Empirical Studies on Community Structure for Networked Web Services

Yuanbin Han[1], Shizhan Chen[1,*] and Zhiyong Feng[1]

[1] School of Computer Science and Technology, Tianjin University, Tianjin, 300072, China

Abstract

This paper presents studies on detecting community structure in web services formed network, which can significantly explore and understand the underlying functionality and behavior of interactions among web services, as well as facilitate the state of art service computing. The community structure in this paper focusing on two typical social characteristics for networked web services: competition and collaboration. Competition-oriented community structure is based on the functional semantics (i.e. the *inputs/outputs* of web services), in which we group web services sharing common interests. Collaboration-oriented community structure is computed by the topological analysis, so that we can cluster web services that interact densely. We present empirical analysis on our dataset and the generated communities for capturing the insight dynamics for web services formed network. Besides, we also present some potential utilities which can accelerate service-oriented computing.

Keywords: *Web service, Network analysis, Semantic, Community structure.*

1. Introduction

Service-oriented computing (SOC) has attracted much attention during recent years[1], which allows people reusing loosely-coupled software applications, by means of service discovering and composing. As the new related innovations continued to emerge, such as cloud computing [2, 3], software as a service, Internet of services [4], service computing plays an increasingly important role to date. Graph-based web service network opens new possibilities for handling the tremendous increase of web services. Graph-based web service networks are based on the interactions among web services. Since using networked web services can capture the pre-computing of some potential composing patterns, it can efficiently construct composite services, which yield lower time complexity, compared with most of other AI-based approaches [5].

Due to advances in network analysis techniques, recently, there has been a considerable amount of efforts focusing on the network analysis for web services formed network [6-12], which can facilitate the state of art SOC. Community structure is the common feature in complex networks [13-15], which describes nodes (actors) who interact heavily or share some common interests. Hence, detecting community structure in web services formed network can explore and understand the underlying functionality and behavior of interactions among web services, as well as simplify large-scaled service-based networks, which can be essentially helpful for service computing.

To accelerate the SOC, *a key issue is how to help users easily understand and navigate the behavior of web service ecosystem*. The objective of this paper is to detect significant communities, as well as explore and understand the underlying functionality and behavior web services, as well as facilitate the state of art service computing. In this paper, we proposed two approaches for mining community structure in web services formed network, which combine semantic techniques and network topological characteristics. As we will show in this paper, these findings can explore both the competition and collaboration features for networked web services.

The roadmap of this paper is structured as follows. Section 2 describes the related work. Section 3 shows the dataset we used, as well as the web service network model. Section 4 presents two approaches for mining community structure, as well as the experimental results, and demonstrates the dynamics of the proposed community structure. Section 5 presents discussions about the proposed methods, while Section 6 presents conclusion of this paper, as well as the limitation and future direction.

2. Related Work

Previous studies have shown that using web services formed network can significantly benefit service composing process, particularly in terms of efficiency. [25, 26] proposed composing methods based on a graph model, and stated that the methods can achieve an acceptable performance. Shin et al. [5] extended the dependency graph by considering the services also as essential

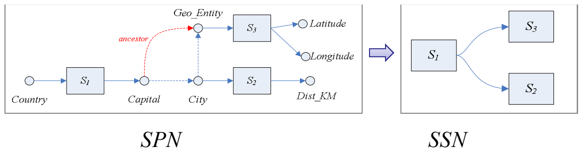
[*] Corresponding author

stakeholder in the dependency relations, and leveraging a two-layered graph model for indexing functional semantics. Experimental results showed their work yielded lower complexity than most of other AI-based approaches.

As already stated in this paper, there has been a tremendous amount of efforts on network analysis for networked services, which are briefly following two research lines [1]: the bottom-up network analysis[6-9,11,12,18] focusing on the network formed by the services repository, and the top-down manner network analysis which the service-based network are triggered by the wisdom of crowds[10]. These efforts are generally motivated by the achievements in the area of complex network and social network, for the purpose of exploring meaningful phenomena and laws of service-based network. Additionally, recently we have witnessed some efforts on network analysis on a new kind of emerged web service called Open API and its composited application Mashup [27]. For instance, [28, 29] proposed to study the network properties for Open API and Mashup Ecosystem by leveraging 2-mode network models, in which the derivative 1-mode model can reveal the insight collaborative laws of Open APIs.

