WEB SERVICES & RECOMMENDER SYSTEMS
A Research Roadmap

Leandro Krug Wives, José Palazzo Moreira de Oliveira
Universidade Federal do Rio Grande do Sul, Brazil

Zakaria Maamar
Zayed University, U.A.E.

Samir Tata, Mohamed Sellami
TELECOM & Management Sud Paris, France

Keywords: Web services, Recommender systems, Research roadmap.

Abstract: In this position paper, we provide a brief overview of Recommender Systems (RS) and Web Services (WS). After, we propose a research roadmap for the challenges and opportunities that could arise following the combined use of WS and RS. While these challenges are expected to hinder this use, we discuss the necessary actions that need to be taken to overcome these challenges and hence, make this use a win-win situation for both WS and RS. We illustrate how the combination of RS to WS takes place in terms of what RS can do for WS and what WS can do for RS. Finally, we conclude by pointing out the actions to take so that this combination turns out successful.

1 INTRODUCTION

The reason of proposing a research roadmap for the Web Services & Recommender Systems association is the following. On the one hand, Web services' adoption is somehow slowed down by various, recurrent issues largely reported in the literature such as the complexity of Web services discovery and Web services reliability (Sapkota 2005; Shen and Su 2007; Margaria 2008). To address these issues, innovative solutions are required and could be built upon different other techniques for instance recommendation (i.e. recommender systems). On the other hand, recommender systems are slowly moving from simple applications (e.g., textbook recommendation) while modern business application's complexity continues to increase. This has, to a certain extent, excluded recommender systems from the list of techniques of choice for the development of these applications, unless the way recommender systems are developed sees some changes. These changes will permit to enhance recommender systems with new capabilities drawn from existing IT technologies for instance Web services.

Web services are paving the way for a new generation of large-scale, loosely-coupled business applications. This is witnessed from the large number of standards and projects related to Web services, e.g., (Bentahar, Maamar et al. 2007; Yu, Bouguettaya et al. 2008), which address a variety of issues such as semantics, high availability, and discovery. These issues, in fact, hinder the smooth automatic composition and deployment of Web services. Composition, which is one of Web services' selling points, handles the situation of a user's request that cannot be satisfied by any single, available Web service, whereas a composite Web service obtained by combining available Web services may be used.

Recommender systems are a special kind of information filtering approaches designed to help users cope with information overload (Adomavicius and Tuzhilin 2005; Werthner, Hansen et al. 2007). These systems can process large volume of information prior to suggesting items to a user according to

Krug Wives L., Palazzo Moreira de Oliveira J., Maamar Z., Tata S. and Sellami M.
WEB SERVICES RECOMMENDER SYSTEMS - A Research Roadmap.
DOI: 10.5220/0002840601190124
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

first, the users' interests or behaviour and second, to other users' past experiences and recommendations (e.g., comments, criticisms, and opinions). This is very appealing when users are overloaded with information and searching for relevant ones turns out a "nightmare". Thanks to recommender systems, the search space could be reduced (in the user's perspective), reducing the number of elements the user has to analyze, providing a positive impact on the complexity and time needed to find relevant information, e.g., (Pu, Chen et al. 2008).

In the remaining of this position paper, we provide a brief overview of Recommender Systems (RS) and Web Services (WS). Afterwards, we illustrate how the combination of RS to WS takes place in terms of what RS can do for WS and what WS can do for RS. Finally, we conclude by pointing out the actions to take so that this combination turns out to be successful.

2 RECOMMENDER SYSTEMS AND WEB SERVICES

According to Werthner et al. (2007), recommender systems are applications that provide qualified advices about products or services that a user might be interested in, though the final word of selecting that product or that service over others goes always back to the user. In addition, RS can assist people make the right decisions when they are swashed with an overwhelming set of alternatives.

There exist several commercial and academic applications that illustrate the use of RS. Examples include Amazon[i], Last.fm[ii], Phoaks[iii] system, and all the Grouplens' projects[iv] (MovieLens, WikiLens, TechLens, etc.). RS can be classified into three categories: content-based, collaborative filtering, and hybrid. This classification takes into account the recommendation model upon which a RS is built. A good survey on RS is proposed in (Adomavicius and Tuzhilin 2005) and (Perugini, Gonçalves et al. 2004).

