

Measuring Credit of Web Service

Bin Xu, Jiayang Ge, Tonglee Chung

Department of Computer Science and Technology

Tsinghua National Laboratory for Information Science and Technology

Tsinghua University

Beijing, China

xubin@tsinghua.edu.cn, {gerjiayang, tongleechung86}@gmail.com

Abstract—Service selection has been a critical concern for constructing a distributed service-based system. So far, research efforts on service selection are mainly based on the criteria of functionality and QoS (Quality of Service). However, the credit of Web service represents the reputation and importance of a Web service, which also provides a key criterion for selecting service. Then the challenge is how to measure the credit of Web service. In this paper, we propose a method for measuring the credit of service through service network. First, we propose a model for constructing exact service network, which is effectively applied to different type of services like WSDL and Web API. Next, an approach is proposed to depict the credit of service from three aspects: popularity, influence and authority. We use about 15000 services from Seekda and ProgrammableWeb in the experiments; the results show that our approach can rank the credit of service effectively, and play a positive role in measuring the credit of a service.

Keywords—credit of service; service network; service selection;

I. INTRODUCTION

Web services are self-contained, modular applications that can be described, published, located, and invoked over a network [1]. With the development of Service-Oriented Architecture (SOA), more and more services are continuously emerging on the Internet. Therefore, service selection is a critical concern for constructing a distributed service-based system. There are existing solution for selecting a Web service based on services' functionality and QoS. Universal Description Discovery and Integration (UDDI) provides a language to describe service features that can help user to carry service location and selection as well as matchmaking according to the functionality of services [2][3]. Many existing approaches provides algorithms to assist QoS-based service selection efficiently and effectively [4][5][6]. Beside these bases, credit of service is a new basis in service selection.

What is credit of service? Credit of service is proposed to evaluate the reputation of a service via service network. Measurement of credit of service in a certain service network is like the assessment that a community gives to a person. A good credit is commendation or honor given to service that provides a trustworthy basis for user when selecting service. Credit of service is an important parameter to

be considered when selecting atomic services for stable platform development. It guarantees that you are getting a valuable service and provider never deprecated which is the key of credit.

Let us use an example about credit from everyday life. What will we consider when selecting personal computer (see Figure 1)? In general, the primary decision criteria for selecting a personal computer are: computer specifications, after sale service and brand reputation. Intuitively, a person puts brand reputation last but in reality everybody cares about this. Recently, Apple's reputation is the best among top brands and people selecting personal computer are more or less comparing Apple with others. Brand reputation guaranteed the quality of the products. In contrast with service selection, service's functionality is like computer specification, quality of service is like after sale service. Credit of service is quite like brand reputation, which is based on factors from the service activities and relationships.

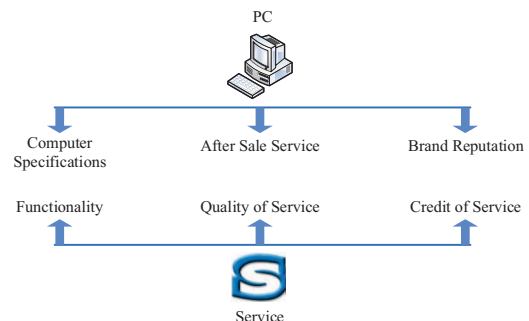


Figure 1. Selecting Personal Computer versus Service

Credit of service gives a whole overview about services' reputation. When selecting service, our approach ranks all the services according to its credit, which makes service selection more trustful. Not like the functionality and QoS which are features from the perspective of isolated service, credit of service is from the perspective of connected services. We measure the credit of service from its ecosystem—service network, which is a novel and proper way to assess

the services for service selection.

In this paper, we put forward a method for constructing exact service network model. The perfect-match model can be effectively applied to different types of services. Then we measure credit of service through service network by calculating service activities and relationship. The findings of this paper include: effectively providing the basis for user to select service. Our measuring approach plays a positive role in measuring the credit of a service. Our contributions are:

- A novel basis to select service: credit of service. A new alternative for user to make selection. It implies services' functionality and non-functional detail and considers service matchmaking or cooperation more.
- An approach to quantify credit of service. The method of measuring is realized through service network, analysis service activities and relationship, finally ranking of each service through credit.

