University of Pittsburgh

Department of Computer Science

CS 2310:
Multimedia Software Engineering

Social Interaction as an extension to SIS (Theory part):

Final Report

Done by:
Rakan Alseghayer (raa88)
Yingjie Tang(yit20)

- ## Introduction:

In this report, I will present my work in the project of Social Network and Interaction as an extension to the SIS abstract machine. I will present the abstract machine, and then show how and what part we have extended to incorporate the social simplified model we have. After that, I will demonstrate the two approaches that I have implemented as a result of our discussions and negotiations. Finally, I will show some examples to overview the abstract social model in isolation from the SIS system.

- ## The Abstract Machine:

Input to the abstract machine is: (P, S, Po, Cycle1, …, Cyclen)

Where the P is the problem set, and S is the solution set.

Cyclei is the computational cycles that the machine go through to change from problem set to another in hope to reach S eventually.

The operators are:

- P1 -enum< P2 where P2 = { y: y is related-to some x in P1, e.g. d(x,y) < D}
- P1 >elim- P2 where P2 = {x: x is in P1 and x is in S}
- P1 >conc= P2 where P2 = {x: x is in P1 and th(x) above predefined threshold t}
- pj +Aijadap= pk is to adjust pj's based on input attribute Aij, for example, by appending it to pj
- pj =propAij+ pk is to output/propagate attribute Aij to peer super-components

Those operators are described here:

1) *Enumeration:*
   Is the operator that extends the problem set and adds to it, so that we explore all the possible available elements that relates to the problem set.
2) *Elimination:*
   This operator will exclude any element in the input set that doesn't match the criteria of elements that belong to S "the solution set."
3) *Concentration:*
   This operator acts as an eliminator except for the fact that it might eliminate some solution elements, and concentrate on non-solution ones.
4) *Adaptation:*
   It gets the elements from the environment, and alters the elements based on the rules of the adaptation.

5) *Propagation:*
   Is the operator that does complement the adaptation by outputting the elements to the environment according to the propagation guides.

Now that we know the operators, we can say that Cycles are nothing but an application of set of operators that produces eventually a transformed set from an input set "based on the used operators."

- ## Course Project Focus Area:

In this project we have started with a simple social model, and applied that on the enumeration operator to see how that social model works. In my part, I have implemented the theoretical part, which is the social model.

- ## The Social Model:

First, we talk about how we defined the social model as an extension to the abstract machine. In order to approach that, we need to know our elements.

Out elements are P1, P2, … Pn, where each Pi represents a person in a group. Each person is defined by their [id, opinion, influencePercentage]. The id is a unique identifier for that person, and the opinion is the subject that would change during our computational cycles. Finally, the influence percentage is the indicator of how influential is that person (i.e. how capable is that person of changing people opinions).

After knowing the nature of our elements, now we describe the problem set that contains those elements. Each problem set is divided into two sub sets: High Influence Group, and Low Influence Group.

The interaction happens in a sub set basis. That is, H is interacting with L and vice versa. As expected, the result of this interaction is a potential change in the opinion of the interacted with group. This change is might happen, or might not. It is a result of the average influence of all the elements in the interacting group.
Assuming that a change happened, there are two possible scenarios: the change in the opinion is either influential or mind changing.

The influential model is changing the interacted with group opinion based on the opinion of the majority of the interacting group. For example, if the majority of the interacting group has the opinion X, then the change in the interacted group would be to the opinion X with some probability.

The other model causes the interacted group to change its opinion to the other one (in case of binary domain of opinions) regardless of what is the opinion of the interacting group.

Now having that said, we move to the formal definitions of the model. From the abstract machine, we have the definition of the enumerator operator:

- P1 -enum< P2 where P2 = { y: y is related-to some x in P1, e.g. d(x,y) < D}

This definition is extended in our model to the following:

- P1 -enum< P2 where P2 = { y: y is x in P1, with a chance that its opinion is change}

Each Pi is represented as a vector:

- Pi = [id, opinion, influence], where id is a unique integer, opinion is a binary variable that represent an opinion, and influence is a percentage.

