Supporting Learners-to-Lerners Interactions Basing on Online Social Networks Information

Pasquale De Meo and Fabrizio Messina and Domenico Rosaci and Giuseppe M. L. Sarné

***Abstract*—E-Learning students can benefit from proper class formation process based on the student needs. In particular, Online Social Networks make available data concerning users' interactions, as skills and trust relationships, that are behind the dynamics of thematic social network groups, and can be exploited to form e-Learning classes. To this aim, we propose a model based on such information, which are properly combined to support the dynamics of e-Learning classes on Online Social Networks. The approach provide a way to give suggestions to users about the best classes to join with and to class adminastrors the best students to accept. The proposed approach has been tested by simulating an e-Learning scenario within a large social network by showing its capability to satisfy all the actors.**

***Index Terms*—Social Networks; Software Agents; Thematic Groups**

I. Introduction

E-Learning (EL) represents a good solution for courses, as it provides time and location flexibility, low costs and information sharing [1]. In this context, among the factors affecting learners progresses there are personal attitudes, initial skills and the level of *mutual trust*, which influences the attitudes of peers to start interactions [2] and minimizes the cold start effect. Given that those information are widely available in Online Social Networks (OSNs), EL activities can benefit from synergies with OSNs. Besides, many OSN platforms [3], [4] support thematic *groups* that, for their relevance, have been largely investigated [5]–[9].

In addition, software agents can support EL class formation processes by suggesting to students (classes) about the best classes (students) to join with (to accept) [10]–[12]. Studies confirmed that, within social communities, users start to interact and share information with other peers also based on the level of mutual trust existing among users [13]–[18]. Besides, also in forming OSN groups existing trust relationships can give a significant contribution, in addition to a similarity criterion [13], [19], [20].

It is obvious that, due to the huge amount of data of OSNs and the huge number of thematics groups, examining the entire space of data to suggest suitable solutions for learners' needs is impracticable. Therefore, based on previous research experiences [15], [21]–[26] we designed a model to manage formation and evolution of e-Learning classes by using user information available on OSNs. These information are linearly combined in a measure, named *convenience*, used to suggest the best class (student) to join with or leave (to accept or remove) to a user (to the class itself). First of all, the skills of a student with respect to a set of topics of interest represent the basic aspect we considered to give teaching-homogeneity to the class [27], in order to balance "supply" and "offer" of support requests (i.e. interactions). Trust represents the second component, which is computed by combining several specific factors – which are related to specific e-Learning concerns – giving a complete trust model based on reliability and reputation criteria and on some countermeasures for erroneous or malicious opinions. The model is designed to assists students and classes by means of personal software agents delegated to create, manage and update the *profiles* of their owners on the basis of information found on the OSNs. The *convenience* measure is exploited by a distributed procedure, named Class Formation (CF), that allows learner/class software agents to appropriately cooperate to form classes.

The experimental trials have shown that running the CF algorithm allows students and class administrators to improve the average value of the convenience within classes.

The rest of the paper is organized as follows: Section II introduces the context and the Expertize, Trust and Advantage measures. The proposed architecture is described in Section III, while Section IV discusses the GF algorithm. Section V presents the experiments we carried out, Section VI examines related literature and, finally, in Section VII we draw our conclusions.

II. e-Learning interactions and measures

Let be $\mathcal{N}$ the set of OSN members, ($||\mathcal{N}|| = N$), $\mathcal{C}$ the set of classes ($||\mathcal{C}|| = C$), with each class $c \in \mathcal{C}$ consisting of a number of learners and at least a teacher. We also suppose that each user $u_i \in \mathcal{N}$ is associated with a software agent [28] a_i able to obtain a view on the u_i background and attitudes and to assist him/her in joining with or leaving classes. Similarly, each manager is assisted by a software agent, denoted as A_i, in deciding whether a new member can be accepted in the class.

We also define a *behavioral measure*, which is related to the interactions carried out by a learner (see Section II-A) and a *trust measure*, which considers the level of mutual trust among OSN members (see Section II-B).

Pasquale De Meo is with the Dept. DICAM, University of Messina, Viale Andrea Doria, 6 - 01010 Messina, Italy, e-mail: p.demeo@dmi.unime.it

Fabrizio Messina is with the Dept. DMI, University of Catania, Viale Andrea Doria, 6 - 01010 Catania, Italy, e-mail: messina@dmi.unict.it

Domenico Rosaci is with the Dept. DIIES, Unversity Mediterranea of Reggio Calabria, Loc. Feo di Vito - 89123 Reggio Calabria, Italy, e-mail: domenico.rosaci@unirc.it

Giuseppe M. L. Sarné is with the Dept. DICEAM, Unversity Mediterranea of Reggio Calabria, Loc. Feo di Vito - 89123 Reggio Calabria, Italy, e-mail: sarne@unirc.it

A. Behavioral Measures

The principle behind the definition of the behavioral measures is that, in order to form classes, a balance between *required* and/or *offered* skills should be desirable, as each learner is interested to improve his/her knowledge by joining with classes where the other members have suitable capabilities and managers of classes are interested to include users holding skills and attitude to interact.

