

Supporting Social Networks with Agent-Based Services

Enrico Franchi, Agostino Poggi, Michele Tomaiuolo

Department of Information Engineering, University of Parma, Italy

ABSTRACT

Current approaches to build social networking systems are based on a centralized architecture because it allows a simple browser-based user experience and makes easier and more efficient to implement many algorithms used in a social networking site (e.g., friend suggestion). However this kind of architecture has many drawbacks for its users, e.g., lack of privacy, lack of anonymity, risks of censorship and operating costs. This paper presents a system, called Blogracy, which uses widespread and stable peer-to-peer technologies, such as distributed hash tables and BitTorrent, for coping with intrinsic defects of centralized architectures and for being the basis of solid distributed social networking platforms. Moreover, Blogracy takes advantages of multi-agent systems for simplifying the implementation of social network services in a decentralized setting.

KEYWORDS

Social Network, Multi-Agent Systems, Peer-to-peer Computing, Information Retrieval, Information Pushing.

INTRODUCTION

Current approaches to build social networking systems are based on a centralized architecture, because such an architecture allows a simple browser-based user experience and makes easier and more efficient to implement many algorithms used in a social networking sites (e.g., friend suggestion). However this kind of architecture has many drawbacks for its users, e.g., lack of privacy, lack of anonymity, risks of censorship and operating costs. Moreover, it makes social networking platforms vulnerable to many kind of attacks: masquerading, which occurs when a user disguises his identity and pretends to be another user, unauthorized access; denial of service; repudiation, eavesdropping; alteration of data; copy and replay attacks; and, in general, attacks making use of social engineering techniques.

In order to overcome both the intrinsic defects of centralized architectures and the general vulnerabilities of social networking platforms, many different approaches have been proposed, both as federated (i.e., consisting of multiple entities cooperating to provide the service, but usually distinct from users) or peer-to-peer systems (with users directly cooperating to provide the service). This paper presents a system, called Blogracy, which uses widespread and stable peer-to-peer technologies, such as distributed hash tables and BitTorrent, for being the basis of solid distributed social networking platforms, and which uses multi-agent systems for simplifying the implementation of services in a decentralized setting. The next section describes the Blogracy system and show how multi-agent systems have been used for the implementation of a set of pervasive and information retrieval services. The following section presents relate work on decentralized social networks, on the use of multi-agent systems for the development of social networking platforms and on agent-based implementations of information sharing services. Finally, the last section concludes the paper by discussing the main results of our work and by introducing the directions we are following for its improvements.

BLOGRACY

Blogracy is an anonymous and uncensorable microblogging platform, built incrementally over BitTorrent (Cohen, 2003), a popular and resilient file-sharing service. The architecture of the platform is modular and is built around a module for basic file sharing and DHT operations, possibly exploiting an existing implementation, and another module providing a set of social services to the local user through a Web interface. Moreover, the platform provides two additional agent based modules respectively providing a set of pervasive services and a set of information retrieval and pushing services. In particular, the current prototype of Blogracy takes advantage of;

- Vuze (Vuze, 2013), a popular BitTorrent client implemented in Java and available as open source software, for implementing the file sharing and DHT operations.
- Open Social (OpenSocial and Gadgets Specification Group, 2013), a set of APIs supporting the sharing of social data, for implementing the social services.
- JADE (Bellifemine et al., 2008), probably the most known agent development environment enabling the integration of agents and both knowledge and Internet-oriented technologies, for implementing the agent-based services. Figure 1 shows the Blogracy system architecture.