The remainder of this paper is structured as follows. In section 2, we presented the formalization of service network, credit of service and measuring. Section 3 described the model for constructing service network in different service type. The experiment is shown in section 4. Section 5 discusses related works. Finally, we conclude the article in section 6 and discuss future work.

II. CREDIT OF SERVICE IN SERVICE NETWORK

Service network reflects matchmaking and cooperation between services. When measuring credit of service through service network, there are two main steps. The first step is constructing exact service network. Constructing an exact service network will affect the result of measuring. The second step is measuring credit of service, which is methodology to quantify credit of service. In this section, we will present the definition of service network, and describe credit of service with measure method.

A. Service Network

A service network is a representation of a set of service where some pairs of service are connected by links. The interconnected services are represented by nodes, and links that connect some pairs of nodes are called edges.

The edges may be directed or undirected. Service composition technique by exploiting matchmaking between input/output parameters of services is called directed service network. In contrast, there is an edge between two services if they appear in the same composition together.

The nodes may be service operation or API service. In general, a WSDL service element consists of operations, and a Mashup is a combination of several APIs.

A service network is an ordered pair $SN = \langle V, E \rangle$ comprising a set of nodes V together with a set of edges E . We represent a set of nodes V as S_n and an edge E represent the relation of service between S_i and S_j as $S_i \rightarrow S_j$.

B. Credit of Service and Measuring

Measuring credit of service by service activities and relationship factor is inspired by the human society, where service activities represents a person's social actions, that actions may be to join a club or to know someone. Service relationship represents a person's social interaction, that includes any relationship between two or more individuals. These two factors are essential representative of person behavior in a community, we can rely on these factors to assess a person's interpersonal relationship.

The credit of service is the reputation of a service. It is commendation or honor given to service, which represents popularity, influence and authority of service within a certain service network. Figure 2 present an example of service activities and relationship in a certain service network.

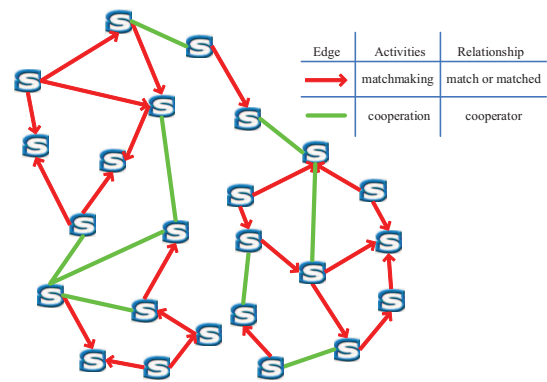


Figure 2. Service Activities and Relationship

Definition 1 (Credit of Service) Services' credit is measured with service activities and relationship factors. It is formalized as

$$Credit = factor(activities) + factor(relationship).$$

The definition shows that credit of service is determined by the service activities and relationship factor. Generally speaking, service activities and relationship in service network is the matchmaking or cooperation between services via network models.

1) *Service Activities*: In a service network, service activities refer to actions which are calculated by reaction of isolated services. The types of service activities involve service matchmaking and cooperation. Therefore, service activities represents the number of links upon a node in service network, and is referred to as degree. In the case of a directed network, there are two separate measures of degree, namely in-degree and out-degree (Figure 3). Accordingly, in-degree is a count of the number of ties directed to the node and out-degree is the number of ties that the node directs to others. The in-degree and out-degree can be interpreted as a form of popularity and gregariousness respectively.

Daniel [7] focus on the influence of social trust and rating mechanisms based on the concept of hubs and authorities in Web-based environment. Furthermore, service activities also indicate the distance metric of service between services in service network, which is referred to as closeness. Closeness can be regarded as a measurement of the range of activities.

Formula (1) gives out the definition of activities factor $factor(act)$. We represent in-degree, out-degree and closeness as $auth$ (authority), hub and clo respectively, and we use the weight w to increase/reduce the significance of corresponding parameter.

$$factor(act) = auth + (w * hub) + (w * clo) \quad (1)$$

We formalize the calculation of authority/hub score [8] of each node as

$$auth(S_i) = \sum_{j=1}^n hub(S_j) \quad (2)$$

where n is the total number of nodes connected to service S_i and S_j is a node connected to service S_i . Formula (2) describes the authority score of a node which is the summation of hub scores of all nodes that point to it.

$$hub(S_i) = \sum_{j=1}^n auth(S_j) \quad (3)$$

where n is the total number of nodes connects to service S_i and S_j is a node connected to service S_i . Formula (3) describes the hub score as the summation of the authority scores of all its linking nodes.

