Name: **Student ID:** **Nr.**

Written Examination	
Lecture:	Systems and Software Engineering
Lecturer:	Prof. Dr.-Ing. K. D. Müller-Glaser
Affiliation:	Institute of Information Processing Technology (ITIV)
Term:	Winter Term 2011/2012
Date:	04.04.2012

Notes

- The exam consists of 19 pages (including this cover page and two additional blank pages). Please check if you got all pages.
- You are allowed to bring one page of notes (both sides). Your notes must be your own, and they must be hand-written.
- This test is to be done individually, and you are not allowed to exchange or share your notes with others during the test. As well as communication with other persons by any means is forbidden.
- The examination duration is 120 minutes.
- Please check your name and student ID on the title page.
- If you need more space, use the blank pages in the back or ask for sheets.

  Add the respective problem number on every additional page.

  Don't write on the back of pages or on your own sheets.
- Do not use pencils for writing or drawing! Statements written with pencils are not considered in the marking!
- If your solution path involves several steps, they must be apparent from your answer.

Question	1	2	3	4	5	6	7	∑
Topic	General Questions	House of Quality	EBNF	Petri Nets	Scheduling	Reliability	UML diagrams	
Total Points	17	8	5	10	15	13	14	82
Result								
Signature								

17

1 General Questions

5

1.a Soft and hard real-time systems

Describe the difference between soft and hard real-time systems. Name an example for both systems.

Hard Real-Time:

If a task is not finished within the predetermined amount of time there can be catastrophic results. For all possible events the maximum response time must be smaller than the minimum time interval between two succeeding events.

Chemical process control, airbag release, heart pacemaker,...

Soft Real-Time:

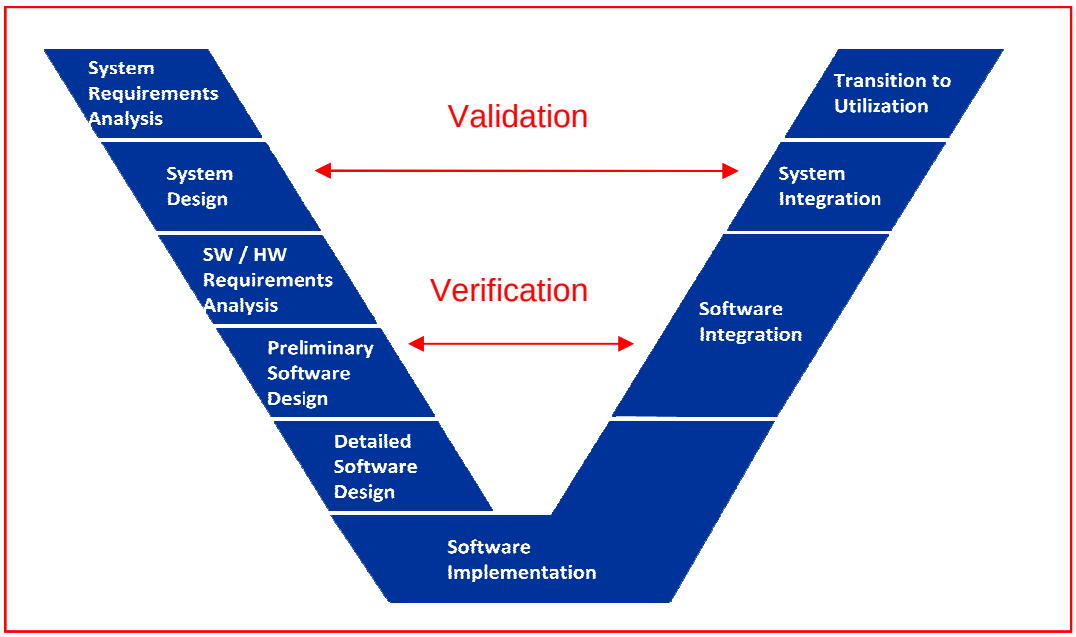
If a task is not finished within the given time span, usability is reduced. Average response time is smaller than average time interval between two succeeding events (occasional waiting times due to overload are accepted but must be eliminated on average)

Less comfort, heating control, letter sorting,...

