IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 4, No 1, July 2013
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

An Effective Web Service Ranking Algorithm based on Quality of Experience of Users

Vandan Tewari[1], Nirmal Dagdee[2] and Aruna Tiwari[3]

**[1] Dept. of Comp. Tech. & Appl. , S.G.S.I.T.S
Indore- 452003,MadhyaPradesh,India**

**[2] Dept. of Computer Engg., S.D. Bansal College of Technology, Umaria
Indore, MadhyaPradesh, India**

**[3] Dept. of Computer Sc.,IIT, Indore,
Indore, MadhyaPradesh, India**

Abstract

With the increasing number of Web services, discovering and selecting best services for a client is becoming very significant. While discovering a user can benefit from experiences of other users. This can actually be exhibited through a collaborative filtering mechanism where a user is able to rate a service based on his experiences. A user can be offered services based on the Quality of Experience (QoE) of all the users which have used the given services in past. The service ratings given by all the users can be aggregated into a single list to prepare the overall service ranking which can be rendered to a client to help him in selection of better service. Further if a user wants to see the service ranking on other aspects such as its popularity etc., an aggregate ranking of services is presented using different ranking parameters. This paper presents a client oriented approach of Service rating and rank aggregation based on user oriented QoE based rating as well as popularity.

Keywords: *Web Service, Quality of Experience, Service Rating, Service Ranking.*

1. Introduction

In the age of globalization, day by day business to business and business to consumer operations are finding huge importance in internet computation around the world. Web services [1] are one means by which we can fulfill all these demands in an easy and efficient way.

Web Services are based on Service Oriented Architecture[2] which enables application-to-application communication over the internet and easy accessibility to heterogeneous applications and devices. As web services become more popular model for Internet computing, the issues of effective and appropriate service discovery become of utmost importance. The web service search using search techniques supported by existing UDDI[3] APIs may not result in the search results that are appropriate to service requestor's needs.

Current proposals for web service discovery presents the same search results to all clients for the similar query. However evidently the different users have different needs and an objective for web service discovery and therefore it is essential that these differences are accounted for while discovering services for a client. Therefore there is an urgent need of identifying the needs of a client for discovery for rendering him the services which he actually desires.

There will be a large pool of discovered services which fulfills the functional requirements of a user. However to select an appropriate service from this pool is still an issue. To help a user in finding a 'good' service, the past experiences of other users might be used. The users may be asked to give their feedbacks in terms of service's overall behavior such as 'value for money', 'satisfaction level', 'trustworthiness' etc which actually represent its Quality of Service (QoS) behavior. These parameters collectively can be thought as Quality of Experience (QoE) and can be used for ranking a service in its pool.

In this paper an Effective Web Service Ranking Algorithm based on Quality of Experience of Web Service users has been proposed.

2. Service Rating and Ranking

A ranking list of n services is just a vector of permutations of integers 1 through n. In contrast a rating of services is assigning a numerical score to each service. A sorted rating list creates a ranking list. The rank of a web services is its relative importance to the other web services in the set. A service ranking model is a method of determining a way in which the ranks are assigned to services in a group. Typically a ranking model uses information available to determine a rating for each service. Once we have the ratings the assignment of the service ranks can be as trivial as sorting the services in the descending order of the corresponding ratings.

When multiple services with the similar functionality are discovered for a user, it is difficult for the user to choose any one of them. To ease of this selection, the services can be ranked on various parameters such as User feedback, Popularity, Cost etc. Evidently these requirements vary from user to user. As discussed earlier the services can be rated those users who have already experienced these services. These users can rate services as per their quality of experience despite being aware of QoS parameters. The services can also be ranked based on the cost which is an important parameter for a client while choosing a service.

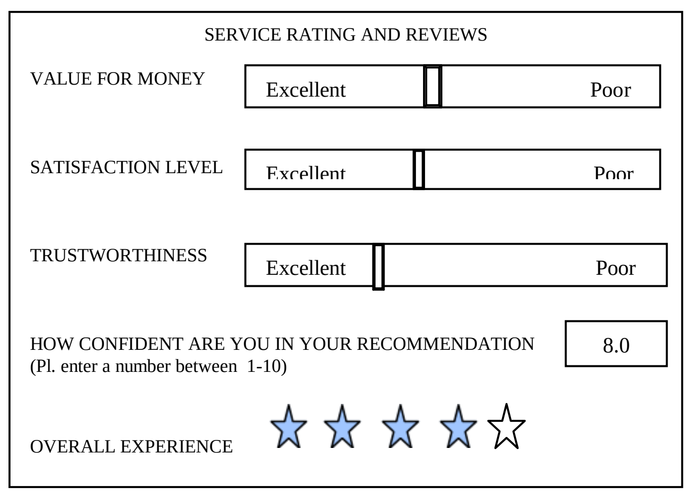
3. Related Work

A noteworthy contribution has been made in field of web service ranking by various researchers. In [4],Le-Hung Vu et al. have presented present a QoS-based semantic web service selection and ranking solution with the application of a trust and reputation management method to address the problem of false QoS rating claimed by providers. The UX architecture presented in [5] proposes use of dedicated servers to collect feedback of consumers and then predict the future performance of published services. In [6], a proposal has been given where services are allowed to vote for quality and trustworthiness of each other and the service discovery engine utilizes the concept of distinct sum count in sketch theory to compute the QoS reputation for every service. However, these reputation management techniques are still simple and can suffer from problems of cheating behaviors. Authors have proposed a model of reputation-enhanced QoS based Web services discovery which uses a reputation manager to assign reputation scores to the services based on customer feedback of their performance in [7]. A discovery agent facilitates QoS-based service discovery using the reputation scores in a service matching, ranking and selection algorithm. In [8] E. Al Masri et al. have introduced the Web Service Relevancy Function (WsRF) used for measuring the relevancy ranking of a particular Web service based on client's preferences, and QoS metrics. The proposal in [9] aims at ranking different Web services published by different cloud platforms, taking advantage of PageRank principle. However the service rating based on collaborative feedback service rating has not been explored for service ranking.

3. Proposed Service Ranking Algorithm

Different users can use different scales for rating services. To aggregate them into a single rating, an algorithm is presented in this section. For a naïve user there may be subtle difficulty in rating a service on Quality of Service using objective values and therefore it is very unlikely for him to provide a feedback. However if the parameters are mapped on natural language descriptions i.e. are subjective, a user may be suitably motivated to provide feedback as rating and therefore for service rating an appropriate GUI has been designed as shown in the figure. The users are simply asked to choose to tag a service in five intervals of quality of experience {Best, very good, good, fair, poor} which is relatively simpler to comprehend. However sometimes a user may not be that confident about his capability of rating and therefore user is also asked to judge his confidence in his ratings. This can actually affect the contribution of rating in overall rating. Figure 1 presents the design of the GUI:

Fig. 1 : GUI for User Rating



3.1 Calculating Overall Ranking from the Five User Selected Parameters:

The users can be asked to provide the rating and reviews of used services using above shown GUI. Here the five parameters can have following ranges: Value of Money, Satisfaction Level Trustworthiness, can be mapped upon (Best, Very Good, Good, Fair , Poor) Where 'Excellent' represents the best rating while poor represents the worst and therefore Best can be mapped on 5 on a scale of 1-5 , while Poor can be treated as 1.

In addition the 'Confidence of Rating' gives us a weight parameter for consideration while calculating the overall rating. The overall experience also can be mapped upon 1-5 where this parameter can represent how well the service has met the functional requirement of a client. For the shown example in Fig 1: GUI , the corresponding overall rating may be calculated as : R=[(3.5+3.5+4.5+4.0)/4]x 0.8=(19.5/4) x0.8= 3.9 ~ V. Good

3.2 Calculating Overall Service Rating using Collaborative Filtering

Once an overall rating is calculated for a user, a final rating list has to be aggregated for all the users who have given feedback for services in terms of rating. In this section a method has been given for rating aggregation so as to produce a final ranking for candidate web services using the Offense-Defense rating and ranking method [10].

3.2.1 The Offence-Defense Model of Rating and Ranking

A natural approach in rating is to first rate individual attributes of each participant service. Each service in the chosen pool can have two strengths: first an offensive strength representing how many times it has been rated better over similar services in its pool and a defensive rating representing how strong (or weak) the offence of other services. The Offense-Defense Method has been proposed by Anjela Y. Govan et al. in [10] for ranking sports teams based on their play statistics. It is a model for rating the overall strength of each team relative to the others. While there are numerous factors that might be taken into account, the approach is to characterize "strength" by combining each team's relative offensive and defensive prowess in a non-linear fashion. To compute offensive and defensive ratings authors start with the assumption that if offensive ratings are large then team has greater offensive strength, i.e. the increased capability of winning in a matchup. On the other hand, smaller defensive ratings will correspond to greater defensive strength, i.e., a low defensive rating indicates that it is hard for the opposition to win in a matchup.

