case, the **dispenser** ouputs *takeApple*
    - *giveChoc* — in this case, the **dispenser** ouputs *takeChoc*
- The **user** may either:
    - insert 10 kr. in the **selector**, select *apple*, and then *takeApple* from the **dispenser**
    - insert 20 kr. in the **selector**, select *choc*, and then *takeChoc* from the **dispenser**

Optional tasks: can you hide *giveApple* and *giveChoc* from the **user**? Can you extend the system to **give change**, and then to sell a **bagel** for 30 kr.?

Exercise 2

Model the system below in CCS. **email your solution to** `hulo@dtu.dk` if you have questions.

The system has 3 processes: an **online shop**, a **warehouse**, and a **customer**

- The **shop** allows the **customer** to choose between two items: *jeans* or *shoes*
- When the **customer** asks to buy an item, the **shop** asks the **warehouse** whether the item is available, by sending *jeansAvailable* or *shoesAvailable*
- The **warehouse** may answer *yes* or *no*:
  - if the **warehouse** says *yes*, the **shop** says *available* to the **customer**, then asks for a *shippingAddress*
  - if the **warehouse** says *no*, the **shop** says *unavailable* to the **customer**

Use Pseuco's CCS Doctor tab to generate the inference tree for the case the **customer** receives the *unavailable* message

Optional: extend the system to support payments with an additional **bank** process:

- after *shippingAddress*, the **shop** also asks a *creditCardNumber* to the **customer**
- then, the **shop** requests a *payment* to the **bank**, which can answer either *accepted*

Exercise 3

Model the system below in Value-passing CCS. **email to** `hulo@dtu.dk` if you have questions.

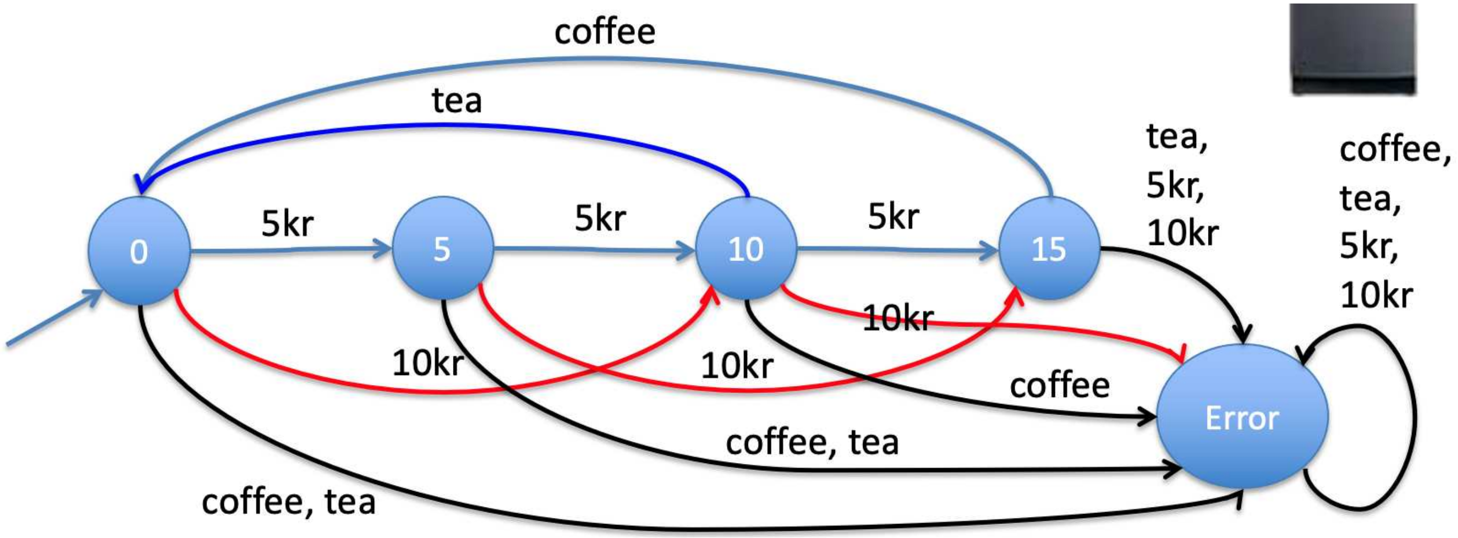
IC is an insurance company which wants to offer a new insurance service for small objects ($\leq 2.000\mathrm{Eur}$) managed completely online for reliable customers. To achieve this, a **customer** who wants his assets to be insured has to provide its *credentials* and the *cost* of the asset with its details (*serial number*, *purchase date*) to **IC**. To ensure that the customer is reliable and the asset inexpensive, IC will then check the customers credentials and past history and estimate the asset's price. If the checks *succeed*, a proposal will be sent back. Otherwise, the request will be *rejected*.

Exercise

- Follow the tutorial "A Tutorial on Uppaal"
  Gerd Behrmann, Alexandre David, and Kim G. Larsen, and implement the models in sections 4 and 5.

Exercise

- Implement the following state machine in uppaal, together with a corresponding set of users (that like tea or coffee)
- Make sure that the vending machine returns money after a timeout of 30 seconds
- Verify that for a multiple user can eventually be served by the vending machine (fairness)



Exercise

- Consider the interaction model you handed in
  - Discuss with your group the type of timing constraints you may have in your model
    - And if you can, implement the model in Uppaal
  - Discuss with your group the type of model-checking properties you may be interested
    - And If you manage to implement the models, try to verify the new properties

Declarative Process Modelling - Solutions Week 7

Hugo A. López

Exercises

Model the following patterns in a DCR-graph (use the simulator to verify if your model is right):

1. **Direct Precedence of a Task.** “Every time B occurs, it should be directly preceded by A.” If B occurs without a directly preceding A, the rule is violated. For instance, traces ⟨ACCAAC⟩ and ⟨ABCAAB⟩ comply to the rule, whereas ⟨ABACB⟩ violates the rule.

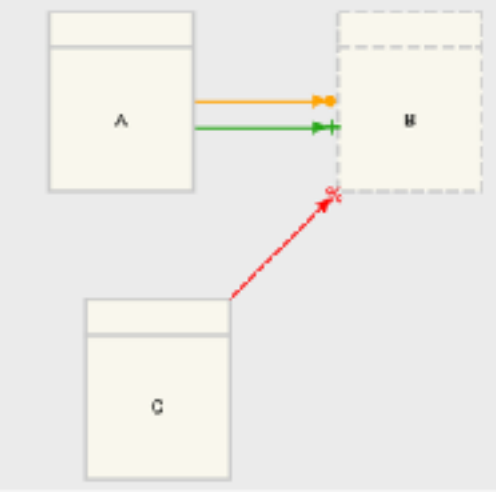
2. **Direct Precedence or Simultaneous Occurrences of Tasks.** “Task *A* must always be executed simultaneously or directly before task *B*.” (hint: consider A/B as non-atomic tasks)

3. **Bounded Existence of a Task:** Task A should be executed exactly k times.” If A occurs less than or more than k times, the rule is violated. For instance, for k = 2, the trace ⟨BCADBCAD⟩ complies to this rule and ⟨BCADBCAAD⟩ violates the rule.