As to the community structure for web services, the Self-Serv project [30, 31] considered community as grouping of services with similarity measures. In [32], the authors presented a framework for gathering services with similar function into communities by combining argumentative agents. By the similar idea with [30, 31], Liu et al. [33, 34] comprised the "*service pool*" and task template in constructing service communities for bridging end-users and services computing. Additionally, the authors also implement a composition approach based on their proposed service community structure. We regard the above efforts as the similar idea of "*abstract services layer*" in our previous project [11], in which the abstract services are functional indices of concrete services by collecting functional similar concrete services. The work presented in this paper can be viewed as a second abstracting of the above mentioned communities by considering the hierarchy of ontology. Moreover, we also studied the topology-based community detection which captures the collaboration-oriented social characteristics.

[1] A detailed statements about literatures related with service-based network analysis were presented in our previous work [18].



Service	Inputs	Outputs	Description
S_1	*Country*	*Capital*	Provide the capital of a country
S_2	*City*	*Dist_KM*	Calculate the distance between 2 cities
S_3	*Geo_Entity*	*Latitude, Longitude*	Inform Latitude and longitude for a geographical entity

Fig. 1. An illustration of *SPN* & *SSN*.

3. Data sets and Models

3.1 Data sets.

The dataset used in our work is OWL-S Service Retrieval Test Collection (*OWLS-TC3*) [2]. We select the most popular 4 domain services in *OWLS-TC*, which related to communication, economy, education and travel domain, since they are most concerned with real-life applications. However, we also abandoned 77 parsed services since they either have no *inputs* or have no *outputs*. The sub-ontology extracted from the 23 different ontologies is used for specifying the semantics of web services, as summarized in Table 1. In doing network analysis, we comprised both *Pajek*[16] and *igraph*[17] tools for visualization and analysis of community.

Table 1: Dataset

# *parsed services* (along with *I/O*s)	867
Extracted sub-ontology size(# *classes*)	271
# *parsed services* used in this work	790
# *parameters* (both *inputs* and *outputs*)	250

3.2 Network Model

In general, functional semantics of a web service can be specified by its *inputs/outputs* parameters (in our dataset, services are stateless, which means that they have no *precondition* and *effect* information). In view of this, we formed the service dataset into two graph models: (1) Service-parameter network (*SPN*), a directed graph which describes the dataflow among web services, and (2) Service-service network (*SSN*) describes the direct relation between services, which can be transformed from *SPN*. The two network models are illustrated in Fig. 1.

[2] *semwebcentral.org*: http://projects.semwebcentral.org. The dataset was also used in our previous work [18], in which readers can refer to more details.

The construction process and network analysis for the two network models were presented in our previous work [18], readers can refer to our previous work for more detailed information.

4. Methodology

4.1 Community structure based on functional semantics

Community structure based on functional semantics was inspired by the visualization of *SPN*, as shown in Fig. 2, in which we use the "*dissimilarities*" [16] distance as the lines value. From Fig. 2, we can clearly witness the clustering phenomenon, where concepts (i.e. *I/O*s) tend to gather into different groups, as the circled areas point out. To further explore what exactly these clusters are, we magnify of the circled area 5 times, as the yellow circled area shown in Fig. 3(a), and we observed that the concepts in the cluster are all "*Time*" related in semantics. More specifically, they are bound up with their "*locations*" in the ontology. As the example shown in Fig. 3(b), the concepts in the same cluster are concept "*Time*" and the descendants of "*Time*".

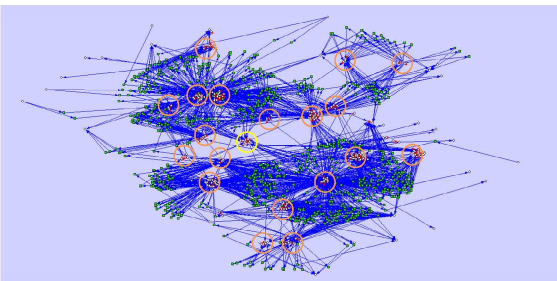
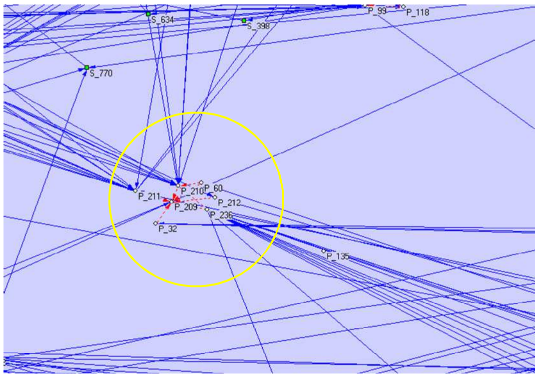


Fig. 2. Visualizations of Clusters in *SPN*, where green boxes denote services, white circles denote *I/O*s.

