Web Mash up Integration to End- Users using QoS and Proficient Assessment

Subbulakshmi.S[1] ,
[1]Assistant Professor,
Sethu Institute of Technology.

Praveen Kumar.B[2]
[2]Assistant Professor,
Velammal Institute of Technology.

ABSTRACT:

"Needs have become more and the search for those needs have become rapid." Everything relies online. Web services are one of the emerging drifts that visualises fundamental to super-rational requests and providing a high fidelity outcome via QoS analysis. Extensive sharing of resources is the vital logic of such web services which enables the users & programmers to exploit data sources available at varied geographical distinctions. For instance, Google Drive is one such web service that offers data portability. The elimination of time old, one to one communication and enhanced many-to-many interactions is accomplished through Web Services, thereby improvising the intellectual standards of common users. Implemented on a Wide area of network applications, web services offer high accuracy, enriched results from existing data.

Keywords:

Scrutiniser, Concurrency, Ontology, Conceptualization.

1. INTRODUCTION

Mashups make use of this famous concept: "*The coolest thing to do with your data will be thought of by someone else*" ie. If you provide data which can be reused this will allow others to develop richer services which you may not have the resources or expertise to develop although service sustainability and serviceability will be needed. A Mashup is an elementary electronic repository that correlates data, methodologies and architecture from two or more originating entities of the web to create new services. Subsequently, this technology implies a simple immediate optimisation With high degree of interactions through API's(Which acts as a liaison between one software and another), data sources to provide the user attractive and accurate results.

2. EXISTING SYSTEM

Data Existence is more vital for imminent projects. The data available here is in a scattered conduct. This makes the clients litter bother about three W's like, what type of data? Where is the data? How to search that? The conventional method had only the QoS rating of Web Services and the selection of the best Mashup was at the jurisdiction of the user online.

This puts the user in a deep state of confusion and a conflict in selecting the best web service among a thousand different services arises. So the conventional probing and presentation slant makes clients little mystified and intricate. For this the details of the service requested should be clear and also securely stored in databases.

3. PROPOSED SYSTEM

*"*All facts at earlier times become outmoded today*"*The challenge in the existing system is that End-to-end service can triumph over by our work where all data from different web services are available in a single Web Mashup. So that drifting for data becomes less. The concurrent access of data is being controlled by applying priority to the user requests. This work mainly deals with the reduction of user confusion in selecting the best Mashup. The selection of the best mashup is done and the best among the available web service is given to the user.

The selection of the best web service is based upon the QoS ratings thus generated. The QoS and Expert generated ratings for these Web services are automatically stored in the respective databases so as to calculate the best service. This implies that a set of vital ingredients in communication are required to have a useful and an effective operational system.

Now our approach exercises the following analyses thereby all Mashup feed are available in the form of Web services and not web applications and thus, all those web services that are published and subscribed, using formal service oriented architecture of existence can be availed with better usability.

Data accumulation

Data accumulation is the process of consolidating and altering the existing data to produce a service that could visualise all possible user requests and providing enriched results. Data accumulation is accomplished in Mashups.

Data Assimilation

Comprehensive data assimilation is necessary so as to provide better coherence of the data that is being used.

4.MQS FRAME WORK:

(1) Mashup Creation

(2)Service Registration

Mashup Creation

"Making and Modernization are vital in a growing technology", likewise creating mashup is a customized one that depends upon composite service participation. To execute the user requests, MQS needs syntactical as well as semantic web services that are available.Here we present a correlated service model based on a SOA frame that depends upon the ontology of similar requests. Web services in this practical scenario are heterogeneous. This implies that it does not rely on single sources; above and beyond it aggregates all web contents, standards, policies and implementations thus providing an unambiguous service automation. It binds the non- functional attributes if the service using QoS .Few amongst them are,

1. Price
2. Availability
3. Reliability
4. Reliability
5. Performance.

These concepts are defined in the Domain ontology. For example school and bus stops are subclasses of place, whereas ontology defines the nature of existence of those entities.

