Dipartimento di Elettronica e Informazione
Politecnico di Milano

prof. Elisabetta Di Nitto, Raffaela Mirandola, Luca Mottola

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (39) 02-2399.3400
Fax (39) 02-2399.3411

Software Engineering II

17 January 2014

Last Name

First Name

Id number (Matricola)

Note

1. The exam is not valid if you don't fill in the above data.
2. Write your answers on these pages. Extra sheets will be ignored. You may use a pencil.
3. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden.
4. You cannot keep a copy of the exam when you leave the room.

Question 1 Alloy (7 points)

Specify in Alloy the following informal statements:

1. A person can have several friends (who are also persons). The friendship relationship is symmetric (that is, if A is a friend of B then B is a friend of A).

2. A function "linked to" can be defined from the friendship relation. The function yields the set of direct or indirect friends of a given person P. That is, it contains all the persons Q such that either Q is a friend of P or Q is a friend of a friend of P or Q is a friend of a friend of ... a friend of P.

3. The predicate unrelated (X, Y), where X and Y are persons, is true when the set of persons linked to X and the set of persons linked to Y are disjoint.

4. The predicate linkedButNotFriend (X, Y), where X and Y are persons, is true if X and Y are not direct friends but they are linked.

5. A couple is composed of exactly two distinct persons.

6. No bigamy is allowed (each person should belong to at most one couple).

7. The predicate coupleButUnrelated(X, Y), where X and Y are persons, is true if X and Y are a couple but they are unrelated.

Solution

```
sig Person {
 friends: set Person
}

sig Couple {
 partner1: one Person,
 partner2: one Person
}{partner1 & partner2 = none}

fact noSelfFriends {
 no p:Person | p in p.friends
}

fact simmetry{
 all p: Person | all p1: Person | p1 in p.friends <=> p in p1.friends
}

fact monogamy {
 all disj c1, c2: Couple | (c1.partner1+c1.partner2) & (c2.partner1+c2.partner2)=none
}

fun linkedTo [p:Person]: set Person {
 p.^friends
}

pred linkedButNotFriend[p1, p2: Person] {
 p1 & p2 = none and (p1 in linkedTo[p2] or p2 in linkedTo[p1]) and not p1 in p2.friends
}
```

```
pred unrelated[p1, p2: Person] {
 p1 & p2 = none and not (p1 in linkedTo[p2] or p2 in linkedTo[p1])
}

pred coupleButUnrelated[p1, p2: Person] {
 unrelated[p1, p2] and some c: Couple | (p1+p2) in (c.partner1 + c.partner2)
}

run coupleButUnrelated
run unrelated
run linkedButNotFriend
```

Question 2 Testing (5 points)

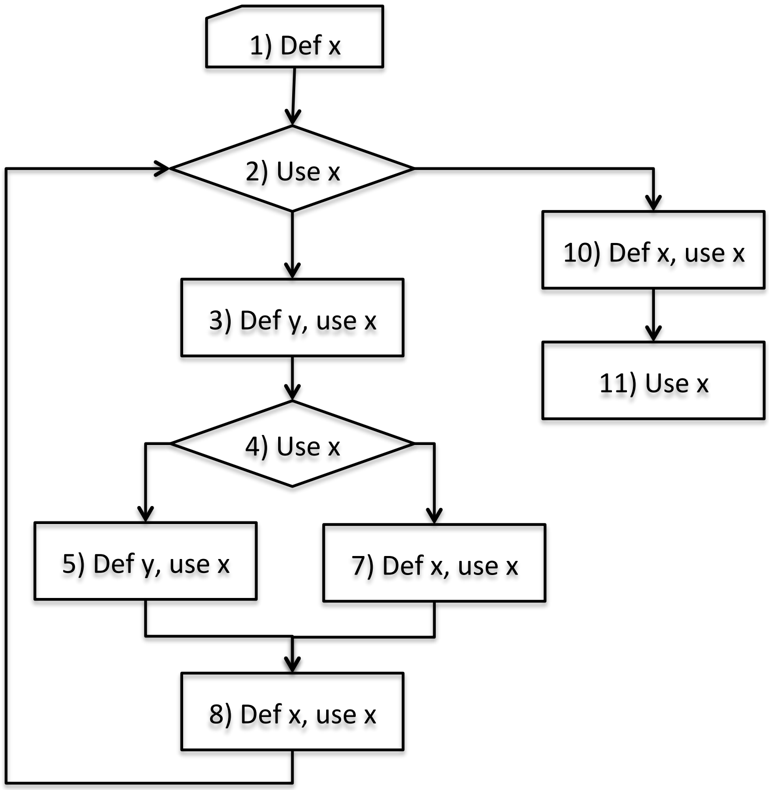
Consider the following fragment of code written in a C-like language:

```
int foo() {
1    x = input();
2   while (x > 0) {
3         y = 2 * x;
4         if (x > 10)
5           y = x - 1;
6         else
7           x = x + 2;
8       x = x - 1;
9    }
10  x = x - 1;
11 return x;
}
```