Classes. Let's define a class c as a tuple $\langle S, W, V_c, o\rangle$ where: (*i*) $S = \{s_1, s_2, \dots, s_m\}$ is the skill set required by the class manager of c; (*ii*) $W = \{w_1, w_2, \dots, w_m\}$ is the weight set used to evaluate the students' skills; (*iii*) V_c is the minimum overall skill grade, computed over the specific skill set S, required to join with c; (*iv*) o is the reference topic or subject or goal of c. More formally, for an OSN user u_k and a skill set S, $V(k,S) = \sum_{i=1}^{m} w_i \cdot g(k, s_i)$, where $g(k, s_i) \in [0,1] \in \mathbb{R}$ is the knowledge grade of u_k for the skill s_i, while $w_i \in [0,1] \in \mathbb{R}$ is set by the class manager to weight g_i with $\sum_{i=1}^{m} w_i = 1$.

The User attitude H**.** The attitude of the user u_k to require and/or offer interactions for his/her skills is computed as:

$$H(k) = \alpha \cdot H(k) + (1-\alpha) \cdot \overline{H}(k)$$

where the new value of $H \in [0,1] \in \mathbb{R}$ combines, weighted by a system parameter $\alpha \in [0,1] \in \mathbb{R}$, the previous value and a contribution $\overline{H}$ for the new interactions computed as:

$$\overline{H}(k) = 1 - \frac{|h(k)_{req} - h(k)_{off}|}{h(k)_{rec} + h(k)_{off}} \quad \text{if} \quad h(k)_{req} + h(k)_{off} \neq 0$$

or $\overline{H}(k) = 0.5$ otherwise, and where $h(k)_{req}$ and $h(k)_{off}$, with respect to u_k, respectively are the evaluation of the interactions for a number N_{req} and N_{off} of skills subset $S_i \subset S$ requested and offered at the new step obtained by:

$$h(k)_{req} = \frac{1}{N_{req}} \sum_{i=1}^{N_{req}} g(k, S_{req,i})$$
$$h(k)_{off} = \frac{1}{N_{off}} \sum_{i=1}^{N_{off}} g(k, S_{off,i})$$

Therefore, when $h(k)_{req} \approx h(k)_{off}$, then $\overline{H} \approx 1$, i.e. the user u_k asks and provides interactions to the same extent. Vice versa, his/her attitude is mainly to offer (or require) interactions, i.e. $\overline{H} \approx 0$.

Class Behavior. The class behavior for the class c_j, denoted as $B(j) \in [0,1]$ characterizes its tendency to offer or require interactions and it is defined as $B(j) = \frac{1}{||c_j||} \sum_{k=1}^{||c_j||} H(k)$.

B. Trust Measure

The second measure is based on the concept of *trust* [29] and it is computed by combining two factors, namely reliability and reputation. The former measure derives by the direct knowledge between truster and trustee due to their *interactions* occurred in the past, while reputation is an indirect knowledge derived by the past interactions occurred among the trustee with other counterparts different from the current truster [30].

An interaction between two generic OSN learners u_p and u_r consists of a process where u_p starts with one or more learning tasks with u_r. Consequently, their software agents a_p and a_r observe the interactions of their owners to register the interaction features (type, topic, duration) and collect feedbacks about other OSN users to compute their respective reputations. Such feedbacks refer to the quality of these interactions (remember that users' skills are evaluated by the *Behavioral measures*).

