NTNU
Norwegian University of Science and Technology

Faculty of Physics, Informatics and Mathematics

Department of Computer and Information Sciences

ENGLISH

Sensurfrist: 22. June

Exam in the subject
TDT4240 Software Architecture

Tuesday 27. May 2004
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!

Problem 1 (10 points): Various questions

Answer these questions short:

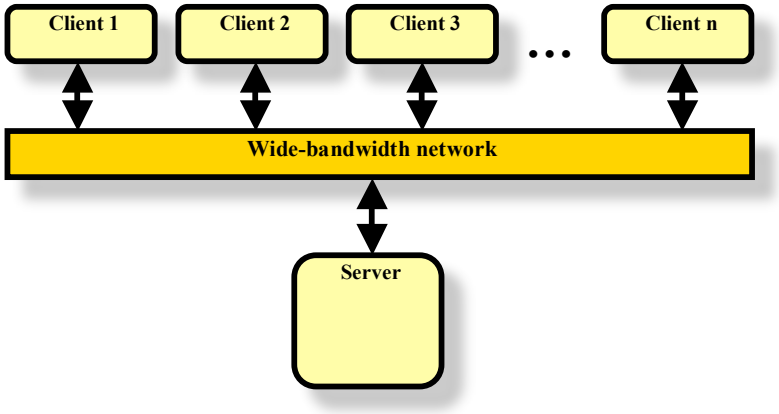
1.1 What is a design pattern?
1.2 What are the advantages by using the Singleton design pattern in a software architecture?
1.3 What is Bass, Clements and Kazman's definition of Software Architecture?
1.4 What is a variation point in a software product line?
1.5 What is a *wrapper* related to Off-The Shelves components and architecture?
1.6 What is *View fusion* in reconstructing software architecture?
1.7 Why can it be necessary to reconstruct a software architecture?
1.8 What is an architectural driver (give examples)?
1.9 What is an architectural strategy?
1.10 What is the motivation for introducing software architecture into software development?

Problem 2 (10 points): Architecture Pattern, Reference Model, Reference Architecture or Architecture

Decide if the following model examples are Architecture Pattern (AP), Reference Model (RM), Reference Architecture (RA) or Architecture (A) according to the book Software Architecture in Practice. Explain and motivate your choice.

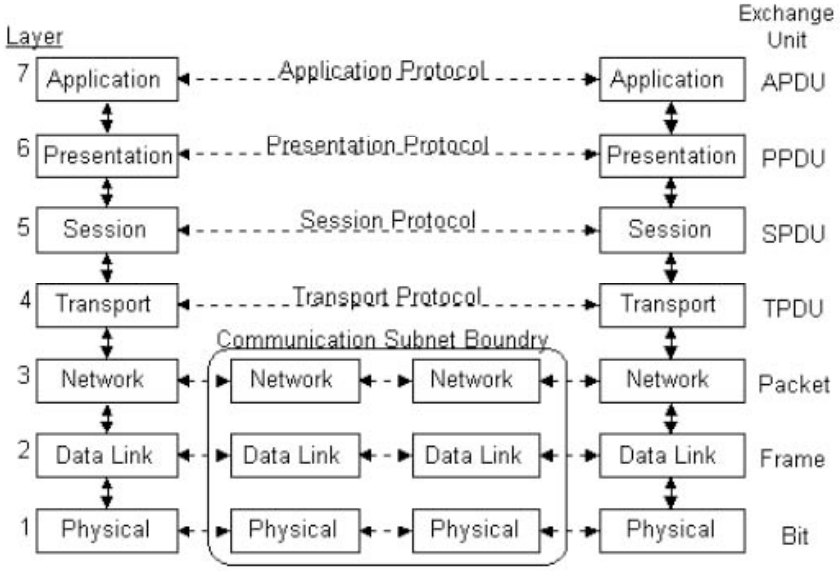
2.1 Client-server model:

Description: A model describing a server providing services to clients through a wide-bandwidth network.