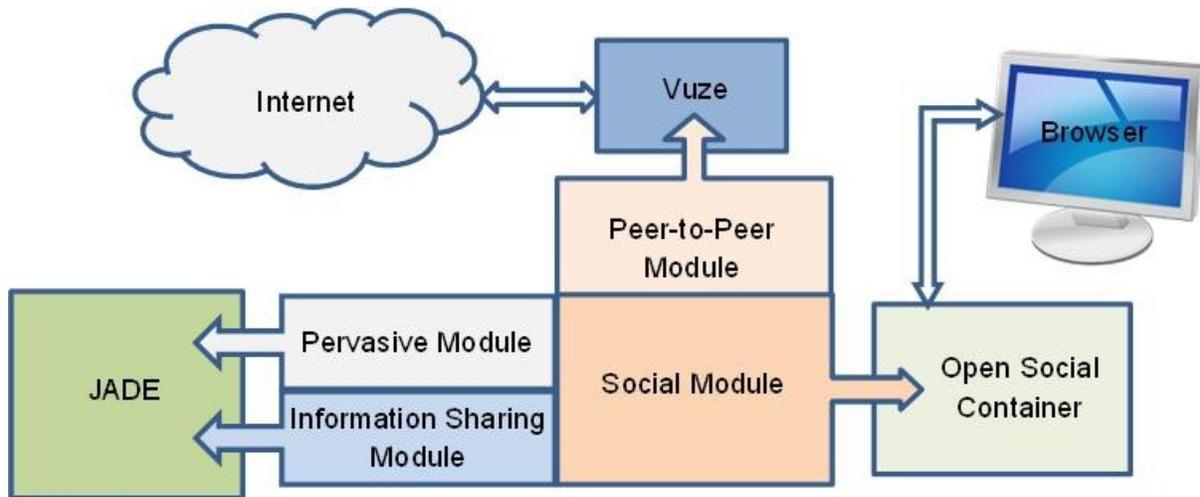


Figure 1. System architecture.

Peer-to-Peer Module

For its basic operation, Blogracy exploits a peer-to-peer file-sharing mechanism and two logically separated DHTs. Users in Blogracy have a profile and a semantically meaningful activity stream, which contains their actions in the system (e.g., add a post, tag a picture, comment a video). One DHT maps the user's identifier with his activity stream, which also contains a reference to the user's profile and references to user generated content (e.g., posts, comments). These references are keys of the second DHT, which are then resolved to the actual files. The files are delivered using the underlying peer-to-peer file-sharing mechanism.

Among the features of public online information systems, and in particular in the case of micro-blogging and social networking applications, anonymity or pseudonymity are often a requirement. But, even under anonymity or pseudonymity, users' content need to be verified for authenticity and integrity. Blogracy uses a key-based identity scheme (Li, 2000), where a user's public key is used directly to represent the user. This way, all content produced by the user can be easily verified against his public key, which is also his own main identifier. Moreover, for assuring anonymity at the lower network level, various anonymizing

technologies exist; varying from simple proxies to complex mix-net schemes, and can be integrated into the platform.

For publishing confidential information, accessible only to a restricted circle of contacts, Blogracy supports attribute-based encryption. Similarly to Persona (Baden et al., 2009), Blogracy privacy model uses attribute credentials for protecting access to sensible content, creating a sort of very flexible “circles”, i.e., parameterized roles to be assigned to users for granting a certain set of access rights. The encryption scheme is based on the CP-ABE (Cyphertext-Policy Attribute-Based Encryption) protocol (Bethencourt et al., 2007).

Once users can be distinguished by their ID, i.e., the hash of their public key, it is also possible to associate additional information with them, including personal profile and personal activity stream. The activities of a user are represented as a flow, which friends and followers are interested into and want to subscribe to. In Blogracy, personal activities are included into an activity stream feed defined on the basis of Activity Streams (Activity Streams, 2013): an open format specification for the syndication of activities taken in social web applications and services. In Blogracy, the personal feed is eventually signed to avoid tampering and then shared using the underlying file-sharing platform.

Social Module

Clearly, an application that does not provide explicit representation for the user’s profile and contacts should not be considered a social networking application. Essentially, in Blogracy users define and manage a list of other users, represented by their IDs. A user is not required to publish his profile, nor the network of his social relations. However, if he does, the profile, containing partial or full information, can be retrieved as any other shared file and its magnet-uri can be also reported in the user’s feed. In case privacy needs to be added, cryptography shall be used. At the current stage, for exporting profiles and contacts, Blogracy adopts Portable Contacts (Portable Contacts, 2013) with OpenSocial extensions, a format which has some benefits from the interoperability point of view, being quite simple and well supported by existing large social networks and mail systems. It also allows to associate tags with each user, thus matching the basic data structure managed by Blogracy.

One of the technical issues of a peer-to-peer microblogging application is data availability. In fact, popular content will quickly gain lots of seeds, while posts published by peripheral users, with few contacts and sparse online presence, will instead suffer poor availability to the extent that it is possible that the publisher remains the only seed for his own new posts. In some systems focused on distributed data storage, like Freenet (Charke et al., 2001), the problem is addressed through multiple replications of all published resources. However, in modern peer-to-peer networks, the hostile behavior of some nodes has to be taken for granted; pollution and other kinds of attacks cannot be underestimated. What we foster, instead, is a replication system based on acquaintances. Essentially, an introducing user is responsible to introduce the invited as smoothly as possible. This kind of mechanisms is thoroughly analyzed in (Douceur & Wattenhofer, 2001; Bernard & Le Fessant, 2009; Rzađca et al., 2010), with special regards to (i) content replication in peer-to-peer storage and (ii) the problem of peers with low availability in completely decentralized systems. In fact, using some kind of fallback strategies for sharing non popular resources may improve the system performance regarding data availability.

