Networked Systems and Services, Fall 2017, Design Task 3, Breach recovery in cross-organizational business process

Assignment
In this assignment, you will design a system according to the specifications given in this document. Your goal is to design the system to be reliable, but also to have a priori decent performance. The document you produce will discuss the design tradeoffs that you have considered.

As tools, you can use any of the reliability techniques we have seen in class, but also any well-known techniques or come up with new mechanisms and solutions.

Task selection
There are three tasks numbered 1, 2, and 3. You can determine which assignment is yours by calculating (“Your student ID” modulo 3) + 1.

Task and submission guidelines
The purpose of this task is to design business processes for a import/export-collaboration working within an example service ecosystem. In that ecosystem breaches between partners need to be recovered following processes dictated and decided on by policy makers of the ecosystem, instead by collaboration members. The text below describes the overall business scenario and structure of the ecosystem. Further down there are instructions for using BPMN and suitable tools for business process design.

As an end result, please provide
- access to your design or printout of your design. Do not forget to tell which tool needs to be used for opening your design if it is in one of the clouds. Screenshots and paper designs are acceptable as well.
- a commenting documentation of your design that identifies the design challenges, motivates your design choices among the alternatives, and discusses the pros and cons of the solution. Please remember to include discussion for the particular questions listed in task steps below.

Background

Networked business refers to the current trend in public domain governance and in the markets of intangible goods (or tangible goods, even their combinations) where groups of enterprises together compete with each other for success. This is in contrast with the tradition of each enterprise searching its success on its own.

In networked business, the organisation of cooperation and its management can be done many different ways. We can approach it from computer science, software engineering, economics, or application area specific (such as health-care) viewpoints.

In the following, we focus to the alignment of business needs in general and the required computing and communication solutions of interest. Making business involves business processes, negotiations, contracting, trust, breach detection, recovery from breach situations, dictionaries for terms used in negotiations, document contents definitions, and regulations on what kind of operation is acceptable.
For simplicity, we use the term service ecosystem as a generic term to grasp a large set of conceptual frameworks that address these needs. Especially, these service ecosystems expect that there are support for modeling and engineering new business service types and collaboration models, for introducing new business service tenders, and to create and manage new collaboration cases. For example, virtual organisation breeding environments found in the literature fit in.

This selected viewpoint considers both public and private domain organisations and therefore, here the terms enterprise, (public) organisation, and (private) company have no difference. The terms enterprise and organisation are often used to refer to all of them.

On this course, we have considered different levels of protocols and discussed reliability support functionalities in them, and studied how different solutions can work together. At this networked business support layer, a few middleware layers up within the OSI application layer, the relevant protocols to consider are business processes.

A business process describes the workflow (including variations) between partners, associated to a certain trigger or an identified goal. As there can be several triggers, there can be several workflows too. A business process is defined in terms of roles and interaction between those roles. Actual persons or software applications (or agents) can occupy those roles when they are capable of fulfilling its requirements and do not violate assignment rules. Role requirements include the service type expressed in terms of interface syntax and externally observable behaviour (= exchange of messages). Assignment rule may for example forbid a person or an agent from playing the accountant role and the account user roles at the same time.

BPMN (business process modeling notation) is one of the current business process design languages. The variation of similar languages shares key concepts like roles and workflow diagrams, messaging, and workflow control structures similar to most programming languages. Special features include transactional tasks with compensation routines for definitions on what to do in failure situations. Since business transactions have side effects in the tangible world, simple database rollbacks are not enough, but the designer must consider how to develop the collaboration situation is steps to something that is acceptable for all parties. Often the process is a forward recovery process: it is not reasonable to try to reach the last coherent shared state with each partner, but instead find a next sensible shared state. For example, our overbooked passenger volunteering to drop out may get compensation in ticket price, night in airport hotel, dinner and same outgoing flight the next day.

This kind of business process modeling languages differ from executable business process languages by leaning towards abstract models. The design is expressed through roles and service types, instead of service implementations. Further, there is space for policies and restrictive criteria, so that the overall behaviour can be adapted to the particular situation in which the model is used. For example, a business process and associate services might be able to provide a dozen of different services every day in seconds, but by policies some service providers limit the rate to one case every day except weekends and bank holidays, due to some marketing and security strategies.

Model-driven engineering and operation environment may support execution of business processes based on this kind of business process descriptions and availability of metainformation on service implementations. Alternatively, the business process models and policies can be used for monitoring and controlling that the expected behaviour is followed.
In the example service ecosystem we assume the following things:

- **Each organisation is autonomous.**
  
  The organisations can choose their computing platform freely and business services they run are encapsulated in a way that hides that platform choice. Business services are made visible for other ecosystem members only as service offers that list the service type (interface syntax, external business processes + role that the offering partner takes.

