Domain-aware reputable service recommendation in heterogeneous manufacturing service ecosystem

Yushun Fan, Keman Huang, Wei Tan, Yang Zhong, Jinhui Yao, Nepal Surya & Shiping Chen

To cite this article: Yushun Fan, Keman Huang, Wei Tan, Yang Zhong, Jinhui Yao, Nepal Surya & Shiping Chen (2015) Domain-aware reputable service recommendation in heterogeneous manufacturing service ecosystem, International Journal of Computer Integrated Manufacturing, 28:11, 1178-1195, DOI: 10.1080/0951192X.2014.961964

To link to this article: http://dx.doi.org/10.1080/0951192X.2014.961964

Published online: 14 Oct 2014.
Domain-aware reputable service recommendation in heterogeneous manufacturing service ecosystem

Yushun Fan\textsuperscript{a}, Keman Huang\textsuperscript{b*}, Wei Tan\textsuperscript{c}, Yang Zhong\textsuperscript{d}, Jinhui Yao\textsuperscript{d}, Nepal Surya\textsuperscript{a} and Shiping Chen\textsuperscript{e}

\textsuperscript{a}Automation Department, Tsinghua University, Beijing 100084, China; \textsuperscript{b}School of Computer Science and Technology, Tianjin University, Tianjin 300072, China; \textsuperscript{c}IBM Thomas J. Watson Research Centre, Yorktown Heights, NY 10598, USA; \textsuperscript{d}School of Electrical and Information Engineering, University of Sydney, Sydney, Australia; \textsuperscript{e}Information Engineering Laboratory, CSIRO ICT Centre, Sydney, Australia

(Received 20 November 2013; accepted 1 September 2014)

Networked manufacturing becomes an important manufacturing method for modern manufacturing enterprises. With the wide adoption of service-oriented architecture and cloud manufacturing, manufacturing enterprises and organisations publish their manufacturing capability, such as resources, processes and knowledge as manufacturing services. A rapidly growing manufacturing service ecosystem can be observed nowadays, which brings the information overload problem for the service selection. Thus, how to organise these services and how to recommend the reputable services become two important issues for the manufacturing service retrieval and reuse. In this paper, the service cluster method WSTPCluster, which develops the topic model based on Latent Dirichlet Allocation, is employed to cluster services into specific domains. As the service ecosystem can be modelled as a heterogeneous service network, a unified reputation propagation method URPM is proposed to calculate the reputation of services and to distinguish the reputable services in each domain. Combining WSTPCluster and URPM, domain-aware reputable service recommendation method is introduced to recommend the high reputable services in each domain for the consumers. Experiments show that this method brings a better performance both in the recommendation accuracy and in the long tail recommendation.

Keywords: manufacturing service ecosystem; domain-aware reputable service recommendation; topic model; service clustering; reputation propagation; heterogeneous network

1. Introduction

With the rapid evolution of the computer and network technologies, as well as the pressing need of international competitiveness in manufacturing era over these years, networked manufacturing (Fan and Huang 2007) has become an important pattern for the collaboration among modern manufacturing enterprises and organisations. Considering the advantage of the service-oriented architecture (SOA) for collaboration, manufacturing enterprises and organisations are trying to publish their manufacturing capability as modern manufacturing services over the Internet (Jang et al. 2008). These manufacturing services – including manufacturing message specification services for the communication between different manufacturing enterprises (Wu, Xi, and Zhou 2008), production system controlling and monitoring services to support reconfiguration of the manufacturing lifecycles and the intelligent functionalities, and machining simulation services for the cross-platform and high-level numerical control simulation (Karnouskos et al. 2010; Phaithoonbuathong et al. 2010) – have been introduced for the collaboration among suppliers and customers over the Internet (Cai, Zhang, and Zhang 2011). Hence manufacturing enterprises can focus on their own core functionality and collaborate with each other to gain more agile, flexible and efficient manufacturing operations (Zupančič, Sluga, and Butala 2012) without impacting their core compatibility (Badr, Peng, and Biennier 2012). Modern manufacturing services have played an increasingly significant role for the development of the manufacturing enterprises (Li et al. 2014).

Furthermore, the promotion of cloud manufacturing (Li, Zhang, et al. 2010; Xu 2012) encourages manufacturing enterprises to publicly provide services. This trend further motivates third-party developers to combine manufacturing services into value-added compositions to fulfill complex requirements (de Souza et al. 2008). Hence the numbers of manufacturing services and their compositions are rapidly increasing, resulting into a rapidly expanding manufacturing service ecosystem (MSE) (Cardoso and Camarinha-Matos 2013).

The MSE can be considered as a complex system which consists of a huge number of manufacturing services, their value-added compositions, manufacturing service providers as well as third-party developers. In the MSE, manufacturing services collaborate with each other in an unforeseen way. This results into dynamic and
complex relations among them. The huge number of manufacturing services with similar functionality brings the information of overload problem (Eppler and Mengis 2004) for service selection. Furthermore, the usability and credibility directly associate with the success of the MSE. Thus, how to organise these massive services to reduce search space for the service selection and how to identify services with high usability and credibility to construct manufacturing process with high quality have become two important issues for the management of MSE.

First of all, organising services into domain has been proved as an effective approach to facilitate the service selection (Platzer, Rosenberg, and Dustdar 2009). However, the systematisation of the services is usually preliminary and artificial (Wang, Zhang, et al. 2011). A few approaches (Pop et al. 2010; Zhou et al. 2013; Elgazzar, Hassan, and Martin 2010; Wu et al. 2014; Zhang et al. 2012) have been proposed to automatically cluster the services into domains. These methodologies cluster the services into domains based on the Web services description language (WSDL) or the quality of services (QoS). However, for most manufacturing services, there is no WSDL or QoS information available for the clustering due to the business interests and privacy protection (Ye and Jacobsen 2013; Dou et al. 2013). In fact, manufacturing services usually represent manufacturing capabilities. Thus the tag and the description of manufacturing services are more crucial compared with common Web services. Furthermore, some manufacturing services offer different domain-specific functionalities which naturally belong to multiple domains. For example, the Ei3² manufacturing services can be used to store, process, analyse and monitor large amounts of machines in manufacturing enterprises; to generate tracking reports; and to integrate with other enterprise computing platforms so that manufacturing enterprises can collaborate with each other. Hence it should belong to domains such as enterprise application, collaborative tools, monitor and analysis, etc. Latent Dirichlet Allocation (LDA) has been proved effective and widely used to automatically detect the latent domains among texts (Blei and Lafferty 2006). Thus, this paper proposes the service cluster method WSTPCluster to detect the functionality domain in the MSE based on LDA and then assign them into different functionality domains.

