NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Sciences

Examination paper for TDT 4242
Software Requirements and Testing

Academic contact during examination: Tor Stålhane

Phone: 73594484

Examination date: June 7, 2014

Examination time: from 09:00 AM to 1:00 PM

Permitted examination support material: Code A
Pocket calculators allowed
All printed and handwritten material is allowed

Other information: -

Language: English

Number of pages: 4

Number of pages enclosed: 2

Checked by:

____

Date Signature

Introduction

In this exam you can score a maximum of 50 points. The remaining 50 points for the semester comes from the compulsory exercises.

If you feel that any of the problems require information that you do not find in the text, then you should

- Document the necessary assumptions
- Explain why you need them

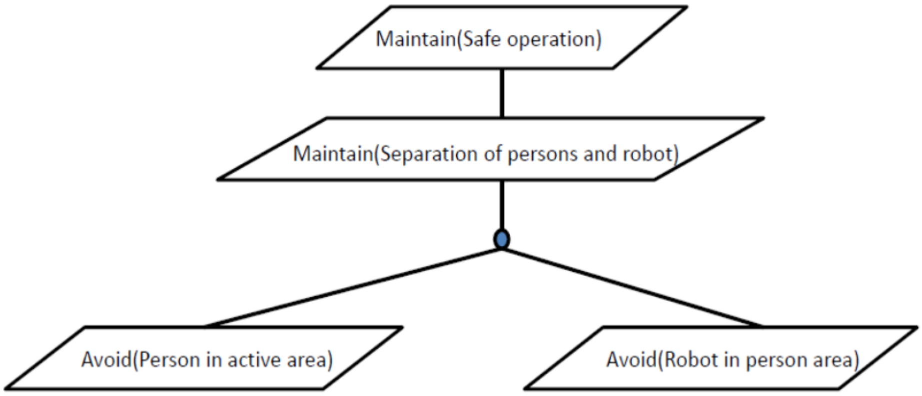
Your answers should be brief and to the point.

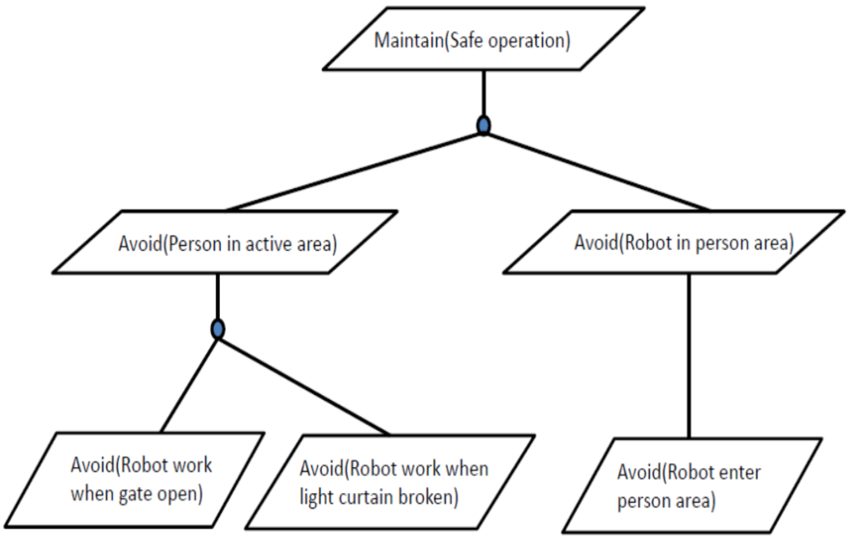
Problem 1 – Requirements engineering (20 points)

1a – Temporal patterns – 10 points

1. Use the defined set of temporal patterns – see appendix 2 – to describe the three upper levels of the requirements to a control unit for a tool cell.

There are several ways to draw these diagrams. The diagrams below are just examples. The circle used in the diagrams is important since it signifies "AND" instead of "OR"





2. Explain the difference between “Maintain(Condition) “ and “Achieve(Condition)”.

Maintain(Condition) => keep the current condition
Achieve(Condition) => move from the state you are in to the described condition

1b – Textual use cases – 10 points

1. Convert the requirements in appendix 1 to textual use cases

Alternative 1: separate use cases

System state	Robot is in area A, gate A is closed	
No	User	Tool cell control system
1	Opens gate A	Robot stops within 20 ms

System state	Robot is in area B, gate B is closed	
No	User	Tool cell control system
2	Opens gate B	Robot stops within 20 ms

System state	Light curtain is not broken	
No	User	Tool cell control system
3	Breaks light curtain	Robot stops within 20 ms

