NTNU
Norwegian University of Science and Technology

Faculty of Informatics, Mathematics and Electronics

Department of Computer and Information Sciences

ENGLISH

Examination results will be announced: 2. July

Exam in the subject
TDT4240 Software Architecture

Saturday 9. June 2012
9:00 am – 1:00 pm

Aids code C:

Simple calculator allowed.
The following 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 an English thesaurus (English-English).

Contact person during the exam:

Alf Inge Wang, phone 7359 4485, mobile phone: 9228 9577

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 5 problems giving a total of 70 points. The remaining 30 points are credits awarded from the software architecture project.

Good Luck!

Meng Zhu and Tosin Daniel Oyetoyan

Controlled 28th of May 2012

Problem 1: Various questions (20 points)

Answer these questions *shortly*:

1.1	What is Bass, Clements and Kazman's definition of Software Architecture (the definition in the textbook)?
1.2	Why is graphics hardware abstraction important in game architectures according to Rollings and Morris?
1.3	What other hardware abstractions than graphics is useful in game architectures according to Rollings and Morris?
1.4	What is *fps* in a game and what parts of the game software affects the fps?
1.5	Draw a sketch of and explain the principle of decoupling game loop.
1.6	What is a token in the analysis of a game architecture?
1.7	What is the purpose of a token interaction matrix in a game architecture analysis?
1.8	What are the three main parts of Perry and Wolf's model of software architecture?
1.9	Why is software architecture important according to the textbook?
1.10	Write the three main areas of tactics described in the textbook related to performance.
1.11	What is the main advantage of using the *Composite* design pattern in a software architecture?
1.12	Give examples of typical response measures for the quality attribute usability.
1.13	What is architectural drift?
1.14	Draw a sketch of the architecture business cycle according to the textbook.
1.15	What is an architectural pattern?
1.16	How are wrappers used in development with Off-The-Shelves components according to the textbook?
1.17	Give five examples of important architectural drivers for a massive-multiplayer game such as World of Warcraft.
1.18	Describe the Attribute-Driven Design Process as described in the textbook.
1.19	What is the purpose of the IEEE1471 standard?
1.20	What is a design pattern?

References:

- Bass, Clements & Kazman: "Software Architecture in Practice"
- Rollings and Morris: "Game Architecture and Design – A New Edition"
- Perry & Wolf: "Foundation for the Study of Software Architecture"
- Coplien: "Software Design Patterns: Common Questions and Answers"
- IEEE: "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems"

Problem 2: Choose most appropriate design pattern (5 points)

Nominees:

a) Singleton
b) Abstract Factory
c) Factory method
d) Composite
e) Observer
f) Template method

Choose the *most appropriate design pattern* (one) for dealing with the 5 problems described below. Motivate for your choices (write reasons for choosing a pattern):

1. Need a mechanism to provide easy management of several configurations of look and feel in a software framework for graphical user interfaces that provides the same core functionality (buttons, windows, text-entry fields etc.).
2. Need a mechanism to manage various shapes in a graphics editor where complex shapes (e.g. a rectangle) are made out of simple shapes (e.g. lines).
3. Need a mechanism to notify clients when a server finds news gathered from the web related to the clients' specifications.
4. An application for a travel agency to manage packages of trips should be developed. You should provide a mechanism to allow all the trips follow the same structure (transport to location, visiting a location, transport back), but each trip package provides variations in transports (to and back), and locations.
5. Provide a global logging mechanism that can be accessed by all parts of a system.

Problem 3: Software Process and Software architecture (5 points)

Describe briefly how software architectural practices and concerns affect the following phases of the software process. Give concrete examples from the software architecture project in the TDT4240 course.

a) Requirement phase
b) Design phase
c) Evaluation phase
d) Implementation phase
e) Testing/Validation phase

Problem 4: CBAM (10 points)

From Table 1, 2 and 3 and the information below, find:

a) **Total benefit** from the 3 architectural strategies
b) **Return-On-Investment** for the 3 architectural strategies
c) **Rank the 3 architectural strategies** according to best investment.

Use straight lines between the data points in utility-response curves.

The data is results from applying the Cost Benefit Analysis Method (CBAM) on an information system for NTNU named TWOTS (Total Waste of Time System) for managing students, teachers, courses, equipment etc.

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

Scenario	Vote	Worst	Current	Desired	Best
1. Availability: Downtime per week	30	60 min	10 min	2 min	0 min
2. Performance: Response time on user requests	20	5 sec	3 sec	1 sec	1 sec
3. Performance: Number of simultaneous user transactions per sec	15	100	500	1500	5000
4. Modifiability: Time to add support for short courses for companies	10	30 days	15 days	1 day	1 day
5. Usability: Average mistakes out of correct operations made by users	20	30%	10%	1%	0%

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

Scenario	Vote	Worst	Current	Desired	Best
1. Availability	30	5	40	80	100
2. Performance	20	10	60	100	100
3. Performance	15	5	30	50	100
4. Modifiability	10	5	20	100	100
5. Usability	20	10	40	80	100

Table 3: Effect and cost of using architectural strategies.

Strategy	Scenario	Cost	Current response	Expected response
1. Replicated servers (hardware and software)	1 2 3	100000	10min 3sec 500	4min 2sec 1000
2. Faster hardware	2 3	20000	3sec 500	2sec 750
3. Improved exception handling	1	5000	10min	8min

Problem 5 Design a software architecture (30 points)

Read the description of AI-Drillo and do an architectural design. Your answer must include:

a) Architectural drivers – 2 points
b) Architectural tactics and patterns – 3 points
c) Process view – 5 points
d) Logical view – 17 points
e) Architectural rationale – 3 points

Motivate for your choices and state your assumptions.

Software for managing a computer-controlled soccer RC car - AI-Drillo

AI-Drillo is computer-controlled car equipped with four distance sensors at each corner of the car and a bumper in front that can hold the ball and be activated to shoot the ball forward. AI-Drillo has also a front camera that can recognize a soccer ball, sidelines of the soccer field and the goals. Further, the car has activators to control the speed of the engine transferred to wheels in the back of the car (forward and reverse), as well as controlling the angle on the front wheels (for turning). It should be possible to use various activators to enable different engines and steering mechanics for the car. The wheels also have a counter that increases by one every time the left front wheel has turned 360 degrees. The goal of the car is to drive around the field and find a ball, then take the ball to the correct goal and score. The car should stop when it has scored 5 goals. Patterns on the sidelines will help the car to find the right goal.

The software of AI-Drillo should manage driving around the soccer field, avoiding other cars or driving out of the field, getting the ball when it is detected, find the goal, and score. Several balls are out in the field for the car to pick up.

It should be possible to replace sensors and activators of the car. The AI-Drillo is a prototype where it must be possible to change the behavior of the car at runtime by updating/replacing software component while the car is running.

Here is a conceptual illustration of the AI-Drillo RC car: