ISSN 0973-9777

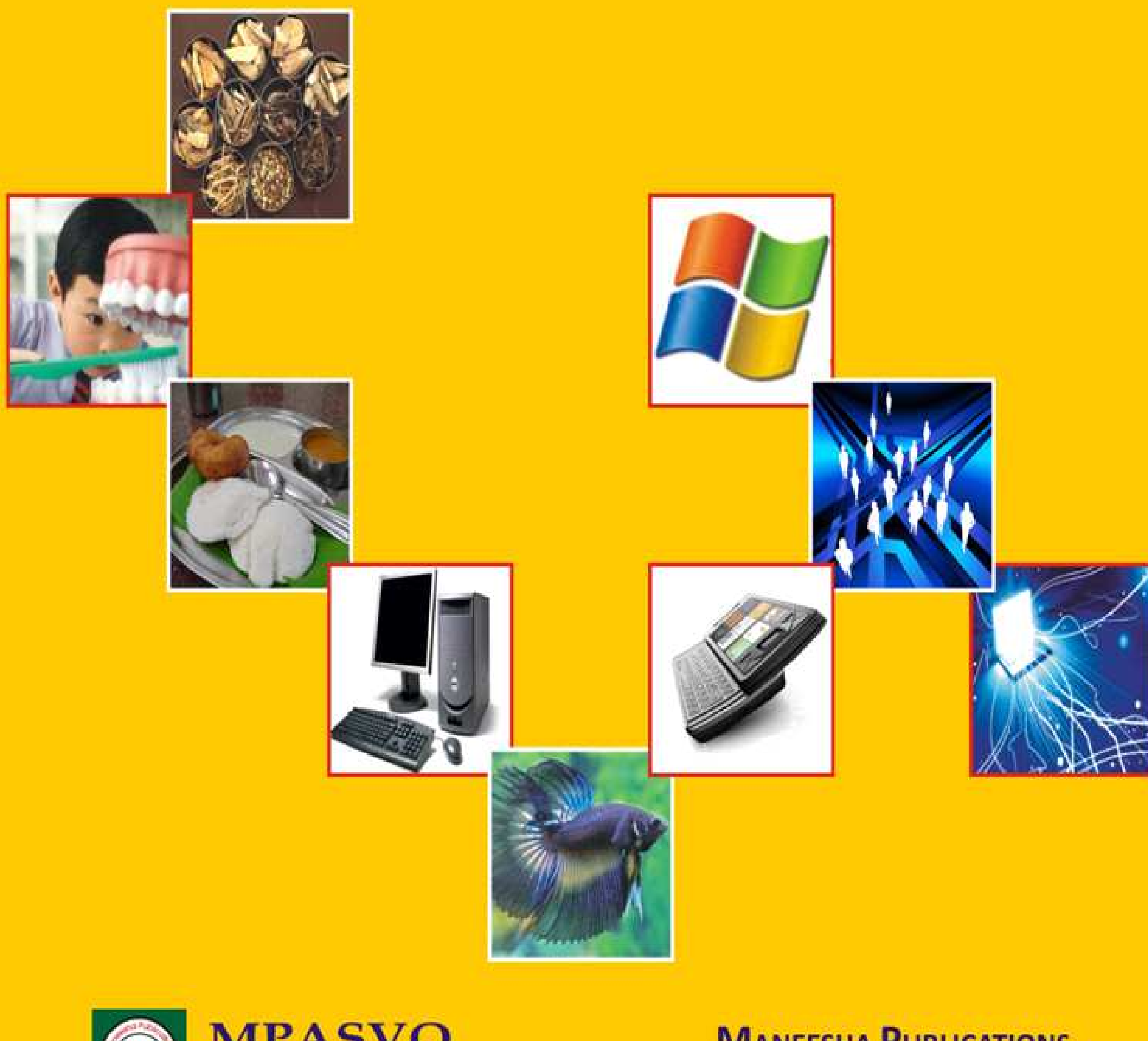
Volume-6 Number-4 July-August 2012

The Indian Journal of Research

ANVIKSHIKI

Bi-monthly International Journal of all Research

Science



MPASVO

Published on behalf of the MPASVO in association with the Member's of Anvikshiki

MANEESHA PUBLICATIONS

www.anvikshikijournal.com

Anvikshiki
The Indian Journal of Research

Bi-Monthly International Journal of All Research

Abstracts and Indexing

http//nkrc.niscair.res.in/browseByTitle.php?Keword=A, ICMJE ICMJE ,www.icmje.org, Academia.edu, banaras.academia.edu, ebookbrowse.com, BitLibrary! http:/ www.bitlib.net/, Tech eBooks ,freetechebooks.com, ARTAPP.NET ,artapp.net,Catechu PDF / printfu.org, File Away ,www.fileaway.info, KMLE ,docslibrary ,http://www.docslibrary.com, MyCelular.ORG ,Android Tips, Apps, Theme and Phone Reviews http://dandroidtips.com, Edu-Doc, http://www.edu-doc.com, www.themarketingcorp.com,Dunia Ebook Gratis duniaebook.net, ,www.cn.doc-cafes.com., Google scholar ,http://scholar.google.co.in, Website : www.onlineijra.com.Motilal Banarasi Das Index,Varanasi, Motilal Banarasi Das Index,Delhi. Banaras Hindu University Journal Index,Varanasi. www.bhu.ac.in, D.K.Publication Index, Delhi. National Institute of Science Communication and Information Resources Index, New Delhi.

Subscriptions

Anvikshiki,The Indian Journal of Research is Published every two months (January,March,May,July,September and November) by mpasvo Press,Varanasi.u.p.India. A Subscription to The Indian Journal of Research : Anvikshiki Comprises 6 Issues in Hindi and 6 in English and 3 Extra Issues. Prices include Postage by Surface mail,or For Subscription in the India by Speed Post. Airmail rates are also available on request. Annual Subscriptions Rates (Volume 3,6 Issues in Hindi,6 Issues in English and 6 Issues of science 2012):

Subscribers

