

A Social Network Approach in Semantic Web Services Selection using Follow the Leader Behaviour

Abstract

Automatic discovery of web services is a crucial task for e-Business communities. Locating and selecting "the best" web service from a vast number of similar services that matches the user's requirements and preferences is a cognitive challenge and requires the use of an intelligent decision making framework. This paper develops a flexible ontological architecture and framework for Semantic Web Service Selection that exploits Goldbaum's innovative "Follow the Leader" model originally designed as an analytic tool for studying social network behavior and evolution. The framework proposes two new ontologies integrated in a recommender system, which guides a user to select the best service that matches their requirements and preferences. We test and evaluate several behavior of market leader scenarios using a simulation agent.

1 Introduction

Semantic Web Services (SWS) extend web services with an explicit representation of meanings using ontological structures and formal reasoning mechanisms. SWS provide advanced facilities for automated discovery, selection, composition, contracting, and execution of dynamic Web Services. Interoperability is the main motivation for Web Services [1], however automation of information use and dynamic interoperability are the objectives of the Semantic Web and Semantic Web Services.

Web Service Selection is a complex process where a service that best satisfies user preferences is selected from a set of candidate services usually returned from a service discovery process based on user requirements [2]. As per the selection criteria, various non-functional properties such as service level agreements (SLA) and quality of services (QoS), can be used and expressed as user preferences. Such non-functional descriptions can capture constraints over the functional and behavioral service descriptions. Selection can thus restrict the consumption of service functionality by a specific condition, e.g., quality of service preference may restrict the usage of a service.

Web services non-functional properties such as response time, throughput, availability, and reliability are difficult for the user to determine and control. Users are usually not willing to spend time describing their detailed preferences to the system, even less to assign weights to them, especially if they do not have a clear understanding of the effects and results of this input. Moreover, users may not even be aware of their explicit preferences. So,

users who want to find new Web Services often seek help from their friends, peers, experts and business partners who have relevant experience. Capturing and specifying user preferences are one of the most complex problems in the selection process. Some approaches make selections based on non-functional properties ontology or synthesize selection using previous queries of similar situation.

To illustrate the challenges involved in SWS selection we provide the following example, which illustrates the key difficulties and at the same time motivates our approach.

Motivating Example:

Bob is seeking an insurance company to insure his home. He lives in the same area as his friend Adam who has already taken out home insurance. Additionally, Bob plans to attend a conference in the USA; he is in the same situation as his colleague Mary. This is the first time Bob buys a house, he does not want to spend much time on analyzing insurance features, he would rather have the same insurance as his friend Adam. Furthermore, he prefers to seek advice from Mary to book a hotel and flight.

Adam's preferences' to insure his house differ from Mary's preferences to book a hotel and a flight. Two sets of different preferences in different contexts that meet Bob's needs.

What if Bob does not know Adam or Mary? Can he get reasonable advice from somebody living in his area? If not then he embarks on a tedious and time consuming process to differentiate between a vast number of services in which all of them may match his request from a functionality perspective, but vary in their non-functional properties.

Semantic web service selection in this paper is designed to provide context-awareness, effective capture and identification of user preferences and high levels of reusability of the best services and preferences based on the notion of market leadership. We model market leadership influences using Goldbaum's [3] novel "Follow the leader" agent model.

The work presented in this paper builds on our observation that the three important technologies Semantic Web Services, social networks and recommender systems possess the kinds of catalytic synergies required to address the complex problem of service selection. All three technologies are currently enjoying unprecedented interest and investment in both research and industry. In this paper we demonstrate the collective power of these technologies using an innovative

and powerful agent behavior model – “Follow the Leader” [3].

The remainder of the paper is organized as follows. Section 2 provides review of three primary technologies: Semantic Web services, social networks and recommender systems. The motivation of this paper is then summarized in Section 3. In Section 4, we first propose and discuss the conceptual model of “Follow the leader”, and then we propose the reference architecture.

In the final section, we evaluate through simulation how accurately our “Follow the Leader” approach recommends services relevant to a user’s preferences.

2 Overview of Key Technology Challenges

In this section, we provide an overview of three research technologies, on which our Market Leadership Selection Framework (MLS) is constructed. We emphasize both technical and philosophical issues that have inspired the “Follow the Leader” architectural design.

2.1 Semantic Web Services

SWS require two main processes in order to execute, namely Service Discovery and Service Selection. Semantic Web Service Discovery seeks to find a match between service requirements and service advertisements based on the semantic description of services. The discovery mechanism does not consider quality of advertised Web services; it considers only the service description.

Web Service Selection is the process that chooses a single service that best satisfies user preferences from a set of candidate services which the Service discovery process identifies [2]. Web Service Selection is a complex and challenging task, especially if it takes a variety of different non-functional properties into account. The service selection involves (1) specifying user functional and non-functional requirements, (2) matching of the service offerings against user requirements, and (3) selecting the best match through aggregation and evaluation matched results.

Different approaches have been employed in semantic web services selection: semantics based, policy based and trust and reputation based. The majority of semantic web services selection approaches are based on the semantics of non-functional properties and QoS where WSDL-S and OWL-S are used to describe the non-functional properties and corresponding interdependencies and hierarchies of properties as shown in [4]. Policy based service selection approaches allow the user to specify the non-functional requirements by encoding these in a QoS policy model as in [5]. Trust and reputation based methods base the selection process on the classification of trust and reputation systems, where feedback is provided from communities or recognized agencies, e.g. methods in [6].

