Social Information Retrieval with Agents

Federico Bergenti
Dipartimento di Matematica e Informatica
Università di Parma
Parma, Italy
federico.bergenti@unipr.it

Agostino Poggi, Michele Tomaiuolo
Dipartimento di Ingegneria dell'Informazione
Università di Parma
Parma, Italy
{agostino.poggi, michele.tomaiuolo}@unipr.it

Abstract—**With the widespread adoption of online social networks as a crucial means for communication, social information retrieval is becoming one of the most interesting areas of research in terms of the large number of–theoretical and practical–issues that it encompasses. We argue that agent technology is central in supporting the decentralization of next generation online social networks and the synergistic pairing of agents and social networks is evident, if nothing else, because members of a social network interact as agents do in a multi-agent system. In this paper we investigate the possibilities that agent technology can offer to social information retrieval and we emphasize the role that agents and multi-agent systems can play by presenting Blogracy, an agent-based online social network system.**

Social information retrieval, online social networks, multi-agent systems

I. Introduction

Nowadays it is common opinion that most *Information Retrieval (IR)* systems and related tasks are more than ever embedded in rich contexts. Documents no longer exist on their own: *(i)* they are connected to other documents; *(ii)* they are associated with the individuals that contributed to produce them, and with the individuals that, possibly partially, consume them; *(iii)* they are dependent of the social networks of their respective producers and consumers; and *(iv)* they are related to the context in which individuals operate. All such features enrich documents and the correct use of them can drastically increase the performance of IR techniques.

Social IR can be broadly defined as the synergistic embedding of information about social networks of individuals and their relationships into IR processes (see, e.g., [10][12][26]).

The traditional models for IR have to do with documents, queries, and their relations. For example, a document is relevant to a query, but a document may reference to other documents and, likewise, a query may be related to another query. In a similar spirit, social networks model individuals and their relations, like friends and family, acquaintances, and collaborators (see, e.g., [4][10][40]). Unfortunately, traditional IR techniques do not model individuals, neither in their role as users of the system, nor as authors of the retrieved documents. This circumstance severely limits the contextual information available to the IR techniques, and the promise of social IR is to boost the performance of IR techniques by means of the integration of socially relevant contextual information. By incorporating individuals into the model, IR techniques gain a greater insight into the documents under observation. New associations between entities become apparent, e.g., individuals appear in their role as information producers or information consumers, queries relate to an individual's information needs, and they describe a topic that falls into the interests of an individual.

The ultimate motivation for social IR is rooted in the belief that an information producer and his/her product cannot be separated, and, likewise, information and its consumers cannot be separated.

Understanding the social context in which the production and consumption of information takes place is especially important when only limited understanding of the information under consideration is available. Traditional IR techniques are based solely on analysing the content of documents and, while very successful in many contexts, they fail when the information of documents under observation is partial. In this sense, social IR can be understood as a formalization of the search techniques that we commonly use to assess the quality of information–by looking at the author's standing in his/her community. The same principle can be applied to other instances of information production and consumption in a social environment: we tend to judge information also on the basis of the reputation that its producers and respective consumers have in their social context.

For these, and possibly many other reasons, we believe that social IR poses new challenges and questions that are worth investigating. We also believe that the quality of contextual information available to social IR techniques heavily depends on the tools that the social IR system can adopt to grasp social aspects relevant to IR tasks.

Individuals and their *social networks* are cornerstones of social IR, but *users* and *online social networks* can perform orders of magnitude better. Social networks are typically described as finite sets of actors and relations defined on them (see, e.g., [40]). In this context, an actor is essentially any social entity, such as an individual, a corporate, or a collective social unit; and a relationship can be any kind of social tie that establishes a link between a pair of actors. Nowadays, the most widely known social networks are Web platforms, often called online social networks, where users not only put or read content, but they are also linked with relationships. The diffusion of online social networks is opening new scenarios for envisaging novel kinds of applications, either to support new social networking activities, or to exploit established relationships among users and use them to offer higher-level services.