- Content-based Recommendation: here, a recommendation is performed by analyzing the similarity between (i) items (artefacts) and a user's interests (represented by her profile in terms of interests, level of expertise, and sometimes needs) or (ii) items themselves. The items that are more similar to the user's profile are suggested first. The similarity identification algorithm depends on the characteristics like price, size, and purpose of the item that is being analyzed. In addition, the similarity functions that are used in these algorithms depend on the nature of the item.
- Collaborative Filtering Recommendation: here, users explicitly evaluate items they know or use by giving specific scores to these items (Herlocker, Konstan et al. 2004). Because it is known that users are always reluctant to engage in any scoring exercise (Claypool, Le et al. 2001), other alternatives exist to address this reluctance and rely, for example, on tracking how a user interacted with the system by registering the items she bought, which is similar to what Amazon does. An item could be any object, document, product, or service. The highly evaluated items are recommended to users using neighbourhood-based algorithms. The main idea is that users with similar profiles tend to have same interests. The similarity among items is also important to identify since similar items may be relevant to users with similar interests. The major limitation of collaborative filtering recommendation is that users do not receive recommendations until some evaluation is performed by another user.
- Hybrid Recommender Systems: Melville et al. (2002) state that content-based and collaborative filtering recommendations "fail to provide good recommendations in many situations". Thus, their combination seems to be a good way to address the limitations of each recommendation type. This is where hybrid RS came into play. There exist many approaches to combine RS techniques. Melville et al., for instance, use content-based techniques to identify similarities among rated- and unrated-items, which permits to minimize the lack of evaluation that was mentioned before. Other authors apply different Recommendation Techniques (RT) and combine or choose their better results (Ahmad Wasfi 1999; Torres, McNee et al. 2004; Thio and Karunasekera 2005). While there is not a complete study yet that states which combination approach to go with, existing studies show that hybrid RS provide better results (Adomavicius and Tuzhilin 2005).

Web service as defined by the W3C is "a software application identified by a URI, whose interfaces and binding are capable of being defined, described, and discovered by XML artefacts, and supports direct interactions with other software applications using XML-based messages via Internet-based applications" (W3C 2004). A WS implements a functionality (e.g., BookOrdering) that users and other peers (i.e., Web services) invoke by submitting appropriate messages. WS are modular and loosely-coupled, providing a simple model of programming and deployment of applications through the Web. The life cycle of a WS consists of three steps usually known as definition and announcement, discovery, and invocation. A fourth step, which is about composition, is usually to this life cycle.

- Definition & Announcement Step: independent providers use the Web Service Description Language (WSDL) to describe the functionalities, operations, and attributes of the WS they develop. Afterwards, the providers announce the Web services to the external community by posting their descriptions on various public and private registries (e.g., Universal Description Discovery and Integration (UDDI) (OASIS 2004), Electronic Business using eXtensible Markup Language (ebXML) (OASIS 1999)).
- Discovery Step: finding WS means screening and querying registries in order to match WS' functionalities to service requesters. A query includes search criteria in terms of functional properties (type of service, inputs, outputs, etc.) and sometimes non-functional properties (response time, execution rate, etc.).
- Invocation Step: it is the process of calling the operation of a WS. As software artefacts can be developed using different languages, these artefacts must follow a common contract to interact with other elements in a platform-neutral environment. In the case of WS, the Simple Object Access Protocol (SOAP) is used so that artefacts can be bound to each other. SOAP is a XML-based remote invocation protocol designed for flexibly composing Internet applications.
- Composition Step: when a user's request cannot be satisfied by any single, available WS, the combination of WS constitutes an alternative (Berardi, Calvanese et al. 2003). Examples of specification languages to compose Web services include the Business Process Execution Language (BPEL) and the Web Services Choreography Description Language (WSCDL).

3 WEB SERVICES & RECOMMENDER SYSTEMS COMBINATION

In this section, we propose a research roadmap for the challenges and opportunities that could arise following the combined use of WS and RS. While these challenges are expected to hinder this use, we discuss the necessary actions that need to be taken to overcome these challenges and hence, make this use a win-win situation for both WS and RS.