In view of this, we attempt to construct communities by mapping clusters in *SPN* into *SSN*, where each community implies that its members (i.e. web services) share common interests. In another angle, members in the same community also mean that they have similar function, or they can achieve similar goals while constructing web service discovering and composition for a certain user requirement. Therefore, we argue that web services in the same community hold competitive relations from the perspective of society, and we call community structure based on the functional semantics as "*competition-oriented*".

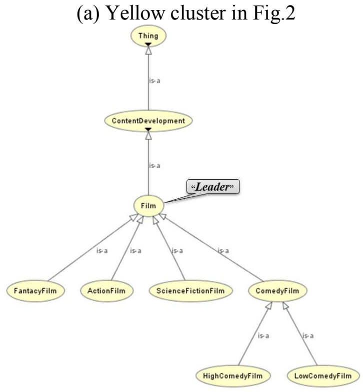
As mentioned before, concepts in the same cluster hold kinship and share a common ancestor. The idea of mining functional semantics-based service communities is simply followed as:

(1) We first divide all *output* parameters in our dataset into groups from the *I/O* ontology, by finding the *output* parameters that have no parents in the *output* set, and we denote the common ancestor shared by concepts in the same cluster as "*Leader*" for each community.
(2) Secondly, we compute all the descendants for each "*Leader*" parameter from the ontology, and form the concept community, as the example shown in Fig. 3(b) (visualized by protégé [19]), in which the "*Leader*" of the concept community is "*Film*".
(3) Finally, by constructing an inverted indexing from the service dataset, which is in terms of "*output-services*", we can compute the service community based on the concept community and mapping from the "*output-services*" inverted indexing.



P-32 Calendar-date; P-60 Day; P-209 Time; P-210 TimeDuration; P-211 TimeMearsure; P-212 TimePosition; P-236 Year

(a) Yellow cluster in Fig.2



(b) Concept community of (a)

Fig. 3 Insight of the cluster in *SPN*, with 5 times magnifying. As can be seen, all the members are time-related concepts.

By using this method, we construct 43 communities, which are marked by the "*Leader*" concepts. Fig. 4 shows the 43 resulted communities, as well as the number of concepts (including the "*Leader*") and services they maintain. Note that the total number of services is 1049, which is more than the number of 790 in the dataset, since some services may belong to more than one community.

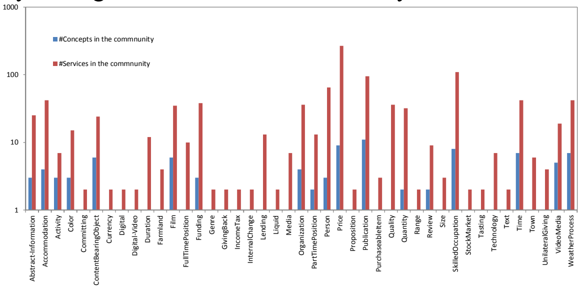


Fig. 4. Communities Based on Functional Semantics.

Competition-oriented community structure based on functional semantics can be helpful when computing composite services, since it can simplify the composing process by inter- and intra-community searching. Moreover, since we also decide the "*Leader*" of each community, it is often beneficial for refining users' requirements. For example, users often require "*Hotel*" at the beginning rather than "*3-star Hotel*".

4.2 Community structure based on topological information

Detecting community structure has been a flourishing research in the flied of social network, and there have been a large number of efforts focusing on efficiently mining communities based on topological information. In this section, we performed one of the efficient community structure detection method proposed by Pons et al. [20] to see how community structure looks like in our dataset.

The network structure is based on the *SSN* model mentioned in Section 3.1. We removed all the loops and isolated nodes (totally 97 isolated services) of *SSN* formed by the dataset in Table 1, and visualized it in Fig. 5.