Institutional : Inland 4,000 +500 Rs. P.C., Single 500+51 Rs.P.C.,Overseas 6000+2000Rs. P.C., Single 1000+500 Rs.P. C.

Personal : 2,500+500 Rs. P.C., Single 500+51 Rs. P.C., Overseas 5000+2000Rs.P.C., Single 1000+500Rs. P.C.

All correspondence related to the Journal should be addressed to

B.32/16 A., Flat No.2/1,Gopalkunj,Nariya,Lanka, Varanasi, U.P.,India
Mobile : 09935784387,Tel.0542-2310539.,e-mail : maneeshashukla76@rediffmail.com,www.anvikshikijournal.com
Office Time : 3-5 P.M.(Sunday off)

Journal set by

Maheshwar Shukla,maheshwar.shukla@rediffmail.com
9415614090

Printed by

mpasvo Press

Maneesha Publication
(Letter No.V-34564,Reg.533/2007-2008)
B-32/16-A-2/1,Gopalkunj,Nariya,Lanka
Varanasi,U.P.,India

Anvikshiki
The Indian Journal of Research
Volume 6 Number 4 July 2012

Science
Papers

PRINT ISSN 0973-9777,WEBSITE ISSN 0973-9777

Letter No.V-34564,Reg.533/2007-2008 ANVIKSHIKI ISSN 0973-9777
INDIAN JOURNAL OF RESEARCH(2012)**6,33-40**
Advance Access publication 26 Apr. 2012

A NOVEL SKELETON TO EXPLORE PROVIDER'S CONTEXT AND USER'S CONTEXT

JITENDRA PRATAP*

Declaration

The Declaration of the author for publication of Research Paper in The Indian Journal of Research Anvikshiki ISSN 0973-9777 Bi-monthly International Journal of all Research: I, *Jitendra Pratap* the author of the research paper entitled A NOVEL SKELETON TO EXPLORE PROVIDER'S CONTEXT AND USER'S CONTEXT declare that , I take the responsibility of the content and material of my paper as I myself have written it and also have read the manuscript of my paper carefully. Also, I hereby give my consent to publish my paper in Anvikshiki journal , This research paper is my original work and no part of it or it's similar version is published or has been sent for publication anywhere else. I authorise the Editorial Board of the Journal to modify and edit the manuscript. I also give my consent to the Editor of Anvikshiki Journal to own the copyright of my research paper.