Another important issue is interoperability with other existing online social networking and micro-blogging platforms. In principle, since Blogracy handles users’ feeds in the form of activity streams, it can also manage similar feeds obtained in other ways, seamlessly integrating content from web blogs and from the peer-to-peer network. Interoperability with more traditional news-feeds, web-based micro-blogging posts, and content distributed over the peer-to-peer network is thus guaranteed, provided that the stream semantics is correct. On the other hand, resources distributed through Blogracy can be

easily replicated over the web. Since the actual system architecture has a web interface, for user operation, it is relatively simple to host a Blogracy instance on a remote node and configure it for public access, acting as a gateway for Blogracy public content.

Finally, apart from requesting updated feeds at startup, followers should be timely notified that one of their followees updated some resource. Traditionally the strategies are: (i) pull, i.e., the observer periodically checks the observed resource for updates or (ii) push, i.e., the update is automatically announced to the observer. Apart from relying on the DHT, Blogracy benefits from the peer-to-peer messaging facility provided by the file-sharing protocol. In fact, for their basic operation, file-sharing systems need to keep track of the peers that are currently seeding or downloading a certain file (sometimes collectively defined as a “swarm”). So, advertising about a new feed is simply a matter of contacting the peers that are sharing the superseded version of the user's feed.

Pervasive Module

Over the OpenSocial container, Blogracy provides some agent-based services for pervasive online social networking, specifically for realizing locality and proximity groups. For this purpose, each node of the social network will hosts multiple agents, with different levels of agency. Some of the more important agents are (i) the neighborhood manager (NM) agent, which cooperates with lower level agents to discover the users in its neighborhood, (ii) the trust negotiator (TN) agent that is involved in the decisions regarding privacy and data access, and (iii) the OpenSocial agent that provides a bridge towards the underlying Blogracy modules.

A user may own multiple nodes (e.g., an instance on the smart-phone 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 device negotiate which should be considered active (i.e., which one determines the user location): (i) the nodes determine which is the device that registered an explicit user action or (ii) they ask the user to select the device he is currently using.

Apart from the personal circles defined by each user, we also have two additional kinds of groups: (i) proximity groups and (ii) location groups. Proximity groups are centered on each member of the social networking system and represents physical closeness to such 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 in time. Each user configures the sticky-ness of his proximity group, i.e., how long the other users are considered part of it after they are no longer physically close to him. 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 union of other groups, i.e., only “friends” are part of a proximity group). The NM agent informs the OpenSocial 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). It is also associated with the computation node that both identifies and supports the group and with a location profile, which can be either hosted on the central server or on the same computational node. 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 use of a trust negotiation protocol is needed since users joining a proximity or 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 policies, without disclosing unnecessary sensible information, yet establishing trust if possible. This protocol is supported by TN agents that perform the negotiation through a generic library

supporting zero-knowledge proof for attribute verification, which facilitates the creation of trust in similar situations (Tomaiuolo, 2010).

Information Sharing Module

The Blogracy system itself relies only on users' nodes for its operation. Thus users need to perform background tasks on their own, in a distributed way. On the basis of the experience gained developing AOIS (Poggi & Tomaiuolo, 2013), we are integrating a layer of autonomous agents into the system, for assisting the user in finding new interesting content and connections and for pushing the local user's activities to followers.

The exchange of information among the users and the agents of the system is driven by the use of both search indexes and ontologies. The search index allows the ranking of information on the basis of the terms contained in a query. The ontology allows to identify additional information on the basis of the terms contained in the ontology that have some semantic relationships (i.e., synonyms, hyponyms, hypernyms, meronyms and holonyms) with the terms contained in the query.

To do it, we adapted Nutch (Apache Foundation, 2013), an open source web-search software, for searching the local repository. It has been done because it is very easy to develop Nutch plugins for extending its capabilities (we used this feature for using its term extraction module for building the topic ontologies) and because is available a Nutch plugin, that extends keywords based search through the use of OWL ontologies (W3C Consortium, 2009). This plugin receives the terms extracted from the information to be indexed by the Nutch software. Then, accessing the WordNet lexical database (Miller, 1995; Princeton University, 2013) through the use of the JAWS Java software library (Southern Methodist University, 2013), for each term it identifies the top terms of the ontology and the other terms extracted from the information that have some semantic relationships (i.e., synonyms, hyponyms, hypernyms, meronyms and holonyms). At the end of this process, all the terms that have a semantic distance greater than the one fixed by the user are removed and then the WordNet ontology is saved as an OWL file.