On the other hand, facing a huge number of services with similar functionality in the same domain, the guideline to help users to select high-quality services is required. The reputation can be regarded as a predictor of the future behaviour (Malik and Bouguettaya 2009). The services with higher reputation will have a higher usability and credibility. Thus the reputation-based trust (Nepal, Malik, and Bouguettaya 2011) plays a major role in conducting business collaboration over the Internet in service-oriented environments (Ridha, Rizvi, and Azzedine 2007; Li, Fan, and Li 2011). Some works aggregate the opinions of other users in the trust network to generate personalised recommendation for consumers based on graph-based trust models, interaction-based trust models and hybrid trust models (Yao et al. 2011; Wang and Vassileva 2007; Sherchan, Nepal, and Paris 2013; Wu et al. 2013). However, performance measurement of manufacturing services is quite different from the product functionality and tolerance (Ellram, Tate, and Billington 2004) or the IT-attributes of services, QoS, from the field of computer science (Huang et al. 2011). Rare MSE offers the QoS information and it is resource-intensive and sometimes impossible to fetch the QoS. Also, in practice, most MSEs do not contain detailed feedback from users. Furthermore, the reputations of the manufacturing services, service providers, service compositions and third-party developers do not isolate but interact with each other (Huang et al. 2013). In order to represent their relations, this paper firstly models the MSE as a heterogeneous network and then proposes a unified reputation propagation method over the heterogeneous network named URPM to calculate the service’s reputation. Finally the reputation is employed to identify core services in each domain. The definition of the heterogeneous network will be discussed in Section 3.

Thus in this paper, combining the topic model-based clustering method and the reputation propagation method over the heterogeneous network, the Domain-aware Reputable Service Recommendation (DRSR) framework is proposed to help the developers to construct high-quality compositions. The main contribution of this paper is as follows.

- A service clustering approach WSTPCluster based on topic model is introduced to automatically detect the functionality domain of services and assign services into different domains.
- A unified reputation propagation method URPM over the heterogeneous network is proposed to calculate the services’ reputation and identify the core services in each domain.
- DRSR framework is present to combine the service clustering and core service identification to guide developers to construct high-quality service compositions. Experiments show that this method gains a better performance both in the recommendation accuracy and in the long tail recommendation.

The rest of the paper is organised as follows. Section 2 discusses the related work. Section 3 introduces the definition of the heterogeneous network model and then formally defines the problems based on a four-phase composition behaviour model. Section 4 clarifies the service clustering method WSTPCluster based on topic model. Section 5 proposes the unified reputation propagation method URPM over the heterogeneous network. Section 6 explains DRSR framework. Section 7 studies
the effectiveness of the proposed method and Section 8 concludes this paper.

2. Related work

2.1. Manufacturing service management

SOA has been widely adopted in the manufacturing sector. Manufacturing service management has attracted increasing attention from industry and academia.

Some research studies focus on the expansion of Web service to publish manufacturing services into MSE. For example, the semantic manufacturing capability profile which contains the Web ontology language (OWL)-based service descriptions is introduced to enhance the Web service description for manufacturing services and then the description logic-based reasoner for semantic matching is employed to help the manufacturing service discovery (Jang et al. 2008). Combining the manufacturing message specification (MMS) and SOA, the manufacturing communication model is shown to support common industrial messaging communication (Wu, Xi, and Zhou 2008).

Other research works study the manufacturing service’s evaluation for the manufacturing services selection. The universal enterprise manufacturing service maturity model is provided to evaluate and level the service development phase which can help in scientific decision-making (Li et al. 2014). Definitions, classifications and measurement methods are presented to evaluate the composition’s flexibility so that the optimal-selection based on flexibility can be enabled (Guo et al. 2012). A performance evaluation method for service-oriented manufacturing network, which combines performance indicators on business, service and implementation level, is proposed for the optimal service selection and composition in the MSE (Huang et al. 2011). Li et al. introduce a method to calculate the trust among services and then the trust-based multi-service selection method is presented to select the optimal services for the composition (Li, Fan, and Li 2011). The conceptual pro-active service ecosystem framework is introduced to study the three-layer-architectural and the five temporal phase composition process in the MSE (Cardoso and Camarinha-Matos 2013).

Due to the rapid increasing of manufacturing services over the Internet, automatically clustering the massive services into domains can reduce the search space to facilitate the service selection, and identifying the reputable services in each domain can help to construct the compositions based on services of high quality. Thus this paper will focus on the service clustering and the service reputation calculation to facilitate the optimal service selection and reused.

2.2. Service clustering

Web service clustering becomes an effective method (Platzer, Rosenberg, and Dustdar 2009) for the service organisation. The most widely used clustering approaches are the semantic-based similarity and non-semantic-based similarity (Chen et al. 2013). For the semantic clustering approaches, the ontology is employed to compute the semantic similarity between Web services (Bianchini et al. 2006; Dasgupta, Bhat, and Lee 2010). For the non-semantic clustering approaches, some studies extract different features from Web services description language (WSDL) documents, such as content, types, messages and service names (Elgazzar, Hassan, and Martin 2010; Liu and Wong 2009), while some studies even exploit the tagging data to improve the clustering performance (Wu et al. 2014; Chen et al. 2014). Some studies (Zhang et al. 2012; Xia et al. 2011) cluster the services based on the QoS, such as service accessibility, service cost, service response time and service reliability.

However, with the wide adoption of the representational state transfer (REST) architectural principles (Thomas 2000) for Web services, most services only provide human-readable (Danielsen and Jeffrey 2013) but not computer-interpretable documentation (Jang et al. 2008) and no WSDL documents are provided for services. Due to business interests and the privacy protection requirement (Ye and Jacobsen 2013; Dou et al. 2013), rare manufacturing services offer WSDL documents or detailed QoS information for consumers. In fact, manufacturing services usually represent manufacturing capabilities; thus the tag and the description of services are more important compared with common Web services. This paper, therefore, introduces the clustering method to organise the services into domains just through the descriptions and tags.

2.3. Service reputation

The concept of reputation-based trust (Nepal, Malik, and Bougguetaya 2011) is not new. Reputation-based trust refers to the subjective assessment that whether the entity will behave as expected (Josang, Ismail, and Boyd 2007). It has been studied in many disciplines including sociology (Mollering 2001), psychology (Cook et al. 2005) and computer science (Gamble and Goble 2011). There are three groups of trust models for social networks: graph-based trust models, interaction-based trust models and hybrid trust models (Sherchan, Nepal, and Paris 2013; Wang and Vassileva 2007). For example, the PageRank algorithms and their derivative methods are widely used to measure the entities’ reputation in the social network (Pujol et al. 2002; Yolum and Singh 2005; Page et al. 1999). Yao et al. introduce the ReputationNet and calculate the service’s reputation based on its designer’s
reputation and its popularity (Yao et al. 2012). These models calculate the reputation based on the opinions of other entities in the social network. Some researchers also calculate the reputation only based on the consumers’ rating (Wu et al. 2013; Wang, Sun, et al. 2011). These models require the details of the ratings from different consumers over time. Recently, some research studies tried to employ the QoS, combined with the Collaborative Filtering (CF), to calculate the reputation of the services for recommendation (Zheng and Lyu 2013; Zheng et al. 2011; Cao et al. 2013).