Abstract

Web service plays an important role in implementing Service Oriented Architecture (SOA) for achieving dynamic business process. With the increased number of web services advertised in public repository, it is becoming vital to provide an efficient web service discovery and selection mechanism with respect to a user's requirement. Considerable efforts have been attained in solving this problem among them semantic based approaches show encouraging result. However, when several semantically equivalent web service candidates are returned by matchmaking process, how to discern which one is the most suitable one is a real challenge.

In this paper we present a framework by which provider's context and user's context has been explored to help understand the real need of a user. This framework uses association rule for context modeling of provider and user from web service composition perspective. After modeling context, a ranking service is invoked to compare web service candidates with user's requirement and then the best suitable potential web service will be selected.

Keywords : web service selection, ranking,, association rule

I. Introduction

The World Wide Web is evolving from a sea of information to a service oriented marketplaces and Web Service (WS) technology is the next wave of Internet computing. Web service is one of the fastest growing areas of information technology in recent years. Web services expose business processes over the Internet and promise more business opportunities by providing a common protocol that can be used by Web applications to communicate with each other over the Web [1].

*M.Tech in C.S, Singhania University (Rajasthan) India.

Traditionally, matchmaking algorithms [2] used in web service discovery considers only one service as a suitable candidate satisfying a service request while combinations or compositions of services are not considered, there fore service registries are expected to store large numbers of services and at the same time the best matching set of services for a given query has to be retrieved in a timely manner Such a combination of requirements makes it difficult to employ full-fledged composition algorithms [2] during the discovery process simply because their time complexity is unacceptable for the discovery purposes.

The remainder of this paper is structured as follows. In section II, we discuss the background of web service selection and composition, section III is an overview of our framework, section IV describes our algorithm with example, section V gives details about our experiment results and section VI is concluded with the future work.

II. Background

A.Web Service

A Web service is an accessible application that other applications and humans as well, can automatically discover and invoke. An application is a Web service if it is [3]: (i) independent as much as possible from specific platforms and computing paradigms; (ii) developed mainly for inter-organizational situations rather than for intra-organizational situations; and (iii) easily compos able (i.e., its composition with other Web services does not require the development of complex adapters)

Web services are, in practice, transient and stateless processes that exist only during service execution, which is triggered by a request coming from a consumer, or client. Services are instantiated to perform specific tasks, thus facilitating scalable, concurrent service provision.[4] The design of a web service is usually defined as a clearly articulated workflow, for the sake of reliability and quality of service.

B. Web Service Selection

The WS selection problem has been extensively studied in the past few years. Previous works have focused on optimizing the selection of WSs for a single activity, while the most recent ones focus on the selection of WSs in order to satisfy the QoS requirements of a workflow (or composite WS). To optimize the selection for the entire process, Zeng et al.[10] proposed computing an optimal set of WSs for each possible execution path in the process based on a weighted combination of QoS measures. When it comes to deciding a WS for a given activity t, the most popular execution path on which t appears is chosen, and the WS assigned to t in that path is selected. To speed up the computation, techniques of integer programming are employed. However, the number of possible execution paths could be huge, especially in the presence of a loop. Besides, the failure of WSs at runtime requires the recomputation of a set of new integer programs, which may not be feasible in practice. Yu et al.[11] proposed a scheme to optimize the end-to-end QoS for various flow structures. They used a utility function derived from the QoS quality and formulated the optimization problem as a general flow problem. Some works concentrate only on a single QoS measure. Menasce [13] proposed a scheme to estimate the throughput of a composite WS from those of its constituent WSs and to use throughput as a basis for selecting WSs. Grassi and Patella[12] proposed a framework to recursively aggregate the reliability of a composite WS based on those of its constituent WSs. Several works have been devoted to the derivation of other QoS measures, such as response time, cost, availability, and fidelity, of a composite WS out of those of its constituent WSs [6]. All of the above-mentioned works adopt a rather simple model that views a WS as the basic unit for composing a composite WS. Our work adopts a different model in which every WS (composite or component) comprises a number of operations, and the invocations of their operations may have to follow some particular orders. In addition, our proposed approach does not need to enumerate all possible execution paths before a process is initiated and allows partial recomputation when a candidate WS is found to fail at runtime, potentially incurring less computation time.

