Recommendation in Interactive Web Services Composition: A state-of-the-art survey

Yassine Jamoussi [1, 2] and Meriem Kasmi[1],

[1] Department of Computer Science, College of Science, Sultan Qaboos University
PO Box 36, Al-Khoudh 123, Muscat, Oman
[2] RIADI Lab, ENSI, University of Manouba, Tunisia
{yessine}@squ.edu.om

ABSTRACT

With the increasing adoption of Web services, designing novel approaches for recommending relevant Web services has become of paramount importance especially to support many practical applications such as Web services composition. In this paper, a survey aiming at encompassing the state-of-the-art interactive Web services composition recommendation approaches is presented. Both Web services composition and recommender systems concepts are introduced and their particular challenges are also discussed. Moreover, the need of using recommendations techniques to support Web services composition is also highlighted. The most relevant approaches dedicated to address this need are presented, categorized and compared.

KEYWORDS

Web services composition, state-of-the-art interactivity, recommender systems.

1. INTRODUCTION and MOTIVATION

Nowadays, Internet has globally proven to be a powerful platform where users can find all information they need. In fact, anyone can have access to the World Wide Web, and people everywhere are expressing their ideas and opinions through Internet. Even companies do not escape this rule as they can encapsulate their business processes and publish them as services using a Web service format [1]. This technology has become a de facto way for sharing data and software as well as integrating heterogeneous applications. Consequently, the number of Web services is tremendously increasing. According to the Web services search engine Seekda[1], there are 28,606 Web services on the Web, offered by 7,739 different providers as of August 2, 2011. Furthermore, several Web services publication websites have appeared such as WebServiceList1[2]and XMethods[3]. This large number of Web services available has led to a challenging problem. That is, users have to choose the best Web service satisfying their needs and this is not easy due to this choice explosion. Moreover, due to the complexity and the diversity of users' demands, a single Web service is usually unable to respond to a specific user request. Thus, one interesting feature is the possibility to create new value-added Web services by composing other existing ones. This process called Web services composition (WSC) has become of paramount importance in the domain of Web services. It aims at integrating fine-grained Web services into large-grained composite ones. Currently, WSC has been heavily studied from both industrial and academic fields. This research area is drawing more and more attention in order to obtain the most relevant WSC and this is the rationale behind this paper.
Our paper focuses on the recommendation in the interactive WSC and offers a survey of state-of-the-art recommendation approaches to support interactive WSCs. Then, it provides a classification of current approaches of interactive WSC recommendation, through

[1] http://webservices.seekda.com/
[2] http://www.webservicelist.com/
[3] http://www.xmethods.net/

which, we hope to contribute in the future research in the area of interactive WSC recommendation.

The remainder of this paper is organized as follows. Section 2 overviews some background information on WSC and recommender systems concepts and describes WSC recommendation in general. A classification of the interactive WSC recommendation approaches is presented in section 3. Section 4 reports the comparative evaluation of the mentioned approaches. Section 5 gives a discussion of the evaluation and finally section 6 sums up the conclusion.

2. PRELIMINARY CONCEPTS

The aim of this section is to give an outline of the key concepts and terminologies that will be needed in the rest of this paper. This will form a basis for the later sections.

2.1 Web services composition

According to the W3C[4] (*World Wide Web Consortium*), a Web service is "a software system identified by a Universal Resource Identifier (URI), whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols". To meet users' requirements, Web services can then be composed as new value-added and cross-organizational distributed applications [2].

The WSC process consists of four steps: planning, discovery, selection, and execution [3]. Planning determines the execution order of tasks. Discovery finds the candidate services for each task. Selection selects the best services from the discovered ones and finally the plan is executed.