However it can very well be employed for ranking services in a pool. The Offence-Defense Method (OD) iteratively produces two ratings for a team i; an offensive o_i and a defensive d_i A simple way to combine them to produce an overall rating is $r_i = o_i/d_i$.

The convergence of the method is guaranteed and is fast. If each web service is treated as a team having a matchup with all potential services as other teams in selection. The method can be effectively applied for ranking. The method of OD rating is as follows: If teams *i* and *j* compete, then let

A = $[a_{ij}]$ be such that a_{ij} is the score that team j generated against team *i* (set a_{ij}= 0 if the two teams did not play each other). Alternately, a_{ij} can be thought of as the number of points that team *i* held team *j* to. When applying for service rating this matrix represents how much better a service has been rated from its opponent service. In other words, depending on how it is viewed, a_{ij} simultaneously reflects relative offensive and defensive capability. To utilize this feature authors have defined the offensive rating of team *j* to be the combination

$$o_j = a_{1j}(1/d_1) + \cdots + a_{nj}(1/d_n)$$

where d_i is the defensive rating of team *i* that is defined to be

$$d_i = a_{1j}(1/o_1) + \cdots + a_{in}(1/o_n)$$

Since o_j's and d_i's are interdependent, these values will have to be determined by a successive refinement technique. Intuitively given A = $[a_{ij}]$:

$$o = A^T \frac{1}{d} \; ; \quad d = A \frac{1}{o}$$

In other words row sums of normalized average distance vector would give us the measure of offensive output while column sum would give us the defensive output.

Rank aggregation is a function which uses several ratings (or ranks) obtained using various models as an input to produce a single rating (or rank) of each team as an output. The simplest aggregation function that can be applied to the Offense-Defense model is $r_i = o_i/d_i$; i.e., the overall rating score of team i is its offensive rating divided by its defensive rating.

3.2.2 Application of OD model for Web Service Ranking

As an example: suppose the following (overall) ratings have been given by a group of four clients to a type (say weather forecast) of services and Overall ratings received by the services are shown in Table 1 & Table 2:

They should be numbered consecutively throughout the text. Equation numbers should be enclosed in parentheses and flushed right. Equations should be referred to as Eq. (X) in the text where X is the equation number. In multiple-line equations, the number should be given on the last line.

Table 1: Ratings Given by Users

Services	User1	User2	User3	User4
S1	Excellent	Very Good	Excellent	-
S4	Fair	-	-	Good
S6	Good	-	Very Good	Very Good
S8	Good	Good	-	Good
S9	-	Good	Excellent	-
S15	-	Poor	Very Good	Excellent

Table 2: Overall Ratings for Services

Service	Ratings
S1	Excellent, Very Good, Excellent
S4	Fair, Good
S6	Good, Very Good, Very Good
S8	Good, good, good
S9	Good, Excellent
S15	Poor, Very Good, Excellent

The above presented table cannot depict correctly the ratings of various users and also some users may be friendly while rating however some may be strict with ratings .To aggregate these differences in scales, a rating aggregation method [11] may be used using normalized rating matrices for all the users $\overline{U}$: Calculating $\overline{U}$ matrices for all users:

Table 3: Normalized matrix for user 1

Service	S1	S4	S6	S8	S9	S15
S1	0	3/9	2/9	2/9	0	0
S4	0	0	0	0	0	0
S6	0	1/9	0	0	0	0
S8	0	1/9	0	0	0	0
S9	0	0	0	0	0	0
S15	0	0	0	0	0	0

Here numerator of (s1,s4) entry means user 1 rated a service 1 three positions above three positions above service 4 . The denominator 9 is the cumulative rating differentials of all ${}^4C_2$ possible matchups between services rated by user1.e.g. difference in ratings between (s1-s4,s1-s6,s1-s8,s4-s6,s4-s8,s6-s8)=(3+2+2+1+1+0)=9 . Similar $\overline{U2}, \overline{U3}, \overline{U4}$ rating matrices can be calculated.

and then calculating the average distance matrix $\overline{U}_{ave}$ can be calculated :