System state	Robot is in area B and both gates are closed	
No	User	Tool cell control system
4.1	-	Detect robot moves into area A
4.2	Opens gate A	Robot stops within 20 ms

System state	Robot is in area A and both gates are closed	
No	User	Tool cell control system
4.1	-	Detect robot moves into area B
4.2	Opens gate B	Robot stops within 20 ms

System state	Robot has stopped for whatever reason	
No	User	Tool cell control system
6, 7	Checks that • no person is in area A or area B • both gates are closed • no person breaks the light curtain	-
6, 7	Push the start button	The robot resumes the job it was working on when it was stopped

Requirements 5, 8 – 11 (and may be requirement 7) are not suited for textual use cases as they are not about user or system behaviour.

Alternative 2: one use case

No.	User	Tool cell control system
1,2	User opens gate to the area where the robot	Robot stops within 20 ms

	is currently working	
3	***User breaks light curtain***	***Robot stops within 20 ms***
4.1	***User opens a gate to the tool cell***	***Robot moves into user occupied area – indicated by robot rail switch***
4.2		***Robot stops within 20 ms***
6.1	***Gates A and B are closed and the light curtain is unbroken***	***-***
7	***Checks that no person is inside the tool cell***	***-***
6.2	***Pushes the start button***	***Robot starts working again***

Problem 2 – Testing methods (15 points)

Appendix 3 shows the state diagram for a seatbelt controller. The controller gives an alarm if a person sits down in the seat and has not buckled up within a certain time. If the alarm is activated, it will keep on until the seatbelt is fastened or the person leaves the seat.

2a – Methods choice – 5 points

It is important for the car manufacturer that the seatbelt controller is well tested.

1. Choose a test method and explain why you chose this method.

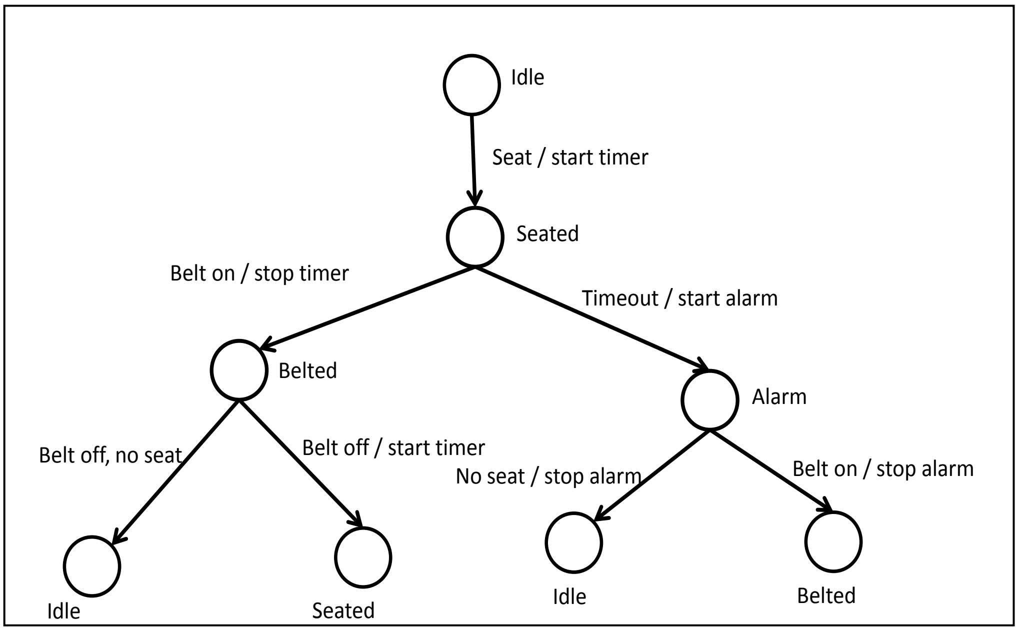
Since the system is described as a state chart, I will use the "round trip path tree" method. This method guaranties full model coverage and, assuming that the model is correct, this implies full path coverage. In addition, round trip path tree guaranties to find

- ***All state control faults***
- ***All sneak paths***
- ***Many of the existing corrupted states***

Note that the explanation is the most important part here. A lot of methods could be used but the merits of the answer lies in the explanation of why the method was chosen.