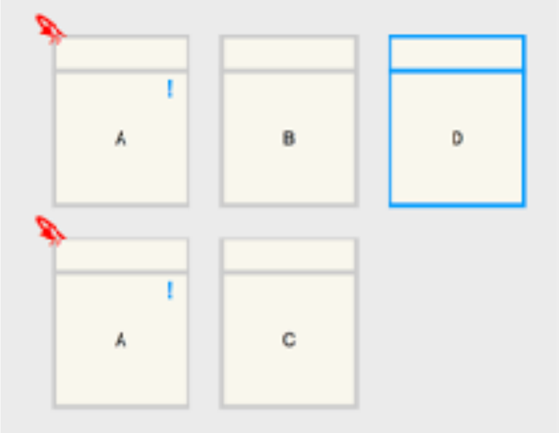
4. **Execution in Between.** “Task B should be performed not before task A has been executed, and not later than C.”

SOLUTIONS

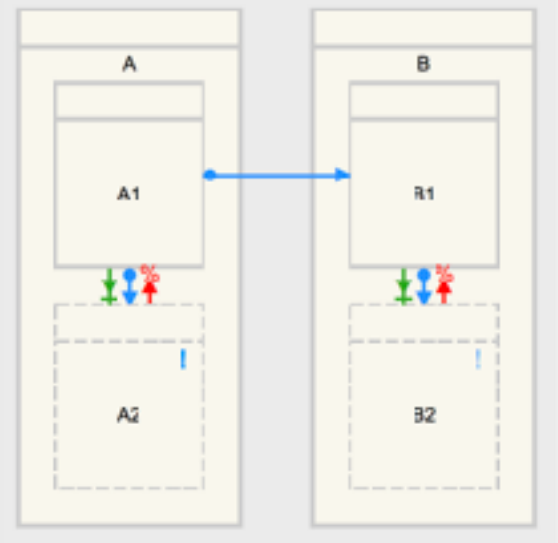
1.



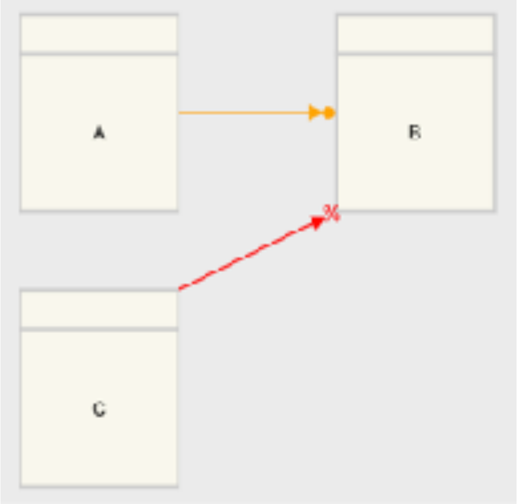
3.



2.

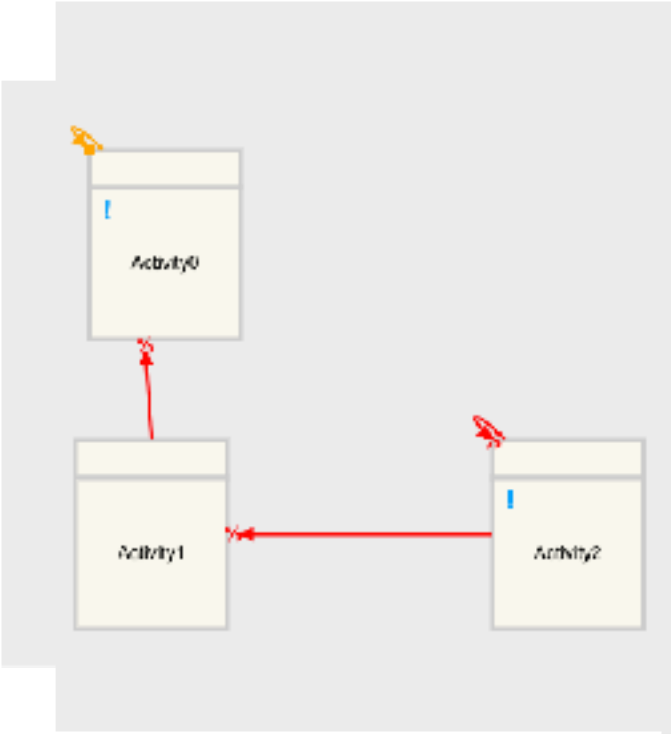


4.

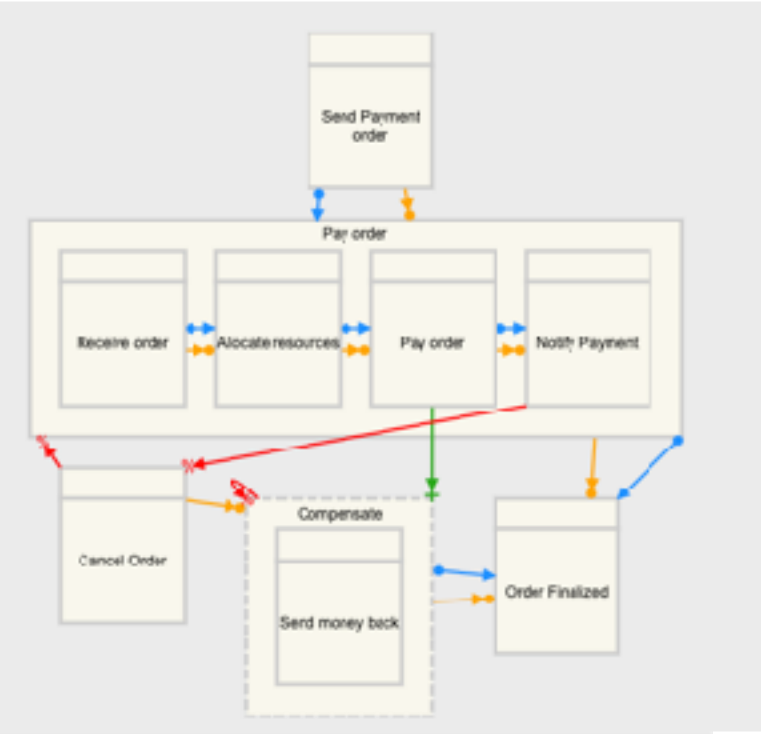


Exercises (Soundness)

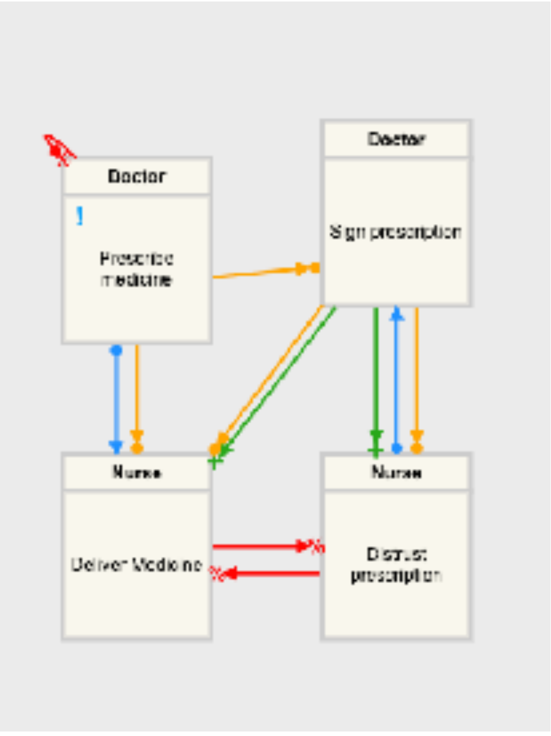
1.Discuss in pairs whether the following processes are sound:



A

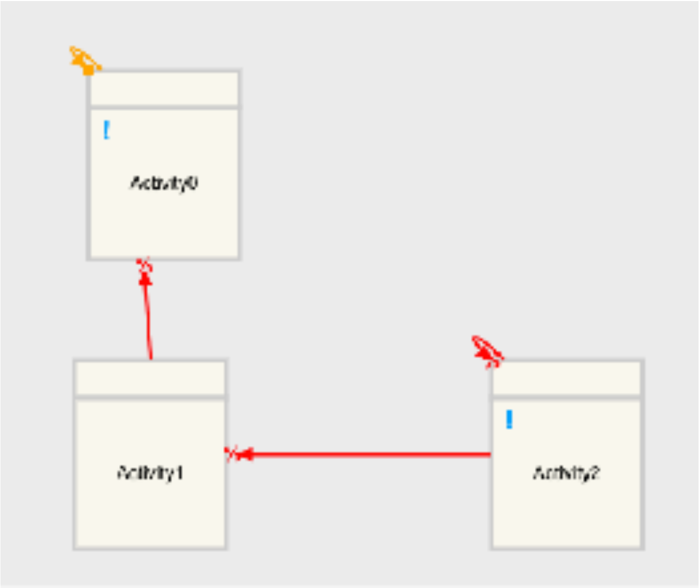


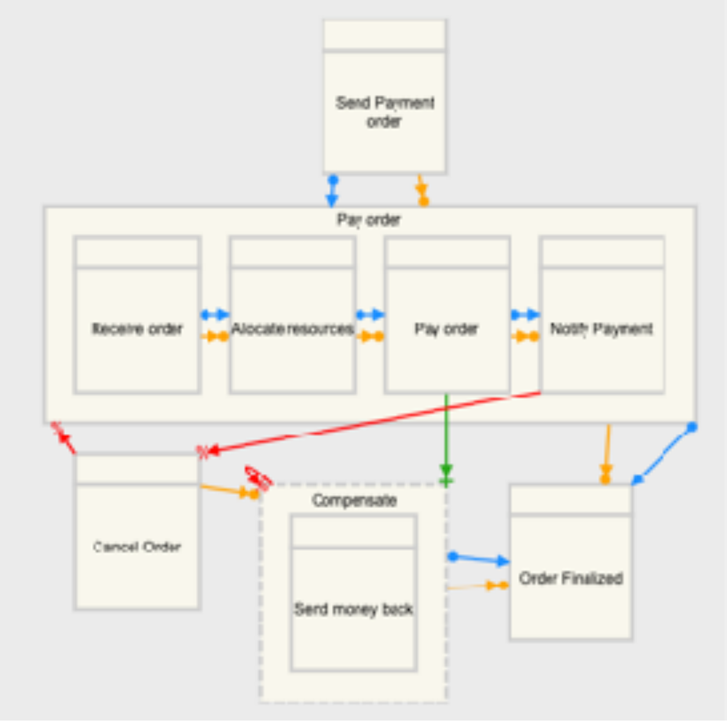
B

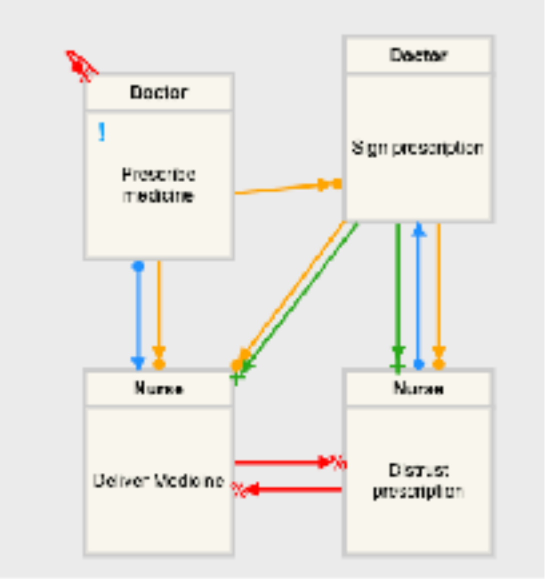


C

SOLUTIONS







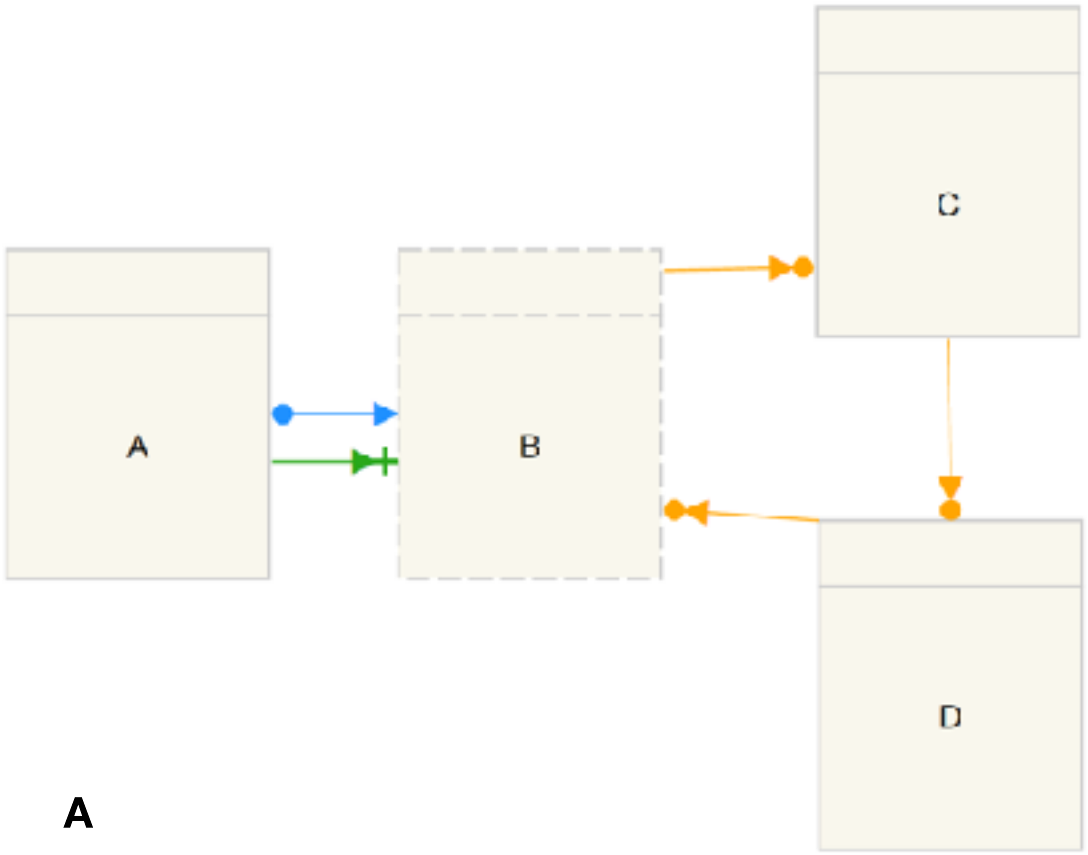
- Sound. The trace <*Activity1, Activity2*> is an accepting trace