These approaches can yield good results when services have complete metadata. However, most of the MSEs actually do not contain detailed feedback from the users and the QoS for each service is resource-intensive to fetch. In fact, the performance measurement of manufacturing services is quite different from the product functionality and tolerance (Ellram, Tate, and Billington 2004) or the IT-attributes of services, QoS, from the field of computer science (Huang et al. 2011). Furthermore, the reputations of manufacturing services, service providers, service compositions and third-party developers do not isolate but interact with each other (Huang et al. 2013) in the MSE. Considering the relations in the MSE, this paper employs the historical information to calculate the reputation of the services in the MSE.

3. Models and problem definition
In order to describe the proposed approach more clearly, this section briefly presents the heterogeneous network model for the MSE and then formally clarifies the problems based on a four-stage composition behavioural model.

3.1. Heterogeneous network model
The MSE consists of massive manufacturing services and the complex relations among these services. Similar to the Web service ecosystem (Huang, Fan, and Tan 2012; Barros and Dumas 2006) and the business service ecosystem (Fan 2010; Huang et al. 2013; Luo, Fan, and Wang 2013), service providers in the MSE publish manufacturing services into the ecosystem and then those services are classified into different domains based on their functionalities. Developers choose one or more services and combine them into a composition or process to fulfil the consumers’ complex requirements. Note that the developer and the consumer can be the same one in the ecosystem. It is easy to identify three relations among the developers, compositions, services and providers.

(1) Supply relation: the service providers supply manufacturing services and offer certain capabilities in the MSE.
(2) Invoking relation: each composition invokes one or more manufacturing services to fulfil a complex requirement.
(3) Developing relation: third-party developers integrate invoked services to construct compositions.

Hence, just as shown in Figure 1, the MSE can be modelled as a network which consists of entities as nodes, including developers, compositions, services and providers as nodes, and relations as edges among them, including supply relation, invoking relation and developing relation. As there are four different kinds of nodes and three kinds of edges in the network, it is obviously a heterogeneous network.

Definition 1 (Heterogeneous network) The manufacturing service ecosystem can be modelled as a heterogeneous network $G = (V,E)$ where $V = \{De, Co, Se, Pr\}$ refers to the four different types of entities, consumers, compositions, services and providers in the ecosystem. Here $De$ refers to the developers who publish at least one composition in the ecosystem. $Co$ refers to the compositions which invoke at least one service. $Se$ refers to the services and $Pr$ refers to the service providers who provide at least one service in the ecosystem. $E = \{E_{DC}, E_{CS}, E_{PS}\}$ refers to the three kinds of relations among the four entities. $E_{DC}$
refers to the developing relations that the developers construct the compositions. \( E_{CS} \) refers to the invoking relations that the compositions invoke related services. \( E_{PS} \) refers to the supply relations that the providers offer the services.

**Definition 2 (Developer–composition network)**

Developer–composition network \( DC \) is used to present the developing relation \( E_{DC} \) between developer \( De \) and compositions \( Co. \) \( DC = \{ De, Co, E_{DC} \} \). It can be denoted as a \( n \times m \) matrix \( D = [d_{ij}]_{n \times m} \) and the element is \( d_{ij} = \begin{cases} 1 \text{ if } De_i \text{ develops } Co_j & \\
0 \text{ otherwise} & \\
\end{cases} \) where \( n \) refers to the number of developers and \( m \) is the number of compositions.

**Definition 3 (Composition–service network)**

Composition–service network \( CS \) is used to present the invoking relation \( E_{CS} \) between compositions \( Co \) and services \( Se \). \( CS = \{ Co, Se, E_{CS} \} \). It can be denoted as an \( n \times s \) matrix \( Y = [y_{jk}]_{n \times s} \) and the element is \( y_{jk} = \begin{cases} 1 \text{ if } Co_j \text{ invokes } Se_k & \\
0 \text{ otherwise} & \\
\end{cases} \) where \( n \) is the number of compositions and \( s \) is the number of services.

**Definition 4 (Provider–service network)**

Provider–service network \( PS \) is used to present the supply relation \( E_{PS} \) between providers \( Pr \) and services \( Se \). \( PS = \{ Pr, Se, E_{PS} \} \). It can be denoted as a \( p \times s \) matrix \( P = [p_{ok}]_{p \times s} \) and the element is \( p_{ok} = \begin{cases} 1 \text{ if } Pr_o \text{ provides } Se_k & \\
0 \text{ otherwise} & \\
\end{cases} \) where \( p \) is the number of providers and \( s \) is the number of services.

Similar to the previous work (Huang, Fan, and Tan 2014), a service–service network can be further extracted from the composition–service network and formalised as follows.

**Definition 5 (Service–service network)**

Service–service network \( SS \) is used to present the service co-occurrence relation in the compositions and it can be denoted by an \( s \times s \) matrix \( SS = [f_{lk}]_{s \times s} \) in which \( f_{lk} = \) the number of compositions the service \( k \) is invoked and \( f_{kl} \) is the frequency that service \( l \) and service \( k \) are used together in the same composition.

Obviously \( SS = Y^T \cdot Y \) and it is easy to get the main diagonal \( \Lambda = [f_{kk}]_{s \times s} \) for service–service network:

\[
f_{kk} = \sum_j y_{jk}
\]

As the service–service network contains only service as nodes in the network, the service–service network is a homogeneous network.

### 3.2. Behavioural model for service composition

In order to construct a composition, the developer firstly needs to select the related domains in the MSE. Secondly he/she picks the reputable service candidates in each domain. Thirdly he/she needs to find a way to connect the service candidates and makes sure that the service candidates can collaborate with each other. Finally he/she needs to build the service mediation (Li, Fan, et al. 2010) or sets up negotiation (Zheng, Martin, and Brohman 2012) if there is any mismatching in composing the selected services as a composition. Hence as shown in Figure 2, the composition behaviours can be classified into four steps.

1. **Service domain selection.** The developer firstly selects several related service domains which may fulfil the requirements, regardless of whether the domains are selected by developer or are suggested by the guideline.
2. **Service candidate identification.** In each domain, the developer selects the service candidates with higher reputation to construct the composition of higher quality.
3. **Service chain search.** For the given service candidates, the developer needs to figure out potential service operational chains which can connect the service candidates. The developer may not own the full knowledge about the composition in the MSE; hence, some potential services or service domains may be needed for the connection. As the service–service network somehow represents the historical collaboration patterns among services, the service chain here refers to a service sequence in the service–service network which can connect the selected service candidates.
4. **Service composition construct.** The developer selects the optimal chain to build the final composition. If there are any mismatches in the chain, the service mediations or negotiations can be developed to fix the mismatch problem.

