2013 IEEE 10th International Conference on Services Computing

A Technique of Analyzing Trust Relationships to Facilitate Scientific Service Discovery and Recommendation

Jia Zhang[1], Petr Votava[2,3], Tsengdar J. Lee[3], Shrikant Adhikarla[1], Isaraporn Kulkumjon (Cherry)[1], Matthew Schlau[1], Divya Natesan[1], Ramakrishna Nemani[2]
[1]Carnegie Mellon University – Silicon Valley, USA
[2]NASA Ames Research Center, USA
[3]Science Mission Directorate, NASA Headquarters, USA
jia.zhang@sv.cmu.edu, petr.votava-1@nasa.gov, tsengdar.j.lee@nasa.gov, rama.nemani@nasa.gov

***Abstract*—**Most of the existing service discovery methods focus on finding candidate services based on functional and non-functional requirements. However, while the open science community engenders many similar scientific services, how to differentiate them remains a challenge. This paper proposes a trust model that leverages the implicit human factor to help quantify the trustworthiness of candidate services. A hierarchical Knowledge-Social-Trust (KST) network model is established to draw hidden information from various publication repositories (e.g., DBLP) and social networks (e.g., Twitter). As a proof of concept, a prototyping service has been developed to help scientists evaluate and visualize trust of services. The performance factor is studied and experience is reported.

I. INTRODUCTION

Software as a Service (SaaS) has been widely considered as a promising new software delivery and provisioning technique to support modern software engineering. While software is published as programmable services (components) on the Internet, software developers can leverage appropriate services as components to build new value-added software, faster than before. In recent years, SaaS has been applied to scientific world and has greatly facilitated scientific application and workflow design and development [1]. However, the *open* feature of the science community has led to many scientific services published onto the Internet with similar functionalities. How to help a scientist select appropriate services remains a challenge.

The Services Computing community has been working on the topic of service discovery for a decade. The last decade has witnessed a holistic set of solutions proposed. Most of the methods analyze either syntactic or semantic meanings of candidate services, and conduct matchmaking processes between the candidate services and the desired requirements before making a selection. In addition, not only functional requirements but also non-functional requirements (QoS features) are used to help select proper services. However, similar to deciding business partners, the human factor may also play an important role. For example, a service published by a reputable research group will be more likely to be considered. For another example, if a scientist knows a collaborator has been using a specific service for a couple of years, she would trust the service more. Such scenarios show that human trust over a candidate service is important for effective service selection.

In this project, we have conducted a comprehensive study of the human factor in scientific service selection. The technical issues on which we targeted are two-fold:

Where can we draw hidden knowledge to help decide the trustworthiness of a candidate service?
How to quantify the trustworthiness of a service?

As illustrated in Fig. 1, our overall idea is to turn trust on services into trust on human (e.g., corresponding service developers). We have established a Knowledge-Social-Trust (KST) network model to compute the trustworthiness of a service developer regarding specific user requirements and context. Our trust model calculates human trust as the summation of two main components which are "knowledge factor" and "social factor." The knowledge factor provides a score of human's reputation which is evaluated from users' expertise, work experience, educational background, and so on. The social factor is evaluated from social interaction and relationship which could cover co-authorship, colleague, friendship, and online conversation like facebook messages, or twitter messages (called tweet). As shown in Fig. 1, this project has demonstrated that a lot of such knowledge can be obtained from various data sources including publication repositories and social networks.

As a newly emerged type of social network, Twitter (http://twitter.com/) is increasingly extending its role from a communication channel into an important platform for seeking and sharing real-time information – a social sensor network. In this project, we have demonstrated that Twitter may contribute to service selection beyond social

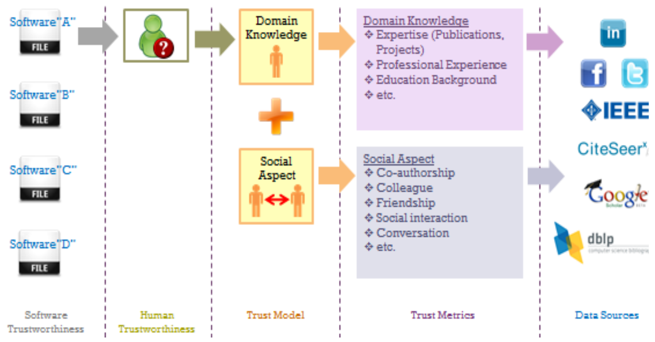


Fig. 1 Trust-empowered service discovery.

978-0-7695-5026-8/13 $26.00 © 2013 IEEE
DOI 10.1109/SCC.2013.104

relationships, including knowledge reputations and service usage history.

Our previous work examines workflow repository myExperiment (http://www.myexperiment.org) and have extracted service usage history information [2], e.g., what types of workflows in which a service is usually used; and how different services are usually used together. In this work, we further demonstrated that hidden knowledge related to human factor can also be extracted to support service discovery and selection.

Our work will potentially be applicable and contribute to various service repositories. Centralized "service yellow page-like" UDDI registries [3] are going out of date, due to their tight binding to SOAP/WSDL services and their over standardization. In recent years, REpresentational State Transfer (REST) service, a light-weight HTTP Request/Response-based service style, has rapidly gained significant momentum [4]. As a result, many REST service-oriented registries have been developed, such as the ProgrammableWeb (http://www.programmableweb.com). As the number of services accumulates at ProgrammableWeb, it is important to facilitate users in querying and finding interested services [5]. Our previous work leveraged the domain knowledge to extend the Support Vector Machine (SVM) technique to verify and support automatic service categorization [6]. Our on-going work reported in this paper will help such service repository to provide personalized (trust-empowered) service discovery and recommendation service.