4

1.b V-Model

Draw a graphical presentation of the different phases described in the system development sub model of the V-Model ‘97. Insert and name the 9 different phases at the respective location.



1.c Validation and Verification

5

Describe the procedures validation and verification and define in which phases of the V-Model they are adopted.

Verification

1. The process of determining, whether the product of a given phase of the system development cycle fulfils the requirements established during the previous phase. "Am I building the product right?"
2. The act of reviewing, inspecting, testing, checking, auditing, comparing or otherwise establishing and documenting whether items, processes, services or documents conform to specified requirements
3. Formal proof of correctness

V-Model: SW / HW integration

Validation

Determination of the correctness of the final system (SW, HW) produced from a development project with respect to the user's needs and requirements "Am I building the right product?"

V-Model: System Integration

1.d

3

What are the most important extensions of the statechart model in comparison to an ordinary finite state machine (FSM)?

Hierarchy

Concurrency

Communication: Transitions can be guarded (conditionally enabled). Furthermore, transitions can be associated with actions. Actions can perform computations on variables, as well as generate new events

2 House of Quality

8

Figure 1 shows a quality function deployment diagram. Name all relevant areas of the diagram and describe the data you insert in these different areas.

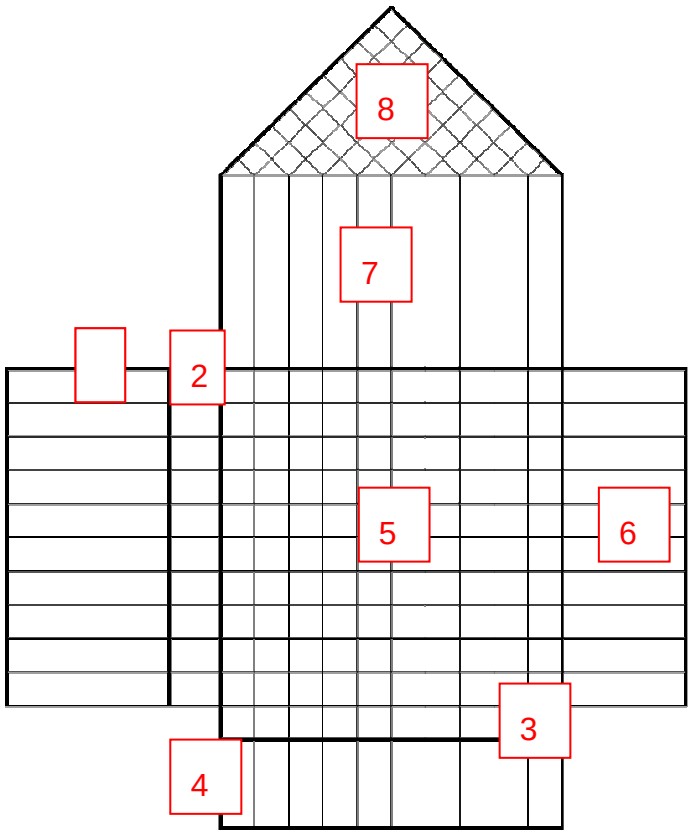


Figure 1: House of Quality

1. **Customer Requirements / What**?: documents a structured list of a product's customer requirements, described in their own words

2. **Weighting of Customer Requirements**: quantifies the relative importance of each of the customer requirements

3. **Prioritization**: relative importance of each quality feature. Calculated of the weightings and interrelation matrix sections.

4. **Technical Benchmarking**: relative technical position of the company's own existing product and competitive product. Helps identify the target level of performance to be achieved in a new product.

5. **Dependencies**: relate to combinations of individual customer and technical requirements. The level of interrelationship discerned is weighted usually in a four point scale

6. **Comparison with Competitor**: compare own existing product with competitive product in terms of the customer requirements

7. **Quality Features / How**?: measurable characteristics of the product which are related to meeting the specified customer requirements

8. **Mutual Influence of Quality Features**: used to identify where the quality features support (‘+’) or impede (‘-’) one another

3 EBNF

5

The syntax of a new programming language is described in EBNF as:

- ID := ′A′ | ′B′ | ′C′ | ‘D’ | ′1′ | ′2′ | ′3′ | ′4′
- S := ′-′ | ′+′
- A := ″=″ | ″!=″
- Start := (S?)ID(ID*)S(ID)(ID*)A(S?)(ID+)S(ID+)

3.a

Fill in Table 1 to check whether each example has the correct syntax and give a reason in case the example doesn’t have the right syntax.