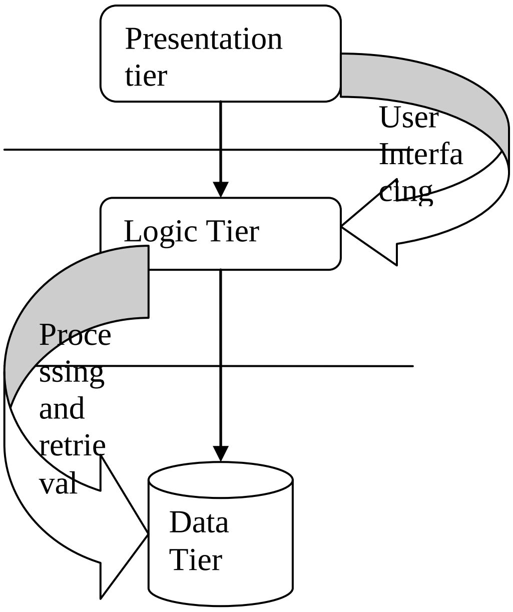

Fig.1 Mashup 3 tier Architecture

5. MQSS ARCHITECTURE

The MQSS is a three tier architecture based system wherein characteristics such as reliability, cost, availability, response time & throughout are validated & then estimated. The framework of the MQSS is illustrated in these 3 tiers where the user queries are fed to the first tier, which is further divided into modules for precision in the upshot of each stage. The modules in all three tiers of architecture include:

-OVA (Optimizer, Validator & Authenticator)
-Ontology Scrutiniser
-Database Hunter.

The modules are highly concrete/ distinct in their pragmatism. This functional independency established by the same, makes the MQSS, "an entity that is capable of performing tasks of various degrees of conceptualisation. Through our effective architecture we achieve best,

- Dataflow
- Code Inspection
- Help
- Feature Usage
- Reframing

Tier 1: The front end of the Mashup QoS Service system would be the user requesting queries by means of interfaces (API's usually). Therein, the control jumps into the first tier of the architecture, the OVA level. OVA as previously referred consists of 3 articulating components namely Optimizer, Validator and Authenticator.

Mashup Optimizers: Optimizers are the most conducive tools that allow user gratification by incessantly monitoring and analysing the various amalgamations of user requests.

Validators: The QoS Validator is an accoutrement which is inclined to validate and check the system personnel request and the final sequel would be optimal and is likely to derive a “better than before” result.

Authenticator: This module relates to the access rights of the service. Data utilization also brings about concerns with respect to data security. Thus the concepts of Access Rights and user Authentication takes centre stage. This component has the jurisdiction over the kind of user who makes use of the service. Registration is this made mandatory such that felonious retrieval of data and service is cut off at the antecedent stages.

Tier 2:

Ontology scrutiniser

Ontology on a general context refers to the theory of existence. It also points to the explicit specification of conceptualization. In Web language terminologies, Domain ontology refers to the entity containing details elaborating the request preference of the current user & the request reference of the predecessors who accessed the respective service previously. The Ontology Web Language (OWL) that is the ontology web language is used in such contexts.

Tier 3:

Reference Repository

QoS provides different parameters from whence or where the system is viewed. This QoS rating given is a system generated one. The parameters such as Cost, Availability, throughput, response time are evaluated based on various mathematical aspects of the service provided.

Nevertheless, the experts’ comments are equally vital. The expert rating is also collected cross referenced with QoS ratings and the consolidated result is given to the user. The upshot thus given to the user is also in the form of a service. This enables the user to make comfortable choice in the likelihood of the complex arena where ‘n’ number of similar services is provided. This reduces the stress on the user to actually choose a better service in an environment that offers similar services.

Since the QoS ratings and expert ratings are stored individually in service databases. These databases contain specific values representing the QoS and expert ratings. Service id is used as the primary key to access and cross reference both the tables / databases. The QoS database consists of attributes like cost, response time, throughput and performance. The QoS Rating table consists of the actual QoS rating which is obtained from the QoS parameters.

The expert database consists of the expert rating for each of the available web service. Both the ratings are then consolidated and the average rating is specified. This enables the user to make a decision at ease given the fact that the most efficient and effective result is provided by the service optimally.

Terminologies Customized:

A Web service is created for Hotel management and transport system separately. Besides, the rating for each is calculated according to the QoS attributes like cost, availability, response time, and throughput. According to the Mashup principle the following relationships are established.

