A Peer to Peer Architecture for a Distributed User Model

Fotis Paraskevopoulos and Gregoris Mentzas

National Technical University of Athens
{fotisp,gmentzas}@mail.ntua.gr

Abstract. The current existence and continuous introduction of new social networks, provide a constant growth of the available information which can be used to create richer user profiles. Due to the disparity of this information centralized approaches to generate user models result in complex systems which can adapt with difficulty to the constant reshaping of information. For this reason there is a trend to create decentralized approaches for distributed user models. There are still two apparent constraints in these, one is the flexibility of the schema used to distribute the user model, and the other is the need for a centralized entity which facilitates communication and interoperability. We propose a peer to peer architecture to achieve a distributed user model. Each peer in the system acts as a standalone user model, it extracts and translates information into pre-defined templates, which are used to ensure interoperability when communicating with other peers, through an internal communication broker.

1 Introduction

Currently there is a plethora of social networks available, and new ones are being introduced constantly. The information found in these, range from information the users share to that generated from the interaction of the user with the system. Information from a single social network (referred as to *information source* from this point on) may be abundant enough to generate rich user profiles, which can be used for personalization services, recommendation services, etc. There is however an advantage in merging information from multiple information sources, to create richer user profiles [1].

In order to cope with this multitude of information sources, user modeling approaches evolved from centralized monolithic ones such as GUMO [8], GeniUS [6], UserML [7] to decentralized ones. A basic constraint of decentralized approaches is the interoperability of the information exchanged between parts of the system [3,4]. This is currently handled either by introducing a central component which handles the information exchange [1][10][9], or by homogenizing the schema to which the information is mapped [2].

In this paper we propose a peer to peer architecture to provide a fully decentralized system to generate and maintain distributed user models. Each peer in the system is made up of a User Model Translator(UMT) which encompasses the necessary logic to extract and translate information from a social network into *key:value* pairs which are grouped into *Templates*. *Templates* are uniquely identifiable by a Universal Unique Identifier(UUID) which is a hash generated from the combination of the *key* parts of the *key:value* pairs. Peers communicate through a *Broker* which can exchange information between any peers in the system provided they support *templates* with matching UUIDs. Our vision is that for any number of information sources available, there will be a matching number of peers.

The structure of this paper is as follows. In section 2 we provide a brief survey of the related work. In section 3 we describe our approach in detail. In Section **??** we go over a "Proof of Concept" which we developed in order to examine the feasibility of such system. Section 4 concludes this paper by going over our planned future work.

2 Related Work

In this section we will go over several of the previous approaches to create decentralized distributed user models.

In [1] the authors introduce a framework which can achieve interoperability of a distributed model via a *centralized* server called *Mypes*, where a single vector based user model is built per user by aligning the information from several information sources using hand crafted rules called *alignments*. PersonisAD [2] is defined as a distributed, active, *scrutable model*, which is capable of gathering information from different sensors about different users and combine their preferences to provide a richer experience. MobiTribe [10] is a system which focuses on mobile devices and its advantage is the fact that the the model itself is distributed in nature. The exchange of information between the devices and applications using the user modeling information is mediated by a centralized content management system. In [5] the authors present a vision of a distributed user model by creating single function stand alone agents which are responsible for storing a single attribute of a holistic vector based user model. Finally in [9] the authors present a decentralized architecture for sharing and re-using life logs from different systems. These are gathered by agents, which then forward the information to a centralized broker.

3 The P2P Distributed User Model

We propose a *P2P* architecture to generate and maintain a distributed user model, which uses pre-defined information exchange templates to communicate the user model. The information provision and exchange is controlled by standalone peers which contain the necessary logic to process, and store the information from a single information source.

Each peer is self contained and can act as stand-alone user model which only handles information from a single source. The objective however is to leverage information from multiple sources, without the need to handle the workload of extracting, storing and processing this information individually. A peer not only extracts information, but it also translates it into pre-determined templates, in order to communicate it with other peers in the system. A peer can act either as a producer, a consumer or both, depending on its desired effect on the entire system.

A peer in the system is composed of three parts, the User Model Translator(UMT), the *Templates* it supports, and its internal *Broker*.

The UMT needs to be developed individually for each peer, and contains the necessary logic to extract information from a particular information source, and consequently translate it into *key:value* pairs for each type of information like in the following example {*name: John*},{*lastname:Smith*} [1].