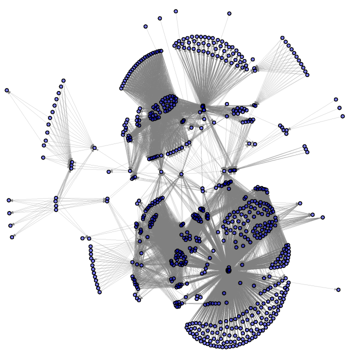
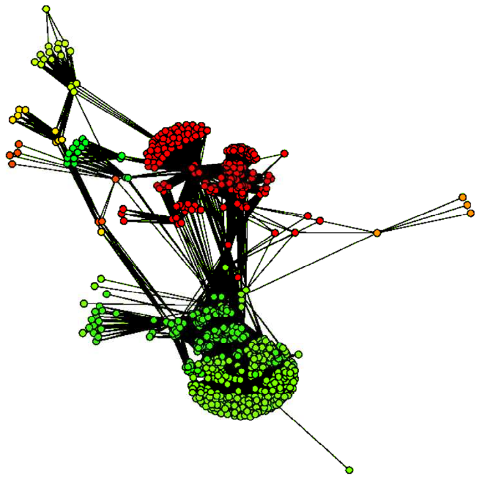


Fig. 5. Visualization of *SSN*

As can be seen from Fig. 5, the *SSN* presented in this paper is a connected direct graph, in which there are 693 web services, 13818 directed links (Note the basic statistics are a little different from the results in our previous work [18] since we removed loops and isolated nodes in *SSN*). The density and average node degree of *SSN* are 0.029 and 39.879 respectively.



Communities	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Size	194	7	4	9	17	275	172	15

Fig. 6. Community Structure Based on *Walktrap* Method

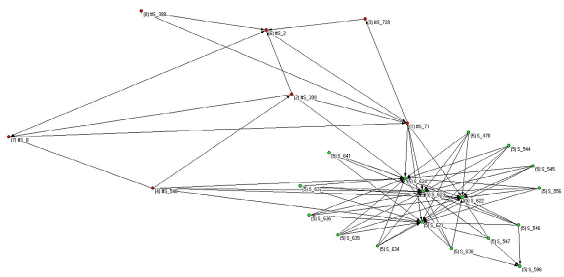


Fig.7. Microanalysis on community C_5. The red circles denotes other communities (i.e., C_1, C_2, C_3, C_4, C_6 C_7, C_8,) which are not detailed (other communities are abstracted into single vertices), green circles are the concrete members(web services) in C_5.

Walktrap [20] was proposed for mining community structure by using random walks in a graph, which is following the fundamental rule that vertices with short random walks tend to form communities. According Pons et al. [20], *Walktrap* method suffers a fairly good complexity of O (mn^2) and space O (n^2) in the worst case, and a time complexity of O (n^2log n) in the most real-world circumstances, where n and m are the numbers of vertices and links respectively in the graph. As to *SSN* we mentioned in the previous paragraph, communities can be computed in seconds. Fig. 6 is the visualization of communities (with different colors) determined by *Walktrap* method, in which 8 communities are computed.

The studying of performance about communities detecting approaches is not the primary task of this paper, since we are more interested in the insight of the function of the formed communities. To do so, we take a microanalysis on the communities by revealing the details.

Table 2. Concrete members of C_5.

WSs	*Inputs*	*Outputs*
S_{478}	Educational-Organization	Lecturer-In-Academia
S_{544}	Higher-Educational-Organization	Lecturer-In-Academia
S_{545}	Higher-Educational-Organization	Lecturer-In-Academia
S_{546}	Higher-Educational-Organization	Professor-In-Academia
S_{547}	Higher-Educational-Organization	Professor-In-Academia
S_{556}	Learning-Centred-Organization	Lecturer-In-Academia
S_{596}	Professor-In-Academia	Address
S_{621}	Researcher	Abstract-Information
S_{622}	Researcher	Address
S_{623}	Researcher	Address
S_{624}	Researcher	Postal-Address
S_{634}	University	Academic-Support-Staff
S_{635}	University	Lecturer-In-Academia
S_{636}	University	Lecturer-In-Academia
S_{637}	University	Lecturer-In-Academia
S_{638}	University	Lecturer-In-Academia
S_{641}	University	Senior-Lecturer-In-Academia

