Software Architecture and Engineering 2015
Project Part 1

Published: March 02, 2015
Submission: March 29, 2015 (midnight); this is a firm deadline

Your task is to design a social networking system. The emphasis of your task is on designing the data model, expressing it in UML, formalizing it in Alloy, and checking whether certain properties hold for your Alloy model.

We intentionally made parts of the system description below ambiguous. It is your responsibility to clarify any ambiguities with your TA (your "customer").

1 Description

Each user registered at the social networking system has a personal profile. Groups of users can have a group profile. Users can have friends, and a friendship relation is always mutual. Each user can select to follow other users and groups in which case their content (that is visible to her) would be presented on his newsfeed. A user can also block another user from seeing any of her content.

Users can post different types of content: photos, posts, (personal) messages, personal details and comments. A post contains text, and may also have links to photos that have been posted before. Users can comment on posted content, such as photos and posts, but they cannot comment on personal messages or personal details. Users can also comment on already posted comments. A message may include text and previously posted photos, and is directed to a specific user to which the message is visible. A message may (at the sender's discretion) be visible to others, but they cannot comment on the message.

Users are identified using their user name and email address, and can include personal details in their profile. For each such personal detail, a user can choose who can see it just like with other content. For example, the user's name may be public (visible to anyone) while the age may be visible only to the user's friends.

To restrict who can access their posted content, users select which circle of users can see each posted content. There are five possible circles:

- Private: contains only the poster (i.e., the user who posted the item).
- Friends: contains the poster and the poster's friends.

- Friends of friends: contains the Friends circle and the friends of the poster's friends.
- Transitive friends: contains all users who have a friend chain to the poster.
- Public: contains all registered users.

Users can create groups of users; for example, to group multiple users with a common interest and who want to share content. The users in a group are called members, and each group member can see any content posted to the group. Some of the group members are designated as the group's administrators. An administrator can add and remove other members from the group, and can also modify and remove content posted to the group. Any content posted to the group is owned by the group, visible to all group members, but can only be modified by group administrators. Users who are not members of a group can still follow the group, but will see content posted to the group only if it is public.

2 UML Model

Task A. Create a UML class diagram of the system as described above. Include all the relevant relations and details. In addition, document any detail that cannot be encoded in the UML class diagram. OCL specifications are not required. Use the best design practices you have seen in the lectures.

3 Alloy Model

Task B. Create an Alloy model of the system described above. Include all the relevant details, relations and facts. Ask your TA in order to clarify any ambiguities. In addition, document any detail that cannot be encoded in the Alloy model.

Task C. Based on your Alloy model, formalize the following predicates in Alloy:

Valentin

Simon

Jan

- canSee(user,content) holds iff the given user can see the given content
- canModify(user,content) holds iff the given user is allowed to modify the given content
- isOnNewsFeed(user,content) holds iff the user should see the content on his newsfeed; this is the case for content posted by users and to groups the user follows, to groups the user is a member of, and messages posted to the user, but only content visible to the user.

The above three predicates must be encoded without quantifying over all the instances of a signature (e.g. over all users or content). For example, if user is an argument to the predicate, instead of:

all u1,u2 : User || u1 **in** user.friends **and** u2 **in** u1.friends ⇒ ...

you should use:

all u1 : user.friends || **all** u2 : u1.friends || ...

which avoids quantifying over all Users.

Task D. Encode and check the following properties in Alloy:

1. Comment chains are acyclic
2. A user can modify only content they can see
3. A user can modify all the content they have created
4. If a post or message includes photos, the photos are visible at least to all the people that can view the post
5. Each group has members
6. A user's newsfeed only has content visible to her
7. A user cannot see any content created by a user that blocks them

If a property does not hold either fix the model to make it hold (if it should hold by the specification), or explain why it shouldn't hold.

Task E. Generate the following instances using Alloy, showing that your model is not overly restricted, or explain why they are infeasible if you cannot generate them:

1. A comment chain that is 5 comments long
2. 3 users that form 7 different groups, each with a different set of members
3. 4 users, where each has at least one friend, but not everyone is a transitive friend of everyone else
4. A user that can see a post of a user that is a friend of friend (but not a direct friend), which has privacy level "friend of friend" and includes a photo from a fourth user
5. A post that includes a photo created not by the poster, where the photo is not public
6. A post with the privacy level "friends of friends", which includes a photo created by a friend of the creator of the post. Some friend of the creator of the post must be able to see the post, but not the photo

4 Deliverables

Submit your solution by email to your TA, including:

1. A Zargo file of the UML class diagram from Task A in ArgoUML
2. An Alloy file with all the required code from Tasks B–E. The file must include short comments that explain your formalization
3. A pdf file that includes:
   - The UML class diagram from Deliverable 1
   - A list of details from the project description that cannot be expressed in the UML model
   - The Alloy model from Deliverable 2
   - A list of properties from Task D that you were not able to check, each with a short explanation why the property does not hold
   - A list of instances from Task E that you were not able to produce, each with a short explanation why the instance is not feasible
   - Diagrams of all the generated instances from Task E

5 Resources

- ArgoUML: http://argouml.tigris.org/
- Alloy: http://alloy.mit.edu/alloy/
- PDF creation: You can use Microsoft Word to create your document and export it to PDF