CS445 / CS645 / ECE451
Fall 2015 — Final Exam

10 December 2015, 4:00pm–6:30pm

Instructor: Daniel M. Berry

Time allowed: 2.5 hours = 150 minutes

No aids allowed (*i.e.,* closed book).

Answer all of the questions on this exam paper.

There are 11 questions for a total of 150 marks.

Plan your time wisely: 1 minute per mark

____

Your Name and Student Number

In the immortal words of the yet to be born Spock of Vulcan,

Live long and prosper!

Q1		/ 10
Q2		/ 15
Q3		/ 25
Q4		/ 8
Q5		/ 6
Q6		/ 15
Q7		/ 20
Q8		/ 15
Q9		/ 8
Q10		/ 18
Q11		/ 10
TOTAL		/150

In this exam, a short underscore of 1 inch (= 2.54 cm) should be filled with one word. A long underscore of 3 inches (= 7.63 cm) should be filled with a phrase consisting of one to several words. In the either case, if you cannot think of exactly the right number of words, then give the best answer that you can and we'll give it as many marks as we can, possibly even full credit! If you cannot even think of just words to fill in, then write an answer as a sentence, and we'll give it as many marks as we can.

In this exam, if you are asked for a simple answer, you need not justify it, unless you are also asked explicity "Why?".

In the following,
"CBS" means "computer-based system".
"NFR" means "non-functional requirement", a.k.a. "quality attribute" or "... requirement".
"RE" means "requirements engineering".
"SRS" means "software requirements specification, written according to some standard, e.g., IEEE".
"UM" means "user's manual".

1. **[10 total marks]** RE Reference Model

   Among the obstacles an airplane flying in the air can face are birds also flying in the air. At normal flying speeds, if an airplane, built to today's standards with today's materials, hits a bird while flying, the airplane will break apart and crash. With today's materials, an airplane, that is built strong enough to not break up if it hits a bird while flying, will be too heavy to fly.

   Consider the validation condition, $D, S \vdash R$, for an airplane, where

   - R is airplane flies $\wedge$ airplane does not crash
   - S is airplane is made of today's materials $\wedge$ airplane gets enough speed to lift off and stay flying
   - D is there are birds in the air

   (a) Is it possible to prove $D, S \vdash R$?

   (b) Give one change to D that will allow $D, S \vdash R$ to be proved with no change to the original S and R.

   (c) Give two changes to S each of which will allow $D, S \vdash R$ to be proved with no change to the original D and R.

   1.

2.

(d) Give two changes to R each of which will allow $D, S \vdash R$ to be proved with no change to the original D and S.

1.

2.

(e) It is generally accepted that the probability of an airplane's hitting a bird during flight, especially at the usual cruising altitudes, is low enough that we *tolerate* that a flying airplane may hit a bird, and do nothing to avoid the resulting crash. In the subquestions (b) through (d) above, check off (with a $\surd$ mark) each change that corresponds to this toleration.