Example	Legal or illegal	Reason
-A43B+5!=BC3+34	illegal	‘5’ is not part of ID
3412=3412	illegal	S is missing after the ‘=’ sign
34-12=34-12	legal	
-+34!=34+ABCD	illegal	Two ‘S’ signs at the beginning are wrong ‘S’ signs between 3 and 4 missing
+A1B2-C3D4=-A1B2+C3-D4	illegal	Only two addend on the right side of ‘=’ are allowed
1+2=3	illegal	S is missing after the ‘=’ sign
3+1=-3+3+4	illegal	Only two addend on the right side of ‘=’ are allowed
-ABCD+1234!=1+ACBD14	legal	
-1+1=-1+1-	illegal	Minus at the end isn’t right. It need to end with at least one ID character
34-D+1!=3-ABCD	illegal	Only two addend on the left side of ‘=’ are allowed

Table 1: EBNF examples

4 Petri nets

10

4.a Transition graph

4

Draw the transition graph of the Petri net given in Figure 2.

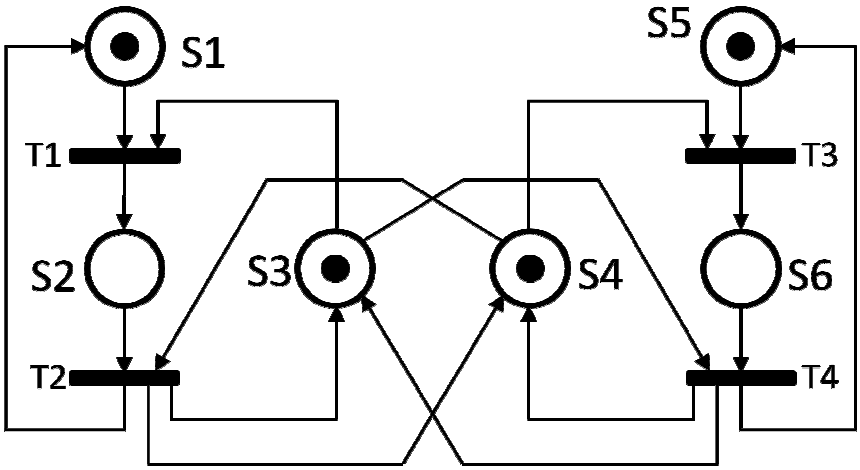
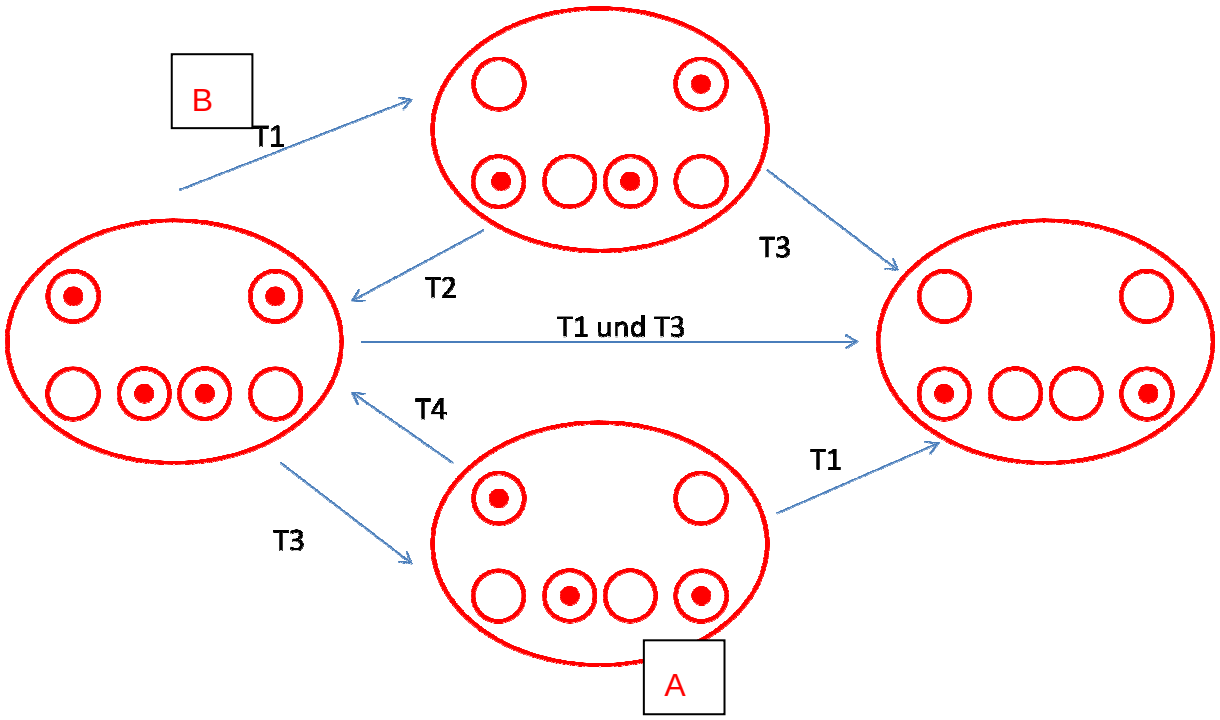