### 3.3. Problem definition

Previous literature works focused on studying the static and dynamic property of the service network (Huang, Fan, and Tan 2012), offering the GPS-like method *ServiceMap* to get the service chains over the service–service network (Tan et al. 2011) and calculating the collaboration possibility of services based on link prediction to help in selecting the optimal service composition (Huang, Fan, and Tan 2014). All these works concentrate on the last two steps: *Service Chain Search* and *Service Composition Construct*. Different from them, this paper will focus on
the first two steps: Service Domain Selection and Service Candidate Identification. In order to facilitate the Service Domain Selection, the first task is obviously to organise the massive services in the MSE into domains and describe the functionality for each domain. Then the developer can select the related domain according to the given requirements. For the Service Candidate Identification, the effective approach is to calculate the reputation of services in each domain and then sort them in descending order according to their reputation. Therefore developers can select the services with higher reputation in each domain to improve the quality of the composition. Hence the two problems discussed in this paper can be summarised as follows.

(1) **Service clustering.** How to automatically cluster services into domains to facilitate the domain selection?

As discussed earlier, the purpose of this paper is to cluster the manufacturing services into domains through their descriptions and tags. The descriptions and tags are in fact a bag of words to describe the capability of manufacturing services; hence the Service Clustering problem can be formally defined as:

**Problem 1 (Service clustering problem)** Given the MSE is defined as $Se = \{S_1, \ldots, S_l\}$, each service in the MSE is labelled with a bag of words, $S_i = \{w_{i1}, w_{i2}, \ldots, w_{il}\}$. Find a method to extract the service domains $Domain = \{d_1, \ldots, d_D\}$ and calculate the membership degree for each service in each domain. $S_i = \{(d_i, m_{ii})\}_{1 \leq j \leq D}$. Here $s$ is the number of services, $l$ is the number of words, $D$ is the number of domains, $m_{i}$ refers to the membership degree of $S_i$ to the domain $d_i$.

(2) **Service reputation calculation.** How to calculate the service’s reputation to help selecting the high-quality candidate services?

Here some notations are introduced to understand the reputation calculation. Let $R_d = [rd_d]_{n \times 1}$ be a $n \times 1$ vector where element $rd_i$ refers to the reputation of developer $i$; $R_r$ be a $m \times 1$ vector $R_r = [ry_r]_{m \times 1}$ where element $ry_j$ refers to the reputation of composition $j$; $R_s$ be a $s \times 1$ vector $R_s = [rx_s]_{s \times 1}$ where element $rx_k$ refers to the reputation of service $k$; $R_p$ be a $p \times 1$ vector $R_p = [rp_p]_{p \times 1}$ where $rp_o$ refers to the reputation of provider $o$; $R_c$ be a $m \times 1$ vector $R_c = [rc_c]_{m \times 1}$ where $rc_j$ refers to the feedback from consumers for the composition $j$. As $R_c$ is the
feedback from consumers, it can be considered as the input for the reputation calculation. Thus the Service Reputation Calculation problem can be formally defined as follows.

**Problem 2 (Service reputation calculation problem)**
Given the networks – developer–composition network $D$, the composition–service network $Y$, the provider–service network $P$ and the service–service network $\Lambda$, as well as the consumers’ feedback for composition $R_c$, find a methodology to calculate the service’s reputation $R_x$.

In the following sections, Section 4 focuses on the service clustering problem and Section 5 deals with the service reputation calculation problem and finally Section 6 combines the clustering method and the reputation calculation method to offer DRSR for the service selection.

### 4. Topic model–based service clustering (WSTPCluster)

In this section, the service clustering method $WSTPCluster$ based on the topic model using LDA will be clarified.

#### 4.1. Content vector extraction

Here a four-step data pre-processing method, which is widely used to extract the meaningful content from a bag of words (Wu et al. 2014), is employed to generate a content vector from the manufacturing services’ descriptions and tags. Hence each service’s capability can be represented as a content vector.

1. **Original vector generating.** Firstly, the description of each service is considered as a bag of words and is split into a word vector according to the space.
2. **Pruning.** Secondly, the noise words and the general words which are not meaningful for the service’s functionality are removed: e.g., the articles such as a, an, the; the prepositions such as in, on, with, by, for, at, about, from, etc.; adverbs such as where, when, quite, etc.; general words such as api, service, user, information, etc.
3. **Suffix stripping.** Thirdly, using a Porter stemmer, the suffixes of all words that have the same stem are stripped. For example, $\text{map}$, $\text{mapping}$, $\text{maps}$, $\text{mappings}$ are replaced with the same stem $\text{map}$.
4. **Spell correcting.** Finally, as the suffix stripping may create some non-sense words, the spell correct tool is used to refine the error word. For example, the $\text{websit}$ generated by the stemmer is corrected as $\text{website}$; $\text{googl}$ is corrected as $\text{google}$, etc.

Taking the manufacturing service $\text{Ei3}^4$ as an example, after the four steps discussed above, a content vector which contains the following 70 words is generated:

application manufacturer gather process analyze large amount machine user provide mobile app web page dashboard report show key performance indicator describe performance uptime quality machine output actionable machine production valuable integrate fully enterprise traditional implementation technical challenge make daunt task company web solve provide tool link shopfloor erp quality enterprise computing platform client communicate cloud server order access specific machine restful call response formatted machine manufacture production monitor enterprise

Note that the words here are not unique. In fact, if a word appears more than one time, the word may be more important for the service. For example, the word enterprise appears three times.

#### 4.2. Topic-based domain extraction

In the service ecosystem, each service offers functionalities which belong to different service domains. Each domain contains a bag of words which can describe the domain’s functionality. Obviously the services and the words are observable while the service domains are latent. Hence it is easy to build the LDA-based method to learn services’ latent domains and calculate their memberships to each domain. As shown in Figure 3, $\theta$ denotes the per-

Figure 3. The LDA model for the service domain extraction. Each service belongs to different service domains. Each service domain contains a bag of words which describe the domain’s functionality.
service domain distribution, $\varphi$ represents the per-domain word distribution.

