Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto and Matteo Rossi

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

Software Engineering II

January 23rd, 2019

Last Name
First Name
Id number (Matricola)

Note

1. The exam is not valid if you do not fill in the above data.
2. Write your answers on these pages. Extra sheets will be ignored. You may use a pencil.
3. Incomprehensible hand-writing is equivalent to not providing an answer.
4. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden.
5. You cannot keep a copy of the exam when you leave the room.
6. The exam is composed of three exercises. Read carefully all points in the text!
7. **Total available time: 2h**

Scores of each question:

Question 1 (MAX 7) _____

Question 2 (MAX 6) _____

Question 3 (MAX 3) _____

Question 1 Alloy (7 points)

Consider a system that supports the review process for scientific articles. An article is authored by one or more researchers and has a content. Authors submit the article. Then, this is assigned to some reviewers and moved to the "under review" state. Reviewers are researchers as well. When all reviews are released, based on their outcome, the article can be in one of the following states: "accepted" or "rejected". More specifically, if all reviews are ACCEPT, then the paper is accepted.

Starting from the following signatures in Alloy:

```
sig Content {
  numberOfPages: Int
} {numberOfPages > 0}

sig Article {
   authors: some Researcher,
   content: Content,
   status: Status,
   reviewers: set Researcher
}
```

A. Complete the specification adding the signatures and facts that allow you to model the description above.
B. Model the following constraints:
   - A reviewer cannot review his/her own articles.
   - Reviewers do not deliver reviews for papers to which they are not assigned.
   - Articles can enter the reviewing process only if they have been assigned at least 3 reviewers.
   - Articles are in the "accepted" state if and only if all reviewers agree on acceptance (if you have already considered this constraint as part of point A, you do not need to address it again here).

Solution
A possible solution for the exercise is the following. Other ones are also possible, of course.

```
sig Content {
  numberOfPages: Int
} {numberOfPages > 0}

sig Researcher {
  review: Article -> Review
}

/* in the signatures below the cardinality lone is used instead of the usual "one"
   because this way the Alloy analyser is not forced to create instances
  if they are not used in the specific world that is generated.
  This is because otherwise the number of instances in this case would be
  relatively large in this case */

abstract sig Review {}
```

```
lone sig ACCEPT extends Review {}
lone sig REJECT extends Review {}

abstract sig Status {}
lone sig Submitted extends Status {}
lone sig UnderReview extends Status {}
lone sig Accepted extends Status {}
lone sig Rejected extends Status {}

sig Article {
   authors: some Researcher,
   content: Content,
   status: Status,
   reviewers: set Researcher
}

fact noAuthorAndReviewerOfTheSameArticle {
  all a: Article | a.authors & a.reviewers = none
}

fact noResearcherProducedUnsolicitedReviews {
  all r: Researcher, a: Article | a in ((r.review).Review) implies
                                   r in a.reviewers
}

fact articlesUnderReviewHaveAtLeast3Reviewers {
  all a: Article | a.status = UnderReview implies #a.reviewers > 2
}

fact acceptedArticleStatus {
  all a: Article | a.status = Accepted iff
                        (a.(a <: (a.reviewers).review)) & REJECT = none
}

pred show[]{ #Article = 1 and some a: Article | a.status = Accepted}

run show for 20
```

Question 2 Requirements (6 points)

A software house has been tasked to create *TSPmonitor*, a crowd-based application that allows users to report on the status of Public Transport in the city. In particular, the application should allow citizens to provide short reports on the status of the bus, tram, and metro lines they take. The reports created through the application include the level of crowdedness of the line, the comfort of the trip, and the punctuality of the service. Reports created through the application are stored for statistics computation and data mining purposes. In addition, if the user allows it, reports are automatically transformed into tweets that make use of predefined hashtags.
The application can also notify users who request it about changes in the detected status of lines (e.g., when a line becomes crowded), and it can suggest alternative paths when problems strike their preferred lines. Also, thanks to its mining capabilities, the application can provide forecasts on the status of lines in the next 3 days.

Q1: Given the system described above and referring to the Jackson-Zave distinction between the world and the machine:

1. For each phenomenon already present in the table, determine whether it is shared or not, and who owns/controls it.
2. In addition to the phenomena already present in the table, identify:
   - One world phenomenon that is not shared with the machine.
   - One shared phenomenon controlled by the world.
   - One shared phenomenon controlled by the machine.
   - One machine phenomenon not shared with the world.

Please use the table below to answer this question (the number of rows does not necessarily correspond to the number of phenomena you can identify for this case).

