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: 14. June

Solution Exam in the subject TDT4240 Software Architecture

Tuesday 24. May 2011
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:

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 Bian Wu

Controlled 15th of May 2011

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	Write some key characteristics of agile software development.
1.3	What is domain-driven design?
1.4	What were the main findings in the paper "Agile Enterprise Software Development Using Domain-Driven Design and Test First"?
1.5	Describe two important differences between computing hardware architecture and software architecture described in Perry & Wolf.
1.6	Describe two similarities between building and software architecture.
1.7	Which three classes of architectural elements are described by Perry & Wolf?
1.8	Write five elements a pattern description should contain in the format Alexandrian form (Coplien).
1.9	What is a pattern language according to Coplien?
1.10	Write the three main areas of tactics described in the textbook for achieving high availability.
1.11	What is the main advantage of using the Singleton design pattern in a software architecture?
1.12	Give examples of typical response measures for the quality attribute availability.
1.13	What is architectural erosion?
1.14	What are the main advantages of using ATAM in a development project?
1.15	What is a variation point in a software product line?
1.16	How are mediators used in development with Off-The-Shelves components?
1.17	What is an architectural driver (give examples)?
1.18	Describe how the quality attribute usability can be related to software architecture?
1.19	What is the relationship between a reference model and a reference architecture?
1.20	Describe shortly the four steps in the process of reconstructing a software architecture as described in the textbook.

References:

- Perry & Wolf: "Foundation for the Study of Software Architecture"
- Coplien: "Software Design Patterns: Common Questions and Answers"

Solution:

1.1 A software architecture of a program or computing system is the structure(s) of the system, which comprise of software elements, the externally visible properties of those elements, and the relationships among them.

1.2 Key characteristics of agile development: Continuous customer feedback, embraces change, test-driven design, focus on working code – not written documents, automated build and testing.

1.3 Domain-driven design is design where the primary focus is on domain and domain logic, and that it should be based on a model.

1.4 Statoil found that that using a new object-oriented architecture using test first and domain-driven design techniques improved the software architecture with respect to performance and code quality, and that the switch to an object database gave additional performance gains.

1.5 Two important differences between computer hardware architecture and software architecture are that for hardware there are a relatively small number of design elements, and scale is achieved by replication of these design elements. For software architecture, there is a large number of possible design elements, and scale is usually achieved through adding specific design elements.

1.6 Similarities between building and software architecture: Multiple views, architectural styles (pattern), style & engineering, style & materials (the two first are most important).

1.7 Three classes of architectural elements: Processing elements, data elements and connecting elements).

1.8 Elements in a pattern description according to Alexandrian form: Pattern name, Problem to solve, Context, Forces or tradeoffs, Solution, Examples, Force resolution or resulting context, and Design rationale.

1.9 A pattern language is a structured collection of patterns that build on each other to transform needs and constraints into an architecture.

1.10 Areas of availability: Fault detection, fault recovery, and fault prevention.

1.11 Advantage of using Singleton is that you can control the number of instances for a specific class (e.g. classes that hold the state of one system or information that must be correct across the system).

1.12 Availability response measures: Time interval the system must be available, availability time, % of uptime, Repair time, Maximum failure time, etc.

1.13 Architectural erosion means that the implementation does not follow the architectural design.

1.14 The main advantages of using ATAM is that assurance that the architecture (master-plan) is correct, better to find problems early than late, unsuitable architecture will be a disaster for a project, force stakeholders to meet, evaluate that the system meet user quality requirements.

1.15 A variation point is the part of parts of an architecture where it is needed to support variation in terms of inclusion/omission of elements, replicate elements, and special versions of elements.

1.16 Mediators are used to repair interface mismatch between OTS and the rest of the architecture by finding what is needed to repair in runtime through planning functions.

1.17 An architectural driver is the most important issues that will have deepest impact on the software architecture and it can be functional, quality and business requirements. Examples of architectural drivers can be the system must be developed in XNA, the system should be easy to extend new functionality, the system must be implemented in 10 weeks, etc.

