NTNU
Norwegian University of Science and Technology

Faculty of Informatics, Mathematics and Electronics

Department of Computer and Information Sciences

ENGLISH

Sensurfrist: 14. June

Exam in the subject
TDT4240 Software Architecture

Tuesday 24. May 2005
9:00 am – 1:00 pm

Aids code C:

Simple calculator allowed.

These specified printed documents are allowed:

- IEEE (2000), "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems", Software Engineering Standards Committee of the IEEE Computer Society.
- Kruchten, P. (1995), "The 4+1 View Model of Architecture", IEEE Software, 12(6).
- English-Norwegian dictionary (or to your native language if your not Norwegian) and/or a English thesaurus (English-English).

Contact person during the exam:

Associate professor Alf Inge Wang, phone 73594485, mobile phone: 92289577

The points show how much each problem is worth in this exam. For each problem, each question has the same weight unless otherwise stated. The exam has 4 problems giving a total of 60 points.

Good Luck!

Controlled 15th of May 2005

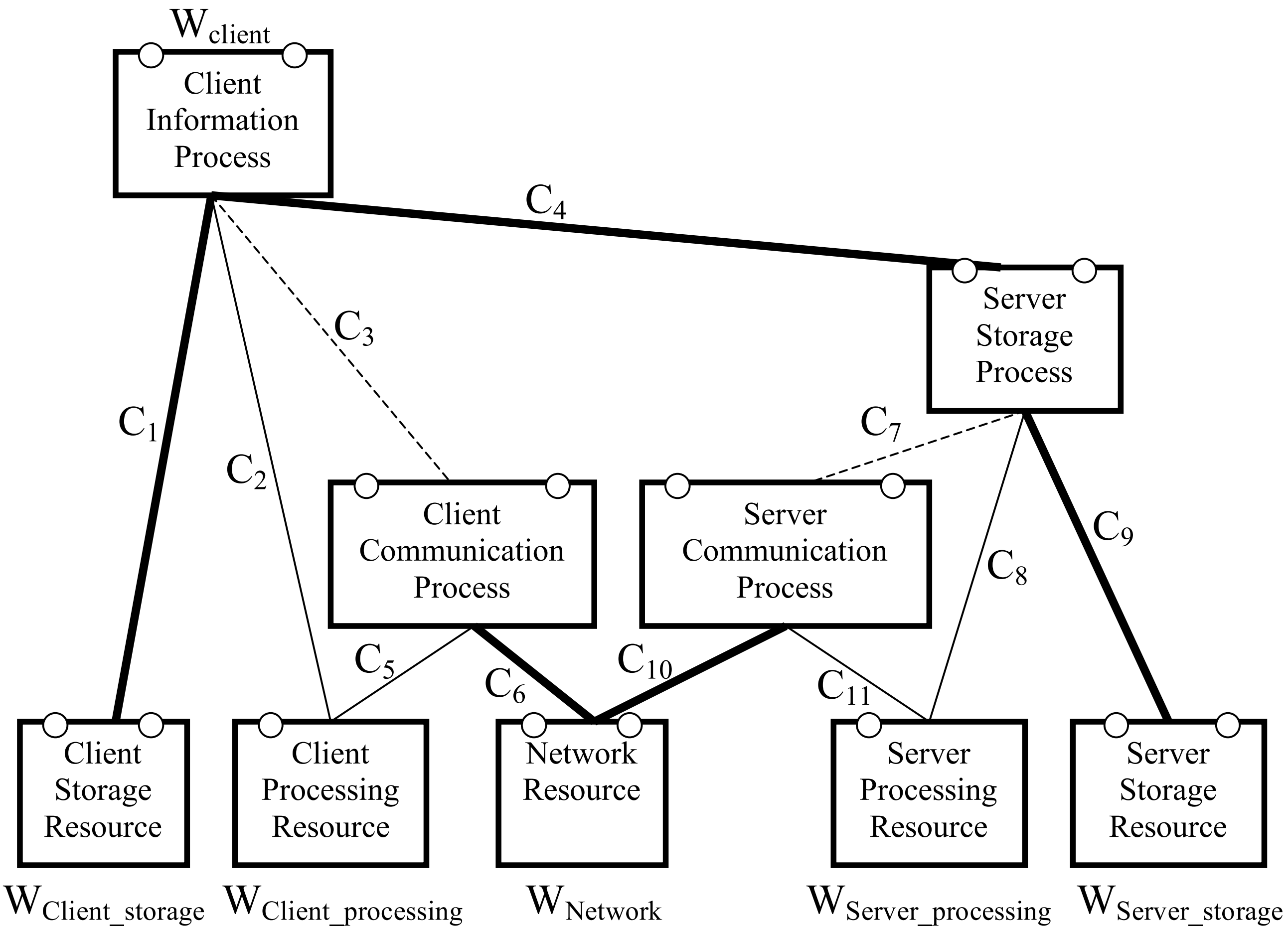
Problem 1 (10 points): Various questions

