Representation of Rules for Relevant Recommendations to Online Social Networks Users

Sarah Bouraga
Dept. of Business Administration,
PReCISE,
University of Namur,
Namur, Belgium
sarah.bouraga@unamur.be

Ivan Jureta
Dept. of Business Administration,
PReCISE,
University of Namur,
FNRS, Brussels
Namur, Belgium
ivan.jureta@unamur.be

Stéphane Faulkner
Dept. of Business Administration,
PReCISE,
University of Namur,
Namur, Belgium
stephane.faulkner@unamur.be

***Abstract*—In our prior work, we identified rules for use in recommendation algorithms on Online Social Network (OSN) in order to increase the relevance of content suggested to a user. The resulting recommendation algorithms filter out and prioritize event types for OSN users (such as photo posts by friends, status posts, shared content, etc.), and are thereby intended to reduce information overload.**

This paper proposes a representation of these rules in a requirements model of a OSN. This is interesting, because recommendation rules influence user behavior, which in turn influences future requirements. If there is a recommendation algorithm, then its behavior should be represented also in requirements models of the system. The paper makes two contributions. We define requirements that OSNs should satisfy in order to produce relevant recommendations of event types to users. We investigate whether an existing requirements modeling language (namely, i-star) can be used to model these requirements.

I. Introduction

Information overload usually refers to the difficulty to make decisions when there is too much information. In such situations, the decision-maker is confronted to so much information that it may be hard to cut through the noise and find the interesting subset, or even only one.

Online Social Networks (OSNs) are particularly prone to producing information overload. Users generate content on OSNs. This content is displayed in various ways to other users. The OSN then displays, either all these event types chronologically (for instance, Twitter and Tumblr notify the individual user with all the event types generated by the other users she follows); or some of the event types seemingly according to the user's induced preferences or profile (for instance, Facebook notifies the individual user with some of the event types generated by her friends).

In a prior work [1], we proposed rules that an OSN could follow in order to propose the most relevant content to the individual users. These rules were represented as recommendation trees, or recommendation algorithms. These recommendation algorithms filter out and prioritize event types for OSN users, and thereby take over some of the decision-making effort from the user.

In this paper, we aim to model these rules using a RE modeling language. More specifically, drawing on our prior work on the identification of rules for relevant recommendations on OSN, we now define and model the requirements that OSNs should satisfy, in order to be said to satisfy (or apply) these rules. We make two contributions: (i) we define requirements that OSNs should satisfy in order to produce relevant recommendations of event types to users; and (ii) we investigate whether an existing requirements modeling language (namely, *i-star*) can be used to model these requirements.

The rest of this paper is organized as follows. Section II reviews the related work. Section III summarizes the background (that is, our prior work on the recommendation algorithms for OSNs); briefly introduces *i-star* (*i**); and proposes the modeling of the recommendation algorithms using *i**. We discuss our results and conclude the paper in Section IV and Section V respectively.

II. Related Work

The design of recommendation systems (RS) is a topic of considerable work. Many studies propose a recommendation algorithm for the generation of personalized suggestions in many fields. The majority of the proposed recommendation algorithms can usually be classified in one of these three recommendation techniques: collaborative filtering (CF), content-based (CB), or hybrid approaches [2]. For instance, Yamamoto et al. [3] proposed recommendation algorithms for TV program suggestions, using a CB approach. Kim et al. [4] proposed a recommendation algorithm, using the *"K-Means Clustering method to reduce the search space"*. They tested their algorithm with the EachMovie dataset, proving that their proposal generated better suggestions than the classical CF approach. Choi et al. [5] proposed an algorithm that generates movie recommendations based on genre correlations. Other examples include Chen et al. [6] who proposed a *"CF recommendation algorithm Based on User Interest Change and Trust Evaluation"*; or Onuma et al. [7] who proposed "TANGENT", a "Surprise-Me" recommendation algorithm that generates *"related, but off the beaten path suggestions to users"*.

