Examination

Advanced of Software Architecture

Course codes **DAT220 & DIT544**

Date:	19 March 2020
Time:	8:00-12:00
Place:	Lindholmen / home
Teacher:	Prof. Dr. M.R.V. Chaudron
Visit to e*xam hall:*	n.a.
Questions:	6 (frontpage + 5 pages)
Results:	Will be posted by 2nd of April 2020
Grade Limits:	Pass (G) 50%, Pass with honors (VG) 75%
Allowed aids:	Ruler, Dictionary

Please observe the following:

- Write in legible English (unreadable means no points!).
- Motivate your answers, and clearly state any assumptions made.
- Start each task on a new sheet!
- Write only on one side of the paper!
- We prefer that you write with a ***pen***, not with a pencil.
  Drawings may be made using a pencil.
- Before handing in your exam, number and sort the sheets in task order!

- Make sure that any photos you include of diagrams are clear and readable

NOTE:

Not following these instructions may result in the deduction of points!

Throughout this exam we will ask you to provide examples to illustrate your answer. Your examples are required to be different from the ones in the book and the lecture slides.

Question 1 (12 pts: 4 + 4 + 4) *General knowledge on software architecture*

In the context of software development, explain how software architecture can help in the following. For each of the questions (a, b & c), provide an example to illustrate your argument. Make your examples as concrete as possible.

a. Achieving shared understanding of a system in large organisations
b. Managing the conceptual integrity of software design during the evolution of a system
c. Identifying technical risks in the development of a system

Question 2 (6 pts: 2 + (2 + 2)) *Requirements, Scenario's*

a. Explain the relation between business drivers and architectural tactics used in a system.
b. Consider a Hospital Information System (HIS). Such HIS are highly dependent on the availability of patient records. For this reason, hospitals often have a passive-stand by database running at a system at a different location to which they switch over to when their main patient-records database fails.
    1. Write a SMART *requirement* for the availability of patient-data for the HIS. Your requirement should include a suitable measure for availability.
    2. Write one *availability scenario* (following the format/formula from the book/slides).

Question 3 (10 pts: 4 + 2 + 2 + 2) *Architectural styles*

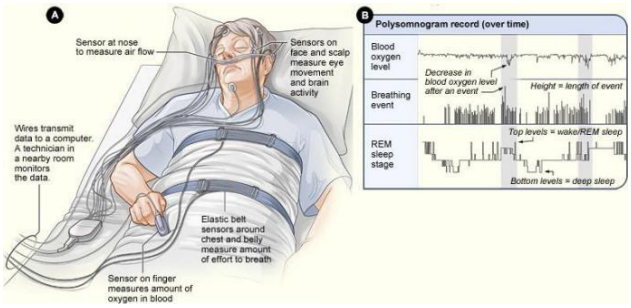
a. Give 2 ways in which the *architectural styles Client/Server* and *Blackboard* are similar and 2 ways in which they are different.

b. Describe a concrete case in which you would use a *Blackboard* style (and not another style) and explain which characteristics of the case decide in favour of the Blackboard and against the other styles (consider as alternatives Client/Server, Pub-Sub and Pipe-and-Filter).

For the following cases, describe what the most appropriate architecture style for the system's architecture is and *motivate why* this is the case.

c. Software for an electroencephalography (EEG, see Figure). Electroencephalography (EEG) is an electrophysiological monitoring method to record electrical activity of the brain. It is typically non-invasive, with the electrodes placed along the scalp. EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. This EEG

technology can be used to detect various types of activities in the brain and from this it can be analysed which parts of the brain are active in response to particular tasks. This in turn can help understand how the brain works or detect any brain damage. A EEG can be done for a short period, when executing some task, or for a longer period – e.g. while sleeping (for 7+ hours).



EEG systems work by placing a number of electrodes on the scalp. These collect signals. These signals are processed through a series of steps: i) amplification, ii) band-splitting into 3 frequency ranges, iii) each of these frequency ranges are filtered for noise, iv) each of these frequency-signals then are band-filtered. The data is recorded with a resolution of milliseconds. For tasks, data can also be visualised in 'real time'.

d. Nomad Tracker: In some countries, such as e.g. Mongolia, some people live still in a nomadic lifestyle – often keeping cattle with them: they live in some area for a relatively short period of time (say weeks). During that time they build a house and let their cattle graze in the region. After some time, they move on to another place. These nomads have two challenge: i) they sometimes want to visit their friends and relatives and need to figure out where they are located, ii) when moving, they want to avoid going to locations where other nomads are housing or have recently housed (and left the area empty of grass for their cattle). Nowadays, mobile telephony has arrived and you are tasked to design an app- in which families can see i) where are their friends, and ii) which regions in their neighbourhood are or were recently 'occupied' .

Question 4 (8 pts: 4 + 4) *Design Principles & Tactics*

a. Explain the design principle of 'information hiding' through a *counter example*. Your example should include: a diagram of at least 2 component and at least one dependency. You should include an example of the signature of an interface (i.e. the method name and the parameters that it includes). Your example must be different from the examples in book and the lecture slides.
b. For systems that need a high reliability, their architecture should 'avoid a single-point of failure'. Give an example of a system that has a component that is a single-point of failure, and show an improved version of that architecture where this weakness has been addressed. You answer should include diagrams that follow UML-conventions.