For SWS to be successful, it is mandatory to have powerful mechanisms to capture user requirements that

are both user friendly and expressive enough to capture a large quantity of complex preferences and the logical relations between preferences. Preferences need to be prioritized and can be associated with a corresponding level of importance (weight). This preference information not only needs to be captured but also managed and matched automatically and effectively by the SWS selection process.

2.2 Social Networks

The increasingly popular Web 2.0 sites like the social networks LiveJournal, Tribe.net, and Facebook provide an unprecedented structured and digitally represented social network. As a result users are now considered as plain web resources [7]. There is great research interest in understanding and harnessing Web 2.0 and some researchers have found social network analysis (SNA) useful in analyzing such networks; others provide models to leverage the semantics of their representation.

Several Ontologies are used to represent social networks. The most popular is FOAF¹, used for describing people, their relationships and their activity. As a consequence, researchers have applied classical SNA methods to FOAF.

The adoption of standardized Ontologies and programs like OpenId for online social networks (see chi.mp for an example) will lead to increasing interoperability between them and to the need for uniform tools to analyze and manage them.

2.3 Recommender Systems

In recent years, research on using recommender systems (RSs) for Web Service discovery and selection has emerged as an effective approach [8]. [9], describe a web service selection framework, which combines a recommender system with semantic matching of service requirements. The system allows users to rate Web Services and provide facilities such as similarity computation, prediction and evolution algorithms for recommending services. [10] adopt a so-called implicit culture framework; where it is possible to elicit the community culture by observing the interactions of people with the environment.

Recommendation strategies use domain-specific or historical information about users [11] or items to decide which specific recommendation technique provides the highest accuracy for the current user. Recommender systems have proven to be useful in various application domains [12]. However, current solutions are usually ad-hoc systems, which are tightly coupled with the application domain and suffer from “cold start” issue.

User feedback can be either implicit or explicit depending on the way user satisfaction is captured [13]. Although explicit methods capture user satisfaction more

¹ www.foaf-project.org

accurately, they are costly and often users do not cooperate.

We believe that an intelligent recommender system should provide context aware recommendations of Web Services without the users needing to have extensive personal experience of all available alternatives.

3 Motivation: The need for more effective solutions

Personalization [14] can be defined as the use of technology and user information to tailor ecommerce and m-commerce interactions between a business and each individual user. User Semantics [15] is a concept of the human's perception of the physical world based on his or her personal background information. User Semantics is influenced by the context of the users such as the location, time, and their social situation such as whether the user is alone, with her friends, co-workers, or boss.

Nowadays users are playing a central role in the creation of web contents [16]. Social networking sites such as FOAF, through which users can maintain an online network of friends or associates for social or business purposes have experienced extensive growth. One of the important issues is how we can find user's relevant information. To address this issue [16] propose a keyword extraction method to extract personal information from the Web.

In order to fully reflect the complexity of the different aspects, they may be represented as personal Ontologies. This will allow clear and explicit representation of a user's specific semantics. It may be possible to represent relationships among the personal Ontologies with other Ontologies to clearly specify where the commonalities and differences arise [15].

User behavior analysis can make web services more intelligent and personal. Many researchers are interested to find user semantics', which already have been on the Web, and apply Semantic Web technologies to them.

Market Leadership

Market leader (best service) is one of the services with high capability. Market leadership can drive the evolution of service adoption, and Goldbaum's "Follow the Leader" model guides the formation of structures in social networks where leaders seek to be early adopters of a subsequently popular service. Users of SWS form a network unto themselves and service providers seek to exploit positive and lucrative network effects. Follow the leaders in dynamic social networks [3], is a model of opinion formation with dynamic confidence in agent-mediated social networks where the profiling of agents as leaders or followers is possible. An opinion leader is specified as a highly self-confident agent with strong opinions. An opinion follower is attracted to those agents in which it has more confidence. In our context an opinion leader for a service is an influential service user that advocates and strongly prefers a given service. The

"Follow the Leader"² model is probabilistic and we have provided a URL to a paper that describes it in detail for those who are not familiar with it.

Goldbaum's [3] model identifies three types of consumers seek input from outside experts:

1. The consumer has a fixed set of preferences but imperfect information concerning the available product options (Web Services). The expert offers information or advice that helps the consumer buy (select) the product that maximizes his exogenous utility.
2. The consumer possesses some innate preferences over the available products, but can be influenced by the opinion of others (peers) and experts. The "expert" may be someone possessing both better information than the general public about the consumer options (as in case 1), but also someone who can provide advice that is consistent with the underlying preferences of a group of consumers.
3. The consumer has no innate preferences. The consumer's tastes are fully fashioned by the influence of peers and experts. In this case, the expert shapes opinion, but need not have any special advantage in evaluating the options.

The expert in all previous cases can be considered as an experienced customer who spends time and effort in analyzing Web Service features, these include their functional and non-functional properties and finally making his decision in using the service, and giving good feedback on the service. Furthermore, he is in a position to strongly recommend the service to be used by his friends if they have similar needs.

Consequently, we can consider each member as part of a social network such as a customer \ user, and each member has a set of friends, such that *MEMBER.Friends* $\leq n$, where $n = \{0, 1, 2, 3, \dots, n\}$. According to [3], a member friend is either a leader or a follower who adopted another leader opinion or recommendation to use a Web Service, then this member adopts whatever his best friend adopted, otherwise the member has no active friends and consequently it acts as an independent (leader).

