SOEN 6481 Software Systems Requirements Specification

Winter 2018

Example Midterm Questions

Name: _____ Total Points:

ID: _____ _____ / ____

Instructions. *This example SOEN 6481 exam contains questions from previous years that you can use to test your preparation. Note that the exam is a* ***closed book*** *exam. Also, note that the actual exam will not necessarily cover the same questions as the ones here (or even the same type of questions)!*

Make sure you familiarize yourself with the exam rules:

- This is a **closed book**, 90 min. exam
- You are reminded of the **Academic Code of Conduct**
- Provide **all answers** in this booklet
- Check if your booklet has all **16 pages**
- Keep your **Student ID** card on your desk
- **Do not detach** any pages from this exam!
- You may write with **pen or pencil**
- The **only** allowed tool is an ENCS-approved calculator
- **No electronic devices** (except an approved calculator) are permitted on your desk or person (e.g., you cannot keep your cell phone in a pocket). These **must** be turned off **and** placed together with your other belongings at the front/side of the exam room.
- You will get marks for brief and precise answers. You will not get marks for long essays or for information that is correct but does not answer the question.
- You cannot leave the room in the first 20min of the exam. You also cannot leave during the last 15min of the exam.
- At the end of the exam, **remain seated** until **all** exams (not just yours!) have been collected.

(2pts) **1.** Name five *defects* that can appear in a requirements document and provide a one-sentence definition for each.

2 pts

1. Name: Omission
   Definition: Contradiction
2. Name: Inadequacy
   Definition: Ambiguity
3. Name: Unmeasurability
   Definition: Noise, overspecification
4. Name: Unfeasibility (wishful thinking)
   Definition: Unintelligibility
5. Name: Poor structuring, forward reference, remorse
   Definition: Opacity

(1pt) **2.** When doing an interview with a stakeholder (e.g., user), the following is **NOT** recommended: *(Check only one answer)*

1 pt

- ☐ Preparing for the interview (e.g., review of domain documents)
- ☐ Using an interview template
- ☐ Asking the users directly about their needs
- ☐ Starting by presenting a solution to the stakeholder's problems
- ☐ Asking context-free questions

(1pt) **3.** To be *complete*, a decision table with N input conditions must have: *(Check only one answer)*

1 pt

- ☐ N columns
- ☐ $2 \cdot N$ columns
- ☐ N^2 columns
- ☐ $\frac{N}{2}$ columns
- ☐ 2^N columns

(3pts) **4.** Consider the following interaction matrix:

	3 pts

Statement	S1	S2	S3	S4	Total
S1	0	1000	1	1	
S2	1000	0	0	1	
S3	1	0	0	1	
S4	1	1	1	0	
Total					

Here, S_{ij} =

- 1: conflict
- 0: no overlap
- 1000: no conflict

(a) (1 pt) Compute the values for the total row and column and insert them in the table above.

(b) (1 pt) Use the formula discussed in the lecture to compute the total number of *conflicts*: _____

(c) (1 pt) Use the formula discussed in the lecture to compute the total number of *non-conflicting overlaps*: _____

(7pts) **5.** Consider the following Defect Detection Prevention (DDP) risk-consequence table for a library loan management system:

	7 pts

Objectives	**Risks**				**Loss of objective**
	Late returns (likelihood: 0.6)	Stolen copies (likelihood: 0.3)	Lost copies (likelihood: 0.1)	Long loan by staff (likelihood: 0.5)	
Regular availability of book copies (weight: 0.4)	0.40	0.60	0.60	0.20	
Comprehensive coverage of library (weight: 0.3)	0	0.20	0.20	0	
Staff load reduced (weight: 0.2)	0.30	0.50	0.40	0.10	
Operational costs decreased (weight: 0.1)	0.10	0.30	0.30	0.10	
Risk criticality					

With

$$Criticality(r) = Likelihood(r) \times \sum_{obj} (Impact(r, obj) \times Weight(obj))$$

and

$$Loss(obj) = Weight(obj) \times \sum_{r} (Impact(r, obj) \times Likelihood(r))$$

(a) (1 pt) What is the meaning of a single table entry, i.e., of each pair (obj, r)?

- [x] (estimated) loss of satisfaction of objective *obj* if risk r occurs
- [ ] relative cost to recover objective *obj* if risk r occurs
- [ ] this is the risk-reduction leverage (RRL)
- [ ] the (estimated) reduction of risk r under objective *obj*
- [ ] None of these options

(b) (2 pts) Compute the values for *Loss of objective* and enter them in the last column of the table.

(c) (2 pts) Compute the values for *Risk criticality* and enter them in the last row of the table.

(d) (1 pt) Which *objective* is most at risk?