- Sound. The empty trace is valid. Moreover, the execution of *Send Payment* order generates a **finite** chain of responses

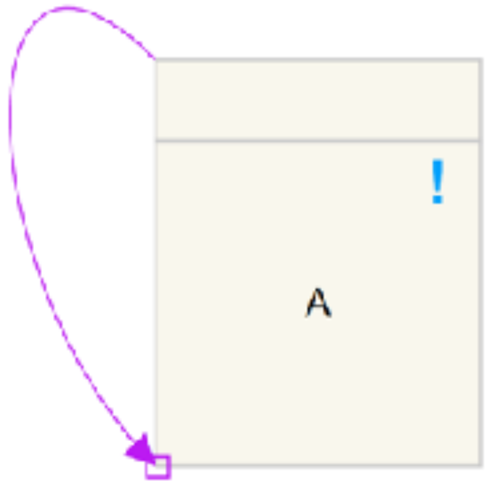
- Sound: The empty trace is not accepting, but executing <*Prescribe Medicine, Sign Prescription, Deliver Medicine*> will generate an accepting trace.

Exercise:

2.Argument whether the following processes are sound:

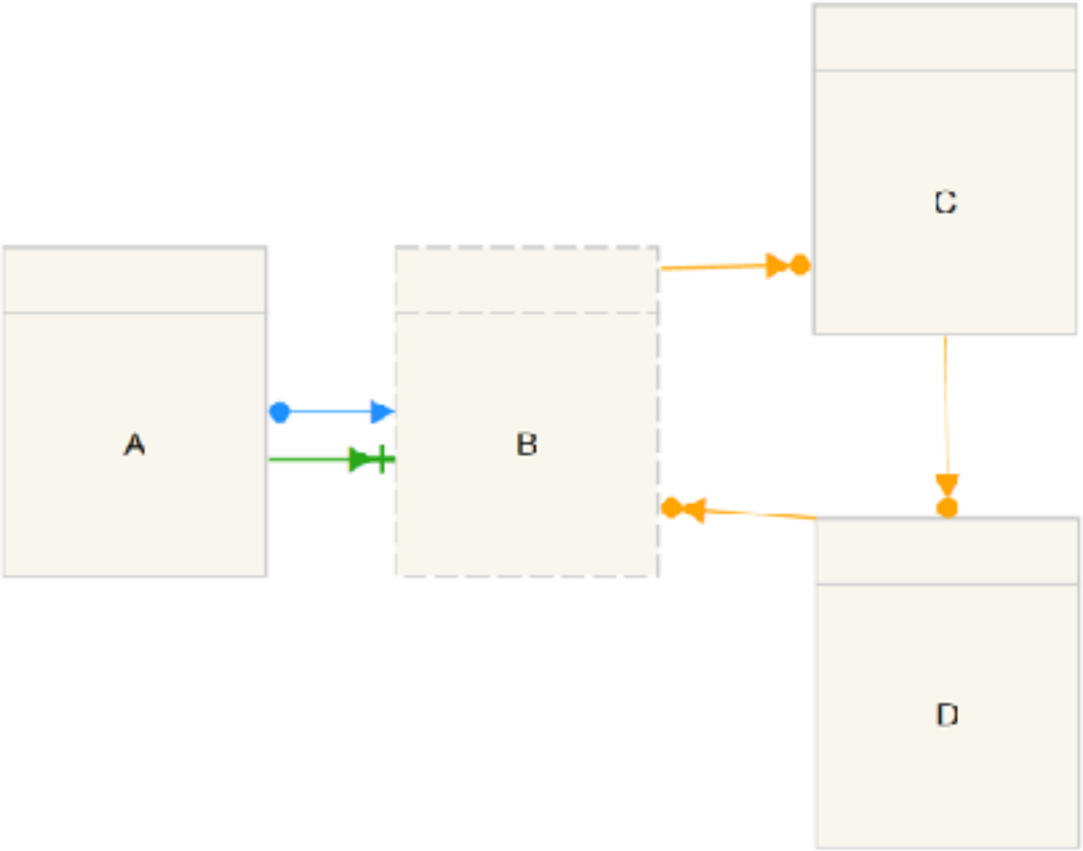


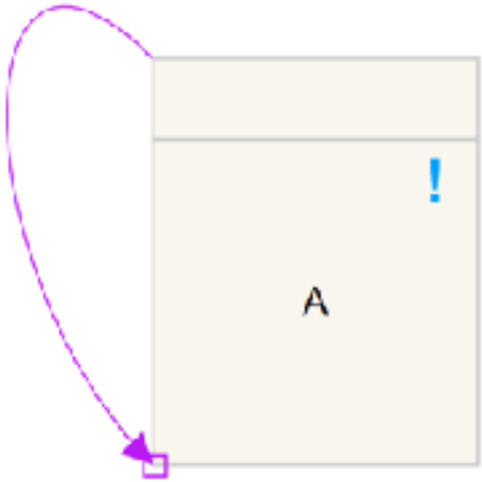
A



B

Solutions



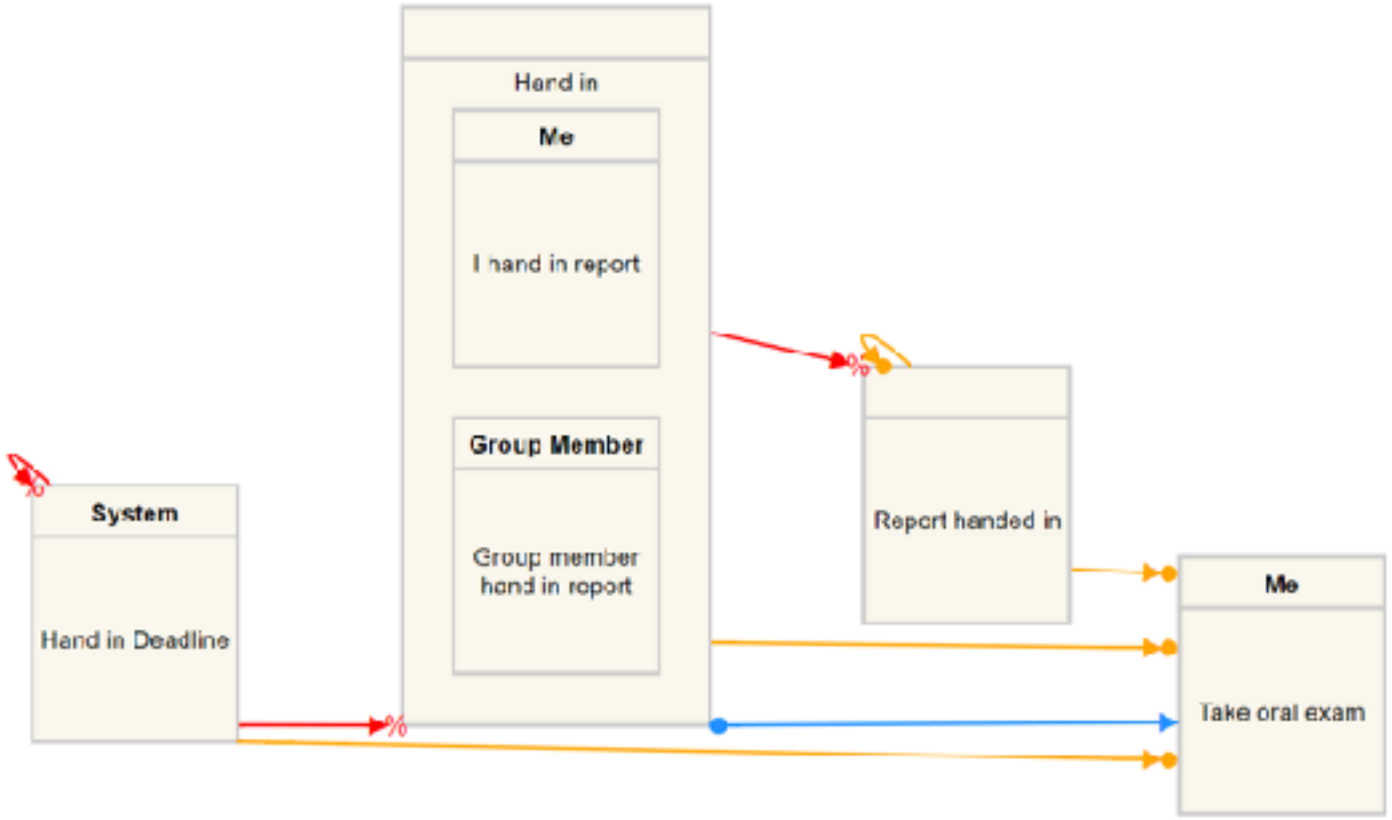


- Sound: if you execute *C* first, you can generate an accepting trace.
