Towards a Trust and Reputation Framework for Social Web Platforms

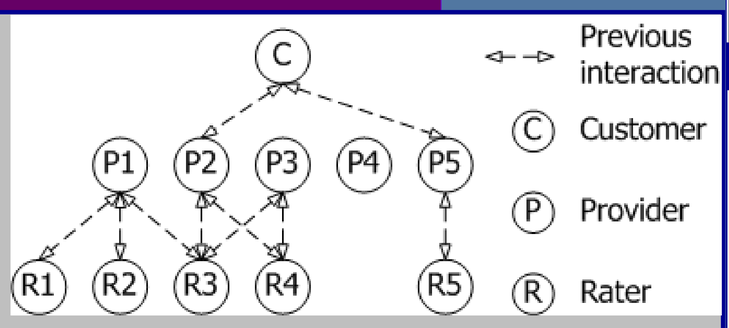
Thao Nguyen[1,2], Luigi Liquori[2], Bruno Martin[1], and Karl Hanks
[1] University Nice-Sophia Antipolis, France
[2] Institut National de Recherche en Informatique et Automatique, France
{Thao.Nguyen,Luigi.Liquori}@inria.fr,
Bruno.Martin@unice.fr, Karl.Hanks@cantab.net

Introduction

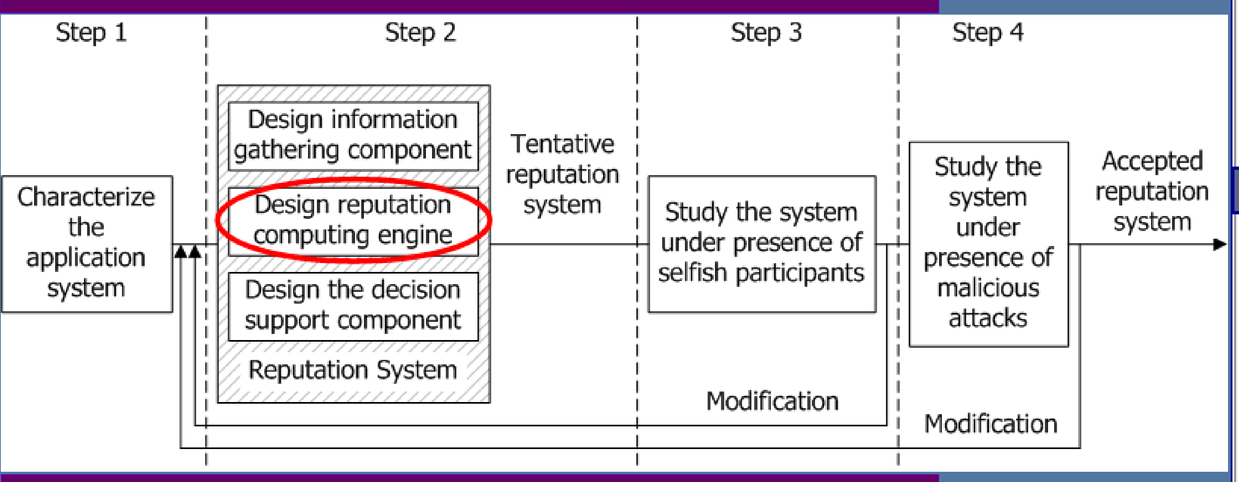
The information concerning the reputation of individuals, which used to be spread by word-of-mouth, now can be broadcast more easily and faster than ever before via the Internet. Exploiting this kind of information, Trust and Reputation Systems (TRSs) represent a significant trend in decision support for Internet-based interactions. They encourage honesty and cooperation among users, resulting in healthier online markets or communities, therefore a better safety for their users. However, TRSs themselves can be the target of attacks, and the major difficulty in designing a reputation system is making it robust against malicious attacks.

Example of "user roles" in a TRS

Consumer C is looking for a service/product which is provided by 5 candidates including SERVICE Providers P1, P2, P3, P4, and P5.
With support from a TRS, C ranks trustworthiness of these candidates and picks out one to trade with. To estimate trustworthiness of the providers, the TRS collects opinions/ratings from Raters R1, R2, R3, R4, and R5, who are previous consumers of the providers, aggregating the ratings with C's personal experience with the providers if it exists.