3.1 What can RS do for WS?

Recommendation seems to offer solutions to the problems of handling large volumes of data (Pu, Chen et al. 2008). Users' profiles, past experiences, and rankings are some of the factors that RS take into consideration in their functioning. With the proliferation of WS-based applications, recommendations would help address some recurrent issues of how to discover WS with reduced efforts, how to suggest/avoid "good/poor" WS based on previous successful/unsuccessful executions, how to compare WS, just to cite some. In this section, we discuss how recommendations could be smoothly woven into the life cycle of a WS.

With the proliferation of trusted and un-trusted WS, their discovery (including registries upon which these WS are posted) constitutes and continues to be complex and tedious, although several improvements are made to the operational mechanisms of registries to address, for example, semantic and security concerns (Trabelsi, Pazzaglia et al. 2006). The discovery process should be performed based on the semantic match between a declarative description of the WS that is being sought, and a description of the WS that is being offered (Paolucci, Kawamura et al. 2002), but this activity is still "complex" and poses problems to allow a broader adoption of WS by businesses. Recommendation could simplify this discovery process by easing and speeding up some of the activities that are related to one of the following three cases:

- Case 1: instead of screening registries, which is the traditional way of discovering WS, a RS could suggest a list of WS that satisfies a user's or client's needs based on previous experiences collected over time. With respect to this user or client, these experiences should feature similar needs, profiles, interests, functionalities, etc. The value-added of RS to this case is to reduce the search space of WS to a specific list of recommended WS, which is the main shortcoming in WS discovery.

- Case 2: in a composition scenario, a RS could suggest additional WS to be included in this scenario based on the components that are already included and the user's profile or based on the composition's functionality. For instance, a delegate attending an overseas conference could be interested in some sightseeing activities, though these activities are not part of the composition scenario associated with this delegate's travel plan. The new component WS will be submitted to the user for approval before inclusion and execution.
- Case 3: to make WS more robust, a RS could suggest peers that would substitute this WS in case of failure. The peers are recommended based on the functional and non-functional characteristics they have in common with the WS to make robust.

WS discovery continues to rely on how they are described. By weaving some Recommender Techniques (RT) into this discovery, WS' descriptions could turn out insufficient, inefficient, and even inappropriate. The WSDL document of a WS uses a number of well-defined, XML-based arguments (e.g., message, port type, binding) that are primarily meant for discovery and invocation purposes. Unfortunately, these arguments cannot sustain the normal functioning of any RS that requires to know, for example, how a WS was rated by users, by whom a WS was recommended in the past, how many times a WS was discovered through recommendation, how many times a user rejected a WS despite positive recommendations and vice versa, etc. As a result, dedicated arguments to describe WS are required and could be related to user/client evaluation (score) over a certain time frame, reputation as defined by users and peers, cases that failed/succeed despite positive/negative recommendations, etc. To monitor and log WS' status and messages would also be helpful to determine its quality and reputation.

Once WS' new descriptions permit meeting the requirements of how RS function, the next step consists of looking into how RTs would affect the announcement of these WS. Some of the aspects to consider include:

- Where are the recommendation details announced? Should these details be posted on existing registries like any regular WSDL detail, or should these details be separated and posted on dedicated registries that would be made accessible to RS only?
- Should all recommendation details be published or just some? In the latter case, what are the details to select?
- When are the announced recommendation details refreshed? Will this be done regularly, continuously, or both? And who is responsible of performing this refreshment?

Another point of what RS can do for WS is about the value-added of recommendation to composition. The purpose here is to promote the reuse of composite WS. Rather than developing composite WS from scratch each user's request is submitted, the user is given the possibility of triggering a composite WS that was developed in the past in response to the requests of other users and hence, was experienced/tested/evaluated over time. This permits to build the history of a composite WS in terms of needs satisfied, problems faced, solutions adopted, and performance assessed. Similarity of needs, profiles, and interests is a pre-requisite to the reuse of composite WS. Because we have pointed out that the descriptions of component WS need extra details that would support the functioning of RS, we expect that composite WS would be subject to the same, namely extra details would be required. These details are for the composition level and could show, for example, competitors of composite WS, performance of composite WS, participating components in compositions, etc.