Ramirez-Cano and Pitt [17], define the relationship between two agents (users) as a confidence function, such that: "an agent (i) increases its confidence in another agent (j) based on how well (j's) opinion meets the criteria specified in i's mind-set. A mind-set represents the set of beliefs, attitudes, assumptions and tendencies that predetermine the way an agent evaluates a received opinion". Subsequently we conclude that user (agent) preferences in selecting a web service are the determinant relationship (factor) between two friends.

² www.business.uts.edu.au/finance/research/wpapers/wp155.pdf

Confidence Relation:

In a social network, agents are considered as nodes on a graph. Ramirez-Cano and Pitt [17] define the *confidence relation* between agents as the vertices of a social network. Each agent has a relationship (friendship) with a specific number of agents. The agent and his friends is a subset (m) of the social network. Each agent (i) has a confidence level with each of its friends ($m - 1$) and itself, confidence relations can be expressed as: $W_i(t) = \{w1, w2, \dots, w_i, \dots, w_j, \dots, w_m\}$, where $w_i(j)$ is the confidence of agent (i) in agent (j), and $0 \leq w_i(j) \leq 1$, means confidence relations are normalized such that: $\sum_{j=1}^m w_i(j, t) = 1$. When ($i=j$) then this refers to the agent self-confidence. If agent confidence in others = 0, this means the agent is a (leader). If the confidence in itself is greater than the confidence in any other friend then it acts as (independent), otherwise the agent acts as a follower (this means its confidence with any of other friends > self-confidence).

Opinion formation occurs at discrete time points. At each point in time, each agent exchanges opinions with other friends. An agent (i 's) opinion changes at time ($t+1$) by weighting each received opinion at time (t) with the confidence in that friend including its own opinion weighted by its self confidence, such that:

$$o_i(t+1) = \sum_{j=1}^m w_i(j, t) o_j(t) \quad (1)$$

The matching between agent (i 's) opinion with another agent's mind-set is defined as an affinity function (a_i) (similarity measure). This function evaluates the linear similarity between an opinion and a given constant (μ), which is a representative reference value of an agent's mind-set for a given issue. Affinity function is defined as:

$$a_i(j, t) = 1 - \frac{|o_j(t) - \mu_i|}{\max |o - \mu_i|} \quad (2)$$

$$\text{Where } \max |o - \mu_i| = \begin{cases} 1 - \mu_i & \text{if } 0 \leq \mu_i < 0.5 \\ \mu_i & \text{if } 0.5 \leq \mu_i \leq 1 \end{cases} \quad (3)$$

(μ) is a constant differ for each agent and constant in time. Thus the affinity function differs for each agent and any of its friends and corresponding to the fitness between opinions and mind-sets.

Therefore, the confidence changes over time based on the affinity between agent and its friends. Agents increase the confidence in those friends whose opinions fit their mind-set. From [17] definition, we can conclude that each *MEMBER* has a set of preferences stored in its mind-set and can be captured through the user interface. If *MEMBER(i)* has a set of preferences *PREF(i)* in a specific domain/context, and *MEMBER(j)* has a set of preferences *PREF(j)* in the same domain/context, and if *PREF(i)* is a

subset of *PREF(j)*, then we can conclude that *MEMBER(i)* is *Follower* to *MEMBER(j)*. A DL reasoner can check whether two concepts subsume each other. From social networks users profiles we can synthesis and classify the user either as a leader or as follower.

As a conclusion, current service discovery and selection is static and inflexible. Moreover, capturing and identifying user preferences and reusability of the best services are challenging issues. These challenges promote us to propose a framework to elicit user preferences from experts to be used in dynamic service selection and composition.

4 Market Leadership Selection Framework

It is a challenge to locate and select the best Web Service, we call it as (Market Leader), in a timely fashion that satisfies the given request when there are a large number of services available. In this paper we combine semantic, recommendation and social network techniques to capture user preferences and matching.

In this section we present our proposed ontology based Market Leadership Selection Framework (see Figure 1, 2 and 3) that address some of the key issues we identified in Section 1. The MLS Framework is composed of several modules and their roles are described in depth in the following subsections.

4.1 Ontologies and OWL-S

Ontologies are explicit shared specifications of conceptualization in a problem domain. They contain commonly agreed knowledge structures, i.e. domain concepts and the relations between them, and also shared terminology for describing these knowledge structures. The domain Ontologies represent concepts related to domain in which the Web Services operate. Domain Ontologies provide the context in which metadata can be interpreted by both humans and machines, whereas the resource Ontology [8] provides a conceptual model for describing resources by which semantic resource descriptions can be generated.

The resource ontology [8] is based on OWL-S upper ontology that partitions a semantic description of a Web Service into three components: the Service Profile, Process Model and Grounding. The Service Profile describes what a service does by specifying its inputs, outputs, preconditions, effects and other properties. The Process Model describes how a service works; each service is either an Atomic Process that is executed directly or a Composite Process that is a combination of other sub-processes. The Grounding contains the details of how an agent can access a service by specifying the details of the communication protocol, i.e. the parameters to be used in the protocol and the serialization techniques to be employed for the communication. OWL-S allows the definition of classes of related services and can establish links to other concepts that describe specific service types and their properties. This makes service

discovery much easier in terms of the built-in links, thus facilitating resource reuse.