WSC can be performed manually, automatically, or semi-automatically. For the first approach, where Web services are entirely composed by hand, users need to be technically skilled. This kind of composition is time-consuming and error-prone without any guarantee that the result will really satisfy the user's demands [4]. In contrast, in automated composition, the whole process is automatically performed without any user intervention required. However, realizing a fully automated WSC presents several open issues. It faces the indecision problem that needs to involve users [5]. The last approach aims at assisting users in the composition procedure. This composition being halfway between the previous two types is called also interactive WSC. Interactive WSC comes therefore to resolve the situation by addressing particular issues, for instance the difficulty of selecting a relevant service among the many available ones.

To sum up, manual composition seems to be, at first glance, the most adaptable to users' needs because it offers them the possibility to define everything as they want. However, it requires a good level of programming knowledge. In this situation, users who are tech-novice must be rejected. With the emergence of the Web 2.0, the composition process has become much more end-user oriented. In fact, this wave of Web has brought new technologies for end-users using graphic tools such as mashups. This technology has emerged as a promising way to enable end-users to combine easily services, in short time and obtain scalable results. Due to these advantages, mashups have become prevalent nowadays and therefore a number of mashup repositories have been established namely ProgrammableWeb.com, myExperiment.org, and Biocatalogue.org. In these repositories, a large number of published services are offered. For example, to April 20 2016, the largest Web services repository ProgrammableWeb.com possesses 7.806 Mashups. Moreover, several commercial mashups development environments were developed such as Yahoo pipes [5]and IBM Mashup Center[6].

[4] https://www.w3.org/

[5] http://pipes.yahoo.com/

2.2 Recommender Systems

The basic idea of recommender systems is to provide users with the most relevant items. They have become a rich research area since the 1990s when the first recommender system, Tapestry, was developed. Since then, there has been much work done in both industry and academia to develop new recommendation approaches. Classic recommendation approaches are usually classified into three categories: Content-based recommendation in which the user is recommended items similar to the ones he/she liked before, collaborative filtering-based recommendation in which the user is recommended items that like-minded users preferred in the past and hybrid approaches combine collaborative and content-based methods. This last category helps to avoid shortcomings of both content-based and collaborative approaches and incorporates the advantages of these two methods.

Over the past few years, both Web services and recommender systems have been active research areas. The marriage between those two concepts has led to the application of recommender systems to Web services. Thus, we talk now about Web service recommender systems which are very useful especially that the available Web services search engines has poor recommendation performance. In fact, those search engines ignore non-functional characteristics of Web services [6] and using them, users should enter correct queries because they are keyword-based. Investing in the Web services field, current recommender systems focus mainly on Web services discovery and selection. We witness a wide range of papers about Web service recommendation mainly Web service discovery [7, 8] and selection [9, 10]. But, it is worth noting that there is not relatively large number of researches devoted to recommender systems for interactive WSC. Such a recommender system may be highly useful to the community especially that since the dawn of Web 2.0, end-users are more and more involved in the composition process in an interactive manner.

In this work, we are particularly interested in recommender systems for interactive WSC. We give thus a classification of their approaches in the following section.

3. CLASSIFICATION of interactive WSC recommendation approaches

This section summarizes related studies that use recommendation to enhance interactive WSC. The classification of these approaches is mainly based on the new emerging trends in recommender systems dedicated to interactive WSC. That is why we have not mentioned classic approaches above. Categories of interactive WSC recommendation approaches presented are: context-aware, social network-based, quality-based, category-aware, time-aware and process-oriented interactive WSC recommendation approaches.

3.1 Context-aware interactive WSC recommendation approach

Context refers to any information that can be used to characterize the situation of entities (users or items). Examples of such information are location and time. Contexts can be explicitly provided by the users themselves or implicitly inferring by the system. Considering contextual information can be very useful and, in certain circumstances, the non-adoption this information in the recommendation process can disorientate the recommendation results. For example, let's consider a travel recommender system. When asked for a vacation recommendation, it can give illogical suggestions if it ignores the temporal context of the request. The vacation recommendation in the winter can significantly differ from the one in the summer, so it is crucial to take into account this contextual information. This

6 https://www.ibm.com/support/knowledgecenter/SSWP9P/welcome

category is preserved to Context-aware recommender systems (CARS) which generate more relevant WSC recommendations by incorporating contextual information of the user.