- [ ] Regular availability of book copies
- [ ] Comprehensive coverage of library
- [ ] Staff load reduced
- [ ] Operational costs decreased
- [ ] None of these options

(e) (1 pt) What is the *highest risk* overall? _____

(8pts) **6.** You are the requirements engineer in an information system project for a video rental store. Simplifying assumptions and details:

8 pts

- It is a stand-alone store, not part of a larger organization.
- Rents only videos, not computer games or other items.
- A "video" can be in any medium: tape, DVD, and so on.
- The rental charge may vary by medium. For example, DVD rentals are more expensive than tapes.
- The store does not sell anything. For example, there are no sales of videos or food.
- All transactions are rentals.
- The input medium by which membership and video rentals are captured is not important.
- Cash-only payments.
- On completion of a rental, the customer receives a transaction report with 'typical' information on it (use your judgement).
- Each renter has a separate membership.

(a) (1 pt) Identify 4 *actors* and give a brief description (3–5 words) for each

- **Actor 1:** Name: _____

  Description: _____
- **Actor 2:** Name: _____

  Description: _____
- **Actor 3:** Name: _____

  Description: _____
- **Actor 4:** Name: _____

  Description: _____

⇒ ***Continued on next page!***

(b) (2 pts) Identify four primary (user-goal level) use cases and related actors (identified by A1, A2 etc.):

- **UC1** Name: _____ Actors: _____
- **UC2** Name: _____ Actors: _____
- **UC3** Name: _____ Actors: _____
- **UC4** Name: _____ Actors: _____

(c) (1 pt) Draw the UML use case context diagram for your actors and use cases:

⇒ ***Continued on next page!***

(d) (3 pts) Write the use case (steps only, no additional details like pre- and postconditions) for "Rent Video" in *essential* style, for the main success scenario (basic flow). (Note: you can have less than 10 steps.)

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____

(e) (1 pt) Write two extensions (alternative scenarios) for the "Rent Video" use case. Provide the step number in the main success scenario and a brief description.

- _____ : _____
- _____ : _____

(7pts) **7.** You elicited the following requirements for a library loan system:

7 pts

1. A book can be on stack if and only if it is not on reserve or on loan
2. A book can be on reserve if and only if it is not on stack or on loan
3. A book can be on loan if and only if it is not on reserve or on stack
4. A book can be requested if and only if it is on stack or on reserve

(a) (2 pts) Translate these requirements into propositional logic:

1. _____
2. _____
3. _____
4. _____

(b) (1 pt) Consider the two requirements 1. and 3. together. Are they *consistent*? Prove or disprove (*Hint: you do not need to create a complete truth table*):

⇒ ***Continued on next page!***

(c) (4 pts) Using a *refutation proof by resolution*, show that the statement

If a book is on loan then it can not be requested

logically follows from the requirements:

(7pts) **8.** Consider the following domain description for email clients:

7 pts

> The client has one mailbox which consists of a number of different folders. Each folder contains a number of messages. A message cannot exist in more than one folder and cannot exist outside a folder. A user can invoke a view on a message and in fact a user may have multiple views, each corresponding to a single message.

(a) (1 pt) Name an appropriate method for identifying *conceptual classes* in this domain description: _____

(b) (1 pt) Use the method to create a list of domain concepts based on the provided description:

- _____
- _____
- _____
- _____
- _____
- _____

(c) (4 pts) Create a domain model for the email client as a UML class diagram. Make sure you show all appropriate details, including associations, multiplicities, and aggregations.

(d) (1 pt) Explain the difference between *aggregation* and *composition*. Give a brief example for each, based on the email domain:

Aggregation: _____

Composition: _____

(6pts) **9.** Consider the following Z schema specification for a birthday book application:

6 pts

$[NAME,\ DATE]$

BirthdayBook

$known : \mathbf{P} NAME$

$birthday : NAME \pfun DATE$

$known = \mathrm{dom}\ birthday$

(a) (3 pts) Write a non-robust Z schema for the *UpdateBirthday* operation, which changes the date of an *existing* entry (i.e., if a name is not in the system, it will not be added by the UpdateBirthday operation).

UpdateBirthday

⇒ ***Continued on next page!***

(b) (1 pt) Now make the operation robust by using the following two schemas for error handling:

$$REPORT ::= ok \mid already\_known \mid not\_known$$

Success

$$result! : REPORT$$
$$result! = ok$$

NotKnown

$$\Xi BirthdayBook$$
$$name? : NAME$$
$$result! : REPORT$$
$$name? \notin known$$
$$result! = not\_known$$

Define a robust version of *UpdateBirthday* that returns *ok* in case of success and *not_known* in case of an error:

$RUpdateBirthday =$ _____

(c) (2 pts) Now show the combined schemas for the *RUpdateBirthday* operation:

UpdateBirthday

(4pts) **10.** Consider the following specification for an automated teller machine (ATM):

4 pts

> A customer arrives at an ATM. He insert his card and then enters the password (PIN). If the password is not accepted he has to re-enter it. If the password is accepted the customer selects the type of transaction he wishes to conduct and then the ATM performs the transaction. If the ATM detects that it has not enough money to fulfil the request it prints an error message, returns the card, and ends the transaction.
>
> Once the transaction is complete the customer has the option to perform more transactions. If no further transactions are selected, the ATM concurrently returns the card and prints a receipt of all transactions and the interaction with the ATM terminates.