Let $\eta_{p,r}$ and $\rho_{p,r}$ be respectively the measures of reliability and reputation that the OSN user u_p (i.e. agent a_p) computes for the OSN user u_r (i.e. agent a_r). The trust measure $\tau_{p,r}$ is obtained by combining the reliability ($\eta_{p,r}$) and the reputation ($\rho_{p,r}$) weighted by means a coefficient $\beta_{p,r} \in [0,1] \in \mathbb{R}$:

$$\tau_{p,r} = \begin{cases} 0.5 & \text{if } I_{p,r} = 0 \\ \beta_{p,r} \cdot \eta_{p,r} + (1-\beta_{p,r}) \cdot \rho_{p,r} & \text{if } I_{p,r} > 0 \end{cases}$$

where $I_{p,r}$ is the number of interactions occurred between the two actors. Note that for new learners the initial trust/reputation is set to 0.5 to contrast whitewashing strategies [31]. For computing $\beta_{p,r}$, we consider that its value increases with the number of interactions occurred between the two learners because their direct knowledge improves over time and decreases when the reliability in providing recommendations decreases (the reputation or some peers may be affected by malicious behaviors); Therefore, the coefficient $\beta_{p,r}$ is computed as:

$$\beta_{p,r} = Max(\beta_1, \beta_2)$$

where $\beta_1 = min\left(\frac{I_{p,r}}{I_{max}}, 1\right)$ and $\beta_2 = 1 - \Omega_{p,r}^{(t)}$. The parameter $\Omega_{p,r}^{(t)}$ is the average *confidence* at time t for the current set of recommenders that provided at least a recommendation to a_p about a_r computed as $\Omega_{p,r}^{(t)} = \frac{1}{||R_{p,r}||} \sum_{i=1}^{R_{p,r}} |\sigma_{p,r}^{(t-1)} - \tau_{q,r}|$ and where $R_{p,r}$ is the set of agents provided an opinion about a_r. It minimizes the effect of untrustworthy opinions by giving more relevance to those mentors evaluated by a_p as the most similar to it. $I_{p,r}$ is the number of interaction that is incremented at each step and when it is greater than the threshold I_{max} then the "knowledge" between two users is considered maximum. As a result, the contribute of the reputation in computing trust decreases as much as the number of the interactions occurred between the two involved learners constantly increases.

1) Computation of Reliability: The reliability measure, $\eta_{p,r} \in [0,1] \in \mathbb{R}$, is computed by u_p (i.e. a_p) about u_r (i.e. a_r) as $\eta_{p,r} = \vartheta_{p,r} \cdot \sigma_{p,r} + (1-\vartheta_{p,r}) \cdot \eta_{p,r}$, where the parameter $\vartheta_{p,r}$ weights in a complementary way the feedback parameter $\sigma_{p,r} \in [0,1] \in \mathbb{R}$ computed on the last interaction occurred between u_p and u_r at time-step t and the value of $\eta_{p,r}$ computed at time-step $(t-1)$.

The parameter $\vartheta_{p,r}$ considers the *relevance* assigned to the interaction between u_p and u_r, let it be $\Psi_{p,r}$. In principle, malicious behaviors aimed to gain good reputation with low value interactions ($\Psi \ll 0.5$) but high reliability ($\sigma \gg 0.5$) can start on interactions of high relevance ($\Psi \sim 1$) (due to a good reputation) but providing poor performance ($\sigma \sim 0$). Therefore, the closer the ratio Ψ/σ to 1, the higher the value of ϑ; the farther the value Ψ/σ from 1, the lower the value of ϑ. A possible choice for ϑ is represented by the adoption of the Gaussian centered in 1, as $\vartheta = e^{-(\Psi/\sigma-1)^2/v^2}$. ϑ acts as a "filter" for those values of σ which, for the correspondent values of Ψ, may reflect a malicious behavior, while large values of v will select only those values of σ for which σ/ϑ

is close to 0 by ensuring that almost the whole history of feedbacks σ is considered in computing η.

2) Computation of Reputation: The reputation measure $\rho_{p,r} \in [0,1] \in \mathbb{R}$ is computed by u_p (i.e., a_p) with respect to u_r (i.e. a_r) as a value ranging in $[0,1] \in \mathbb{R}$:

$$\rho_{p,r} = \frac{1}{||R_{p,r}||} \sum_{q=1}^{||R_{p,r}||} \tau_{q,r}$$

Through the usual meaning of these indexes, $0/1$ means that u_r is totally unreliable/reliable.

C. Convenience Measure

Behavioral and trust measures are combined to measure the *convenience*, for a user, to join with the class c_j. The asymmetric nature of the trust measure implies also the asymmetry of the convenience. In particular, let ϕ be a parameter computed as $\phi = \frac{(1-|H(k)-B(j)|)}{||c_j||}$, where $||c||$ is the number of users (i.e. agents) affiliated with c. Then the convenience ($\gamma_{u,c}$) for the user u to join with the class c, and that ($\eta_{c,u}$) of the class c to accept the affiliation request of a user u are computed as:

$$\gamma_{k,j} = \phi \sum_{a_i \in c_j} \tau_{k,i} \qquad \eta_{j,k} = \phi \sum_{a_i \in c_j} \tau_{i,k}$$

Both measures increase with the difference between the behaviors of a_k and c_j. As a consequence of the asymmetric nature of trust, the procedure described in Section IV is distributed among the agents assisting learners and those assisting class managers. As it will be discussed in the experimental Section, the aim of the distributed procedure is to let the system to reach a balance in terms of convenience among all the considered actors of the proposed OSN EL scenario.