The remainder of the paper is organized as follows. In Section 2, we discuss related work. In Section 3, we introduce our KST network model. In Section 4, we present trust algorithms over the KST network. In Section 5, we present network visualization. In Section 6, we present prototyping system implementation. In Section 7, we draw conclusions and discuss future work.

II. Related Work

In this section, we discuss related work focusing on social networks and archived publication records.

A. *Social networks*

A number of social networks have emerged in the last 10 years. Some of them focus on professional relationships, represented by LinkedIn (http://linkedin.com). Some other social networks focus on personal relationships, represented by facebook (http://facebook.com). Since these traditional social networks have been well studied in the literature, in this paper we focus on a newcomer of the family of social networks, the Twitter (http://twitter.com/).

Twitter has been used as a social sensor network, a real-time data source supporting a variety of information seeking and sharing applications. Each tweet allows to carry up to 140 characters, derived from the capacity constraint of a text message on a cell phone. Therefore, tweet processing efforts can be controllable and tweets can quickly disseminate information. Two major application areas have been identified: real-time analysis of scenario-based information dissemination trend [7-10] and recommendation-oriented applications [11-14].

For the first application area, Twitter is envisioned as *social sensors* [7]. For the second application area, Twitter is used to enhance recommendations. Such related work has proved the effectiveness of leveraging Twitter to facilitate information dissemination and recommendation. However, to our best knowledge, twitter has not been used to facilitate service discovery to date. Particularly, we propose a novel service recommendation method that leverages both real-time society opinions and past experiences.

B. *Archived publication records*

DBLP is a known metadata repository of publications in computer science (http://dblp.uni-trier.de/), including journal and conference papers, book chapters, theses, etc. To date, it has indexed more than 2 million articles. DBLP stores publication record metadata in an XML file, together with a published Data Type Document (DTD) file describing its grammar. For example, for a journal paper, DBLP stores its paper title, authors, journal name, volume, number, month, year, and pages.

Over 400 publications have reported various types of statistical studies over the DBLP dataset, including co-authorship analysis [15, 16], community analysis [17], field analysis [18, 19], clustering [20, 21], and accessibility analysis [22]. In contrast to the existing work on bibliography datasets, we aim to extract hidden knowledge to help quantify trust relationships among researchers. Particularly, we are interested in two types of information: a person's knowledge (reputation) level in a given field; direct or indirect trust relationship due to co-authorship relation.

Sponsored by US NSF, CiteSeer^x (http://csxstatic.ist.psu.edu/) provides a scientific search engine focusing on the literature in computer and information science. CiteSeer^x and DBLP are both known publication metadata repositories. They use proprietary strategies to select indexed publications; thus, their datasets do not completely overlap. In this paper, as a proof of concept, we used the DBLP dataset because of our familiarity with its data format. We plan to integrate CiteSeer^x as another publication metadata archive in our future work.

III. KST Network Model

We propose a hierarchical network model to evaluate and select trustworthy service candidates. From bottom up as shown in Fig. 2, the model comprises three layers: a knowledge network (K-Net), a social network (S-Net), and a trust network (T-Net). K-Net aims to help decide the knowledge expertise level of a person; S-Net aims to help decide the social relationship among people; T-Net aims to

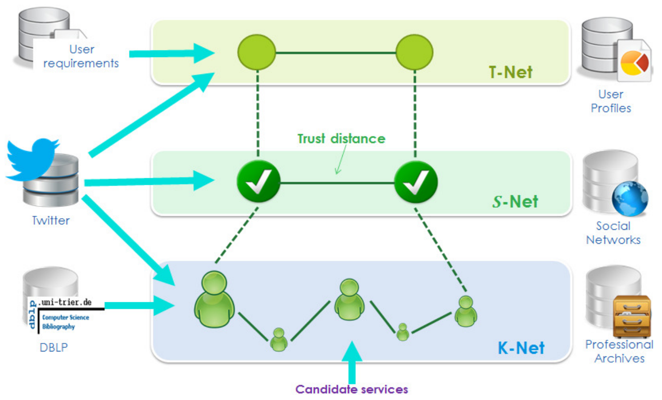


Fig. 2 Hierarchical KST network model.

help decide the trust level among people according to specific user requirements on services.

A. K-Net

K-Net is built upon professional archives, represented by publication record repositories such as DBLP, CiteSeer^x, and IEEE and ACM digital libraries. A person's knowledge reputation over a specific domain (or specific fields or topics) can be represented by an accumulated function of her publication records in the corresponding domain as shown below. For each publication record, several factors impact its significance, including publication channel (C), publication time (T), and similarity between the article content to user query (S). Each factor can be assigned different coefficient weights to indicate the importance of the factor. The sum of all coefficient weights should remain to be 1.

$$k_{p,d} = \sum f(\alpha C, \beta T, \gamma S), \qquad \alpha + \beta + \gamma = 1$$

If a researcher publishes a paper in a reputable channel, it indicates that the researcher has conducted some reputable work in the field. If a researcher publishes constantly in a reputable channel during a time period, it is reasonable to consider the researcher knowledgeable in the field.

A publication channel itself usually carries a reputation in corresponding fields. For example, an IEEE Transactions journal is generally considered as a highly reputable publication channel in the related computer science fields. In computer science, journals/magazines and conferences typically adopt different ranking systems. For journals and magazines, several international organizations strive to identify and index significant journals and magazines in various fields. A well-adopted criterion is the impact factor (IF). Thomson Reuters publishes annual journal citation reports[1], applying citation analysis to assign an impact factor (IF) to "good" journals and magazines. A newly launched journal may apply to Thomas Reuters to be indexed, and it is up to Thomas Reuters to make the decision based on its impact in the domain (e.g., computer science). Another two widely adopted lists are Science Citation Index (SCI) and EI-Compendex, both identifying significant journals and magazines in a number of fields.