Figure 2: Petri Net



4.b Analysis of a Petri net

2

Name four characteristics to describe a Petri net.

Reachability, Liveliness or deadlock, safeness, conservation, reversibility, boundedness

4.c Analysis of a Petri net

4

Analyze the Petri net given in Figure 2, in regards of the characteristics you listed in section 4.b.

Reachability: M_0=(1,0,1,1,1,0)

Reachable markingsR(N,M_0): M_1=(0,1,0,1,1,0), M_2=(0,1,0,0,0,1), M_3=(1,0,1,0,0,1).

See transition graph in 4.a.

Liveliness or **deadlock**: not alive, since there is one deadlock

Safeness: the Petri net is 1-safe

Boundedness: the Petri net ist bounded since only one token is in a place at any time

Conservation: the total number of tokens decreases

Reversibility: the Petri net is not reversible

15

5 Scheduling

10

5.a

Four tasks with different priority should be executed on one processor.
The Table 2 shows the features of these tasks.

Task	Processing Time	Priority (0 is highest)	Arrival time	Deadline
A	52	3	T + 0 ms	T + 125 ms
B	21	1	T + 9 ms	T + 130 ms
C	39	0	T + 12 ms	T + 80 ms
D	15	2	T + 15 ms	T + 95 ms

Table 2: Scheduling tasks

You are at a certain point in time T, all tasks request processor-time at this point in time. Plot the processing of the given tasks into the following diagrams (see next page) under consideration of the scheduling methods named below.

Round Robin (time slice 10ms) (Task Queue)
TDMA (time slice 15ms) (Cycle A-B-C-D, A starts at T)
Priority Scheduling