As Fig. 7 shows, services are densely connected inside the community C_5, while having less links with other communities. From Table 2, we can see that services are densely connected by sequential ties. For instance, there are four most popular services, i.e. "*S_621*", "*S_622*", "*S_623*" and "*S_624*", which have more degrees than other services in C_5, since they have the input parameter "*Researcher*" that can be called by other services with the output parameter "Researcher" (Note concepts "Lecturer-In-Academia", "*Professor-In-Academia*", "*Academic-Support-Staff*" and "*Senior-Lecturer-In-Academia*" are semantically related with "*Researcher*", thus four most popular services can be called by other services, as defined in our network model), which means that they are the "*succeed*" services of other services in the community. This demonstrates that community structure based on *Walktrap* method reveals the fact that services in the same community usually cooperate frequently, which is the feature of "*Collaboration-oriented community structure*" we mentioned.

Fig. 8 is the visualizing of another community C_6, which has the most member population among the 8 communities. It shows that C_6 also follows the fact that services are densely connected inside, while sparsely connected outside.

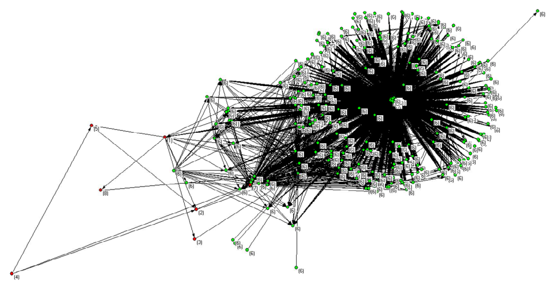


Fig.8. Visualization of Community C_6 (with 275 members).

5. Discussions

In this section, we discuss both pros and cons of the two proposed community detecting approaches, and present some potential utilities for facilitating SOC.

In the case of detecting community structure based on the functional semantics in Section 4.1, we emphasized that it is competition-oriented, since these communities group web services with similar functionality or corresponding to common requirements. Therefore, a possible usage for competition-oriented communities is that we can facilitate composing process by dividing composing into two phases: inter-community service composing and intra-community service discovery. One of our essential ongoing work is focusing on designing hierarchical service composing based on the model illustrated in Fig. 9. In the first step, we construct abstract composition by inter-community searching from users' requirements, which can quickly construct an abstract composite flow, and we can also refine users' requirements by leveraging user interactions with ontology. In the second step, service discovery is proposed for binding concrete services within the communities. Thus we can obtain the composite services by combining the two steps.

In the matter of limitations about community structure based on the functional semantics, services with same parameters may functionally differ from each other since semantics based on *input* and *output* parameters are short of sufficient context information. Therefore, services in the same community may be functionally different in some circumstances. This is serious existing in the community leading by "*Price*" summarized in Fig. 4. For instance, two services related with coffee and cars maintaining the same output "*Price*" are grouped into the same community, though they functionally differ from each other in reality context. We attempt to address this issue by extending the functional semantics with text descriptions and tags of services (Our ongoing work for annotating text descriptions and tags is by leveraging DBpedia [21], Spotlight [22] and Yago[23] ontology).

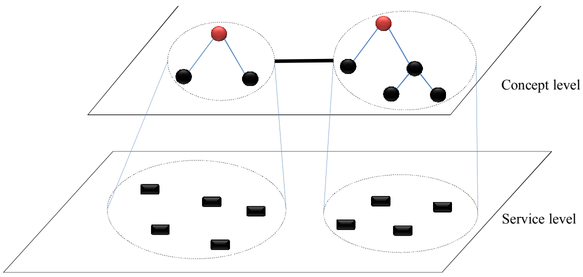


Fig.9. Two layered model for service selection and composition, where the red nodes in concept level can be used for refining users' fuzzy requirements.

For detecting services community structure based on *Walktrap* method stated in Section 3.2, we mentioned that services in the same community are densely connected inside, while sparsely connected outside. This can significantly benefit service composing, since the priority of service selection for a certain composing can be determined by communities. Therefore, a reasonable complexity can be achieved by confining services in the same community, which can characterize the priorities in composing. Furthermore, collaboration-oriented community structure can be also helpful to service recommending. We leave this as our future work.

