Mining and Composition of Emergent Collectives in Mixed Service-Oriented Systems

Daniel Schall and Florian Skopik
Distributed Systems Group
Vienna University of Technology
Argentinierstraße 8/184-1, A-1040 Vienna, Austria
{lastname}@infosys.tuwien.ac.at

***Abstract*—Complex service-oriented systems typically span interactions between people and services. Compositions in such systems demand for flexible interaction models. In this work we introduce an approach for discovering experts based on their dynamically changing skills and interests. We discuss human provided services and an approach for managing user preferences and network structures. Experts offer their skills and capabilities as human provided services that can be requested on demand. Our main contributions center around an expert discovery method based on the concept of hubs and authorities in Web-based environments. The presented discovery and interaction approach takes trust-relations and link properties in social networks into account to estimate the hub-expertise of users. Furthermore, we show how our approach supports flexible interactions in mixed service-oriented systems.**

***Keywords*-mixed service-oriented systems, expert discovery, emergent collectives, hubs and authorities, ExpertHITS**

I. Introduction

The Web has evolved from a distributed repository of content to an interactive medium in which people actively shape the availability of information and services. Part of this evolution is often called Web 2.0 and characterized by the emergence of knowledge sharing and online service composition platforms. The transformation of how people interact on the Web has been poorly leveraged in existing SOA (service-oriented architecture)-based systems [1]. In traditional composition scenarios, services are created from the top down, without considering the availability and preferences of people, constraints and relationships, and the support of dynamic, ad-hoc interactions. The Web's people-centric nature has led to an unusual role for humans in information systems. Certain problems that are hard for software services to solve are outsourced to humans. As an example, task-based platforms for human computation (i.e., *crowdsourcing*) including Amazon Mechanical Turk[1] enable access to the manpower of thousands of people on demand by creating human-tasks that are processed by the crowd.

The service-oriented computing paradigm and Web services technology offer the ideal technical grounding for virtualizing the capabilities of knowledge workers in large scale computing environments [2]. Techniques for human computation in service-oriented systems have not yet uncovered its full potential. The research performed in this work centers on harnessing human capabilities within service-oriented environments while leveraging the new innovations provided by the Web 2.0 paradigm. People offer services to engage in different Web-based interaction scenarios driven by often intrinsic properties such as social interest, evolving skills and trust relations; and community dynamics.

In this paper we utilize Human-Provided Services (HPS) [2] enabling flexible interactions in service-oriented systems. In HPS we advocate that people should be able to define services following the same principles as SOA. To avoid the need for parallel systems of software-based services (SBS) and HPS, we define a *mixed service-oriented system* comprising SBS and HPS. Experts offer their skills and capabilities as HPS that can be requested on demand. Thus, mixed service-oriented environments are difficult to manage due to changing interaction patterns, behavior, and faults resulting from varying conditions in the environment. To address these complexities, we propose mining of interactions focusing on human characteristics in SOA.

In this work we present the following key contributions: (i) We discuss the discovery and interactions in mixed service oriented systems comprising HPS and SBS. (ii) Estimation of user reputation based on a context-sensitive algorithm. Our approach, called *ExpertHITS*, is based on the concept of hubs and authorities in Web-based environments. (iii) An approach for community reputation (the hub-expertise of users) influenced by trust relations. Dynamic link weights are based on trust and user rating influenced by the query context. ExpertHITS is calculated online, thus fully personalized based on the expert-requester's preferences (i.e., the demanded set of skills). (iv) Implementation and evaluation of our approach demonstrating scalability and effectiveness of the algorithm.

This paper is organized as follows. In Section II we present a motivating example detailing the need for flexible composition and interactions models. Section III introduces the fundamental idea and basic concepts of our approach. A detailed description of the proposed discovery approach is provided in Section IV. Our evaluation is presented in Section VI. Finally, we discuss related work in Section VII and conclude the paper in Section VIII.

[1] Amazon Mechanical Turk: http://www.mturk.com/

II. Crowdsourcing Scenario

A motivating scenario for discovering experts on demand and flexible interaction support is depicted in Figure 1. The process model may be composed of single tasks that are either processed by corresponding Web services or are assigned to responsible persons (see WS-HumanTask [3]).We assume that the task owners in this process exchange only electronic files and interact by using communication tools. While various languages and techniques for modeling such processes already exist, for example Business Process Execution Language [4] (BPEL), we focus on another aspect in this scenario: *discovery and interactions with trusted experts*. A language such as BPEL demands for the precise definition of flows and input/output data. However, even in carefully planned processes with human participation, for example modeled as BPEL4People activities [5], ad-hoc interactions and adaptation are required due to the complexity of human tasks, people's individual understanding, and unpredictable events. Especially, if people have not yet worked jointly on similar tasks, it is likely that they need to set up a meeting for discussing relevant information and process artifacts. Personal meetings may be time and cost intensive, especially in cases where people belong to different geographically distributed organizational units. Various Web 2.0 technologies, including forums, Wiki pages and text chats, provide well-proven support for online-work in collaborative environments. However, several challenges remain unsolved that are addressed in this paper:

(i) Who is the right expert that can assist in solving problems which people face while participating in the process? (ii) How can third parties (experts) be contacted and informed about the current situation and how can they easily be involved in ongoing collaborations? (iii) Based on which decision are experts selected, which information needs to be exchanged, and how can such scenarios be supported in service-oriented systems?