Hence the generative processes of LDA model can be summarised as follows:

1. For each domain $d = 1, \ldots, D$, draw a multinomial $\varphi_d$ from a Dirichlet prior $\beta$. $\varphi \sim \text{Dirichlet}(\beta)$

2. For each service, $S_i \in S$: 
   a. Draw the domain distribution $\theta_i \sim \text{Dirichlet}(\alpha)$.
   b. For each word $w_{i,j}$ in the service’s content vector $C_i$:
      i. Generate a word $z_{i,j}$ from the multinomial distribution $z_{i,j} \sim \text{Multinomial}(\theta_i)$;
      ii. Generate a word $w_{i,j}$ from the multinomial distribution $w_{i,j} \sim \text{Multinomial}(\varphi_{z_{i,j}})$;

In order to train the LDA model and estimate the parameters, the Gibbs sampling process and the EM algorithm can be employed to solve the problem (Blei, Ng, and Jordan 2003). Due to the space limitation, the details for the Gibbs sampling process and EM algorithm are not discussed here and anyone who is interested please refer to Blei, Ng, and Jordan (2003).

After the training, the $D$ service domains in the MSE can be extracted and the membership degree for each service in each domain is generated. In fact, the relation between the functional words and services are transformed as a three-layer network, just as shown in Figure 4.

Thus the services and service domains in the ecosystem can be defined as:

1. **Service domain**: $\text{Domain} = \{d_1, \ldots, d_D\}$ where $D$ is the number of domains. $d_j = \{(w_l, f_{jl})| 1 \leq l \leq N \}, 1 \leq j \leq D$ where $w_l$ refers to a content word and $f_{jl}$ the frequency of the word $w_l$ in the domain $d_j$.

2. **Service**: $S = \{S_1, \ldots, S_s\}$ where $s$ is the number of services in the ecosystem. $S_i = \{(d_j, m_{ij})| 1 \leq j \leq D \}$ where $m_{ij}$ refers to the membership degree of $S_i$ in the domain $d_j$.

Furthermore, from the service perspective, the service domain can be defined as:

3. **Service domain**: $\text{Domain} = \{d_1, \ldots, d_D\}$ where $D$ is the number of domains. $d_j = \{(s_i, m_{ij})| 1 \leq i \leq n \}, 1 \leq x \leq D$ where $m_{ij}$ refers to the membership degree of $S_i$ in the domain $d_j$.

5. **Unified reputation propagation framework**

In the MSE, the reputation of services, providers, compositions and developers influence each other which brings a great challenge for the reputation calculation. Considering the relations among these four entities in the MSE, this section presents the unified reputation propagation method (URPM) over the heterogeneous network to calculate the services’ reputation and then discusses some simplified models.

5.1 **Unified reputation propagation method (URPM)**

In the MSE, the following basic phenomenon can be used for the reputation calculation:

**Phenomenon 1.** Highly reputable providers offer many highly reputable services which are invoked in many highly reputable compositions.

**Phenomenon 2.** Highly reputable developers develop many highly reputable compositions which invoke the highly reputable services.

Furthermore, the compositions are used to fulfill the consumers’ requirement and the consumers offer some feedback to the compositions. Thus the compositions with high reputation get high feedback from the consumers.

**Phenomenon 3.** Highly reputable compositions attract high consumers’ attention.

Based on the above phenomenon, adding the reputation influence from services to providers and the reputation influence from compositions to developers, it is easy
to get the reputation propagation framework which is shown in Figure 5.

(1) Composition reputation

The reputation of the composition comes from three parts: the reputation of the invoking services, the reputation of the developers and the feedback of the consumers.

\[ R_y^+ \leftarrow \mu R_c + \alpha D^TR_d + \lambda Y^{-1}R_x + \xi_y \]  

(2)

Here \( \mu + \alpha + \lambda = 1 \). \( \mu R_c \) refers to the reputation from the feedback of the consumers; \( \alpha D^TR_d \) refers to the reputation from its developers; \( \lambda Y^{-1}R_x \) refers to the reputation from its invoking services; \( \xi_y \) refers to the random factors. Then the reputation is normalised as:

\[ R_y \leftarrow \frac{R_y}{1 \bullet R_y} \]  

(3)

(2) Developer reputation

The reputation of the developer comes from the compositions he/she has ever published.

\[ R_d^+ \leftarrow \beta DR_y^+ + \xi_d \]  

(4)

Here \( \beta DR_y \) refers to reputation from the compositions the developers have ever published and \( \xi_d \) refers to the random factors. Then the reputation is normalised as:

\[ R_d \leftarrow \frac{R_d}{1 \bullet R_d} \]  

(5)

(3) Service reputation

The service reputation comes from the reputation of the compositions it has been invoked and the reputation of its providers.

\[ R_x^+ \leftarrow \omega Y^TR_y^+ + \theta P^TR_p + \xi_x \]  

(6)

Here \( \omega + \theta = 1 \). \( \omega Y^TR_y^+ \) refers to the reputation updating from compositions, \( \theta P^TR_p \) refers to the reputation from the providers and \( \xi_x \) refers to the random factors. Then the reputation is normalised as:

\[ R_x \leftarrow \frac{R_x}{1 \bullet R_x} \]  

(7)

(4) Provider reputation

The reputation of the providers comes from the services he/she has published in the ecosystem.

\[ R_p^+ \leftarrow \rho PR_x^+ + \xi_p \]  

(8)

Here \( \rho PR_x \) refers to the reputation from services, \( \xi_p \) refers to the random factors. Then the reputation is normalised as:

\[ R_p \leftarrow \frac{R_p}{1 \bullet R_p} \]  

(9)

Giving different combinations of the parameters, the reputation propagation models for the service reputation obviously will be different. The following section discusses several simplified models by setting the same specific parameters.

5.2. Model simplification

5.2.1. Composition–service model

In this model, the impacts of reputation from the developers and providers are ignored, which means \( \alpha = 0 \), \( \beta = 0 \), \( \theta = 0 \), \( \rho = 0 \). Considering whether the feedback of the services’ reputation to the compositions is taken into account, two simplified methods can be easily extracted from the UPRM to calculate the services’ reputation.
Here the random factors in the model are ignored and the following is obtained:

\[ R_x^+ = \lambda Y \Lambda^{-1} R_c + \mu R_c \]  \hspace{1cm} (11)

\[ R_x^- = \omega Y^T (\lambda Y \Lambda^{-1} R_c + \mu R_c) \]
\[ = \omega \lambda Y^T Y \Lambda^{-1} R_c + \omega \mu R_c Y^T R_c \]  \hspace{1cm} (12)

\[ R_x^t = \omega \mu [(\omega \lambda Y^T Y \Lambda^{-1})^{t-1} + \ldots + I] Y^T R_c \]

As \( \omega \mu Y^T R_c \) is the constant, the iterative equation for the reputation of each service can be rewritten as:

\[ r_{x_i}^+ = \omega \lambda \sum_k f_{ki} r_{x_k} + c \]  \hspace{1cm} (13)

Apparently, the reputation of the service is related to the reputation of its neighbours which is similar to the PageRank algorithm method (Page et al. 1999) based on the service network. Thus it is named as PPRM.