C.Web Service Composition

Service Composition refers to the process of creating customized services from existing services by a process of dynamic discovery, integration and execution of those services in a deliberate order to satisfy user requirements [2].

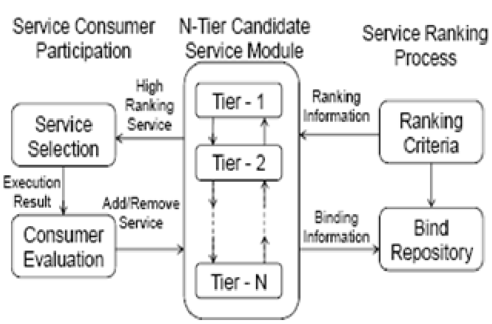
Consider the task of comparing products on the web. In the absence of automated composition of services, the user invests considerable resources visiting numerous sites, determining appropriate service providers, entering his preferences repeatedly, integrating or aligning the different type of results coming from different sites. We would prefer that the user enters information once and receives the expected results from the most appropriate services with minimal additional assistance.

Service composition is required to improve the quality of E-business [9]. It provides two important benefits. First, more automation, it reduces the amount of human interaction needed to exchange or transfer information. Second, more flexibility, the sub-services can be composed depending on the client's requirement. Satisfying the client is the heart of doing business or providing services.

III Frame Work

In this section, we present our adaptive framework, which not only selects the most highly linked services when new services are needed, but also retains as much as possible the initial pre-deployed selection of services. Our framework also has a feature that selects services among those within the same tier.

In addition, when an execution of a selected service produces a failure in a service composition, the framework will remove the service from the registry. After a previously failed service is recovered, the framework computes the link analysis score of the public service registries to check how the service is being used by other consumers. Intuitively, in the long run, only "good" services will be kept in different tiers and available for selection by the consumer. Our framework is realized by our Algorithm.



Initially, information from public service registries is captured and recorded locally in the *binding repository*, and the potential services by the consumer are kept in an *N-Tier Candidate Service Module* (*N- TCSM*). Formally, an *N-TCSM* is a list of *N* elements, denoting *N* tiers. Each element is a set of candidate services (of the same or different service types).

To handle a user request, the *Service Selection* component selects a set of candidate services from the *NTCSM* so that the service consumer can form its service composition.

For a service consumer who requests *m* services of the same kind from the *N-TCSM*, the algorithm will do the following: Starting from $i = 1$, if k ($\geq m$) such services are available in *N- TCSM*[*i*], then select *m* services randomly from these *k* services; otherwise, select the services from *N-TCSM*[*i*], and continue to select the remaining $m - k$ services from the subsequent *NTCSM*[*i*+1], *N-TCSM*[*i*+2], ... until a total of *m* services have been selected.

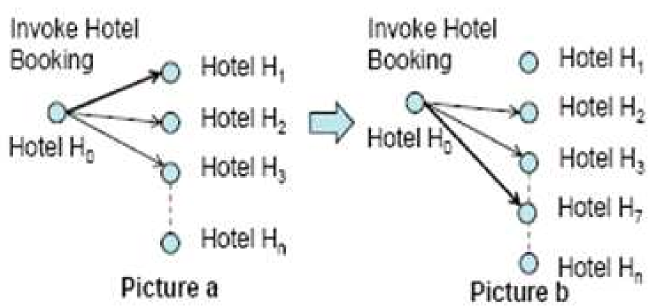
After executing a selected service *e* from *N-TCSM*[*i*] in a service composition, if a failure results, *e* will be removed from *N- TCSM*[*i*]. Such handling is done by the *Consumer Evaluation* component. Furthermore, the corresponding binding entity of *e* will also be removed from the binding repository. Whenever a service *e* has been removed from a tier, a new service *x* with the highest estimated ranking but not in the *N-TCSM* will be placed initially into the lowest tier of the *N-TCSM*. The actual tier of the newly added service will be determined by the *Ranking Criteria*.