Table 4: Average Distance Matrix

Service	S1	S4	S6	S8	S9	S15
S1	0	0.0833	0.1181	0.0833	0.0278	0.1458
S4	0	0	0	0	0	0
S6	0	0.0635	0	0	0.0357	0
S8	0	0.0278	0	0	0	0.0556
S9	0	0	0.0625	0	0	0.1181
S15	0	0.0714	0.0357	0	0	0

The oValue and dValue of the services is as follows:

Here **oValue** represents the offensive vector calculated as sum of row values for each service i.e. $oValue = \bar{R}_{ave}\,E$ where E is the vector of all ones. The **dValue** represents the defensive vector calculated as sum of column values for each service i.e. $dValue = E^T \bar{R}_{ave}$. Now the rank of each service can be calculated as $R = oValue/dValue$ Using R The final rating list may be calculated as:

Table 5: O-D Values and Overall Rank

Service	oValue	dValue	R=oValue/dValue	Rank
S1	0.4583	0	∞	1
S4	0	0.246	0	6
S6	0.0992	0.216	0.459	4
S8	0.0834	0.0833	1.001	3
S9	0.1806	0.0635	2.844	2
S15	0.1071	0.3195	0.335	5

3.3 Rank Aggregation for Web Service Discovery

While selecting service, a user must have the freedom of choosing the attributes on which the services should be ranked and therefore a scheme has been presented in this section based on various parameters of service ranking. The services may be ranked not only on service rating but on Popularity i.e. invocation frequency as well as cost to be paid while using the service.

3.3.1 Borda Count Method for Rank Aggregation

Borda's method [12] is a “positional" method, it assigns a score corresponding to the positions in which a candidate appears within each voter's ranked list of preferences, and the candidates are sorted by their total score. Given k full lists, Borda’s method assigns a k-element position vector to each candidate (the positions of the candidate in the k lists), and sorting the candidates by the L_1 norm of these vectors. A primary advantage of positional methods is that they are computationally simple and can be implemented in linear time on a RAM (i.e., in O(nk) time for k full lists of n candidates).As an example if there are three ranked lists on popularity, collaborative rating and QoS Rating where 1 represents the best candidate, then Borda Rank is calculated as :

Table 6: Aggregated Rank of Services

Service	OD Rank	Popularity	QoS of Service	Borda Count	Borda Rank
S1	1	2	1	4	1st
S4	6	6	6	18	6th
S6	4	5	5	14	5th
S8	3	3	2	8	3rd
S9	2	1	3	6	2nd
S15	5	4	4	13	4th

3.4 Algorithm for Client Oriented Rank Aggregation

Algorithm : Client Oriented Rank Aggregation
//The presented algorithm calculates the aggregated
//ranks as per user’s chosen parameters for web
//service ranking
Input: User Chosen Parameters for Ranking
Output: Aggregated Ranked List of Services
BEGIN
Step 1.Take Choice of ranking parameters from user
Step 2. As per the chosen parameters, run the ranking algorithms to generate individual ranking lists.
Step 3. Calculate the Borda Count method with respect to given ranked lists.
Step 4. Provide the aggregated ranked list.
STOP

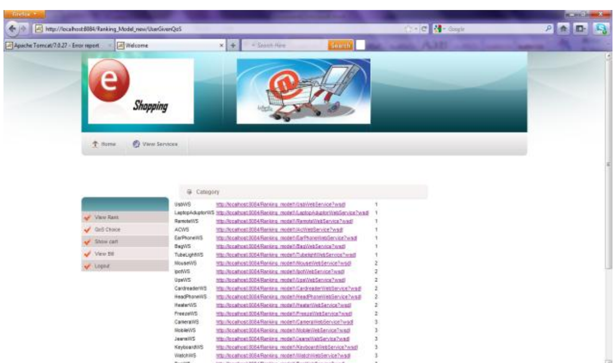
4. Results & Test Cases

For testing the method of service ranking based on Borda Count, a simple test domain for e-shopping had been taken where e- shopping services were ranked based on popularity, QoS of Offered service and User feedback. The aggregated ranked list has been shown in the Table 7 below. It has been observed that Borda Count method suffices well for a small data set of services and is computationally feasible in web service environment for ranking.

