ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Different Syntactic Methods and Clustering For Web Service Integration

Praveenkumar Arjun Patel, Prof. Umesh Laxman Kulkarni

PG Student, Dept. of CSE, DYPCET, Shivaji University, Kolhapur, India

Department of Computer Engineering, Mumbai University, Mumbai, India

ABSTRACT: Service Integration is the act of combining the resulting data of various complementary web services into a common scenario. Such an approach is assuring with respect to the discovery of new types of knowledge. Before service integration processes can be executed, it is necessary to predict which web services are required candidates. Knowledge Discovery in Services (KDS) process is specialized to address a repository of open services that do not contain semantic annotations. In these conditions, specialized methods are required to determine equivalences among open services with sensible precision. It commences a bottom-up process for KDS that adapts to the environment of services for which it operates.

KEYWORDS: Clustering, Equivalence Processing, TSM-LP, Web services

I. INTRODUCTION

Thousands of modular software capabilities discoverable and usable by disparate organization are supported by Service-Oriented Computing (SOC). Underlying SOC are specifications, such as the Web Service Description Language (WSDL), and the invocation methods supported by SOAP or REST that translate machine interfaces and interactions. Such open approaches provide an environment where web services are used.

Prototype of service-oriented computing is Web 2.0 that overlays the notion of it. Combination of fundamental model and Web 2.0 is advantages for anyone user to be the prime stakeholder. Due to collaboration technologies and market oriented environments user will interact faultlessly such as, any user may manipulate web services in a most appropriate style to their everyday activities. The integration of the resulting data into a common vision can be of important while business process composition is unnecessary.

Service integration is the parallel execution of two or more services to create an integrated data providing with a more complete description about some object or task.

II. LITERATURE SURVEY

Now a day's service integration receives a great deal of knowledge from academia and industry. Tools and techniques that instrument the service integration process and subsequently conceive of the results are involved in much of the current work. This part of work largely relates to the great research area of data and information integration. Knowledge Discovery in Databases (KDD) [1] is initial idea from which we considers knowledge discovery but instead of databases or data mining, we consider new knowledge that can be achieved when aggregating complementary services.

Knowledge Discovery in Services (KDS) [1] is a systematic process for discovering web service candidates for service integration that may ultimately find out new knowledge. Within this approach, there is a customized improvement life cycle that software engineers can use to generate new applications based on service integration techniques.KDS also find outs the aspects of the web service specifications that are most valuable for determining service integration qualification.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

This includes two main tests.
1. In circumstances where web service-based semantic characterizations are not available, high-accuracy syntactical approaches must be in put to suppose equivalences among services using direct and indirect information from service specifications (Equivalence Processing).
2. Features that make two of more services capable of integration or service integration (Clustering) must be well recognized and flexible as the environment of service repositories progress.
In this paper the process that intersects the areas of data integration and service mashup. This paradigm for discovering data from web service provisions is similar to the well-established approaches associated with KDD. This paper discusses the likenesses of the two processes. In addition, the state-of-the-art approaches related to the underlying KDS areas of equivalence processing, clustering, and filtering [1].

Although the area of data integration has had a longstanding background, the use of these techniques to accomplish mashups has only just recently started to be addressed. A majority of the work in this area addresses the tools and environments. That supports the visualization and presentation of mashup results [2], [3]. Other projects describe enabling techniques for preparing services for mashup [4]. There are also projects that investigate the policy for protecting data in mashup environments and instituting enterprise policy for modernizing systems using the information resulting from successful mashups [5].