  But, if you were to start with *A*, you will generate a deadlocked process

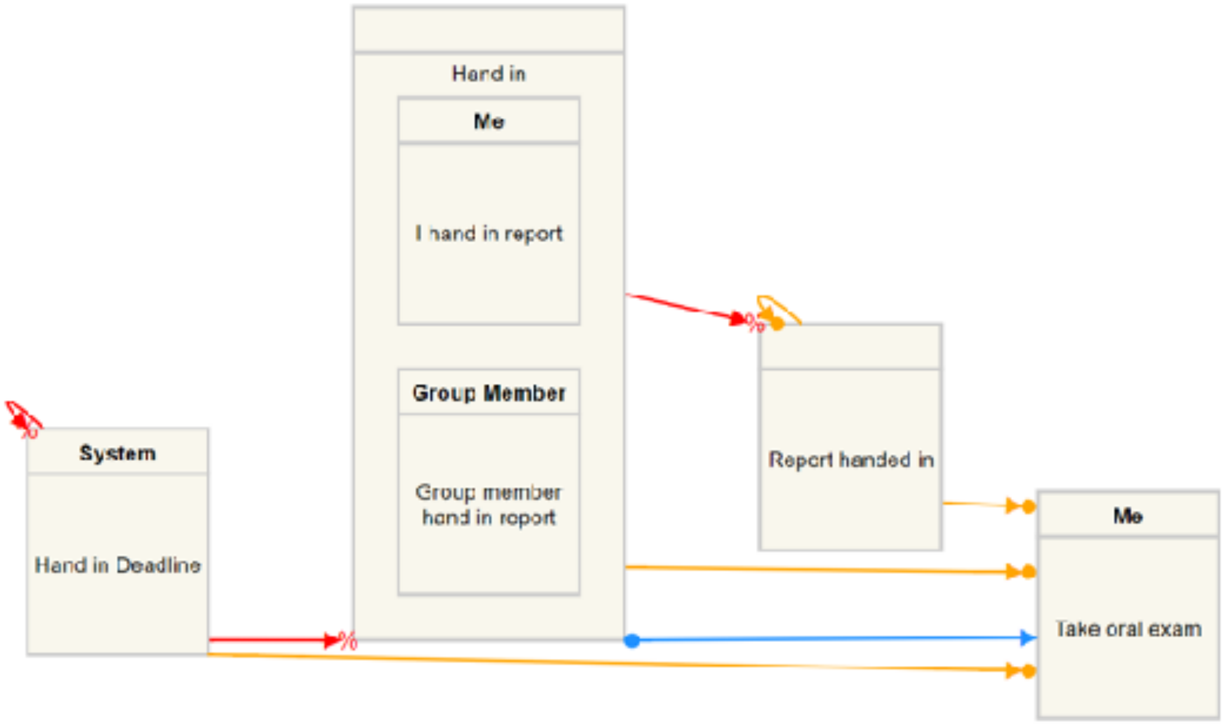
- Unsound: *A* can only be executed if the source of the milestone (*A* itself) is not pending, which is not the case.

Exercise

3.Argument whether this process is sound:



Solution



The process is sound: even if you have *Report handed in* with a cyclic dependency blocking *take oral exam*, this does not block the execution as any execution of the activities in the nesting *hand in* will exclude it.

02291 System Integration

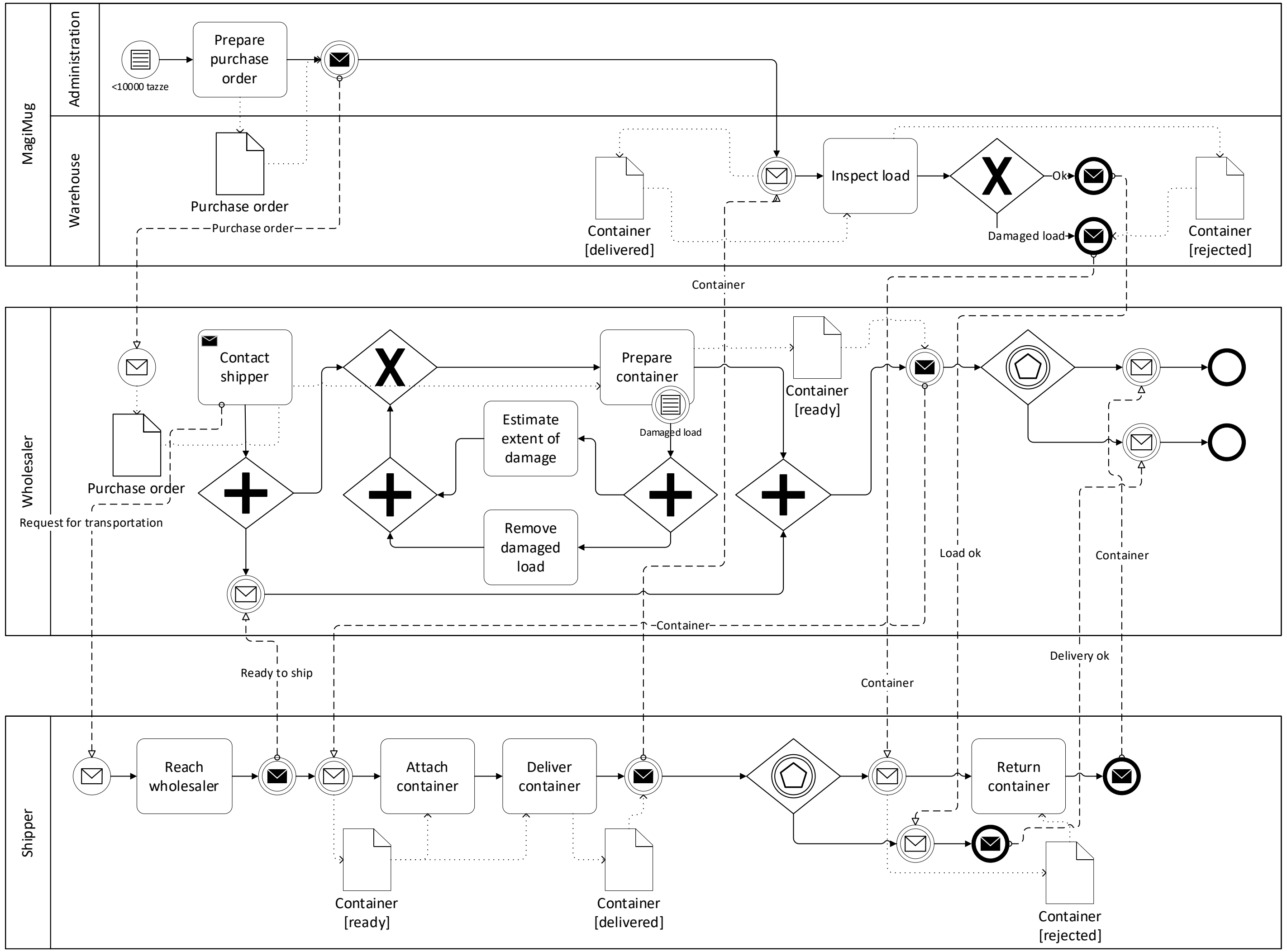
Introduction to BPMN: Solutions to exercises