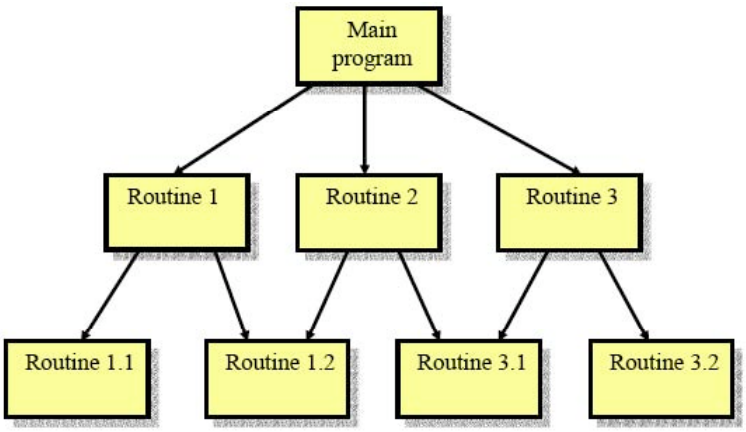
2.2 OSI model:

Description: Describing how applications can communicate through network using layers:
7) Provides services to the applications
6) Converts the information
5) Handles problems which are not communication issues
4) Provides end to end communication control
3) Routes the information in the network
2) Provides error control between adjacent nodes
1) Connects the entity to the transmission media



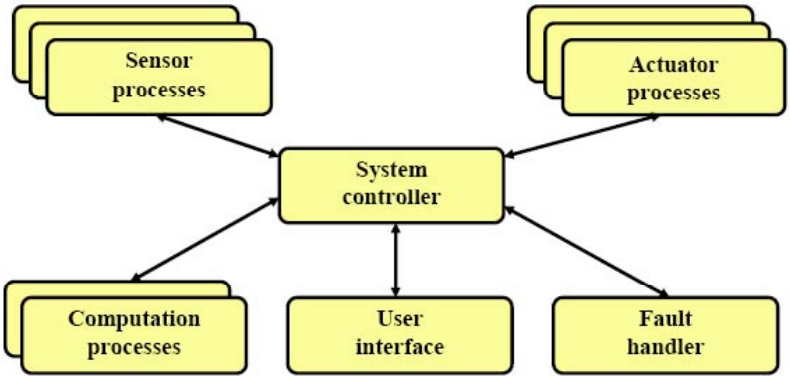
2.3 Call-return model:

Description: Top-down subroutine model where control starts at the top of a subroutine hierarchy. The model is only applicable to sequential systems. The main program can call Routines 1, 2 and 3. Routine 1 can call 1.1 or 1.2 etc.



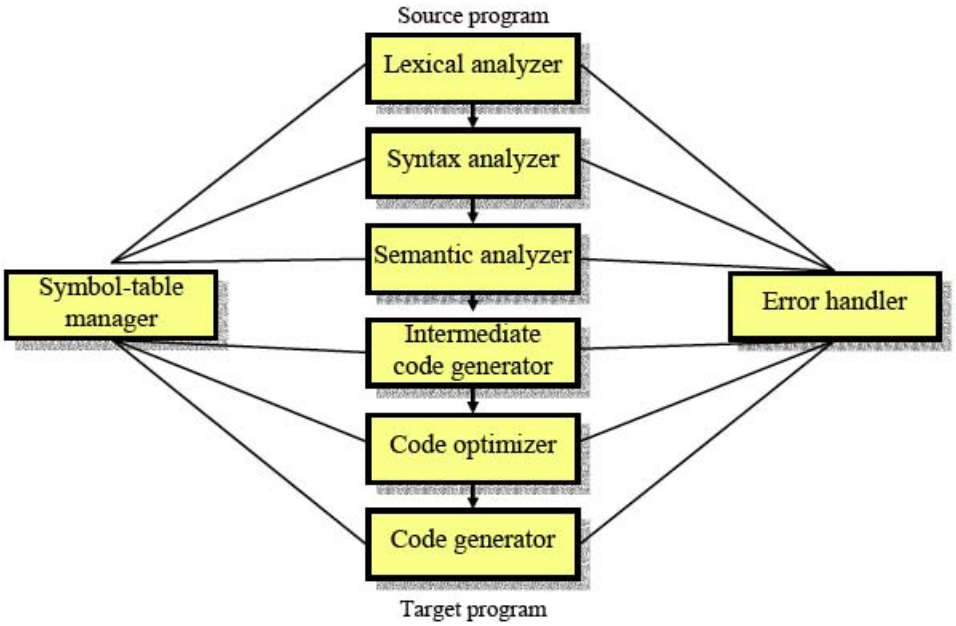
2.4 Centralized management of control model:

Description: This model is often used in "soft" real-time systems which do not have a very tight time constraints. The central controller manages the execution of a set of processes associated with sensors and actuators.



2.5 Compiler model:

Description: This model describes the phases of a compiler. The phases represent the necessary functionality to implement a compiler.