III. RESEARCH METHOD

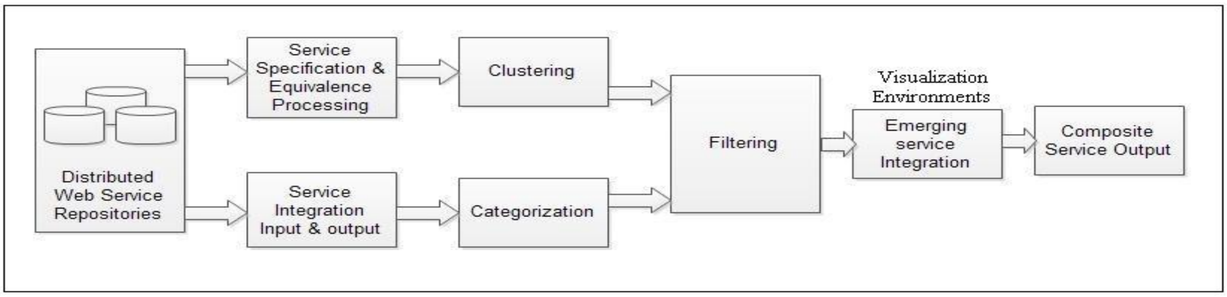

Distributed Web Service Repositories
Service Specification & Equivalence Processing
Clustering
Service Integration Input & output
Categorization
Filtering
Visualization Environments
Emerging service Integration
Composite Service Output

Fig.1 Proposed Web Service Integration Architecture

IV. EQUIVALENCE PROCESSING

In Conventional software engineering places the match of software interface for has explored. Our approach compares the similarity search of earlier approach by making use of markup language interface which is available within service oriented computing domain. Also the approaches to service integration are similar to the fundamental techniques for the discovery and composition of web services. Semantic and syntactic techniques are the two common approaches to composition, consumption and possibly service integration. Generally semantic approaches support the integration of web services and functionality of semantic description.

Algorithm similarly attempts to understand the trends of the software developers such as Subsumption Relation, Common Subsets and Abbreviations in Naming that name their web services. In order to understand these trends, we manually download and verified the functionality of many open web services. Web services arrive from a number of web services repositories.

User interfaces are main query and sub queries. The proposed work searches in distributed web service repositories of Universal Description Discovery and Integration (UDDI) as per the query. This can be a service name, operation name, type name and descriptive fields from the service specification. Tendency-based Syntactic Matching-Levenshtein Distance and Letter-Pairing (TSM-LP) [1] algorithm is used for equivalent processing between input query and specific part of web service.

Tendency-based Syntactic Matching- Levenshtein Distance/Letter Pairing Algorithm (TSM-LP)
TSM-LP(Si , Sj) TSM-L function
LD (Si , Sj) Levenshtein distance function
FT1(Si) Trend-based threshold
FT2(Si) Trend-based threshold for letter pairing
Si, Sj Two strings for comparison
Length() String length functions
CS Web services category (such as Business)

FT1(Si)
*temp = [(Length(Si) * 2) / 3] – 2*
return temp
FT2(Si)
temp = Sensitivity (CS)
return temp

TSM-LP(Si , Sj)
if (LD(Si, Sj) <= FT1(Si)) or
(LP(Si, Sj) >= FT2(Si)) or
(Si ⊆Sj or Sj ⊆Si)
(Si > 3 and Sj > 3) and
(Si < 15 and Sj < 15)
return TRUE
else
return FALSE

Table 1 -TSM-LP Pseudo code

A. Discovering Trends in Web Service Message Naming:-

KDS approach uses specialized parsing tools to assess services from the bottom-up. We sought and found trends across the open web services. The three most common trends are listed as follows:

1. Subsumption Relationships—The strong trend for web service developers is to use part names based on common names. Similar messages aim to have strong subsumption relationships when using common names. For example, there are equivalent services where name = fname; name = middle name, and name = user name.
2. Common Subsets— This is similar to subsumption relationships, some part of web service names are related to have common subsets. For instance, we established equivalent component names last name = user name.
3. Abbreviations in Naming—Another strong trend was common names reduced into abbreviations. For example, circuit = ckt or communication = comm.

B. Using Natural Language Approaches to Exploit Trends in Message Matching:-

Numerous syntactic approaches were applied to take advantage of Trend 2 and 3 when matching services**.**