1.18 Usability can be related to software architecture in terms of usability issues that will impact the structure of components in the system, such as undo/redo functionality, keeping model of the user, the program state, etc.

1.19 A reference architecture is the combination of a reference model (describing the functionality of a system) and an architectural pattern (describing the software component structure).

1.20 Reconstruction process: 1) Information extraction (extract information from various sources e.g. source files, documentation, executables, file structure etc.), 2) Database construction (convert information from step 1 to a database format), 3) View fusion (combine information in the database to improve data and remove ambiguities), and 4) Reconstruction (build abstractions and representations using the database to query hypothesis about patterns)

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

Nominees:

a) Model-view-controller
b) Pipe-and-filter
c) Layered
d) Blackboard
e) Hierarchical task tree
f) Control Loop
g) Peer-to-peer

Choose the *most appropriate architectural pattern* (one) for these 5 short descriptions of systems. Motivate for your choices:

1. Software for a system consisting of sensors for temperature, light, humidity and air pressure that will open or close windows in a building depending on the sensor readings.
2. Software for taking raw data from seismic samples and transforming the data through several stages until the data can be used for visualization.
3. Software for managing and representing the artificial intelligence (AI) of a strategy game where the strategy is represented in several levels of complexity from atomic actions up to strategic plans.
4. Software agent system where the software agents communicates through a repository by storing information objects, reading information objects, updating information objects and deleting information objects.
5. A system for managing air traffic that consist of an user interface that uses flight management and sector management that uses a set of classes providing general aeronautical services that uses general support mechanisms such as network, storage, resource management etc.

Solution:

1. Control loop
2. Pipe-and-filter
3. Hierarchical task tree
4. Blackboard
5. Layered

Problem 3: ATAM (5 points)

Do the step 6 (Analyze the architectural approaches) using the ATAM process based on the following information about an online-store for video games named Paystation network..

Utility tree:

- Security:
    - Scenario S1: It should be less than 1% chance to get unauthorized access to customer data. (L,M).
    - Scenario S2: 95% of unauthorized intruders should be identified (M,H).
- Availability:
    - Scenario A1. The system must be available 95% of the time (M,H).
- Usability:
    - Scenario U1: The user should be able to learn all Paystation network features within 10 minutes (M,H)

Identified architectural tactics:

- AT1: The server should have a hot-restart within 30 seconds if it fails.
- AT2: The server should have watchdog to check if the servers is running.
- AT3: Schedule time-critical components wisely.
- AT4: Structure the system to have semantic coherence.
- AT5: Use information hiding.

Solution:

Of the four scenarios S1, S2, A1, and U1 only A1 is needed to be analyzed, as there are no tactics for security and usability. Also only the tactics AT1 and AT2 need to be considered, as there are now scenarios for performance (AT3) and modifiability (AT4 and AT5).

Scenario#: A1 Scenario: The system must be available 95% of the time (M,H)
Attribute(s): Availability
Environment: Normal operations
Stimulus: External or internal fault
Response: Restart server in consistent state
Response measure: The system should be available 95% of the time.

Architectural Decisions:	Sensitivity	Tradeoff	Risk	Nonrisk
AT1: Hot restart		T1	R1	
AT2: Watchdog	S1			N1

Reasoning: The tactics AT1 and AT2 will help to make the system more available, but only if the system itself is very stable and protected against external stimulus that can bring the system down.

Architectural diagram: None

Sensitivity points, Tradeoff points, Risk and Nonrisk:

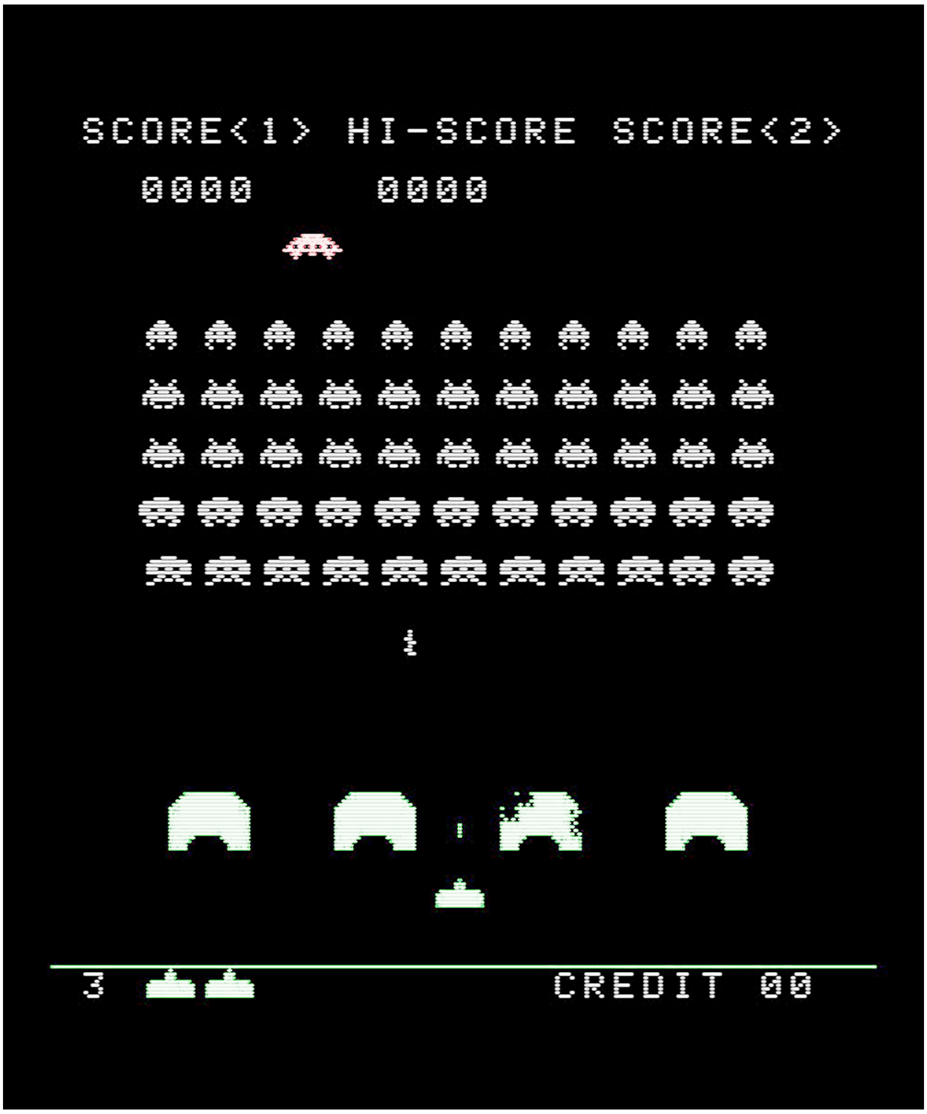
S1	AT2 is only positive for availability
T1	AT1 is positive for availability, but could impact performance as it is necessary to continuously store a consistent state the system could be restored to.
R1	AT1 could be a risk to the project due to issues related to performance.
N1	AT2 should be a safe nonrisk decision.

Risk themes: None

Problem 4: Creating a Game Architecture (10 points)