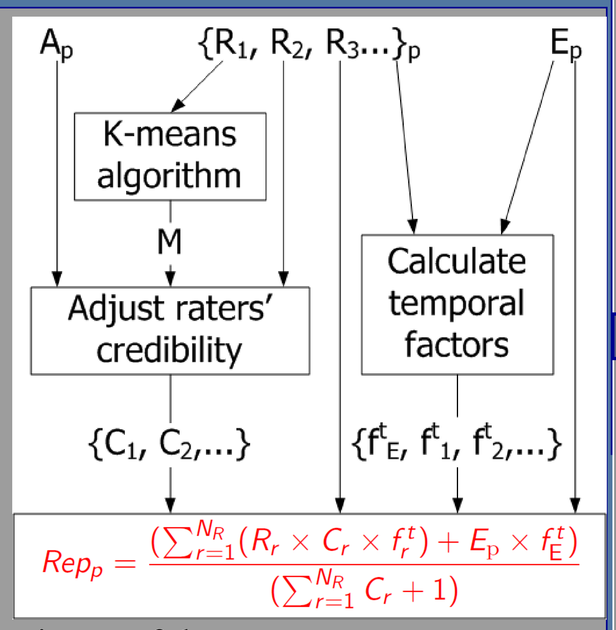
Designing Process



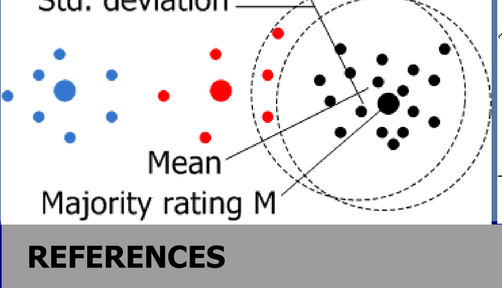
Reputation Computing Engine

Notational conventions

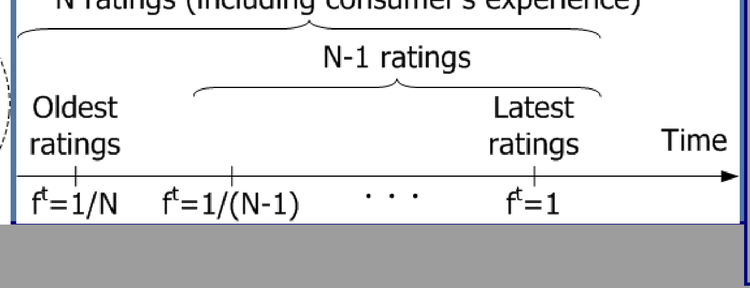
R_r : rating/feedback shared by rater r, including a time-stamp
$\{R_1, R_2, \ldots\}_p$: set of collected ratings
C_r : Credibility of rater r
E_p : first hand experience of the consumer with provider p
A_p : last computed reputation score of provider p if consumer has interacted with the provider
M : majority rating, which is an outcome of K-means clustering algorithm
f^t_r , f^t_E: temporal factors corresponding to rating by rater r and personal experience of the consumer
Rep_p : computed reputation score of the considered provider p