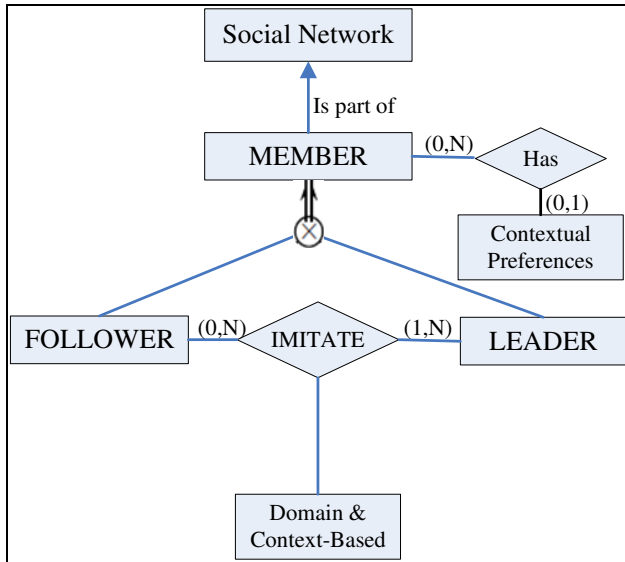


Figure 1: Follower-Leader Ontology

4.2 Follower-Leader Ontology

The MLS Framework uses the Follower-Leader Ontology as shown in Figure 1, where each user can be classified according to their behavior as either a follower or a leader. Within a specific domain and context each follower imitates one or more leaders based on innate (personal) preferences. Some followers don't have such preferences because simply they do not know such preferences prior to the Web Service selection process. Such innate (personal) preferences play a constraint on selecting their leader and consequently on selecting the Web Service. So, followers who have some preferences finally chose their leader on the basis of matching their preferences with the perceived preferences exhibited by the leaders. Each leader has specific preferences within a domain and context. Searching social networks for a specific member, if the member exists then inducing the social networks on the web each member identity (follower/leader) is identified, if no friend is found then member is considered to be an independent leader. If a follower is recognized then its friends can be discovered, all these friends are retrieved to find the best match within that domain and context. If none of those friends have preferences in that domain\context, then an iterative searching process to be implemented on friends of available friends until there are no more to search for or a friend of a friend has preferences that have been recognized.

4.3 User Preferences Ontology

The MLS Framework also uses the User Preferences Ontology as shown in Figure-2, user preferences can be

classified according to the main category (Functional \ Non-Functional), each preference has a set of attributes such as: Name, Description, Category, Scope: (local \ Global), Unit, Operator (=, >, <), Value and Importance-Level (0 – 10).

For example: in a financial domain, we consider four context examples (Home-loan, Shares, Home-Insurance, and Life-Insurance).

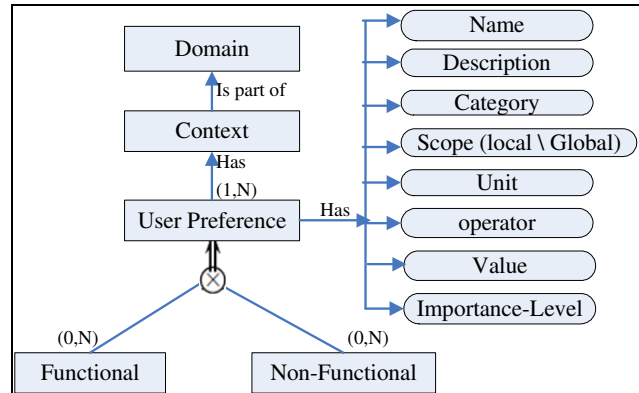


Figure-2: User Preferences Ontology

For Home-loan, we can consider the following preferences (1) Functional Preferences such as: Term, Interest rate, Redraw Facility, Establishment fees, reoccurring fees \ frequency...etc (2) Non-Functional Preferences such as Trust, reputation, Security, Reliability, response time ... etc.

4.4 The MLS Framework Overview

A user in this context refers to any person who directly interacts with the system, or an agent that interacts on user behalf. All agents can be classified as leaders. Follow the Leader Architecture as depicted in Figure-3 is composed of the following modules:

USER MODULE: communicates with user to capture user query, user identity, identify domain and context then pass (query, identity and domain \ context information) to leader module. This module provides also, user preferences negotiation, result presentation and feedback absorption.

LEADER INDUCER MODULE: provides facility to synthesize about a leader for a follower or unknown user identity as described in Follower-Leader Ontology. It utilizes Social Networks Analysis (SNA) approaches [7], to perform analysis on social networks.**LEADER MODULE:** this module acts as a recommender system, identifies user style (Follower, Leader) from its database and from its communication with **LEADER INDUCER MODULE**. For a leader, the Leader module passes the query with preferences to Domain \ Context Preferences Module. For a follower, the leader module, find the best match service or at least the best match preferences to the

query from its DB. This module used also to store leader details, preferences, query, selected service (URI), and feedback rating. This module acts as the heart of the recommender system.