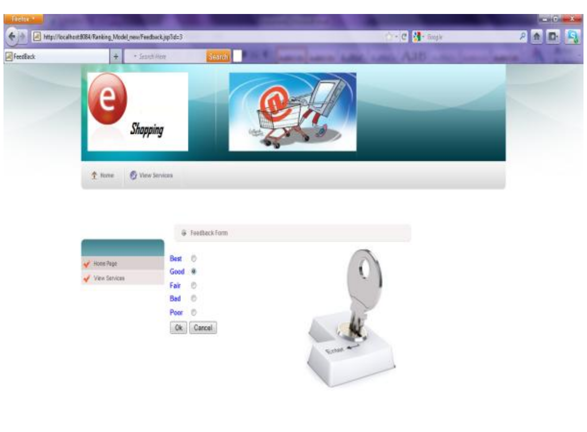
Table 7 : Overall Results for Rank Aggregation

Web Service Name	Rank By Cost	Rank By Feedback (OD Rank)	Rank By Popularity	Rank By Cost & Popularity	Rank By Feedback & Popularity	Rank By Cost, Feedback & Popularity
Ipadws	2	3	4	2	5	2
LaptopWS1	6	2	6	5	6	6
PCWS1	5	1	2	3	2	1
EarphoneWS	12	1	5	10	4	10
SmarttabWS1	6	2	3	5	3	4
IpodWS	8	2	13	7	10	12
UpsWS	7	2	8	6	7	9
HeadphoneWS	11	5	13	11	11	15
LaptopWS3	4	3	7	4	7	7
CameraWS	3	2	10	2	9	8
LaptopWS2	1	2	9	1	8	5
MobileWS	8	1	1	6	1	3
PalmtopWS	9	3	12	9	10	13
KeyboardWS	10	1	11	8	9	11
PCWS2	7	5	14	9	12	14

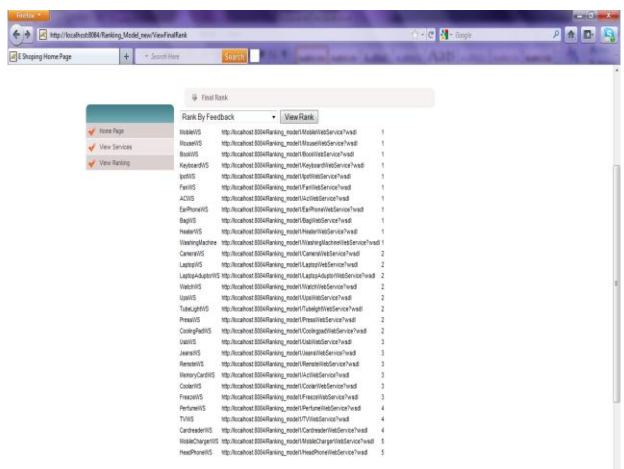
Test Case 1: Ranking based on User Given Quality of Service



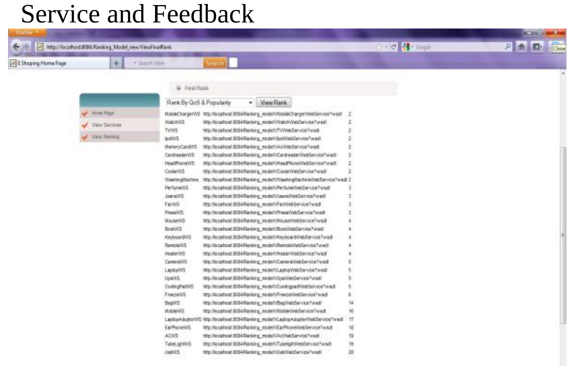
Test Case 2: User feedback of Web Services



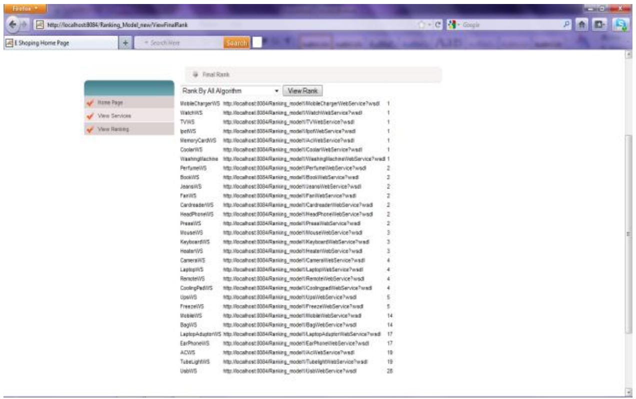
Test Case 3: Rank of Web Services on user Feedback