Each agent platform acts as a “peer” of the system and is based on four main agents: a personal assistant (PA), a repository manager (RM), an information finder (IF), and an information pusher (IP).

A PA monitors the local user's actions in the platform and learns the user's profile, beyond information provided explicitly. This agent receives the user's queries, forwards them to the available information finders and 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. In fact, as a social network becomes larger and more richly interconnected, users unavoidably face some form of information overflow. A personal agent, on the basis of a user's profile, can arrange presented data in a way to give evidence to the most interesting bits.

An RM manages both the search indexes and the ontology associated with a repository of documents. In particular, a user can ask its RM to create a repository for a specific topic indicating the set of terms (named ontology top terms). The RM creates the repository and an associated ontology containing the top terms and the semantic relationships (i.e., synonyms, hyponyms, hypernyms, meronyms and holonyms) among them. Each time the user adds a document in the repository, the RM populates the ontology with the terms extracted from the document that satisfy a semantic distance constraint with a least one of the ontology top terms. Moreover, when the PA of a friend of its user sends a query for searching information in a specific repository, the RM uses the keyword of the query for populating the ontology. Of course, the user has the possibility of directly modifying the ontology and can share and merge her/his ontology with the ones of her/his friends.

An IF is an agent that searches information on the repositories contained into the node where it lives. It provides this information both to its user and to her/his friends. An IF receives users' queries, finds appropriate results and filters them on the basis of its user's access policies. In particular, when it receives a query from the PA of a friend of its user sends it a query for searching information in a specific repository, then it checks if the querying user has the access to at least a part of the information about the topic stored in the corresponding topic repository, and, if it happens, searches the information on the basis of both the received query and a set of additional queries obtained by replacing each keyword of the received query with the possible substitutes contained in the topic ontology, orders the results and then sends the results to the remote PA. Moreover, it sends the received query to the local RM and IP agents to respectively update the repository ontology and for updating the interests of the profile of the sender of the query.

An IP is an agent that monitors the changes in the local repository and pushes the new information to the PA of interested subscribers who are currently connected. The IP can forward content produced both by the local user and by remote acquaintances to other contacts, according to privacy preserving policies and to recent queries made by other users. In particular, when a user adds information in a specific topic repository, then the RM informs the IP about the new information and the IP checks if such new information satisfy some queries maintained in the profiles of the remote users and if it happens, then the IP either sends such information to the PA of the remote user (if the corresponding Blogracy platform is alive) or maintains such an information until such a platform becomes alive again. Moreover, when an IF receives a query from a friend of its user, it propagates the query to the local IP. In this case, the IP sends the same query to the friends of its user that are not friends of the sender of the query. If some of them have further information for the query, then it informs the PA of the sender about them, and finally the PA presents to her/his user the list of users that might have interesting information.

RELATED WORK

Various solutions are being proposed to overcome the centralized architecture of the most widespread social networking platforms. Many of these proposals follow a federated approach, allowing users registered on a certain server to create relationships with users of other servers. Others are full-fledged peer-to-peer systems, usually based on a distributed hash table (DHT).

Among the federated social networking systems, two of the best known are Diaspora (Diaspora, 2013) and StatusNet (StatusNet, 2013). Diaspora servers communicate by means of an ad-hoc federation protocol and the standard Salmon protocol for comments. Users can (i) participate in the network by setting up their own server, which is named a "pod", or (ii) exploit already existing pods. StatusNet (formerly known as Laconica) adheres to the OStatus standard protocol (W3C Consortium, 2013) for the interconnection of various servers. Using a number of existing protocols, StatusNet shows quite strong interoperability with other networks. Concerning the completely distributed solutions, their origin can be traced back to Freenet (Charke et al., 2001), which is meant as a distributed, cooperative, uncensored and secure file system. It uses a "best-effort" unreliable routing algorithm to find content and namespaces, over both "OpenNet" and "DarkNet" connections. Various quite popular uncensored forums are built on Freenet, but they usually suffer a large amount of spam coming from anonymous sources. Similarly to other more recent systems, Freenet uses an anonymizing layer based on a variation of Chaum's mix-net scheme (Chaum, 1981).

Specifically in the field of social networking, various systems are being developed on the basis of peer-to-peer communications and DHT indexing. Among these, PeerSoN (Buchegger et al., 2009), Persona (Baden et al., 2009) and Safebook (Cutillo et al., 2009) are the most interesting.

