Towards Soft Computing for Service Discovery

Dexin Zhao1, Zhiyong Feng1, Shizhan Chen1, Qing Yu2 and Wenjie Li2
1School of Computer Science and Technology, Tianjin University, Tianjin, 300072, China
2School of Computer Science and Technology, Tianjin University of Technology, Tianjin, 300191, China
Email: qiqiharxin@163.com

Abstract—Current existing service discovery mechanism is lack of dealing with imprecise or vague information on semantic level. This paper proposes a novel approach that service advertisements or requests can be expressed in natural language involving linguistic variables, which is considered as an attempt to achieve the vision of soft computing. This new service discovery method has three features by comparing with existing architectures: First, service description is on semantic level, especially allows users to express their requests involving imprecise or vague data in a natural way. Second, the approach is based on the theory of fuzzy logic, which is the principal constituent of soft computing. Third, the architecture has ontology support which makes the service descriptions be understandable by all the participants.

Index Terms—Service discovery, Soft Computing, Fuzzy logic, Linguistic Variable.

I. INTRODUCTION

With the enormously increasing number of services that become available in internet, service discovery has been a well-recognized challenge in distributed environments [1]. Service discovery protocols are designed to minimize administrative overhead and increase usability. They can also save system designers from trying to foresee and code all possible interactions and states among devices and programs at design time. By adding a layer of indirection, service discovery protocols simplify pervasive system design [2].

Recent trends in mobile and ubiquitous computing have created new requirements for automatic configuration of network resources. In responds to these requirements, a variety of service discovery protocols have been proposed, which attempt to provide automatic “discovery” and configuration of network services. Examples include Sun Microsystems’ Jini Network Technology [3], Microsoft’s Universal Plug and Play [4], and Internet Engineering Task Force’s Service Location Protocol [5]. Currently most these existing service discovery techniques are on syntactic level, which are lacking of rich semantic representation, especially lacking the ability for representing imprecise or vague information of services. For the ubiquitous computing environments to become practical, an infrastructure needs to be supported, which allows users to discover, deploy, compose, and synthesize services automatically. Such an infrastructure must be semantic-based so that applications can reason about a service’s capability to a level of detail. The increasing complexity and distributed nature in service discovery are prompting us to investigate how to represent imprecise information of services that can be understood by a computing device, and how to make the users feel natural and easy to express their requests. The efforts of our proposed approach on dealing with uncertainty information are coincident with the idea of soft computing, which is the state-of-the-art approach to artificial intelligence. Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation [6].

The goal of this paper is to increase the efficiency of the service discovery and to allow imprecise or vague terms in the process. The proposed approach exploits the theory of fuzzy logic as a description and reasoning methodology to present fuzzy information. Generally, computing involves manipulation of numbers and symbols. By contrast, in daily life, we take full advantage of the information expressed in a natural language, employ mostly words in computing and reasoning, and obtain conclusions expressed as words from premises. In our service discovery framework, a word is viewed as a label of a fuzzy set of points, which drawn together by similarity, and the fuzzy set plays the role of a fuzzy constraint on a linguistic variable.

The rest of the paper is structured as follows. Section 2 introduces the basic notions of fuzzy logic and soft computing. The emphasis is on the two important concepts in fuzzy logic: linguistic variable and generalized constraint, which form the ideas about our representation of fuzzy data. Section 3 presents our solution to the service discovery, which has the ability to process the fuzzy information in a natural language. Finally, Section 4 gives the concluding remarks of our work.
II. FUZZY LOGIC AND SOFT COMPUTING

The role model for soft computing is the human mind. The basic ideas underlying soft computing in its current incarnation have links to many earlier influences, among them professor Zadeh has made a significant contribution on the research about fuzzy set theory [7]. Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost. The principal constituents are fuzzy logic, neurocomputing and probabilistic reasoning.

As one of the principal constituents in soft computing, fuzzy logic provides a methodology to express imprecise information. In fact, fuzzy information plays a pivotal role in humans’ cognition, which serves a way of achieving data compression. A familiar instance to illustrate the point is the problem of parking a car. In our daily life, it is an easy task to park a car because the final position of the car is not specified precisely. The difficulty of parking will increase geometrically with the increase in precision, and eventually the task becomes impossible to accomplish. In fuzzy logic, confidence values increase in precision, and eventually the task becomes impossible to accomplish.

A linguistic variable, and its linguistic values are membership function interpreted as a label of a fuzzy set which is characterized by a vocabulary of English.