In some domains (such as mathematics and economics), conferences are usually counted as premature publication channels. On the contrary, computer science conferences are considered important publication channels. As a matter of fact, top conferences have higher reputations over normal-level journals. One major reason is the unique feature of rapid technology evolution of computer science. Regarding the reputation of a conference, a set of factors are used as criteria. One important factor is the acceptance rate of research papers of a conference. The lower the acceptance rate, in general the higher reputable a conference is. Other factors include the number of submissions, the reputation of the program committee members, the scale of the participation group, and the number of citations received by its published papers.

The degree of knowledgeable in some specific topics (tags, in taxonomy) may need to be tied to a time period. For example, a researcher who published heavily in a field ten years ago may not work on some newly emerged topics in the field.

Even if a person is reputable in a domain in general, the similarity between the content of her published articles and the expected user query may also need to be considered. Consider a scenario when a user query is about analyzing KEGG data in life science. If the developer of a candidate service has published an article in *Science* magazine regarding algorithms of processing KEGG data, then the corresponding service may be granted more attention.

B. S-Net

The social network has an enormous amount of information in various formats. For instance, Twitter is a social micro-blogging tool which can be used to extract information about interests of people in reference to a specific field or context. On the other hand, we have more formal information in the form of research journals and publications like IEEE and ACM. One objective of the project is to identify and analyze such useful information using APIs.

From the publication history, we can evaluate whether a researcher likes to stay with a more stable collaboration group, or participate in various collaboration relations. Exceptions involve students. A faculty member may advise students and co-author with them. Since students keep on graduating, the collaboration relationship between a faculty and a student under his/her advice should not be used to study the stability of collaboration relationships. Under such circumstances, affiliation information will help to answer the question of whether a researcher is a faculty or students. Such information may be derived from university website,

[1] http://thomsonreuters.com/products_services/science/science_products/a-z/journal_citation_reports/

from personal website (for the period when a particular paper is published), and from professional social network such as LinkedIn.

To limit our project within the boundary of information which could possibly be retrieved, we will consider the number of retweet as the trust metric retrieved from the Twitter social network for the knowledge factor. The reason that we focus on retweet rather than tweet is that people tend not to retweet unbeneficial messages or general messages.

Additionally, co-authorship retrieved from DBLP dataset can be considered as one trust metric of the social factor. It could be implied whether person A is trusted by person B or not and vice versa. The simple reason is that if they do not trust each other, they would not have co-authored papers.

C. T-Net

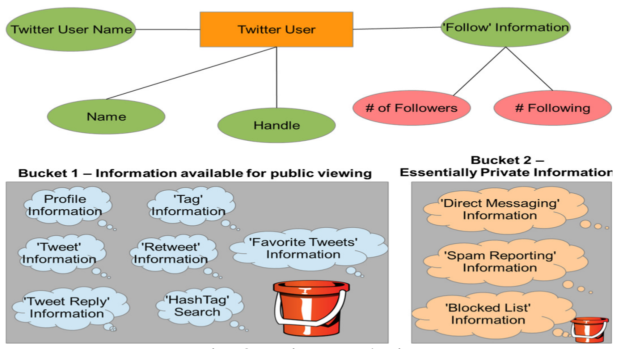
The software trust model is intended to provide a singular quantitative trust value for each software module with respect to any user and context as well. Some contextual factors that decide the relevance of a module include the pertaining domain and time of software creation. The trust model of a software developer may comprise of two aspects: reputation based on knowledge and reputation based on social relationships. Reputation based on knowledge may include the number of relevant publications of the developer, number of followers on twitter and so on, whereas the social relationships can be determined from the list of people who have co-authored with the individual.

IV. KST-Based Trust CalCulation

A. KST-based trust algorithm

As shown in Fig. 2, our trust model calculates human trust as the summation of two main components which are "knowledge factor" and "social factor." Under specific context, the two factors can be assigned different weights.

In this project, data sources that we studied include social networks, such as Facebook, Twitter, and LinkedIn and publication sources like IEEE, Google Scholar, CiteSeer, and Digital Bibliography and Library Project (DBLP). However, we have confronted with many limitations in accessing those data sources. To name a few, all publication sources do not provide open APIs to query their information. Also, the number of query per hour is limited in open APIs provided by many social networks. For example, Twitter limits 150 queries per hour. Finally, some meaningful information may be associated with privacy guard. Fig. 3 illustrates an analysis of the concepts in Twitter dataset. Some information can be retrieved publically (e.g., profile information, tweet information). Some other information is considered private data, such as direct messaging information, and blocked list information. Getting permission from a specific user is required for private data. Consequently, as a proof of concept, we used offline dataset: 5M tweet messages recorded within 1 month


Twitter User Name
Twitter User
'Follow' Information
Name
Handle
# of Followers
# Following
Bucket 1 – Information available for public viewing
Profile Information
'Tag' Information
'Tweet' Information
'Retweet' Information
'Favorite Tweets' Information
'Tweet Reply' Information
'HashTag' Search
Bucket 2 – Essentially Private Information
'Direct Messaging' Information
'Spam Reporting' Information
'Blocked List' Information
Fig. 3 Twitter analysis.

and DBLP publication information which is published during 1936 to 2012. To map users from tweet dataset with DBLP, we wrote code to send REST APIs to acquire twitter users' information including their full names.