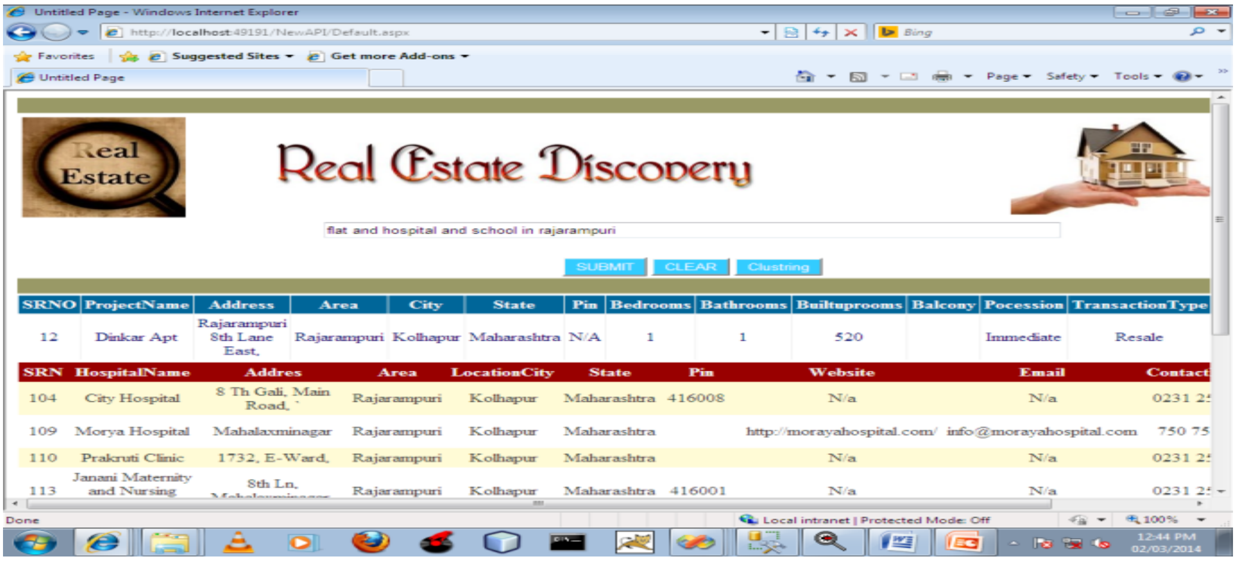
The Levenshtein distance (LD) (besides called the edit distance) assesses of similarity between two strings. LD is the smallest number of deletions, insertions, or substitutions which is necessary to transform a original string into a target string. The larger the LD, the more dissimilarity of the strings depends on LD. From preexisting sources, we adapt implementations of the LD algorithm from preexisting sources. The LD algorithm is successful when evaluating abbreviations with full strings. LD is also booming for similar strings that are changed to create uniqueness. There were a number of occurrences where zeros are substituted for the letter O or numbers are added to the end of a message name.

LD is helpful for similar strings, but it is not useful for strings that have similar subsets that do not have exact subsumption relations as in Trend 1. For example, last name and surname are equivalent but neither is a subset of the other. We employed the use of Letter Pairing for the account of instances of this nature. The Letter Pairing (LP) approach is an algorithm that can be used to match strings that have common subsets. Using the LP algorithm, two strings are separated into letter pairs. STR1 would be separated into ST, TR, and R1, and STR2 would be separated into ST, TR, and R2. A ratio of the number of identical pairs and the total number of pairs is calculated. This percentage is exercised to assess the similarity of the two strings.

C. Different Approaches for Syntactical Message Matching:-

The overarching matching algorithm is called Tendency-Based Syntactic Matching- Levenshtein Distance and Letter Pairing (TSM-LP). This algorithm exploits the previously mentioned trends using LD and LP. However, as a part of the service integration experimentation we will assess using the following four methods of syntactical matching:

- Equality—This method tests if two parts of message are syntactically the same.
- Subsumption—This method tests if the two parts of message have a subsumption relation (i.e., message part 1 is a subset of message part 2 or vice versa). Note: Subsumption contains equality.
- Levenshtein Distance (TSM-L)—This method adds to equality and subsumption by also testing to see if message parts are similar based on edit distance. Three thresholds based on varying strictness manage the algorithm. The three thresholds are determined based on the degree of self-similarity for the particular category for which the service is related.
- Letter-Pairing (TSM-P) This method extends equality and subsumption by inspecting similarity using letter pairing. Strictness thresholds are based on repository similarity and align with TSM-L.
- Levenshtein Distance/Letter-Pairing (TSM-LP)—This method merges all of the prior mentioned methods.



flat and hospital and school in rajarampuri

SRNO	ProjectName	Address	Area	City	State	Pin	Bedrooms	Bathrooms	Builtuprooms	Balcony	Procession	TransactionType
12	Dinkar Apt	Rajarampuri 8th Lane East,	Rajarampuri	Kolhapur	Maharashtra	N/A	1	1	520		Immediate	Resale