Draw a UML *activity diagram* for this description. Show explicitly the allocation of the activities among the customer and the ATM (use swimlanes).

(6pts) **11.** Consider the following specification for a course planning system:

6 pts

> The course section life cycle starts from its planning. Once the decision for opening the registration for the course is received, the course is opened. While the course is opened, the requests for registering can be accepted. The course will not be actually taught until the class size reaches a certain minimum. The requests to register are accepted until the course reaches the predefined maximum number of students, or the registration deadline has passed; in both cases, the course section becomes closed. If the class size is below the minimum, the class is cancelled. Closing the section when there are not enough students will have the same effect as cancelling it.

(a) (4 pts) Draw an UML state machine diagram to specify Course Section behaviour. Use hierarchical states where appropriate:

(b) (2 pts) *True* or *False*? ***Note:*** Each correct answer gives you $\frac{1}{2}$ points, each wrong answer $-\frac{1}{2}$, "don't know" 0 (however, you will never get a negative score for this question).

State machines can be used to describe legal system events within a use case.

☑ True ☐ False ☐ Don't know

Activity diagrams are useful to model use cases with many alternative flows or extensions.

☑ True ☐ False ☐ Don't know

State machines are an alternative modeling technique to domain models in RE.

☐ True ☑ False ☐ Don't know

State diagrams can show how a *single* object behaves across *different* use cases.

☑ True ☐ False ☐ Don't know

(6pts) **12.** You are developing the *SocialArt* web application, based on RDF/RDFS web technologies.

6 pts

(a) (3 pts) Consider the following RDF triples (in Turtle format):

```
01  BASE  <http://example.org/>
02  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
03  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
04  PREFIX schema: <http://schema.org/>
05  PREFIX dcterms: <http://purl.org/dc/terms/>
06  PREFIX wd: <http://www.wikidata.org/entity/>
07
08  <bob#me>
09      a  foaf:Person ;
10      foaf:knows <alice#me> ;
11      schema:birthDate "1990-07-04"^^xsd:date ;
12      foaf:  topic_interest   wd:Q12418 .
13
14  wd:Q12418
15      dcterms: title   "Mona_Lisa" ;
16      dcterms: creator  <http://dbpedia.org/resource/Leonardo_da_Vinci> .
17
18  <http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
19      dcterms: subject  wd:Q12418 .
```

Draw a labeled, directed graph corresponding to these triples:

(b) (2 pts) Now extend your domain model (RDF Schema) by adding a *has_age* property to record the age of a person. The *domain* of *has_age* is `foaf:Person` and the *range* is `xsd:integer`. Show the additional triples using either RDF/XML or Turtle:

(c) (1 pt) Using your domain model extension from (b), provide a triple stating that Bob (`<bob#me>` in (a)) is 25 years old:

— End of Exam —

Note: This sheet will also be provided in the actual exam

Truth tables for $\neg$, $\wedge$, and $\vee$

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Truth tables for $\leftrightarrow$ and $\rightarrow$

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Equivalence Rules

Equivalence Rule	**Name**
$p \Leftrightarrow \neg\neg p$	double negation
$p \rightarrow q \Leftrightarrow \neg p \vee q$	implication
$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$	De Morgan's laws
$p \vee q \Leftrightarrow q \vee p$ $p \wedge q \Leftrightarrow q \wedge p$	commutativity
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	distributivity
$p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$ $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$	associativity

Inference Rules

Inference Rule	**Name**
$\left.\begin{array}{r} p \\ q \end{array}\right\} \Rightarrow p \wedge q$	conjunction
$\left.\begin{array}{r} p \\ p \rightarrow q \end{array}\right\} \Rightarrow q$	*modus ponens*
$\left.\begin{array}{r} \neg q \\ p \rightarrow q \end{array}\right\} \Rightarrow \neg p$	*modus tollens*
$\left.\begin{array}{r} p \rightarrow q \\ q \rightarrow r \end{array}\right\} \Rightarrow p \rightarrow r$	chaining
$\left.\begin{array}{r} p \vee q \\ \neg p \vee r \end{array}\right\} \Rightarrow q \vee r$	resolution
$p \wedge q \Rightarrow p$	simplification
$p \Rightarrow p \vee q$	addition