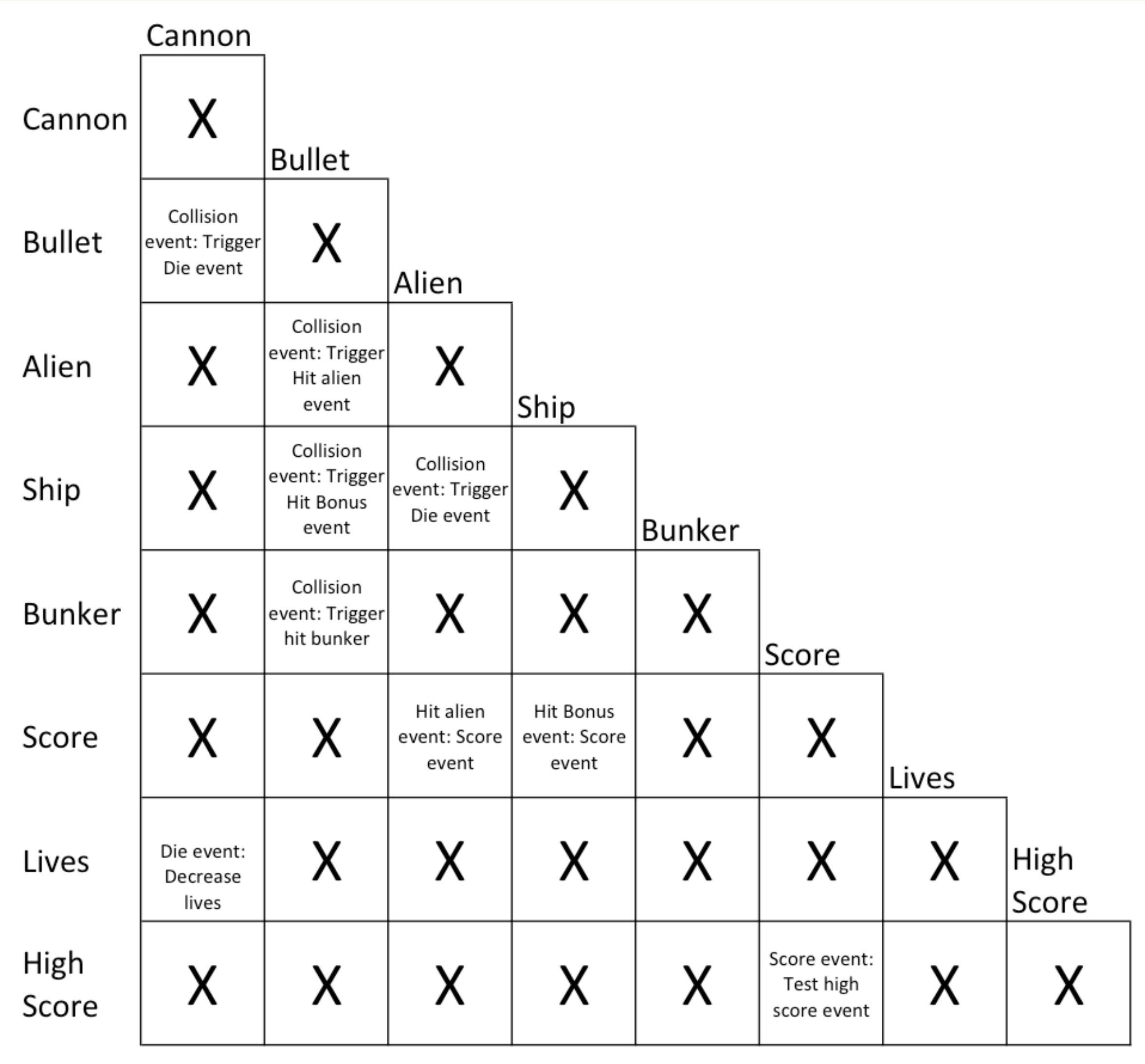
Use the first two steps (1. Find tokens, 2. Analyze interaction and events) in the method described in “Creating a Game Architecture” (Rollings & Morris) to analyze the architecture for the game described below.

Description of the game Space Invaders (see screenshot of the game below):
Space Invaders is a two-dimensional fixed shooter game in which the player controls a laser cannon by moving it horizontally across the bottom of the screen and firing at descending aliens. The aim is to defeat five rows of eleven aliens that move horizontally back and forth across the screen as they advance towards the bottom of the screen. The player defeats an alien, and earns points, by shooting it with the laser cannon. As more aliens are defeated, the aliens' movement and the game's music both speed up. Defeating the aliens brings another wave that is more difficult, a loop which can continue indefinitely. The aliens attempt to destroy the cannon by firing at it while they approach the bottom of the screen. If they reach the bottom, the alien invasion is successful and the game ends. A special "mystery ship" will occasionally move across the top of the screen and award bonus points if destroyed. The laser cannon is partially protected by several stationary defense bunkers—the number varies by version—that are gradually destroyed by projectiles from the aliens and player.