We formalize the calculation of closeness score [9][10] of each node as

$$col(S_i) = \frac{1}{\sum_{S_j \in S_n} d_{SN}(S_i, S_j)} \quad (4)$$

where $S_{SN}(S_i, S_j)$ is the distance between service S_i and S_j in service network SN , and S_n is the set of service.

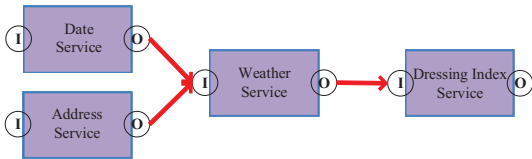


Figure 3. Service Activities and Relationship in Weather Service Network: Weather Service's in-degree is equals 2, out-degree is equals 1 and the number of service activities is 3. Date Service and Address Service has direct relationship to Weather Service, and indirect relationship to Dressing Index Services.

2) *Service Relationship*: In service network, service relationship is the relations between two or more isolated services. There are two type of relationship: direct and indirect relationship (Figure 3). We use service relationship to measure the influence of a service in a service network. To the best of our knowledge, eigenvector centrality is a measure of the influence of a node in a network [11][12]. We use eigenvector centrality to measure most influential service(s) within a service network. Furthermore, to calculate service relationship also indicates the number of times a service acts as a bridge along the shortest path between two other services, also referred to as betweenness.

Formula (5) gives the definition of relationship factor $factor(rela)$. We represent eigenvector centrality and betweenness as eig and $betw$ respectively, we use the weight w to increase/reduce the significance of parameters.

$$factor(rela) = eig + (w * betw) \quad (5)$$

Eigenvector centrality is a relative score recursively defined as a function of the number and strength of connection to its neighbors and those neighbors' centralities. Let A denote the adjacency matrix of a service network, $\Gamma(S_i)$ denote the neighborhood of S_i and the set of nodes S_i is connected to directly. Eigenvector centrality of S_i is calculated as

$$eig(S_i) = \frac{1}{\lambda} \sum_{j \in \Gamma(S_i)} S_j = \frac{1}{\lambda} \sum_{j=1}^n A_{i,j} S_j \quad (6)$$

where λ is a constant, formula (6) can be rewritten in vector formula (7) where $S_i = S_1, S_2, S_3, \dots, S_n$ is the vector of eigenvector centrality scores of all nodes. The vectorized representation is

$$S_i = \frac{1}{\lambda} A S_i \quad (7)$$

where λ is an eigenvalue and S_i is the corresponding eigenvector of matrix A . So several eigenvector pairs exist for an adjacency matrix A .

We formalize the calculation of betweenness score [10][13][14] of each node as

$$betw(S_k) = \sum_{S_i \neq S_k \neq S_j \in S_n} \frac{\sigma_{S_i S_j}(S_k)}{\sigma_{S_i S_j}} \quad (8)$$

where $\sigma_{S_i S_j}$ and $\sigma_{S_j S_i}$ is the number of shortest paths from S_i to S_j and $\sigma_{S_i S_j}(S_k)$ is the number of shortest paths from S_i to S_j that S_k lies on.

III. CONSTRUCTING EXACT SERVICE NETWORK

A service network comprises of a group of services (operations or APIs) and matchmaking or cooperation that indicates they provide overlapping or identical functionalities. Web service matchmaking is the process of matching

two or more services together, usually for the purpose of composition. Many Web service matchmaking solutions are based on keyword matching which is supported by category browsing of UDDI. However, the keyword based matching cannot fully capture real functions of Web service. To address this limitation, we developed a method to assess the match-degree of Web service for matchmaking. The matchmaking process is based on a lightweight semantic comparison of signature specifications in WSDL. Chantal [15] study about parameter and operation network structure by highlighting topological differences, and give guidelines on the use of those features to guide a composition search process. In this paper, we choose WordNet and Edit Distance based matching schemes and applied perfect-match model for constructing exact service network. Exact service networks guarantees any two service operation which are linked together are matched, and users can compose them according to their requirement. We also applied perfect-match model in Mashup ecosystem and the experimental results show good performance.