5.2.2. Developer–composition–service model

In this model, the impacts from the service providers and the random factors are ignored which means \( \theta = 0, \rho = 0, \xi_r = \xi_d = \xi_c = 0 \) and then:

\[ R_x^+ = \mu R_c + \alpha D^T R_d + \lambda Y \Lambda^{-1} R_c \]

\[ R_d = \beta DR_x, R_x = \omega Y^T R_y \]  \hspace{1cm} (15)

Hence:

\[ R_x^+ = \mu R_c + (\alpha \beta D^T D + \lambda \omega Y \Lambda^{-1} Y^T) R_y \]

\[ R_x^t = \mu [\alpha \beta D^T D + \lambda \omega Y \Lambda^{-1} Y^T]^{t-1} + \ldots + I] R_c \]  \hspace{1cm} (16)

Here \( t \) refers to the number of iterations. Obviously if set \( \alpha = 0 \), this model reduces to the PPRM method; further if set \( \lambda = 0 \), it reduces to be the TPRM method. As this model considers the developers’ reputation, it is named as Developer-aware Reputation Propagation Method (DRM).

The illustrations of these simplified models are shown in Figure 6.

5.3. Consumers’ feedback strategy

From the discussion above, it can be seen that the reputation is related to the consumers’ feedback for the compositions. In order to show the influence of the consumers’ feedback, two consumers’ feedback strategies are defined as:

(1) Equivalent feedback strategy (EI)

The hypothesis here is that the consumers’ feedback for each composition is equivalent, hence \( R_c = \frac{1}{m} 1 \).
For the EI strategy, the influence from the consumers’ feedback is obviously ignored. Thus the reputation calculation is only based on the relations among the four entities in the MSE.

(2) Popularity-based feedback strategy (PI)

The PI strategy takes the consumers’ feedback on the compositions into account. As most MSE does not offer the detail of feedback information but only provides the rating and visited number. Thus in this paper, the feedback of the composition is defined as the product of its rating and visited number. So $R_c = \frac{cp_i \times \text{Visited}(Co_i)}{\sum_k \text{Rate}(Co_k) \times \text{Visited}(Co_k)}$ (17)

Here $\text{Rate}(Co_i)$ refers to the rating of the composition $Co_i$ and $\text{Visited}(Co_i)$ refers to its visited number.

6. Domain-aware reputable service recommendation

6.1. Method

Until now, the WSTPCluster has been used to calculate each service’s membership degree for each domain, it is easy to get the services with top membership degree in each domain and consider them as the domain-specific services.

Based on the URPM, the services’ reputations, which can reflect the collective perception from the historical information, are calculated. Thus the services in each domain can be re-ranked by their reputations so that the top trustworthy services in each domain can be recommended to the developers. Thus the recommendation for the developers is offered in the ‘one domain one list’ pattern. Here it is named as DRSR. The detail is shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Combining WSTP Cluster and URPM for Recommendation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm: DRSR</td>
</tr>
<tr>
<td>Input:</td>
</tr>
<tr>
<td>(1) The MSE $G = {V, E}$ $V = {De, Co, Se, Pr}$ $E = {E_{DC}, E_{CS}, E_{PS}}$</td>
</tr>
<tr>
<td>(2) URPM Propagate Parameters: $\alpha, \beta, \lambda, \omega, \theta, p, \mu$</td>
</tr>
<tr>
<td>(3) WSTPCluster Parameters: Domain Number $D$, Domain Service Number $Q$, Recommend Service Number $K$</td>
</tr>
<tr>
<td>Output:</td>
</tr>
<tr>
<td>(1) WSTPCluster:</td>
</tr>
<tr>
<td>01. Running WSTPCluster to extract the $D$ domains in the service ecosystem</td>
</tr>
<tr>
<td>02. Calculate each service’s membership degree to each domain</td>
</tr>
<tr>
<td>(2) URPM:</td>
</tr>
<tr>
<td>03. Run URPM to calculate each service’s reputation based on the historical information</td>
</tr>
<tr>
<td>(3) Domain-aware reputable recommend</td>
</tr>
<tr>
<td>04. For each domain, get the services with top-$Q$ membership degree and consider them as domain-specific services</td>
</tr>
<tr>
<td>05. Re-rank the domain-specific services based on their reputation and recommend the top-$K$ services for the developers</td>
</tr>
</tbody>
</table>

6.2. Implementation

Based on the discussion above, Figure 7 presents a simplified architecture for the DRSR in the MSE:

Composition editor offers a client browser for the developers including the domain view module and the service chain view module. Domain View (DV) module helps the developer to get the related domains for the compositions and visualise the domains as a domain network. Service Chain View (SCV) module shows the selected services
for each domain and the service chains to connect the service candidates, as well as the potential compositions for the developers.

**Recommendation engine** consists of four engines for the domain-aware service recommendation. **Service Cluster Engine (SCE)** employs the service contents to cluster the services into different domains and to calculate the services’ membership degree to the service domains; **Service Network Engine (SNE)** constructs the heterogeneous network model and then calculates the service reputation and service link possibility to form the potential service network. **Domain Service Register (DSRe)** collects data from SCE to label tags for each domain and collects data from SNE to sort the services in each domain based on their reputations. **ServiceMap Engine (SME)** uses the information from SNE to search the potential service chains for the selected service candidates in SCV and then calculates their possibilities. Here SCE and SNE run off-line while DSRe and SME are the real-time engines.

### 7. Empirical Study on ProgrammableWeb

#### 7.1. Experiment data-set

ProgrammableWeb\(^5\) is by far the largest online repository of services and their compositions. Though it is not a specific data-set consisting of manufacturing services, each service in ProgrammableWeb contains the information such as name, provider, category, publication date, tags and description; each compositions in it contains the information such as name, created date, developer, the list of services in it, description, visited number and the user rating; and each developer in it contains the information including name and the compositions he/she registered. Hence ProgrammableWeb is selected as an example to study the effectiveness of the proposed approach in this paper. The data-set consisting of services, compositions and developers from June 2005 to March 2013 is fetched from the repository in this paper.

In order to examine the approach’s performance, the data-set is separated into two sets: one set contains the compositions published from June 2005 to August 2012 and it is used as the **Training Data-Set**; the other one contains the compositions published from September 2012 to March 2013 and it is used as the **Testing Data-Set** to quantify the effectiveness of the proposed method. Table 2 summarises some basic statistics of the experiment data-set.