SRN	HospitalName	Addres	Area	LocationCity	State	Pin	Website	Email	Contact
104	City Hospital	8 Th Gali, Main Road, `	Rajarampuri	Kolhapur	Maharashtra	416008	N/a	N/a	0231 2
109	Morya Hospital	Mahalaxminagar	Rajarampuri	Kolhapur	Maharashtra		http://morayahospital.com/	info@morayahospital.com	750 75
110	Prakruti Clinic	1732, E-Ward,	Rajarampuri	Kolhapur	Maharashtra		N/a	N/a	0231 2
113	Janani Maternity and Nursing	8th Ln,	Rajarampuri	Kolhapur	Maharashtra	416001	N/a	N/a	0231 2

Fig.2 Process Query for Equivalence Processing

IV. CLUSTERING

A. Convert XML file to Database table

Once equivalences are identified between input query and specific parts of web services, web services are linked together by input and output messages. XML file is created for data fetched from web services with convert XML file to Database table.

B. Create Clusters

The K-Nearest Neighbour algorithm is one of the foundation algorithms in instance based learning. This algorithm starts from associating each instance to its corresponding point in an n-dimensional space. The closest values of an instance are determined using the formula for Euclidean distance. Thus, considering an instance x and $a_r(x)$ the value of the attribute r of the instance x, the following vector that describes this instance can be defined:

$<a_1(x), a_2(x),\ldots, a_n(x)>$

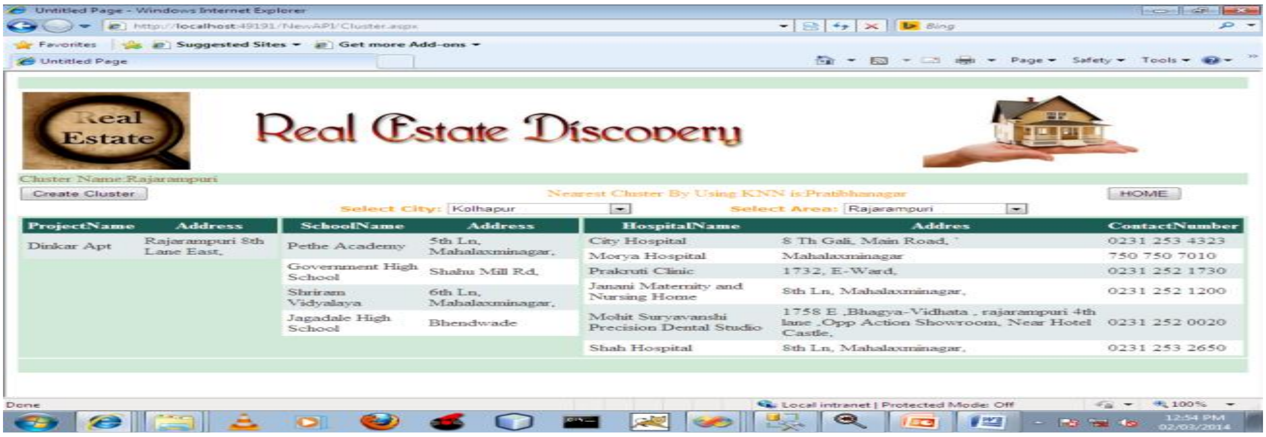
Now the distance among two examples x_i and x_j is defined like this:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

In the "nearest neighbour" finding, the goal function can present discreet values as well as real ones.

C. Cluster Creation

Cluster Rajarampuri



Untitled Page - Windows Internet Explorer

http://localhost:49191/NewAPI/Cluster.aspx

Real Estate Discovery

Cluster Name:Rajarampuri

Create Cluster

Nearest Cluster By Using KNN is:Pratibhanagar

HOME

Select City: Kolhapur

Select Area: Rajarampuri

ProjectName	Address
Dinkar Apt	Rajarampuri 8th Lane East,

SchoolName	Address
Pethe Academy	5th Ln, Mahalaxminagar,
Government High School	Shahu Mill Rd,
Shriram Vidyalaya	6th Ln, Mahalaxminagar,
Jagadale High School	Bhendwade