PeerSoN is a system designed to provide encryption, decentralization and direct data exchange in the field of social networks. The first prototype of PeerSoN is designed around a PKI, though some studies are being conducted for weakening this assumption. Each user has a unique ID, possibly computed as a hash of the user's email. The DHT is used to trace the user's network presence. An index file, containing a list of new content generated by the user, is also registered in the DHT.

Persona is designed as a set of social networking services. It uses an interesting Attribute-Based Encryption protocol for protecting access to users' content. It allows each user to create various groups of "friends", by assigning proper attribute credentials. Content can then be associated with a publication policy and made available only to a restricted audience.

Safebook is based on a DHT and a network of socially close peers, defined Matryoshka. Peers in a user's Matryoshka are trusted and support the user by anonymizing communications and replicating content and profile information. Safebook exploits a more traditional certification authority. In fact, a user's public key cannot be calculated from his identity, and all public/private key pairs are generated locally by the peers.

Multi-agent systems have always been considered one of the most important ingredients for the development of distributed information management systems and for providing the different services needed in such systems (Klusck, 2001). In particular, several interesting works demonstrate: how multi-agent systems are a suitable means for the management of information in a community of users, how they can take advantage of a peer-to-peer network for performing a distributed search of information and how the use of ontologies and user profile allows an improvement of the quality of their work.

DIAMS is a multi-agent system that provides services for users to access, manage, share and learn information collaboratively on the Web (Chen et al., 2000). DIAMS can be considered one of the most complete multi-agent infrastructures for the management and retrieval of information in a community of users. In fact, it supports the searching and retrieval of the information from local and/or remote repositories and it encourages the collaboration among its users by supporting the sharing and exchange of information among them.

ACP2P (Agent Community based Peer-to-Peer) is an information retrieval system that uses agent communities to manage and search information of interest to users (Mine et al., 2004). In the ACP2P system, an agent works as a delegate of its user and searches for information that the user wants by coupling the typical propagation of the query on the peer-to-peer infrastructure. It supports the community with the identification of the agents that may have such information through the use of the experience gained in its previous interactions. The experimental results of the use of the ACP2P system demonstrated that the use of the agent experience provides a higher accuracy in retrieving information.

CinemaScreen is a recommender system, which combines collaborative filtering and content-based filtering (Salter & Antonopoulos, 2006). The first method requires matching a user with other users with similar behaviors and interests. The second method requires matching the items on the basis of their characteristics (CinemaScreen, in particular, deals with genres, actors, directors etc.). While both mechanisms exhibit weaknesses in particular situations, their combination allows better performances since the very beginning of the system activity. The system is built in the form of an intelligent agent, but apparently it is modeled as an essentially centralized system.

On the other hand, pSearch is a decentralized information retrieval system (Tang & Dwarkadas, 2003). In this system, which is P2P but non-flooding, document indices are distributed through the network according to a classification of document content. The document semantics is generated and managed through a technique called Latent Semantic Indexing (Wiemer-Hastings, 1999). The resulting system is proven to be efficient in the number of nodes to contact to perform a search.

In (Hotho et al., 2006) a social resource sharing system is presented. In this case, it uses a form of lightweight knowledge representation, called folksonomy. In fact, the conceptual structures of 'taxonomy' are created bottom-up by 'folks', thus creating an emergent semantics, instead of using the more rigid approach of the traditional Semantic Web

Sanchez and his colleague proposed an integrated agent-based ontology-driven multi-agent system that automatically retrieves Web pages that contain data relevant to the main concepts of a specific domain (Sanchez et al., 2006). The multi-agent system is based on the use of a Web-based ontology learning method able to automatically build ontologies for any domain (Moreno et al., 2004), and then on a set of agents that use such ontologies for the retrieval, filtering and classification of information.

Social networks and multi-agent systems share both the structure and the scope, since they are composed of individuals connected with some kinds of relationship and they are realized for accomplishing individual and/or common goals. A multi agent system is a system composed of multiple interacting agents; therefore, it is natural to think about synergies between social network and multi-agent system research and application. In fact, multi-agent system models, techniques and technologies have been used and have important potentialities for the study of social networks and the development of social network models (Franchi & Poggi, 2012).

ReferralWeb is an agent based interactive system for reconstructing, visualizing, and searching social networks on the World-Wide Web, whose main focus is selecting an expert of a given field in one's (extended) social network (Kautz et al., 1997a; Kautz et al., 1997b). In ReferralWeb a social network is modeled 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 documents publicly available on the Web, e.g., home pages, co-authorship in published papers or organization charts in institutional websites. 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 and providing a maximum social radius (the number of "links" in the chain connecting the person performing the query with the expert); iii) request a list of documents written by people "close" to a given expert.