With this in mind, we believe that online social networks are still not sufficient because in their current incarnation they tend to be highly centralized and to form, sometimes huge, *islands*. The recent clamor about the PRISM program and the release of classified documents by Edward Snowden [19] has also raised many questions about the privacy issues of current social networking applications. We think that social IR can be taken to its full potential by eliminating the boundaries of current online social networks and by fostering IR tasks that may break across networks. We believe that agent technology is crucial to enable such an envisioned decentralization of online social networks because of the inherent decentralized nature of multi-agent systems (see, e.g., [9][29]) and because of their intrinsic characteristics in terms of management of trust, privacy, and reputation (see, e.g., [2][7][36][37][39]).

In the following section we outline the major features that the synergistic pairing of online social networks and agent technology offers, and we survey recent research effort that explored such a combination in various contexts. Finally, we present an agent-based online social network system, namely Blogracy [15][16], that promotes decentralization and that is therefore a solid base for taking social IR to its full potential.

II. Agent Technology and Online Social Networks

In order to understand the relationship between multi-agent systems and social networks it is important to understand the intrinsic computational properties of social networks. The first insights on such properties came from Milgram's experiment that led to the investigation of the so-called *small world phenomenon* [28]. In Milgram's experiment, a group of randomly chosen people received the name and address of another randomly chosen person living in a distant city. Then, people were asked to route a mail message toward the target person chosen only among their friends or close acquaintances. The experiment pointed out that: *(i)* people are connected through very short chains of acquaintances, with a 5-6 links length, in average; and *(ii)* people is able to route the messages to the target person using local information and performing local actions.

A result of the Milgram's experiment is that the behavior of people was similar to that of rational autonomous agents. In fact, every person choses his/her successor in his/her list of acquaintances considering elements like geographical proximity or profession similarity, which is essentially using only local and elementary information to pursue a global complex goal, with no need to use their humanity. From our point of view, this is a particularly relevant conclusion, since it points to the emergence of a global behavior from local strategies, a feature that is one of the key properties of multi-agent systems.

More recently, the studies on the small world problem led to two computationally-based approaches to search for people within social networks (a comprehensive review of different algorithms and their performance is presented in [1]). The original experiment of Milgram led to a machine-based approach consisting in the problem of looking for a remote agent given its unique identifier. A successive approach deals with finding a specific agent who matches a given criterion, e.g., having a given capability or expertise. This is quite similar to the problem of navigating one's social network in search for someone with a given expertise or for an answer to a specific question. In an enterprise setting, this is the problem of looking inside the organization for someone able to solve a specific problem or able to answer to a specific question. When solved with agent-based techniques, this problem resembles the *collaborative filtering* one and is usually termed as *expert finding*, and authors use such definitions interchangeably.

The expert finding problem is similar to Milgram's original problem in that the social network of each node is the search space in which the request is processed. It should be emphasized that both problems strongly rely on the local search ability and the occurrence of the small world phenomenon, i.e., on the fact that two random individuals are preferably mostly connected by short chains of acquaintanceships. If social networks were not searchable it would be impossible to efficiently find a person matching some criteria unless personally known and, then, the Milgram's experiment would have failed. On the other hand, if the chains were very long, the search would be not feasible.

A pioneering research on this subject was done in [24][25]. These papers describe *ReferralWeb*, an agent based interactive system for reconstructing, visualizing, and searching social networks on the Web whose main focus is selecting an expert of a given field in one's (extended) social network.

In ReferralWeb a social network is modelled by a graph where the nodes represent individuals and an edge between nodes indicates that a direct relationship between the individuals has been discovered. For ReferralWeb a direct relationship is implied when the names are in close proximity in any document publicly available on the Web, e.g., home pages, co-authorship in published papers, or organization charts in institutional Web sites. ReferralWeb does not require its users to fill a user profile describing their skills.

The constructed network is then used to guide the search for people or documents in response to user queries. A person can: *(i)* ask to find the chain between himself/herself and a named individual; *(ii)* search for an expert in a given topic providing a maximum social radius (the number of links in the chain connecting the person performing the query with the expert); and *(iii)* request a list of documents written by people close to a given expert.