Exercise 1 – MagiMug

MagiMug is a manufacturer of promotional mugs, which are sold to companies, universities, hotels, etc. In particular, MagiMug buys white ceramic mugs from a wholesaler and screen prints on them a picture provided by the customer.

When the number of cups in stock is less than 10000 units, MagiMug's administration prepares a purchase order and sends it to its wholesaler. Once the order has been received, the wholesaler contacts a shipper. Then, while waiting for the shipper to reach its premises, the wholesaler prepares a container to be shipped to MagiMug. If, while preparing the container, part of the load gets damaged, the activity is stopped, and the extent of the damage is estimated. At the same time, the damaged load is removed from the container. Once these two activities are complete, the container preparation is repeated. Once the container is ready and the shipper has reached the wholesaler’s premises, the container is given to the shipper, who attaches it to his/her truck, and delivers it to MagiMug. Once the container is received, MagiMug's warehouse workers check the integrity of the load. If everything is in order, this is reported to the shipper, who in turn notifies the wholesaler of the success of the activity, and the process ends. If, on the other hand, part of the load is damaged, the shipper takes the container back to the wholesaler and the process ends."

Exercise 1 – BPMN collaboration diagram

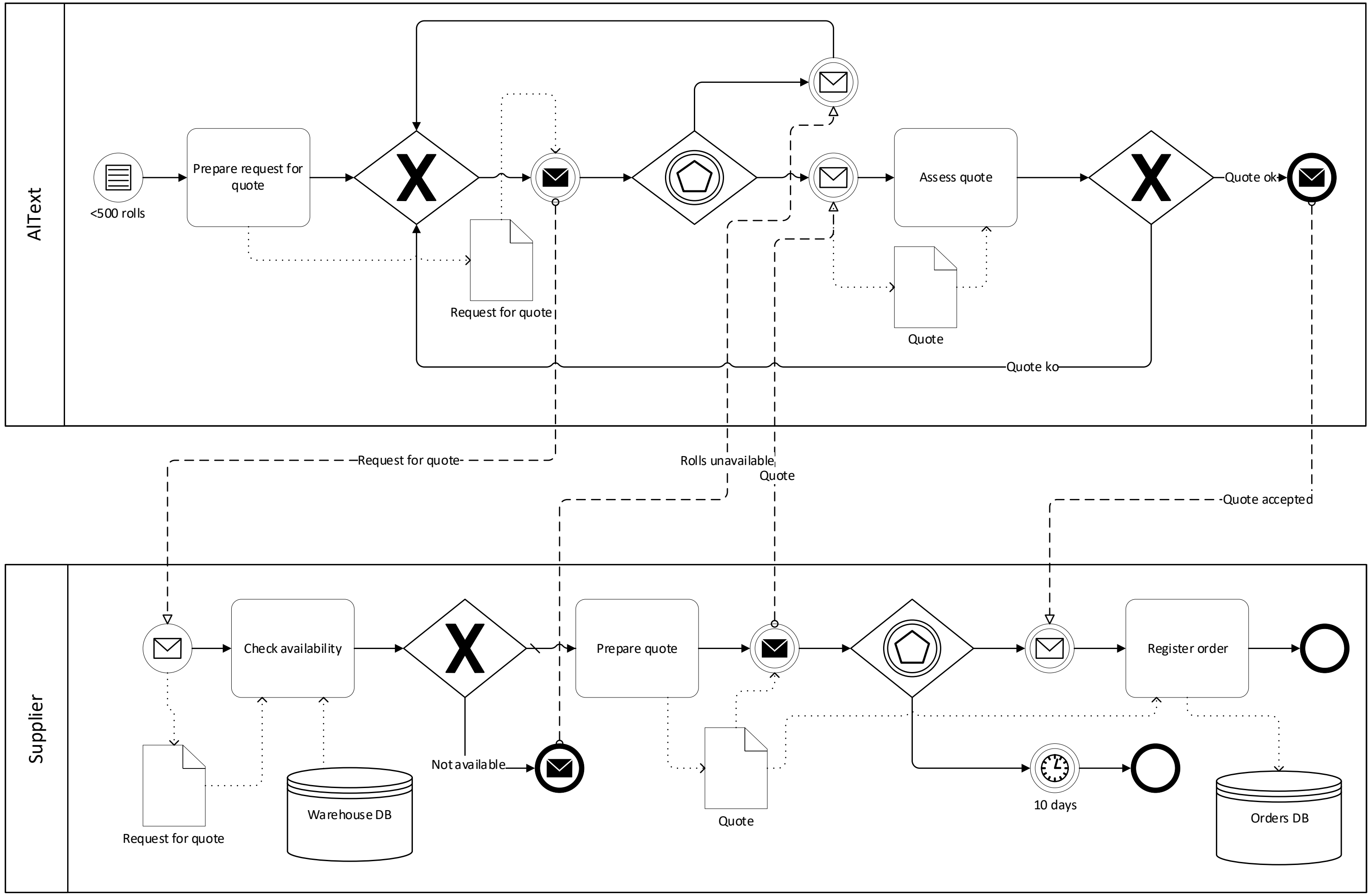


Exercise 2 – AlText

AlText is a publisher of printed academic textbooks and scientific journals.