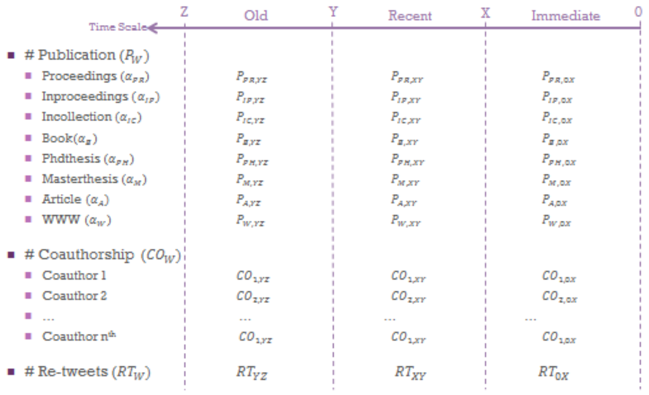
Currently we consider the impact of publication, retweet, and co-authorship to the trust value based on time. These factors are time-dependent because a recent one typically implies the higher trust than a past one. However, such a reason is not logical when considering the number of citations because the paper cited in the past does not mean that it is lower quality than the one which is just cited. Hence, we will not consider time factor for the number of citations. We separate time scale into three parts as shown in Fig. 4: old, recent, and intermediate. "Intermediate" covers the period since now to the past X month; "recent" means the past X month until the past Y month; "old" means the period since the past Y month until the past Z month.

As the impact of trust metrics is different, weight for each trust metric should be assigned. The notation for the weight for each trust metric is described in Fig. 4. The normalized number of publication, retweet, and co-authorship separated by type and time period is illustrated as well.

For each publication type listed in Fig. 4, e.g., article (denoted as A), its trust value is computed as a summation of the normalized number of publication categorized by time period which is multiplied by the time factor:

$$T(article) = t_O P_{A,YZ} + t_R P_{A,XY} + t_I P_{A,0X} \qquad (1)$$

Consequently, we combine every publication type and its



Time Scale	Z — Old — Y	Recent — X	Immediate — 0
# Publication (P_W)			
Proceedings (α_{PR})	$P_{PR,YZ}$	$P_{PR,XY}$	$P_{PR,0X}$
Inproceedings (α_{IP})	$P_{IP,YZ}$	$P_{IP,XY}$	$P_{IP,0X}$
Incollection (α_{IC})	$P_{IC,YZ}$	$P_{IC,XY}$	$P_{IC,0X}$
Book(α_B)	$P_{B,YZ}$	$P_{B,XY}$	$P_{B,0X}$
Phdthesis (α_{PH})	$P_{PH,YZ}$	$P_{PH,XY}$	$P_{PH,0X}$
Masterthesis (α_M)	$P_{M,YZ}$	$P_{M,XY}$	$P_{M,0X}$
Article (α_A)	$P_{A,YZ}$	$P_{A,XY}$	$P_{A,0X}$
WWW (α_W)	$P_{W,YZ}$	$P_{W,XY}$	$P_{W,0X}$
# Coauthorship (CO_W)			
Coauthor 1	$CO_{1,YZ}$	$CO_{1,XY}$	$CO_{1,0X}$
Coauthor 2	$CO_{2,YZ}$	$CO_{2,XY}$	$CO_{2,0X}$
...	...	...	...
Coauthor nth	$CO_{1,YZ}$	$CO_{1,XY}$	$CO_{1,0X}$
# Re-tweets (RT_W)	RT_{YZ}	RT_{XY}	RT_{0X}

Fig. 4 Part of trust factors.

weight with publication weight as the trust value for publication:

$$T(K_{DBLP,Pub}) = \sum_{i\epsilon\{pub\_types\}}\left[\alpha_i \sum_{j\epsilon\{I,R,O\}} t_j P_{i,j}\right] \quad (2)$$

The trust value for citation is derived from the number of citation multiplied by the normalization factor:

$$T\left(K_{DBLP,Cite}\right) = \sum_{i\epsilon\{publications\}}[\eta_i C_i] \quad (3)$$

The trust value for retweet part is a summation of the normalized number of retweet messages categorized by time period which is multiplied by the time factor:

$$T\left(K_{TW,RT}\right) = \sum_{i\epsilon\{I,R,O\}}[t_i RT_i] \quad (4)$$

Knowledge factor of trust model is calculated by the summation of publication (equation (2)), citation (equation (3)), and retweet (equation (4)) which are multiplied by their own weights because each trust metric did not impact the final trust value equally:

$$KF(K) = P_W K_{DBLP,Pub} + C_W K_{DBLP,Cite} + RT_W K_{TW,RT} \quad (5)$$

For co-authorship, we calculate trust value of a specific collaborator by the summation of the frequency of co-authoring with author ith categorized by time period and weight with time factor:

$$T(coauthor, author_i) = \sum_{j\epsilon\{I,R,O\}} t_j CO_{i,j} \quad (6)$$

The trust value of a co-authorship is calculated by combining the impact of co-authorship with all co-authors and differentiating the impact of different co-authors by multiplying with Knowledge factor of that co-author:

$$T\left(coauthorship, S_{DBLP,Co}\right) = \sum_{i=1}^{\#coauther}\left[K_i \sum_{j\epsilon\{I,R,O\}} t_j CO_{i,j}\right] (7)$$