If the universe of discourse is $U=[0,150000]$ and $x$ is a numerical vocabulary such as 8000, then $\mu_{large}(8000)$ is the grade of membership of 8000 in the above membership function. In addition, $\mu_{large}(x)$ can be interpreted as the degree to which $x$ fits the perception in a specified service context.

We divide linguistic values into two basic types according to their linguistic structures. One is consisted by primary terms, such as old, young, another type of linguistic value is compound terms consisted by primary words with some modifiers according to syntax rules, such as very young, quite comfortable etc. Other terms can be viewed as composite words, which may be generated through the combination of the above basic types of linguistic values. In fact, Primary terms and their derivatives are fuzzy subset of the domain, so the factual meanings of the linguistic values are defined by relevant membership functions. That is, a linguistic value can be derived from a fuzzy set $A$ which is defined as

$$A = \{ (x, \mu_A(x)) \mid x \in X \}$$

where $\mu_A(x)$ is the membership function of $x$ and $X$ is a collection of objects denoted by $x$. $\mu_A(x)$ maps $x$ to the membership space denoted as $M \in [0,1]$.

The modifiers, such as very, extreme, little, etc. are often used in service description or consumers’ requests to express fuzzy data, they are linguistic modifiers which modify the meaning of the primary terms. On account of that, it is necessary to give the membership functions of the modified operators. The constraint modification rules that we proposed are as below:

**Definition 1.** If $A$ is a primary term, and its membership function is $\mu(x)$, then

- extremely $A = \text{op}_1(A)$, where $\mu_{\text{op}_1(A)} = [\mu(x)]^3$
- very $A = \text{op}_2(A)$, where $\mu_{\text{op}_2(A)} = [\mu(x)]^2$
- more or less $A = \text{op}_3(A)$, where $\mu_{\text{op}_3(A)} = [\mu(x)]^{3/2}$
- roughly $A = \text{op}_4(A)$, where $\mu_{\text{op}_4(A)} = [\mu(x)]^{1/4}$
- not $A = \text{op}_5(A)$, where $\mu_{\text{op}_5(A)} = 1 - \mu(x)$

where $\text{op}_1, \text{op}_2, \text{op}_3, \text{op}_4$ and $\text{op}_5$ stand for operators. Representing linguistic terms by means of fuzzy sets, this corresponds to: extremely $A \subseteq$ very $A \subseteq$ more or less $A \subseteq$ roughly $A$. It is worth noting that these rules are commonly used in service description and may be adjusted with the special context of the service.

B. Generalized Constraint

In service querying and advertising, it is usually the case that what is expressed is not exact an instance of a service, but rather a fuzzy description of the acceptable instances. To describe this problem naturally, an efficient way is expressing constraints over the linguistic variables of the service. So, we induct the concept of the generalized constraint from fuzzy logic to our description of services. In our method, information

![Membership function for large vocabulary](image_url)
is assumed to consist of a collection of propositions expressed in a natural or synthetic language including linguistic variables, and information is conveyed by constraining the values of linguistic variables.

Let \( X \) be a variable which takes values in a universe of discourse \( U \). Generally, a generalized constraint on the values of \( X \) is represented as

\[
p \rightarrow X \text{is} a \ R
\]

where \( \text{is} a \) is a variable copula which defines the way in which \( R \) constrains \( X \). More specifically, the role of \( R \) in relation to \( X \) is defined by the value of the discrete variable \( r \). The principal types of constraints and the values of \( r \), which define them, are the following [8]:

- \( e \): equal (abbreviated to \( = \));
- \( d \): disjunctive (possibilistic) (abbreviated to \( \epsilon \));
- \( c \): conjunctive;
- \( p \): probabilistic ...

In addition to the types of constraints defined above, there are some more specialized and less commonly used constraint types. As an illustration of the generalized constraints, when \( r = e \), the constraint is an equality constraint and is abbreviated to \( = \). In this case, \( X \text{eq} a \) means that \( X = a \). When \( r = d \), the constraint was disjunctive, and “\( \epsilon \)” is abbreviated to “\( \epsilon \)”, which leads to the expression \( X \text{is} R \). Constraints induced by propositions expressed in a natural language are for the most part possibilistic in nature. This is the reason why the simplest value, \( “r = \epsilon” \), is chosen to define possibilistic constraints.

A conclusion can be got from the concept of the generalized constraint, that is, information may be viewed as the constraints on linguistic variables. For example, the proposition “Cathy is young”, conveys information about Cathy's age by constraining the values that the variable \( \text{Age} \) (Cathy) can be taken.