III. The multi-agent e-Learning architecture

In the proposed approach, OSN users (i.e. learners) are supported by intelligent software agents [33] capable to perform all the activities aimed at organizing classes basing on the measures presented in Section II. All the agents execute a set of tasks which are briefly summarized below, and categorized as *Learner Agent Behavior* and *Class Agent Behavior*.

The Learner Agent Behavior. The behavior of a learner agent consists of several tasks periodically executed to maintain data useful to run the CF algorithm (see Section IV). Let u_k be the generic learner and a_k his/her agent, the following tasks are triggered by the learner and executed by the agent as: (*i*) Any interaction of learner u_k with one or more peers will trigger agent a_k to update Behavioral measures; (*ii*) Any reliability change of a user u_j that interacted with a peer, will trigger a_k to update the Reliability measure; (*iii*) The Convenience measure will be updated for any change in the Reliability measure; (*iv*) Behavioral and Trust measures are periodically sent to the class agent once and if they have been recalculated; (*v*) The generic software agent a_k will assist user u_k to take decision about joining with or leaving classes.

The Class Agent Behavior. The behavior of a Class Agent consists of several tasks executed periodically to maintain data useful to run the CF algorithm (see Section IV). Let c_j be a class and A_j the associated software agent, the following tasks are triggered by the interactions among software learner agents a_k and the class agent A_j as: (*i*) Any message of a_k containing updated Behavioral and/or Convenience measure will trigger agent A_k to update Behavioral and/or Convenience measures for the whole class; (*ii*) Whenever the Behavioral measure of the class c_j has changed, A_j will send the updated measure to all the learner agents of the class c_j; (*iii*) A_k will assist the class manager of c_j to take decision about the requests coming from agents a_k to join with or leave classes.

IV. The distributed procedure for Class Formation (CF)

In our approach, each *Learner Agent* has: (*i*) to update all the proposed measures whenever one or more interaction occurred, (*ii*) to send the new values to its class agents and (*iii*) to assist its own user to take decision about joining with or leaving classes by executing the CF algorithm (For this aim, it will receive behavioral and trust measures from its own class agents).

Each *class agent* has (*i*) to wait for learner agents messages in order to update the proposed measures of the entire class, (*ii*) to send updated behavioral measure for allowing learner agents to update their own convenience measures and (*iii*) to assist its own class manager to take decision about the requests coming from learner agents to join with or leave classes.

A. The distributed CF procedure.

Let T be the time between two consecutive steps of the CF procedure executed by the generic learner agent, in order to join with a set of classes of the same topic. We also suppose that agents can query a distributed database named CR (Class Repository) on which the list of the classes is stored.

The CF procedure performed by the learner agents (See Fig. 1(a)). Let X_n be the set of the classes the agent a_n is affiliated to, and N_{MAX} the maximum number of classes an agent can analyze at time t, with $N_{MAX} \geq |X_n|$. Besides, suppose that a_n stores into a cache the class profile of each class contacted in the past and the timestamp d of the last run of the CF procedure for that class. Let the timestamp ξ_n and $\chi_n \in [0,1]$ be two thresholds fixed by the agent a_n. The ratio of the procedure for the learner agent is to improve the convenience in joining with a class. Therefore, firstly the values of convenience are recalculated if older than ξ_n (lines 1-4). Then, candidate classes are sorted in a decreasing order based on their Convenience value (line 5). In the loop in lines 7-16 the N_{Max} classes are selected. If the classes in the set L_{good} are not in the set X_n, then agent a_n could improve the convenience of the owner if the classes accept the user for joining with.