Zhao(2010). Zhao et al. [11] provided a comprehensive and extensible platform for service consuming and navigation named HyperService. With HyperService, non-technical users can easily search and navigate among services as they surf a document web with hyperlinks. Based on the input keywords and their navigation context, a set of relevant services is recommended. Every time users select a service among the suggested set, another set of services is recommended in accordance with the selected service. Thus, interactively-assembled Web services are brought to end users through a web2.0 style interactive user interface. Semantic Engine is the function kernel of HyperService. It provides the functions of automatic relation discovery, user behavior analysis (usage count, the rating score, etc.), service accessibility by keyword searching as well as context aware recommendations performed thanks to content-based methods. Services that fit user's current navigation context, having the best global importance factors (the more times a service is linked to other services, the more popular it is) and are of users' interests deducted are recommended and displayed to users in an interesting and user-friendly way.

3.2 Social network-based interactive WSC recommendation approach

A social network is a graph representation of interactions between entities. This network has the potential to assume an important role in helping in decision-making situations. Let us recall some decisions that we make in our daily life, such as buying a new product or applying for a job in a particular company. Intuitively, we often ask our friends who have already had experience with that product or that company for their opinions. Therefore, social networks influence and even can change our views in decision-making scenarios. This awareness was fostered in a numerous academic fields such as recommender systems. The idea of using social networks in recommender systems stems from other additional advantages mentioned in [17].

Maaradji (2010). Maaradji et al. [5] introduced the SoCo framework which relies on the retrieved knowledge from social networks modeling users' interactions to advise end-users on which Web services to select as they interactively undertake WSC tasks. Through a GUI offering drag/drop functionality for mashups, end-users receive a sorted list of candidate services that are relevant to be successor of the current one. They end up with a composition diagram representing the final WSC. SoCo provides dynamic recommendations through an integrated social-aware recommender system. Its recommendations are built upon user profile (containing of his/her interests, preferences, and the history of his interactions with the system) and social proximity extracted from social networks as well as the previously-built compositions. This information is used to estimate a recommendation confidence. Web services recommended are the most trusted ones i.e. having high recommendation confidence values.

Xu (2013). Xu et al. [12] leveraged multi-dimensional social relationships among users, topics, mashups, and services to recommend the services needed to build mashups. Using a coupled factorization algorithm, services needed for the current mashup construction are predicted, ranked and delivered to the end-user at once. The recommended services are not functionally similar. They are rather the whole services needed and they are delivered to end-users not step-by-step but at once based on users' functional specifications and implicit requirements which are inferred by the topic

ISBN: 978-1-941968-37-6 ©2016 SDIWC

model. Users have just to select proper services and compose them by performing “drag and drop” actions.

3.3 Quality-based interactive WSC recommendation approach

So far, interactive WSC recommendation approaches have not focused on the internal properties. Precisely, quality issues have not been invoked and there is a lack of proposals addressing quality-based interactive WSC recommendation. However, quality assessment may be instrumental when selecting services for composition. For instance, in case two services are functionally-similar, quality can be a discriminant factor. In line with this view, dealing with quality issues in the interactive WSC recommendation may be a promising research area.

Picozzi (2010). Picozzi et al. [13] proposed a model aiming at supporting end-users in the recognition of the most suitable components. The quality-based proposed recommendation approach computes the quality of mashups to produce high-quality mashup recommendation. This value is actually an aggregated quality measure calculated on the basis of the quality of each component in the mashup. End-users who have already shaped final or intermediate mashup can get mashups recommendations about possible extensions of a given mashup. They can extend a particular mashup based on a certain recommendation and continue to extend the obtained mashup by considering other recommendations, realizing thus an interactive composition. The recommended mashups are ranked on the basis of a quality-driven recommender algorithm.