The key idea of ReferralWeb is to use the social network to make more focused and effective searches. It is not meant to be a tool to create social networks, i.e., to help people socializing. ReferralWeb also emphasizes the importance of the referral chains themselves as means to build trust on the selected experts.

MARS is a multi-agent referral system that finds experts on the basis of personal agents able to learns the user's preferences and interests, and able to build an expertise model of the other users on the basis of their responses [42]. Each user is assigned an agent who: *(i)* learns the user's preferences and interests, and *(ii)* maintains a view of its user's acquaintances, that are used to prioritize incoming queries, possibly issuing referrals when other users might be more suitable to answer a given query. Each agent first rates, according to the user's feedback, those agents that provided an answer and those agents that referred to them and, then it modifies its neighbors accordingly. Consequently, the referral system evolves to reflect the changes in the social network.

A response to a query specifying what information is being sought, if given, may consist of an answer or a referral, depending on the query and on the expertise of the answering agent. If an agent is reasonably confident that its expertise matches the query, it directly answers; otherwise, it yields referrals to other supposedly expert agents.

Each agent maintains models of its acquaintances. An agent sends its query initially only to some of its neighbors, that are the individuals with the closest acquaintances. The agent who receives a referral may pursue it even if the referred party is not already an acquaintance; good acquaintances are going to be promoted to neighbors on an intuitive basis. When new neighbors are considered, some of previous ones will be discarded, since the number of neighbors is bounded. The authors of MARS decided that reputation should increase slowly, but it should fall out quickly, and that rewards and penalties are greater for agents nearer to the answering agent. This implies that a bad decision results in bad reputation, but if agents just started a chain of referrals leading to a bad agent, then the penalty is modest.

The expertise model is captured through a classical vector space model [38]. Term vectors are used to express both the profile of the user and the acquaintance model for each of its acquaintances. Since a term vector also models the required expertise, the cosine of the angle between the user vectors with the subject vector yields the competence of a user in a given subject. Intuitively, when there are two agents with expertise in the same direction, the one with the greater expertise is more desirable.

Each agent learns its user's profile and its acquaintance models based on an evaluation of the received answers as well as on the referrals that led to them. A referral graph, which is local to each agent, encodes how the computation spreads, as a query originates from an agent, and referrals or answers are sent back to this agent.

Yenta is a matchmaking system that helps people with similar interests to get in touch [14]. Yenta agents do not query the Web; instead, they scan user's e-mails, Usenet posts and (possibly) documents in order to discover their users' interests and hobbies. The idea is that many potentially interesting people do not write publicly and so they become invisible to tools relying on public data. Collected data are then used to introduce users' to each other. Considering that in the 90's Web communities were built around the idea of common interests rather than on personal acquaintance, the system was a truly distributed social networking system, at least for the time.

Shine (SHared INternet Environment) [41] is a fully peer-to-peer framework for network community support. The system has been implemented and a presented in [41]. The framework provides design guidelines and enables different applications to share program components and to cooperate, and it features a peer-to-peer architecture through which personal agents can flexibly form communities where users can exchange information with peer agents. Essentially, Shine is a middleware for collaborative workspaces especially tailored to implement various collaborative workspaces.

Shine provides a personal agent to each and every single user and three core modules compose each agent: the person database, the plan execution module and the communication module. In addition, one or more applications are installed in each agent. Such applications provide their services to the user by means of functionalities of the core modules via a dedicated API.

The person database of Shine holds data on people and on personal agents. The data include information on the agent and on the user whom the agent is associated with, as well as other agents and people known to the agent. An agent holds the data required to form a community that is suitable to the user in the person database and it exchanges data among other agents when necessary. In the Shine architecture, the user and his/her personal agent correspond in a one-to-one manner. Therefore, in the person database, data on both a user and his/her personal agent are stored without distinguishing between them.