When the paper rolls that AlText has in stock fall below 500 rolls, a request for a quote for 2000 rolls is prepared and sent to one of its supplier. The supplier, upon reception of the request, verifies if it has enough paper rolls in its warehouses. If the supplier does not have enough rolls, it terminates its process by informing AlText. AlText then sends the request for a quote again to another supplier. If a supplier has enough rolls available, it prepares a quote, which is sent to AlText. If, after assessing the quote, AlText is satisfied with it, AlText notifies the supplier that the quote has been accepted, the supplier registers the order, and the process ends on both sides. If AlText is not satisfied with the quote, it sends the request for a quote again to another supplier. If a supplier does not receive the notification within 10 days since the quote was sent, it ends the process.

Exercise 2 – Main process



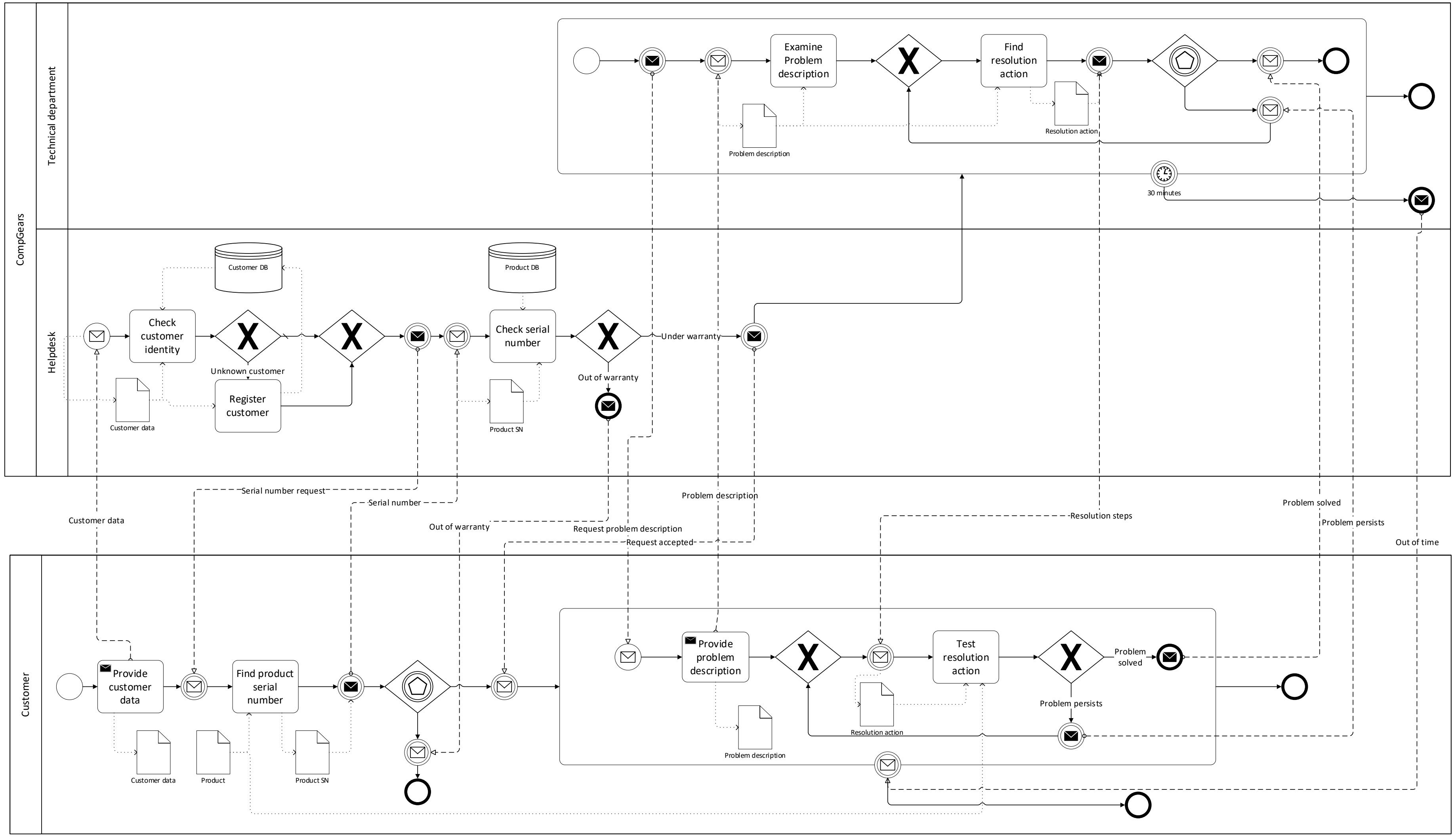
Exercise 3 - CompGears

CompGears, a consumer electronics company, wants to model its helpdesk's technical support process.

The process begins when a customer contacts the company's helpdesk with a technical support request, communicating his/her identity. First, the helpdesk operator checks whether the customer is already present in the system and, if not, (s)he registers the customer. Then, the helpdesk operator asks the customer to provide the serial number of the product. Once this information is received, the helpdesk operator checks if that product is still under warranty. If not, the helpdesk operator ends the process by notifying the customer that the product is out of warranty. Otherwise, (s)he notifies the user that the request has been accepted and forwards it to an operator from the company's in-house technical department.

First, the technical operator asks the customer to describe the problem. Once this information is received, (s)he examines the symptoms and then looks for a possible resolution action, which is provided to the customer. The customer then verifies whether that resolution action solves his problem and communicates the outcome to the technical operator. If the resolution action is successful, the process ends on both sides. If not, the technical operator searches again for a resolution action. This is repeated until the problem is resolved. If the technical operator is busy for more than 30 minutes on a specific support request, the process ends by inviting the user to repeat the entire procedure again.