Test Case 4: Rank of Web Services using Quality of Service and Feedback



Test Case 5: Rank of Web Services by Quality of Service attribute, Popularity and User Feedback



5. Conclusions

An aggregate ranking algorithm based on user feedbacks/service rating has been proposed in this paper which not only calculates an overall rating given by a user but also calculates the overall service ranking for the candidate set of services chosen for a user. This way a user has been benefited from the experiences of his peers. Further there can be other parameters on which a service may be ranked such as Service usages i.e. popularity and QoS parameters. Hence an aggregate ranking algorithm has been proposed for overall ranking based on combination of these parameters which takes into account the fact that user should be able to choose the parameters on which the services are being ranked. This Service Rating & Ranking algorithm has been incorporated in the any service discovery architecture. A Service rating approach has been presented which is user friendly and encourages to a user to give rating for a used service.
In this paper the rating and ranking issues that were handled are:

- To calculate Overall service rating based on subjective ratings supplied by a user through a given interface.
- To do the rate aggregation of rating lists provided by group of users.
- To rank the services based on aggregated rate list using rating vectors.
- To present a method for aggregating ranking lists based on different parameters to provide a final ranking list to a client.

The user benefits significantly from usage of these techniques as he is offered better services which are more appropriate to him in a ranked fashion. This in term allows him a better judgment while choosing the best service out of the pool of services which are functionally viable for him.

References

[1] Web Services www.w3schools.com/webservices/ws_intro.asp.
[2] Service Oriented Architecture http://java.sun.com/developer/Books/j2ee/jwsa/
[3] Universal Description Discovery and Integration (UDDI) version 3.0.2 UDDI SpecTechnical Committee Draft, Dated 20041019,http://www.uddi.org/pubs/uddi_v3.htm,
[4] Le-Hung Vu; Porto, F.; Aberer, K.; Hauswirth, M., "An Extensible and Personalized Approach to QoS-enabled Service Discovery," In proceedings of11th International Database Engineering and Applications Symposium, IDEAS 2007,pp.37-45,doi:10.1109/IDEAS.2007.4318087 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4318087&isnumber=4318075
[5] Z. Chen, C. Liang-Tien, B. Silverajan, L. Bu-Sung, "UX - An Architecture Providing QoS-Aware and Federated Support for UDDI", *Proceedings of ICWS'03.*
[6] F. Emekci, O. D. Sahin, D. Agrawal, A. E. Abbadi, " A Peer- to-Peer Framework for Web Service Discovery with Ranking", *Proceedings of ICWS'04.*
[7] Z. Xu, P. Martin, W. Powley, and F. Zulkernine, "Reputation-Enhanced QoS-based Web Services Discovery," in IEEE International Conference on Web Services (ICWS 2007), pp. 9-13
[8] Eyhab Al-Masri, Qusay H. Mahmoud, "QoS based Discovery and Ranking of Web Services", IEEE, 2007 pp 529-534.
[9] WSRank: A Method for Web Service Ranking in Cloud Environment2011 Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing
[10] Anjela Y. Govan, Amy N. Langville and Carl D. Meyer. Offense-defense approach to ranking team sports. Journal of Quantitative Analysis in Sports. 2009, 5(1), pp. 1-17, 2009
[11] Amy N. Langville, Carl D. D. Meyer, Who's #1?: The Science of Rating and Ranking ,Princeton University Press,2012,pp.176-181.
[12]Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. 2001. Rank aggregation methods for the Web. In Proceedings of the 10th international conference on World Wide Web (WWW '01). ACM, New York, NY, USA, 613-622. http://doi.acm.org/10.1145/371920.372165

Vandan Tewari is currently pursuing Ph.D. program in Computer engineering in RGPV University, M.P., India. She is presently working as Asst. Prof. in Dept. of C.T.A at S.G.S.I.T.S., Indore. Her area of research is Web Services, Personalization, Data Mining and, Databases

Dr. Nirmal Dagdee is a professor of computer science and has about 25 years of teaching experience in various engineering colleges. His areas of active research are data security, SOA and soft computing. He has authored several research papers that are published in reputed journals and conference proceedings. Currently, he is director of S. D. Bansal College Of Technology, Indore, India.

Dr. Aruna Tiwari is working as Assistant professor at Dept. Of Computer Sc. & Engg. at IIT, Indore. She has published more than 25 papers in Journals of repute. Her major area of research are Soft Computing, Data Mining and Databases