The topics of customized recommendation, and/or of the design of a RS for OSN have also been discussed in the

TABLE I
CORE, NEUTRAL, AND OPTIONAL EVENT TYPES

	Core	Neutral	Optional	Undetermined
Profile	Name, Birthday Relationship status Profile picture, Foreign languages, Sports, Youth movement, Music and movie	Website, Job(s) Cousins	Ethnicity, Phone number, Parents, Uncles and aunts Religious and Political beliefs, Résumé	Mother language Industy(ies), Siblings Qualifications Areas of expertise Favorite quote About me
Relationships	Unidirectional		Bidirectional	
Content	Short text, Comment on a media, Tag friends on a media, Share a photo, video, Receive a message Create, and Join a group	Comment on: profile info, a status; Tag friends on a status	Long text	Comment on a relationship status, Like on a status, on a media, Share a status, Tag media, Send message
Privacy				Semi public, Public
Recommendations			Users, Public figure	Content
Connection			Sign in	Share

literature. However, the proposed RS focus mainly on the recommendation of friends. Examples include Hsu et al. [8], who proposed LJMiner, a RS destined for LiveJournal. LJMiner is a hybrid system, combining the analysis of structural links, with the analysis of content. Guy et al. [9], [10] developed a RS to provide friend recommendation for an enterprise OSN. Xie [11] proposed a RS that generates friend recommendations, based on (i) user activity, (ii) interest analysis, and, optionally, (iii) domain knowledge. Also, Chen et al. [12] proposed a comparison of four recommendation algorithms that provide people suggestions on OSN, namely “Content Matching”, “Content-plus-Link”, “Friend-of-Friend”, and “SONAR”.

These contributions differ from our work here, because we are interested in a RS that will classify content as relevant or irrelevant for the individual users, that is, a RS that will recommend event types; and not a RS that will provide the user with friend recommendations.

The topic of item recommendations on an OSN has also been researched. Konstas et al. [13] proposed a collaborative RS that generates track recommendations for the OSN “last.fm”. Their RS base their suggestions on (i) the connection between users, (ii) the items, and (iii) the tags. Guy et al. [9], [10] proposed a RS that provides customized item recommendations to users of an enterprise OSN. Their recommendation technique is based on two elements, namely the people and the tags.

Finally, several authors have carried out studies about mobile phones notifications. Shirazi et al. [14] analyzed the kinds of mobile notifications that users like and dislike. In their study, a notification was considered as a piece of information users receive about a *“variety of events, such as the arrival of message, a new comment on one of their social network posts, or the availability of an application update”*. Similarly, Mashhadi et al. [15] explored the perceived importance of mobile phones notifications. Pielot et al. [16] studied how users deal with a notification, and discovered that users check the notification within a few minutes of arrival, *“regardless of whether the phone was in silent mode or not”*.

Our work here differs from theirs because we aim to propose event types recommendations based on rules, instead of, exclusively, the data induced by user activity. Also, we adopt a different approach to the term “notification”. The studies mentioned above focused on notifications mobile phone users receive to alert them of something new (such as a new message for instance). Here, what we call “notification” is the fact that the OSN proposes the event type to the user; and not only the “alert” that, for instance, Facebook users receive when one of their friends like a photo they posted. Finally, as far as we know, no research has been conducted about the modeling of a recommendation rules in requirements models, and more specifically, using *i-star*.

III. RECOMMENDATION ALGORITHMS MODELLING

A. Background and OSN Recommendation Trees

In a prior work, we surveyed 450 Bachelor students of the University of Namur regarding their perceived relevance of a list of event types; and we surveyed 150 students to understand factors, which influence this perceived relevance.

An event type is an activity generated by a user. An instance of an event type created by a user may produce a notification for her “friends” or connections on the OSN. An example of event type can be: “Sharing a photo”. An instance of this event type would be: “User A shares a photo X on her profile at time Y”. Another example of an event type would be: “Post a comment on a status”. An instance of the latter would be: “User A posts a comment on the status of user B at time Q”.

The question is then: *“Should we notify user B of the event type generated by user A?”*

Tables I, II and III summarize our findings.

Table I classifies the event types as core, neutral, and non-core by students; as well as undetermined (that is, the event types that did not yield significant results).