The CF procedure performed by the class agent - Fig. 1(b). Let K_c be the set of the agents affiliated to the class c, and K_{MAX} the maximum number of learners allowed to be within the class c [1] with $||K_c|| \leq K_{MAX}$. Suppose that the class agent A_c stores into a cache the profile P of each user u

[1] For convenience it is assumed the same for all the classes and topics

```
Input:
  X_n, N_MAX, ξ_n, χ_n;
  Y = {c ∈ C} a random class set : |Y| ≤ N_MAX, X_n ⋂ Y = {0},
  Z = (X_n ⋃ Y)
for c ∈ Z : d_c > ξ_n do
    Send a message to A_c to retrieve the profile P_c.
    Compute γ_{u_n,c}
end for
Let be L_good = {c_i ∈ Z : i ≤ j → γ_{u_n,c_i} ≥ χ_n}, with
    |L_good| = N_MAX
j → 0
for c ∈ L_good ∧ c ∉ X_n do
    send a join request to A_c
    if A_c accepts the request then
        j → j + 1
    end if
end for
for c ∈ {X_n − L_good} ∧ j > 0 do
    Sends a leave message to c
    j → j − 1
end for
```

```
Input:
  K_c, K_MAX, ω_c, π_n, a_r, Z = K_c ⋃ {a_r};

1: if (V(r, S_c) < V_c ∨ |K_c| ≥ K_MAX) then
2:     Send a reject message to a_r
3: else
4:     for a ∈ K_c do
5:         if d_u ≥ ω_c then
6:             ask to a its updated profile
7:         end if
8:     end for
9:     for a ∈ Z do
10:        compute η_{c,a}
11:    end for
12:    Let be K_good = {a ∈ Z : γ_{c,a} ≥ π_c}
13:    for a ∈ K_c − K_good do
14:        send a leave message to a.
15:    end for
16:    if a_r ∈ K_good then
17:        the request of a_r is accepted
18:    end if
19: end if
```

Fig. 1. CF algorithm. Top (a): Learner Agent. Bottom (b): Class Agent

managed by his/her learner agent $a \in K_c$ and the timestamp d_u of its acquisition. The procedure run by A_c is triggered whenever a join request by a learner agent a_r (in the interest of u_r) is received by A_c (with the profile P_r). Let the timestamp ω_c and $\pi_n \in [0, 1]$ be two thresholds fixed by the agent A_c. If the class has reached this maximum, no more students will be accepted. By lines 4-8 the class agent asks the updated profile of its students to update their convenience $\gamma_{c,a}$ (lines 9-11) so that a new sorted set $K_{good} \subset \{K_c \bigcup a_r\}$ is built (line 12). Then, the class agent (*i*) will send a *leave* message to all the learner agents a having a convenience $\gamma_{c,a}$ or (*ii*) if $a_r \in K_{good}$ (line 16), the agent request is accepted.

V. Experiments

In order to evaluate the described approach, we performed some experiments to investigate on the convergence of the CF algorithm described in Section IV. As a measure of the internal convenience for a class c_j, we introduced the concept of *Average Convenience* (AC), computed as the average of all the measures of convenience $\eta_{j,i}$ computed by $c_j \in \mathcal{C}$ for all its students $u_i \in c_j$. To measure the global convenience of all the classes of $\mathcal{N}$, we computed the mean $MAC = \sum_{c_j \in \mathcal{C}} AC_j / ||\mathcal{C}||$ and the standard deviation $DAC = \sqrt{\sum_{c_j \in \mathcal{C}} (AC_j - MAC)^2 / ||\mathcal{C}||}$.

A first test involved three scenarios consisting of 50, 100, and 200 e-Learning classes, as summarized in Table I. To compute the convenience, we assumed that 20% of OSN members is unreliable. Behavioral coefficients h_{req} and h_{off} and the values of trust (τ), have been sampled from a normal distribution [34] around specific mean and standard deviation (stdev), see Table I. In particular, τ_r is the mean of generated trust values for reliable users, while τ_u is the mean for unreliable users. Moreover, for this set of experiments, the ratio $r = \frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$ was set to 1. Besides, the starting composition of classes is random. Table II shows the results of the execution of the CF algorithm for the three scenarios reported in Table I, that shows the initial value of MAC/DAC (epoch $T_0 = 0$) and the final one (epoch $T_e = 20$). Indeed, we have verified that after 20 epochs of executions, the MAC has reached a very stable value. It can be observed that, the improvement, in terms of MAC, at the end of the experiments, is about the 8% for all the configurations and, since the ratio $\frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$ is the same for the three scenarios without relevant variations, the subsequent were driven by r.