The set of elements is divided into two subsets (Figure 1):

- H = {P1, P2, ..., Pi}
- L = {Pj, ..., Pn}
- Where H $\cup$ L = P

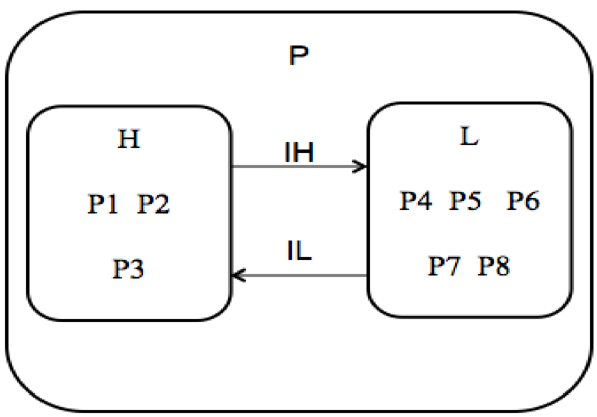

Figure 1: An example of a problem set with 8 elements.

During each interaction cycle, the interaction between the groups is represented by IH and IL, where IH is the percentage with which the group H affects the L group. The opposite holds for IL (Figure 1). The probability with which an interaction will change an opinion of an element in the other group is calculated as follows:

- IH = $\frac{\sum_{i=1}^{|H|} P.influence}{|H|}$
- IL = $\frac{\sum_{i=1}^{|L|} P.influence}{|L|}$

thus, each interaction cycle is a two way interaction. From H to L (H → L) and from L to H (L → H). However, the percentage of changing the opinion of elements is different in each interaction.

- **The Interaction Models:**

When the chances occur and the element Pi is to have its opinion changed, there are two ways to change that as mentioned above:

a. The Influencing Model:
In this mode, if the majority of the interacting group has the opinion X, then the element that is interacted with will have the opinion changed to X (even if it is already has X as its opinion).
b. The Mind Changing Model:
In this model, the opinion of the interacting group is irrelevant. Hence, if the element is going to change its opinion, it will change it to an opinion that is (not its current one) (e.g. flipping the opinion in binary domain opinions).

- **Demonstration of the Models:**

Here I will present some scenarios that I have crafted in isolation of the SIS test bed, also, a demonstration video of those experiments is available at: http://screencast.com/t/n48teU5c

First, I will show the behavior of the influential mode with two crafted examples:

1) | H | = 20
| L | = 180
H group influencing percentage = [7%, 10%]
L group influencing percentage = [1%, 3%]
5% in H have their opinion for '0'
95% in L have their opinion for '0'
Number of interactions is: 30 (bidirectional)

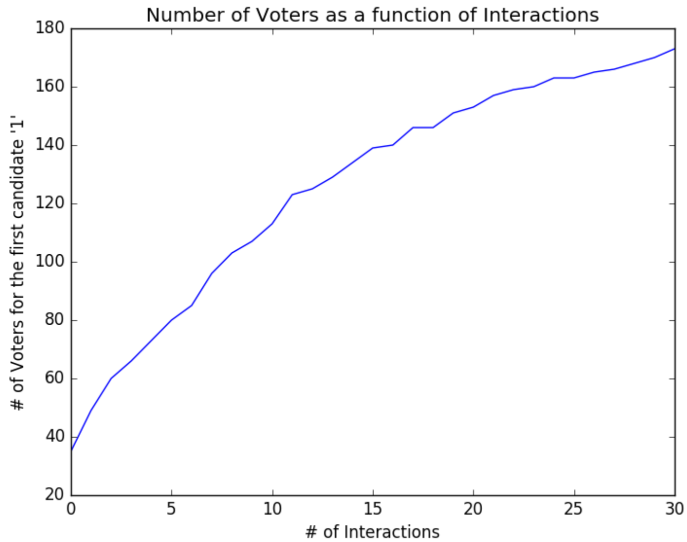

Figure 2: Almost a logarithmic function of interaction

2) | H | = 100
   | L | = 100
   H group influencing percentage = [80%, 90%]
   L group influencing percentage = [10%, 20%]
   5% in H have their opinion for '0'
   95% in L have their opinion for '0'
   Number of interactions is: 30 (bidirectional)