We also investigated the students regarding the categories of event types. The results showed that students want to be notified when their friends generate an event type belonging to the Profile, Link, or Content categories; whereas, they do not perceive as relevant event types belonging to the Recommendation, Privacy, or Connection categories.

TABLE II
CORE, OPTIONAL, AND DISPENSABLE CONTENT CATEGORIES

Categories	Core	Neutral	Optional
Profile	X		
Link	X		
Content	X		
Recommendations			X
Privacy			X
Connection			X

As far as the factors influencing the perceived relevance of event types; only 4 yielded significant positive results, namely:

- the reception of an alert: *"Has the OSN sent an actual alert to the user about the event type?"*,
- the commonalities: *"Is the event about something the user and her friend have in common?"*,
- the closeness of the friend who generates the event type: *"Are the user and her friend close, or are they merely acquaintances?"*,
- the quality of the friends involved, or tagged in the event type: *"Is (Are) the friend(s) who is (are) tagged in the event type close with the user or are they merely acquaintances?"*.

TABLE III
CLASSIFICATION OF FACTORS AS RELEVANT/IRRELEVANT, AND INFLUENCEABLE AND NON INFLUENCEABLE: SUMMARY

Factors	Relevant	Irrelevant	Undetermined
Influenceable	Alerts	Icons, Order of presentation, Starred content, Show friends	Preview
Non Influenceable	Commonalities, Closeness of friend, Quality of friends	Gender of contact, Experience of contact, Frequency of publication, Popularity of , friend, Number of friends, Number of likes, Legend, Tagged people, Location, Emoticons	

The other factors were considered as irrelevant in the explanation of the perceived importance of event types; or they did not provide significant results (Preview).

Based on these results, we proposed two recommendation algorithms. The first one, represented in Figure 1, can be applied for a user who logs into the OSN rarely, and/or a user who is very popular (that is, a user who has many friends on the OSN). In other words, the first algorithm is to be applied for users who will have potentially many event types to be proposed when she logs in. In applying the first set of rules, the RS would propose to the user only the core content. However, if an individual uses the OSN everyday and/or if he has few friends on the OSN, and if we only follow the first decision tree, then we might risk not having enough content to propose. Hence, we can turn to the second set of rules, represented by the second recommendation algorithm in Figure 2, and propose the neutral event types under several conditions, that is if the factors are verified. In Figure 2, "N" stands for "Notify", and "NN" stands for "Do not notify".

B. I-Star

The motivation behind the *i** framework is the modeling and reasoning about organizations: their environments, as well as their ISs. Two modeling elements compose the framework, namely the Strategic Dependency (SD) model and the Strategic Rationale (SR) model. They are used, respectively, for the description of the dependency relationships between various actors; and for the description of stakeholder interests and concerns, and how they might be dealt with by different configurations of systems and their environments [17], [18].

The Strategic Dependency (SD) model focuses on the intentional relationships between actors, which allows a deeper understanding of the whys. Various dependency types exist, differentiating between the types of freedom and constraint [17], [18].

The Strategic Rationale (SR) model provides a way to model the intentional constructs within each actor. The intentional elements (goals, tasks, resources, and softgoals) in the SR model are linked by (i) means-ends, and (ii) task-decomposition relationships. The former explain *"why an actor engage in some tasks, pursue a goal, need a resource, or want a softgoal"*; while the latter describe hierarchically the intentional elements making up a routine [17], [18].

C. OSN Recommendation Rules Modeling

We will first consider the SD model of the OSN. Several actors are to be taken into account. The OSN and the User are *Positions*; the Generator and the Receiver are *Roles*; and User X and User Y are *Agents*.

The *Position* of User covers the *roles* of Generator and Receiver of event types. *User A* and *User B* are instantiations of both roles, i.e., User A can be a Generator, as well as a Receiver. The same is true for User B.