Although mining community structure based on topological methods is gaining momentary attention in the field of complex network and social network, there have been still enormous challenges in leveraging it for real-world networks. For this paper, *Walktrap* method heavily depends on the topological information of services formed network, but seldom considers the practical semantics of networks, which will lead to the result that some small but semantically meaningful communities might be covered by large communities. It has been stated that considering semantic aspect of information in community structure detecting can achieve desirable results for practical context [24]. We leave this as our future direction for mining semantically meaningful "*Collaboration-oriented community structure*".

6. Conclusions

In this paper, we have suggested two meaningful community detecting methods, which followed two basic lines: (1) the competition-oriented community structure based on functional semantics which derived from the behavior and semantic property of web services, as well as (2) collaboration-oriented community structure computed by topological analysis.

We showed that these findings had a series of meaningful implications for service computing. Firstly, service community structure can track the challenge of constantly growth of web services, for which we can using the idea like the "*Autonomous System*" within the Internet (for web services, i.e. the community) to administer the vast amount of services, as well as to search composite services efficiently. Besides, the community structure based on functional semantics provided a gateway for user requirements refinement. Ultimately, it would be interesting to study service recommending based on the underlying attracting nature of the collaboration-oriented community structure. All these are our priority concerns in the immediate future.

As stated previously, one of the shortcomings is that we only consider the semantics of parameters, which are insufficient for representing the functional information of web services. Our ongoing work is planning to address this issue by adding the semantic information of text description, as well as tags from the wisdom of crowds. Our another interesting is concerning service communities in heterogeneous networks for OpenAPIs and Mashups, in which we consider heterogeneous types of entities(multi-mode networks) in Mashup Ecosystem, such as providers, users, APIs, Mashups, tags, data formats, protocols.

Acknowledgments

This work was funded by NSFC grant 61173155, National High-Tech Research and Development Program of China grant 2007AA01Z130 and the 985 Project of Tianjin University grant 2010XG-0009.

References

[1] M. P. Papazoglou, P. Traverso, S. Dustdar and F. Leymann, "Service-oriented computing: State of the art and research challenges", IEEE Computer, Vol.40, No.11, 2007, pp.38–45.