Yenta is a matchmaking system that helps people with similar interests to get in touch (Foner, 1997). Yenta agents do not query the web; instead, they scan user's emails, Usenet posts and (possibly) documents in order to discover their users' interests and hobbies. The idea is that many potentially interesting people do not publicly write and are consequently invisible to tools relying on public data. Collected data are then used to introduce users' to each other. Considering that in the nineties web communities were built around the idea of common interests rather than personal acquaintance, the system was a truly distributed social networking system for the time.

Community Organizer is a system where agents help the users by gathering and exchanging information, visualizing contexts, and recommending or assisting their users in making a choice (Hattori et al., 1999). Each user has a personal agent and a set of additional community agents have the function of providing shared information, knowledge, or contexts within the community and act as mediators for informal communications between people. In particular, each personal agent acquires the user profile and visualizes potential communities around the user. The community agent collects the user profiles and maintains information on potential communities. Upon a request from a personal agent, the community agent first computes potential communities around the owner of the personal agent, and then sends the necessary data (users in the potential communities and their relevance) to the personal agent.

Shine is a fully peer-to-peer framework for network community support (Yoshida et al., 2003). The framework also provides design guidelines and enables different applications to share program components and cooperate and 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. 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, such as receiving a message from another agent, user input or notification about a change in the community.

MARS is a multi-agent referral system that finds experts on the basis of personal agents able to learn the user's preferences and interests and able to build an expertise model of the other users on the basis of their responses (Yu & Singh, 2003). The expertise model is captured through a classical vector space model (Salton & McGill, 1983) and each personal agent maintains the models of its neighbors. In particular, the model is updated by a personal agent on the basis of the responses of its neighbors. The response can be an answer of its user or a referral: if the agent is reasonably confident about the expertise of its user matches the query, it directly answers; otherwise, it suggests a referral to another personal agent. Since the number of neighbors is bounded, some of them will be discarded to make place for new ones.

MAGNet, is a middleware based on software agent technology that enables social networking services for users in the mobile network domain (Basuga et al., 2009). The first experimentation showed that its services can enable mobile users to define and customize their social relations with other users, as well as use the created relations to plan and manage group events. MAGNet is built taking advantage of both JADE (Bellifemine et al., 2008) and FOAF (Brickley & Miller, 2005). 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 (Gursel & Sen, 2009). In particular, each agent is associated with a user and 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 (Lerman et al., 2007); 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.

CONCLUSIONS

In this paper we proposed Blogracy a novel peer-to-peer social networking platform that leverages existing, widespread and stable technologies such as DHTs and BitTorrent. Although the primitives offered by those technologies were created with other goals in mind, however, they could be used with minor modification in our system. In particular, we introduced a key-based identity system, a model of social relations for distributing resources efficiently among interested readers and a set of agent-based services for semantic information retrieval and pushing. Such agent-based services show how social network services can be simple to be implemented also in a decentralized setting and are the most important component that distinguishes Blogracy from precedent works on social networks that are based on a decentralized architecture and/or take advantages of agents.

Considering individual security threats, Blogracy is designed to show resilience against censorship and centralized control over published data. Its completely distributed architecture and the replication of popular data, typical of file sharing systems, provide also resistance against DoS attacks. Moreover, traditional signatures and timestamps, together with CP-ABE, are used as means to protect against eavesdropping, alteration, copy and replay, repudiation, unauthorized access. Conversely, it opposes limited resistance against masquerading and some form of social engineering, exactly for the absence of any centralized control.

Up to now, Blogracy system was only experimented in some “artificial” communities involving researchers and students of our University. However, the results of the experimentation encouraged us in continuing its development. Besides to continue its experimentation and validation, current activities are devoted to: i) the improvement of the current agent-based services and the addition of new ones, ii) the development of a client for Android mobile devices, and iii) the development of a set of tools for simplify the management of the ontologies associated with the information repository. Moreover, given that the performance limitations of P2P direct messaging, we are also considering the creation of multicast tree over DHT based networks. Similar solutions have been developed only over Pastry (Castro et al., 2002) and not yet over Kademlia (Maymounkov et al., 2002).