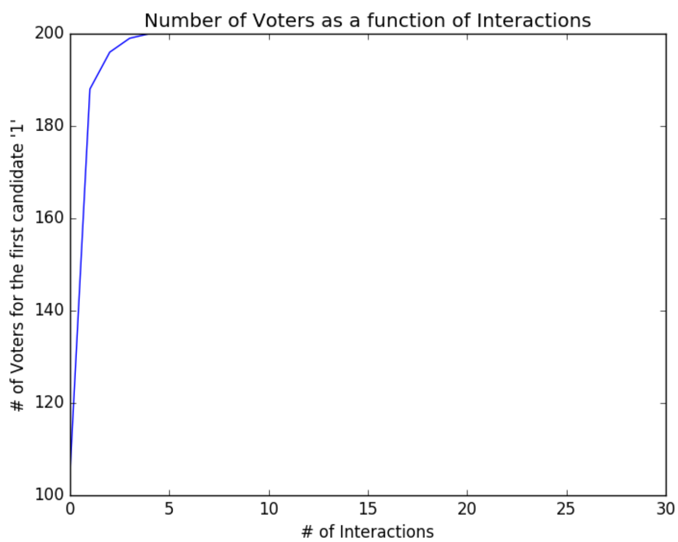

Figure 3: shows a power like curve

From the above two examples, we see that the influence percentage, controls the speed of convergence towards the "united opinion".

Now, I will show the behavior of the Mind Changing Model with two crafted examples as well (I will fix the same exact parameters to show the different behavior):

1) | H | = 20
   | L | = 180
   H group influencing percentage = [7%, 10%]
   L group influencing percentage = [1%, 3%]
   5% in H have their opinion for '0'
   95% in L have their opinion for '0'
   Number of interactions is: 30 (bidirectional)

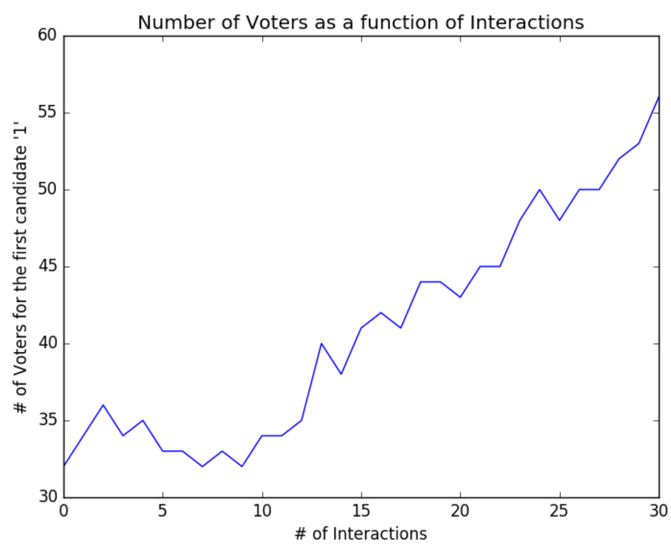

Figure 4: An increasing function of interaction with minor fluctuations

2) | H | = 100
   | L | = 100
   H group influencing percentage = [80%, 90%]
   L group influencing percentage = [10%, 20%]
   5% in H have their opinion for '0'
   95% in L have their opinion for '0'
   Number of interactions is: 30 (bidirectional)