### 7.2. Evaluation metrics

#### 7.2.1. Ground truth definition

The goal of this paper is to identify the domain-aware reputable services for the developers. The hypothesis here is that the higher the service’s reputation is, the higher possibility that the service will be reused in the future. Thus in order to evaluate the performance of recommendation, the existence status of the services in the testing period is used as the ground truth.

\[ y(Se_j) = \begin{cases} 1 & \text{Se}_j \text{ exist in the testing period} \\ 0 & \text{otherwise} \end{cases} \]  

(18)

As the invoked frequencies of different services during the testing period are different, the frequency \( f(Se_i) \) can be used as the ground truth. If a service does not appear in the testing period, then \( f(Se_i) = 0 \).

#### 7.2.2. Accuracy measurement

In order to evaluate the performance of the reputation propagation methods, this paper use the area under the receiver operating characteristic (ROC) curve (AUC) due to its effectiveness for the unbalanced data-set (Fawcett 2006). Learning from the Wilcoxon–Mann–Whitney algorithm (Bradley 1997), the services in Training Data-Set are separated into two parts based on the ground truth: the services reused in the Testing Data-Set are considered as the positive instances \( I^+ \) and the other as the negative instances \( I^- \). Hence the AUC metric is defined as follows:

\[
AUC = \frac{\sum_{i=1}^{\|I^+\|} \sum_{j=1}^{\|I^-\|} f(s_i^+, s_j^-)}{\|I^+\| \|I^-\|}
\]  

(19)

where \( \|I^+\| \) refers to the number of the positive instances, \( \|I^-\| \) refers to the number of the negative instances, \( s_i^+ \in I^+ \), \( s_j^- \in I^- \) and \( f(s_i^+, s_j^-) \) is the indicator function which is defined as follows:
\[
f(s_i^+, s_j^-) = \begin{cases} 
1 & r(s_i^+) > r(s_j^-) \\
0.5 & r(s_i^+) = r(s_j^-) \\
0 & r(s_i^+) < r(s_j^-) 
\end{cases}
\]  

(20)

where \( r(s) \) refers to the reputation of the service \( s \).

In order to evaluate the performance of DRSR method, this paper further considers the Mean Average Precision (MAP). This is because MAP has been commonly used in the information retrieval and recommendation system (Yue et al. 2007):

\[
\text{MAP@K} = \frac{1}{D} \sum_{j=1}^{D} \frac{1}{m_j} \sum_{h=1}^{m_j} \frac{h}{\pi(s_{j,h})} 
\]

(21)

where \( D \) is the number of domains, \( m_j \) is the reused number of services in domain \( j \) and \( \pi(s_{j,h}) \) is the position of the reused service \( s_{j,h} \) in the ranking list for domain \( j \).

7.2.3. Long tail measurement

Recommending the popular services for developers is easier but more trivial (Yin et al. 2012), while the peripheral services in long tail tend to be more specialised in function and they can bring more innovation and value (Yu and Woodard 2009). Hence a discounting factor based on the popularity is introduced into MAP to get the Mean Average Popularity–Inverse Precision (MAPIP) as follow:

\[
\text{MAP@K} = \frac{1}{D} \sum_{j=1}^{D} \frac{1}{m_j} \sum_{h=1}^{m_j} \frac{h}{\pi(s_{j,h}) \cdot \text{pop}(s_{j,h})} 
\]

(22)

where \( \text{pop}(s_{j,h}) \) refers to the service’s popularity in the Training Data-Set.

7.3. Performance comparison

7.3.1. Reputation propagation performance evaluation

As discussed in Section 5, this paper introduces four reputation methods (URPM, PRRM, TPRM, DRM) and two consumers’ feedback strategies (EI, PI). Hence there exist eight reputation propagation methods: URPM+EI, URPM+PI, PRRM+EI, PRRM+PI, TPRM+EI, TPRM+PI, DRM+EI, DRM+PI. In order to show the effectiveness of the proposed methods, this paper considers the following basic line methods.

1. Top-degree-based reputation method (TDRM):

   The reputation of each service is its network degree in the service network, which means that the higher frequency the service collaborates with others, the higher reputation the service gain.

2. Homogeneous PageRank reputation method (HPRRM):

   The reputation of each service is its page rank value in the service network.

   From the definitions of the 10 methods, it can be seen that: the services’ reputation in HPRRM is based on the PageRank algorithm over the service network; hence, the PageRank algorithms and its derivative methods (Pujol et al. 2002; Yolum and Singh 2005; Page et al. 1999) can be considered as HPRRM in this paper. PRRM+EI and PRRM+PI are similar to the PageRank-based methods. However, they are calculated over the heterogeneous network. The services’ reputation in TDRM and TPRM+EI is based on the invoked frequency; the services’ reputation in TPRM+PI further considers the feedback from the consumers. As there is no detail feedback from the consumers, the TDRM, TPRM+EI and TPRM+PI can be considered as the simplified methods for the consumers’ rating-based methods (Wu et al. 2013; Wang, Sun, et al. 2011) in the MSE. The DRM+EI and DRM+PI take the reputation from the developers, the feedback from consumers into account; thus these two models are similar to the ReputationNet (Yao et al. 2012) and can be considered as the variants in the MSE.

   Also the TDRM and the HPRRM are based on the service network, a homogeneous network which nodes are all services. Hence the TDRM and the HPRRM can be considered as the methods based on the homogeneous network. However, the other eight methods are all based on the networks which contain more than one type of nodes. Thus they are considered as the methods based on the heterogeneous network.

   For TDRM, given the service network, the network degree for each service is calculated and considered as its reputation. For HPRRM, given the service network, the PageRank algorithm is run and then the page rank value is considered as each service’s reputation.

   For the other eight methods, given the heterogeneous network, setting the iteration number as 10, the iteration methods defined in Section 5 are used to calculate the service’s reputation. Finally the AUC metric can be calculated based on the definition discussed before.

   Obviously, different propagate parameters result in different performances.

   For the TPRM, as the parameters \( \mu, \omega \) does not affect the reputation calculation due to the normalisation phase, the parameters are set as \( \mu = \omega = 1 \).

   For the PRRM, similarly, \( \omega \) does not affect the calculation, hence it is set as 1. For the parameters \( \mu, \lambda \), as \( \mu + \lambda = 1, u, \lambda \in (0, 1) \), this paper varies \( \mu \) from 0 to 1 in the step 0.001, and for each \( \mu, \lambda \) pair, run the iteration and calculate the AUC metric. Finally choose the optimised \( \mu = 0.999, \lambda = 0.001 \) with the largest AUC as the final parameters.

   For the DRM, \( \beta, \omega \) do not affect the calculation, hence they are set as 1. For parameter \( \alpha, u, \lambda \), as \( \mu + \lambda + \alpha = 1, \alpha, u, \lambda \in (0, 1) \), this paper varies \( \alpha, u, \lambda \)
with a step 0.001 and for each combination \((a, u, \lambda)\), calculates the AUC metric and get the optimised parameters with the largest AUC as the final parameters. For the data-set in this paper, the optimised parameters are \(a = 0.1, u = 0.899, \lambda = 0.001\).