Deadline Scheduling

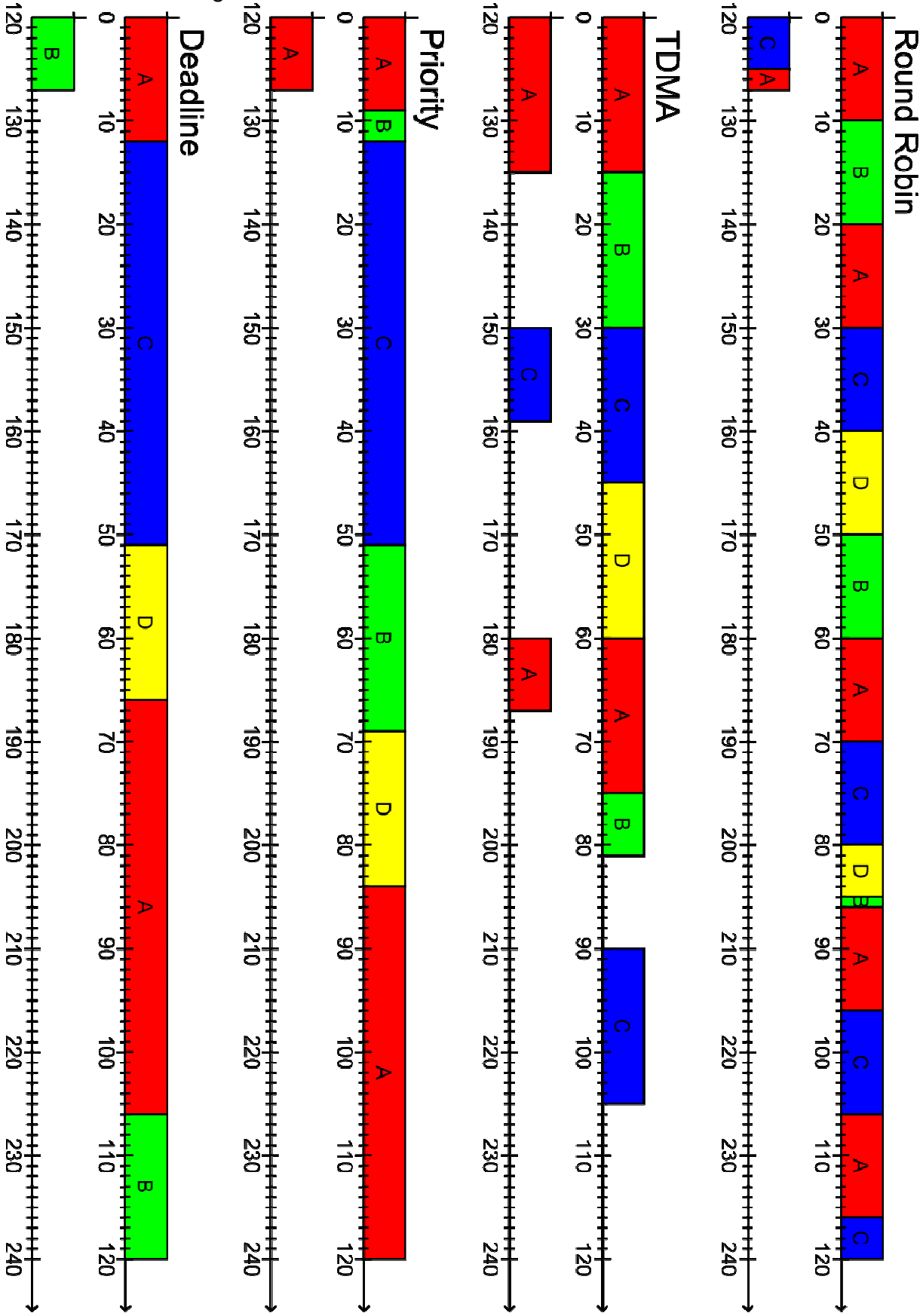


Figure 3: Scheduling

5.b

4

Calculate the maximal and the average response time of Round Robin and Deadline scheduling.

$$T_{Res,Max}(\text{RR}) = T_{Res,A} = 127\,\text{ms}$$

$$T_{Res,Avg}(\text{RR}) = \frac{T_{Res,A} + T_{Res,B} + T_{Res,C} + T_{Res,D}}{4}$$

$$= \frac{(127-0) + (86-9) + (125-12) + (85-15)}{4} = \frac{127 + 77 + 113 + 70}{4} = 96{,}75\,\text{ms}$$

$$T_{Res,Max}(\text{Deadline}) = T_{Res,B} = 118\,\text{ms}$$

$$T_{Res,Avg}(\text{Deadline}) = \frac{T_{Res,A} + T_{Res,B} + T_{Res,C} + T_{Res,D}}{4}$$

$$= \frac{(106-0) + (127-9) + (51-12) + (66-15)}{4} = \frac{106 + 118 + 39 + 51}{4} = 78{,}5\,\text{ms}$$

5.c

1

Which different priorities must be assigned to the tasks so that Deadline Scheduling finds a valid schedule?

Deadline scheduling is independent from priorities.

13

6 Reliability

3

6.a

Calculate the reliability of the system given by the block diagram in Figure 4 as function of the reliabilities of the involved blocks. Assume stochastic independence of the reliability for the different blocks in the block diagram.