Cappiello (2012). Cappiello et al. [14] illustrated the incorporation of quality based recommendations in the mashup development process to enable end-users to complete and/or improve their mashups. Thus, an assisted composition process in which quality and role of the candidate services are the driver of mashup recommendations was stressed in this work. As in [20], the quality of the composition is computed as a weighted aggregation of the quality of the single components. Weights reflect roles i.e. importance of each candidate service within the composition. Once the user selects the first candidate component, the quality-based ranking algorithm is executed. It proceeds according to two steps: i) the categorization of the component to include in the current mashup using collaborative filtering mechanisms ii) the selection of a particular component, belonging to the defined category in i). Another interesting functionality of this algorithm lies in recommending similar but higher-quality compositions when applied on final ready-to-use mashups.

3.4 Category-aware interactive WSC recommendation approach

A WSC is a mixture of functionally-different Web services. It is quite obvious thus that the recommendation result contains services from various categories. However, most existing interactive WSC recommendation approaches do not provide candidate services ranked per category; they are given all in a single diverse list. This can lead to meaningless service ranking. Additionally, mashup composers are usually not clear about which categories they need. As long as relevant service categories are not explicitly provided, the user friendliness of recommendation will be decreased [15].

Xia (2015). A novel category-aware service recommending method is proposed in [15]. It is actually a three-step approach to overcome the aforementioned restrictions and offers a performing category-aware service recommendation for mashup creation. In fact, after receiving a requirement text from a user, the category-aware recommendation engine analyzes it to infer the categories of services that are going to be implied in the mashup composition task. Then, the engine searches for candidate services belonging to the deducted categories and ranks them within these categories. Finally, the recommendation engine returns "per category service candidate ranking lists". The user selects from each category a service and thus the composition is executed. Once a mashup requirement is received, the service category relevance prediction starts. Combining machine learning and collaborative filtering, the approach decomposes mashup requirements and explicitly predicts relevant service categories. This implies that the problem where users not clear about the needed service categories for mashup creation is henceforth solved through this approach. Finally, based on a distributed machine learning framework, services are recommended in the form of "Category per candidate service ranking lists". Hence, the meaningless service ranking issue is overcome.

3.5 Time-aware interactive WSC recommendation approach

Popular Web service ecosystems such as ProgrammableWeb.com are extremely dynamic and continuously evolving over time. This is due to the large number of incoming services joining the repository, simultaneously with many others perishing, becoming thus deprecated. For example, as we have mentioned before, to April 20 2016, there are 7.806 mashups available in ProgrammbeWeb.com but 1.530 of them are deprecated. This situation has led to the emergence of few but valuable efforts centered around time-aware recommendation approaches.

Zhong (2015). Zhong et al. [16] extended their work in [15] to include the time factor reflecting the evolutivity of the ecosystem. Based on their model in [15], they developed a time-aware service recommendation framework for mashup creation. It is composed of three components: temporal information (TI) extraction, mashup-description-based collaborative filtering (MDCF) and service-description-based content matching (SDCM).
TI predicts service activity in the near future based on usage history. The predicted value corresponds to the service popularity score in recent time frame. To do this, TI predicts topic i.e. category activity first and then infers the service activity because directly predicting service activity will face the sparseness problem.
MDCF recommends services using collaborative filtering techniques applied on mashups having similar functional descriptions with the functional requirements of the new required mashup. Similarity measurement is a key mean here. Once the set of most similar historical mashups is obtained, the relevance score of services with respect to the new required mashup can be evaluated.
As for SDCM, it computes content similarity between the functional requirements of the new required mashup and the content description of services based on LDA.
Popularity scores from TI and relevance scores from MCDF and SDCM are integrated to generate the ranked recommended list of services for the new required mashup.

3.6 Process-oriented interactive WSC recommendation approach

Wijesiriwarna (2012). Wijesiriwarna et al. [18] proposed a guided process-oriented mashup platform called SOFAP for software analysis composition. This platform allows different software projects stakeholders to access software analyses, compose them into workflows and execute the obtained composition. Every time users select a service, the recommendation engine provides him/her with the next possible services to add to the composition schema. In case of a wrong selection or following an incorrect control-flow pattern, recommendation engine gives a real-time feedback to the user. Once finished, the composed workflow is passed to the mashup run-time for the execution. Also, meaningful workflow templates are stored in the SOFAS repository, allowing future users to reuse the existing templates.