Phenomenon	**Shared nor not**	**Who controls it**	**Short explanation (if the name you assign to the phenomenon is not self-explanatory, or you need to specify some conditions on the phenomenon)**
Bus drivers strike			
Accident occurs between tram and car			
Citizen reports crowdedness of bus			
Citizen reports delays in tram passages			
Citizen reports heating off in bus			
Citizen sets preference saying that she usually takes line X			
TSPmonitor informs citizens that her usual bus is crowded			
TSPmonitor advises citizens to take alternative line			
TSPmonitor sends tweet informing that bus is crowded			

TSPmonitor computes forecast of level of crowdedness of bus			
TSPmonitor computes alternative path for user			

Q2: Define in natural language one specific goal for *TSPMonitor*, one domain assumption and one requirement referring to the phenomena identified above.

Solution

Phenomenon	**Shared nor not**	**Who controls it**	**Short explanation (if the name you assign to the phenomenon is not self-explanatory, or you need to specify some conditions on the phenomenon)**
Bus drivers strike	N	W	
Accident occurs between tram and car	N	W	
Citizen reports crowdedness of bus	Y	W	
Citizen reports delays in tram passages	Y	W	
Citizen reports heating off in bus	Y	W	
Citizen sets preference saying that she usually takes line X	Y	W	
TSPmonitor informs citizens that her usual bus is crowded	Y	M	
TSPmonitor advises citizens to take alternative line	Y	M	
TSPmonitor sends tweet informing that bus is crowded	Y	M	

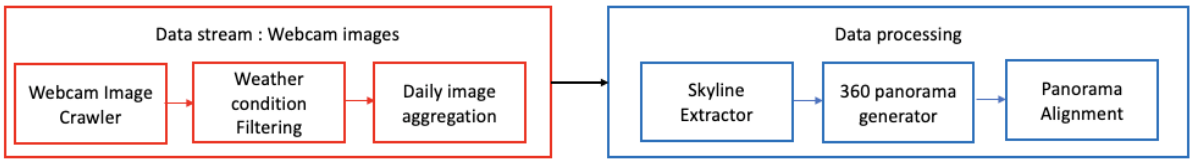
TSPmonitor computes forecast of level of crowdedness of bus	N	M	
TSPmonitor computes alternative path for user	N	M	
People board bus	N	W	
User gives permission to publish tweets	Y	W	
TSPmonitor forecasts that tomorrow at 8am bus 45 will be very crowded	Y	M	
TSPmonitor stores report from user	N	M	

Goal: Users want to be kept up-to-date on the level of crowdedness of the buses that they usually take.

Domain assumption: user signaling her preference for line X is accurate.

Requirement: When more than N citizens (with N a configurable system parameter) report that bus X is crowded, the system informs all users that habitually take bus X that the bus is crowded

Question 3: Design (3 points)
Consider the following figure which represents the high level architecture of a real software system called SNOW.



The goal of SNOW is to exploit the operational value of information derived from public web media content, specifically from mountain images contributed by users, to support decision making in a snow-dominated context. The system performs the following activities:

- crawls geo-located images from heterogeneous sources that are installed on the mountains (*Webcam Image Crawler*);
- filters the images that are not clear enough (*Weather condition Filtering*);
- aggregates the images of the same day to cope with possible issues such as varying illumination conditions and moving obstacles thus generating the daily image to be consider in the next steps (*Daily image aggregation*);

- identifies individual peaks, and extracts a snow mask from the portion of the image denoting a mountain (*Skyline Extractor*, *360 panorama generator*, *Panorama Alignment*).

Assume that the availability of the Data stream part is 99.999% and the availability of the Data processing part is 99.98%. Answer to the following questions:

1. What is the total availability of SNOW? You do not need to provide a precise value. The order of magnitude of such availability together with an explanation would be sufficient.
2. Given the availability of the Data stream part (99.999%), what can you infer about the availability of its sub-components?
3. Assume that you want to improve the availability of the data processing part by replicating it. How many replicas would you need to achieve 99.999%?

Solution

1. Being a series of two parts, the total availability of SNOW is determined by the weakest link, that is, the data processing part. So, its value is close to 99.98%
2. The data stream component is itself a series of multiple components. We can deduce that the sub-component with the lowest availability has about 99.999% availability.
3. We should configure multiple pipelines, each composed of the three sub-components of Data processing. This is possible under the assumption that there are no obstacles to the replication of all three sub-components and that each of the replicated pipelines can autonomously perform the needed work. In this case, the availability of the whole subsystem would be $1 - \prod_{i=1}^{n}(1 - A_i) = 1 - (1 - 0.9998)^n$. If n = 2 we get to 1-4/1000000 which is certainly higher than 99.999%.