Exercise 3 – BPMN collaboration diagram



02291 System Integration

Introduction to DMN: Solutions to exercises

© Giovanni Meroni

Exercise 1

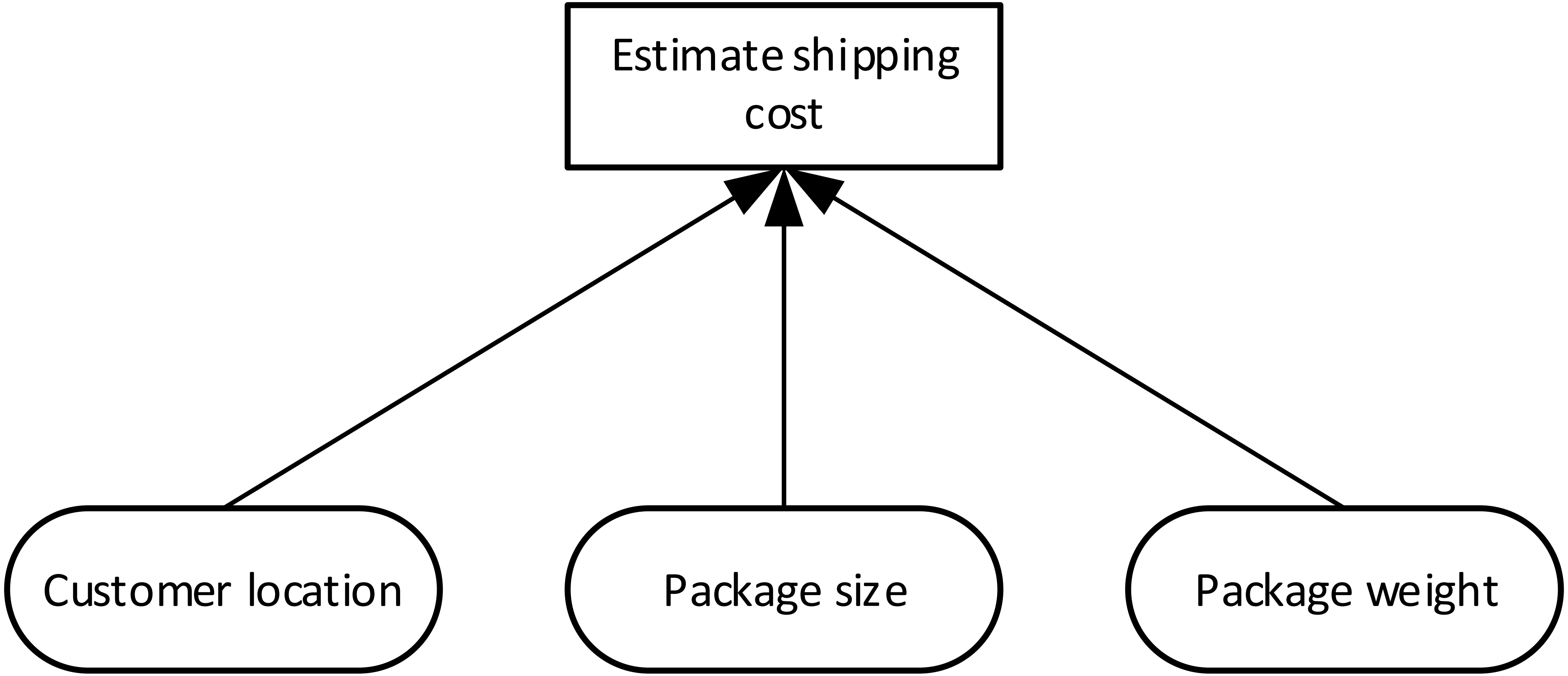
The decision task “Estimate shipping cost” of Example 3 is organized as it follows:

- If the customer is in EMEA, the weight of the package is lower than 10 Kg, and the volume of the package is smaller than 10 cm3, then the cost amounts to 100 DKK. Otherwise, the cost is 300 DKK.
- If the customer is in America or APAC, the weight of the package is lower than 5 Kg, and the volume of the package is smaller than 7 cm3, then the cost amounts to 200 DKK. If the weight of the package is greater than 20 Kg, and the volume of the package is bigger than 30 cm3, then the cost amounts to 2000 DKK. Otherwise, the cost amounts to 1000 DKK.

Implement this task with DMN:

- Draw the DMN diagram of this task
- Implement the decision with a decision table that uses the First (F) hit policy.
- Modify that decision table to use the Any (A) hit policy instead.
- Modify that decision table to use the Unique (U) hit policy instead.

Exercise 1 – DMN diagram



Estimate shipping cost

Customer location

Package size

Package weight

Exercise 1 – First decision table

Estimate shipping cost (F)	Customer location {EMEA, APAC, America}	Package weight [0,50]	Package volume [0,100]	Shipping cost [0,1000] 1000>0
1	EMEA	< 10	< 10	**100**
2	EMEA	-	-	**300**
3	-	< 5	< 7	**200**
4	-	> 20	> 30	**2000**
5	-	-	-	**1000**

Exercise 1 – Any decision table

Estimate shipping cost (A)	Customer location {EMEA, APAC, America}	Package weight [0,50]	Package volume [0,100]	Shipping cost [0,1000] 1000>0
1	EMEA	< 10	< 10	**100**
2	EMEA	>= 10	-	**300**
3	EMEA	-	>= 10	**300**
4	APAC	< 5	< 7	**200**
5	APAC	> 20	> 30	**2000**
6	APAC	>= 5	<= 30	**1000**
7	APAC	<= 20	>= 7	**1000**
8	America	< 5	< 7	**200**
9	America	> 20	> 30	**2000**
10	America	>= 5	<= 30	**1000**
11	America	<= 20	>= 7	**1000**

Exercise 1 – Unique decision table

Estimate shipping cost (U)	Customer location {EMEA, APAC, America}	Package weight [0,50]	Package volume [0,100]	Shipping cost [0,1000] 1000>0
1	EMEA	< 10	< 10	**100**
2	EMEA	>= 10	< 10	**300**
3	EMEA	< 10	>= 10	**300**
4	EMEA	>= 10	>= 10	**300**
5	APAC	< 5	< 7	**200**
6	APAC	> 20	> 30	**2000**
7	APAC	>= 5	<= 30	**1000**
8	APAC	<= 20	> 30	**1000**
9	APAC	< 5	[7,30]	**1000**
10	America	< 5	< 7	**200**
11	America	> 20	> 30	**2000**
12	America	>= 5	<= 30	**1000**
13	America	<= 20	> 30	**1000**
14	America	< 5	[7,30]	**1000**

Exercise 2

- Given the following decision table:

Determine discount (?)	Fidelity level {gold, silver, bronze}	Items bought [0,100]	Total value [0,100000]	Discount [0,30] 30>0
1	Gold	> 2	-	**30**
2	Gold	-	-	**10**
3	Silver	-	> 10000	**20**
4	Silver	> 2	-	**10**
5	Bronze	> 2	> 1000	**10**

- Which single hit policies would not cause any violation?
- Would all the previously identified hit policies lead to the same outcome?
- Is the table complete?

Exercise 2

- Given the following decision table:

Determine discount (?)	Fidelity level {gold, silver, bronze}	Items bought [0,100]	Total value [0,100000]	Discount [0,30] 30>0
1	Gold	> 2	-	**30**
2	Gold	-	-	**10**
3	Silver	-	> 10000	**20**
4	Silver	> 2	-	**10**
5	Bronze	> 2	> 1000	**10**

- Which single hit policies would not cause any violation? **First and Priority**
  - Rules 1 and 2, 3 and 4 could hold simultaneously (thus violating Unique) and would give different results (thus violating Any)

Exercise 2

- Given the following decision table:

Determine discount (?)	Fidelity level {gold, silver, bronze}	Items bought [0,100]	Total value [0,100000]	Discount [0,30] 30>0
1	Gold	> 2	-	**30**
2	Gold	-	-	**10**
3	Silver	-	> 10000	**20**
4	Silver	> 2	-	**10**
5	Bronze	> 2	> 1000	**10**

- Would all the previously identified hit policies lead to the same outcome? **Yes**
  - When 1 and 2 hold, First would select 1, giving 30% discount. Priority would also select 1 since 10 has lower priority than 30 (i.e., 10 < 30).
  - When 3 and 4 hold, First would select 3, giving 20% discount. Priority would Priority would also select 3 since 10 has lower priority than 20 (i.e., 10 < 20).

Exercise 2

- Given the following decision table:

Determine discount (?)	Fidelity level {gold, silver, bronze}	Items bought [0,100]	Total value [0,100000]	Discount [0,30] 30>0
1	Gold	> 2	-	**30**
2	Gold	-	-	**10**
3	Silver	-	> 10000	**20**
4	Silver	> 2	-	**10**
5	Bronze	> 2	> 1000	**10**

- Is the table complete? **No**
  - Discount is not specified for Silver customers with 2 or less items bought and 10000 DKK or less total value.
  - Discount is not specified for Bronze customers with 2 or less items bought or 1000 DKK or less total value.