Recommendation would not be complete without taking full benefits of previously executed composite WS in terms of outcomes produced, obstacles met, exceptions raised, and alternatives adopted. Business Process (BP) mining is commonly used for tracking execution (van der Aalst, Reijers et al. 2007), and is a good source of data for recommendation.

3.2 What can WS do for RS?

We now discuss how current RTs could be subject to enhancement or adaption in response to WS' characteristics. These characteristics show, for example, how WS are described, discovered, invoked, and composed.

Content-based RTs rely on how items to analyze and then to suggest are described in terms of content. When it comes to WS, their descriptors (i.e., WSDL documents) are the content to be analyzed. These documents are XML-based and contain specific metadata about WS. This means that WSDL documents are semi-structured and use a set of predefined tags that contain textual, categorical, and numerical values. The following list shows how the nature of these tags would affect the actions to take during recommendation.

- Compared to textual documents, WSDL documents are more structured and present a strict syntax and semantics. Furthermore, WSDL documents do not have to cope (to a certain extent) with the vocabulary issue (morphological variations, synonymy, etc.) that is usually present on textual documents and difficult identification of similar content. Therefore, it is expected that minor changes in the current content-based RTs would happen. These changes would primarily permit to handle the hierarchical structure and semantics of WS.
- WS do not usually have textual attributes (in the sense of free and unstructured texts). As WS description languages are usually a variation of XML with predefined tags and limited set of attributes, their contents are like semi-structured data and thus more subject to categorical or numerical manipulation techniques. Categorical and numerical attributes are thus, extracted out of the WSDL tags of a WS. These tags could be semantically annotated (using ontology) so that a meaning is given to each tag. To simplify the development of a recommendation model rather than going through the "hassle" of adapting existing RTs, it would be interesting to assign priorities and weights to the different WSDL tags. This would, for instance, help the newly developed recommendation model in ranking WS with respect to different dimensions such as price, availability, stability, etc.

Collaborative-based RTs are based on user collaboration, i.e., users must submit feedback (e.g., opinions, ratings) about the items they use or consume, and the system recommends items based on users with similar interests and opinions. When it comes to WS, RTs can be used to minimize the search space using previous experiences. The refinements or adaptations that must be performed in RTs are related to collecting data about the WS's life cycle and their interaction (sequence, order of use/activation, status, quality), using specific monitors (Cruz, Campos et al. 2004), since users do not directly interact with WS (in fact, most of the time they even do not know they are using WSs).

Hybrid-based RTs must be designed along the following dual goals: deal with WS' descriptors (purpose of content-based techniques) and take into account the way WS interact and are used (purpose of collaborative-filtering techniques). The former goal is related to collecting feedbacks on, for instance, the evaluation (or any other kind of feedback such as the number of activations or uses) of the recommended or used WS, from any independent peer (preferably trusted) or entity involved in overseeing the progress of the life cycle of a WS. The latter goal is related to the lack of details that follows the similarity analysis of descriptions of WS. For this purpose, hybrid techniques would use descriptions of WS and feedbacks that are collected out of the bodies responsible for the monitoring of these WS.

4 SUMMARY

In this position paper, a research roadmap for combining Web services and recommender systems was discussed. This roadmap offered a glimpse of the research opportunities and challenges that such a combination would offer. What Web services can do for recommender systems and the other way around permitted to suggest an overview of first, the solutions that Web services could offer to address the limitations of recommender systems and second, the mechanisms that recommender systems could offer to sustain the adoption of Web services. Doubtless combining Web services and recommender systems should be a win-win situation for both, and real business applications that would benefit out of this combination will just prove this situation.

ACKNOWLEDGEMENTS

This work was partially supported by CAPES/COFECUB project ADContext, and by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil.

REFERENCES

Adomavicius, G. and A. Tuzhilin (2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." IEEE Trans. on Knowl. and Data Eng. 17(6): 734-749.

Ahmad Wasfi, A. M. (1999). Collecting user access patterns for building user profiles and collaborative filtering. Proceedings of the 4th international Conference on intelligent User interfaces Los Angeles, California, United States, ACM, New York, NY: 57-64.