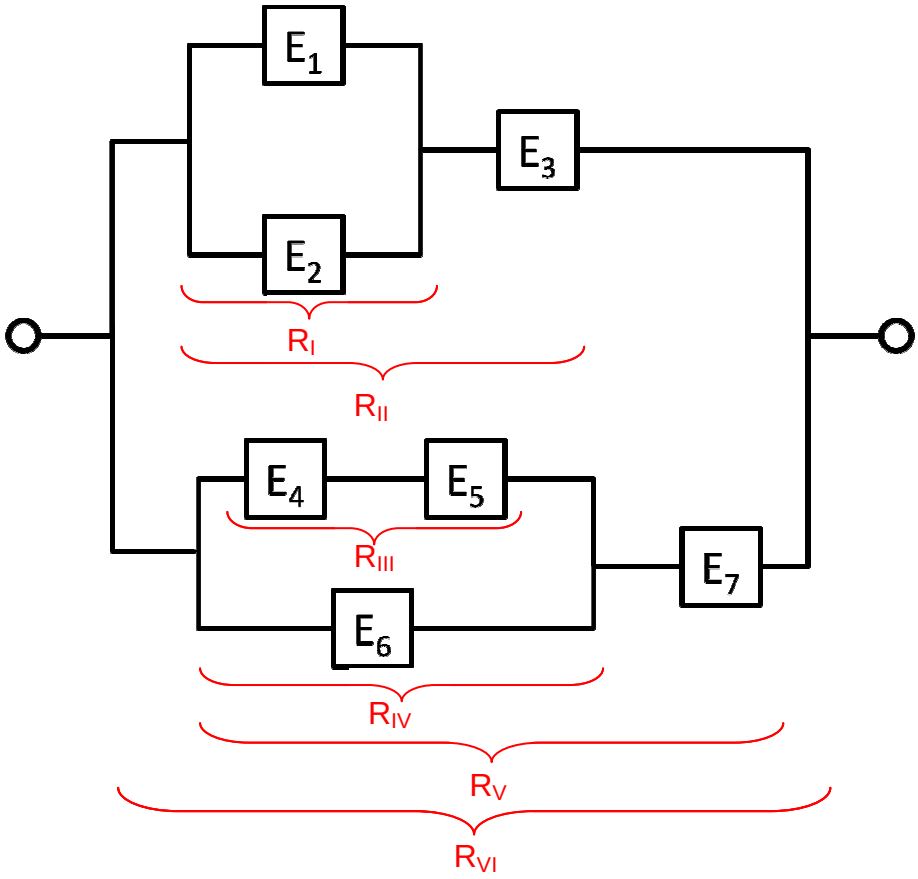


Figure 4: Reliability block diagram of a system

$R_I(t) = R_1(t) + R_2(t) - R_1(t)* R_2(t)$

$R_{II}(t) = R_I(t)* R_3(t)$

$R_{III}(t) = R_4(t)* R_5(t)$

$R_{IV}(t) = R_{III}(t) + R_6(t) - R_{III}(t)* R_6(t)$

$R_V(t) = R_{IV}(t)* R_7(t)$

$R_{VI}(t) = R_{II}(t) + R_V(t) – R_{II}(t)* R_V(t)$

6.b

1

The constant failure rate of a module is $\lambda = 10^{-3}$ / h.

Calculate the mean time to failure (MTTF) for this module

MTTF = $\int_0^\infty R(t)dt = \frac{1}{\lambda} = 10^{3}\ h$

6.c

3

Figure 5 shows the failure rate of a system over time.

Name the three phases in the diagram.

What is the name of a curve with such a shape?