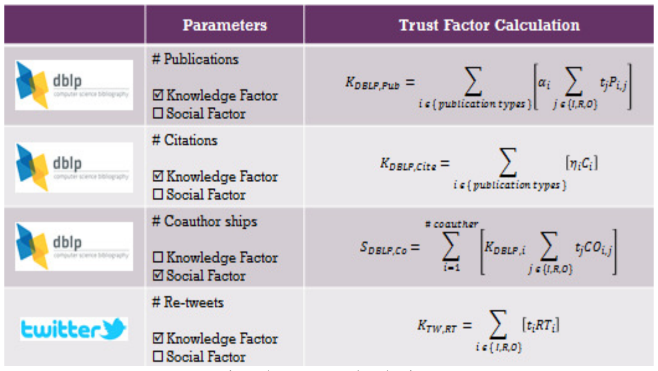
Social factor of the trust model is derived from the weighted co-authorship as illustrated below. This authorship weight makes the trust model more flexible to add more social trust metric in the future.

$$SF(S) = [CO_W S_{DBLP,Co}] \quad (8)$$

The final trust value is derived by the summation of knowledge factor and social factor weighted by their own weights as in equations (9) – (10). The breakdown calculation is summarized in Fig. 5. Note that more attributes can be added to the formula.

$$Trust\ Value = K_W K + S_W S \quad (9)$$

$$Trust\ Value = K_W[P_W K_{DBLP,Pub} + C_W K_{DBLP,Cite} + RT_W K_{TW,RT}] + S_W[CO_W S_{DBLP,Co}] \quad (10)$$



	Parameters	Trust Factor Calculation
dblp computer science bibliography	# Publications ☑ Knowledge Factor ☐ Social Factor	$K_{DBLP,Pub} = \sum_{i\epsilon\{publication\ types\}}\left[\alpha_i \sum_{j\epsilon\{I,R,O\}} t_j P_{i,j}\right]$
dblp computer science bibliography	# Citations ☑ Knowledge Factor ☐ Social Factor	$K_{DBLP,Cite} = \sum_{i\epsilon\{publication\ types\}}[\eta_i C_i]$
dblp computer science bibliography	# Coauthor ships ☐ Knowledge Factor ☑ Social Factor	$S_{DBLP,Co} = \sum_{i=1}^{\#coauther}\left[K_{DBLP,i} \sum_{j\epsilon\{I,R,O\}} t_j CO_{i,j}\right]$
twitter	# Re-tweets ☑ Knowledge Factor ☐ Social Factor	$K_{TW,RT} = \sum_{i\epsilon\{I,R,O\}}[t_i RT_i]$

Fig. 5 Trust calculation.

To give a proof of concept of our Trust Model, we have given weighted parameters a set of values. Fig. 6 illustrates an example how to calculate trust value. They are in accordance to the relative impact on the final scale. For instance, the Publication channel types are arranged in accordance of their importance and assigned weights in a relative and normalized fashion.

$$K_{DBLP,Pub} = \sum_{i\epsilon\{publication\ types\}}\left[\alpha_i \sum_{j\epsilon\{I,R,O\}} t_j P_{i,j}\right]$$

$$\text{where} \quad \alpha_A = 0.075, \alpha_B = 0.125, \alpha_{IC} = 0.15, \alpha_{IP} = 0.175$$

$$\alpha_M = 0.1, \alpha_{PH} = 0.125, \alpha_{PR} = 0.2, \alpha_W = 0.05$$

$$=> K_{DBLP,Pub} = \alpha_A * [t_R \cdot P_{A,R}] = 0.075 * [0.5 * 3] = 0.1125$$

$$K_{DBLP,Cite} = \sum_{i\epsilon\{publication types\}}[C_i]$$

$$=> K_{DBLP,Cite} = C_A = 2$$

$$K_{TW,RT} - \sum_{i\epsilon\{I,R,O\}}[t_i RT_i]$$

$$=> K_{TW,RT} = [t_I RT_I + t_R RT_R + t_O RT_O] = [0.3*3 + 0.5*6 + 0.2*10] = 5.9$$

$$S_{DBLP,Co} = \sum_{i=1}^{\#coauther}\left[K_{DBLP,i} \sum_{j\epsilon\{I,R,O\}} t_j CO_{i,j}\right]$$

$$=> S_{DBLP,Co} = K_{DBLP,1} * [t_R CO_{1,R}] + K_{DBLP,2} * [t_R CO_{2,R}] + K_{DBLP,2} * [t_R CO_{2,R}] + K_{DBLP,3} [t_R CO_{3,R}]$$

$$= 2.21 * [0.5*3] + 1.5 * [0.5*1] + 3.5*[0.5*1] + 4.0*[0.5*1]$$

$$= 7.82$$

$$Trust\ Value = 0.75[0.3K_{DBLP,Pub} + 0.5K_{DBLP,Cite} + 0.2K_{TW,RT}] + 0.25S_{DBLP,Co}$$

$$= 0.75*(0.3*0.1125 + 0.5*2 + 0.2*5.9) + 0.25*7.82$$

$$= 3.62$$

Fig. 6 Example of trust calculation.

B. *Performance Measurement*

Our trust network analysis requires filtering data based on its relevance to specific contexts. As the data amount becomes big, data analytics and data visualization pose significant performance concern. Thus, we took Solr into consideration. Solr (http://lucene.apache.org/solr/) is a popular, fast open-source enterprise search platform from the Apache Lucene project. We are especially interested in its features of full-text search and near real-time indexing. Solar uses a vector space model that yields fast search results. Meanwhile, Solr is written in Java and runs as a standalone enterprise search server with a REST-like API. In our project we loaded dataset in the form of XML.