2b – Testing methods – 10 points

1. Write four complete tests for the seatbelt controller, given the method you chose in problem 2a-1



ID	***Start state***	***Event***	***Condition***	***Reaction***	***New state***
1	***Idle***	***Seat***	***-***	***Start timer***	***Seated***
2	***Seated***	***Belt on***	***-***	***Stop timer***	***Belted***
3	***Belted***	***Belt off***	***-***	***Start timer***	***Seated***
4	***Seated***	***Timeout***	***-***	***Start alarm***	***Alarm***

Problem 3 – Scenario testing (15 points)

Even if we can test all the functions and elements in the tool cell control unit described in appendix 1, the company where the control unit will be installed require that we also need a good test of the system in operation. They therefor as us to perform a scenario test

1. Explain what a scenario test is and why it can be important

Strictly speaking there are two types of scenario tests. We should focus on type 2 - scenarios used as a script for a sequence of real actions in a real or simulated environment. When it comes to realism, scenario testing type 2 is the ultimate testing method. The goal of scenario testing is to test how the system will behave

- ***In real word situations – described by scenarios***
- ***With real users, supplied by the system owner***
- ***With real customers – if necessary***

A scenario tests is done as follows:

- ***The environment is arranged according to the scenario description. Customers are instructed as needed.***
- ***A person – the game master – reads out each step of the scenario.***
- ***Users and customers react to the situations created by the game master.***
- ***The events resulting from each scenario is documented – e.g. by a video camera or by one or more observers – for later assessment.***

2. Select two scenarios and explain why they are important

Two important scenarios:

- ***The robot working in one of the areas should stop when somebody enters from the other area.***
- ***The robot is working in area A and an operator is in area B when somebody opens the gate to area A.***

3. Give a complete description of the two scenarios.

Scenario 1:

- ***The robot is in operation in are A. Both gates are closed.***
- ***An operator opens the gate to area B to pick up a product.***
- ***The robot should keep on working***
- ***The operator approached area A. When he breaks the light curtain, the robot should stop***
- ***He leaves the tool cell but forget to close the gate to area B***

- ***He pushes the restart button but nothing happens***
- ***He close the gate to area B and repeat the restart***
- ***The robot starts working again***

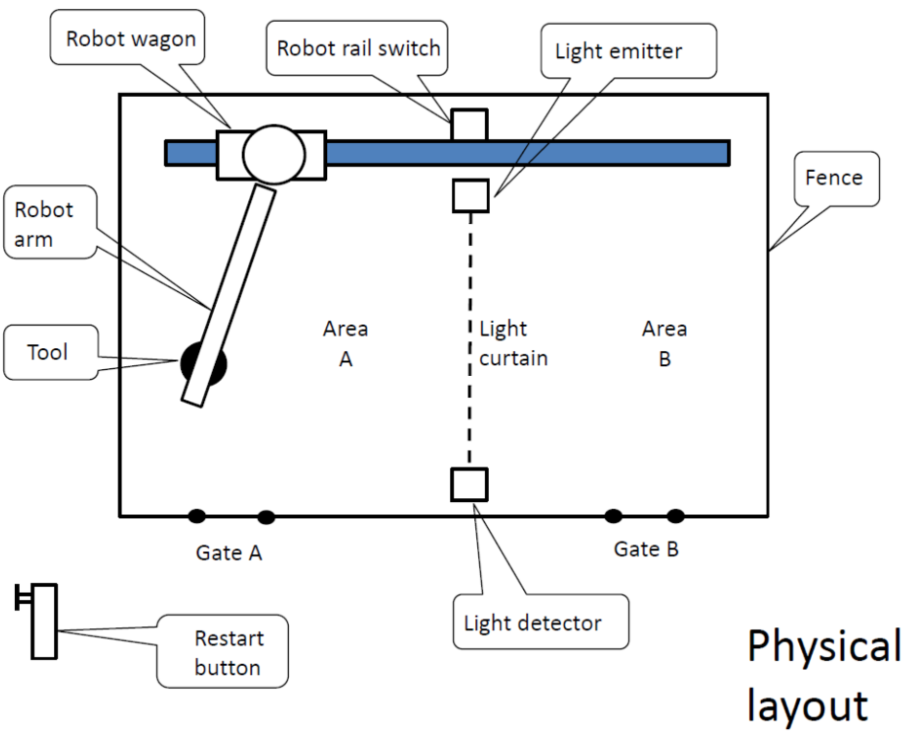
Scenario 2:

- ***The robot is in operation in are A. An operator is working in area B. The gate to area B is open.***
- ***Somebody is opening the gate to area A. The robot stops working***
- ***The operator leaves area A and closes the gate.***
- ***The robot should not start again.***
- ***The same operator pushes the restart button but nothing happens***
- ***He asks the operator in area B to get out and close the gate behind him.***
- ***When this is done he repeats the restart and the robot starts working again.***
- ***The operator who was working in area B opens gate B and the robot keeps on working in area A***