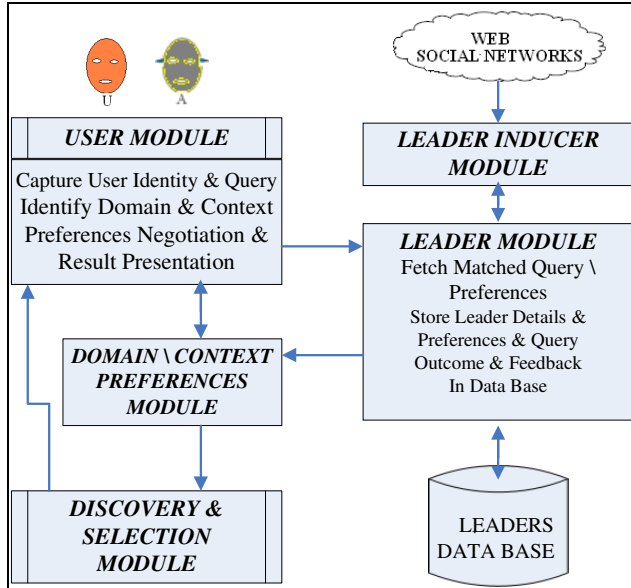


Figure-3: Follow the Leader Architecture

DOMAIN \ CONTEXT PREFERENCES MODULE: this module maintains a database for optimal metrics for preferences associated with each context in specific domains. It used to configure preferences, which are to be used in Discovery & Selection Module. In early stages of implementation neither leaders nor PREFERENCES MODULE have any knowledge about preferences, so at this stage preferences are drawn from Service Profile and non-functional properties including QoS ontologies.

DISCOVERY & SELECTION MODULE: this module performs two functionalities: (1) match-making of functional requirements specified in user requirements with specifications of existing services, and (2) selection, i.e., choosing a service with the best quality among those able to satisfy a user's preferences. For Web Service Discovery & Selection this framework can adopt any semantic based approach utilizing Web Service Ontologies, semantic metadata and a DL-based reasoning engine; such as the one outlined in [8], "Semantic-Enabled Resource Discovery".

5. Service Selection Approach:

Based on Market Leadership discussed in section (3) and 4.4 The MLS Framework, user can be semantically recognised either as a leader or a follower. Leaders are capable of providing prosperous functional and non-

functional selection criteria. A discovery and matching process has been conducted over published services advertisements and all functional selection criteria has been met, the outcome from this process returned the set (S_i) of matched services considered for QoS selection process.

5.1 Notations and Definitions:

In the following section we use the following notations:

P = Published capabilities, R = Request(query), M = Matched, U = User, S = Atomic Service

User Query (Request):

Let U_R = Set of user attributes captured from user query (request)

User query is defined as a tuple $\langle U_F, U_N \rangle$, U_F represents user functional requirements to be all satisfied (compulsory), where as U_N , represents non-functional (QoS) requirements. $U_R = U_F \cup U_N$

Let (m) be the number of non-functional (QoS) attributes that considered as constraints over the selection process, then U_N can be defined as:

$$U_N = q_r^m : (q_r^1, q_r^2, \dots, q_r^j \dots q_r^m) \quad (1 \leq j \leq m),$$

$$q_r^j = \{q_N^j, q_O^j, q_V^j, q_W^j\} \text{ where :}$$

N is QoS attribute Name,

O refers to the operator constraint (\leq, \geq)

V refers to the value constraint

W refers to the weight constraint

For example user might search for: (home loan interest rate $\leq 5.5\%$, with weight 10)

Each user (QoS) attribute in his query associated with a priority (weight) over other attributes, such that:

$$W = w_r^m : (w_1, w_2, \dots, w_j \dots w_m) \quad (1 \leq j \leq m)$$

So, each (QoS) attribute is either to be maximized or minimized, based on the operator constraint, for example the user is looking for the highest interest rate on his deposit while he is looking for the minimum ticket cost.

Functional Match Services:

Let S_P = Set of published services by service provider

Let S_M = Set of matched services considered for QoS selection process after the discovery process has been conducted, then S_M is the set of services that satisfied the functional requirements constraints, such that:

$$S_M \subseteq S_P, S_M = U_F \cap S_P$$

$S_M = \{S_M^n : (S_M^1, S_M^2, \dots, S_M^n)\}$ each item represents an atomic service, these atomic services are common in their functionality.

Each atomic service have a set of QoS attributes mapped to user non-functional (QoS) attributes, such that:

$$S_i = q_i^m : (q_i^1, q_i^2, \dots, q_i^j, \dots, q_i^m), (1 \leq j \leq m)$$

Or can be represented as:

$$S_i = q_{im} : (q_{i1}, q_{i2}, \dots, q_{ij}, \dots, q_{im}), (1 \leq j \leq m)$$

Although some atomic services satisfy functional requirements and some QoS attributes are not available, we use the worst offered value for non-available attributes. Since the value of a missing QoS attribute can be predicted by using either average values (q_{avg}) or the worst-case values (q_{max}) or (q_{min}) [18]. Using (q_{avg}) will likely select the most suitable services since it reflects the actual service behavior more accurately, while using (q_{max}) or (q_{min}) will select services with a stronger guarantee.

5.2 Selection Algorithm:

1. For each non-available QoS attribute (q_i^j) value in service S_i where ($1 \leq j \leq m$), use the worst offered value from other services.
2. Normalize weights (priorities) given by user, such that:
$$\sum_{j=1}^m w_j = 1$$
3. Compute utility function for each atomic service.
4. Select the atomic service with the highest utility function as the best candidate. If two or more have the same utility value, then select randomly from them. Since our proposed system maintains feedback and reputation, then select the service with higher reputation as the best candidate from equal utility.