### III. FL-BASED SERVICE DISCOVERY APPROACH

In this section, we present our solution to the service discovery based on FL (fuzzy logic), which allows users to express their queries with fuzzy information in a natural language. To support rich representation of service, the service description is critical to service discovery application. Unfortunately, a number of service discovery protocols do not take into account the importance of service description and do not address it fully [9]. In our approach, we adopt fuzzy logic to address the representation and inference on imprecise information.

Another feature of our approach is the ontology support to the process of service discovery. An ontology is not only the simple specification of a set of terms, but also expresses relationships between them. It is an important aspect that will facilitate the interoperability between clients and services across various parts in the environments. While, common ontology infrastructures are often either missing from or are not well represented in the existing service discovery architectures. There are existing many ontology efforts, and to design an ontology is beyond the scope of our work. During the development of the service discovery prototype, the terms that are expressed in the service are defined in OWL [10], and we only require descriptions to refer to the ontology in order to mediate between diverse information sources. OWL is a W3C recommendation for such a language and it is an extension to the RDF [11], which lets us create ontology for arbitrary domains and instantiate the ontology to describe resources. The supporting fuzzy rules and sets are stored as OWLJessKB files. OWLJessKB, a knowledge based tool, is able to interpret OWL syntax.

\[
p = \text{Today seems to be a very hot day}
\]

The proposition may be expressed as \( \text{Today has the probability (such as 0.6) to be a very hot day} \). According to the generalized constraint types mentioned in section 2, the proposition type is \( X \text{isp} R \).

Assume that the explanatory database is chosen to be \( ED = \text{POPULATION}[\text{Date};\text{Weather}] + \text{HOT}[\text{Weather};\mu] \), in which POPULATION is a relation with arguments Date and Weather, Hot is a relation with arguments Weather and \( \mu \) and \( + \) is the disjunction. The meaning of \( p \) is defined by two procedures on it. The first procedure acts on ED and returns the constrained variable \( X \), and the second procedure acts on ED and returns the constraining relation \( R \). In this case,
the constrained variable is the Weather, which in terms of ED, may be expressed as:

\[ X = \text{Weather(Today)} = \mu_{\text{Weather}} \text{POPULATION}[\text{Date} = \text{Today}] \]

This expression specifies the procedure which acts on ED and returns \( X \). More specifically, in this procedure, \( \text{Date} \) is instantiated to \( \text{Today} \) and the resulting relation is projected on \( \text{Weather} \), yielding the weather of today. The constraining relation \( R \) is given by \( R = \text{HOT}[\text{Weather}; \mu^2] \), which implies the intensifier very interpreted as definition 1.

A simple instance can be an illustration to fuzzy constraint propagation, in fuzzy logic, a Compositional Rule is:

\[
\begin{align*}
X & \text{ is } A \\
(Y, X) & \text{ is } R \\
Y & \text{ is } B
\end{align*}
\]

where \( A \) and \( R \) are fuzzy relations, \( B = A \circ R \) denotes the composition of \( A \) and \( R \). For example, if \( A = \text{"small"} \) and \( R = \text{"approximately equal"} \), then \( Y \) is small. The Compositional Rule is an important fuzzy constraint propagation rule, which is one of the inference rules governing fuzzy constraint modification.

IV. CONCLUSION

Due to the large number of services, finding, selecting, and using a service is a difficult task. Service discovery provides a link between what an application or user wishes to do, and it is essential to achieving Weiser’s vision of ubiquitous computing [12]. A significant amount of recent research has focused on effective discovery of services, but we found that the main shortcoming is that these service discovery protocols are intolerant of imprecision and partial truth due to the bivalent logic foundation. As a result, existing service discovery architectures do not have a capability to provide sufficient support to operate on perception-based information which is prevalent in the real world.

This work investigates the issue of automatic service discovery which is characterized by multiple requirements such as rich representation, constraint specification, ontology support, and the imprecise data expression. We exploit fuzzy logic, which involves a fusion of natural languages and computation with linguistic variables, to address the representation and inference on the imprecise or vague information.

Based on our proposed approach, the ability of higher expressive power, an enhanced ability to model real-world problems is obtained, which fits the guiding principle of soft computing. It is worth noting that, FL-based service discovery is not a replacement for conventional methods of numerically based computing. Rather, it is an additional tool which significantly enhances the ability of methods of design and analysis to deal with real world problems.

REFERENCES