We state that:

- The OSN depends on:
  - The Generator to generate an Event Type (ET),

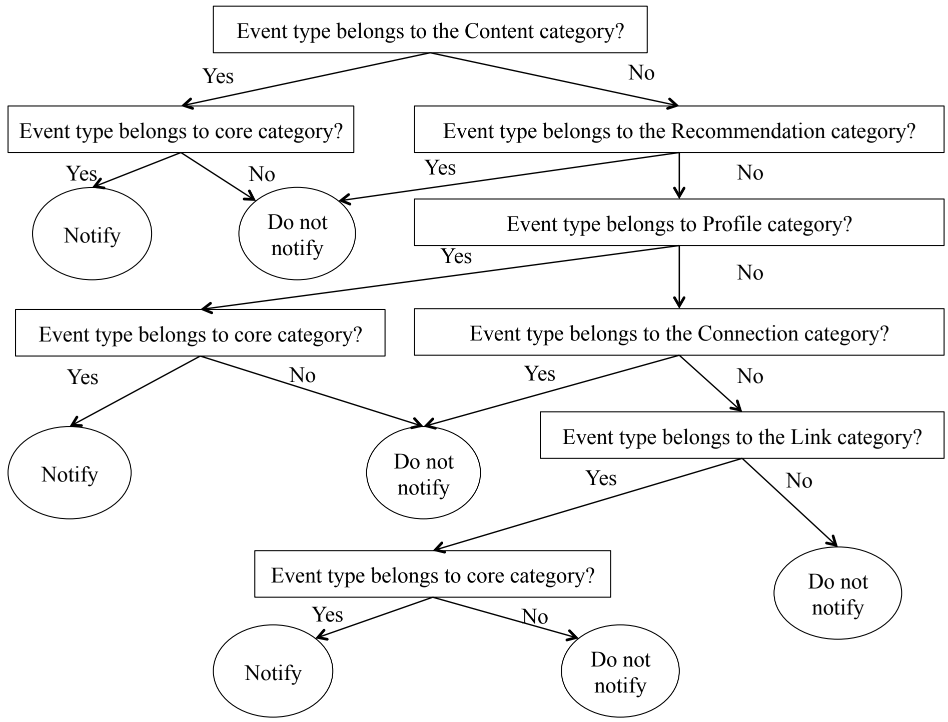

Fig. 1. Decision Tree Based on Event Types

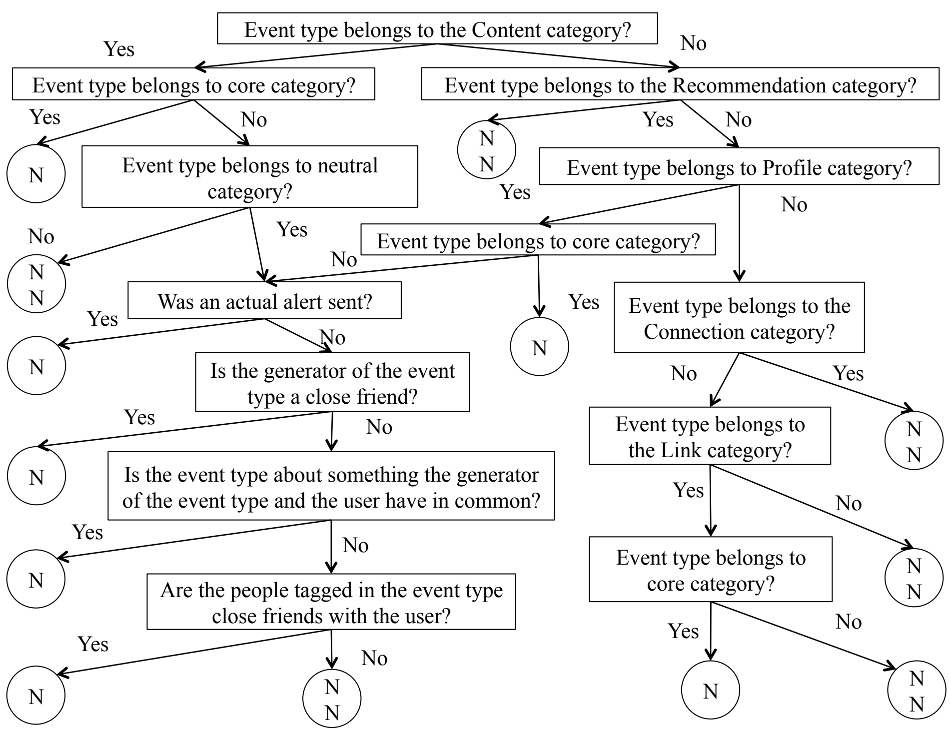

Fig. 2. Decision Tree Based on Event Types

- The Receiver to decide to reply to the proposed ET or not,
  - The User for a better user involvement on the OSN, that is, use the OSN on a regular basis,
  - The User to use the OSN, that is, be(come) a member, share/post ETs, look at proposed ETs, browse the OSN,
- The Receiver depends on the OSN to propose relevant ETs,
- The User depends on the OSN to mitigate information overload.