In order to support communities, Shine's authors added the concept of person set. Each community is represented in the person database as a person set and the framework provides operations for dealing with such sets, e.g., functions to broadcast messages to the members of a community. In this way Shine agents can flexibly determine the range of broadcasting by regarding a person set as the destination list.

In Shine a peer-to-peer network is formed directly connecting the communication modules of groups of agents. The function of such modules is simply to exchange messages with each other. Given the fact that the agents live in a ubiquitous computation environment, the module is layered so that only the lower layer depends on the environmental details.

Agents in Shine are goal-driven through plans: a plan is description of agent action rules. Multiple plans are executed concurrently in the plan execution module of each agent. Some plans are prepared to perform services of applications while other plans are provided by Shine to do fundamental or common tasks. A plan acts in response to external events, e.g., receiving a message from another agent, a user input or a modifications in the person database.

SNIS is a multi-agent system where agents utilize the connections of a user in the social network to facilitate the search for items of interest [21]. In particular, each agent is associated with a user and it observes the user's activities and, in particular, the ratings and comments provided by the user to items retrieved from the social network. SNIS has been experimented in the Flickr domain [27]; the system scans photos posted by all of the user's contacts and gathers statistics about their categories and user comments (which represent user interest) and such information is used to facilitate the search for items of interest.

III. Blogracy

It is common opinion that multi-agent systems can play an important role to support completely decentralized or federated social networking platforms. Indeed, one of the very specific features of multi-agent systems is the sociality of agents, i.e., their ability to communicate in a semantic way (see, e.g., [6][32]) and to develop trust relationships among them. Moreover, agents can express their communication acts by means of acknowledged standards for interoperability among diverse systems, like FIPA [13], and they can exchange

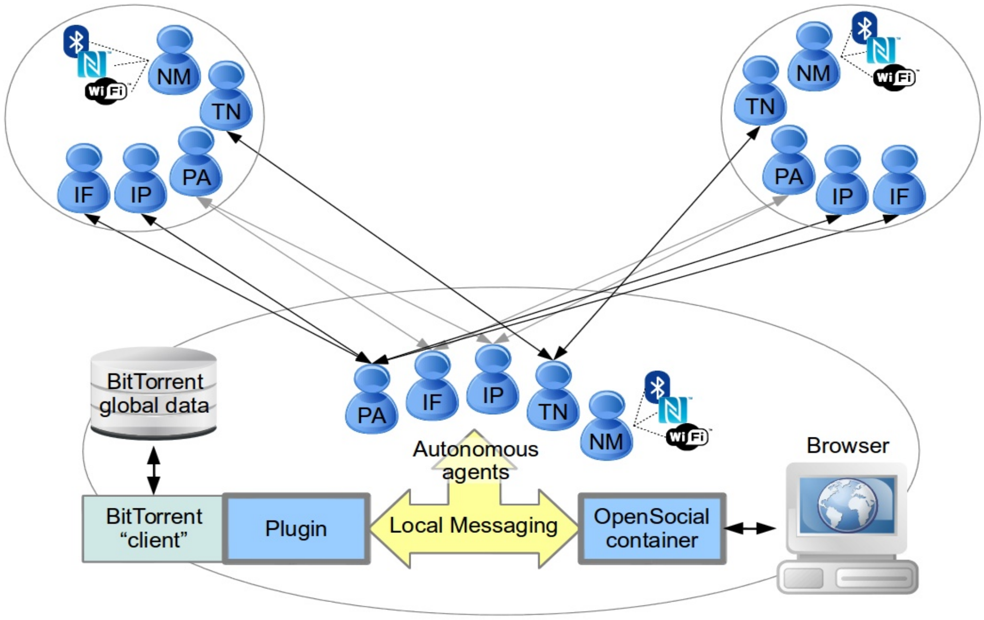

Figure 1. The multi-agent architecture of Blogracy.

messages directly in a peer-to-peer way. Therefore, it is not surprising that these two technologies are often applied together for developing advanced social platforms.