Utility Function:

The utility function U_i from using service S_i as adopted from [18], is defined as:

$$U_i = \sum_{i=1}^m w^j * A_i^j \quad (4a) \quad (1 \leq j \leq m) \text{ where,}$$

$$A_i^j = \begin{cases} \frac{q_i^j - \mu^j}{\sigma^j} & \text{if Maximize, } op = (\geq) \\ 1 - \frac{q_i^j - \mu^j}{\sigma^j} & \text{if Minimize, } op = (\leq) \end{cases} \quad (4b)$$

w^j is the weight for the (jth) attribute such that,

$$0 < w^j < 1,$$

weights are normalized such that: $\sum_{j=1}^m w^j = 1$.

μ^j, σ^j Are the average and standard deviation of the (jth) attribute for all candidates.

If $\sigma^j = 0$ of any attribute, then set $\sigma^j = 1$ to avoid division by zero, this case is very likely to happen if all services offered the same value for that attribute.

5.3 Service Capability Measure:

In our simulation model, we analyzed the behavior of the service with the highest attraction, we call it "Market Leader". Our analysis shows that the market leader has the following special characteristics:

1. Availability of all QoS attributes in the published service.
2. Most of QoS attributes values offered are the best over all other candidates.

Accordingly, we introduce a relative measure of each service capabilities with respect to other candidates, assuming all attributes with equal weight. Synthesize the attributes to be maximized and those to be minimized, *capability measure* is defined as follows:

$$C_i = \sum_{i=1}^m w_p^j * B_i^j \quad (1 \leq j \leq m) \text{ where,}$$

$$B_i^j = \begin{cases} \frac{q_i^j - Min^j}{Max^j - Min^j}, \dots \text{if Maximize, } op = (\geq) \\ \frac{Max^j - q_i^j}{Max^j - Min^j}, \dots \text{if Minimize, } op = (\leq) \end{cases} \quad (5)$$

Where w_p^j is the predicted weight for the (jth) attribute such that, $0 < w_p^j < 1$,

Weights are normalized such that: $\sum_{j=1}^m w_p^j = 1$.

Weights can be predicted based on the experience of the recommender system with the user queries by computing average weight of each attribute over a specific period of time. Initially the recommender system initializes all weight values with equal values (10). Although this prediction is based on different users behaviors, it still gives a better indication than equal weights given to all attributes.

Max^j, Min^j are the max and min of the (jth) attribute

for all service candidates. If $Max^j - Min^j = 0$, then set

$Max^j - Min^j = 1$ to avoid division by zero, this case is

very likely to happen if all services offered the same value for that attribute.

6. Simulation Results and Analysis:

We have developed a Service Selection Simulation tool utilizing the NetLogo platform [19]. The user interface as shown in figure-4, and we use it to analyze and evaluate the validity of our approach, in the following section we outline the testing environment and associated outcomes.

6.1 Simulation model:

Our model as shown in figure-5 composed of a fixed number of atomic services (30) with the same functional properties and varied in their (QoS) attributes, each atomic service maintains the following information: (service code, a list of QoS attribute codes and corresponding values). Each service' information is hidden from others. That information is static during any simulation session.

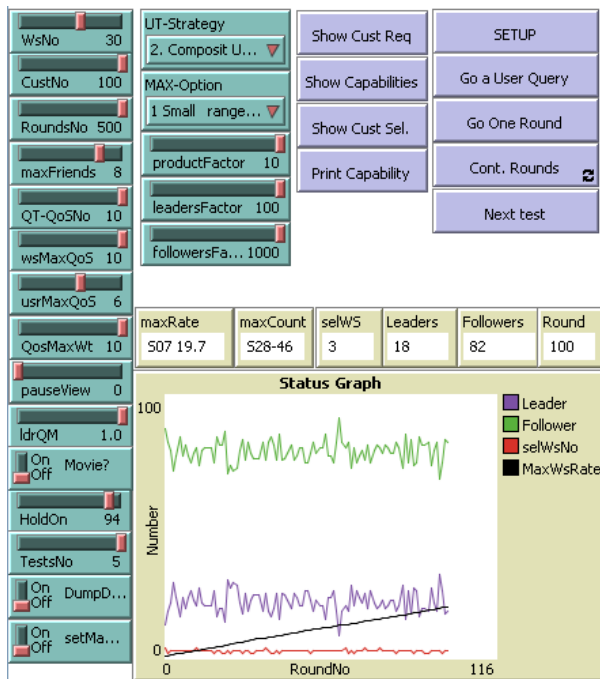


Figure-4: Service Selection Simulation Interface

Each simulation session is composed of a fixed set of rounds (500). In each round a number of customers (100) enter their queries to the system. Each customer has the following information: (ID, a list of attribute QoS codes and corresponding values and weights –priorities- and a set of variable number of friends).

The system evaluates customers based on their query and friends information, if the customer has no friends and has number of QoS attributes greater than a predefined threshold value, then the customer qualified as a “leader” otherwise it qualified as a “follower”. So, a follower has at least one friend. As well, the system

maintains a rating algorithm for services based on their number of QoS attributes provided by each service and customers’ feedback. All services have the same rate in the beginning of each session.