□ Response time □ 1/ performance

□ Throughput □ performance

□ Availability □ performance

□ cost□ performance

We designate the symbols for Mashups as,

□ Response time for mashup is denoted as Mu(R)

□ Throughput for mashup is denoted as Mu(T)

□ Availability for mashup is denoted as Mu(A)

The conventional method of creating web page is now replaced with service created for each web page. The web service is created by use of Service oriented architectural languages mainly XML [Extensible Markup Language] and WSDL [Web Service Description Language]. Each web page is now converted into a web service thereby generating a WSDL service that carries out its own functionality and methodologies.

A database is masqueraded as an INFORMATION BASE (IB) to carry out all searches for determining a rating so that the best web services mashup can be scrutinised and the user query gets optimal result. The result obtained is not through a single and narrow search, besides that it takes deep and widest search and thus gives us an efficient outcome.

6. MQSS FRAMEWORK PROCEDURE

1. Start
2. Get input from the user as a query (in XML tree format).
3. If Umis an authorised individual then login to the Web Service WS1.
   3.1 Check if the query Q1 is true.
   3.2 If true then traverse into Web Service WS2.

   3.3 Mapping of similar entity is done such that the current query Q1 and the another query Q2 are cross referenced.
   3.4 Display Web Service WS3 so as to provide supplemented results.
4. The rating is assessed with all the QoS values such as Mu(R), Mu(T), and Mu(A).
5. End.

7. MASHUP SERVICE REGISTRATION

Every service in the global network should be registered and got the appropriate acceptance and recognition in this competing world. Now our Mashup that we have created should also be registered (i.e.) Reliable. Reliability is an important aspect of service being exploited. The incognito of the reliability of the Mashup service is specified by the rating. If the rating is high then it is reliable such that the Mashup what we have exposed to the user can be taken into account and accepted. This is done to manage security via Trust. There may be several service providers that offers similar services in the internet For example of Person A searches for a venue in India, this can be provides by two service providers:

Google Maps (http://maps.google.com.au) and WhereISMaps (http://www.whereis.com). [1]

Person A would like to ensure that all the services selected in response to his/ her query are the best services compared to other similar services.

However, selection of the best (or trusted) services in the web becomes difficult because of the following:

- There is no essential power to monitor the services. Services operate in an unwrapped environment.
- There is no strong rigid framework for service providers. Clients have to interact with strange services.
- In addition, services may commit to provide a certain level of QoS but may fail to deliver. Thus, a major challenge in building an end-to-end support service for Mashup engineering includes providing a trust framework for enabling the selection of services based on trust parameters.

*SID	SNAME	QoS VAL
101	KPN	8.8
102	SRS	6
103	PARVEEN	5.5
104	SPS	3.3
105	YOHA	2
106	SHIVAM	2.1
107	SSM	2

Fig.3 QoS calculation of each web services

*SID	CNAME	SNAME	QoS Val	EXP Val
101	Sai	KPN	8.8	**?**
102	Dev	SRS	6	**?**
103	Shakthi	PARVEEN	5.5	**?**
104	Shashank	SPS	3.3	**?**
105	Aadarsh	YOHA	2	**?**
106	Shiva	SHIVAM	2.1	**?**
107	Sudhir	SSM	2	**?**
108	Venkat	NPR	3	**?**

Fig. 4 Expert Database

8.DISCUSSION

DATA REDUNDANCY

Data redundancy is normally described as data repetition this redundant data is avoided and thus this mashups provide us a unique content without any duplication.

CONTROLS DATA CONCURRENCY

Data concurrency is referred as multiple accesses of data simultaneously. Here we offer a high degree of concurrency based on priority of user request.

DATA SECURITY

Normally the database used here is named as database hunter where data is in a secure environment so that third party intervention is avoided.

9. RESULTS

The result is produced in such a way that best Web Service is offered as a link to the Web Mashup. This redirects not the Web Applications or to any Websites but to the Mashup which is the combination of three Web Services we've created. The instance cited in the paper also elaborates the final Mashup utilised offering multiple Web Services of difference genres but related contexts at a single go. This Web Mashup consists of three different Web Services viz., Logistics, Tourists, and Hotels.