HospitalName	Addres	ContactNumber
City Hospital	8 Th Gali, Main Road, '	0231 253 4323
Morya Hospital	Mahalaxminagar	750 750 7010
Prakruti Clinic	1732, E-Ward,	0231 252 1730
Janani Maternity and Nursing Home	8th Ln, Mahalaxminagar,	0231 252 1200
Mohit Suryavanshi Precision Dental Studio	1758 E ,Bhagya-Vidhata , rajarampuri 4th lane ,Opp Action Showroom, Near Hotel Castle,	0231 252 0020
Shah Hospital	8th Ln, Mahalaxminagar,	0231 253 2650

Done

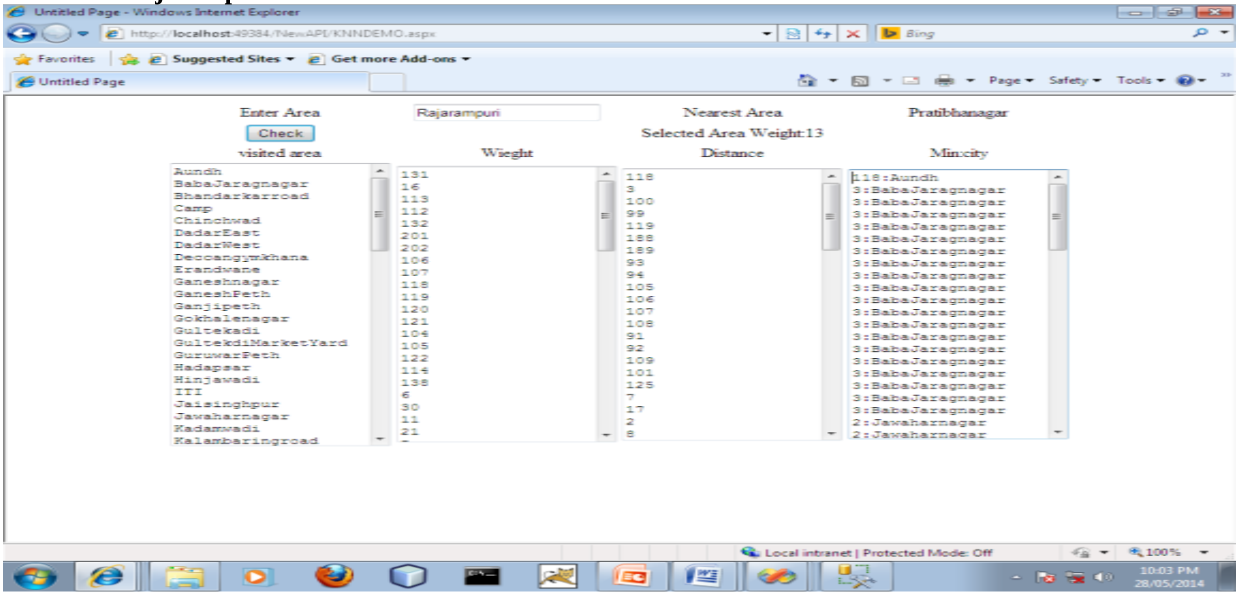
Local intranet | Protected Mode: Off

12:54 PM
02/03/2014

Fig.3: Cluster "Rajarampuri"

Create clusters by area wise in each city. Every area has assigned weight. Weight is increased by 1 as per consecutive areas in the particular city. Nearer areas having minimum weight difference calculated using Euclidean Distance.

D. Cluster Rajarampuri Demo



Untitled Page - Windows Internet Explorer

http://localhost:49384/NewAPI/KNNDEMO.aspx

Enter Area: Rajarampuri

Check

Nearest Area: Pratibhanagar

Selected Area Weight:13

visited area	Wieght	Distance	Min:city
Aundh	131	118	118:Aundh
BabaJaragnagar	16	3	3:BabaJaragnagar
Bhandarkarroad	113	100	3:BabaJaragnagar
Camp	112	99	3:BabaJaragnagar
Chinchwad	132	119	3:BabaJaragnagar
DadarEast	201	188	3:BabaJaragnagar
DadarWest	202	189	3:BabaJaragnagar
Deccangymkhana	106	93	3:BabaJaragnagar
Erandwane	107	94	3:BabaJaragnagar
Ganeshnagar	118	105	3:BabaJaragnagar
GaneshPeth	119	106	3:BabaJaragnagar
Ganjipeth	120	107	3:BabaJaragnagar
Gokhalenagar	121	108	3:BabaJaragnagar
Gultekadi	104	91	3:BabaJaragnagar
GultekdiMarketYard	105	92	3:BabaJaragnagar
GuruwarPeth	122	109	3:BabaJaragnagar
Hadapsar	114	101	3:BabaJaragnagar
Hinjawadi	138	125	3:BabaJaragnagar
ITI	6	7	3:BabaJaragnagar
Jaisinghpur	30	17	3:BabaJaragnagar
Jawaharnagar	11	2	2:Jawaharnagar
Kadamwadi	21	8	2:Jawaharnagar
Kalambaringroad			