0-t_1: Early failure period or infant mortality phase

t_1 - t_2: Intrinsic Failure Period or phase of constant failure rate

t_2 - ∞: Wearout Failure Period or wear out phase

Name: Bathtube curve

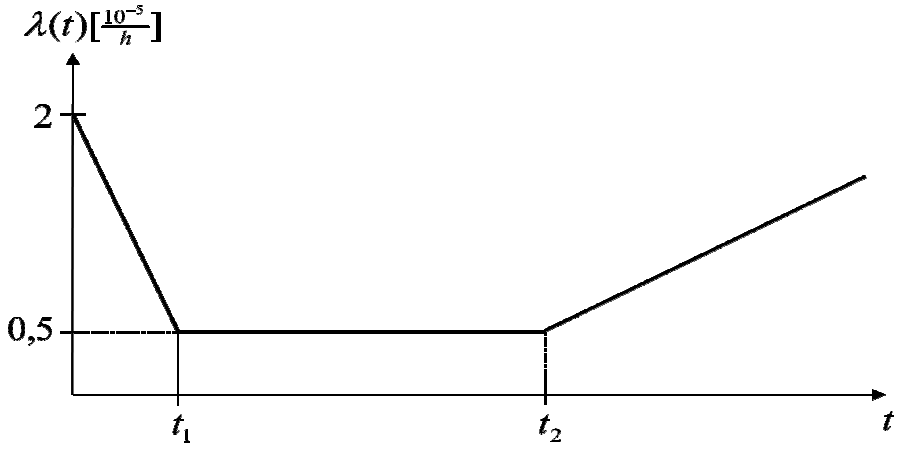


Figure 5: Failure rate over time

6

6.d

Give the function for the failure rate according to the curve shown in Figure 5 and calculate the reliability of this system.

The gradients of the function are given in the following:

$0 \le t \le t_1$: – see Figure 5

$t_1 < t \le t_2$: –

$t > t_2$: –

$$\lambda(t) = \frac{10^{-5}}{h} \cdot \begin{cases} -\frac{3}{2t_1}t + 2 & 0 \le t \le t_1 \\ 0{,}5 & t_1 < t \le t_2 \\ \frac{1}{4}t + \frac{1}{2} - \frac{1}{4}t_2 & t > t_2 \end{cases}$$

$$R(t) = e^{-\int_0^t \lambda(t)dt} = e^{-I(t)}$$

$$I(t) = 10^{-5} \cdot \begin{cases} \int_0^t -\frac{3}{2t_1}t + 2 = \left[-\frac{3}{4t_1}t^2 + 2t \right]_0^t & 0 \le t \le t_1 \\ \int_0^t \lambda(t) = \int_0^{t_1} \lambda(t)dt + \int_{t_1}^t \lambda(t)dt = \left[-\frac{3}{4t_1}t^2 + 2t \right]_0^{t_1} + \left[\frac{1}{2}t \right]_{t_1}^t & t_1 < t \le t_2 \\ \int_0^t \lambda(t) = \int_0^{t_2} \lambda(t)dt + \int_{t_2}^t \lambda(t)dt = \left[\frac{3}{4}t_1 + \frac{1}{2}t \right]_0^{t_2} + \left[\frac{1}{8}t^2 + (\frac{1}{2} - \frac{1}{4}t_2)t \right]_{t_2}^t & t > t_2 \end{cases}$$

$$I(t) = 10^{-5} \cdot \begin{cases} -\frac{3}{4t_1}t^2 + 2t & 0 \le t \le t_1 \\ \frac{3}{4}t_1 + \frac{1}{2}t & t_1 < t \le t_2 \\ \frac{1}{8}t^2 + \frac{1}{8}t_2{}^2 + \left(\frac{1}{2} - \frac{1}{4}t_2\right)t + \frac{3}{4}t_1 & t > t_2 \end{cases}$$

14

7 UML Diagrams

Description of a bank software system:

A customer can open an account at the bank. A customer can open an arbitrary number of additional accounts. For every new customer the name, the address and the date, when she or he has opened the first account, are recorded.

Distinguish between current accounts and savings accounts.