The *Ranking Criteria* component provides the ranking facility by applying the link analysis algorithm to the *binding repository*. Candidate services in the *N-TCSM* will change their belonging tiers according to the relative scores of the services computed by the link analysis algorithm. It should fully fill *N-TCSM*[*i*] with services with the highest scores, followed by filling *N-TCSM*[*i*+1] with the remaining services.

IV Algorithm

1. Look up the service for the current activity; get the competitive candidate services from registry.
2. Invoke the evaluation framework to rank the candidate services based on the composition context which composites the services according to the user's requirements (containing knowledge of previously invoked services, the current candidate services and also user context).
3. Invoke the highest scoring service first. If there is no service available, the execution has failed.
4. If an error occurs when the service is invoked, then record this error information. Return to step 3 to select the next best service. This step adds failure tolerance, which in a distributed setting is essential.
5. If the invocation finishes successfully, log the execution details to the context store. Move to next activity and return to step 1.

Let us use a scenario in the *TripHandling* application to further illustrate the algorithm. Suppose *n* candidate hotel services are bound to the workflow step *HotelBooking*, as shown in Figure 5(a). We can apply the algorithm to compute their service ranking scores. Based on the scores, the *Ranking Criteria* will rearrange the services in different tiers in the *N-Tier Candidate Service* accordingly.



Let H_0 be the hotel booking service for the *TripHandling* process, while in fact, H_0 invokes external services to do the actual booking. Suppose service H_1 (among compatible services H_1, H_2, …, H_n as shown in Figure 5(a)) is selected and executed. Unfortunately, this particular execution of H_1 results in a failure, and hence H_1 is removed from the *N-TCSM* of H_0. The link <H_0, H_1> shown in bold in the figure is also removed from the SO network. Then, the algorithm selects a new hotel booking service (say, H_7) with a high estimated ranking to replace H_1, as shown in Figure 5(b), and the link <H_0, H_7> is added into the SO network .

A. Personalized Association Rule Generator: Association rule is one of the most important techniques in the domain of data mining. There are two important concepts in association rule, *support* and *confidence*.

The main idea is to utilize the association rules among different web services to identify the dependence between the given web service and other web services. Formally, given a user u_k, we can find a set of neighbors of that user. Afterwards, a database containing all web service composition transactions by those users during the specified time interval is constituted, which is called that user's personal composition database. We can then association rule mining algorithm, e.g. Apriori algorithm [5,6], a most popular and widely applied association rule mining method, to get all association rules within this database. Given any web service $s \in S$, we can select a web service set S' = {*s*1, *s*2,…, *s*w} that satisfying:

$support\ (s \rightarrow S') > \gamma$

$confidence(\ s \rightarrow S') > \delta$

B. Ranking Mechanism: After building personal profile for each user, whenever a user is submitting a web service query request, the personal association rules will be employed with other web services in the composition to facilitate the final ranking process. Formally, given a set of candidate C={c1, c2, c3,…, cm} and a set of web services S={*s*1, *s*2,…, *s*n}, which are web services being joined with one of the candidates to fulfill the particular task, the ranking steps include:

1) Calculate selection weight for each candidate under circumstance of S. In this study, we only consider one-to-one association rule. The assumption is to simplify the problem. Selection weight can be defined as:

$$\text{Weight(Ci)} = \sum_{j=1,sj \in S}^{n} \text{confidence(Sj} \rightarrow \text{Ci)}$$

2) Sort *weight*(*i c*) and return the top N candidates.

V. Evaluation

This section reports on the experimentation of our proposal.

A. Experiment Design: We use the *TripHandling* application[7] to evaluate our work. It generates, in total, 200 hotel service consumers, 1000 hotel service providers, and 100 service registries. We use a 3-*Tier Candidate Service Module* in the simulation, and label the three tiers as *reliable* tier, *available* tier, and *backup* tier, respectively.