Definition 2 (Perfect-Match) Given a set of services $\{S_n\}$, connecting of source service S_i to target S_j conform to $type_match(S_i.type, S_j.type)$, where $S_i.type$ and $S_j.type$ is the set of parameters type and APIs respectively. There are two cases: (1) $S_i.type \subset S_j.type$, we refer to as partial match, and (2) $S_i.type = S_j.type$, we refer to as full match. In these two case $S_i \in S_n$ and $S_j \in S_n$ are a set of vectors, it may consist of operations' parameters or APIs list.

Definition 2 is the definition of perfect-match model which is used for constructing exact service network in both WSDL service and Mashup ecosystem. Figure 4 shows the application of perfect-match model in WSDL service and Mashup ecosystem.

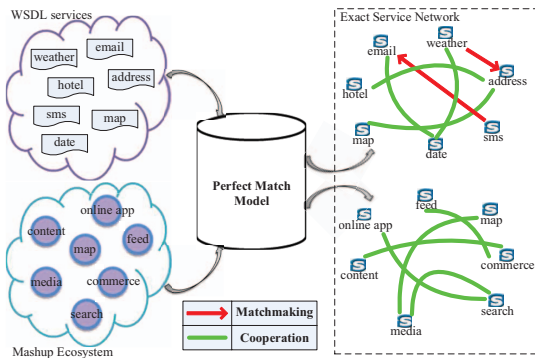


Figure 4. Application of Perfect Match Model

A. WSDL Service

We capture the Input/Output descriptions which belong to each operation that contains a set of parameters. For each pa-

rameter refers to name and type which defines Input/Output messages. We deal with word-splitting using space, capital letter and special character (such as “_”). As for stop-word removal, we considered WSDL stop-words especially. For example, “parameters”, “session”, “response”, “id” which frequently appears in parameter name and has no semantic meaning. We also considered word stem, for example the stem of the noun “titles” is “title”.

1) *Words Similarity*: Through the preprocessing of data, we found some character set are terminology words or acronym. Such character set are referred to as terms, the others we referred to as words. Terms and words are two disjoint set. We adopted WordNet and Edit Distance for calculating words similarity. Because the term doesn't exist in the lexical database of WordNet, so we deal with terms similarity calculation using Edit Distance. When calculating similarity between word and term, we assume the similarity is zero. We give out the function of calculating Words Similarity $wSim$, such that

$$wSim(x, y) = if \begin{cases} WordNet & x \in word \& y \in word \\ 0 & x \in word \& y \in term \\ EditDistance & x \in term \& y \in term \end{cases}$$

2) *Operations Match Degree*: Let V_1 and V_2 be the word vector of parameters of operations op_1 and op_2 . Formula (9) gives out the definition of Operations Match Degree $opMatch$.

$$opMatch(op_1, op_2) = \frac{wSim(V_1, V_2)}{|V_1||V_2|} \quad (9)$$

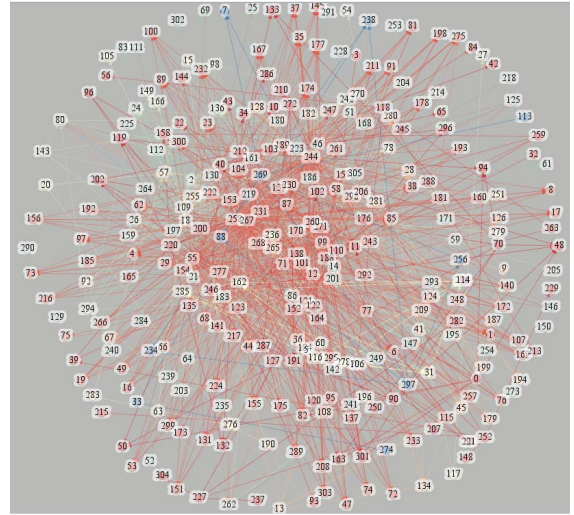


Figure 5. WSDL Services Operation Network

Based on perfect-match model, we calculate match degree for each pair of operations in our data set (Table I). The result in Figure 5 shows, the WSDL service operation

network is a directed network that comprised of 306 service operations as nodes and 1157 matchmaking as edges (see more details on <http://sminer.org/analysis.jsp>). In order to construct exact service network, we filtered out match degree that is lower than 0.98.