Answer these questions short:

1.1 Describe in your own words what a software architecture is?
1.2 What is the difference between architectural patterns, design patterns, and idioms?
1.3 What are the advantages of using the Abstract Factory design pattern in a software architecture?
1.4 How can different stakeholders of a software system influence the software architecture? Give examples for Developer, Marketing, End-user, Maintainer, and Customer.
1.5 What is the relationship between functionality and quality attributes?
1.6 What is a quality attribute scenario and what is the purpose of it?
1.7 What is an enterprise architecture and how does it relate to the business environment?
1.8 Describe the 3 main areas availability tactics must cover?
1.9 What is Attribute-Driven Design (ADD)?
1.10 Describe short the four steps of the reconstruction software architecture process.

Problem 2 (10 points): Software Architecture and Performance

The figure below shows a Structure and Performance (SP) diagram.



Here are the operations (small circles) for the components in the diagram:

- Client Information Process: Read article, Write article.
- Server Storage Process: Select tuple, Update tuple.
- Client Communication Process: Send message, Retrieve message.
- Server Communication Process: Send message, Retrieve message.
- Client Storage Resource: Store block, read block.
- Client Processing Resource: Perform instruction.
- Network Resource: Send package, retrieve package.
- Server Processing Resource: Perform instruction.
- Server Storage Resource: Store block, read block.

Answer the following questions:

2.1 What does the diagram above describe and what is it used for?

2.2 What does the three different relationships mean (strong/bold line, normal line and dotted line)?

2.3 What is the purpose of the operations (small circles) in the diagram?

2.4 What is the purpose of the Cs (C_1 to C_{11}) in the diagram?

2.5 Find the work performed in $W_{Network}$ as a function of W_{client}?

Problem 3 (10 points): The ATAM

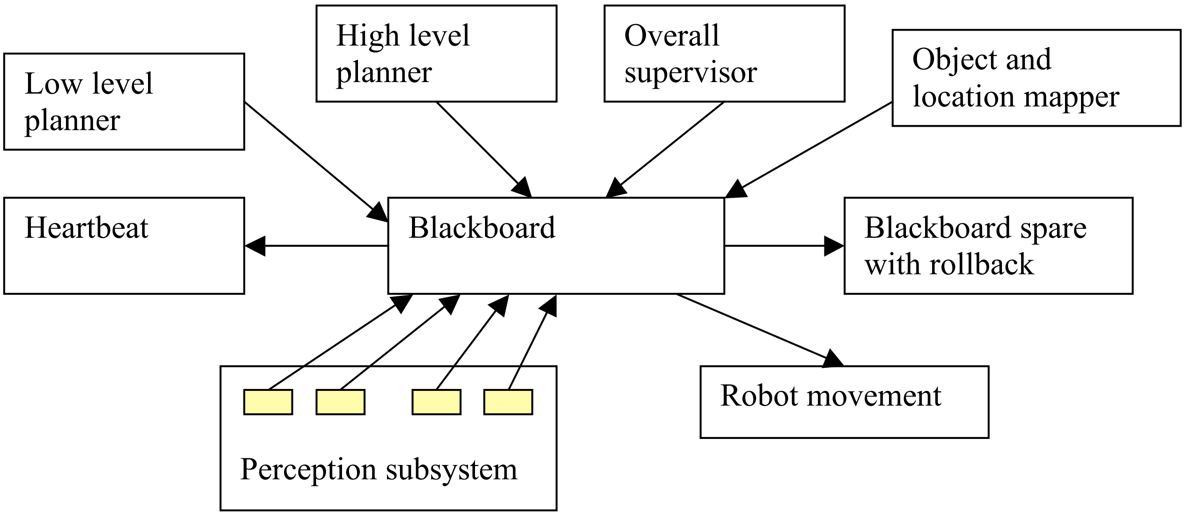
Read through the description of the robot controller architecture below and perform a *quick* ATAM evaluation of the architecture. Produce the following outputs:

- A utility tree
- Analysis of architectural tactics used
- List of sensitivity and tradeoff points
- Set of risks and non-risks
- Set of risk themes

Please motivate for your choices and state your assumptions:

Robot controller architecture

The focus of the architecture of this robot controller is on availability. A simplified illustration of the logical view is shown in the figure below:



Here is a short description of the different modules:

- *Overall supervisor*: Is the module that has the overview of the state and data of the robot and makes the decisions on what to do.
- *High level planner*: Plans high-level tasks like find ball, pick up ball, bring to light etc.
- *Low level planner*: Decomposes high-level tasks into movement commands.
- Object and location mapper: Identifies object and the surroundings of the robot.
- *Heartbeat*: Received control information from blackboard.
- *Perception subsystem*: Manages sensor information.
- *Robot movement*: Manages the motors of the robot.
- *Blackboard*: Manages the robot data.
- *Blackboard spare*: Manages the robot data.

The robot should be able to move around, pick up objects and bring the objects to the light. The downtime of the robot should not exceed 1 minute per hour, and if the robot controller goes down, it should be available within 5 seconds.

Problem 4 (30 points): Create an architecture

Read the description below and do the following:
4.1 Identify the most important quality attribute(s) for the system described below.
4.2 Identify architectural driver for the system described below.
4.3 Choose and describe suitable architectural tactics for the problem described below, and describe how the tactics affect the quality attributes.
4.4 Create architecture views of the system described below. The architecture must be described in two views according to the 4+1 view model: *Scenario view and Logical view*.

Motivate for your choice of quality attributes, architectural drivers and the architectural tactics used in your architecture.

Software for Digital Camera

The software described here is software that is used in the controller of various digital cameras. The software should be able to provide different levels of functionality depending on the price segment of the camera and the software should be able to be used for various kinds of hardware configurations (buttons, screen, data storage and optical components).
The digital camera consist of these hardware components:

- Controller (CPU, memory): managing the other components, provide interface with the user etc.
- Controller buttons (vary from camera to camera). Typically buttons for on/off, flash, take picture, menu, navigation, zoom etc.
- Digital screen (can vary from camera to camera in size, colour depth etc).
- Permanent data storage (typically flash-memory, memory stick, SD-cards etc).
- Optical component with an interface to control zoom, focus, etc....

Here is a list typical functionality provided by the camera:

- Turn on/off camera.
- User controlled optical functionality (zoom out, zoom in, flash etc).
- Camera controlled optical functionality (auto focus, lens opening etc).
- Power save functionality (shut down camera if not used etc).
- Storing, retrieve and delete pictures.
- Processing images (rotate, enhance, etc).
- Display pictures, information to the user on the camera's screen.
- Camera set up (storage options, GUI-options, language options etc.)