Local intranet | Protected Mode: Off

10:03 PM
28/05/2014

Fig.4: Cluster "Rajarampuri" Demo

Create clusters using K-Nearest Neighbour Algorithm such as Rajarampuri cluster

V. RESULTS

Fig.1 shows Proposed Web Service Integration Architecture. Table 1 contains pseudo code for Tendency-Based Syntactic Matching- Levenshtein Distance and Letter Pairing (TSM-LP) algorithm. Equivalence processing takes input query from user that query processes on Flat Web Service, Hospital Web Service and School Web Service. Fig.2 shows Equivalence Processing output for flats availability, as well as nearby hospital and school for children in Rajarampuri at Kolhapur. Fig. 3 shows result of Rajarampuri cluster and Fig 4 shows Rajarampuri cluster demo using K-Nearest Neighbour Algorithm

VI. CONCLUSION

Equivalence processing module is implemented by using TSM-LP algorithm. Input Query and Service Specification of web services are matched. Input Query processes different Web Services such as Flat Web Service, Hospital Web Service and School Web Service. Create clusters using K-Nearest Neighbour Algorithm such as Rajarampuri cluster, Kothrud cluster and Dadar East cluster.

REFERENCES

[1] M. Brian Blake, Michael F. Nowlan, "Knowledge Discovery in Services (KDS): Aggregating Software Services to Discover Enterprise Integrations," *IEEE Transactions on, Knowledge and Data Engineering, vol.23, no.6, pp.889-901, June 2011.*

[2] S. Cetin, N.I. Altintas, H. Oguztuzun, A. Dogru, O. Tufekci, and S.Suloglu, "A Integration-Based Strategy for Migration to Service-Oriented Computing," *Proc. IEEE Int'l Conf. Pervasive Services,2007.*

[3] A. Jhingran, "Enterprise Information Integrations: Integrating Information,Simply," *Proc. 32nd Int'l Conf. Very Large Data Bases,2006.*

[4] M. Sabbouh, J. Higginson, S. Semy, and D. Gagne, "Web IntegrationScripting Language," *Proc. 16th Int'l Conf. World Wide Web, 2007.*

[5] J. Zou and C.J. Pavlovski, "Towards Accountable Enterprise Integration Services," *IEEE Int'l Conf. E-Business Eng., pp 205-212,Oct. 2007.*

[6] M. Brian Blake, "Knowledge Discovery in Services," *IEEE Internet Computing, vol. 13, no. 2, pp. 88-91, Mar. 2009.*

[7] Semih Cetin, N. Ilker Altintas, Halit Oguztuzun, Ali H. Dogru, Ozgur Tufekci, and Selma Suloglu, "A Integration-Based Strategy for Migration to Service-Oriented Computing," *Proc. IEEE Int'l Conf. Pervasive Services,2007.*

[8] McIlraith SA, Son TC, Zeng H, Semantic Web services. *IEEE Intell Sys 16(2):46–53*, 2001.

[9] Chakraborty D, Perich F, Joshi A, Finin T, Yesha Y, "A reactive service composition architecture for pervasive computing environments". *In: Proceedings of the 7th personal wireless communications conference, Singapore, pp 53–62, October 2002.*

[10] Chakraborty D, Perich F, Avancha S, Joshi A "DReggie: a smart service discovery technique for e-commerce applications". *In: Proceedings of the workshop at the 20th symposium on reliable distributed systems, New Orleans, October 2001.*

[11] Bussler C, Fensel D, Maedche A "A conceptual architecture for Semantic Web enabled Web services". *SIGMOD Rec 31(4):24–29*, 2002.

[12] Cardoso J, Sheth A "Semantic e-workflow composition". *Technical report, LSDIS Lab, Computer Science, University of Georgia*, 2002.