Solution:

1. Find tokens: laser cannon, bullet, alien, mystery ship, defense bunker, lives, score, and high score.
2.



	Cannon	Bullet	Alien	Ship	Bunker	Score	Lives	High Score
Cannon	X							
Bullet	Collision event: Trigger Die event	X						
Alien	X	Collision event: Trigger Hit alien event	X					
Ship	X	Collision event: Trigger Hit Bonus event	Collision event: Trigger Die event	X				
Bunker	X	Collision event: Trigger hit bunker	X	X	X			
Score	X	X	Hit alien event: Score event	Hit Bonus event: Score event	X	X		
Lives	Die event: Decrease lives	X	X	X	X	X	X	
High Score	X	X	X	X	X	Score event: Test high score event	X	X

Problem 5 Design a software architecture (30 points)

Read the description of the SMP system below and do an architectural design. Your answer must include:

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

Motivate for your choices and state your assumptions.

Software for Surveillance and Monitoring of Patients (SMP)
The SMP hardware unit is used at hospitals to monitor the health of patients. The unit receives signals from various sensors e.g. pulse, temperature and blood pressure, which will be displayed on a LCD display, a screen or printed out on a on-device mini printer. The unit can be operated by buttons, a keyboard or a touch-based screen, depending on the version of the unit the customer buys. The operator can set values for when alarms should be fired off and enter data about the patient. The hardware unit will use sensors from various vendors. At given extreme values specified by the operator or by default values, the unit will initiate an alarm using text or symbols on the screen and high sound to warn that the patient is in a life-critical state and need immediate help. A more expensive version of the unit will also notify nurses or doctors by sending an SMS to registered cell phone numbers.

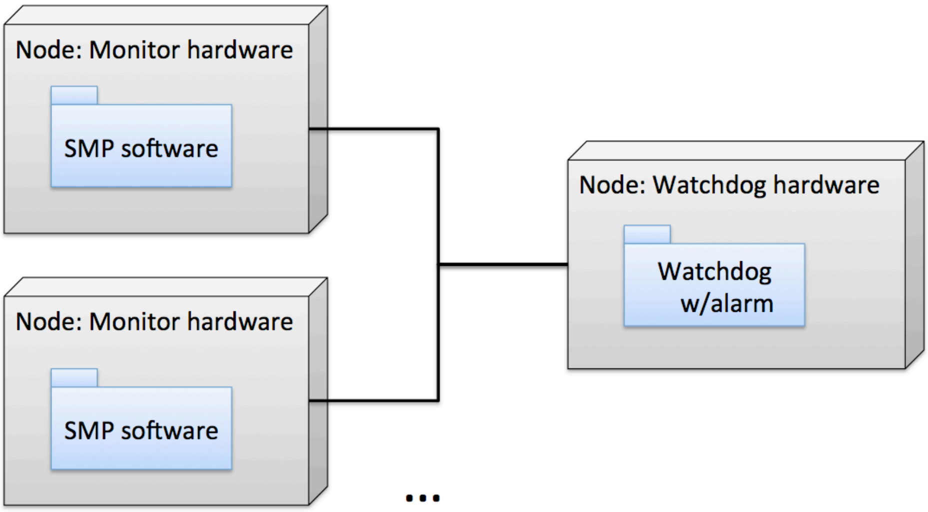
Here one example of how a SMP hardware unit can look like:

Solution:

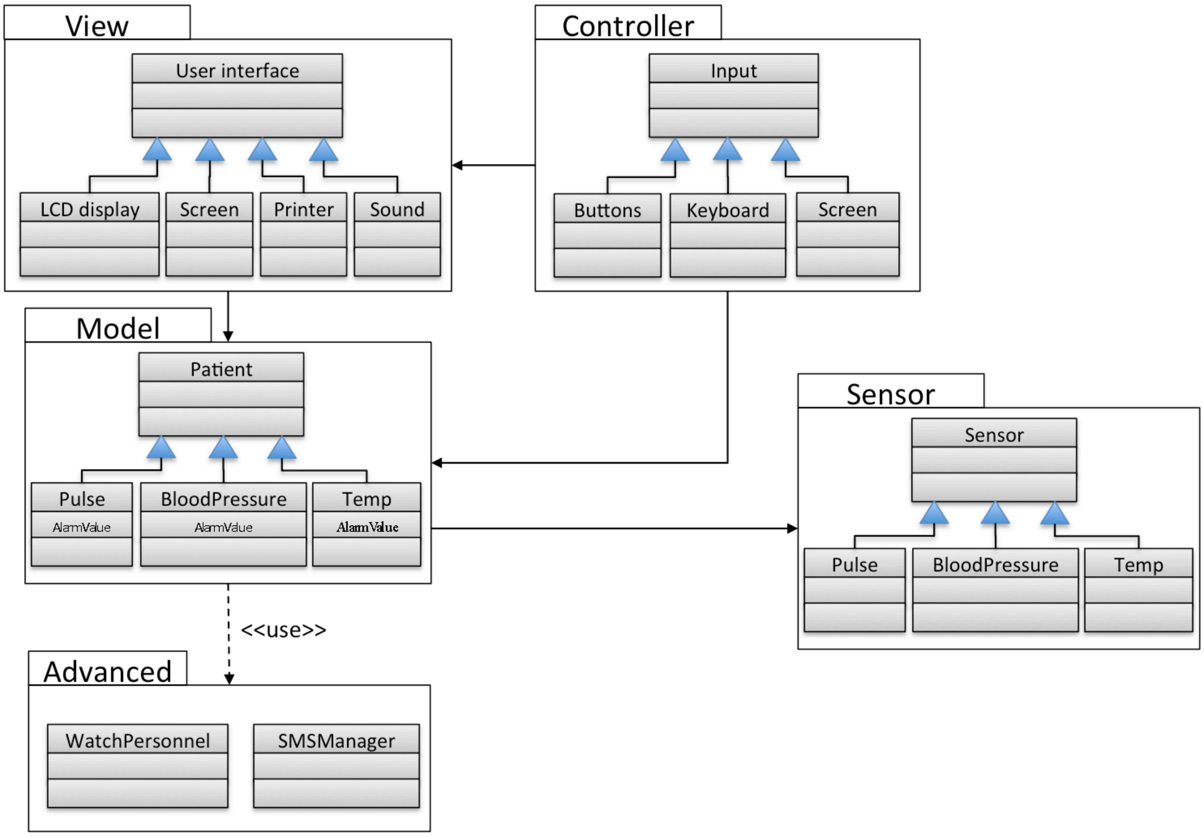
a) Architectural drivers: The most important architectural drivers are availability (very important as the device is used to monitor life-and-death situations), and modifiability (different configurations in terms of sensors, functionality, input and output)

b) Architectural tactics and patterns: It is very important to use tactics and patterns that promote modifiability and availability. To ensure that the unit is running it needs some ping-echo or heartbeat mechanism from an external hardware unit to know that the system is alive. Further it needs a mechanism to restart the system if it is detected to if it is not working, and fire off an alarm if the system cannot be restarted. Further the whole system must use all means available to ensure to make a system with as little defaults as possible and catch defaults in the system if found (catch exceptions etc). For the modifiability, we can apply the model-view-controller pattern as the input and output varies a lot from configuration to configuration. Further, we should apply an architectural pattern that allows variations in output, input, sensors, and a possibility to add the advanced SMS functionality.

c) Physical view: The physical view consists of two types of hardware nodes connected through wired or wireless network: The monitor hardware running the SMP software and a separate watchdog unit that will listen for alive signals from the monitor units. If it does not get a signal, it will try to restart the monitor unit with a restart signal to the monitor hardware, or else fire off an alarm.



d) Logical view: The logical view uses the model-view-controller as a start-point and has added support for sensors in addition. Further an additional package for providing the advanced functionality of sending SMS to personnel on watch.



e) The architecture was design for high availability and high modifiability in mind. The high availability is achieved through the watch-dog configuration along with the fast restart or alarm if the unit does not work. In addition, the system should be implemented carefully to avoid defects that can cause the unit to crash. Modifiability is achieved mainly through the model-view controller (as both the view and controllers can be different for different version of the system. In addition, there is an advanced package that takes care of the more expensive units with SMS-warning support.