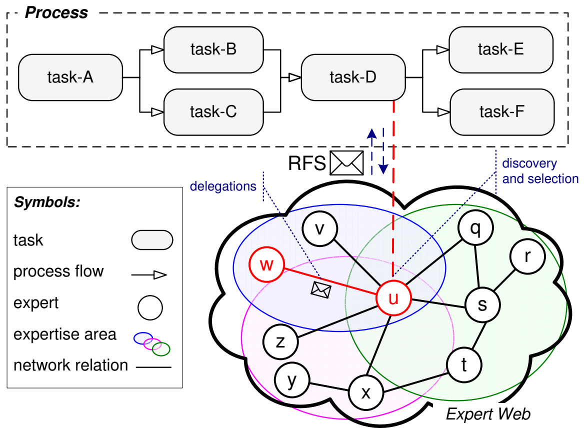


Figure 1. Discovering and including experts for online help and suppport.

Emergent collectives. Here we propose the *Expert Web* consisting of connected experts that provide help and support in a service-oriented manner. The members of this expert web are either *humans*, such as company employees offering help as online support services or can in some cases be provided as software-based services. Applied to enterprise scenarios, such a network of experts, spanning various organizational units, can be consulted for efficient discovery of available support. The expert seekers send *requests for support*, abbreviated as RFSs. Experts may also delegate RFSs to other experts in the network, for example when they are overloaded or not able to provide satisfying responses. Following this way, not only users of the expert network establish social relations to experts, but also relations between experts emerge.

III. Discovery of Emergent Collectives

In this section we will detail the basic concepts enabling the discovery of experts. Our approach is based on the following idea: given a search query containing the set of relevant skills, who is the expert (i) satisfying these demanded skills and (ii) how well is this expert connected to other people having similar expertise. From the Expert Web point of view, finding the right expert by performing skill matching is not sufficient. We also need to consider whether the expert will be able to delegate RFSs to other peers in the Expert Web.

A. Trust Emergence

Traditional rating and ranking models usually neglect social aspects and individual preferences. However, actors in the Expert Web may not be compatible with respect to working style and behavior. As a consequence, social aspects need to be considered and require dynamic interaction models. In this paper, we focus on *social trust* to support and guide delegations of requests. In contrast to a common security perspective, social trust refers to the flexible interpretation of previous collaboration behavior [6], [7] and the similarity of dynamically adapting interests [8]. Especially in collaborative environments, where users are exposed to higher risks than in common social network scenarios [9], and where business is at stake, considering social trust is essential to effectively guide human interactions. Relying on these works, we define trust in the Expert Web as follows:

> *Trust reflects the expectation one expert has about another's future behavior to perform delegated RFSs dependably, securely, and reliably based on experiences collected from previous interactions.*

Various metrics can be calculated by analyzing interaction logs. Relation metrics describe the links between actors by accounting for (i) recent interaction behavior, (ii) profile similarities (e.g., interest or skill similarities), (iii) social and/or hierarchical structures (e.g., role models).

B. Hubs and Authorities

In this work we utilize the concept of hubs and authorities in Web-based environments. This concept has been introduced by Kleinberg [10] to rank Web pages in search queries using the 'HITS algorithm' (Hyperlink-Induced Topic Search). The notion of *authorities* in social or collaborative networks can be interpreted as a measure to estimate the relative standing or *importance* of individuals in social networks. Compared to methods such as PageRank [11], the main advantage of the HITS model is that hub *and* authority scores are calculated for each node in the network.

Applying this idea in our scenario, a member of the Expert Web may receive an RFS and delegate work to some other peer in the network (characterizing hubs in the network). Receivers of the delegated work, however, expect RFSs fitting their skills and expertise (i.e., being an authority in the given domain). Careless delegations of work will overload these peers resulting in degraded processing time due to missing expertise. Within the Expert Web, authorities give feedback using rating mechanism (e.g., a number on the scale from 1 to 5) to indicate their satisfaction — whether a particular hub distributes work according to their skills and interest. Thus, a 'good hub' is characterized by a neighborhood of peers that are satisfied with received RFSs. On the other hand, delegation of work is strongly influenced by *trust*, for example, whether the initial receiver of the RFS (hub within the Expert Web) expects that the peer will process work in a reliable and timely manner. Thus, RFS receivers need to be trusted by influential hubs that are highly rated to be recognized as authoritative peers in the Expert Web.