  The organisations can freely choose which business services they provide, when, and to whom those are available. They can choose which collaboration cases they participate and which they reject. These decisions can depend on the organisational business strategy and the governance base on it. As computer scientists, we tend to note such rules as business policies, because the policies we can further use as parameters that modify the behaviour of a service. For example, a loyalty program in a company may state a policy that all loyal customers get a lower price or additional services. The same way organisations can capture rules from the national laws for example, into their policies to follow.

- **Service ecosystem provides support facilities for all the member organisations.**
  
  These support facilities may include
  
  - discovery of services based on their type,
  - repository of business process models (and chunks of them) and
  - ecosystem level policy set.

  These facilities can be considered as an infrastructure provider with corresponding units.

- **When autonomous organisations work together they do not integrate their computing systems.**

  Therefore, there is no access or administrative rights by one organisation to another's computing system. When there is a single decision-maker for the whole distributed system, using access control lists works, but when systems are indeed under different controllers, this is not the case. Instead, contracting, control of collaboration, and expressing service requests take place on the abstract level, using the ecosystem-widely known models as known terms.

  **When the involved organisations agree on the use of services they use three deontic logic terms: obligation, prohibition and permission.** Please note that the negations of the terms are not simple, and not all agree on them either. "Not obligated" may or may not mean the same as "obligated not to" or even "permitted".

  Furthermore, each organisation is independent in deciding what deontic logic terms it connects to outgoing requests and incoming requests. For example, let's consider a client-server situation where there is a contract that the client uses the service X from the server:

  - Client obligated to use X, Server situation prohibits all use \( \rightarrow \) outcome: failure
  - Client permitted to use X and others, server situation prohibits use of X \( \rightarrow \) potentially some other service with same service type is used successfully
  - Client obligated or permitted to use X, server permits / is obligated in this situation \( \rightarrow \) successful event within collaboration
  - Client prohibited to use X \( \rightarrow \) no collaboration intended
Here we had a simple client-server case, but the organisations might have a negotiated contract in place as well:
  o Contract says A obligated to provide service X for all, A refuses as its internal policy says X is prohibited (in the situation) \(\rightarrow\) outcome: breach (failure to conform to the contract

**Just alike business processes are defined,** also failure handling paths must be defined. Business processes may fail for different reasons: when a more tempting business opportunity occurs, an enterprise may choose the sanctions for dropping an old collaboration over letting the new opportunity escape.

It should be remembered, that when business processes control the flow of tangible items (parcels, carriers, humans, bills and notes), rollbacks are not sufficient. If an airline has oversold flight tickets for your second leg trip and boards someone else instead of you, it does not feel right if they just return you ticket price for that leg and leave you at that strange airport. Instead, they should do forward recovery: they should activate a corrective business process that takes you to your destination, preferably in time, through an alternative route or by another airline. Unless if you prefer returning home directly as you would miss your meeting anyway.

Decision-making for selecting the appropriate corrective process involves questions like
  o Under which circumstances the corrective process can be triggered?
  o Who gets to be part of decision-making when selecting the right recovery process?
  o Who decides the policies that regulate recovery processes and situations triggering them?

**Business scenario for the task**

Consider a group of import & export companies, each from a different country that have some contradictory rules about safe transport or legal merchandizes.

These companies have expertise on the international market situation, directives on international commerce, and international market laws. They do not own any fleet for transporting tangible goods though, but have to outsource all transportation. They have agreed that each of them only use their national transporting services (and we assume there is no international services in use). The companies have a contract on how they bill clients and how they split costs and profits.

In the service ecosystem, in which their collaboration started, there is a group of directives (policies, allowed and denied business processes) they have to obey in order to keep their membership status. They gain a lot of market status knowledge through the ecosystem. The directives give guidance on what kind of goods can be imported to which countries (illegal products to possess or sell), what kind of transport limitations there are in each country (weight of lorries, width of roads, availability of airports). The ecosystem knowledge and services can be presented as an additional organisation.

**Your task**

Design the collaboration between these partners and refine that with breach recovery processes. Use BPMN (business process modeling notation) for modeling, either with one of the tools mentioned below, or with paper and pen.

**Work in steps**
1. Design the key business processes between the partners using BPMN
o Supporting processes can be left out for simplicity and size. Use rather abstract blocks when there is no structural essential things to include.
o Focus on exchange of key information, remember to include messages where only control of cooperation is required (swapping terms, requesting to discard unnecessary service, asking for confirmation that a service is on its way when it takes long, exchanging models etc)

2. Identify potential breaches

3. Design a couple of alternative recovery business processes

4. Consider where in your system architecture you can place them and why. What benefits and drawbacks there are for these choices?

5. Did you remember to consider who in the collaboration scenario gets to decide which of your recovery processes is used?

6. Complete your overall design by ensuring all collaboration partners know when (if) the purchase reached the final location and know whether any breach recovery (with potential sanctions) took place during the collaboration operation.