For the second set of experiments we assumed a variable value of $r = \frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$, as shown in Table III, ranging from 0.1 to 0.9. A value $r < 1$ say us that users, in overall, can join more places ($N_{max} \cdot |U|$), than the total allowed ($K_{max} \cdot |C|$). In particular, the best improvement, in terms of MAC, is for $r = 0.4$ (+20%), $r = 0.5$ (+16%) and $r = 0.6$ (+20%). It means that, from one hand, finding a class to improve the personal convenience γ is a bit more difficult for the user when $r < 1$, therefore the CF algorithm helps to improve the MAC with respect to the random composition of classes. Nevertheless, the algorithm clearly needs a certain degree of freedom to give some benefits. Therefore, when r is very small, the improvements, in terms of MAC are comparable to those given for values of r close to 1. In overall, these results point out that the CF algorithm gives, on average, a relevant

TABLE I
CF algorithm. Simulation parameters

Sc.	$\|C\|$	$\|U\|$	K_{Max}	N_{Max}
1	50	200		
2	100	400	20	5
3	200	800		

	τ_r	τ_u	$\{h_{Req}, h_{off}\}$
mean	0.8	0.3	0.5, 0.5
stdev	0.2	0.2	0.2, 0.2

TABLE II
Results with $r = 1.0$

	Sc 1		Sc 2		Sc 3	
	MAC	DAC	MAC	DAC	MAC	DAC
T_0	0.63	0.12	0.62	0.12	0.62	0.12
T_e	0.67	0.10	0.66	0.10	0.67	0.12

TABLE III
MAC AND DAC

	r=0.1		r=0.2		r=0.3		r=0.4		r=0.5	
	MAC	DAC	MAC	DAC	MAC	DAC	MAC	DAC	MAC	DAC
T_0	0.61	0.07	0.59	0.02	0.60	0.03	0.60	0.04	0.60	0.08
T_e	0.61	0.08	0.63	0.08	0.69	0.04	0.70	0.10	0.73	0.06
	r=0.6		r=0.7		r=0.8		r=0.9			
	MAC	DAC	MAC	DAC	MAC	DAC	MAC	DAC		
T_0	0.60	0.07	0.63	0.09	0.63	0.09	0.62	0.11		
T_e	0.70	0.09	0.69	0.08	0.68	0.08	0.67	0.10		

improvement of the convenience for the classes.

In order to test the effectiveness of the trust model we have verified, by simulations, that the class formation algorithm will lead to high and stable values of average convenience. Simulations have shown that the CF algorithm will lead significant benefits in terms of average quality of interactions.

VI. Related Work

Group/class formation is an important task to promote EL activities and obtain effective results [36]. In particular, forming random groups/classes may cause absence of participation and motivation [37]. A recent survey on group/class formation [38] analyzes about 250 works. The authors discovered that the 20% of studies on group formation in collaborative EL and a 20% of them adopt probabilistic models, while the remaining studies rely on various AI techniques. Among them, an interesting work deals with strategies for group formation based on individual behaviors [39] obtained by monitoring communication data. The results show that the students participation in small groups is correlated with their behavior in the class. Therefore, authors suggest to use these information to allocate heterogeneously initial classes into small groups. It partially differ from our approach that is aimed at grouping individuals with similar behaviors, in terms of "positive" and "negative" interactions. Besides, a relevant component in our proposal are the trust relationships from OSN data that in [39] is neglected.

A recent survey [40] dealt with the recommender systems for Technology Enhanced Learning (TEL). These systems recommend a wide variety of EL resources but their basic requirements are different from other domains and, therefore, specific methods must be adopted to evaluate them. Our work includes a recommender system for learners focused on the interactions occurring among them. In [41], two new collaborative team leadership and operational models for EL including indexes of trust, reflexivity and shared procedural knowledge are proposed. They attempt to improve practice in EL in team-based lifelong learning projects. The authors stated that EL teams take benefit from collegian participation in a trusted environment. Also, social skills and knowledge sharing are considered key aspects that, through collegiality and mutual trust, will enable to build innovative, fast-moving EL projects.

In [42] is analyzed the state-of-the-art of the "socialization" of EL activities and an automated approach to find proper people to form EL groups in OSN is described. As in our work, it is considered that, in addition to the common criteria to form groups, OSNs allow the access at a myriad of relationship data. By means of these data, suitable metrics can be created to weight the "edges" between users. The proposed algorithm to form groups simply explores the whole OSN to find a minimal number of proper candidates to form a group able to optimize a group EL experience. Differently, we exploit the concept of trust by combining reliability and reputation. Finally, in [43] the student use of Facebook at the University of Cape Town is analyzed by showing positive benefits to build EL micro-communities on Facebook but certain existing challenges, as including ICT literacy and uneven access, remain opened.

VII. Conclusions and Future Work

Class formation in e-Learning is a critical task for the quality of such activities. In this work we focused on a distributed algorithm supported by a trust model and some behavioral measures based on information coming from the OSN (i.e. users trust relationships, interaction quality, historical attitude to interact with peers) to improve the metrics for dynamic class composition in OSNs. This flexibility is aimed at improving the quality of learning experiences and it is obtained by combining information about trust and previous interactions in a unique measure named "convenience". In this work we have shown a first set of experimental results obtained by simulating an artificial scenario with a variable number of users and groups. The results have shown that the class formation algorithm will lead to high and stable values of average convenience.