As a proof of concept, the datasets that we loaded to Solr are Twitter and DBLP. We had four SOLR cores defined for the same, which were basically mapped to Tweet dataset, DBLP User, co-authorship dataset, and publication dataset. As mentioned before, Solr Index Server requires input interface as XML files. In this case, data-preprocessing is needed. We use Java Architecture for XML Binding (JAXB) to write XML files from Java objects.

```
public static void xmlWriter(TwitterUser twitterUser, String filename)
{
    JAXBContext context = JAXBContext.newInstance(TwitterUser.class);
    Marshaller m = context.createMarshaller();
    m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE);
    File file = null;
    file = new File(filename);
    m.marshal(twitterUser, file);
}
```

The segment code above shows how to generate an XML file from a Java object. For the fields that we want to write

XML, we have to mark "annotation" on the top of the class and the "Get" function in each field.

Once the formats are defined, the following steps help to load data onto the Solr server:

1. Copy XML files under the path apache-solr-4.0.0-BETA/example/example-DIH/solr/<core directory>/xml/. For example, DBLPUser core for Solr will have its files copied under apache-solr-4.0.0-BETA/example/example-DIH/solr/dblpuser/xml/.

2. Change to the directory path apache-solr-4.0.0-BETA/example/ by using:

```
cd apache-solr-4.0.0-BETA/example/
```

3. Restart the Solr server by using the command:

```
java -Dsolr.solr.home="./example-DIH/solr/" -jar start.jar
```

Data from Solr can be retrieved using an HTTP GET and receive XML, JSON, CSV or binary results. Once the data is indexed onto the Solr, we had to use that data with respect to the context, in order to calculate the knowledge and social factors for our Trust model described in the last section.

V. TRUST VISUALIZATION

As a result of our algorithm, a user will be able to distinguish between multiple service providers. The networks shown in Fig. 7 were generated after processing all data from our datasets. They depict the social impact of each other on the entire social network. Each node represents an individual, for whom we are analyzing Trust. In general, the nodes can represent any entity whose Trust we are analyzing. Each node carries its Trust factor calculated from the algorithm described in the last section.

Each node is associated with two features: size and color. The size of a node depicts its knowledge or in layman terms, the impact of the user on the entire social network (not the tiers considered alone). In other words, the size of a node represents the reputation in terms of knowledge factor and the aggregation of social factor from all other nodes. The edge from one node to another node shows the social impact of the starting node onto the other. The edges are directional because the impact that one node has on the other depends on their knowledge factor. For example, the impact of Steve Jobs on a student typically may be different from the impact of the student on Steve Jobs.

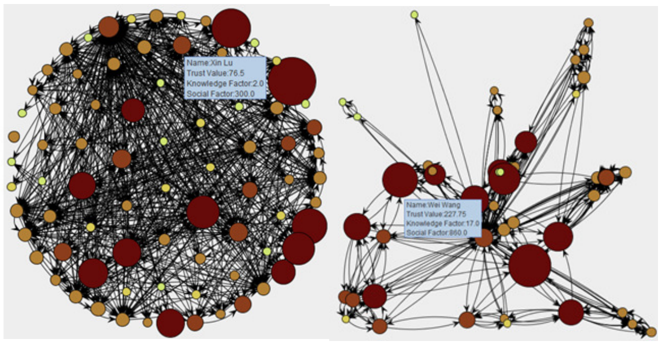


Fig. 7 Visualization of service trust.

In some cases, the node sizes are similar and the color gradient comes into play. The darker shade a node has, the higher knowledge factor it possesses. The edges between different nodes depict relationships. A relationship is strictly directional due to the asymmetric nature of the interaction itself. For instance, a graduate student might be highly impacted by Steve Jobs but the reverse may not be necessarily true. In terms of the algorithm, the edges depict the social relationship factor. In fact, the node size is also dependent on the number of edges incident on itself. The reason is that, a node encapsulates knowledge and social factor to study the social impact of a person on the entire network.

After studying various social network visualization tools, we decided to adopt JUNG [23] to build our visualization framework mainly due to its embedded rich graph mining algorithms. In addition, the JUNG framework offers a good object-oriented programming support, and a rich selection of vertex icons and graph layouts.

Different layouts may help illustrate different properties of a social graph. For example, Fig. 7(a) shows a KKLayout of JUNG provides an overall view of all relationships; Fig. 7(b) shows an ISOM layout which concentrates on different levels of the social kernel. Moving mouse over a node, the detailed information of the node will be shown: person's name, trust value, and knowledge and social factors.

VI. DESIGN AND DEVELOPMENT OF PROTOTYPING SYSTEM

A. *Prototyping system*

The process of analyzing the trust relations becomes exponential in complexity as there are huge numbers of nodes in KST network to analyze. To solve this challenge, we use the contextual information related to domain of analysis while querying from Twitter and DBLP. The benefits are two-fold: the analysis becomes more profound as it gives us more precise trust metrics; the complexity of the problem will be reduced nearly by ten folds. Another challenge of building the KST network is the bootstrapping problem, which is a classical and widely known problem where to establish trusted computing base.