REFERENCES

- Activity Streams (2013). Activity Streams Documentation. Retrieved July 3, 2013, from <http://activitystrea.ms>.
- Apache Foundation (2013). Nutch Software. Retrieved July 3, 2013, from <http://nutch.apache.org>.
- Baden, R., Bender, A., Spring, N., Bhattacharjee, B., & Starin, D. (2009). Persona: an online social network with user-defined privacy. In *ACM conference on Data communication* (pp. 135-146), Barcellona, Spain: ACM.
- Basuga, M., Belavic, R., Slipcevic, A., Podobnik, V., Petric A., & Lovrek, I. (2009). The MAgNet: Agent-based Middleware Enabling Social Networking for Mobile Users. In *10th International Conference on Telecommunications* (pp. 89-96), Zagreb, Croatia: IEEE.
- Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: a Software Framework for Developing Multi-Agent Applications. Lessons Learned. *Information and Software Technology Journal* 50, 10-21.
- Bernard S., & Le Fessant, F. (2009). Optimizing peer-to-peer backup using lifetime estimations. In *2009 EDBT/ICDT Workshops* (pp. 26-33), New York, NY, USA: ACM.
- Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-Policy Attribute-Based Encryption. In *IEEE Symposium on Security and Privacy* (pp.321-334), Oakland, CA, USA: IEEE.
- Brickley, D., & Miller, L. (2005). FOAF Vocabulary Specification 0.98. Retrieved July 3, 2013, from <http://xmlns.com/foaf/0.1>.
- Buchegger, S., Schiöberg, D., Vu, L., & Datta, A. (2009). PeerSoN: P2P social networking: early experiences and insights. In *Second ACM EuroSys Workshop on Social Network Systems* (pp. 46–52), Nuremberg, Germany: ACM.
- Castro, M., Druschel, P., Kermarrec, A. M., & Rowstron, A. I. (2002). SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *Journal on Selected Areas in Communications*, 20(8), 1489-1499.
- Chaum, D.L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), 84–90.
- Chen, J.R., Wolf, S.R., & Wragg, S.D. (2000). A Distributed Multi-Agent System for Collaborative Information Management and Sharing. In *9th ACM International Conference on Information and Knowledge Management* (pp. 382-388), McLean, VA, USA: ACM.
- Clarke, I., Sandberg, O., Wiley, B., & Hong, T. (2001). Freenet: A distributed anonymous information storage and retrieval system. In H. Federrath (Ed.), *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability* (pp. 46-66), New York, NY, USA: Springer.

- Cohen, B. (2003). Incentives build robustness in BitTorrent. In *First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA.
- Cutillo, L.A., Molva, R., & Strufe, T. (2009). Safebook: a Privacy Preserving Online Social Network Leveraging on Real-Life Trust. *IEEE Communications Magazine*, 47(12), 94-101.
- Diaspora (2013). Diaspora Social Network Web Page. Retrieved July 3, 2013, from <https://joindiaspora.com>.
- Douceur, J., & Wattenhofer, R. (2001). Competitive hill-climbing strategies for replica placement in a distributed file system, In J. Welch (Ed.), *Distributed Computing, Lecture Notes in Computer Science, volume 2180* (pp. 48–62), Berlin, Germany: Springer.
- Foner, L. (1997). Yenta: a multi-agent, referral-based matchmaking system. In *First International Conference on Autonomous Agents* (pp. 301-307), Marina del Rey, CA, USA: ACM.
- Franchi, E., & Poggi, A. (2012). Multi-Agent Systems and Social Networks. In M.M. Manuela Cruz-Cunha, G.D. Putnik, N. Lopes, P. Gonçalves, E. Miranda (Eds.), *Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions* (pp. 84-97), Hershey, PA, USA: Information Science Reference.
- Gursel, A., & Sen, S. (2009). Improving Search In Social Networks by Agent Based Mining. In *21st international Joint Conference on Artificial Intelligence* (pp. 2034-2039), San Francisco, CA, USA: Morgan Kaufmann.
- Hattori, F., Ohguro, T., Yokoo, M., & Matsubara, S. (1999). Socialware: Multiagent systems for supporting network communities. *Communications of the ACM*, 42(3), 55-61.
- Hotho, A., Jäschke, R., Schmitz, C., & Stumme, G. (2006). Information Retrieval in Folksonomies: Search and Ranking. In *The Semantic Web: Research and Applications, Lecture Notes in Computer Science, volume 4011* (pp. 411-426), Berlin, Germany: Springer.
- Kautz, H., Selman, B., & Shah, M. (1997a). Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3), 63-65.
- Kautz, H., Selman, B., & Shah, M. (1997b). The Hidden Web. *AI Magazine*, 18(2), 27-36.
- Klusck M (2001). Information Agent Technology for the Internet: A survey. *Data & Knowledge Engineering*, 36(3), 337-372.
- Lerman, K., Plangrasopchok, A., & Wong, C. (2007). Personalizing results of image search on Flickr. In *AAAI workshop on Intelligent Techniques for Web Personalization* (pp. 65-75), Vancouver, Canada: AAAI Press.
- Li, N. (2000). Local Names in SPKI/SDSI. In *13th IEEE workshop on Computer Security Foundations* (pp. 2–15), Cambridge, UK: IEEE.
- Maymounkov, P., & Mazieres, D. (2002). Kademia: A peer-to-peer information system based on the xor metric. In P. Druschel, F. Kaashoek, & A. Rowstron (Eds.), *Peer-to-Peer Systems, Lecture Notes in Computer Science, volume 2429* (pp. 53-65). Berlin, Germany: Springer.
- Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39-41.
- Mine, T., Matsuno, D., Kogo, A., & Amamiya M. (2004). Design and implementation of agent community based peer-to-peer information retrieval method. In *Cooperative Information Agents VIII, Lecture Notes in Computer Science, volume 3191* (pp. 31-46), Berlin, Germany: Springer.