B. Mashup Ecosystem

In the Mashup ecosystem, a Mashup consists of a combination of mapping APIs (e.g., Google Maps) and Atom data APIs (e.g., from CNN) that provides a new service that displays a listing on the map. The application of Mashups adds a new value to the combined Web APIs that is not conceived individually. Therefore, a combination of mapping APIs represented two or more API cooperating in a Mashup.

We define the APIs mapped in the Mashup as *composition-service network (CSN)*, and formalize the *CSN* as a $m \times n$ matrix, where m is the number of compositions and n is the number of services [16][17]. *CSN* is formalized as

$$CSN = [CS_{ij}], 0 \leq i \leq m, 0 \leq j \leq n \quad (10)$$

where $CS_{ij} = 1$ if composition S_i invokes service S_j .

We get an additional networks: a *service-service network (SSN)* in which two services are connected if they appear in the same composition. *SSN* is formalized as

$$SSN = [SS_{ij}] = CSN^T \cdot CSN, 0 \leq i, j \leq n \quad (11)$$

where SS_{ij} is the number of compositions in which service S_i and service S_j are invoked, SS_{ii} is the number of composition where service S_i is invoked.

We calculate *SSN* with the perfect-match model in our data set (Table I). The result Figure 6 shows this Mashup APIs network is an undirected network that comprised 999 APIs as nodes and 3147 cooperation as edges (see more details on <http://sminer.org/analysis.jsp>).

IV. EXPERIMENT

Our experiment measures credit of service. First, we describe our data set, then present our report related with service activities and relationship. Finally, we described the rank of services' credit and compare with PageRank and hot service/Mashup which calculate the number of its operations/APIs.

A. Data Set

We obtained the service data set from Seekda. Seekda is the most comprehensive search engine for Web service until 2011. The data set which we had collected is describes as WSDL file. Each service contains information such as name, description and provider information. According to WSDL definitions, the action supported by the service is defined as operation. Our data set contains more than 88 thousand operations which belong to over 9 thousand services. Furthermore, we obtain ProgrammableWeb data

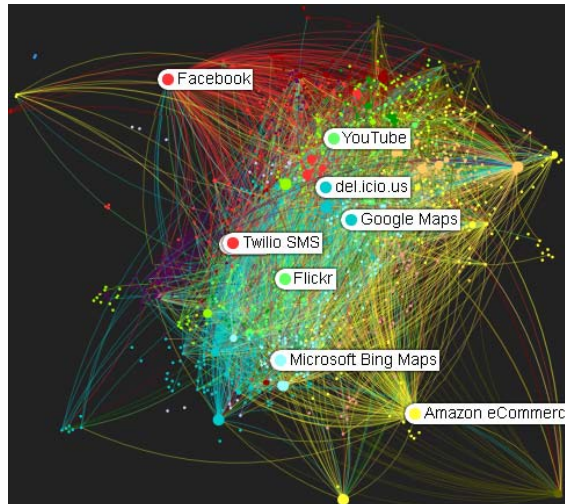


Figure 6. Mashup APIs Network

regarding Mashup. In this data set, each Mashup contains information such as Mashup name, provider, description, tag and API list. We get a collection of 6726 Mashup, and 999 API from 1133 API providers (Table I).

Table I
DATA SET STATISTIC

Number of WSDL service	9256
Number of service operation	88383
Number of Mashup	9726
Number of API	999
Number of API provider	1133
Average number of service operation per WSDL	9.5
Average number of API per Mashup	5.8
Average number of API provider per API	1.0

B. Statistic Report

In our experiment, we calculate service activities and relationship using service authority, hub, closeness, eigenvector centrality and betweenness in service network. The statistics and data set are shown in Figure 7. In the result, we set our weight w to 0.85. From the observation, we can see the difference of directed and undirected network. In Figure 7 (a) and (b), the score of authority and hub distribution is the same as in undirected network, which is presented in APIs network (see more details on <http://sminer.org/analysis.jsp>). Generally speaking, the higher the services' authority and hub the more active the role is in a certain network. Closeness measures services' ability, that is how many other services it can reach through relatively shorter paths. Eigenvector centrality measures influence of a node in a network. A service with higher betweenness value imply

that it will more frequently reside on shortest paths between any other two services.