4. Comparative evaluation

In order to evaluate the interactive WSC recommendation approaches classified in the previous section, the following criteria have been selected. Table 1 summarizes the result of the comparative evaluation.
The adopted criteria are described as follow:

- Personalization (P): It refers to the fact of recommending services according to users' interests. To do so, user behavior for example can be incorporated and analyzed into the recommendation model to generate more personalized and accurate services.
- Recommended items (RI): Although the discussed approaches are dedicated all to generating relevant interactive WSCs, their outputs are not delivered in the same form. In [11, 5, 14], a ranked list of similar candidate services that are relevant to be successor of the current one are recommended. In [12], the recommended services are ranked but not functionally-similar. They are rather all candidate services needed delivered at once. In [18], similar services are recommended to add to the composition schema but the ranking issue was not invoked. In [13], ranked mashups are provided to end-users in order to help them in extending a particular mashup. In [16, 17], "per category service candidate ranking lists" because [17] uses the same model as [16] extending by the factor time. We denote thereafter these five forms of recommended services as: Ss,r which refers to "List of similar ranked services" , Sns,r for "list of not similar ranked services", Ss,nr which refers to "List of similar not ranked services", Ms,r for "List of similar ranked mashpus" and C(Ss,r) for "List of categories of similar ranked services".
- Interactivity level (IL): This criterion describes to which extent a user is involved in the interactive WSC. In [11], each time a user selects a service among the suggested set, another set of services is recommended in accordance with the selected service. Thus, the user is highly involved in the WSC process, since he/she should select candidate services one by one from the suggested sets to fulfill the WSC task. It is also the case in [5, 14]. If the selection of candidate services is done simultaneously, interactivity level is lower. It is the case in [12, 15, 16]. When the user selects a whole mashup at one time, the interactivity is much lower such as in [13]. We respectively symbolize these three cases by (+++), (++) and (+).

Regarding the remaining criteria, they are those that we have explained above and adopted to categorize interactive WSC recommendation approaches. Those criteria are: Context awareness (C), Social Network awareness (SN), Quality awareness (Q), Category awareness (Cat), Time awareness (T) and process orientation (P). In fact, we noticed that being of any class of approach does not mean excluding other classes. In contrast, this may improve recommendation accuracy and yield better results. Yet, there are different awareness levels towards these criteria. We distinguish 3 levels among the different studied papers: extreme awareness for those which particularly focus on that criterion, medium awareness for papers which adopt that criterion to refine more their recommendations but are not mainly structured around it and the no-awareness level for works which do not invoke that criterion at all. These three levels are respectively symbolized by: (++), (+) and (-).

ISBN: 978-1-941968-37-6 ©2016 SDIWC

Criteria / Paper	Zhao et al. (2010) [11]	Maaradji et al. (2010) [5]	Xu et al. (2012) [12]	Picozzi et al. (2010) [13]	Cappiello et al. (2012) [14]	Xia et al. (2015) [15]	Zhong et al. (2013) [16]	Wijesiriwardana et al. (2012) [18]
Context awareness	++	-	-	-	-	-	-	-
Social ntwrk awareness	-	++	++	-	-	-	-	-
Quality awareness	-	-	-	++	++	-	-	-
Category awareness	-	-	+	-	+	++	+	-
Time awareness	-	-	-	-	-	-	++	-
Process orientation	-	-	-	-	-	-	-	++
Personalization	✓	✓	✓					
Recommended items	Ss,r	Ss,r	Sns,r	Ms,r	Ss,r	C(Ss,r)	C(Ss,r)	Ss,nr
Interactivity level	+++	+++	++	+	+++	++	++	+++

Table 1. Comparative evaluation of interactive Web Services composition recommendation approaches

5 DISCUSSION

The work of Zhao et al. [11] presents the main advantage of taking into account context in their recommendation approach. This feature is crucial and can even leverage recommendation process because users' interests can change if ever being in a particular place or at a particular date. Recommendations are also personalized in this work. Nevertheless, its composition process highly involves users and this limits their effectiveness.