You are to accomplish the following:
- draw the control flow graph of the program;
- provide the use-definition information for variables x and y;
- point out a potential issue with this code that data flow analysis would be able to spot;
- using symbolic execution, identify the path condition that allows the program to follow the path 1 2 3 4 5 8 9 2 3 4 6 7 8 9 2 10 11.

Solution

1. Control flow diagram that includes def-use definitions



2. Def-use pairs (not explicitly required in the text)

x	<1, 2> <1, 3> <1, 4> <1, 5> <1, 6> <1, 7> <1, 8> <1, 10> <7,8> <8, 2> <8, 3> <8, 4> <8, 5> <8, 7> <8. 8> <8, 10> <10, 11>
y	No pair exists

3. The potential issue spotted by data flow analysis is exactly that y is defined by never used in the program.
4. Symbolic execution

   1. x = X
   2. X>0
   3. y=2*X
   4. X>10
   5. y=X-1
   8. x = X-1
   2. X-1 > 0
   3. y = 2*(X-1)
   4. X-1<=10
   7. x = X-1+2=X+1
   8. x = X+1-1=X
   2. X<=0

   The path condition for the given path is then X>10 and X<=11 and X<=0

   This is clearly an inconsistent condition. Thus, the given path is impossible.

5. **Questions 3 and 4 Planning and development (10 points)**

Politecnico exploits the Beep platform to support the management of course material. The application allows students to register to the system, visualize the courses to which they are enrolled, select other courses to which to enrol, download the documents associated to a course and, when allowed by the instructor, upload documents and participate in forum discussions.

Instructors can activate courses, visualize enrolled students, publish course materials, manage one or more discussion fora.

Politecnico wants to extend the existing application as follows:

- Allow instructors to associate the same forum to different courses or classes of the same course.
- Allow instructors to define and publish forms similar to those offered as part of googleDrive and to require students to fillin these forms.
- Enable the integration with a configuration management system so that students can develop their software assignments under the support of the configuration management system (possibly offered by an external system as googleCode or github), and can have the result of their work loaded on Beep in the simplest and most automatic way.

The Beep development team is composed of an expert project manager, three persons very expert on the Beep system and two persons just graduated in "Ingegneria Informatica".

A) Please explain how you would proceed with the definition of a plan for the activities of the development group. In particular, the group has to take care of the following goals: 1) Beep has to be maintained in its normal operation and any problem that should occur has to be corrected; 2) the new extensions have to be defined, designed, developed, tested and deployed without causing problems to the users of the system. Identify the activities to be executed and their dependencies. Moreover, propose a possible allocation of resources to these activities or a criterion to allocate them. You do not need to calculate costs and efforts in quantitative terms.

B) Assuming that we have finalized the planning phase, focus on the requirement analysis and on the definition of a system integration and test plan. More specifically:
   - define one or more use case diagrams where you identify the actors interacting with the new extensions and the associated use cases (you do not need to describe these cases in detail);
   - explain how you want to proceed with the integration and the corresponding testing activity.

You can add any hypothesis you think is useful, provided that you specify it explicitly in your solution.

Solution

The extensions to be developed are three. Thus, we can assume that we organize the team in three subgroups, each focusing on the development of one of these extensions. Moreover, one of the groups will have to take care, part time, of the management and maintenance of the existing platform. If we pair experts and junior team members we can exploit the opportunity of the joint work to have the experts training the juniors.

One of the experts, let's call her ExpA, and one of the junior members, let's call him NewA, could allocate a small portion of their time to monitor the operation of the current platform and to solve problems when they arise. For instance, we can assume that ExpA spends 5% of her time in this activity while NewA spends 10%. This way ExpA will also have the chance to train NewA on the structure and operation of the application.

From a detailed analysis of the extensions we can understand that the first one is the simplest. It consists in transforming the relationship between course and forum from 1 to n into m to n, in modifying the operation that, given a course, allows the user to visualize all forums associated to it, and in modifying the operation for creating a new forum so that the user is be asked to which courses he/she wants to associate the new forum. We could also add a new operation that modifies the visibility of a forum in a set of course even after its creation.

The second extension is not particularly complex as well. Beep already offers functions to allow instructors to create an html page. We would need to add: a function to guide the user in the creation of a form, a function that acquires the data from course users and, finally, a function that allows the instructor to gather the collected results. This last one could simply consists in the generation of an electronic sheet that the instructor could elaborate off-line, out of the Beep platform.
The third extension is the most complex one as it requires that the project team studies the way configuration management systems work. Without this preliminary study it is difficult to decompose the extension in smaller functions.
Given these considerations, the team could be organized as follows:
The development of the third functionality could be assigned to ExpA, who could focus on this for 95% of her time, and to NeoA, who could focus on it for the 90% of his time. This way, both team members will be allocated 100% (part on the existing application and part on the third extension).
The work on the third extension will start immediately with the study of the most common configuration management systems (namely, git and svn). Then an accurate analysis will be performed that will result in the identification and design of the functions to be implemented. The task will end with the implementation and testing of all functions associated to the third extension. We can provide a first estimation for the duration of this activity, but this will have to be re-evaluated after the initial study.
The development of the first functionality could be assigned to another expert, let's call him ExpB. Knowing the Beep system, he will develop it quite easily.
The development of the second functionality will be assigned to the group composed of the third expert, let's call this ExpC, and of the second junior, NeoB.
All development and testing activities will take place in an environment separated from the one used for the normal operation of the platform.

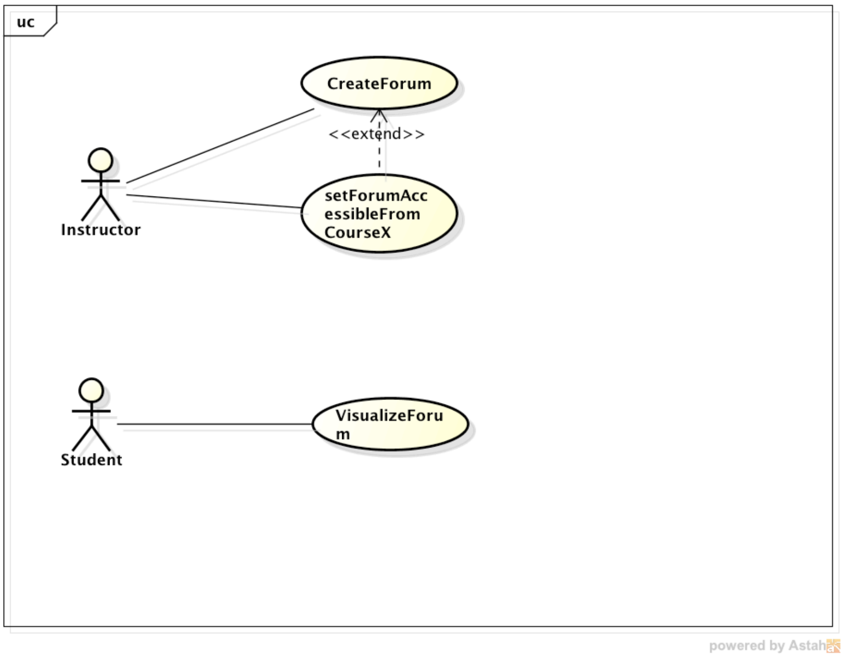
At the end of his work, ExpB will define the integration and acceptance tests for the extension of which he is responsible as well as for the others and will start executing the tests (of course, the unit tests will have to be performed as part of the development activities).
After passing all integration and acceptance testing, the extension of which ExpB is responsible will be released for beta testing to a limited number of instructors and students.
At this point ExpB, in cooperation with ExpC and NeoB, will manage the integration and testing of the second extension and then, helped by ExpA and NeoA, of the third extension.

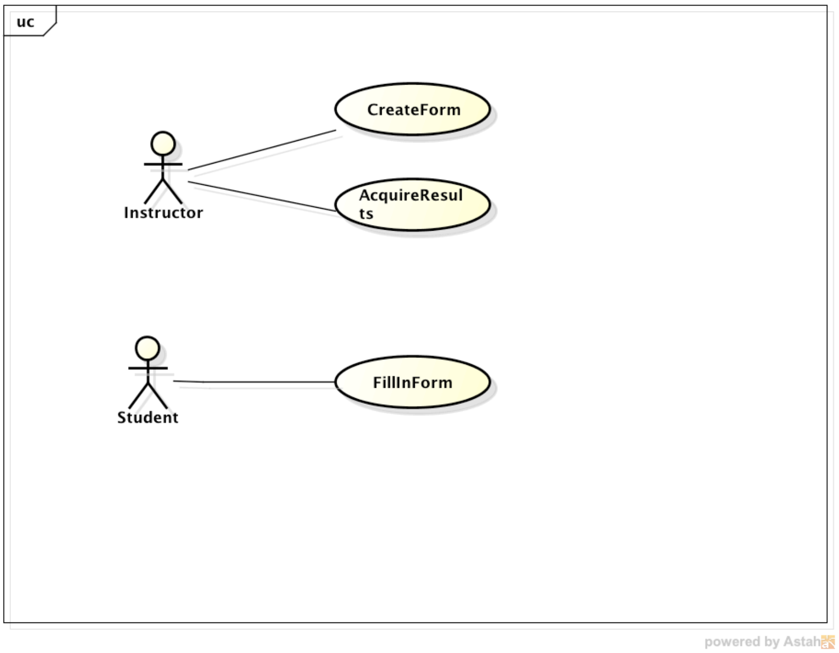
The project manager will monitor the activities and help the others in case of need.

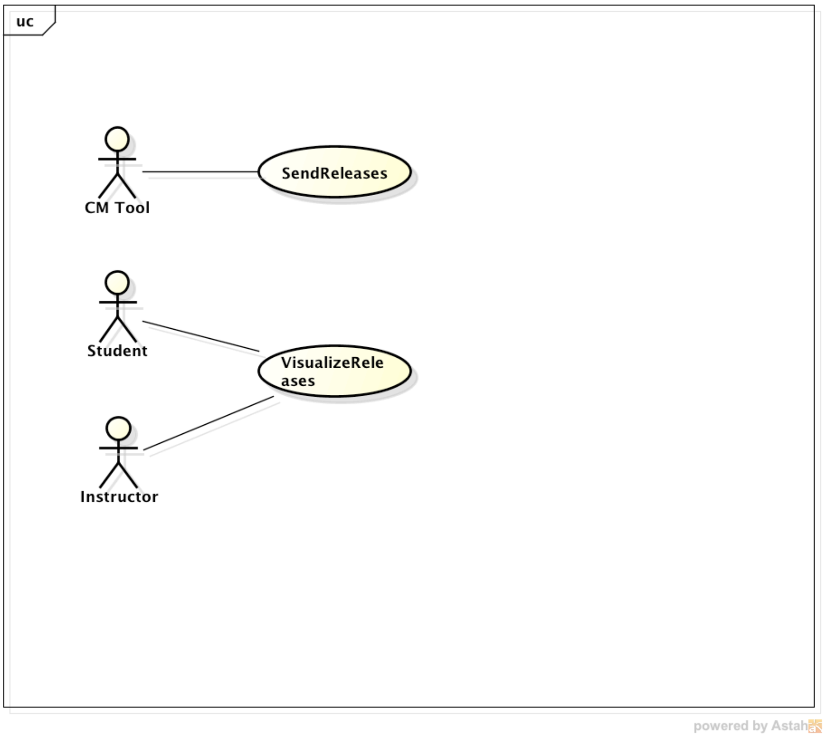
The risks associated to this planning are related to two main factors:

- If ExpB is unavailable for any reason, the work associated to the development and integration of the first extension will be blocked. This risk is mitigated by the fact that the other two experts will be able to take over with a minimum time for analyzing the status of ExpB's work.
- The work concerning the integration of the configuration management system remains quite vague at the time of the planning. As a consequence, it is not possible to estimate, with a reasonable level of confidence, the needed effort and cost. To mitigae this risk the project manager decides to monitor this activity continuously and to reassess the situation periodically to be ready to replan or refocus the work.

Attori e use case diagrams


uc
CreateForum
<<extend>>
setForumAccessibleFromCourseX
Instructor
Student
VisualizeForum
powered by Astah


uc
CreateForm
Instructor
AcquireResults
Student
FillInForm
powered by Astah


uc
CM Tool
SendReleases
Student
VisualizeReleases
Instructor
powered by Astah