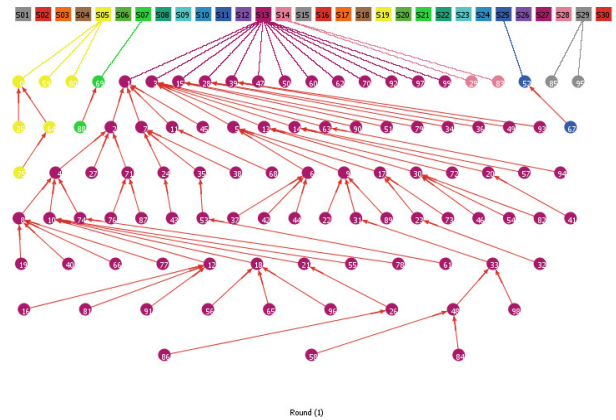


Figure-5: Sample view of the model

Each simulation session starts by setting the services and customers with their corresponding information. Each round starts with a new set of customers with new query and friends information. No communication between customers is assumed even if they are friends. In each round every customer passes its information to the system, if the customer is a qualified leader, then the system performs the service selection process, selecting the best service from available services according to the selection process mentioned previously. If the customer is a follower, then the system selects a friend from customer friends to follow. For simplicity considerations, current friends selection is random from available friends information. However in real life scenario, friends can be discovered using social network analysis tools. Friend relationships could be captured directly on the basis of Confidence Relation proposed in section (3). Each customer provides feedback to the system about his satisfaction, and this feedback can be used to influence the evolution of the underlying network connections, **furthermore it can also be used to drive machine learning algorithms which could help to improve the user’s service selection performance**

By the end of each transaction, the system updates its registry for each service with its new rate. Service rate influenced by three factors (1) Number of QoS attributes that are offered, (2) Number of Leaders selecting that service and (3) Number of followers who follow a leader selected that service. Each factor has a corresponding coefficient factor reflects its importance. In our model all services are given (1) for the first factor, while factor (2) = number of leaders / 100, and factor (3) = number of followers / 1000.

6.2 Simulation results:

6.2.1 Market Leader Analysis:

Market leader is one of the services with high capability. It shows high attraction ratio in the first couple of rounds as shown in figure-5 and table-1 it refers to (S07), and continues attracting leader customers and their followers. It is not necessary to be the leader in each round. For example in rounds (5, 12, 14, 15, 21) other competitors lead the round. From table-1, we note four competitors (S05 S03 S07 S16) with the highest capability measure among the (30) providers. From multi-test results market leader shows attraction ratio of about (40%) of all transactions, while other competitors share the rest based on their capability measure.

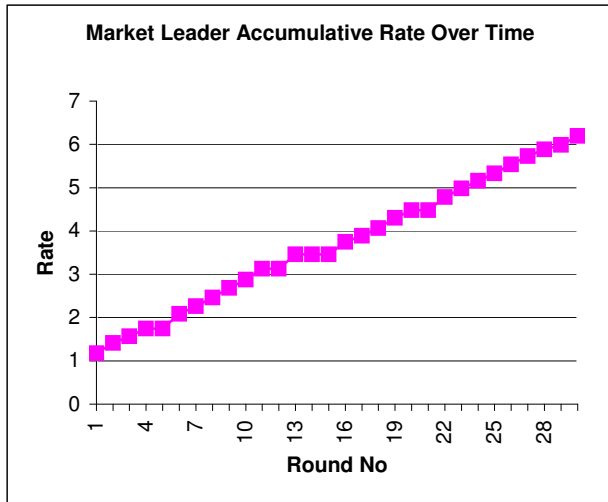


Figure-5: Market Leader (S07) Accumulative Rate Over Time

6.2.2 Customer Behavior Analysis:

From the simulation settings, we note that the number of customers who are qualified as “Leaders” varies in each round, with a maximum of (36) and a minimum of (10). On average (21) customers are qualified as leaders each round, which concludes that the model promotes adequate number of leaders’ opinions to establish the recommender system knowledge base.

7 Evaluation and Discussion

In evaluating our framework, we are interested in two measures. The first measure was to determine the ratio of users who act as leaders or followers to overall population within a specific period of time and the second measure being the impact of Available services on Followers and Leaders selection.

Round No	Market Leader(S07) Rate	Round Leader	Round Leader Rate	Best Services among 30
1	1.17		1.17	S05 S03 S07
4	1.74		1.74	S05 S03 S07
5	1.74	S05	1.15	S03 S05 S07
6	2.08		2.08	S05 S07 S03
11	3.13		3.13	S03 S07 S05
12	3.13	S03	1.78	S07 S03 S05
13	3.46		3.46	S07 S05 S03
14	3.46	S03	1.97	S03 S07 S16
15	3.46	S03	2.08	S03 S05 S07
16	3.75		3.75	S05 S03 S07
17	3.89		3.89	S03 S07 S05
19	4.30		4.30	S03 S07
20	4.48		4.48	S07 S03 S05
21	4.48	S03	2.52	S05 S07 S03
22	4.78		4.78	S05 S03 S07
30	6.20		6.20	S16 S03 S07

Table-1: Market Leader (S07) Accumulative Rate Over Time

1. Number of Leader customers varies in each round, with an average of (21%) of the population being qualified leaders, which demonstrates that the model promotes adequate number of leaders with high self confidence to establish the recommender system knowledge base.
2. Based on the previous conclusion, our recommender system can support around (79%) of users who act as followers, providing them with high quality recommendations. This conclusion highly supports the usability of recommender system.
3. Followers follow other friends with high confidence. This concept supports the usability of the recommender system to recommend similar services for similar customers within a specific domain and context.
4. Market leader service is a service with high capability and leaders can achieve the highest utility from its usage. It shows high attraction ratio among all other candidates, this verify the usage of our selection algorithm.
5. If the user does not have any friends, then such a user acts as an independent leader, who needs to explore all available preferences in such domain/context. The recommender system provides him with facilities to build his own preferences.