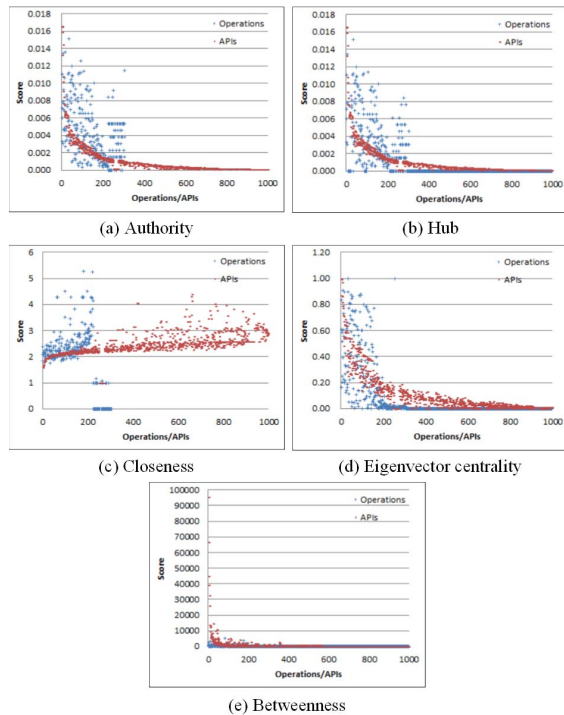


Figure 7. Statistic Report and Data Set: The quantity of operations and APIs vs the value of authority, hub, closeness, eigenvector centrality and betweenness. Note that there are 306 operations and 999 APIs.

C. Ranking

The aim of this paper is provide a basis for user to select service, our result of measuring credit of service is a score between 0 and 1. The score will be basis for user to make a choice in service selection. To reduce detailed measures to a sequence of ordinal numbers, rankings make it possible for user to quickly evaluate complex information and are applicable to Web service search engine. In our experiment, we compare credit of service rank with PageRank and ranking by composition times of APIs, which statistic comes from composited time in Mashup ecosystem. In Table II , we listed top 20 service operation ranking. We compared credit of service rank with PageRank and found that our approach effectively render ranking for each operation. In Table III, we listed 20 APIs ranking, through this experiment we can see high composition times in Mashup ecosystem didn't result in high score of credit. So credit of service not only measures the popularity from pure quantity but also from influence and authority, namely service activities and relationship. However, when user select a service they should compare credit of service in the same category. For example, the credit of Facebook is higher than Twitter in social network service category in our data set. In addition,

our result of measuring credit of service plays a positive role. For example, BibleData is equal to CivicAddressPointRange score in PageRank but different in credit of service. In conclusion, measuring the credit of service considered various factor, namely service activities and relationship. Our approach can effectively measure each service's credit and hold discriminative power contrast to PageRank.

V. RELATED WORK

The main contribution of this paper is proposing a new basis for user to select service: credit of service. We try to use network-based method to measure credit of service. Measuring credit of service is different from trust, reputation and QoS based service selection.

A trust and reputation mechanism is a mechanism using consumers' feedback to identify good services from bad ones [18]. Recently trust and reputation mechanisms have also been applied to Web service systems [19]. Maximilien [20] proposed a framework to achieve service selection in a manner that considers the preferences of service consumers and the trustworthiness of providers. Sandip [21] developed a trust mechanism, which determines the number of users to query given a target guarantee threshold likelihood of choosing high-performance processors. Stefania [22] proposed a methodology for addressing trust in Semantic Web Services (SWS) based applications and presented a prototype of a trust-based selection in IRS-III, based on WSTO which an ontology for managing trust in SWS. Holger [23] used trust and reputation mechanisms in clients to infer expectations of future providers' behavior. Trust and reputation mechanisms are the public's opinion about the character or standing of a Web service. Furthermore, QoS has been discussed a lot in literatures and seen as the major criteria for selecting Web services. QoS is defined in various ways and measured by different metrics, such as performance, reliability, and scalability. In our previous work [5], we have solved QoS-driven semantic Web service composition problem and we can handle a large-scale service composition in a very short time. Zeng [4] proposed a global planning approach for QoS-driven service selection in composed service. Tao [6] proposed a broker-based architecture and several efficient heuristic algorithms to find the optimal service in composed service under QoS constrains. Alrifai [24] proposed an approach combining global optimization and local optimization to efficiently find optimal services.