These early requirements are represented in Figure 4.

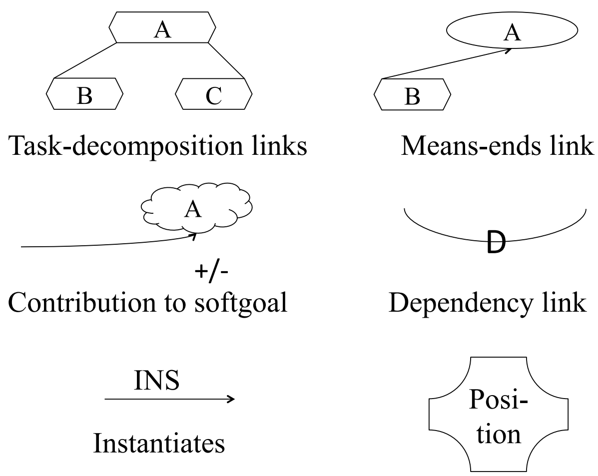

Fig. 3. Legend for Figures 4 and 5

We will now consider the SR model.

The *Agent* User A is an instance of the *Role* Generator, and an instance of the *Role* Receiver. The *Agent* User B is an instance of the *Role* Generator, and an instance of the *Role* Receiver.

We will detail the intentional constructs of the OSN:

- The OSN has the internal task of creating and maintaining the desire/interest for the user to use the OSN; decomposed into:
  - The goal "Propose relevant recommendations",
  - The goal "Allow users to generate ETs/Propose features",
- The internal goal "Allow users to generate ETs/Propose features" can be achieved by:
  - Allowing users to post profile ETs,
  - Allowing users to post link ETs,
  - Allowing users to post content ETs,
  - Allowing users to post recommendations ETs,
  - Allowing users to post privacy ETs,
  - Allowing users to post connection ETs,
- The internal goal "Propose relevant recommendations" can be achieved by:
  - The task "Gather ET", decomposed into:
    - The task "Gather original/starting ET",
      - Depends on the Generator for the resource "Original ET",
    - The task "Gather reply to an original/starting ET",
      - Depends on the Generator for the resource "Reply",
  - Implement RS, decomposed into:
    - The task "Gather user information" (about frequency of use, and popularity),
    - The task "Apply decision tree", decomposed into:
      - The task "Decide to notify the user with the ET",
      - The task "Decide to not notify the user with the ET",

The internal goal *"Allow users to generate ETs/Propose features"* contributes positively to the softgoal *"Better user involvement"*. The more features the users have at disposal, the more fun they will have with the OSN, and the more involved they will get. The internal goal *"Allow users to generate ETs/Propose features"* contributes negatively to the softgoal *"Mitigate information overload"*. The more features the users have at disposal, the more ETs they will generate, and the more information/content there will be.

The internal goal *"Propose relevant recommendations"* contributes positively to the softgoal *"Mitigate information overload"*. The RS will filter out irrelevant ETs, and will notify the user only with the most relevant ones. The internal goal *"Propose relevant recommendations"* contributes negatively to the softgoal *"Better user involvement"*. The amount of ETs that can be notified to the user will be reduced, the user will then have less ETs to browse.

We will now turn to the intentional constructs of the other actors:

- The Generator has the internal task of "Generating ET", decomposed into:
  - The task "Generate original ET",
  - The task "Generate reply",
- The Generator depends on the OSN for the resource "Feature",
- The Receiver has the internal task of "Use the OSN passively", decomposed into:
  - The task "Observe proposed ET",
    - Depends on the OSN for the resource "Notification",
  - The task "Decide if she will reply to the proposed ET", decomposed into:
    - The task "Decide to reply, to send notification",
    - The task "Decide to not reply",
- If the Receiver decides to reply to an event type, then the Receiver depends on the Generator, that is, on the other role, to act on this decision and to generate the reply.