We randomly select 0.5%, 1%, and 2% of 1000 hotel services and put them into the service pools of the *reliable*, *available*, and *backup* tiers, respectively. After an invocation of a service, if it encounters a failure, the service is removed from the 3-*TCSM*. If a service has been removed from the 3-*TCSM*, a new service from the remaining service provider pool will be selected and added to the 3-*TCSM* according to our framework by applying our algorithm. Since their priorities may be changed, the candidate services in the 3-*TCSM* are re-divided into different tiers according to the new ranking information.

Initially, each service consumer randomly selects a set of services to compose its required service. This experimental setting simulates a random sample of pre-deployment information for initial service selection. The initial settings of all service consumers are listed in Table 1, in which columns L1 to L4 individually present the number of services in each category used by all service consumers.

T A B L E 1 *Initial statistics of 3-TCSM (metric M-Q).*

Tier	L1	L2	L3	L4	
Tier-1 (Reliable)	543	544	562	551	
Tier-2 (Available)	602	609	601	588	Overall
Tier-3 (Backup)	625	580	592	603	Quality
Total Count	1770	1733	1755	1742	M-Q
Failure Rate	0.01%	0.1%	1%	10%	2.77%

We choose the following two metrics as the effectiveness measures to evaluate our approach. The overall quality of the 3-*TCSM* (denoted by *M-Q*) is defined as the average failure rate of all candidate services in the 3-*TCSM*.

The average failure rate per service invocation (denoted by *M-FR*) is defined as the average failure rate experienced by a consumer over a sequence of user requests for that service composition since the first request. For instance, if 5 failures result from a service composition when fulfilling a sequence of 100 user requests, then *M-FR* is 0.05. Since PageRank is highly representative, we choose it as the link analysis algorithm *F* as input to our algorithm in the experiment. Referring to [8,9], we terminate the execution of PageRank after 200 iterations. We invoke *our algorithm* 4096 (= 212) times, and repeat the experiment 10 times to report the average result.

B. Data Analysis: After the simulation, the statistics of services as kept in the 3-*TCSM* is shown in Table 2. The overall quality *M-Q* after adopting our approach is 1.92%. Compared with the *M-Q* value of 2.77% for the initial setting in Table 1, the overall quality of the 3-*TCSM* services has improved by 31%.

T A B L E 2 *Average statistics of 3-TCSM after running simulation (metric M-Q).*

Tier	L1	L2	L3	L4	
Tier-1 (Reliable)	1156	852	142	50	
Tier-2 (Available)	614	617	629	541	Overall
Tier-3 (Backup)	625	584	596	594	Quality
Total	2393	2053	1367	1185	M-Q
Failure Rate	0.01%	0.1%	1%	10%	1.92%

We observe from Tables 1 and 2 that, using our approach, the chance of selecting a high-quality candidate service (say, from categories L1 and L2) has increased significantly.

We sample the 4096 invocations of our algorithm at 2*i* steps, where *i* ranges from 0 to 12. The sampling results of our approach and the random approach are plotted as two data series in Figure 6. The *X*-axis indicates the number of service invocations, and the *Y*-axis indicates the failure rate.

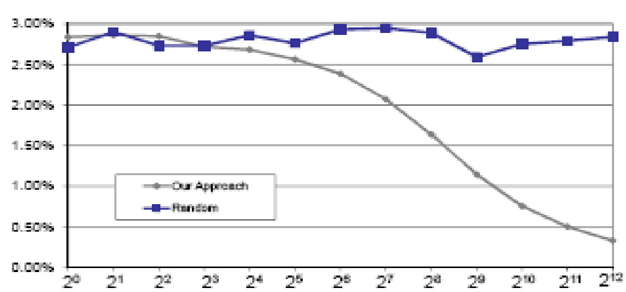


Figure 6. Comparison with random (static) (metric M-FR)
[*X*-axis: no. of invocations; *Y*-axis: failure rate].

We observe from Figure 6 that, as the number of service invocations increases to 212, the average failure rate of executed services (*M-FR*), randomly selected from Tier-1, is 0.326%, while that of the random approach is 2.84%. Thus, the average failure rate using our approach is only 11.5% of that using the random approach.