However, existing solution of selecting a Web service is decided by general service properties. These properties are either service functionality or non-functional detail with no service activities and relationship. Service activities and relationship are measure from service network which functionality and QoS are interdependent with other services. This solution connect isolated services to become service network. To measure service activities and relationship in a certain service network namely measure credit of service.

Table II
TOP 20 OF SERVICE OPERATION RANKING

Operations Name	Ranking		Credit of Service	PageRank Score
	Credit	PageRank		
AllegroWebApiService	1	1	1.000000000	0.016954217
ZipCode	2	2	0.442756532	0.011247000
IncarflashwSDL	3	3	0.410398340	0.009889000
BibleData	4	4	0.407540500	0.007988000
CivicAddressPointRange	5	4	0.369828503	0.007988000
MediaFinanzService	6	5	0.326481870	0.005878000
IGN.Boards.Web.Common.Services.TopicServices	7	6	0.312228206	0.005515000
meltWS	8	7	0.310568023	0.005168000
CodeGenService	9	8	0.304677320	0.004750000
ACEServices	10	9	0.283446541	0.004729000
TransformationService	11	9	0.271729108	0.004729000
ResearchService	12	9	0.251880229	0.004729000
TradeService	13	9	0.236169382	0.004729000
CodicEngine.NetWebService	14	10	0.229611246	0.004471000
stock	15	11	0.196392244	0.004304000
OfferService	16	11	0.194551619	0.004304000
ISBN	17	11	0.192574814	0.004304000
currencyService	18	11	0.177445064	0.004304000
BSEWebService	19	11	0.156880210	0.004304000
LicenceProtector-ASP.Net-WebService	20	12	0.144304687	0.004300867

Table III
APIs RANKING

APIs Name	Ranking			Credit of Service	PageRank Score	Composition Times
	Credit	PageRank	Composition Times			
GetGlue	1	1	211	1.000000000	0.021161	7
Singly	2	2	679	0.7002060014	0.017605	1
TwitPic	3	3	69	0.4727866785	0.014588	29
Dropbox	4	4	188	0.4102277956	0.013424	8
eSideWalk	5	5	679	0.3407436114	0.013104	1
All for Good	6	6	679	0.2743645233	0.010226	1
SimpleAPI	7	25	391	0.1544562794	0.004962	3
HotUKDeals	8	7	679	0.1425589540	0.008819	1
deCarta	9	10	679	0.1386090260	0.007118	1
CNET	10	8	136	0.1300318497	0.007750	12
Biblia.com	11	44	325	0.1122097349	0.003505	4
BioID Web Services	12	35	490	0.1013694940	0.003716	2
Google AJAX Language	13	9	75	0.0990936112	0.007129	26
GeoNames	14	42	23	0.0901092375	0.003536	81
Follow The Money	15	22	325	0.0898884127	0.004979	4
Flickr	32	114	4	0.0446924853	0.002077	614
Amazon eCommerce	303	385	5	0.0006912940	0.000734	412
Facebook	343	381	6	0.0003670772	0.000741	371
Google Maps	345	606	1	0.0003664441	0.000415	2394
Twitter	366	432	2	0.0002736660	0.000635	742

The credit of service offers a promising way to solve the Web service selection problem, that not only considers ser-

vices' functionality and non-functionality but also considers services matchmaking and cooperation.

VI. CONCLUSION

In this paper, we propose a novel basis for user to select service: credit of service. A service with high credit score of credit represents popularity, influence and authority of service. Based on constructing exact service network model, our measurements in credit of service implies services' functionality and non-functional detail and consider more on service matchmaking or cooperation. In addition, we present the methodology to quantify credit of service. Our measure approach plays a positive role in measuring credit of service. The experiment shows credit of service not only measures the popularity from pure quantity but also from influence and authority, namely service activities and relationship. In future work, based on this research, we will apply this method to our service search and mining systems (<http://sminer.org/index.jsp>) in service recommendation section.