C. Personalized Expert Queries

In this work, we define this concept as *expert hubs* that are well-connected (i.e., social network structure *and* connections based on joint collaborations) given a particular query context. Delegation is important in flexible, interaction-based systems because it becomes clear that expert hubs will attract many RFSs over time, thus presenting bottlenecks in terms of processing time needed to work on RFSs. On the other side, being a hub in the Expert Web also means that a person *knows* many other experts in similar fields of interest. We argue that the likelihood of being able to delegate RFSs to other experts greatly increases depending on the *hubness* of a person due to the embedding of a hub in expert areas (e.g., communities or interest groups). The major challenge in this scenario is that hubness needs to be calculated on demand based on a query. The query determines the *context* through the set of skills relevant for discovering experts.

An important aspect of the presented approach is to select interactions based on (query) context information. We assume that each interaction (e.g., based on delegated RFSs) is associated with context *tags* based on a skill taxonomy. The following steps outline our approach at a high level, which will be detailed in subsequent sections. First, matching is performed based on the query context. A skill model is used (not detailed in this work) to retrieve the set of qualified users. Second, expert hubs are discovered using link and interaction information.

Let us start formalizing this concept by discussing two scenarios as depicted in Figure 2. First, a query (see Q^A or Q^B) is specified either manually by a (human) expert seeker or derived automatically from a given process context, for example a predefined rule denoting that a particular set of skills is needed to solve a problem. The purpose of a query is to return a set of experts who can process RFSs, either by working on the RFSs or delegation. Thus, Q^A would return H^A as the user who is well connected to *authorities* in query context Q^A. There are two influencing factors, i.e., relations, determining hub- and authority scores: (i) how much hubs *trust* authorities (depicted as filled arrows from hubs to authorities) and (ii) *ratings* hubs receive from authorities (open arrows to hubs). Trust mainly influences the potential number of users (e.g., known by H^A) who can process delegated RFSs. On the other hand, receivers can associate ratings to RFSs to express their opinion whether the delegated RFSs fit their expertise. Q^B may demand for a different set of skills. Thus, not only matching of actors is influenced, but also the set of interactions and ratings considered for calculation ExpertHITS (i.e., only the set of RFSs and ratings relevant for Q^B).

Note, single *interactions* that lead to trust relations, as well as single *rating actions* that lead to rating relations are not depicted by Figure 2. A single arrow may in fact depict a number of *actions* (interactions or rating actions).

This approach provides a powerful tool for expert discovery because reputation (for example, within communities) is expressed as hub-expertise by weighting trust relations in personalized scopes (through the query context) and feedback-ratings. Also, we believe that our approach is difficult to cheat on because hub-expertise is influenced by how well hubs are connected to multiple authorities who propagate their expertise back to hubs. We will further elaborate on this concept in the following sections.

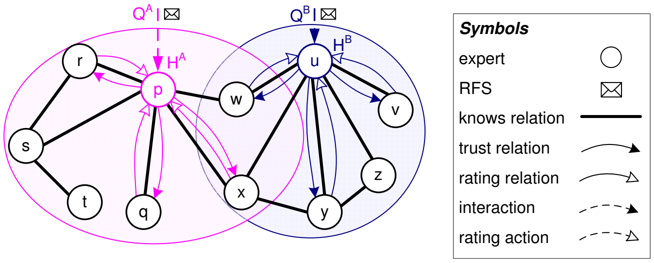


Figure 2. Hubs with different areas of expertise.

IV. Discovery of Expert Hubs

Here we present our expert discovery algorithm that is influenced by *social trust* and *rating mechanisms*. Our algorithm accounts for context information and weighted links between actors. Context is utilized by considering relations of experts in different scopes. Thus, the goal of our algorithm is to find hubs with respect to context. In the following, we discuss the basic flow of *actions* in the Expert Web. The actions include delegation of RFSs, ratings of requests, and advanced delegation patterns.

A. Hub Discovery

Let us assume that a query Q is specified to discover an expert hub (see Figure 3(a)). Every query influences the set of prior ratings (arrows pointing to u) and interactions (i.e., actions) that need to be considered when calculating hub- and authority scores. Consider the query context Q comprising actions related to the demanded set of skills. In this case, u has been discovered as the entry point to the Expert Web denoted as H^B. The matching algorithm to select related actions is not detailed in this work.

B. Delegation Actions

In Figure 3(a), user u receives an RFS issued towards the Expert Web. Since u represents the hub expert, u may decide to delegate the request to one of its neighbors v, w, y, z, which can be discovered through `knows` relations [12] (Figure 3(b)). Also, multiple delegation receivers of the same RFS can be selected if, for example, u has limited trust in its neighbors v, w, y, z.

