

Third Assignment REBS

UNIVERSITY OF COPENHAGEN



Formalities

- Deadline: 18 January 11:00 am
- Includes Part A (last week) and Part B (this week)
- Include a PDF with the report and if there is any functional code, include sources and/or links to repositories.
 - If there are any additional packages, provide a way to automate installation (maven/ant/build scripts)

Part A: Business Process Compliance

- In this first part of the assignment you will
 - Elicit reference models of laws from real legislations
 - Extend your conformance checking tool in assignment 1 to account for compliance dimensions
 - Implement compliance checking in order to compare models

Exercise A.1

Generate a DCR graph of the policies in §42 of the Danish Consolidation Act for Social Services (Serviceloven). You may use the Process highlighter and the DCR annotation guidelines located in the “Files” section of this course

- *42.-(1) The municipal council shall pay compensation for loss of earnings to persons maintaining a child under 18 in the home whose physical or mental function is substantially and permanently impaired, or who is suffering from serious, chronic or long-term illness. Compensation shall be subject to the condition that the child is cared for at home as a necessary consequence of the impaired function, and that it is most expedient for the mother or father to care for the child.*
- *(2) The requirement in subsection (1) above that the child shall be cared for at home shall not apply to any child mentioned in subsection (1) who has been placed in care under section 52(3) (vii) in connection with the child's hospital visit. It is a condition that the presence of the mother or father at the hospital is a necessary consequence of the child's functional impairment and that such presence is most expedient for the child.*

Exercise A.2

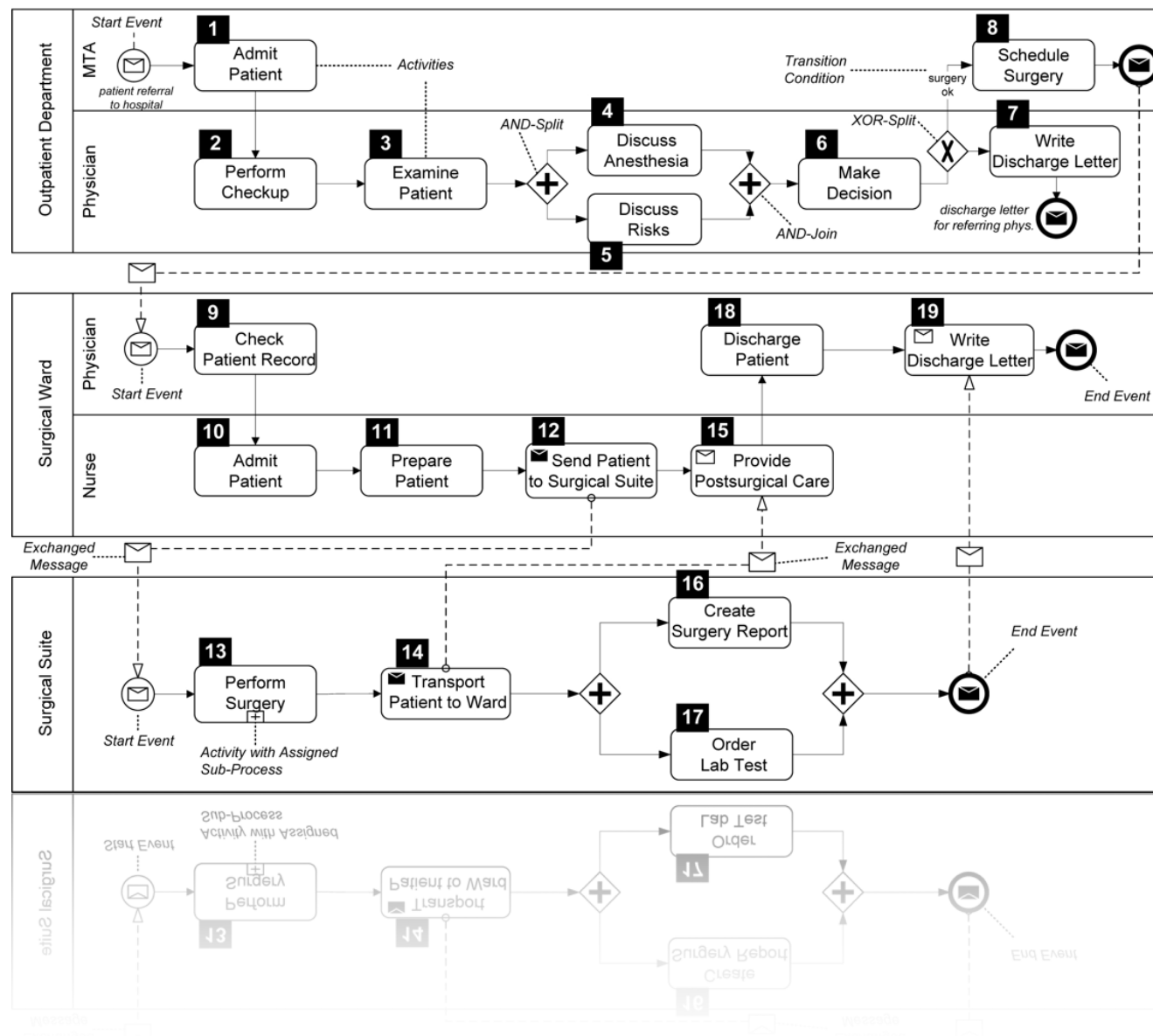
- Construct a set of accepting and non-accepting traces based on activities identified in Exercise A.1.
- Extend the implementation of your conformance checking algorithm so it accounts for the compliance diagnostic criteria in this slide.
- Show via examples how the policies in Exercise A.1 are violated, compliant or weakly compliant.