In particular, multi-agent systems have been used as: *(i)* an underlying layer, or middleware, for developing social networking platforms; and *(ii)* a technology to increase the autonomous and intelligent behavior of existing systems.

For the first type of applications of multi-agent systems, many of the distinguishing features of multi-agent systems can be fully exploited. Indeed, multi-agent systems provide semantic communication among agents, which is handy for expressing all the different actions that users can perform in a social platform. The different types of messages can be understood according to their meaning and applied according to existing trust relations among the users and their respective agents. In addition, complex negotiation protocols can help creating acknowledgements and trust among users, in an automatic or assisted way, without exposing sensitive data. Mobility can also be useful for moving the computation closer to data, if massive analysis has to be performed, but it can also be handy for adding functionality to a node of a decentralized social platform or to a user's client application.

In the second type of applications of multi-agent systems, agents are mainly used because of their proactive and reactive behaviors that can provide recommendations of both users and content, and that can enable the personalization of results. Reactive abilities are particularly important in a social networking environment where interesting events happen frequently and where users can be easily distracted by the huge information flow, which is associated with highly connected social networks. Sensing the environment and executing automatic tasks can reduce this overload significantly. Goal-oriented behaviors, on the other hand, can support users in prosecuting their long term objectives about friend and content discovery, i.e., to discover known persons registered to the network, to make new acquaintances with users with common interests, to find interesting content hidden in less relevant data or from new sources.

Both kinds of features of agents and multi-agent systems have been already integrated in the design of *Blogracy* [7][15][33], an agent-based system whose goal is to provide users with adaptive and composite services on top of core features. At the lower level, Blogracy uses widespread and stable peer-to-peer technologies, such as distributed hash tables and the BitTorrent protocol, for coping with the intrinsic defects of centralized architectures and to become the basis of solid distributed social networking platforms. At the higher level, it takes advantage of multi-agent systems for simplifying the implementation of social network services in a decentralized setting.

The architecture of Blogracy is modular and composed of two basic components: *(i)* an underlying module for basic file sharing and DHT operations, built as an extension of existing implementations, and *(ii)* an OpenSocial container, i.e., a module providing the services of the social platform to the local user through a Web interface. Additionally, Blogracy supports autonomous agents to provide recommendations of both users and content, personalization of results, and trust negotiation mechanisms.

Blogracy relies only on users' nodes for its operation. Therefore, users need to perform background tasks on their own, in a distributed and decentralized way.

A layer of agents takes charge of assisting the user in finding new interesting content and connections, and in pushing the local user's activities to followers.

Figure 1 sketches the multi-agent architecture of Blogracy. A *Personal Assistant (PA)* monitors the local user's actions in the platform and it learns the user's profile, beyond information provided explicitly. The PA receives the user's queries, forwards them to the available *Information Finders (IF)* and it presents the results to the user. Moreover, a PA provides the local user with recommendations about possibly interesting content and connections available in the network. Another task performed by the PA is the personalization of results. Indeed, as a social network becomes larger and more richly interconnected, users unavoidably face some form of information overflow. A PA, using a user's profile, can arrange presented data in a way that highlight the most interesting pieces of information.

An IF is an agent that searches information on the repository contained in the node where it lives, through an automatic TF-IDF indexing algorithm and explicit hashtags associated with local posts. It provides such information both to its user and to other trusted users. An IF receives users' queries, finds appropriate results and filters them by using its user's access policies. An *Information Pusher (IP)* is an agent that monitors the changes in the local repository and that pushes new information to the PA of interested subscribers that are currently connected. An IP can forward content produced by the local user and by her/his remote acquaintances to other contacts, according to privacy preserving policies and to recent queries made by other users.

Over the OpenSocial container, Blogracy can also provide functionalities for pervasive online social networking, specifically for creating locality and proximity groups. In this case, the system has to rely on highly adaptive services both to sustain the basic operations of the location-based social networking and to provide advanced functionalities. For this purpose, each node of the social network has to host multiple agents, with different levels of agency [31][34][35]. Some of the more important agents are:

- The *Neighborhood Manager (NM)* agent, which cooperates with lower level agents to discover the users in its neighborhood;
- The *Trust Negotiator (TN)* agent, which is involved in the decisions regarding privacy and access rules; and
- The *OpenSocial (OS)* agent, which provides a bridge towards the underlying modules of Blogracy.