These requirements are modeled in Figure 5.

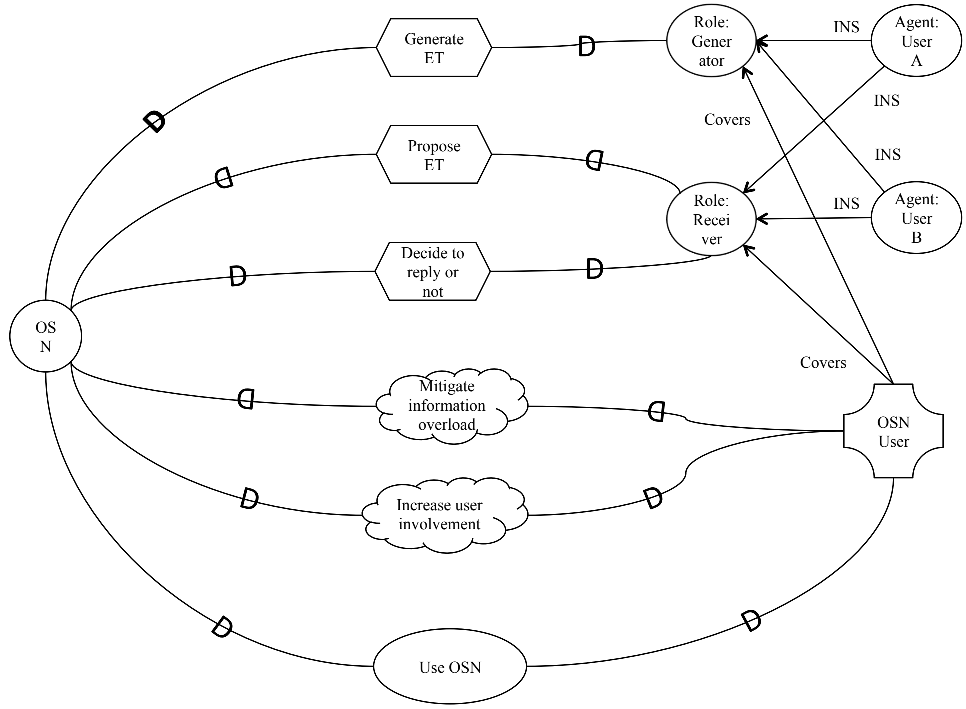

Fig. 4. Strategic Dependency Model

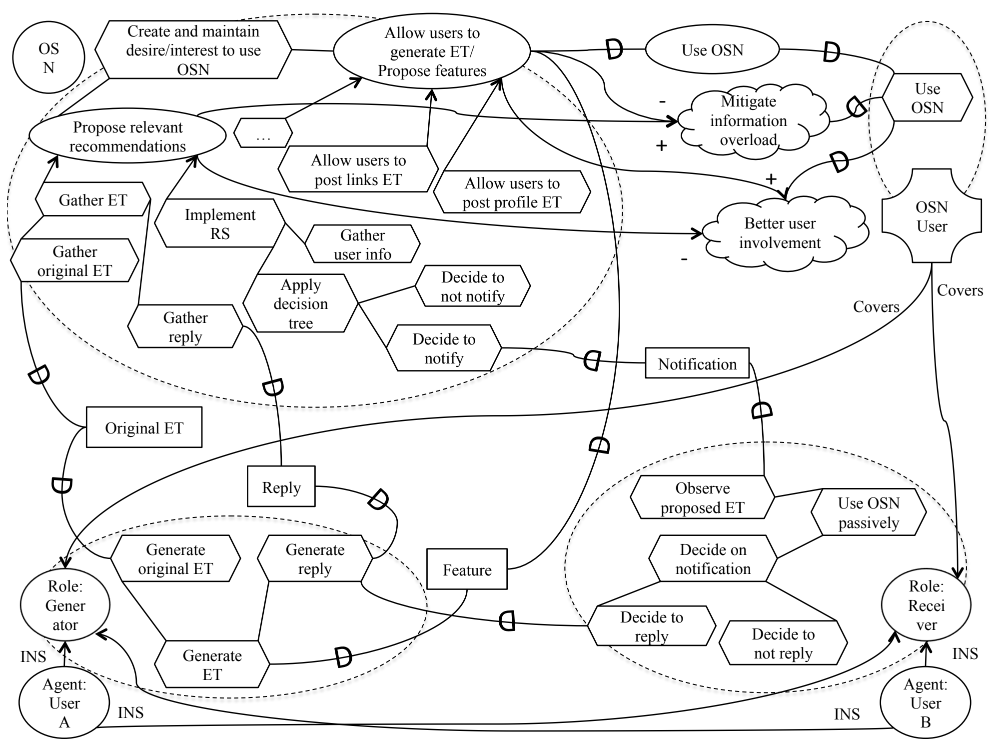

Fig. 5. Strategic Rationale Model

IV. Discussion and Limitations

In Section III, we attempted to model our recommendation algorithm in a requirements model of an OSN, using the notations provided by an existing RE language, namely *i**. We believe this is relevant because recommendation rules influence user behavior, which in turn influences future requirements. If there is a recommendation algorithm, then its behavior should be represented also in requirements models of the system.

We tried to translate the notion of cycle that we can observe in OSN use. More specifically, lets consider two users, User A and User B. User A shares a photo. The recommendation algorithm is applied to decide if the sharing of the photo generated by User A is an event type that should be proposed to User B. If the application of the recommendation algorithm results in the decision to propose the event type, then the User B can react to the event type by, for instance, commenting on the photo. This comment on the photo is again an event type. The recommendation algorithm will also be applied in order to decide if the event type should be notified to other users. Hence, we can observe, and with our *i** models we tried to translate, this notion of cycle; represented in Figure 6.

In summary, we know which event types are perceived as important by OSN users. We believe that it is relevant to have insight into the event types that users want to see in priority, given that the design of new systems in general, and OSNs in particular, involves deciding what to show to users. Knowing which event types are more important to which target user group should help inform such decisions, so that we would try to show the most important contents to user when they connect to the OSN; and depending on the time they spend on the OSN, leave less important content for later in the period of time the user is logged in.