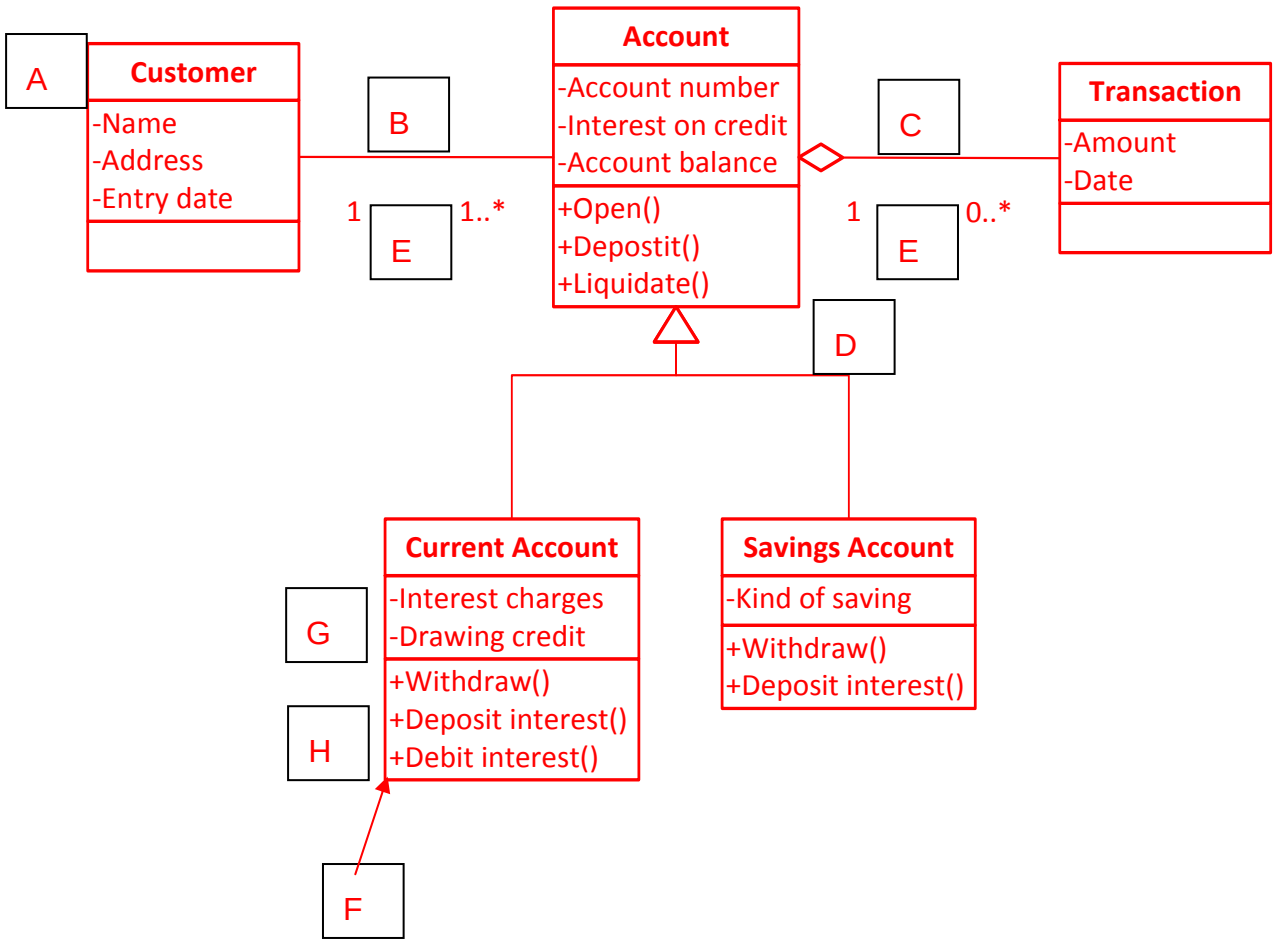
For every account an individual rate of interest on credit is fixed and for current accounts an individual rate of interest charges is fixed too. In addition, every account has an unambiguous account number. About every savings account the kind of saving is stored.

A customer can deposit and withdraw amounts. Furthermore the interest on credit is credited and in case of current accounts interest charges are debited. In order to calculate the amount of interest the date and the amount of every transaction has to be noted down. A customer may liquidate each of his accounts if she or he wants to.

7.a Class diagram

11

Identify the classes, attributes, operations, associations and inheritance structures in the description of the bank software system above and design a class diagram. Also identify the visibility of the attributes and operations.



3

7.b Object diagram

Model two possible object diagrams, based on the class diagram you designed in exercise 7.a.