A user may own multiple nodes (e.g., an instance on the smartphone and an instance on his home computer) and, since the actual location of the user is important for our application, the nodes in the different devices negotiate which one should be considered active (i.e., which one determines the user location). In fact, the nodes can either determine the device that registered an explicit user action or they can ask to the user to select the device he/she is currently using.

Apart from the personal circles defined by each user, we also have two additional kinds of groups: *proximity groups* and *location groups*. Proximity groups are centered on each member of the social networking system and they represent physical closeness to such a member. Proximity groups are extremely fluid in the sense that users can physically move and consequently the set of users belonging to a proximity group varies over time. Each user configures the hysteresis, or sticky-ness, of his proximity group, i.e., how long other users are considered part of it after they are no longer physically close to him/her. Although a proximity group may be entirely public, for privacy reasons it is safer to consider only proximity groups that are subset of other groups (or of the set union of all groups, i.e., only "friends" are part of a proximity group). The NM agent informs the OS agent when users enter and leave the proximity group and the latter notifies the OpenSocial container about it.

On the other hand, a location group is associated with the users in the proximity of a given location (e.g., a classroom or a museum room) and it has a host (i.e., a node) that both identifies and supports the group. Moreover, a location group is associated with a location profile maintained either on the central server or on its host. In fact, a location, although logically different from a regular user, works in the same way and a location group is essentially a proximity group for the location.

The availability of a generic TN agent is also important since users joining a proximity group or a location group are not necessarily connected a priori in the social network, and they may need to acknowledge their profile attributes before practical social interaction. Such a negotiation requires the controlled exchange of credentials and of policies, without disclosing unnecessary sensible information, yet establishing trust if possible [2][36][37][39]. In [7], a generic library supporting zero-knowledge proof for attribute verification is presented. The same mechanisms can also facilitate the creation of trust in social networks.

Agents present different degrees of autonomy and intelligence. For example, lower level agents are mostly reactive; e.g., they inform the NM agent when a new node is discovered. The NM agent itself has some degrees of autonomy and intelligence, and it has the following duties: *(i)* it aggregates information from the agents that discover new peers; *(ii)* it informs the OS agent of the state of neighborhood; *(iii)* it tries to present a consistent view, merging data from the different sources; and *(iv)* it configures the discovering agents according to high-level criteria, such as battery consumption and hardware availability.

The OS agent is basically a gateway to the OpenSocial container and it translates the other agents' requests for the OpenSocial container. A TN agent is a true agent that performs potentially complex negotiations on its user's behalf and, depending on the configuration, it may work in full autonomy.

IV. Conclusions

This paper outlines a very promising line of research: the use of the entire spectrum of agent technology to provide concrete support to innovative social IR tasks. Agents and multi-agent systems naturally models social networks, and they can even implement large-scale online social networks as nowadays agent technology is considered a mature tool capable of supporting mission-critical, large-scale software systems (see, e.g., [3][5][8][17][22][23][30]).

Moreover, the peculiar management of decentralization and the sophisticated treatment of privacy and reputation issues make agents and multi-agent systems ideal tools to provide insightful contextual information to social IR techniques. In particular, Blogracy breaks the traditional centralized approach to the implementation of online social networks and it opens to new sources of contextual information that can be obtained by observing documents and individuals across multiple social networks. All in all, the decentralization that agent technology ensures define a novel features of documents, individuals, and relations: how they spread across different online social networks, and how they change in such a spreading over time. Moreover, the simple fact of observing individuals and documents overlapping different online social networks is immediately usable as a relevant source of contextual information.

In conclusion, we argue that agent technology provides solid and mature tools to support the design and implementation of novel social IR tools, and we believe that no effective social IR can take place if it would restrict to a single, even if enormous, online social network.