(Bonus) Exercise A.3.

- Create a compliance checker for DCR graphs. To this aim, you need
 - To create an interface that allows to specify (or upload) a DCR graph of a compliance rule, and a DCR graph of a process model, as well as the term alignments
 - To implement the algorithms for compliance checking given before, providing diagnostics of whether full or weak compliance has been achieved as a result of the analysis
- You are welcome to reuse modules from previous assignments

Exercise A.4

- Consider the topmost pool of the **BPMN** model on the right:
- a. Write DCR compliance rules for the following compliance rules:
 - The outpatient department always makes a decision in this process
 - After performing a checkup, either schedule a surgery or write a discharge letter should be executed, but not both in the same trace.
 - After discussing anesthesia and the risk of the patient, the process must follow with a decision.
- Attempt to create an equivalent DCR process model describing the behaviors of the topmost pool, and ensure the process is compliant with rules a.1 — a.3 by using the compliance checker developed in Exercise A.3



Part B

- In this part you will use Jolie to implement subscriber to a MQTT-XES broker in HiveMQ
- You can try to run your exercises with several logs (for all the broker url is `broker.hivemq.com`):
 - `pmcep/REBS/#`
 - `pmcep/Hospital log/#`
 - `pmcep/Disco Example Log/#`
 - `pmcep/Hospital Billing - Event Log/#`
 - `pmcep/DreyersFond/#`

Exercise B.1

Below is an implementation of a server process subscribing to the call center log (called Disco Example Log) published at the HIVEMQ broker.

```
include "mosquitto/interfaces/MosquittoInterface.iol"
include "console.iol"

execution {concurrent}

inputPort Server {
  Location: "local"
  Protocol: sodep
  Interfaces: MosquittoReceiverInteface
}

outputPort Mosquitto {
  Interfaces: MosquittoInterface
}

embedded {
  Java:
    "org.jolielang.connector.mosquitto.MosquittoConnectorJavaService" in Mosquitto
}
```

B.1 (continued)

```
init {
    request << {
        brokerURL = "tcp://broker.hivemq.com",
        subscribe << {
            topic = "pmcep/Disco Example Log/#"
        }
        // I can set all the options available from the Paho library
    }
    setMosquitto@Mosquitto (request)()
}

main {
    receive (request)
    println@Console("topic :      "+request.topic)()
    println@Console("message :    "+request.message)()
}
```

- using the jar files JolieMosquittoConnector.jar and org.eclipse.paho.client.mqttv3-1.1.2-20170804.042534-34.jar which should both be placed in a folder called lib in the same location as the subscriber source file.

Exercise B.1.1

- Run the subscriber. It should (after some warnings) receive events from the broker on the call center example with topics at the format pmcep/Disco Example Log/CaseX/Activity, where X is a number denoting the ID of the case and Activity is the activity, e.g. Inbound Call.
- 2.1 Run the subscriber. It should (after some warnings) receive events from the broker on the call center example with topics at the format pmcep/Disco Example Log/CaseX/Activity, where X is a number denoting the ID of the case and Activity is the activity, e.g. Inbound Call.
- 2.2 Run the subscriber in two shells. Explain what happens and how this is different than for a message queuing system and actors.
- 2.3 Implement two new subscribers that uses the topic filters to only listens to respectively Inbound Email and the Inbound call activities as the Email client and the [call center example](#) above.
- 2.4 Try to implement a subscriber that counts how many time each kind of activity happens.