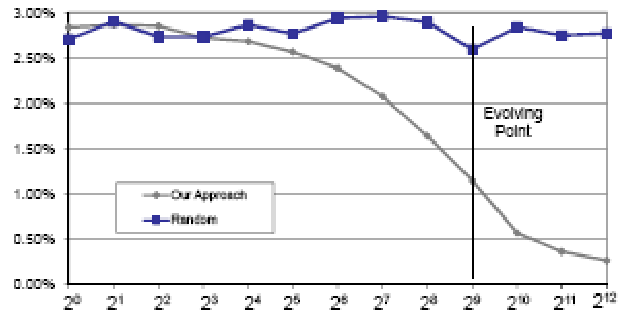


Figure 7. Comparison with random (evolving at 29)
(metric M-FR) [*X*-axis: no. of invocations; *Y*-axis: failure

Figure 7 shows a comparison with the random approach under the scenario of evolving service quality for our Experiment

B. The *X*-axis and *Y*-axis are the same as those of Figure 6. By comparing Figures 6 and 7, we observe that our approach can work even better after the quality evolving, such as achieving a failure rate of 0.265% after 212 invocations in Figure 7 as against a failure rate of 0.326% in Figure 6. This observation shows that our approach is promising in solving the problem of evolving service quality. Owing to space limitation, we have to leave other details of the experiment and analysis to future publications.

VI Conclusion and Future work

Selecting the best suitable services to complete a complex composite service is an important research topic. We have formulated the dynamic WS selection problem in a dynamic and failure-prone environment. We introduce a Web service selection method combining more than two services within the same tier or a different tier.

Furthermore, in this paper the ranking mechanism only considers one-to-one association rule like *si*®*sj*. This assumption is to simplify the problem. However, dependency also exists between more than two web services, such as (*si*,…*sj*) ®sk.

In the future, we will analyze more composition scenarios and consider completeness of the composition context.

References

1. M. Chhabra, H. LU, *"Towards Agent Based Web Services"*, 6th IEEE/ACIS International Conference on Computer andInformation Science (ICIS 2007), 2007.

2. M. Paolucci, T. Kawamura, T. R. Payne & K. P. Sycara. Semantic matching of web services capabilities. In *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2002.

3. Wang, Jiying, *"Information Discovery, Extraction and Integration for the Hidden Web"*, Proc. VLDB PhD Workshop, 2003.

4. Mcilraith, S., Son T., Zeng, H. *"Semantic Web Services"*.IEEE Intelligent Systems, SpecialIssue on the Semantic Web, Volume 16, No. 2, pp. 46-53,March/April, 2001.

5 R. Agrawal, T. Imielinski & A.N. Swami, *"Mining Association Rules between Sets of Items in Large Databases"*, In Proceedings Of 1993 ACM International Conference on Management of Data, Washington D.C, USA, 1993, pp. 207-216.

[6] R. AGRAWAL & R. SRIKANT, *"Fast Algorithms for Mining Association Rules in Large Databases"*, In Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, 1994, pp. 487-499.

[7] *Travel Handling*. IBM Bpel Repository available at www.alphaworks.ibm.com/tech/bpelrepository.

[8] S. BRIN & L. PAGE. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1–7): 107–117, 1998.

[9] M. SOBEK. *A Survey Of Google's Pagerank. Efactory Gmbh And Co.,*USSELDORF, GERMANY, 2002/03. Available at http://pr.efactory.de/.

[10] L. ZENG, B. BENATALLAH, M. DUMAS, J. KALAGNANAM & Q.Z. SHENG, *"Quality Driven Web Services Composition,"* Proc. 12th Int'l Conf. World Wide Web (WWW '03), pp. 411-421, 2003.

[11] T. YU, Y. ZHANG & K.J. LIN, *"Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints,"* ACM Trans. Web, vol. 1, no. 1, pp. 1-25, 2007.

[12] V. GRASSI & S. PATELLA, *"Reliability Prediction for Service- Oriented Computing Environments,"* IEEE Internet Computing, vol. 10, no. 3, pp. 43-49, 2006.

[13] D.A. MENASCE, *"Qos Issues in Web Services,"* IEEE Internet Computing, vol. 6, no. 6, pp. 72-75, 2002.