ACKNOWLEDGMENT

This work is supported by China National Science Foundation under grant No.61170212 and China National High-Tech Project (863) under grant No.SS2013AA010307. Beijing Key Lab of Networked Multimedia also supports our research work.

REFERENCES

- [1] Jinghai Rao and Xiaomeng Su. A survey of automated web service composition methods. In *SWSWPC'04*, pages 43–54, 2004.
- [2] Yutu Liu, Anne H. H. Ngu, and Liangzhao Zeng. Qos computation and policing in dynamic web service selection. In *WWW (Alternate Track Papers & Posters)'04*, pages 66–73, 2004.
- [3] Massimo Paolucci, Takahiro Kawamura, Terry R Payne, and Katia Sycara. Importing the semantic web in uddi. In *Web Services, E-Business, and the Semantic Web*, pages 225–236. Springer, 2002.
- [4] Liangzhao Zeng, Boualem Benatallah, Anne H. H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang. Qos-aware middleware for web services composition. *IEEE Trans. Software Eng.*, pages 311–327, 2004.
- [5] Bin Xu, Sen Luo, Yixin Yan, and Kewu Sun. Towards efficiency of qos-driven semantic web service composition for large-scale service-oriented systems. *Service Oriented Computing and Applications*, 6(1):1–13, 2012.
- [6] Tao Yu, Yue Zhang, and Kwei-Jay Lin. Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Transactions on the Web (TWEB)*, 1(1):6, 2007.
- [7] D. Schall and F. Skopik. Mining and composition of emergent collectives in mixed service-oriented systems. In *Commerce and Enterprise Computing (CEC), 2010 IEEE 12th Conference on*, pages 212–219, Nov 2010.
- [8] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *JOURNAL OF THE ACM*, 46(5):604–632, 1999.
- [9] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.
- [10] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.
- [11] Muhammad Usman Ilyas and Hayder Radha. Identifying influential nodes in online social networks using principal component centrality. In *ICC'11*, pages 1–5, 2011.
- [12] Phillip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, pages 191–201, 2001.
- [13] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.
- [14] J. M. Anthonisse. The rush in a directed graph. 1971.
- [15] Chantal Cherifi and Jean Francois Santucci. On topological structure of web services networks for composition. *CoRR*, pages –1–1, 2013.
- [16] Wei Tan, Jia Zhang, and Ian T. Foster. Network analysis of scientific workflows: A gateway to reuse. *IEEE Computer*, pages 54–61, 2010.
- [17] Keman Huang, Yushun Fan, and Wei Tan. An empirical study of programmable web: A network analysis on a service-mashup system. In *ICWS'12*, pages 552–559, 2012.
- [18] Yao Wang and Julita Vassileva. A review on trust and reputation for web service selection. In *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on*, pages 25–25. IEEE, 2007.
- [19] Yao Wang and Julita Vassileva. Toward trust and reputation based web service selection: A survey. *International Transactions on Systems Science and Applications*, 3(2):118–132, 2007.
- [20] E Michael Maximilien and Munindar P Singh. Toward autonomous web services trust and selection. In *Proceedings of the 2nd international conference on Service oriented computing*, pages 212–221. ACM, 2004.
- [21] Sandip Sen and Neelima Sajja. Robustness of reputation-based trust: boolean case. In *AAMAS'02*, pages 288–293, 2002.
- [22] Stefania Galizia, Alessio Gugliotta, and John Domingue. A trust based methodology for web service selection. In *ICSC'07*, pages 193–200, 2007.
- [23] Holger Billhardt, Ramn Hermoso, Sascha Ossowski, and Roberto Centeno. Trust-based service provider selection in open environments. In *SAC'07*, pages 1375–1380, 2007.
- [24] Mohammad Alrifai and Thomas Risse. Combining global optimization with local selection for efficient qos-aware service composition. In *Proceedings of the 18th international conference on World wide web*, pages 881–890. ACM, 2009.