As for [5] and [12], they are both social-based but Xu et al. [12] exploited also the idea of category awareness to have much better results. Besides, they both make use of social information to support service recommendation but there is a difference pertaining to their models of relationships. In [12], relationships are multi-dimensional including users, topics, mashups, and services while, in Maaradji's [5], the social network models only users' interactions. Unlike [5], the recommended services are not functionally similar in [12]. They are all the services needed for the composition, delivered to end-users at once. This is an interesting feature offered by Xu's work keeping an effective level of interactivity in the WSC task. As a future outlook, Xu et al. announced their will to build an online collaboration platform based on their approach. According to them, recommendations will be improved since the model will be more used, and thus it will collect more useful information on composition patterns. This is an appealing idea to which we are also interested.

Picozzi' work [13] and Cappiello's work [14] are among the few proposals addressing quality issues in interactive WSC recommendation. They both proposed extensions of users' mashups in order to enhance the overall mashup quality. This latter is perceived as a weighted aggregation of single components qualities. Weights reflect roles i.e. importance of each candidate service within the composition. Thus, these two works present the advantage of, in addition to being quality based, they are also role-based. In [13], the interactivity level is very low because its recommendations are high-quality ranked mashups to add to the current one. In contrast, users are highly involved in WSC in [14] since they have to select for each service belonging to their current mashup, another one from a ranked list of similar services. We note also that [14] performs an automatic categorization to recommend users with more relevant services. However, neither [13] nor [14] have involved personalization in their recommendation method.

Xia et al. [15] proposed a category-aware service recommending method. Its leading advantages are the fact that common restrictions (meaningless service ranking and low user friendless of recommendations) within existing recommendation methods are alleviated in their approach. Services are recommended in the form of "Category per candidate service ranking lists", so end-users have just to select from each category the service they need and the composition will be executed. Experiments conducted by authors have proved that their approach not only improves recommendation precision but also the diversity of recommendation results. Zhong et al. [16] provide almost the same model is described but including the time dimension when recommending service for mashup creation. Precisely, recommendations are also "Category per candidate service ranking lists" having relevant service activity in the near future. Despite their good performance, these two approaches do not provide personalized recommendations.

Wijesiriwardana et al. [18] proposed a process-oriented intactive WSC recommendation approach. This work differs different from the other presented ones in many aspects. We mention the collaborative dimension of the proposed platform as well as the composition model based on workflow. Moreover, the recommendations provided are dedicated to sotware projects membres such as developers, architectures and testers. One attractive advantage here is that the recommendation engine gives a real time feedback to the user in case of a wrong selection or incorrect control-flow pattern. Thus, users are very well supported in this work.

6 CONCLUSION

In conclusion, this paper has provided an overview and evaluation of current interactive WSC recommendation approaches. An introduction to recommender systems as well as WSC was presented in which we particularly focused on interactive or semi-automatic WSC. We raised and highlighted the need for a synergy between recommender systems and interactive WSC. Therefore, we studied the most prominent emerging approaches of interactive WSC recommendation and classified them into categories. We tried to cover all interactive WSC recommendation categories to get the most exhaustive possible classification. We also supported this classification by several criteria in order to evaluate and compare the approaches. Finally, a summary of the comparison and evaluation of the approaches are presented and discussed.

REFERENCES

[1] Tang, J.: A Service Composibality Model to Support Dynamic Cooperation of Cross-Enterprise Services. In: IFIP International Federation for Information Processing, Information Technology for Balanced Manufacturing Systems, (ed.) Shen, W. 2006, vol. 220, pp. 213—222. Springer, Boston (2006)

[2] Zou, G., Lu, Q., Chen, Y., Huang, R., Xu, Y., Xiang, Y., QoS-Aware Dynamic Composition of Web Services Using Numerical Temporal Planning.