REFERENCES

[1] L. Adamic, and E. Adar, "How to search a social network," Social Networks, vol. 27, no. 3, pp. 187-203, 2005.

[2] F. Agazzi, and M. Tomaiuolo. "Trust Negotiation for Automated Service Integration", in CEUR Workshop Proceedings 1099, WOA 2013.

[3] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, "JADE: a Software Framework for Developing Multi-Agent Applications. Lessons Learned," Information and Software Technology Journal, vol. 50, pp. 10-21, 2008.

[4] F. Bergenti, E. Franchi, and A. Poggi, "Selected Models for Agent-based Simulation of Social Networks," in Social Networks and Multi Agent Systems Symposium (SNAMAS 2011), pp. 27-32. 2011.

[5] F. Bergenti, and A. Poggi, "An Agent-Based Approach to Manage Negotiation Protocols in Flexible CSCW Systems," in 4th Int. Conf. on Autonomous Agents, pp. 267-268. 2000.

[6] F. Bergenti, A. Poggi, M. Tomaiuolo, and P. Turci, "An Ontology Support for Semantic Aware Agents". Lecture Notes in Computer Science, 3529/2006, pp. 140-153. Springer. 2006.

[7] F. Bergenti, L. Rossi, and M. Tomaiuolo, "Towards Automated Trust Negotiation in MAS," in WOA 2009. Parma, Italy. 2009.

[8] F. Bergenti, A., Poggi, and M. Somacher, "A collaborative platform for fixed and mobile networks," Communications of the ACM, vol. 45, no. 11, pp. 39-44, 2002.

[9] R. Bordini, M. Dastani, J. Dix, and A. Fallah-Seghrouchni, "Multi-Agent Programming: Languages, Platforms and Applications". Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 15. Berlin, Germany: Springer, 2005.

[10] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, "Social cloud: Cloud computing in social networks," in 2010 IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, 2010, pp. 99-106.

[11] W. Croft, D. Bruce, T. Metzler, and T. Strohman, "Search engines: Information retrieval in practice". Addison-Wesley, 2010.

[12] D. Horowitz, and D. K. Sepandar. "The anatomy of a large-scale social search engine," in Procs. 19th Int. Conf. on WWW, 2010, pp. 431-440.

[13] FIPA (2014, February 10). "FIPA Specifications" [Online]. Available at http://www.fipa.org/.

[14] L. Foner, "Yenta: a multi-agent, referral-based matchmaking system," in First International Conference on Autonomous Agents, Marina del Rey, CA, USA, 1997, pp. 301-307.

[15] E. Franchi, A. Poggi and M. Tomaiuolo. Open Social Networking for Online Collaboration. International Journal of e-Collaboration 9(3), 2013.

[16] E. Franchi, and M. Tomaiuolo, "Distributed Social Platforms for Confidentiality and Resilience". In: Social Network Engineering for Secure Web Data and Services, IGI Global, 2013.

[17] F. Gandon, A., Poggi, G., Rimassa, and P. Turci, "Multi-Agents Corporate Memory Management System," Applied Artificial Intelligence Journal, vol. 16 no. 9-10, pp. 699-720, 2002.

[18] M. Genesereth, and S. Ketchpel "Software Agents," Communications of the ACM, vol. 37, no 7, pp. 47-53, 1994.

[19] M. Greene, "Where Has Privacy Gone? How Surveillance Programs Threaten Expectations Of Privacy". The John Marshall Journal of Information Technology & Privacy Law, 30(4), 5. 2014.

[20] D. Greenwood, and M. Calisti, "Engineering Web service-agent integration," in IEEE International Conference on Systems, Man and Cybernetics (Vol. 2), The Hague, Netherlands, 2004, pp. 1918-1925.

[21] A. Gursel, and S. Sen, "Improving Search In Social Networks by Agent Based Mining," in 21st International Joint Conference on Artifical Intelligence, San Francisco, CA, USA, 2009, pp. 2034-2039.