The rules we designed are based on what the users want to see when they log in. Once a user sees an event type she finds interesting, she then has to decide whether or not she wants to reply to the event type. We are interested in modeling this interaction, this loop (modeled in Figure 6); in which the rules play a significant role.

We believe that our models in Figures 4 and 5 are correct, in the sense that we translated the two roles that each user can have: the generator of an event type, and the receiver of the event type. And we modeled this information, using the notions offered by *i**, that is, tasks, goals, softgoals, as well as dependency links, and means-ends and task decomposition relationships. The notion of loop is modeled through the use of the notions “Position”, “Role”, and “Agent”. We consider two Agents, User A and User B. Each user can take on both Roles; the Position User can cover both roles.

However, our models suffer from two major limitations. Firstly, this notion of cycle mentioned above could be modeled even more clearly. Our models show that any user can take on both roles, but we miss the clear steps of the loop of (i) observing event types, (ii) deciding if to send notification, and (iii) sending notification. The clear notion of timing is missing.

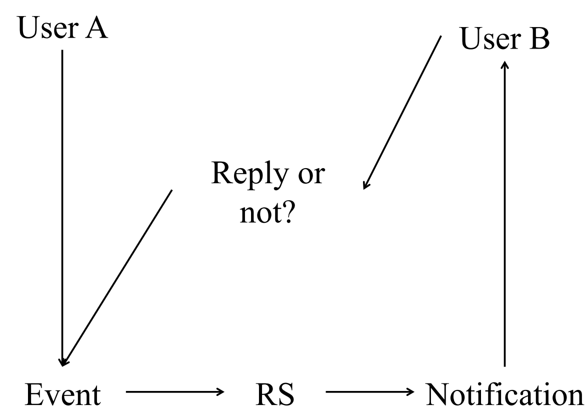

Fig. 6. Cycle of OSN Usage

Secondly, in the SR model, we do not take into account, for now, the activity, the history of the user; that is, the implicit data gathered by the OSN about users. While we know that this information could be valuable for the generation of event types recommendation; as it has been proven by the common recommendation techniques mentioned in Section II (for instance, CF is an efficient recommendation technique that bases its recommendation on the preferences of the user).