By translating the information into a basic exchangeable format, it can be grouped to correspond to any of the *Template*s which the agent supports. A *Template* is the specifications of a group of *key:value* pairs. The previous *key:value* examples could be part of the "Basic Profile" template. Each template is identified by a Universally Unique Identifier (UUID) which is generated using the *key* part of the *key:value* pairs. The process to generate the UUID starts by concatenating the string values of templates' *keys* into a single string and subsequently we generate a hash of that value. The resulting hash is the UUID of the template and it determines its *syntactical signature* and uniqueness, not its semantic one. There is no requirement for any peer in the system to support the exact same types or number of templates. Any peer can introduce new templates into the distributed model, which are defined by their UMT. These can be discovered and supported by other peers, through the internal *Broker*.

We need to determine the Level of Interoperability(LI) between any two peers. The *LI* between any **n** number of peers is determined by the number of matching *template UUIDs* they support. The *LI* is determines whether two peers can communicate and exchange information at all.

Each peer contains its own *broker*. Through the *broker* a peer can determine the *LI* between other peers, the list of available templates in each peer, and the rules and pre-conditions in place to access the information from a particular peer. Not all *templates* of a peer are publicly accessible, only those that have been specified in the broker configuration.

Currently there is no *LI* threshold set in place to determine whether any two peers can communicate, but rather the *brokers* themselves analyze the lists of *UUID*s which are exchanged during the initial handshaking process. It is possible for *brokers* to request the syntax of a *template* pertaining to a UUID which it does not currently support.

4 Future Work

In this paper we presented a P2P architecture to achieve a decentralized distributed user model. We believe this approach allows for the continuous expansion of the user model, in the form of a genealogy of peers. The information exchanged between peers can be further reused to create connections and inferences which are not immediately present from a single information source, without the constraints and overhead of extracting and analyzing the information from multiple information sources by a single pre-defined non-extensible service.

There are two aspects of the peer which need to be extended, the first is to provide a method by determining the emphsemantical signature of the templates, and the second is to extend the brokers contain the necessary functionality to enforce cross-boundaries policies, and expiration policies to ensure the privacy of the users information[11].

We have deployed a live version of the P2PUM which has four active peers in p2pum.imuresearch.eu.

References

1. Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. User Modeling and User-Adapted Interaction 23(2-3), 169–209 (Apr 2013)
2. Assad, M., Carmichael, D.J., Kay, J., Kummerfeld, B.: PersonisAD: distributed, active, scrutable model framework for context-aware services. In: LaMarca, A., Langheinrich, M., Truong, K.N. (eds.) Pervasive Computing, pp. 55–72. No. 4480 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2007)
3. Carmagnola, F.: From user models to interoperable user models. AI Communications 22(2), 121–123 (2009)
4. Carmagnola, F., Cena, F., Gena, C.: User model interoperability: a survey. User Modeling and User-Adapted Interaction 21(3), 285–331 (Aug 2011)
5. Dim, E., Kuflik, T.: User models sharing and reusability: a component-based approach. In: UMAP Workshops (2012)
6. Gao, Q., Abel, F., Houben, G.J.: GeniUS: generic user modeling library for the social semantic web. In: Pan, J.Z., Chen, H., Kim, H.G., Li, J., Wu, Z., Horrocks, I., Mizoguchi, R., Wu, Z. (eds.) The Semantic Web, pp. 160–175. No. 7185 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2012)
7. Heckmann, D., Krueger, A.: A user modeling markup language (UserML) for ubiquitous computing. In: Brusilovsky, P., Corbett, A., Rosis, F.d. (eds.) User Modeling 2003, pp. 393–397. No. 2702 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2003)
8. Heckmann, D., Schwartz, T., Brandherm, B., Kröner, A.: Decentralized user modeling with UserML and GUMO. In: PROCEEDINGS OF THE WORKSHOP ON DECENTRALIZED, AGENT BASED AND SOCIAL APPROACHES TO USER MODELLING (DASUM 2005. pp. 61–65 (2005)
9. Iyilade, J., Vassileva, J.: A decentralized architecture for sharing and reusing lifelogs. In: UMAP Workshops (2013)
10. Thilakarathna, K., Petander, H., Mestre, J., Seneviratne, A.: {MobiTribe}: cost efficient distributed user generated content sharing on smartphones. IEEE Transactions on Mobile Computing PP(99), 1–1 (2013)
11. Wang, Y., Kobsa, A.: Privacy in cross-system personalization. In: Intelligent Information Privacy Management Symposium, Stanford, CA (2010)