2. **[15 total marks]** RE Reference Model and Domain Models

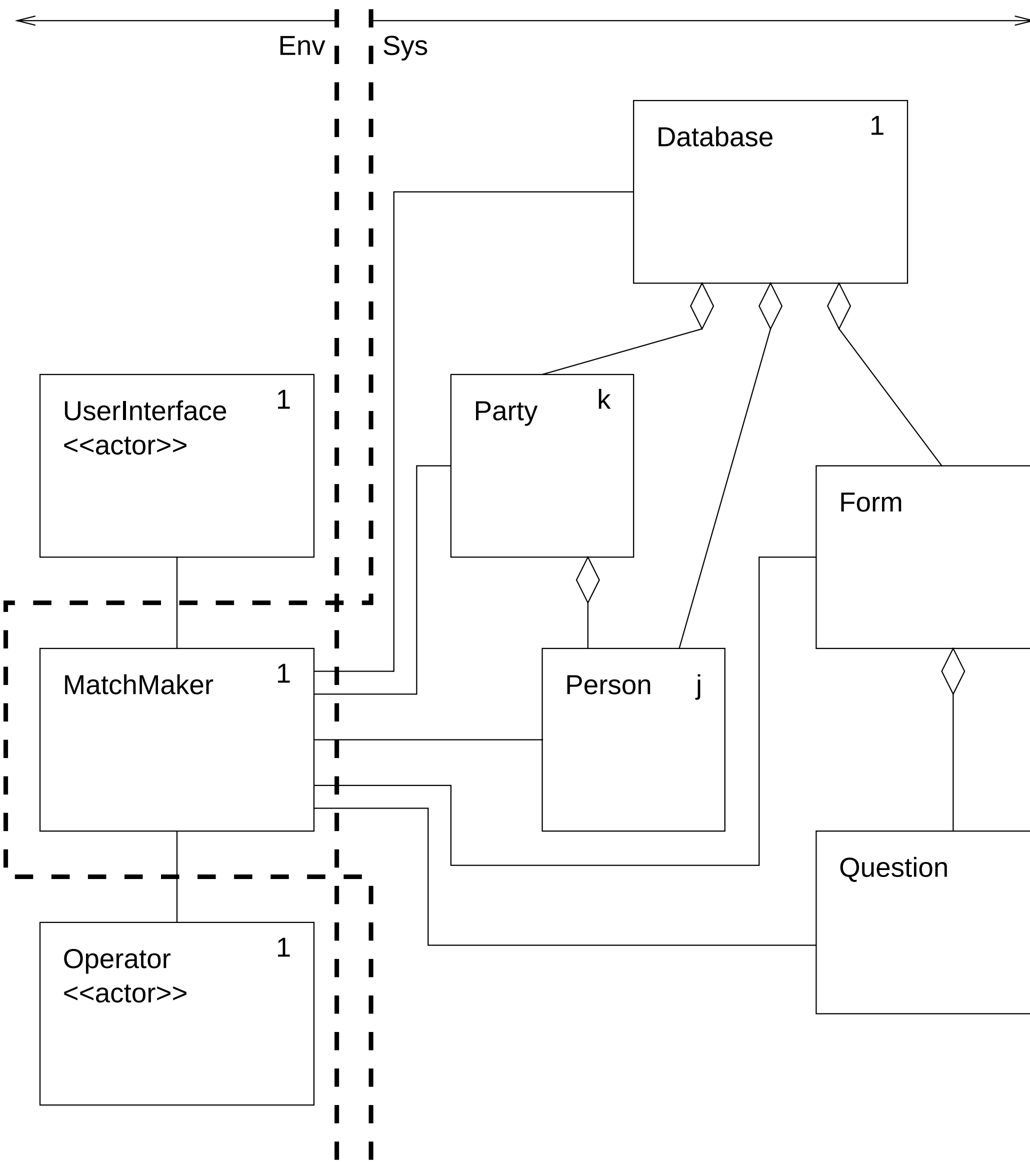
   (a) For an airplane that allows both human piloting and auto piloting, draw on the rest of this page a domain model that shows the entities: airplaneSurface, cockpitControls, humanPilot, obstaclesInTheAir, radar, restOfAirplaneIncludingComputerAndEngines, and wind, and the operations: beDetectedBy, operateControls, and pushOn. This domain model should show all classes and their multiplicities all operations in the proper classes, and all necessary links between classes.

   THERE IS A SECOND PART TO THIS QUESTION.

   (b) Superimpose on your domain model a division of the world into an *Env*; a *Sys*; and their intersection, the *Intf*. You may use any notation to indicate the division, including (1) labeling the classes with "*E*" and "*S*" and (2) drawing labeled regions

3. **[25 total marks]** Domain Models of Matchmaker

Please see below the main domain model of the Matchmaker that was produced as part of Berry's solution to Deliverable 2:

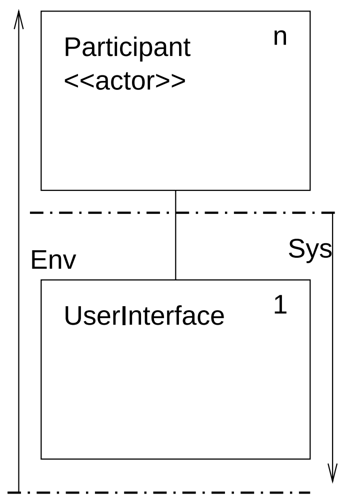


In this model, the Matchmaker was shown as being used by the actors UserInterface and Operator. While the Operator actor is fully human and can make mistakes on input, the UserInterface actor is software and it only *represents* human Participants. The UserInterface presents only (1) correctly and completely filled forms and (2) correct, full, and validated payments.