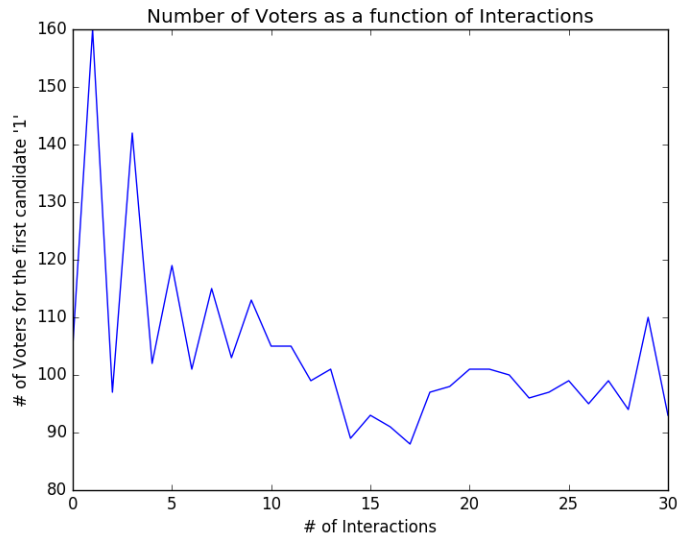

Figure 5: shows a physical wave like curve

The interesting part here is that the behavior is not predictable, it changes based on the people change their minds, hence, the fluctuation.

The code is available at this link:
https://dl.dropboxusercontent.com/u/19443460/code.zip

- **SIS Testbed Design:**

In our first simulation, the enumeration function should be defined to find out all the possible problem sets at before the final vote. Say the message coming to B super component says another group is supporting Hilary, then Hilary is added to the enumeration results as the previous Obama in super component A. The adaptation function is quite a tricky part in this simulation, we can set further design of the attributes of profiles more elaborate. In addition to the one attribute to tag each profile in each groups, we should add another vector which represents the probability of a specific profile electing a specific candidate as the president of United States. The whole attribute set of a specific profile should be p1 = <tag,<cand1, cand2>>, where tag uniquely identify the profile and cand1 and cand2 is the probability of electing candidate 1 or candidate 2 as the president of United States. The sum of cand1 and can2 should be 1 and the candidate with the highest value will be elected by this specific profile. When the message from other components comes, it will influence the judgement of this profile and the value of cand1 and cand2 will be changed and thus there is a chance for the electing target for a specific profile to change.

In the last step of the propagation, we use the majority vote to get the election target of the whole group. We want to see how the opinions from another group influence the opinions of a specific group.