As future work, we will perform a further experimental campaign in order to verify that the convergence to high values of convenience leads to significant benefits in terms of average quality of interactions. Moreover, a further set of experiments is needed to verify the effectiveness of the trust model to limit malicious behaviors, in order to give trust values which reflect the actual behavior, in terms of overall reliability, of the students.

References

[1] J. L. Moore, C. Dickson-Deane, and K. Galyen, "e-learning, online learning, and distance learning environments: Are they the same?" *The Internet and Higher Education*, vol. 14, no. 2, pp. 129–135, 2011.

[2] J. Mason and P. Lefrere, "Trust, collaboration, e-learning and organisational transformation," *Int.l J. of Training and Development*, vol. 7, no. 4, pp. 259–270, 2003.

[3] https://www.facebook.com, 2016.

[4] https://www.twitter.com, 2016.

[5] P. Grabowicz, L. Aiello, V. Eguiluz, and A. Jaimes, "Distinguishing topical and social groups based on common identity and bond theory," in *Proc. of the ACM Int. Conf. WSDM 2013*. ACM, 2013, pp. 627–636.

[6] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group Formation in Large Social Networks: Membership, Growth, and Evolution," in *Proc. of the 12th ACM SIGKDD Int. Conf.* ACM, 2006, pp. 44–54.

[7] S. Kairam, D. Wang, and J. Leskovec, "The life and death of online groups: Predicting group growth and longevity," in *5th ACM int. conf. on Web search and data mining Proc. of*. ACM, 2012, pp. 673–682.

[8] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "Hyson: A distributed agent-based protocol for group formation in online social networks," in *Multiagent System Technologies*. Springer Berlin Heidelberg, 2013, pp. 320–333.

[9] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "Improving the compactness in social network thematic groups by exploiting a multi-dimensional user-to-group matching algorithm," in *Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on*. IEEE, 2014, pp. 57–64.

[10] V. Vasuki, N. Natarajan, Z. Lu, B. Savas, and I. Dhillon, "Scalable affiliation recommendation using auxiliary networks," *ACM Trans. on Intelligent Systems and Technology*, vol. 3, no. 1, p. 3, 2011.

[11] J. Gorla, N. Lathia, S. Robertson, and J. Wang, "Probabilistic group recommendation via information iatching," in *Proc. of the Int. World Wide Web Conf. (WWW '13)*. ACM Press, 2013, pp. 495–504.

[12] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "A distributed agent-based approach for supporting group formation in p2p e-learning," in *AI* IA 2013: Advances in Artificial Intelligence*. Springer International Publishing, 2013, pp. 312–323.

[13] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné, "Trust and compactness in social network groups," *Cybernetics, IEEE Transactions on*, vol. 45, no. 2, pp. 205–216, Feb 2015.

[14] W. Tan, S. Chen, J. Li, L. Li, T. Wang, and X. Hu, "A trust evaluation model for e-learning systems," *Systems Research and Behavioral Science*, vol. 31, no. 3, pp. 353–365, 2014.

[15] A. Comi, L. Fotia, F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "Forming homogeneous classes for e-learning in a social network scenario," in *Intelligent Distributed Computing IX*. Springer International Publishing, 2016, pp. 131–141.

[16] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "A trust model for competitive cloud federations," *Complex, Intelligent, and Software Intensive Systems (CISIS)*, pp. 469–474, 2014.

[17] F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "A trust-based, multi-agent architecture supporting inter-cloud vm migration in iaas federations," in *Internet and Distributed Computing Systems*. Springer International Publishing, 2014, pp. 74–83.

[18] A. Comi, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarnè, "A qos-aware, trust-based aggregation model for grid federations," in *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*. Springer Berlin Heidelberg, 2014, pp. 277–294.

[19] T. Snijders, "Network dynamics," *The Handbook of Rational Choice Social Research. Stanford University Press*, pp. 252–279, 2013.

[20] D. Rosaci and G. M. L. Sarné, "Matching users with groups in social networks," in *Intelligent Distributed Computing VII*. Springer, 2013, pp. 45–54.

[21] A. Comi, L. Fotia, F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "Using semantic negotiation for ontology enrichment in e-learning multi-agent systems," in *Complex, Intelligent, and Software Intensive Systems (CISIS), 2015 Ninth International Conference on*. IEEE, 2015, pp. 474–479.