In our Expert Web application scenario, `knows` is a bidirectional relation between users. A relation becomes active if both users acknowledge that they are connected to each other (v `knows` u *and* u `knows` v), a simple yet effective mechanism to support growth in social networks (e.g., newcomers and bootstrapping problem) while preserving user control. Notice, `knows` relations do not contain context related information such as tags. The context of interactions is derived from delegated RFSs (tags or type of RFS classified by using a skill taxonomy). To support growth in networks (e.g., how can newcomers become members of communities), we introduce an advanced interaction pattern in the Expert Web depicted by Figure 4(a).

C. Triad Delegation Pattern

An authority may need to delegate an RFS received from the hub to somebody who is known to the authority, but not the hub. This pattern is shown in Figure 4(a). Hub u delegates an RFS to y, which is in turn delegated to x and, thus, being responsible for processing the RFS.

If ties (i.e., through `knows` relations) between the pairs (u, y) and (y, x) exist, it is likely that x will attempt to establish a connection to u as well. This concept is known as *triadic closure* in social networks [13] and can be applied to support interaction patterns in service-oriented systems.

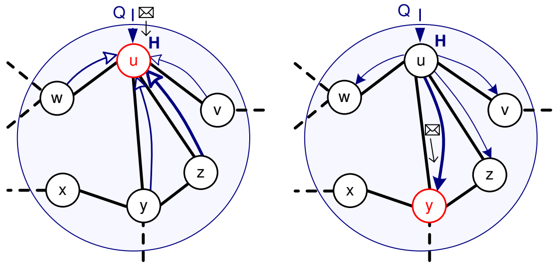
The *triad interaction pattern* (see [7]) enables x to connect to hubs and helps increasing its authority in the Expert Web. As mentioned previously, `knows` is a bidirectional connection and needs to be acknowledged by u.

D. Rating Procedure

An RFS is delivered back to the expert seeker from the Expert Web; i.e., the selected hub u depicted in Figure 4(b). The main argument of our model is to determine those hubs that are well embedded in expertise areas (e.g., communities). Thus, the hub-score should be influenced by feedback ratings denoting the level of satisfaction of authorities. Ratings are subjective opinions of authorities with respect to RFSs received from hubs, i.e., whether RFSs fit the expertise area of authorities. In the final step, RFSs are rated (see dashed open arrows) expressing the precision of received delegations. Indeed, such ratings are also given to RFSs traversing the Expert Web through triad delegation patterns. Given the scenario in Figure 4(a) and Figure 4(b), automatic propagation of ratings (e.g., if a delegated RFS from u to y was further delegated to x) is currently not considered in our model. Thus, x rates the RFS received from y and similarly y from u.

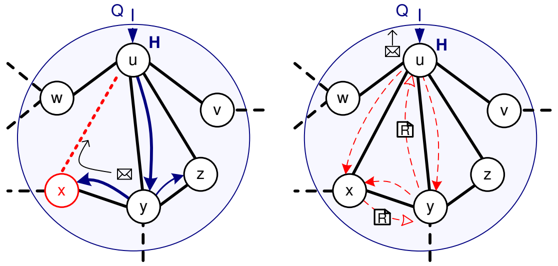
E. Trust Updates

Trust relations, based on experts' behavior are periodically updated with recent interaction data. Those interactions (reflected by filled dashed arrows) are aggregated to interaction metrics that are interpreted by pre-defined rules to infer trust.



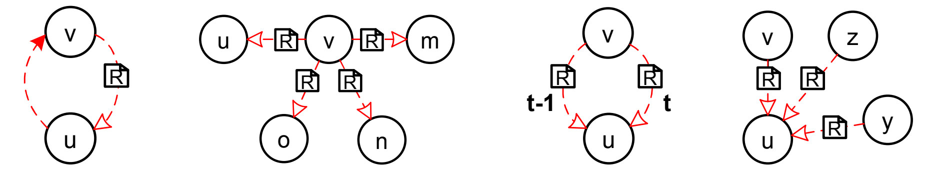
(a) Discovery of expert hub. (b) Selection of authority.

Figure 3. Hub discovery and trust-based selection of authorities.



(a) Delegation of RFS. (b) RFS reply and rating.

Figure 4. Advanced interaction patterns and feedback ratings.



(a) RFS. (b) Rating hubs. (c) EMA update. (d) Rated hub.

Figure 5. Update mechanisms for feedback ratings.

V. Metric Calculation

Considering the loosely structured and dynamically bound Expert Web example, it is important to derive metrics that can be used to infer trust relations in an automated manner.