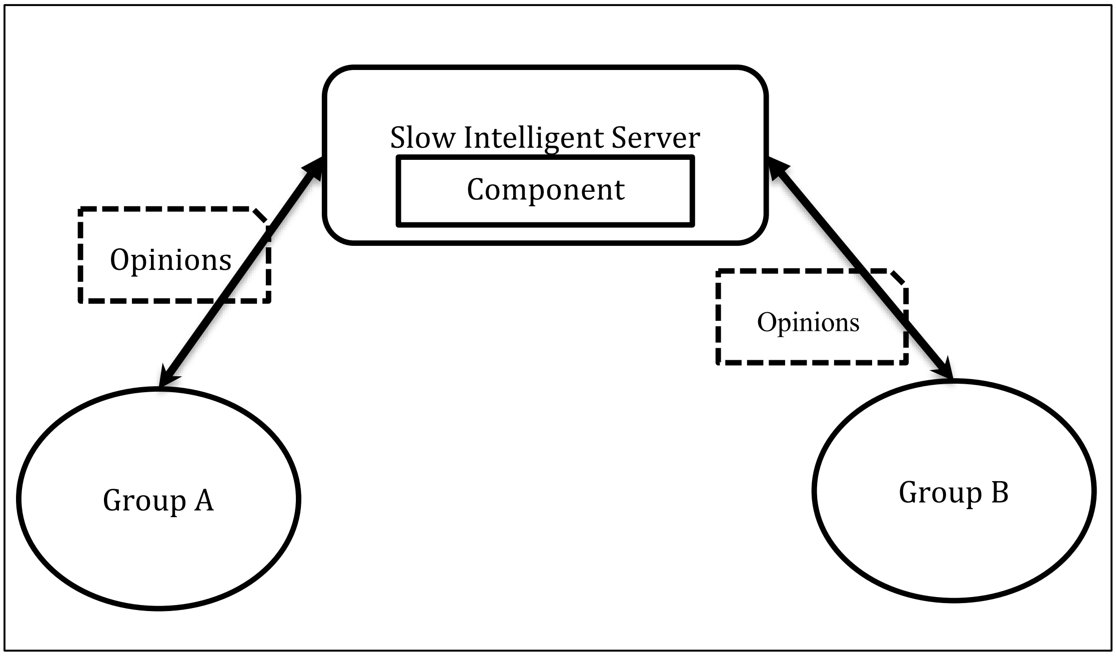

Figure 6. Components Design in SIS

- **SocialNetwork GUI:**

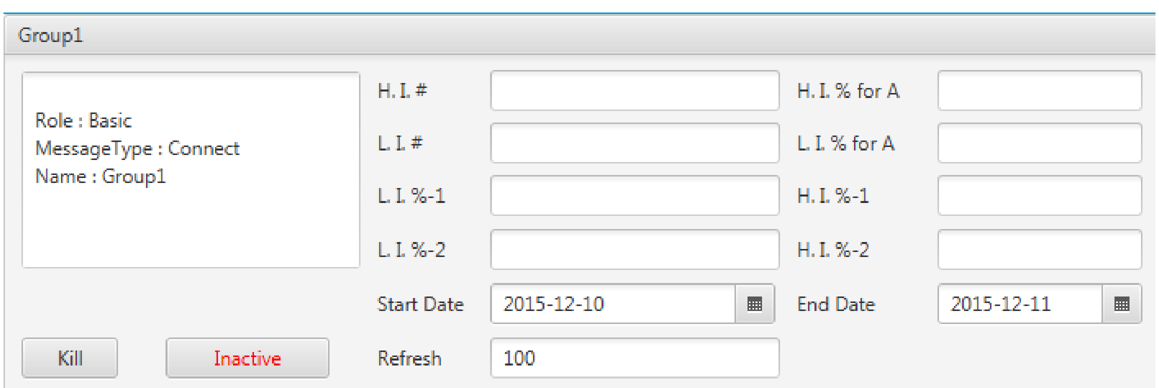

Figure 7. SocialNetwork GUI

The SocialNetwork GUI helps users to set up different parameters to simulate the social network model. With different setups, the information in the social network will finally converge into different values.

1. H. I. #: The number of profiles in the high influence group.
2. L. I. #: The number of profiles in the low influence group.

3. H. I. % for A: The percentage of people vote for A in the high influence group.
4. L. I. % for A: The percentage of people vote for A in low influence group.
5. L. I. %-1: The start point of the range of influence for the low influence group.
6. L. I. %-2: The end point of the range of influence for the low influence group.
7. H. I. %-1: The start point of the range of influence for the high influence group.
8. H. I. %-2: The end point of the range of influence for the high influence group.

• Conclusion:

In conclusion, I hope that I can further pursue this project to a next level with a more detailed modeling. After all, to mimic the real elements and people, we need to simulate the real communication between elements within a group. A group should be a tag (such as a social class of influence). Also, I suggest that including extra information to the person element that shows its capabilities of interacting with other elements will be a big progress in this model. This in my opinion should be the next step in order to evaluate the interaction in a more realistic manner.

• References:

[1] Professor SK Chang webpage lecture on the “Abstract Machine for SIS”.

[2] Professor SK Chang webpage lecture on the relationship between patterns and Index Cells.

[3] Christo Wilson , Bryce Boe , Alessandra Sala , Krishna P.N. Puttaswamy , Ben Y. Zhao, User interactions in social networks and their implications, Proceedings of the 4th ACM European conference on Computer systems, April 01-03, 2009, Nuremberg, Germany.