Fig. 8 illustrates how we address the bootstrapping problem in the node selection process. We start from selecting a set of trusted experts (E) based on the context (for instance, cloud computing). The Knowledge factors for all of them are calculated as discussed in Section IV.*A*. We then query for the first level of connections (L_1) based on Twitter relations, followed by the second level of connections (L_2) by retrieving the direct connections for level L_1. Afterwards, we conduct a second level of query for finding all the nodes in DBLP. Such information is used to calculate the Knowledge factor for all the nodes at levels L_1 and L_2, which are in turn used to build a bi-directional graph. Edges represent the Social factor, which is calculated

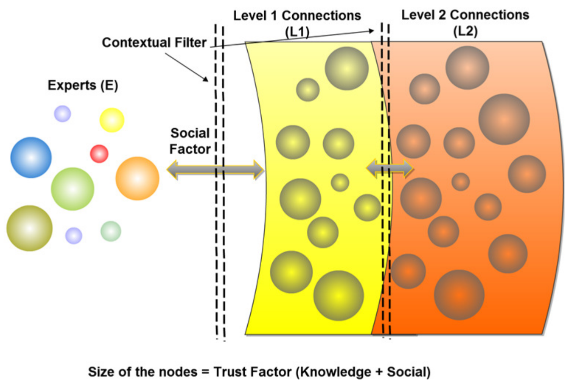


Fig. 8 KST network construction.

based on the Knowledge factor of the respective nodes. These relations are bi-directional as the impact of one node on the other depends on its own Knowledge factor. Once the internode relations for all the selected nodes are calculated, for each node we aggregate the social impact due to others, in order to obtain the Social factor for the node. The knowledge factor along with the social factor then form the trust value, which is represented as the node weight in the graph shown in Fig. 7.

Such an algorithm poses a challenge of recursive completeness. Let's consider for instance level L_1 from Fig. 8, the trust factor for a node would also depend on the social impact from level L_2 nodes connected to it, which in turn would depend on the next-level nodes and this goes on recursively. An important observation is that social factor of a node 'x' from level L_2 on a node 'y' from L_1would depend on the knowledge factor of 'x,' and this knowledge factor of 'x' is independent of next-level nodes. Thus, this means if we consider until level L_2 as in Fig. 8, we can calculate the complete trust factor for level L_1 nodes, irrespective of level L_2 nodes being incomplete which can be ignored in the final trust graph.

To retrieve Twitter's user information online, REST query is sent for each user using Twitter's ID or username. The data from the Twitter is used to add weight to the Trust value of the user obtained from the user. In this way, we obtain all the Twitter usernames for the list of DBLP users for which we are calculating the Trust values, by making use of a regular expression based on complete name information for the DBLP users. Furthermore, we use the obtained Twitter username to retrieve the Trust factor from Twitter by making use of the retweet information for the user.

The software environment includes: Apache Solr version 4.0.0-BETA, Java SDK version 1.5, and Eclipse IDE (optional, but makes it easy to run). Programmable Web maintains the description of various software modules and the relationship they maintain amongst each other. This information helps us link the developer information to their software creations.

We have developed a prototyping system whose architecture is illustrated in Fig. 9. The trust engine computes the weights of different components in the newly

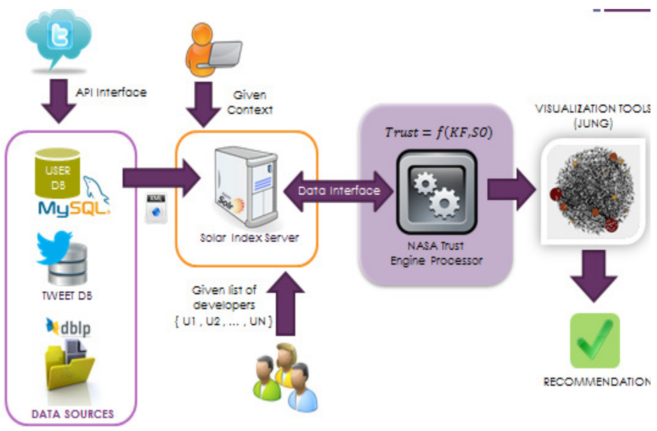


Fig. 9 Prototyping system.

created trust graph network. The log repository is used to track all state of the elements in the graph. This database can be later used as a mechanism to re-apply some calculations to the graph. The graph visualization engine helps a user to understand the social trust graph of various developers in a suitable context. The recommendation engine gives an output of the software that can be reused by the developer.

B. *Discussions and limitations*

In this project, our first constraint lies on a problem of user's behaviors when creating their profiles in social networks. This restriction prevents us from retrieving the targeted dataset as a result. For instance, some users might create their profiles without using their real names, which causes a difficulty to query their actual profiles. For another example, some users might not state any personal information and thus, makes it impossible to distinguish the targeted users from others who create their accounts using the same names. Moreover, potential issues exist when integrating datasets from multiple social networks, if some people do not establish their complete profiles in some social networks.

After our preliminary study, we have confronted with limitations in using open APIs to obtain data from various social networks. Firstly, no open APIs exist for some publication networks such as IEEE and ACM digital libraries. Google Scholar has already cancelled its open API option and has prevented crawlers from querying inner HTTP contents as well. Secondly, some social networks are highly strict in preserving user's privacy. For example, LinkedIn does not allow developers to query other person's information without obtaining their permission. Lastly, many open APIs including Facebook and Twitter have restricted the number of query per day. The number of Tweet messages is limited as well.

Consequently, this project selects several publically available archives and networks whose APIs allow us to obtain sufficient information to develop the KST network as a proof of concept. Our prototype is thus primarily created based on the parameters extracted from obtained datasets. However, it should be noted that our trust model is flexible enough to be expanded to support more parameters from other social networks and professional archives.

VII. Conclusions and Future Work

In this project, we have explored how to answer the two research questions posed in Section I. We have demonstrated that we can hidden knowledge from social networks, to support measurement of trust relationship of software services, so as to facilitate service discovery and reuse. We have developed a Knowledge-Social-Trust network-based trust algorithm to help quantify trustworthiness of a service. A context-aware recommendation prototyping framework has been built.