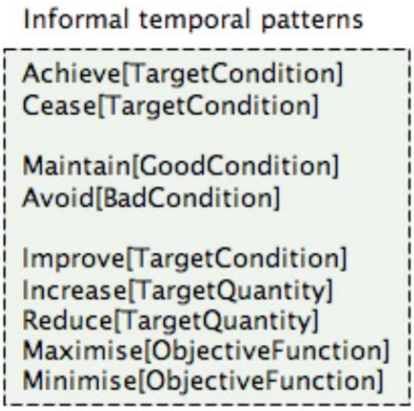
Appendix 1 – Tool Cell

Requirements – natural language

1. If the robot is in area A and gate A is opened, the robot shall stop within 20 ms.
2. If the robot is in area B and gate B is opened, the robot shall stop within 20 ms.
3. If the light curtain is broken, the robot shall stop within 20 ms.
   => Personnel enters robot area.
   Light curtain: one or more pairs of light emitters and light detectors. If the light beam from emitter to detector is broken, a signal is sent to the controller.
4. If the robot rail switch is activated and the gate is open in the area that the robot moves into, the robot shall stop within 20 ms.
   => Robot enters personnel area.
5. Stop robot => stop tool, robot wagon and robot arm
6. If the system has stopped it can only be restarted by pressing the restart button. The restart button shall only work if both gates (gate A and gate B) are closed and the light curtain is un-broken – the light detector receives light from the light emitter.
7. The operator shall check that there are personnel in neither area A nor area B.
8. The gate shall remain open if an operator has entered the corresponding area.
   => Used as an indicator that there is personnel in the area
9. Each gate has a gate sensor
10. Area A and area B are separated by a light curtain consisting of a light emitter and a light detector
11. On the rail, area A and area B are separated by a rail switch which is changed when the robot moves past it



Appendix 2 – Temporal patterns for requirements

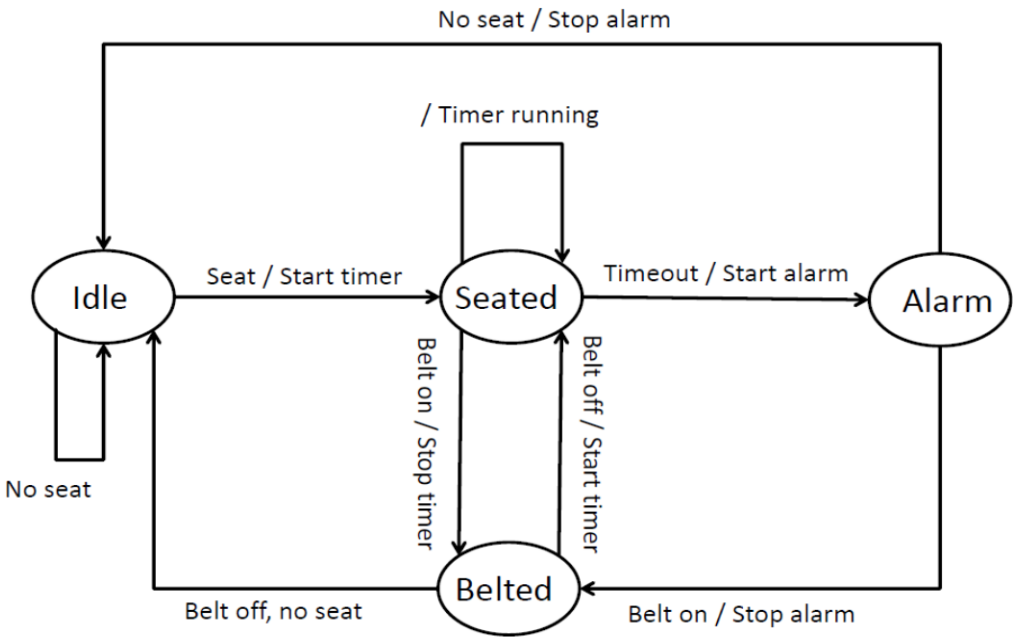

Informal temporal patterns

Achieve[TargetCondition]
Cease[TargetCondition]

Maintain[GoodCondition]
Avoid[BadCondition]

Improve[TargetCondition]
Increase[TargetQuantity]
Reduce[TargetQuantity]
Maximise[ObjectiveFunction]
Minimise[ObjectiveFunction]

Appendix 3 – Seat belt alarm – one alarm per seat



- "Seat" – someone is sitting in the seat
- "No seat" – no person is sitting in the seat