For the UPRM, \(\beta, \rho\) do not affect the calculation, hence they are set as 1. For the other parameters, \(\omega + \theta = 1, \mu + \lambda + \alpha = 1, \omega, \theta, \alpha, u, \lambda \in (0, 1)\), similarly, they are varied with a step 0.001 and it is easy to get the optimised parameters \(\omega = 0.999, \theta = 0.001, \alpha = 0.1, u = 0.899, \lambda = 0.001\).

Table 3 summarises the AUC for all the 10 methods.

From Table 3, it can be seen that the four reputation propagation methods are sensitive to the feedback strategies and the methods with PI strategy outperform the ones with EI strategy, which means that in practice, taking the consumers’ feedback into account can help to improve the performance. Also all the methods based on the heterogeneous network gain higher performances than the methods based on the homogeneous network. The UPRM method with PI strategy actually achieves the best performance. Comparing with TDRM, it gains a 17.5% improvement. The explanation is that the heterogeneous network contains richer information, not only the relations among the four entities (providers, services, compositions and developers) in the ecosystem, but also the differences among the compositions.

From the experiments shown above, it can be concluded that: (1) The consumers’ feedback for compositions can improve the accuracy of the reputation ranking. (2) The heterogeneous network which contains richer information can gain a better performance for the reputation ranking than the homogeneous network.

### 7.3.2. Domain-aware reputable recommendation performance

In order to study the performance for the domain-aware reputable recommendation methods, this paper considers the baseline method which only ranks the services based on the membership degree to the domain:

**Domain membership-based recommendation method (DMRM):** The services in each domain are re-ranked based on its domain membership degree and the ones with the top-K domain membership degree are recommended to the developers.

First of all, for the services in the Training Data-Set, WSTPCluster is run to detect the service domains based on the services’ descriptions and tags. Here the domain number is set as 40, \(D = 40\), and the service number in each domain is set as 50, \(Q = 50\). Figure 8 shows the 40 domains and the top 5 words in each domain which are generated by the WSTPCluster. Also each service’s membership degree distribution to different domains is calculated. Taking the service Ei3 as example, just as shown in Figure 9, it can be seen that Ei3 belongs to four different domains: Domain36 offers cloud-based management platform for enterprise, Domain33 offers the monitor and analyse tools over Internet, Domain20 means the services are based on restful format, Domain 5 represents collaborative tools for project management.

After the clustering, the TDRM, HPRRM, TPRM+PI, PRRM+PI, DRM+PI and UPRM+PI are run for the Training Data-Set to calculate the services’ reputation. At the same time, the DMRM is used to get the services’ membership to each domain and the membership is considered as its domain-specific reputation. For the services in each domain, they are re-rank based the reputation based on the seven different methods. Finally, the Testing Data-Set is used to evaluate the accuracy and long tail performances for each method. From Figure 10, it can be seen that DMRM performs the worst because it only considers the service’s functionality but ignores the historical information. TPRM achieves a better performance than HPRRM in MAP but worse in MAPIP, which means that HPRRM can recommend the long tail services better than the TPRM. This is because TPRM is based on the service network degree while HPRRM is based on the page rank value. UPRM with PI achieves the best performance in MAP and MAPIP, TPRM with PI, PRRM with PI, and DRM with PI gain a comparable performance. Compared with TPRM, UPRM with PI gets 27.04% improvement for the average MAP. For the average MAPIP, it achieves 26.43% improvement than HPRRM.

Hence it can be concluded that taking the historical information into account can help to improve the recommendation performance while the domain-aware reputable recommendation method based on heterogeneous network can gain higher performances in both the accuracy and the long tail recommendation compared with the methods based on the homogeneous network.

### 8. Conclusion

With the wide adoption of SOA in the manufacturing sector (Huang et al. 2011) and the speedy promotion of the cloud manufacturing (Xu 2012), the manufacturing enterprises and organisations are turning to publish their manufacturing capabilities as services to collaborate with each other over Internet. Thus a rapidly growing
Figure 8. 40 domains with top 5 words generated from the services’ descriptions.

Figure 9. Membership degree distribution and the top words in the related domains for the service Ei3.

Figure 10. Domain-aware service recommendation performance comparison. The larger MAP or MAPIP is, the better the performance is. (a) URPM with PI achieves the best performance in MAP for different top-K (b) URPM with PI achieves the best performance in MAPIP for different top-K.
manufacturing service ecosystem (MSE) which consists of large number of manufacturing services and their compositions can be observed nowadays. Automatically clustering manufacturing services into different domains and identifying the reputable ones in each domain can help to facilitate service selection and construct the composition/process of high quality. Hence they have become two important issues for the management of the MSE.

In this paper, a heterogeneous network model is introduced to model the MSE. The four-step composition behaviour model is firstly proposed to define the service clustering problem and the service reputation calculation problem. Secondly, the WSTPCluster, the service clustering method based on LDA topic model, is employed to automatically detect the service domains through the services’ descriptions and tags. Thirdly, the URPM, over the heterogeneous network, is presented to calculate the services’ reputation according to the complex relations in the MSE and the feedback from the consumers. Finally combining WSTPCluster and URPM, DRSR method is proposed to help the developers to select high-quality services in each domain.

Based on the comprehensive set of experiments on the real data-set, ProgrammableWeb, it can be seen that: (1) The reputation propagation methods which consider the consumers’ feedback gain 0.8% ~ 10.3% improvements in AUC. (2) Compared with the methods only based on the homogeneous service network, the URPM over the heterogeneous network gains 17.5% improvements in AUC. (3) Compared with the method just based on the functionality, the DRSR method achieves a significant improvement in both accuracy and long tail recommendation. Also it achieves 27.04% improvements in average MAP and 26.43% improvements in average MAPIP than the methods based on homogeneous service network.

The limitation of this paper is that the data for experiment are not specific to manufacturing services. However, the proposed approaches can be used directly in the MSE because they are just based on the services’ descriptions, tags and the historical collaboration information.

In the future, some empirical data specific from the manufacturing sectors such as the well-known clothing manufacturing ecosystem in Ningbo (Li, Fan, and Li 2011) or the Boeing Company which consists of 13,000 + professional service suppliers in more than 100 countries (Huang et al. 2011) will be collected and used to improve the effectiveness of domain-aware reputable recommendation method. Also as the MSE is not static but evolving over time, how to take the evolution mechanism into account may be helpful to improve the performance.

Funding
This work is partially supported by the National Natural Science Foundation of China [grant number 61174169], the National Key Technology R&D Program [grant number 2012BAF15G01] and the Doctoral Program Foundation of Institutions of Higher Education of China [grant number 20120002110034].

Notes
5. www.programmableweb.com

References