K-means algorithm



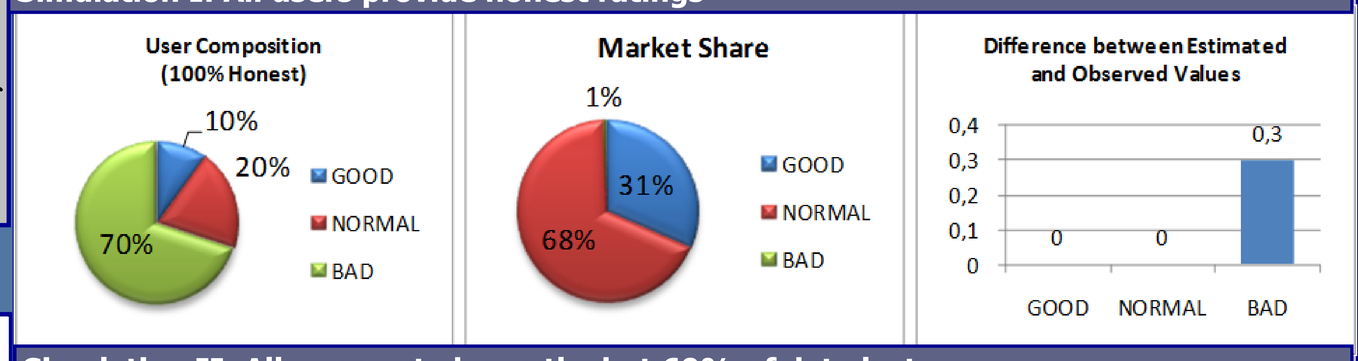
Calculate temporal factors



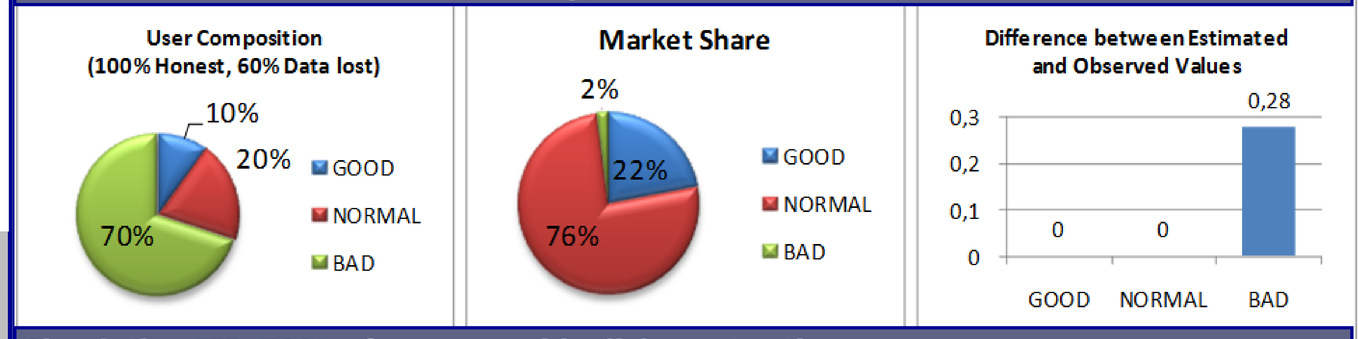
Simulation results

- 200 users in total, 10000 transactions (each transaction is requested by a random user; the rest of users are candidate providers)
- Each user can provide *GOOD*, *NORMAL*, *BAD*, or *GOODTURNBAD* service; and submit *HONEST*, *DISHONEST*, or *COLLUSIVE* (favoring users in the same group) ratings.

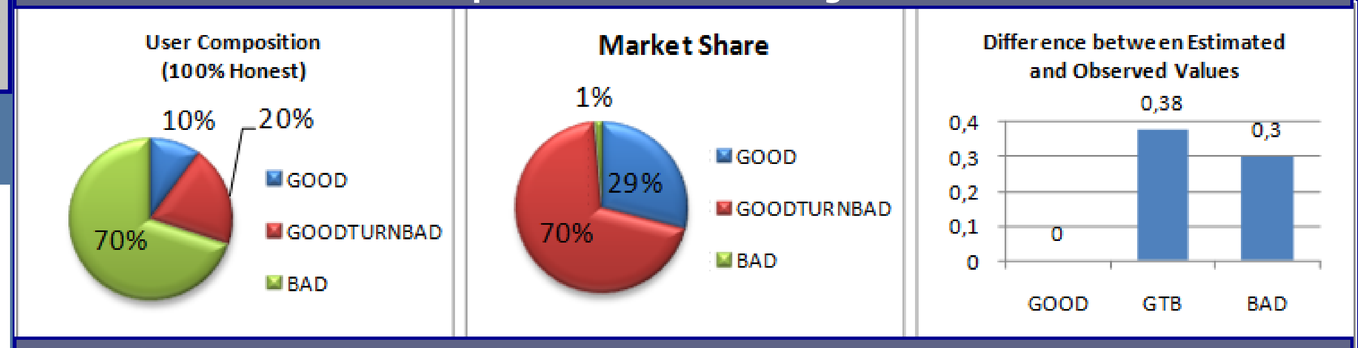
Simulation I: All users provide honest ratings



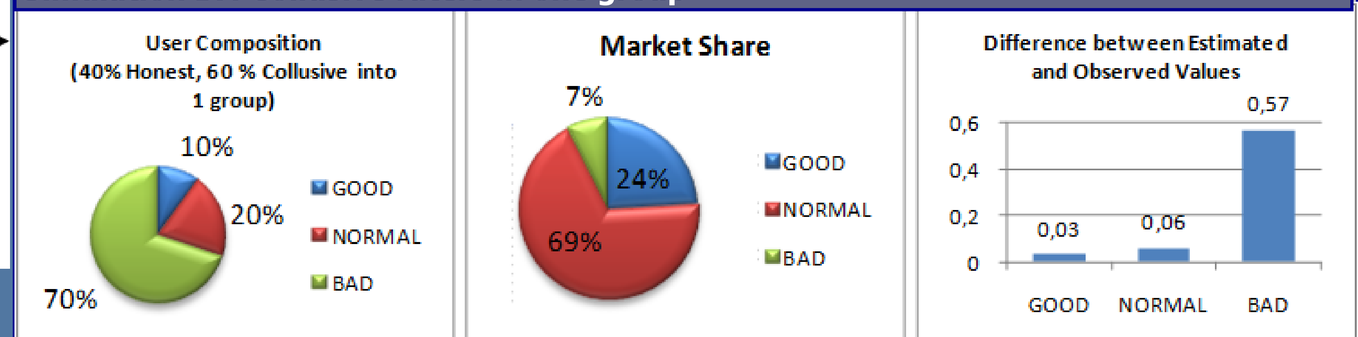
Simulation II: All users rate honestly, but 60% of data lost