[2] Y. Wei, M.B. Blake, "Service-oriented Computing and Cloud Computing: Challenges and Opportunities", IEEE Internet Computing, Vol.14, 2010, pp.72-77.
[3] A. A. T. Innocent, "Cloud Infrastructure Service Management -A Review", International Journal of Computer Science Issues, Vol. 9, No 2, 2012, pp.287-292.
[4] J. Cardoso, K. Voigt, M. Winkler, "Service Engineering for the Internet of Services", in 10th International Conference on Enterprise Information Systems, 2008, pp.15-27.
[5] D. Shin, K. Lee, T. Suda, "Automated Generation of Composite Web Services Based on Functional Semantics", Journal of Web Semantics, Vol.7, No.4, 2009, pp.332-343.
[6] H. Cai, "Scale-free Web Services", in Proc. of ICWS, 2007, pp.288–295.
[7] H. Kil, S. Oh and D. Lee, "On the Topological Landscape of Web Services Matchmaking", in Proc. VLDB Workshop on Semantic Matchmaking and Resource Retrieval, 2006, pp.178: 19-34.
[8] H. Kil, S. Oh , E. Elmacioglu, W. Nam, D. Lee, "Graph Theoretic Topological Analysis of Web Service Networks", World Wide Web, Vol.12, No.3, 2009, pp.321-343.
[9] S. Oh, D. Lee, and S.R.T. Kumara, "Effective Web Service Composition in Diverse and Large-Scale Service Networks", IEEE Transactions on Services Computing, Vol.1, No.1, 2008, pp.15-32.
[10]W. Tan, J. Zhang, and I. Foster, "Network Analysis of Scientific Workflows: a Gateway to Reuse", IEEE Computer, Vol.43, No.9, 2010, pp.54-6.
[11]S. Chen, Z. Feng, H. Wang, "Service Relations and its Application in Services-Oriented Computing", Chinese Journal of Computers, Vol.33, No.11, 2010, pp.2068-2083.
[12]S. Chen, Y. Han, Z. Feng, "Social Network Analysis on the Interaction and Collaboration Behavior among Web Services", in Proc. AAAI Spring Symposium, 2012, pp. 9-15.
[13]M. Girvan, M.E.J. Newman, "Community structure in social and biological networks", in Proc. National Academy of Sciences, 2002, pp.99:7821-7826.
[14]M.E.J. Newman, "Detecting community structure in networks", European Physical Journal, Vol.38, No.2, 2004, pp.321-330.
[15]M.A. Porter, J.P. Onnela, P.J. Mucha, "Communities in Networks", Notices of the American Mathematical Society, Vol.56, No.9, 2009, pp.1082-1097.
[16]W. de Nooy, A. Mrvar, and V. Batagelj, “Exploratory Social Network Analysis with Pajek,” Cambridge University Press, 2005.
[17]G. Csardi and T. Nepusz, “The igraph software package for complex network research,” InterJournal, Vol. Complex Systems, 2006, pp. 1695.
[18]Y. Han, S. Chen, Z. Feng, “Optimizing Service Composition Network from Social Network Analysis and User Historical Composite Services”, in Proc. AAAI Spring Symposium, 2012, pp. 39-45.
[19] http://protege.stanford.edu
[20]P. Pons, M. Latapy, “Computing Communities in Large Networks Using Random Walks,” J. Graph Algorithms Appl. Vol.10, No.2, 2006, pp.191-218.
[21]C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, “DBpedia-A crystallization point for the Web of Data,” Journal of Web Semantics, Vol.7, No.3, 2009, pp.154-165.
[22]P. N. Mendes, M. Jakob, A. García-Silva and C. Bizer, “DBpedia Spotlight: Shedding Light on the Web of Documents,” in Proc. 7th International Conference on Semantic Systems (I-Semantics '11) , 2011, pp.1-8.
[23]F. M. Suchanek, G. Kasneci, G. Weikum, “Yago: a core of semantic knowledge,” in Proc. 16th international conference on World Wide Web, 2007, pp.697-706.
[24]W. Wu, Y. Xiao, Z. He, W. Wang, and T. Yu, “Mining Hidden Communities in Social Networks Based on Weight Information,” Journal of Computer Research and Development, Vol.46, No.z2, 2009, pp. 540-546.
[25]S.V. Hashemian, F. Mavaddat, “A Graph-Based Framework for Composition of Stateless Web Services,” in Proc. 4th IEEE European Conference on Web Services, 2006, pp.75-86.
[26]H. Elmaghraoui, I. Zaoui, D. Chiadmi, L. Benhlima, “Graph based E-Government web service composition”, in International Journal of Computer Science Issues, Vol.8, No 1,2011, pp.103-110.
[27]http://www.programmableweb.com/
[28]S. Yu, and C. J, Woodard, “Innovation in the Programmable Web: Characterizing the Mashup Ecosystem,” in Proc. 2nd International Workshop on Web APIs and Services Mashups, 2008, pp.136-147.
[29]G. Ji, J. Zhang, Z. Zhao, and Y. Han, “Service Collaboration Network: A Novel Mechanism for Web Service Management,” in Proc. 12th International Asia-Pacific Web Conference (APWEB '10), 2010, pp. 85-91.
[30]B. Benatallah, Quan Z. Sheng, M. Duma, “The self-serv environment for web service composition,” IEEE Internet Computing, 2003, 7: 40–48
[31]H. Y. Paik, B. Benatallah, F. Toumani, “Towards self-organizing service communities,” IEEE transactions on systems, man and cybernetics, Vol. 35, No.3, 2005, pp.408-419.
[32]J. Bentahar, Z. Maamar, D. Benslimane, P. Thiran, “An Argumentation Framework for Communities of Web Services,” IEEE Intelligent Systems, Vol. 22, No.6, 2007, 75-83.
[33]X. Liu, G. Huang, H. Mei, “A Community-Centric Approach to Automated Service Composition,” Science in China Series F: Information Sciences, Vol.53, No.1, 2010, pp. 50-63.
[34]X. Liu, G. Huang, W. Pei, H. Mei, “Discovering Homogeneous Web Service Community in the User-Centric Web Environment,” IEEE Transactions on Services Computing, Vol. 2, No.2, 2009, pp.167-181.

Yuanbin Han is currently a PhD student in Tianjin University, China. His current research interests include service computing and semantic web.

Shizhan Chen received his PhD from Tianjin University in 2010 and is a lecturer at Tianjin University, China. His current research interests include service computing and SOA.

Zhiyong Feng is a full tenured professor and head of the School of Computer Science and Technology, Tianjin University, Tianjin, China. His current research interests include knowledge engineering, service computing, and security software engineering.