A. ExpertHITS

We discuss the formal model for our expertise ranking algorithm consisting of (i) hub score $H(u;Q)$ of user u in query context Q and (ii) authority score $A(v;Q)$ of user v in the same query context Q.

$$H(u;Q) = \sum_{v \in \mathtt{knows}(u)} w_{vu}^{Q} A(v;Q) \qquad (1)$$

$$A(v;Q) = \sum_{u \in \mathtt{knows}(v)} w_{uv}^{Q} H(u;Q) \qquad (2)$$

- $H(u;Q)$: Hub score of u acting as a reliable entry point to the Expert Web brokering RFSs to authorities. Hubs are identified based on the demanded set of skills, knows relations connecting u to other experts *and* feedback ratings received from prior delegations.
- $A(v;Q)$: Authority score of user v. Authorities are skilled users (experts) that are connected to influential hubs. Authority means that users process RFSs received from hubs in a reliable, trustworthy manner.
- w_{uv}^{Q}: Trust influences the delegation behavior of hubs by selecting authorities based the success of interactions; in our example successfully delegated and processed RFSs.
- w_{vu}^{Q}: Denotes the connection strength of an authority v to hub u. The weight is calculated using information from ratings given by v to RFSs received from u.

B. Rating Mechanism

We calculate the rating $\tilde{r}_{vu}$ as the exponential moving average (EMA) to smoothen the sequence of ratings (i.e., short-term fluctuations). EMA gives more importance to recent ratings while not discarding older ratings. This model is a simple yet effective method. Figure 5 shows the following steps:

- Consider interactions between u and v (hub u delegating RFSs to the authority v) as an interaction trace $u \overset{t-n}{\rightarrow} v, \ldots, u \overset{t-1}{\rightarrow} v, u \overset{t}{\rightarrow} v$. We obtain a set of ratings $\{r_{vu,t-n}, \ldots, r_{vu,t-1}, r_{vu,t}\}$, where v rates u (Figure 5(a)).
- We define η with $0 < \eta < 1$ as a coefficient to smoothen previous ratings of v. In particular, the factor η is based on the ratings v has given to all other delegating HPSs (Figure 5(b)). Therefore, the factor η expresses the relationship between the two sets of ratings $X = \{r_{vu,t}, r_{vo,t}, r_{vn,t}, r_{vm,t}\}$ and $Y = \{r_{vu,t-1}, r_{vo,t-1}, r_{vn,t-1}, r_{vm,t-1}\}$. The calculation of η is performed based on the correlation coefficient $correl(X,Y)$ and a mapping of the coefficient $\eta = \frac{correl(X,Y)+1}{2}$ to obtain strictly positive values.
- We update the set of ratings into $\tilde{r}_{vu}$ associated with delegations from u to v (Figure 5(c)). The update rule is defined as:

$$\tilde{r}_{vu,t} = \eta r_{vu,t} + (1-\eta)\tilde{r}_{vu,t-1} \qquad (3)$$

- Finally, we calculate aggregated ratings $AR(u)$ for u based on inbound rating links $\mathtt{inlinks}(u)$ (Figure 5(d)). AR is calculated as the weighted sum of smoothened ratings:

$$AR(u) = \sum_{v \in \mathtt{inlinks}(u)} w(v)\tilde{r}_{vu,t} \times \left[\sum_{z \in \mathtt{inlinks}(u)} w(z) \right]^{-1} \qquad (4)$$

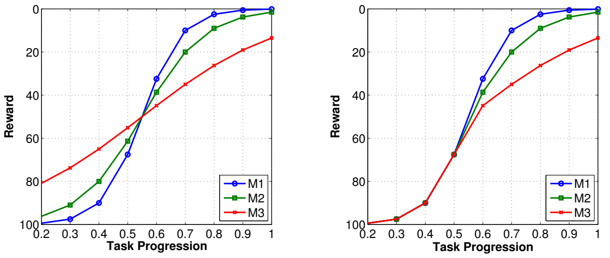
Currently we assume equal importance for each weight using $w(v) = \frac{1}{|\mathtt{inlinks}(u)|}$. The weight could be adjusted based on the trustworthiness of ratings to prevent malicious rating behavior.

C. Trust Weights

The weight w_{uv}^{Q} can be interpreted as how much u trusts v in processing RFSs in a reliable manner (Eq. 5). Specifically, experts' behavior in terms of reliability and RFS processing successes, are periodically updated with recent captured interaction data.

$$w_{uv}^{Q} \equiv \frac{\text{succ. delegations from } u \text{ to } v}{\sum_{w \in \mathtt{knows}(u)} \text{succ. delegations from } u \text{ to } w} \qquad (5)$$

Reliability and processing success of RFSs are based on a *task rewarding schema*. Each RFS is associated with a task ht. Tasks model RFS states such as *accepted*, *inprogress*, *finished* or *aborted*. Thus, task rewards are automatically



(a) Initial rewarding model.

(b) Refined rewarding model.

