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: 1. September

Continuation Examination in the subject TDT4240 Software Architecture

Saturday 9. Aug 2008
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:

Not available

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!

Problem 1: Various questions (15 points)

Answer these questions in short:

1.1 What is Bass, Clements and Kazman's definition of Software Architecture (the definition in the textbook)?
1.2 What is the purpose of the ATAM architecture evaluation?
1.3 What is an architectural pattern (style)?
1.4 What is architectural erosion?
1.5 Describe the *Composite* design pattern and advantages of using it.
1.6 Describe the CBAM process (9 steps)
1.7 What is the purpose of Utility-Response Curve in CBAM?
1.8 How do you calculate the total benefit obtained from an architectural strategy in CBAM?
1.9 Describe the *process* in the textbook for reconstructing a software architecture.
1.10 What kind of information can be useful to extract to reconstruct a software architecture?
1.11 Name the different types of architectures that are included in an enterprise architecture according to Telenor (Sigurd Thunem)?
1.12 Name the three main areas the textbook describes for availability tactics.
1.13 What is the purpose of the GRASP (design) patterns?
1.14 What is a (design) pattern language?
1.15 What are the three techniques described in the textbook for repairing interface mismatch?

Problem 2: Choose the correct architectural pattern (10 points)

Buff Tore is hired as a software architect in the Pear software company, where his job is to choose the correct architectural pattern for the following two systems Pine and Smash. Buff Tore can choose among the following architectural patterns:

1. Pipe and filter
2. Layered
3. Blackboard
4. Task Control
5. NASREM

2.1 Choose and motivate for your choice of architecture pattern for the Pine system (5 points)

The Pine system should have the following characteristics:

- The main parts of the system will remain the same for many years (stable).
- The Pine system should be possible to tailor for each customer, typically the user-interface and look & feel of the system.
- The Pine system should be possible to run on various operating systems and hardware platforms.

2.2 Choose and motivate for your choice of architecture pattern for the Smash application (5 points)

The Smash system is an application for changing audio in real-time by applying audio effects. The system has the following characteristics:

- Several audio effects can be applied to the input audio one after another.
- The implementation of the audio effects might be changed to improve performance.
- New audio effects might be added to the application in later releases.

Problem 3: ATAM (5 points)

Do the step 6 (Analyze the architectural approaches) in the ATAM process based on the following information about a system for selling tickets over the web:

Utility tree:

- *Usability:*
  - *Scenario U1. The client user should be able to correctly use 90% of the functionality of the application after using the system for 5 minutes (M,H).*
- *Performance:*
  - *Scenario P1: 10000 simultaneous users should have a response time less than 1 second under normal operation (H,M).*

Identified architectural tactics:

- *Reduce the number of events processed* – when more than 8000 users use the system at the same time, the number of users above 8000 will get lover processing priority.
- *Replication of server PCs* – The server application is replicated on two PCs to process more user requests.
- *Scheduling of resources* – the scheduling of processing resources is based on first come – first served.

Problem 4 (10 points): The CBAM

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

- **Total benefit** obtained from the 3 architectural strategies.
- **Return-On-Investment** for the 3 architectural strategies
- **Rank the 3 architectural strategies** according to best investment.

Use straight lines between the data points in the graph.

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

Scenario	Vote	Worst	Current	Desired	Best
1. Performance (Time to pick up and deliver 4 balls to the light)	30	20min	10min	4min	3min
2. Availability (How many times the robot get stuck during the mission picking up balls)	20	30 % fail	10% fail	4% fail	0 % fail
3. Availability (How many times the robot controller software crashes)	15	20% crash	5% crash	0% lost	0% lost
4. Modifiability (Time to add readymade module)	20	60min	30min	15min	5min
5. Testability (Time to test robot movements)	15	5hours	2hours	20min	5min

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

Scenario	Vote	Worst	Current	Desired	Best
1. Performance	30	30	60	90	100
2. Availability	20	20	70	90	100
3. Availability	15	30	80	100	100
4. Modifiability	20	15	60	70	100
5. Testability	15	5	30	90	100

Table 3: Effect and cost of using architectural strategies.

Strategy	Scenario	Cost	Current response	Expected response
1. Use route-planning	1	2000	10min	7min
2. Improved wall detector	2 3	800	10% fail 5% crash	6% fail 5% crash
3. Improved exception handling	2 3	400	10% fail 5% crash	10% fail 2% crash

Problem 5 Create an architecture (30 points)

Read the description below and do the following:

5.1 Identify the ***architectural drivers*** for the system described below (5 points)

5.2 Choose and describe suitable ***design and/or architectural patterns*** for the problem described below, and describe how the patterns affect the quality attributes (5 points)

5.3 Create ***architecture views*** of the system described below. The architecture must be described in two views according to the 4+1 view model: *Logical and Scenario view* (20 points)

Motivate for your choice of quality attributes, architectural drivers, design patterns and the architectural patterns used in your architecture. State your assumptions.

Software for a simple digital video camera

The software described here is software that is used in the controller of various simple video cameras. The software should be able to provide different levels of functionality depending on the price segment of the video camera and the software should be able to be used for various kinds of hardware configurations (buttons, screen, data storage and optical components).
The video 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, video record, video player, night shot, menu, navigation, zoom etc.
- Digital screen (can vary from camera to camera in size, colour depth etc).
- View finder (small screen in the back of the camera).
- Permanent data storage (typically hard drive, flash-memory, memory stick, SD-cards etc).
- Optical component with an interface to control zoom, focus, etc....

Here is a list typical functionality offered by the video camera:

- Turn on/off camera.
- Playback video on screen.
- Record video to permanent data storage.
- User controlled optical functionality (zoom out, zoom in etc).
- Camera controlled optical functionality (auto focus, lens opening etc).
- Power save functionality (shut down camera if not used etc).
- Storing, retrieve and delete videos.
- Processing video effects.
- Display information to the user on the camera's screen.
- Video camera set up (storage options, GUI-options, language options etc.)