[22] ——, "Supporting knowledge sharing in heterogeneous social network thematic groups," in *Complex, Intelligent, and Software Intensive Systems (CISIS), 2015 Ninth International Conference on*. IEEE, 2015, pp. 480–485.

[23] P. De Meo, F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "Similarity and trust to form groups in online social networks," in *On the Move to Meaningful Internet Systems: OTM 2015 Conferences*. Springer International Publishing, 2015, pp. 57–75.

[24] A. Comi, L. Fotia, F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "An evolutionary approach for cloud learning agents in multi-cloud distributed contexts," in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2015 IEEE 24th International Conference on*. IEEE, 2015, pp. 99–104.

[25] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "Recommending users in social networks by integrating local and global reputation," in *Internet and Distributed Computing Systems*. Springer International Publishing, 2014, pp. 437–446.

[26] A. Comi, L. Fotia, F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "A distributed reputation-based framework to support communication resources sharing," in *Intelligent Distributed Computing IX*. Springer International Publishing, 2016, pp. 211–221.

[27] H. Songhao, S. Kenji, K. Takara, and M. Takashi, "Towards new collaborative e-learning and learning community using portfolio assessment," in *World Conf. on E-Learning in Corporate, Government, Healthcare, and Higher Education*, vol. 2008, no. 1, 2008, pp. 1270–1275.

[28] S. Franklin and A. Graesser, "Is it an agent, or just a program?: A taxonomy for autonomous agents," in *Intelligent agents III Agent Theories, Architectures, and Languages*. Springer, 1997, pp. 21–35.

[29] T. Grandison and M. Sloman, "Trust management tools for internet applications," in *Trust Management*. Springer, 2003, pp. 91–107.

[30] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *HICSS '00: Proc. of the 33rd Hawaii Int. Conf. on System Sciences*, vol. 6. IEEE Computer Society., 2000.

[31] G. Zacharia and P. Maes, "Trust management through reputation mechanisms," *Applied Artificial Intell.*, vol. 14, no. 9, pp. 881–907, 2000.

[32] S. Garruzzo, D. Rosaci, and G. M. L. Sarné, "Masha-el: A multi-agent system for supporting adaptive e-learning," in *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, vol. 2. IEEE, 2007, pp. 103–110.

[33] M. Wooldridge and N. Jennings, "Intelligent agents: Theory and practice," *The knowledge eng. review*, vol. 10, no. 2, pp. 115–152, 1995.

[34] K. Hopkins, G. Glass, and B. Hopkins, *Basic statistics for the behavioral sciences* . Prentice-Hall, 1987.

[35] S. Garruzzo, D. Rosaci, and G. M. L. Sarné, "Isabel: A multi agent e-learning system," in *Intelligent Agent Technology, 2007. IAT'07. IEEE/WIC/ACM International Conference on*. IEEE, 2007, pp. 485–488.

[36] F. Rennie and T. Morrison, *E-learning and social networking handbook: Resources for higher education*. Routledge, 2013.

[37] P. Dillenbourg, "Over-scripting cscl: The risks of blending collaborative learning with instructional design." *Three worlds of CSCL. Can we support CSCL?*, pp. 61–91, 2002.

[38] W. Cruz and S. Isotani, "Group formation algorithms in collaborative learning contexts: A systematic mapping of the literature," in *Collaboration and Technology*. Springer, 2014, pp. 199–214.

[39] N. Jahng and M. Bullen, "Exploring group forming strategies by examining participation behaviours during whole class discussions." *European J. of Open, Distance and E-Learning*, 2012.

[40] M. Erdt, A. Fernandez, and C. Rensing, "Evaluating recommender systems for technology enhanced learning: A quantitative survey," *Learning Technologies, IEEE Trans.*, vol. 8, no. 4, pp. 326–344, 2015.

[41] J. Jameson, G. Ferrell, J. Kelly, S. Walker, and M. Ryan, "Building trust and shared knowledge in communities of e-learning practice: collaborative leadership in the jisc elisa and camel lifelong learning projects," *British J. Educat. Tech.*, vol. 37, no. 6, pp. 949–967, 2006.

[42] S. Brauer and T. C. Schmidt, "Group formation in elearning-enabled online social networks," in *Interactive Collaborative Learning (ICL), 2012 15th Int. Conf. on*. IEEE, 2012, pp. 1–8.

[43] T. E. Bosch, "Using online social networking for teaching and learning: Facebook use at the university of cape town," *Comm.: South African J. for Comm. Theory and Research*, vol. 35, no. 2, pp. 185–200, 2009.