In: *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 18—31 (2014)

[3] Claro, D. B., Albers, P., Hao, J.-K..: Web Services Composition. In: Semantic Web Services, Processes and Applications. (eds.) Cardoso, J. and Sheth, A. P., Springer (2006)

[4] Sheng, Q.Z., Qiao, X., Vasilakos, A.V., Szabo, C., Bourne, S., Xu, X.: Web services composition: A decade's overview. J. Infor. Sciences. 280, 218--238 (2014)

[5] Maaradji, A., Hacid, H., Daigremont, J., Crespi, N.: Towards a Social Network Based Approach for Services Composition. In: Proceedings of the IEEE International Conference on Communications, Cape Town, pp. 1-5(2010)

[6] Yao, L., Sheng, Q. Z., Ngu, A. H. H., Yu, J., Segev, A.: Unified Collaborative and Content-Based Web Service Recommendation. In: IEEE Transactions on Services Computing, vol. 8, no. 3, pp. 453-466 (2015)

[7] Ma, J., Sheng, Q. Z., Liao, K., Ngu, A. H.: WS-Finder: A Framework for Similarity Search of Web Services.In Service-Oriented Computing. pp. 313-327.(2012)

[8] Xu, Z., Martin, P., Powley, W., Zulkernine, F.: Reputation-Enhanced QoS-based Web Services Discovery., In: Proceedings of the 2007 IEEE International Conference on Web Services (ICWS), Salt Lake City, UT, pp. 249–256 (2007)

[9] Zibin, Z., Hao, Z., Michael R., L., Irwin, K.: WSRec: A Collaborative Filtering Based Web Service Recommender System. In: IEEE International Conference on Web Services, pp.437-444 (2009)

[10] Zheng, Z., Ma, H., Lyu, M. R., King, I.: QoS-Aware Web Service Recommendation by Collaborative Filtering. In: IEEE Transactions on Services Computing, vol. 4, no. 2, pp. 140–152 (2011)

[11] Zhao, C., Ma, C., Zhang, J., Zhang, J., Yi, L., Mao, X. HyperService: Linking and Exploring Services on the Web. In: Proceedings of the 2010 International Conference on Web Services, Miami, FL, pp. 17-24 (2010)

[12] Xu, W., Cao, J., Hu, L., Wang, J., Li, M.: A Social-Aware Service Recommendation Approach for Mashup Creation. In: Proceedings of the 20th IEEE International Conference on Web Services, Santa Clara, CA, pp. 107-114 (2013)

[13] Picozzi, P., Radolfi, M., Cappiello, C., Matera, M.: Quality-based Recommendations for Mashup Composition. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 360–371. Springer, Heidelberg (2010)

[14] Cappiello, C., Matera, M., Picozzi, M., Daniel, F., Fernandez, A.: Quality-Aware Mashup Composition: Issues, Techniques and Tools. In: Proceedings of the Eighth International Conference on the Quality of Information and Communications Technolog, Lisbon, pp. 10-19 (2012)

[15] Xia, B., Fan, Y., Tan, W., Huang, K.., Zhang, J., Wu, C.: Category-aware API clustering and distributed recommendation for automatic mashup creation.IEEE Transactions on Service Computing, vol. 8, no. 5, pp. 674–687(2015)

[16] Zhong, Y., Fan, Y., Huang, K., Tan, W.,Zhang, J.: Time-Aware Service Recommendation for Mashup Creation. IEEE Transactions on Services Computing, 356-368 (2015)

[17] He, J., Chu, W. W.: A Social Network-Based Recommender System (SNRS). In: Annals of Information Systems: Special Issue on Data Mining for Social Network Data, Vol.12, pp. 44–74 (2010)

[18] Wijesiriwardana, C., Ghezzi, G., Gall, H.: A Guided Mashup Framework for Rapid Software Analysis Service Composition. In: the 19th IEEE conference on Software Engineering Conference, Asia-Pacific, pp. 725-728 (2012)