Figure 6. RFS rewarding model.

given to RFSs to measure the degree of success. For example, fast and reliable processing of tasks yields higher rewards, thereby resulting in higher trust in a collaboration partner. To model task rewards, we use a function belonging to the family of sigmoid functions with the general form $P(t) = \frac{1}{1+e^{-t}}$ (see Figure 6). Sigmoid functions are typically used to model systems that saturate at large values of t, for example, the processing time of RFSs. The detailed mechanisms of the rewarding model can be found in [14]. Here we briefly discuss the most important parameters. The task rewarding function f_T based on the task processing time $PT(ht)$ for a given task ht is defined as:

$$f_T(PT(ht)) = \frac{\psi}{1 + EXP(-\frac{PT(ht)-\sigma}{\delta})} \tag{6}$$

Table I

REWARDING MODEL AND RELATED SYMBOLS.

Symbol	Description
$f_T(PT(ht))$	Rewarding function based on processing time $PT(ht)$.
τ	Time interval (expiration time) after which ht fails.
ψ	Saturation of $f_T : [0, \tau] \rightarrow [0, \psi]$, $\psi \in [0, 1]$.
σ	Parameter to define the horizontal displacement of f_T.
δ	Parameter to define the steepness of f_T's slope.
$f_{T\langle M \rangle}$	Task rewarding function for model $M(\psi, \sigma, \delta)$.
M_n	Depicts a rewarding model identified by the index n.

A plot of this function based on different parameters (see M_n in Table I) is shown by Figure 6(a). The basic idea is to use different models (e.g., $M1, M2, M3$) to account for the *risk* that a particular *type of RFS* will not be processed by an HPS. Risk is automatically calculated (parameter δ – Table I) based on *finished* versus *aborted* tasks within the Expert Web community. On other words, the task rewarding function f_T should fall less steeply if a particular type of RFS tends to be aborted by the community. To model risk for the task progression spectrum that is based on the task processing time, f_T needs to be refined as a stepwise function:

$$f^*_{T\langle M \rangle}(PT(ht)) = \begin{cases} f_{T\langle M_\Lambda \rangle}(PT(ht)) & \text{, if } \frac{d^2 f_{T\langle M \rangle}}{dt} < f''_{T\langle M \rangle} \\ f_{T\langle M \rangle}(PT(ht)) & \text{, otherwise} \end{cases} \tag{7}$$

Figure 6(b) shows the refined rewarding function. Progression (based on processing time) towards a particular point (the inflection point of f_T – see 0.5 on horizontal axis) results in equal rewards regardless of the model ($M1, M2, M3$). Beyond this point, RFSs are differently rewarded depending on the risk modeled by M_n. For example, given $M3$ that models RFSs with higher risks, higher rewards are given because a successfully processed RFS becomes more valuable to the delegating hub. The benefit of this approach is that the rewarding function f_T undergoes a self-configuration process by selecting a particular model M_n automatically based on monitored interactions. Instead of selecting a schema manually, rewarding adjusts based on the emergence of interaction behavior and dynamics within the Expert Web.

VI. Evaluation

In our experiments we focus on the performance of ExpertHITS as well as the influence of trust and ratings on hub/authority scores. In this work, we do not deal with performance issues due to network delay or end-to-end characteristics of the entire system. Here we focus on ExpertHITS calculation time under different conditions.

A. Experimental Setup

In all our tests we used a machine with Intel Core2 Duo CPU 2.50 GHz, 4GB RAM, running Java 1.6 and an OSGi Java container for hosting services. A query service invoking the ExpertHITS algorithm including metric calculation has been implemented in Java. The ExpertHITS algorithm has been implemented on top of a Java-based graph toolkit [15].

B. Generation of Synthetic Interaction Graph

The approach we take is to generate artificial interaction data imitating real collaboration environments. For this purpose, we adopt the *preferential attachment* method [16] which provides a realistic model for science collaboration scenarios. Specifically, a collaboration network is modeled as an undirected graph $G = (N, E)$ comprising a set of nodes N and edges E establishing connections between nodes. The probability of establishing a new connection to a given node is proportional to its degree distribution. Using this basic network structure, we generate interactions (delegations and ratings) associated with edges.

Assuming a scale free network with power law distribution, hubs play a central role, thereby generating a large amount of delegations. This behavior is taken into account when generating artificial interactions by estimating that 80% of delegations are initiated by about 20% of network HPSs; thereby immitating hub behavior of certain nodes.

C. Performance Results

Complexity is a crucial factor in order to support personalization of queries. We analyze different networks comprising actors and interactions that have already been matched with a specific query context Q. The system must be able to handle multiple requests simultaneously. We analyze the performance of ExpertHITS under different load conditions. At this stage, we focus on small-scale (100 nodes) and medium-scale (1000 nodes) networks. ExpertHITS results are calculated for 50-500 concurrent requests. A queue holds instances of the constructed network. A thread pool instantiates worker threads to calculate personalized ranking scores based on query preferences.