(a) This model means that the designer and implementor of Matchmaker do not have to

____ filling in forms, making payments, and the fact that a human Participant will make mistakes during the filling of a form and the making of a payment.

Another domain model:



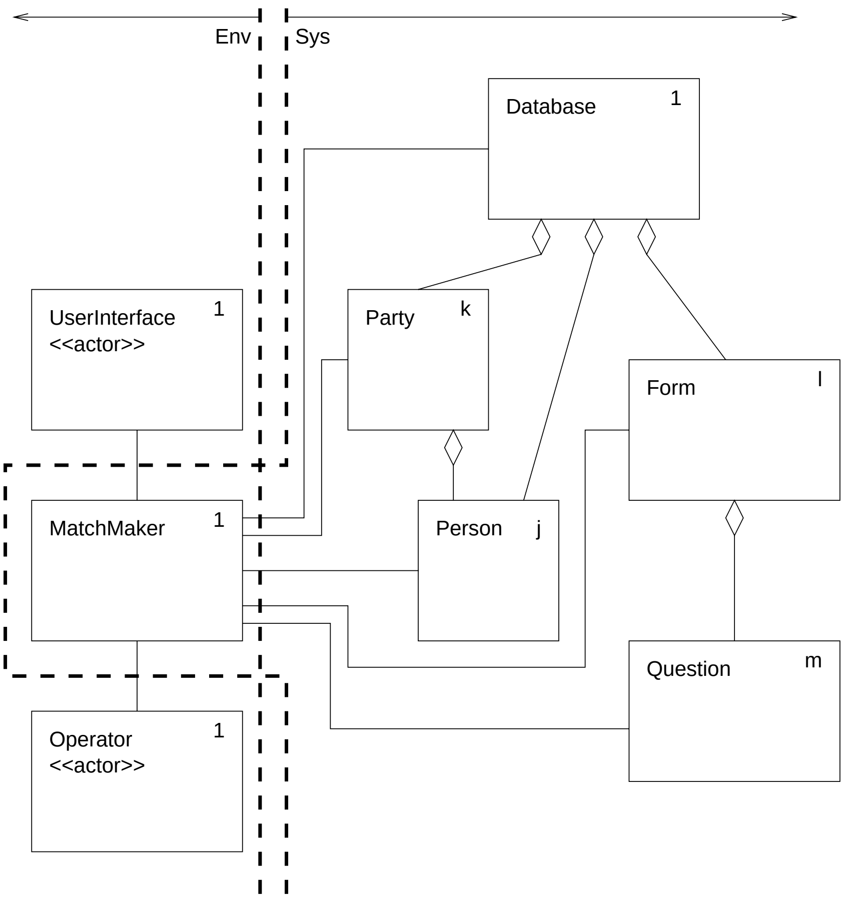
describes the interaction between the human Participants and the UserInterface. The UserInterface offers to the Participants use cases (operations) that allow incrementally filling in a form at a Website, question by question, and making payments, all with the possibility of making mistakes. To the designer of Matchmaker, this UserInterface could be regarded as off the shelf.

However, let's pretend that the company that was to produce this UserInterface has decided for some reason not to do it. Now we have to redesign the Matchmaker so that it now interacts directly with the human Participants at a Website. This new Matchmaker thus has to deal with mistakes that the Participants can make on input.

(b) On the next page, modify the existing Matchmaker domain model to get a domain model for the new Matchmaker. The existing domain model is reproduced slightly smaller to allow you room for adding any needed classes and links. You may write over any text that you need to change.