[22] M.N. Huhns et al., "Research Directions for Service-Oriented Multiagent Systems," IEEE Internet Computing, vol. 9, no. 6, pp. 65-70, 2005.

[23] N. Jennings, J. Corera, and I. Laresgoiti, "Developing industrial multi-agent systems," in First International Conference on Multi-Agent Systems (ICMAS-95), 1995, pp. 423–430.

[24] H. Kautz, B. Selman, and M. Shah, "Combining Social Networks and Collaborative Filtering," Communications of the ACM, vol. 40 no. 3, pp. 63-65, 1997.

[25] H. Kautz, B. Selman, and M. Shah, "The Hidden Web," AI Magazine, vol. 18, no. 2, pp. 27-36, 1997.

[26] S.M. Kirsch, M. Gnasa, and Armin B. Cremers. "Beyond the Web: Retrieval in social information spaces". Procs. 28th European Conf. on Advances in Information Retrieval, 84-95, Springer-Verlag, 2006.

[27] K. Lerman, A. Plangrasopchok, and C. Wong, "Personalizing results of image search on Flickr," in AAAI workshop on Intelligent Techniques for Web Personlization, Vancouver, Canada, 2007, pp. 65-75.

[28] S. Milgram, "The small world problem," Psychology today, vol. 1, no. 1, pp. 61-67, 1967.

[29] J. Muller, "Architectures and applications of intelligent agents: A survey," Knowledge Engineering Rev., vol. 13, no. 4, pp. 353-380, 1998.

[30] A. Negri, A. Poggi, M. Tomaiuolo, and P. Turci, "Dynamic Grid Tasks Composition and Distribution through Agents," Concurrency and Computation: Practice and Experience, vol. 18, no. 8, pp. 875-885, 2006.

[31] A. Negri, A. Poggi, M. Tomaiuolo, and P. Turci, "Agents for e-Business Applications", in: 5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS-2006), pp. 907-914. ACM. 2006.

[32] A. Poggi, "Developing ontology based applications with O3L," WSEAS Trans. on Computers, vol. 8, no. 8, pp. 1286-1295, 2009.

[33] A. Poggi, and M. Tomaiuolo. "A DHT-Based Multi-Agent System for Semantic Information Sharing", in Studies in Computational Intelligence, 439/2013, pp. 197-213. Springer, 2013.

[34] A. Poggi, M. Tomaiuolo, and P. Turci, "An Agent-Based Service Oriented Architecture". 157-165, In: WOA 2007, Genova, Italy.

[35] A. Poggi, M. Tomaiuolo, and P. Turci, "Extending JADE for agent grid applications", in Proc. WET ICE 2004, pp. 352-357. IEEE. 2004.

[36] A. Poggi, M. Tomaiuolo, and G. Vitaglione, "A security infrastructure for trust management in multi-agent systems", in Trusting Agents for Trusting Electronic Societies, pp. 162-179. Springer. 2005.

[37] A. Poggi, M. Tomaiuolo, and G. Vitaglione, "Security and trust in agent-oriented middleware", in On The Move to Meaningful Internet Systems (OTM 2003), pp. 989-1003. Springer. 2003.

[38] G. Salton, and M. McGill, "Introduction to modern information retrieval". New York, NY, USA: McGraw Hill. 1983.

[39] M. Tomaiuolo, "dDelega: Trust Management for Web Services". International Journal of Information Security and Privacy (IJISP), 7(3), 53-67, 2013. ISSN:1930-1650. doi:10.4018/jisp.2013070104.

[40] S. Wasserman, and K. Faust, "Social network analysis: methods and applications". Cambridge, UK: Cambridge University Press. 1994.

[41] S. Yoshida, K. Kamei, T. Ohguro, and K. Kuwabara, "Shine: a peer-to-peer based framework of network community support systems," Computer Communications, vol. 26 no. 11, pp. 1199-1209, 2003.

[42] B. Yu, and M Singh, "Searching social networks," in Second international joint conference on Autonomous agents and multiagent systems, NY, USA, 2003, pp. 65-72.