Finally, as mentioned in Section III-A, we distributed the surveys only to Bachelor students of the University of Namur. We discovered that the latter almost exclusively use Facebook. Hence, the profiles of all the respondents are extremely similar. Thus, the results are valid for this specific OSN, and for this specific profile.

V. Conclusion

In this paper, building on our prior work, we modeled recommendation rules using a RE modeling language, namely *i**. More specifically, drawing on our prior work on the identification of rules for relevant recommendations on OSN, we made two contributions: (i) we defined requirements that OSNs should satisfy in order to produce relevant recommendations of event types to users; and (ii) we investigated whether an existing requirements modeling language (namely, *i**) can be used to model these requirements. We believe our contribution can have implications for the design (and more specifically, for the RE phase) of future OSNs that would aim to propose more relevant event types to the individual users.

In order to address these questions, we used *i** to model the recommendation algorithms we designed in a prior work. In other words, we modeled the requirements (induced by the recommendation rules) for the design of an OSN that aims to mitigate information overload; that is, for the design of an OSN that aims to propose the most relevant event types to the individual users. The resulting models were proposed in Figures 4 and 5.

We also discussed our results, and the limitations of our work.

Future work will consist in analyzing a broader range of profile, in order to design a more complete set of rules and thus, a more complete decision tree; instead of having a decision tree based solely on the specific profile of students. We will also work on the validation of the proposed (and future) classification tree(s).

References

[1] S. Bouraga, I. Jureta, and S. Faukner, "Rules for relevant recommendations to online social network users," (in review).

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[3] N. Yamamoto, M. Saito, M. Miyazaki, and H. Koike, "Recommendation algorithm focused on individual viewpoints," in *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*. IEEE, 2005, pp. 65–70.

[4] T.-H. Kim, Y.-S. Ryu, S.-I. Park, and S.-B. Yang, "An improved recommendation algorithm in collaborative filtering," in *E-Commerce and Web Technologies*. Springer, 2002, pp. 254–261.

[5] S.-M. Choi, S.-K. Ko, and Y.-S. Han, "A movie recommendation algorithm based on genre correlations," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8079–8085, 2012.

[6] Z. Chen, Y. Jiang, and Y. Zhao, "A collaborative filtering recommendation algorithm based on user interest change and trust evaluation." *JDCTA*, vol. 4, no. 9, pp. 106–113, 2010.

[7] K. Onuma, H. Tong, and C. Faloutsos, "Tangent: a novel,'surprise me', recommendation algorithm," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 657–666.

[8] W. H. Hsu, A. L. King, M. S. Paradesi, T. Pydimarri, and T. Weninger, "Collaborative and structural recommendation of friends using weblog-based social network analysis." in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, 2006, pp. 55–60.

[9] I. Guy, I. Ronen, and E. Wilcox, "Do you know?: recommending people to invite into your social network," in *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 2009, pp. 77–86.

[10] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, "Social media recommendation based on people and tags," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 194–201.

[11] X. Xie, "Potential friend recommendation in online social network," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*. IEEE, 2010, pp. 831–835.

[12] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 201–210.

[13] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 195–202.

[14] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, "Large-scale assessment of mobile notifications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 3055–3064.

[15] A. Mashhadi, A. Mathur, and F. Kawsar, "The myth of subtle notifications," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 2014, pp. 111–114.

[16] M. Pielot, K. Church, and R. de Oliveira, "An in-situ study of mobile phone notifications," in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, 2014, pp. 233–242.

[17] E. S. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. IEEE, 1997, pp. 226–235.

[18] E. Yu, "Modelling strategic relationships for process reengineering," *Social Modeling for Requirements Engineering*, vol. 11, p. 2011, 2011.