Bentahar, J., Z. Maamar, et al. (2007). "An Argumentation

Framework for Communities of Web Services." IEEE Intelligent Systems 22(6): 75-83.
Berardi, D., D. Calvanese, et al. (2003). A Foundational Vision for e-Services. Proceedings of Ubiquitous Mobile Information and Collaboration Systems Workshop (UMICS’2003) held in conjunction with The 15th International Conference On Advanced Information Systems Engineering (CAiSE’2003). Klagenfurt/Velden, Austria: 28-40.
Claypool, M., P. Le, et al. (2001). Implicit interest indicators. Proceedings of the 6th international conference on Intelligent user interfaces. Santa Fe, New Mexico, United States, ACM: 33 - 40
Cruz, S. M. S. d., M. L. M. Campos, et al. (2004). Monitoring e-business Web services usage through a log based architecture. IEEE International Conference on Web Services, IEEE: 61-69.
Herlocker, J. L., J. A. Konstan, et al. (2004). "Evaluating collaborative filtering recommender systems." ACM Trans. Inf. Syst. 22(1): 5-53.
Margaria, T. (2008). The Semantic Web Services Challenge: Tackling Complexity at the Orchestration Level. 13th IEEE International Conference on Engineering of Complex Computer Systems, 2008. ICECCS 2008, IEEE: 183-189.
Melville, P., R. J. Mooney, et al. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002). Edmonton, Canada: 187-192.
OASIS. (1999, 1999). "ebXML." Retrieved 10 September 2008, from URL: http://www.ebxml.org/.
OASIS. (2004, October 2004). "UDDI Specification Draft 3.0.2." Retrieved 20 September 2008, from URL: http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm.
Paolucci, M., T. Kawamura, et al. (2002). Semantic Matching of Web Services Capabilities. Proceedings of the First International Semantic Web Conference on The Semantic Web, Springer-Verlag. 2342: 333-347.
Perugini, S., M. A. Gonçalves, et al. (2004). "Recommender Systems Research: A Connection-Centric Survey." J. Intell. Inf. Syst. 23(2): 107-143.
Pu, P., L. Chen, et al. (2008). "Evaluating product search and recommender systems for E-commerce environments." Electronic Commerce Research 8(1): 27.
Sapkota, B. (2005). Web Service Discovery in Distributed and Heterogeneous Environment. Proceedings of the WWW Service Composition with Semantic Web Services (wscomps05) Workshop. Compiegne, France
Shen, Z. and J. Su (2007). On Complexity of The Tightening Problem for Web Service Discovery. IEEE International Conference on Service-Oriented Computing and Applications, 2007. SOCA '07, IEEE: 29-38.
Thio, N. and S. Karunasekera (2005). Client profiling for QoS-based Web service recommendation. Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific.
Torres, R., S. M. McNee, et al. (2004). Enhancing digital libraries with TechLens+. Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries Tuscon, AZ, USA, ACM, New York, NY: 228-236.
Trabelsi, S., J. C. Pazzaglia, et al. (2006). Secure Web Service Discovery: Overcoming Challenges of Ubiquitous Computing. Proceedings of the European Conference on Web Services (ECOWS'2006). Washington, DC, USA.
van der Aalst, W. M., H. A. Reijers, et al. (2007). "Business process mining: An industrial application." Inf. Syst. 35(5): 713-732.
W3C. (2004). "Web Services Architecture Requirements: W3C Working Group Note 11 February 2004." 2010, from http://www.w3.org/TR/wsa-reqs/.
Werthner, H., H. R. Hansen, et al. (2007). Recommender Systems. Proceedings of the 40th Annual Hawaii International Conference on System Sciences, HICSS'07. Hawaii, IEEE Computer Society, Washington, DC, USA: 167.
Yu, Q., A. Bouguettaya, et al. (2008). "Deploying and Managing Web Services: Issues, Solutions, and Directions." The VLDB Journal 17(3): 537-572.[v]

---

i http://www.amazon.com/
ii http://www.lastfm.comq/
iii http://www.cs.indiana.edu/~sithakur/l542_p3/
iv http://www.grouplens.org/