In future research we aim to build a context model to incorporate user background, project profile, and other situational information to refine our trust model. We plan to integrate our trust network model with our earlier developed service usage history network. We also plan to develop a plugin service to strengthen a scientific workflow engine (e.g., VisTrails) in context-aware service and workflow discovery.

VIII. Acknowledgement

This work is partially supported by National Aeronautics and Space Administration, under grant NASA NNX12AQ95G.

XI. References

[1] D.D. Roure, C. Goble, and R. Stevens, "The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows", *Future Generation Computer Systems*, 2009, 25: pp. 561-567.
[2] J. Zhang, W. Tan, J. Alexander, I. Foster, and R. Madduri, "Recommend-As-You-Go: A Novel Approach Supporting Services-Oriented Scientific Workflow Reuse", in Proceedings of *IEEE International Conference on Services Computing (SCC)*, 2011, Washington DC, USA, Jul. 4-9, pp. 48-55.
[3] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*, 2007, Springer.
[4] C. Pautasso, O. Zimmermann, and F. Leymann, "Restful Web Services vs. "Big"' Web Services: Making the Right Architectural Decision", in Proceedings of *the ACM 17th International Conference on World Wide Web (WWW)*, 2008, Beijing, China, Apr. 21-25, pp. 805-814.
[5] K. Gomadam, A. Ranabahu, M. Nagarajan, A.P. Sheth, and K. Verma, "A Faceted Classification Based Approach to Search and Rank Web APIs", in Proceedings of *IEEE International Conference on Web Services (ICWS)*, 2008, Beijing, China, pp. 177-184.
[6] J. Zhang, J. Wang, P.C.K. Hung, Z. Li, N. Zhang, and K. He, "Leveraging Incrementally Enriched Domain Knowledge to Enhance Service Categorization", *International Journal of Web Services Research (JWSR)*, 2012, 9(3).
[7] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake Shakes Twitter Users, Real-Time Event Detection by Social Sensors", in Proceedings of *the 19th International Conference on World Wide Web*, 2010, Raleigh, NC, USA, Apr. 26-40, pp. 851-860.
[8] M. Cheong and V. Lee, "Integrating Web-Based Intelligence Retrieval and Decision-Making from the Twitter Trends Knowledge Base", in Proceedings of *the 2nd ACM Workshop on Social Web Search and Mining* 2009, Hong Kong, China, Nov. 2-6, pp. 1-8.
[9] A.L. Hughes and L. Palen, "Twitter Adoption and Use in Mass Convergence and Emergency Events", *International Journal of Emergency Management*, 2009, 6(3-4): pp. 248 - 260.
[10] M. Demirbas, M.A. Bayir, C.G. Akcora, Y.S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-Sourced Sensing and Collaboration using Twitter", in Proceedings of *IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2010, Jun., pp. 1-9.
[11] N.A. Diakopoulos and D.A. Shamma, "Characterizing Debate Performance via Aggregated Twitter Sentiment", in Proceedings of *the 28th International Conference on Human Factors in Computing Systems*, 2010, Atlanta, GA, USA, Apr., pp. 1195-1198.
[12] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi, "Short and Tweet: Experiments on Recommending content from information streams", in Proceedings of *the 28th International Conference on Human Factors in Computing Systems*, 2010, Atlanta, GA, USA, Apr. 10-15, pp. 1185-1194.
[13] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha, "Time is of the Essence: Improving Recency Ranking using Twitter Data", in Proceedings of *the 19th International Conference on World Wide Web*, 2010, Raleigh, NC, USA, Apr. 26-30, pp. 331-340.
[14] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum, "An Empirical Study on Learning to Rank of Tweets", in Proceedings of *the 23rd International Conference on Computational Linguistics (COLING)*, 2010, Beijing, China, Aug., pp. 295-303.
[15] M. Biryukov, "Co-author Network Analysis in DBLP: Classifying Personal Names", in *Modelling, Computation and Optimization in Information Systems and Management Sciences*, 2008, Springer, pp. 399-408.
[16] T.-H. Huang and M.L. Huang, "Analysis and Visualization of Co-authorship Networks for Understanding Academic Collaboration and Knowledge Domain of Individual Researchers", in Proceedings of *International Conference on Computer Graphics, Imaging and Visualisation*, 2006, Jul. 26-28, pp. 18-23.
[17] M. Biryukov and C. Dong, "Analysis of Computer Science Communities Based on DBLP", *Lecture Notes in Computer Science*, 2010, 6273/2010: pp. 228-235.
[18] F. Reitz and O. Hoffmann, "An Analysis of the Evolving Coverage of Computer Science Sub-fields in the DBLP Digital Library", *Lecture Notes in Computer Science*, 2010, 6273/2010: pp. 216-227.
[19] Y. Zeng, Y. Yao, and N. Zhong, "DBLP-SSE: A DBLP Search Support Engine", in Proceedings of *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT)*, 2009, Sep. 15-18, pp. 626-630.
[20] S. Alwahaishi, J. Martinovic, V. Snásel, and M. Kudelka:, "Analysis of the DBLP Publication Classification Using Concept Lattices", in Proceedings of *Annual International Workshop on DAtabases, TExts, Specifications and Objects (Dateso)*, 2011, Pisek, Czech Republic, Apr. 20, pp. 132-139.
[21] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis", in Proceedings of *the 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT)*, 2009, Saint-Petersburg, Russian Federation, Mar. 23-26, pp. 565-576.
[22] S. Lawrence, "Free Online Availability Substantially Increases a Paper's Impact", *Nature*, 2001, 411(6837).
[23] JUNG, Accessed on, Available from: http://jung.sourceforge.net/.