(c) How is the interface between the new Matchmaker and Participants different from the interface between the old Matchmaker and the UserInterface?

(d) What use cases does the new Matchmaker have that the old Matchmaker did not have?


Env
Sys
Database 1
UserInterface <<actor>> 1
Party k
Form l
MatchMaker 1
Person j
Question m
Operator <<actor>> 1

This page is left blank to provide working space.

4. **[8 total marks]** World Model for Matchmaker

Recall that two components of the world for any CBS are *Sys* and *Env*.

Suppose that each Participant, P, who uses the new Matchmaker, introduced in the previous question, is to be allowed to see the contents of
1. P's own form and
2. the forms of only those others with whom P has been matched, in dating mode or in party mode. These others are called P's "actual matchees".

In the underscore before each *Sys* data item below, classify the data item as being in one of the two subregions of *Sys* in a world model:

1. S' = *Sys* – *Env*
2. I = *Sys* ∩ *Env* = *Intf*.

Note that an item is in I only if *all* of it is eligible to be in I.

_______________ loop indices

_______________ P's form data

_______________ the form data for all Participants

_______________ the form data for all of P's actual matchees

_______________ the form data for all of P's potential (but not actual) matchees

_______________ the list of a party's Participants

_______________ the list of all Participants

_______________ the list of all current parties

______________ the list of all of P's potential (but not actual) matchees

______________ the list of all parties

______________ the list of engaged couples (a data item needed by the Gale–Shapley algorithm)

______________ the list of matched couples for a party

______________ the ranking of all participants according to P's and their form data

5. **[6 total marks]** Graduate Student Lectures

(a) Each algorithm presented for either the Stable Marriage Problem (SMP) or the Stable Roommates Problem (SRP) assumes that at the start, you have for each individual, I, to be matched,

a list of ______________________________, ranked according to I's ________.

In a bipartite party, there are two groups of equal size, and each individual from one group is matched with someone from the other group.

In a unipartite party, there is one group whose size is an even number, and each individual is matched with someone from the (same) group.

(b) A stable matching among one or two groups of individuals is one in which there *does not exist any* pair of individuals, A, and B, (Please complete the definition here:)

(c) If you use the Gale-Shapley SMP algorithm to find matches for a bipartite party, are you guaranteed to find a stable matching that matches everyone?

(d) If you use the Irving SRP algorithm to find matches for a unipartite party, are you guaranteed to find a stable matching that matches everyone?

(e) In the end, is it essential for the users of Matchmaker's party mode that the matches be stable?

Why or why not?

6. **[15 total marks]** User Interfaces

Consider the following two paradigms for the operation of a text editor E on a file F stored on a disk drive D:

1. Save on D a back up copy, F.bak, of F;
For each edit operation, E updates the original F on D;
When the user exits E, F on D is already up to date.

2. Copy F into F' in the main memory M;
For each edit operation, E updates F' in M;
When the user exits E, he or she must copy F' from M to F on D to get F to be up to date.

(a) Which paradigm is closer to the old paper-and-pencil-with-eraser paradigm?

(b) In that paradigm, what entity corresponds to the paper?

(c) In this same paradigm, what entity corresponds to the pencil with eraser?

(d) In the old paper-and-pencil-with-eraser paradigm, what does saving on D a back up copy, F.bak, of F correspond to?

(e) The message from E, "Do you want to save changes before exiting editor?" implies which paradigm: 1, 2, or both?

(f) Write the message that a Paradigm 1 editor, E, should emit before obeying a user's command to exit E.

7. **[20 total marks]** SRSs, UMs, and Ambiguity

(a) Consider the sentence,

The word "word" has a "w" and three other letters.

This sentence has "word" used both for its meaning and as itself. Circle the occurrence of "word" used for its meaning. Underline the occurrence of "word" used as itself.

Write and punctuate correctly a sentence that uses "W" both for its meaning and as itself. (This sentence does not have to be true in the sense of matching the real world.)

(b) Consider the sentence

The (security) (sub)system(s) (is)are implemented.

which is typical of what one sees in requirements documents.
How many grammatically correct sentences does this sentence represent?