Note for Contributors

SUBMISSION OF PAPERS

Contributions should be sent by email to Dr. Maneesha Shukla Editor-in-Chief, Anvikshiki, The Indian Journal of Research (maneeshashukla76@rediffmail.com). www.onlineijra.com

Papers are reviewed on the understanding that they are submitted solely to this Journal. If accepted, they may not be published elsewhere in full or in part without the Editor-in-Chief's permission. Please save your manuscript into the following separate files-***Title; Abstract; Manuscript; Appendix.*** To ensure anonymity in the review process, do not include the names of authors or institution in the abstract or body of the manuscript.

***Title*:** This title should include the manuscript, full names of the authors, the name and address of the institution from which the work originates the telephone number, fax number and e-mail address of the corresponding author. It must also include an exact word count of the paper.

***Abstract*:** This file should contain a short abstract of no more than 120 words.

MANUSCRIPT: This file should contain the main body of the manuscript. Paper should be between 5 to 10 pages in lenth,and should include only such reviews of the literature as are relevant to the argument. An exact word count must be given on the title page. Papers longer than 10 pages (including *abstracts*, *appendices and references*) will not be considered for publication. Undue length will lead to delay in publication. Authors are reminded that Journal readership is abroad and international and papers should be drafted with this in mind.

References should be listed alphabetically at the end of the paper, giving the name of journals in full. Authors must check that references that appear in the text also appear in the References and *vice versa*. Title of book and journals should be italicised.

***Examples*:**

BLUMSTEIN,A.and COHEN,J.(1973),'A Theory of Punishment' *Journal of Criminal Law and Criminology*,64:198-207

GUPTA,RAJKUMAR(2009),*A Study of The Ethnic Minority in Trinidad in The Perspective of Trinidad Indian's Attempt to Preserve Indian Culture*, India: Maneesha Publication,

RICHARDSON,G(1985),Judicial Intervention in Prison Life', in M. Maguire ,J. Vagg and R. Morgan, eds., *Accountability and Prisons*,113-54.London:Tavistocs.

SINGH,ANITA.(2007),*My Ten Short Stories*,113-154.India:Maneesha Publication.

In the text,the name of the author and date of publication should be cited as in the Harvard system(e.g.Garland 1981: 41-2;Robertson and Taylor 1973;ii.357-9)If there are more than two authors, the first name followed by *et al.* is manadatory in the text,but the name should be spelt out in full in the References. Where authors cite them as XXXX+date of publication.

Diagrams and tables are expensive of space and should be used sparingly. All diagrams, figures and tables should be in black and white, numbered and should be referred to in the text.They should be placed at the end of the manuscript with there preferred location indication in the manuscript(e.g.Figure 1 here).

Appendix: Authors that employ mathematical modelling or complex statistics should place the mathematics in a technical appendix.

NOTE : Please submit your paper either by post or e-mail along with your photo, bio-data, e-mail Id and a self-addressed envelop with a revenue stamp worth Rs.51 affixed on it. One hard copy along with the CD should also be sent.A self-addressed envelop with revenue stamp affixed on it should also be sent for getting the acceptance letter. Contributors submitting their papers through e-mail, will be sent the acceptance letter through the same. Editorial Board's decision will be communicated within a week of the receipt of the paper. For more information, please contact on my mobile before submitting the paper. All decisions regarding members on Editorial board or Advisory board Membership will rest with the Editor. Every member must make 20 members for Anvikshiki in one year. For getting the copies of 'Reprints', kindly inform before the publication of the Journal. In this regard, the fees will be charged from the author.

Search Research papers of The Indian Journal of Research Anvikshiki-ISSN 0973-9777 in the Websites given below

http://nkrc.niscair.res.in/BrowseByTitle.php?keyword=A

National Knowledge Resource Consortium

www.icmje.org

www.scholar.google.co.in

www.kmle.co.kr

www.fileaway.info

www.banaras.academia.edu

www.www.edu-doc.com

www.docslibrary.com

www.dandroidtips.com

www.printfu.org

www.cn.doc-cafes.com

www.freetechebooks.com

www.google.com

IJRA

www.onlineijra.com

ISSN 0973-9777

09739777

₹ 1500/-