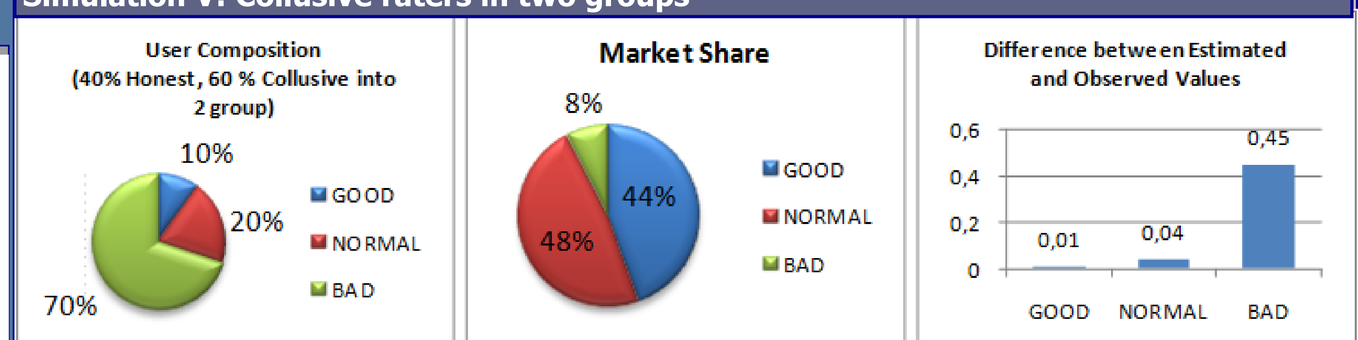
Simulation III: 70% of users provide dishonest ratings



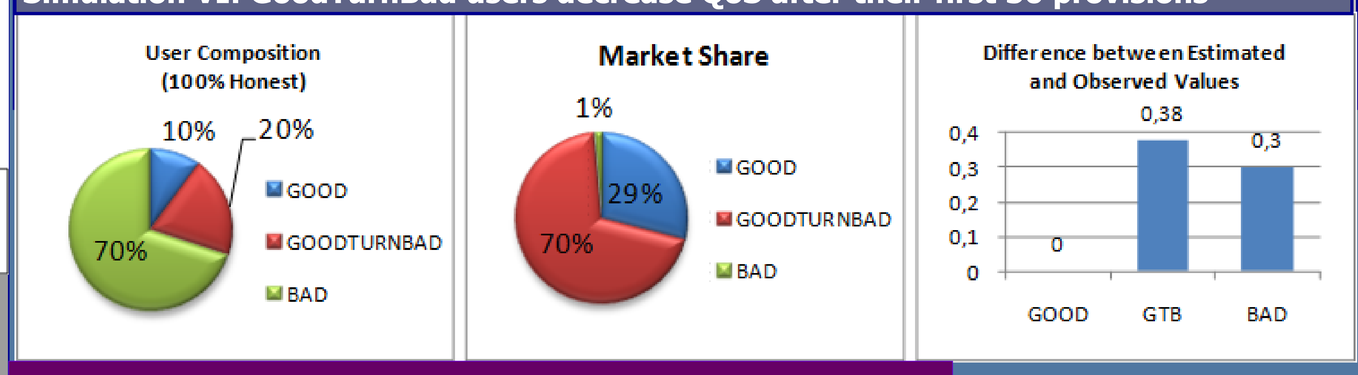
Simulation IV: Collusive raters in one group



Simulation V: Collusive raters in two groups



Simulation VI: GoodTurnBad users decrease QoS after their first 50 provisions



Conclusions and Future Work

We have proposed a research methodology which can be used to study the robustness of many TRSs, and also implemented a model which is a simplified and modified version of the RateWeb engine [2]. We found a flaw in the engine, making it vulnerable to *milking reputation* attack which is corresponding to **Simulation VI**. For that reason, we will design a new RCE and develop a simulator with measures applicable to social-web applications.

REFERENCES

1. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)
2. Malik, Z., Bouguettaya, A.: Rateweb: Reputation assessment for trust establishment among web services. The International Journal on Very Large Data Bases 18(4), 885–911 (2009)
3. Dellarocas, C.: Online reputation systems: How to design one that does what you need. MIT Sloan Management Review 51(3) (spring 2010)