(c) In the standard IEEE SRS about a CBS to be developed, Section 1.1, titled "Purpose" is about the

__________ of the __________ and is not about the __________ of the __________.

(d) A good UM for a CBS is written from the viewpoint of a user of the CBS. It is usually structured as a list of __________ __________, one per section, each describing one way that the user might want to use the CBS. Within each section about one __________ __________, the manual describes typical __________s for achieving the section's __________ __________. The descriptions show the detailed interaction between the user and the CBS, often with the help of __________ diagrams.

(e) There are some CBSs that interact with the environment, e.g., the wind, birds, the weather, and not people. How can you write a UM for such a CBS?

(f) The following sentence has the word only in the standard place before the main verb.

The spam filter only marks the e-mail messages that it considers to be spam.

This placement of only *may* be incorrect with respect to the intended meaning of the sentence, but then again, it *may* be correct. In reading it, we cannot be certain what the writer meant. Give three interpretations for this sentence.

(g) What is the common property that holds for all kinds of ambiguity, fuzziness, generality, imprecision, incompleteness, inconsistency, and vagueness that appear in written or spoken specifications of, or statements about, requirements of a CBS?

Therefore, what must the requirements analyst do?

8. **[15 total marks]** Cost Estimation and NFRs or Quality Requirements

(a) The conundrum (contradiction) about cost estimation for a CBS is that the cost estimation is ______________

at the very time that there is not enough ______________ to do it well, namely at the very beginning of the project. The estimated cost to build a CBS is needed in order to know whether building the CBS is

______________ at what the customer is willing to ______________ for it to be built.

(b) We know that the number of lines of communication c in a group grows quadratically with the number of people n in the group. Specifically,

$c = \frac{n \times (n-1)}{2}$.

On the assumptions that

1. each person adds 6 hours of work per day (8 hours - 1 hour for lunch - 1/2 hour for each of two breaks), and

2. each line of communication soaks up 1 hour of work per day,

with how many people in the group, does the required communication add up to *more* than the hours of work provided by the people in the group?

(c) For what kind of CBS is "the CBS is fun to use" a relevant requirement?

(d) Why is "the CBS is fun to use" considered a non-functional requirement (NFR)?

(e) Given this property of the fun NFR, describe one way to decide whether a CBS meets the fun NFR.

9. **[8 total marks]** Verification, Validation, and Inspection

Validation is showing that you are building the ______________ ______________ according to the CBS's ______________,

while verification is showing that you are building the ______________ ______________ according to the CBS's ______________,

(a) Taking the above distinction into account, if $D, S \vdash R$ is the validation formula, then for a program P, what can be considered the verification formula?

(b) Recall that there are several kinds of review, including formal inspection. What other kinds of review does a formal inspection contain as component parts?

(c) Manual testing of a state machine model with input can be considered what kind of review or inspection?

10. **[18 total marks]** State Machines and Linear Temporal Logic

(a) Consider the following specification written in Temporal Logic:

$\Box(\mathit{Initial} \Rightarrow (\mathit{Initial}\ \mathcal{W}\ (\mathit{WhiteSpace} \vee \mathit{Letter} \vee \mathit{Digit} \vee \mathit{Otherwise})))$
$\Box((\mathit{Initial} \wedge \mathit{WhiteSpace}) \Rightarrow \bigcirc \mathit{Initial})$
$\Box((\mathit{Initial} \wedge \mathit{Digit}) \Rightarrow \bigcirc \mathit{Num})$
$\Box((\mathit{Initial} \wedge \mathit{Letter}) \Rightarrow \bigcirc \mathit{Id})$
$\Box((\mathit{Initial} \wedge \mathit{Otherwise}) \Rightarrow \bigcirc \mathit{Error})$

$\Box(\mathit{Id} \Rightarrow (\mathit{Id}\ \mathcal{W}\ (\mathit{Letter} \vee \mathit{Digit} \vee \mathit{Otherwise})))$
$\Box((\mathit{Id} \wedge (\mathit{Letter} \vee \mathit{Digit})) \Rightarrow \bigcirc \mathit{Id})$
$\Box((\mathit{Id} \wedge \mathit{Otherwise}) \Rightarrow \bigcirc \mathit{Initial})$