7. Make sure you discuss the choices you made and assumptions behind them.
   o Remember to include into your submission commentary on balancing of system reliability and business expectations in the scenario
   o What is your definition of reliability here?
   o What are the facilities for supporting reliability?
   o *What scale of costs is caused? As a measure of cost, use the number of messages exchanged between partners for each service type request, considering best and worst cases. Is this a reasonable amount of messages for such task?*

8. Collect your design documentation and discussion paper and submit them through Moodle. Remember to include instructions if you use external services as a platform.

**Tools and advise**

Please find introduction to business process modeling through these links:

o Basic tutorials
  o We do not intend to learn the whole language, but to use pools. Lanes, activities, gateways, events, transactions, restrictions, alternative paths based on choices, data acquired through conversations across organizational boundaries
  o Each of the tools include their own tutorials, but these mentioned here are a bit more on the generic level (language patterns instead of tool details), shorter for getting started or more clearly spoken and slower, in case you wish to move into those directions
OMG BPMN 2 tutorial or google for BPMN tutorial (http://www.omg.org/cgi-bin/doc?dtc/10-06-02)

Application example: applying for leave (https://www.visual-paradigm.com/tutorials/bpmn-tutorial-with-example.jsp)

One of the tool use videos in youtube, starting a set of several videos, using yet another modeling tool (https://www.youtube.com/watch?v=WtOzW8Ck5LY&index=1&list=PLp9kfsZpijeKnBpqnCaSIXmIYZP8rN1QL)

Quick quide for notations (google figures for BPMN 2 poster or use the chart in the tool you use) (https://www.slideshare.net/jlaznik/bpmn2-0-posteren)

Short video sequence on some of the modeling phases
  - Capturing the idea (https://www.youtube.com/watch?v=y1O1K_zrQ5o)
  - Thinking of independence: use of hand-overs (https://www.youtube.com/watch?v=_2KtxcDteU)
  - How to automate business process (https://kb.heflo.com/knowledge-base/automate-business-process/)

Note that most examples in tutorials instruct on orchestrated cases (single controller) while we need a choreographed case now (peers controlling each their autonomous unit)
  - an autonomous organization must have a role (separate set of swimlines) while a unit or computing system may have a swimline associated to that role
  - Each role (and swimline, unless you have a justified reason) must have a start node and an end node as the behaviours of separate services of organisations and units are independent (choreographies → starts and stops per role, orchestrations → may have sequences crossing swimlines within a single role)
  - A hand-over would map to a choreography diagram (which we do not need here) or crossing the white space between roles

Suggested tools
  - QPR Enteprise architect suit (needs installation)
  - https://www.gennymodel.com/bpmn-online-tool (free trial)
  - https://www.signavio.com/enterprise-architecture-with-archimate-2-1/# (free trial – upper right corner)
  - https://www.modelio.org/ (free)

Grading hints

There is a lot of space for using your imagination here, no need to check any laws or such for this design. The scenario is not fully realistic. Consider it a distributed game of multiple partners, each having different rules to obey.

Keep reflecting the reliability aspects in the game. Consider the viewpoints of clients in the different countries, each of the collaborating partners, each shipment and each import case. What are the potential breaches or failures?
While the overhead cost of the global system seems huge, the same facilities support engineering, correct local operation, agility on market place, following and adapting to regulatory systems and other well worth goals in addition to the reliability goal. In that sense, there are a lot of modern features in the scenario.

From BPMN, very basic use skills suffice. On paper, syntax errors do not matter. In complete models do not matter elsewhere, you can focus on making your point on the spots that you see essential. Make sure you understood the key concepts the right way.

The group of acceptable solutions is infinite. In grading, emphasis is on the way the design
- addresses autonomy of organisations and their dependence on policy makers at the same time;
- uses available status information from all organisations for sitational decision-making when choosing alternative process sequences; and
- utilizes shared ecosystem facilities.

DO NOT use UML – the size of the task explodes – UML solutions are not considered. This is not a programming task, but a task of finding the right level of abstraction and learning to see how the context ecosystem changes successful designs.

**Deliverables**
Design document. The document should explain how you have solved the problems and provide answers to the questions from Requirements section.

Use the template provided at [http://www.acm.org/publications/proceedings-template](http://www.acm.org/publications/proceedings-template) for your document. Expected length of the document would be around 6 pages in that template, depending on the number of figures that you use. Note that this is not a hard limit in any sense; it is only intended to give you an idea of the extent of the report.

**Timeline**
The assignment is due on October 17th at 23:00. No extensions will be given.

**Return**
Return as one PDF file. Name the file “username_X.pdf” where X is the number 1, 2, or 3 corresponding to the assignment number. Return the file via Moodle. Note that the files will be shared with other students for reviewing, so only indicate your name in the file.