Table II
AVERAGE PROCESSING TIME CONCURRENT REQUESTS (SECONDS).

Requests	50	100	200	300	400	500
Small-scale	9	16	20	21	24	25
Medium-scale	199	325	390	986	1432	1663

Small-scale networks can be processed in a real-time manner requiring in our experiments in the worst case up to 12 seconds. On average, 19 seconds can be expected under different load conditions (50-500 concurrent requests). The results of medium-scale networks are compared with small-scale networks in Table II. Computing ExpertHITS in such networks takes up to several minutes when serving concurrent requests (i.e., on average 390s at a load of 200 requests). Load conditions in the range between 300-500 concurrent executions of the algorithms results on average in response times between 15-25 minutes. Given our initial online help and support example, we believe it is sufficient to compute ExpertHITS in this magnitude because illustrated processes, for example in enterprise crowdsourcing, do not demand for hard computational (time) constraints. Scalability and reduced processing time can be achieved by using multiple servers and load balancing mechanisms. These mechanisms are subject to our future work and performance evaluation.

D. Hub Quality and Relation to HITS

To study the results of ExpertHITS, we define a set of ranking evaluation metrics in the following.

- Absolute ranking change returns the ranking change of u in a given result set. The ranking change $RC(u)$ is defined as follows:

$$RC(u) = pos(u)_{HITS} - pos(u)_{ExpertHITS} \tag{8}$$

- We define *quality* as the aggregated trust weights of u's neighbors. Quality is calculated as

$$\mathcal{Q}(u) = \sum_{v \in \texttt{knows}(u)} \sum_{z \in \texttt{inlinks}(v)} w_{zv} \tag{9}$$

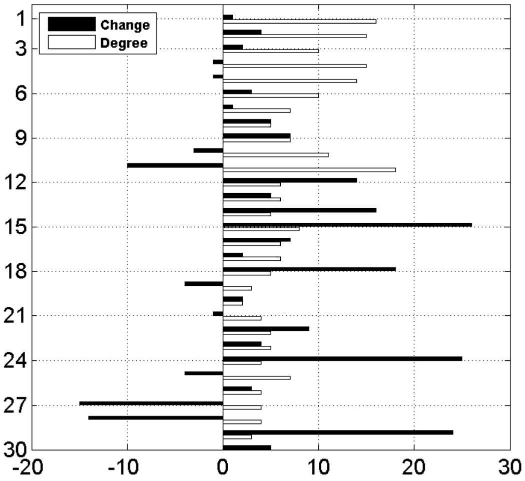


Figure 7. Ranking change and node degree.

Rank	Quality $\mathcal{Q}$	Rating
1	3.7	0.8
2	3.0	0.7
3	3.0	1.4
4	3.0	0.5
5	3.0	0.5
6	1.0	0.8
7	1.0	1.6
8	1.0	1.8
9	1.0	0.8
10	1.0	0.3
11	0.4	0.9
12	1.0	1.1
13	1.0	0.9
14	1.0	1.1
15	1.0	0.3

(a) Hub quality and ratings (1-15).

Rank	Quality $\mathcal{Q}$	Rating
16	1.0	1.3
17	1.0	0.9
18	1.0	0.9
19	1.0	2.7
20	1.0	4.1
21	1.0	1.5
22	1.0	1.0
23	1.0	0.9
24	1.0	1.1
25	1.0	0.2
26	1.0	1.3
27	1.0	1.2
28	1.0	0.8
29	1.0	1.5
30	1.0	2.5

(b) Hub quality and ratings (16-30).

Figure 8. ExpertHITS ranking results: ranking change, quality, and ratings.

To test the effectiveness of ExpertHITS, we performed experiments to study the impact of ratings and trust on expert rankings. In the following figures, we show the top-30 ranked experts in a small-scale network (100 nodes). Results are sorted based on the *position* within the result set (see vertical axis in Figure 7 and rank in Figure 8). Figure 7 shows the degree of network nodes and ranking changes obtained by comparing ranking results using the HITS algorithm without taking trust or ratings into account.

In Figure 8, one can see that all nodes within the top segment received high ratings given a high degree of links which is the desired property of ExpertHITS. Different levels of quality (i.e., quality mainly being 1 of ranked nodes between the positions 6-30) can be explained by the impact of node degree on quality. Some nodes are demoted (negative ranking change) since the node (e.g., see 11) has received low ratings even though the node has a high degree of links. On the other hand, nodes get promoted (positive ranking change) if they exhibit sufficient high ratings (see 15) or high quality (see 20 which was promoted a few positions only due to limited degree). Overall, ExpertHITS exhibits the demanded properties of promoting well-connected and rated hubs, thereby guaranteeing the discovery of reliable entry points to the Expert Web.