$\Box(\mathit{Num} \Rightarrow (\mathit{Num}\ \mathcal{W}\ (\mathit{Digit} \vee \mathit{Otherwise})))$
$\Box((\mathit{Num} \wedge \mathit{Digit}) \Rightarrow \bigcirc \mathit{Num})$
$\Box((\mathit{Num} \wedge \mathit{Otherwise}) \Rightarrow \bigcirc \mathit{Initial})$

$\Box(\mathit{Error} \Rightarrow (\mathit{Error}\ \mathcal{W}\ (\mathit{false})))$
$\Box((\mathit{Error} \wedge \mathit{true}) \Rightarrow \bigcirc \mathit{Error})$

Draw the specified finite state machine.

(b) Now, recognize that in each state with an $Otherwise$ event, $Otherwise$ means something different. For any state, $Otherwise$ means "any event but the other events that emerge from the same state". Define each of the three $Otherwise$s in terms of the other predicates.

1. $Otherwise$ of $Initial$:

2. $Otherwise$ of Id:

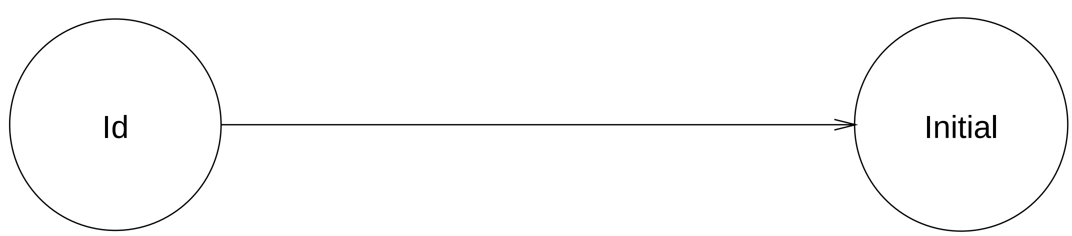
3. $Otherwise$ of Num:

(c) In the FSM you made for (a), consider the transition from Id to $Initial$ under the event $Otherwise$. The basic FSM notation indicates neither any conditions on the transition nor an action to happen when a transition is taken. The UML state machine notation allows specifying both conditions on the transition and an action to happen when a transition is taken.

Assume that $Otherwise(x)$ means that the actual otherwise character that triggers the $Otherwise$ event is available to be used in the transition's conditions and actions by mentioning the parameter x.

On the transition line in the diagram below, write the UML expression associated with this transition that says

"Whenever in state Id,
if the input is the otherwise character x and the x is a punctuation character ($punct(x)$) then
first the current value of $token$ is emitted ($emit(token)$),
and then $token$ is assigned the value of x.
Finally, the next state is $Initial$."



11. **[10 total marks]** The Requirements Iceberg

(a) What are the two main effects of spending more time studying the problem to be solved by a CBS, i.e., doing more up-front RE for the CBS?

1.

2.

According to a study of large NASA projects, these two effects combine to mean that there will be

______________ cost ______________ in the total development of the CBS.

(b) The LRT construction in Kitchener-Waterloo is being run as a "design–build" project, that is, designing, i.e., requirements analysis, and building happen at the same time.

In software engineering terms, this is taking an ______________ approach to building the LRT.

As a consequence, just as with agile software development, the ______________ and ______________ needed for the construction are unpredictable.

The discovery of previously unknown buried infrastructure in the streets that are dug up for the construction results in expensive dealing with the buried infrastructure or rerouting of the LRT or both. For software, we are often able to determine all requirements before starting to implement the software, thus making the development of the software as cheap, as fast, and as predictable as possible. Why is determining buried infrastructure before starting construction of the LRT not possible?

Why is a three-stage process, (1) determining all buried infrastructure throughout Kitchener-Waterloo, *then,* (2) using what was determined to make good plans for the LRT, and *finally,* (3) building the LRT throughout Kitchener-Waterloo, not feasible