To handle the “cold start” issue in other recommendation systems, our frame work proposes an evolutionary approach to use a *PREFERENCES MODULE* to build a preferences database from Service Profile and non-functional properties including QoS ontologies. Hence, a Domain\Context preferences database is built through evolutionary usage of the system.

In order to create an efficient user personalized service, our framework recommends the best service based on user preferences in a specific Domain\context. This improves user query performance to discover and select best and frequent Web service, and provides the highest accuracy in its recommendation for the current user.

So, we can conclude that our framework provides an effective approach to capture user preferences within a specific domain and context, manages the “cold start” issue and provides an efficient user personalized service.

8 Conclusion

Locating and selecting the best Web Service, that satisfies a given service request from a large number of available services is a conceptually and computationally challenging problem. The MLS Framework demonstrates that combining semantic web, social networking and recommender system techniques provides a powerful and effective solution to the web service selection problem.

Our MLS Framework for Semantic Web Service selection is based on functional and non-functional properties. Functional properties are bounded by the resource ontology described in section 4.1. To differentiate between non-functional properties, we propose a domain \ context preferences ontology to capture user preferences semantically. The MLS Framework captures user’ preferences based on a novel “Follow the Leader” model that utilizes social networks structures and user behaviors, is a friendly and expressive approach used to capture large quantities of preferences and the logical relations between preferences and corresponding level of importance.

In future work, we intend to extend the MLS Framework by implementing *confidence relation* described in section 3 to model richer relationships and to enhance model performance.

References

1. Martin, D., et al., *Bringing Semantics to Web Services with OWL-S*. World Wide Web, 2007. **10**(3): p. 243-277.
2. Vitvar, T., et al., *X- Semantically-enabled service oriented architecture: concepts, technology and application*. Service Oriented Computing and Applications, 2007. **1**(2): p. 129-154.
3. Goldbaum, D., *Follow the Leader: Simulations on a Dynamic Social Network*. 2008. **UTS Finance and Economics Working Paper No 155**, <http://www.business.uts.edu.au/finance/research/wpapers/wp155.pdf>.
4. Maximilien, E.M. and M.P. Singh, *A framework and ontology for dynamic Web services selection*. Internet Computing, IEEE, 2004. **8**(5): p. 84-93.
5. Liu, Y., A.H. Ngu, and L.Z. Zeng. *QoS computation and policing in dynamic web service selection*. 2004: ACM New York, NY, USA.
6. Maximilien, E.M. and M.P. Singh. *Toward autonomic web services trust and selection*. 2004: ACM New York, NY, USA.
7. Erétéo, G., et al., *A State of the Art on Social Network Analysis and its Applications on a Semantic Web*. 2008.
8. Chen, L. and F. Tao, *An Intelligent Recommender System for Web Resource Discovery and Selection*. Intelligent Decision and Policy Making Support Systems, 2008.
9. Manikrao, U.S. and T.V. Prabhakar. *Dynamic Selection of Web Services with Recommendation System*. 2005.
10. Kokash, N., A. Birukou, and V. D Andrea, *Web Service Discovery Based on Past User Experience*. LECTURE NOTES IN COMPUTER SCIENCE, 2007. **4439**: p. 95.
11. Drachsler, H., H. Hummel, and R. Koper, *Personal recommender systems for learners in lifelong learning: requirements, techniques and model*. 2007.
12. Birukou, A., et al. *IC-service: a service-oriented approach to the development of recommendation systems*. 2007: ACM Press New York, NY, USA.
13. Kritikopoulos, A., M. Sideri, and I. Varlamis. *Success Index: Measuring the efficiency of search engines using implicit user feedback*. 2007: PCI.
14. Kim, J. and E. Lee, *Semantic Web Recommender System Based Personalization Service for User XQuery Pattern*. LECTURE NOTES IN COMPUTER SCIENCE, 2005. **3828**: p. 848.
15. Scherp, A. and R. Jain, *Towards an ecosystem for semantics*. 2007.
16. Mori, J., Y. Matsuo, and M. Ishizuka. *Finding user semantics on the web using word co-occurrence information*. 2005.
17. Ramirez-Cano, D. and J. Pitt. *Follow the Leader: Profiling Agents in an Opinion Formation Model of Dynamic Confidence and Individual Mind-Sets*. 2006: IEEE Computer Society Washington, DC, USA.
18. Yu, T., Y. Zhang, and K.J. Lin, *Efficient algorithms for Web services selection with end-to-end QoS constraints*. 2007.
19. NetLogo, *NetLogo Home Page*. On line at: <http://ccl.northwestern.edu/netlogo/>. 2009.

© [2009] IEEE. Reprinted, with permission, from [Al-Sharawneh, J.; Williams, M.-A. A social network approach in Semantic Web Services Selection using Follow the Leader behaviour. Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009]. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it