Question 5 (10 pts: 2 + 2 + 2 + 2 + 2) Reverse Architecting

The objective of Reverse Architecting is to reconstruct all relevant architectural information from the source code (and possibly other available artefacts) of a system. Ideally this is done fully automatically, but often (much) manual work is needed for achieving this.

For each of the following pieces of information, describe whether they are easy or difficult to reverse engineer from the source code (briefly motivate your answer):

a. Component diagrams
b. Sequence diagrams
c. Use case diagrams
d. Principles guiding the design and evolution of the architecture
e. Describe 2 challenges that one runs into when trying to reverse engineer sequence diagrams from a running system.

Question 6 (34 pts) Architectural design

Read the description below and make an architectural design – detailed instructions are given below the case description. Points per sub-questions are mentioned at each sub-question. Motivate your choices and state any assumptions you make.

Bio-Imaging System (BIS2020)

Automation has been finding its way into bio-chemical research. One application is the automated analysis of large samples that are subjected to some 'treatment'. In this approach, a matrix of small containers is filled with biological samples (e.g. cells), and across this matrix a treatment (some chemical or bacteria) is added in different concentrations to the various samples. This samples-matrix is then stored in an environment that is controlled for temperature and for a particular amount of time. Experiments typically take 24 hours. After this time, the matrix is automatically moved into an 'image processing' chamber of the same machine. This image processing facility takes high resolution (1920x1080 pixels) photos of each container of the matrix. These photos are then transferred to an image-processing module. This module can perform various types of analysis on the photos: i) it can analyse the colours of the sample (through spectrography), ii) it can perform a count of the number of colonies of cells in the sample, iii) it can calculate the surface area of the container that is covered with cell-colonies. The system is extensible with new image-analysis algorithms.

The settings for one 'run' of a matrix are configured by the lab-researcher. These settings include: treatment (which chemical and how its concentration should range over samples), time, temperature and types of analyses to perform. These settings can be done at the machine itself or via a remote connection from a desk-top computer in the lab.

The results of these analyses then undergo reviews by human experts. Human experts view the analyses and can manually change them or add annotations. The human experts can do this through a screen that is attached to the system, but more typically these experts do this

work remotely by connection to their own desktop – which may be in a different lab- to the computer of the BIS2020 system. The BIS2020 system then produces a table with the results of the analysis. This data can be exported in various formats (such as comma-separated-file, excel) such that the researcher can upload these in his tools for doing statistical analyses (in a separate tool). All data (images, annotation, reviews) is archived on a separate computer and backed-up on a remote storage system. All data can be searched and revisited: reviews can be updated any time in the future. All versions of any updates to the reviews are archived.

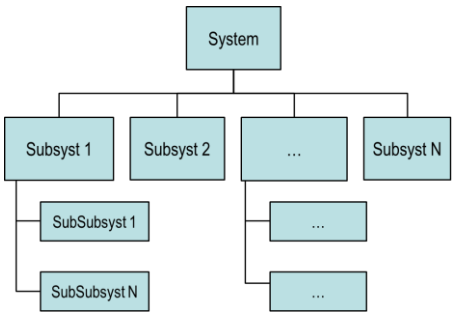
Figure: Matrix of samples, Image analyses of count of cell-colonies and area, Lab Researcher reviewing results, Report

a.(2) Stakeholders: mention 2 stakeholders of the system.

b.(3) Architectural Drivers: mention 3 architectural drivers and motivate why they are drivers.

c.(4) Use case view: represent at least 2 types of users.
Your Use case view should include at least 3 significant use cases.

d.(4) Functional decomposition: Create a functional decomposition diagram for your system. Your notation can follow this example (right hand side):
For each component, describe its main responsibility in at most 1 sentence.



e.(4) Logical/Structural view: Use a suitable UML diagram & notation to create a design of the architecture of the system.

f.(2) Describe which architectural style(s) are used in which parts of your architecture design.

g.(4) Propose a design for the API/interface (i.e. method-name(s) and parameters) for doing image processing. Keep in mind that the system should be extensible with new image-processing methods.

h.(4) Process/Behaviour view: Use a suitable UML diagram & notation to design the behavioural views of the system. For this you should address the following functionalities of the system:

1. A matrix of containers is loaded into the system, the system is configured, the samples are analysed and all generated data is stored and archived.
2. A human expert from an external lab searches for a particular run of an experiment, reviews the experiment's images and analysis-results on his own computer. He then adds an annotation which is archived in the BIS2020 system.

i.(4) Create a deployment view for your system design.

j.(3) Describe 3 main sensitivity-points of your architecture (including deployment infrastructure; i.e. compute-capacity, network-capacity and storage-capacity) if images are not only recorded at the end of an experiment run, but every 15 minutes.

Remember that the models that you create for e., h. and i. should be consistent with each other.