VII. Related Work

We believe that models and algorithms to determine the expertise of users are important in future service-oriented environments. The notion of service-orientation is not only applicable to Web services. Service-orientation in human collaboration is becoming increasingly important. Task-based platforms allow users to share their expertise [17]; or users offer their expertise by helping other users in forums or answer communities [18]. By analyzing email conversations [19], the authors studied graph-based algorithms such as HITS [10] and PageRank [11] to estimate the expertise of users. The authors in [20] followed a graph-entropy model to measure the importance of users. The work by [21] followed a graph-based approach and applied HITS as well as PageRank in online communities (i.e., a Java question and answer forum). While the above cited works attempted to model the importance of users based on interactions and information flow; they ignore the fact that interactions typically take place in different *contexts*. Approaches for calculating personalized PageRank scores [22], [23] were introduced to enable topic-sensitive search on the Web, but have not been applied to human interaction analysis. We propose a model where expertise analysis is performed considering context information. Our proposed algorithm can be computed online, while most other approaches require offline calculation (mining).

VIII. Conclusion and Future Work

In this paper we introduced a new approach for supporting reputation- and trust-based discovery and interactions in Expert Webs. Our approach is based on the HPS concept enabling knowledge workers to offer their skills and expertise in service-oriented systems. Unlike traditional top-down models, we proposed the combination of preplanned process steps *and* tasks that can be outsourced to the Expert Web to solve emergent problems. We demonstrated a novel approach for estimating expert reputation based on link structure and trust relations. Trust information is periodically updated to capture dynamically changing interaction preferences. We have shown that ExpertHITS can be computed in an online manner, thereby enabling full personalization at runtime. Existing approaches in personalized expertise mining algorithm typically perform offline interaction analysis. Our empirical evaluations have shown that ExpertHITS exhibits the desired properties; trust and rating weights influence hub- and authority scores. These properties ensure that our algorithm discovers experts which are well-connected to other experts.

In our future work we will evaluate our approach in real collaboration environments to capture more realistic interaction data.

Acknowledgements

This work was supported by the European Union FP7 project COMPAS, grant no. 215175.

References

[1] D. Schall, S. Dustdar, and M. B. Blake, "Programming human and software-based web services," *Computer*, vol. 43, pp. 82–85, 2010.

[2] D. Schall, H.-L. Truong, and S. Dustdar, "Unifying Human and Software Services in Web-Scale Collaborations," *IEEE Internet Computing*, vol. 12, no. 3, pp. 62–68, 2008.

[3] M. Amend et al., "Web Services Human Task (WS-HumanTask), Version 1.0." 2007.

[4] OASIS, "Business Process Execution Language for Web Services, Version 2.0." 2007.

[5] A. Agrawal et al., "WS-BPEL Extension for People (BPEL4People), Version 1.0." 2007.

[6] D. Artz and Y. Gil, "A survey of trust in computer science and the semantic web." *J. Web Sem.*, vol. 5, no. 2, pp. 58–71, 2007.

[7] F. Skopik, D. Schall, and S. Dustdar, "Modeling and mining of dynamic trust in complex service-oriented systems," *Information Syst.*, 2010.

[8] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *Trans. on the Web*, vol. 3, no. 4, 2009.

[9] C. Dwyer, S. R. Hiltz, and K. Passerini, "Trust and privacy concern within social networking sites: A comparison of facebook and myspace," in *AMCIS*, 2007.

[10] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford Digital Library Technologies Project, Tech. Rep., 1998.

[12] "Friend-of-a-friend," Website, 2010, http://xmlns.com/foaf/spec/.

[13] D. J. Watts, *Six degrees: The science of a connected age*. WW Norton & Company, 2003.

[14] D. Schall, "Human interactions in mixed systems - architecture, protocols, and algorithms," Ph.D. dissertation, 2009.

[15] "Java universal network/graph framework," Website, 2010, http://jung.sourceforge.net/.

[16] A. Reka and Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, June 2002.

[17] J. Yang, L. Adamic, and M. Ackerman, "Competing to share expertise: the taskcn knowledge sharing community," in *Int. Conf. on Weblogs and Social Media*, 2008.

[18] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *WSDM*. ACM, 2008, pp. 183–194.

[19] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang, "Graph-based ranking algorithms for e-mail expertise analysis," in *DMKD*. ACM, 2003, pp. 42–48.

[20] J. Shetty and J. Adibi, "Discovering important nodes through graph entropy the case of enron email database," in *LinkKDD*. ACM, 2005, pp. 74–81.

[21] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *WWW*. ACM, 2007, pp. 221–230.

[22] T. H. Haveliwala, "Topic-sensitive pagerank," in *WWW*. ACM, 2002, pp. 517–526.

[23] G. Jeh and J. Widom, "Scaling personalized web search," in *WWW*. ACM, 2003, pp. 271–279.