Problem 3 (10 points): The CBAM

From the table 1, 2 and 3 and the information below, compute (use straight lines between the data points in the graph):

- Total benefit obtained from the 3 architectural strategies.
- Return-On-Investment for the 3 architectural strategies and find what architectural strategy is the best investment.

The data is results from applying the Cost Benefit Analysis Method (CBAM) on a web-system for selling tickets for cinemas, theatres, music concerts, and sports events.

Total Benefit Obtained can be computed using this formula: $B_i = \Sigma (b_{i,j} \times W_j)$ where:

- B is benefit for each architectural strategy *i*
- $b_{i,j}$ is the benefit by using strategy I to its effect on scenario j
- W_j is the weight of scenario j

Table 1: Results from prioritizing scenarios with worst, current, desired and best response levels.

Scenario	Vote	Worst	Current	Desired	Best
1. Performance (simultaneous users)	20	1000	3000	5000	10000
2. Availability (server failure)	40	10 % fail	5% fail	1% fail	0 % fail
3. Availability (transactions lost)	25	4% lost	2% lost	0% lost	0% lost
4. Usability (% of users need help)	15	40%	20%	0 %	0%

Table 2: Results from assigning utility to the various scenarios.

Scenario	Vote	Worst	Current	Desired	Best
1. Performance	20	5	70	90	100
2. Availability	40	0	60	80	100
3. Availability	25	15	80	100	100
4. Usability	15	20	60	100	100

Table 3: Effect and cost of using architectural strategies.

Strategy	Scenario	Cost	Current response	Expected response
1. Increase computational efficiency	1	500	3000	4000
2. Active redundancy	2 3	1000	5% fail 2% lost	2% fail 1% lost
3. Support for cancel/undo	4	900	20%	5%

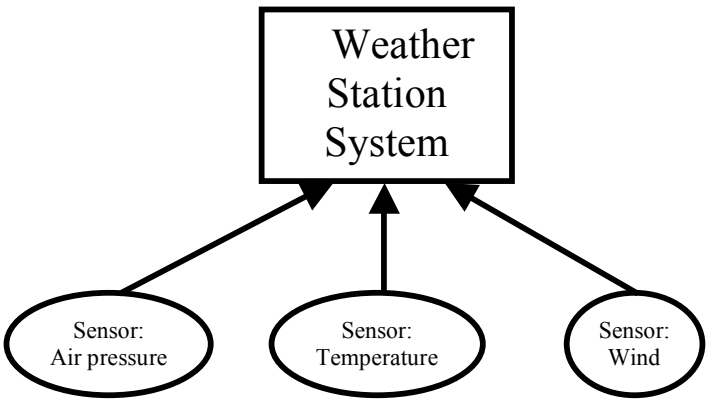
Problem 4 (30 points): Create an architecture

Read the description below and do the following:

- Identify the most important quality attribute(s) for the system described below.
- Identify architectural driver for the system described below.
- Choose and describe suitable architectural tactics for the problem described below, and describe how the tactics affect the quality attributes.
- Create a software architecture for the system described below. The architecture must be described in two views according to the 4+1 view model: Logical view and process view.
- Motivate for your choice of quality attributes, architectural drivers and the architectural tactics used in your architecture.

Weather Station System (WSS)

The weather station system (WSS) is a software system for producing weather data and weather reports gathered from physical weather sensors where the WSS is located. The WSS operates without any human operator, and gets weather data from three sensors measuring temperature, wind, and air pressure as shown in the Figure to the right.



Stakeholders of WSS can be divided into three main groups:

- *Web users* accessing weather data as web pages (HTML) provided by WSS using a web-browser.
- Web sites accessing WSS to receive weather data as XML. These web sites use the weather data to present weather information on their own web pages.
- People walking in the mountains or people in boats accessing the WSS through mobile devices (Personal Data Assistant or mobile phones) or portable PCs using the WAP-protocol on a GSM mobile network. These stakeholders are dependent on weather data from WSS in short intervals, to travel safely in the mountains or at the sea. This information is provided in Wireless Markup-Language (WML).

WSS can provide the following information to other systems (in HTML, XML or WML):

- The temperature, wind speed, and/or air pressure right now.
- The temperature, wind speed, and/or air pressure at a given time.
- Weather report (temperature, wind speed and/or air pressure) for a given time interval (e.g. from 1200-1500 every 30 minutes, 19th of May 2004).

The weather data is also used by the Norwegian Weather Association to create yearly weather reports that are used to compare weather year-by-year, month-by-month, and day-by-day. This data is also used to look at trends in global heating and other weather effects that can be results of environmental changes.