Moreno, A., Riano, D., Isern, D., Bocio, J., Sanchez, D., & Jimenez, L. (2004). Knowledge Exploitation from the Web. In *Practical Aspects of Knowledge Management, Lecture Notes in Computer Science, volume 3336* (pp. 175-185), Berlin, Germany: Springer.

OpenSocial and Gadgets Specification Group (2013). Open Social Specifications. Retrieved July 3, 2013, from <http://docs.opensocial.org>.

Poggi, A., & Tomaiuolo, M. (2013). A DHT-Based Multi-Agent System for Semantic Information Sharing. In C. Lai, G. Semeraro, E. Vargiu (Eds.), *New Challenges in Distributed Information Filtering and Retrieval, Studies in Computational Intelligence, volume 439* (pp. 197-213), Berlin, Germany: Springer.

Portable Contacts. (2008). Portable Contacts Specifications. Retrieved July 3, 2013, from <http://portablecontacts.net/draft-spec.html>.

Princeton University (2013). Wordnet Software. Retrieved July 3, 2013, from <http://wordnet.princeton.edu>.

Rzadca, K., Datta, A., & Buchegger, S. (2010). Replica placement in P2P storage: Complexity and game theoretic analyses, In *International Conference on Distributed Computing Systems* (pp. 599–609), Genova, Italy: IEEE.

Salter, J., & Antonopoulos, N. (2006). CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. *IEEE Intelligent Systems*, 21(1), 35-41.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval*. New York, NY, USA: McGraw Hill.

Sanchez, D., Isern, D., & Moreno, A. (2006). Integrated Agent-Based Approach for Ontology-Driven Web Filtering. In *Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, volume 4253* (pp. 758-765), Berlin, Germany: Springer.

Southern Methodist University (2013). JAWS Software. Retrieved July 3, 2013, from <http://lyle.smu.edu/~tspell/jaws>.

StatusNet (2013). StatusNet Social Network Web Page. Retrieved July 3, 2013, from <http://status.net>.

Tang, C., Xu, Z., & Dwarkadas, S. (2003). Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *ACM SIGCOMM* (pp. 175–186), Karlsruhe, Germany: ACM.

Tomaiuolo, M. (2010). Trusted computing or distributed trust management? In I. M. Portela, M. M. Cruz-Cunha (Eds.), *Information Communication Technology Law, Protection and Access Rights: Global Approaches and Issues* (pp. 397- 415) Hershey, PA, USA: IGI Global.

Vuze (2013). Vuze Software. Retrieved July 3, 2013, from <http://www.vuze.com>.

W3C Consortium (2009). OWL 2 Web Ontology Language Overview. Retrieved July 3, 2013, from <http://www.w3.org/TR/owl2-overview>.

W3C Consortium (2013). OStatus Community Group. Retrieved July 3, 2013, from <http://www.w3.org/community/ostatus>.

Wiemer-Hastings, P.M. (1999). How Latent is Latent Semantic Analysis? In *16th international Joint Conference on Artificial Intelligence* (pp. 932-941), San Francisco, CA, USA: Morgan Kaufmann.

Yoshida, S., Kamei, K., Ohguro, T., & Kuwabara, K. (2003). Shine: a peer-to-peer based framework of network community support systems. *Computer Communications*, 26(11), 1199-1209.

Yu, B., & Singh, M. (2003). Searching social networks. In *Second International Joint Conference on Autonomous Agents and Multi Agent Systems* (pp. 65-72), Melbourne, Australia: ACM.