10.CONCLUSION

"How is Mashup related to cloud computing?"

In this paper we have considered cloud computing to be associated with web mashups. Cloud is a technology that shares resources; the main entity that we need for this sharing of resources is a network where we feed our facts. The network used here is the internet which has inclusive communication and interaction all over the globe. Numerous searches, evaluation, validation based on different criteria are being done by the sharing of resources.

12. FUTURE ENHANCEMENT

At this extent we have our application being done with the help of programming languages like JAVA and Database SQL YOG. In the future this application can be enhanced as Unique Mashup software such as Drop Box, and Google Drive that contain all data searches with a high degree of accuracy and more memory. The following were the findings of the paper.

∗ Eventually, addressing issues related to user requests will be an important step towards commendably supporting user programming of mashup services instead of applications.

∗ The only resource that we would need in such creation of a Mashup is the
"INTERNET".

13.REFERENCES

[1] Yahoo Pipes, http://pipes.yahoo.com/pipes, 2010. IBM Mash up Center, http://www-01.ibm.com/software/info/mashup-center, 2010.

[2] Intel Mashmaker, http://mashmaker.intel.com/web, 2010.

[3] B. Medjahed, A. Bouguettaya, and A.K. Elmagarmid, "Composing Web Services on the Semantic Web," VLDBJ., vol. 12, no. 4,pp. 333-351, 2003.

[4] R. Fikes and N.J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, vol. 2, pp. 189-208, 1971.

[5] A.L. Blum and M.L. Furst, "Fast Planning through Planning Graph Analysis," Artificial Intelligence, vol. 90, nos. 1/2, pp. 281-300, 1997.

[6] J. Rao, P. Ku¨ ngas, and M. Matskin, "Composition of Semantic WebServices Using Linear Logic Theorem Proving," Information Systems, vol. 31, nos. 4/5, pp. 340-360, 2006.

[7] H. Salkin and K. Mathur, Foundations of Integer Programming.NorthHollard, 1989.

[8] B. Medjahed and A. Bouguettaya, "A Multilevel ComposabilityModel for Semantic Web Services," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 954-968, July 2005.

[9] Q. Yu and A. Bouguettaya, "Framework for Web Service QueryAlgebra and Optimization," ACM Trans. Web, vol. 2, no. 1,pp. 1-35, 2008. I.F. Ilyas, W.G. Aref, and A.K. Elmagarmid, "Supporting Top-kJoin Queries in Relational Databases," Proc. 29th Int'l Conf. VeryLarge Data Bases, pp. 754-765, 2003.

[10] C. Boutilier, R.I. Brafman, C. Domshlak, H.H. Hoos, and D. Poole,"CP-Nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements," J. Artificial IntelligenceResearch, vol. 21, pp. 135-191, 2004.

[11] A. Tversky, "Contrasting Rational and Psychological Principlesof Choice," Wise Choices: Decisions, Games, and Negotiations, R.J. Zeckhauser, R.L. Keeney, and J.K. Sebenius, eds., pp. 5-21,Harvard Business School Press, 1996.

[12] B.A. Mellers, A. Schwartz, and A.D.J. Cooke, "Judgment andDecision Making," Ann. Rev. Psychology, vol. 49, pp. 447-477, 1998.

[13] H. Wang, S. Shao, X. Zhou, C. Wan, and A. Bouguettaya, "WebService Selection with Incomplete or Inconsistent User Preferences," Proc. Int'l Joint Conf. Service-Oriented Computing (ICSOC)/ServiceWave, pp. 83-98, 2009.

[14] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman,"Reputation Systems," Comm. ACM, vol. 43, no. 12, pp. 45-48, 2000.

[15] E. Bertino, E. Ferrari, and A.C. Squicciarini, "Trust-X: A Peer-to-Peer Framework for Trust Establishment," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 827-842, July 2004.

[16] H. Skogsrud, B. Benatallah, F. Casati, and M.Q. Dinh, "Trust-Serv: A Lightweight Trust Negotiation Service," Proc. 30th Int'l Conf.Very Large Data Bases, pp. 1329-1332, 2004.

[17] Z. Malik and A. Bouguettaya, "Evaluating Rater Credibility for Reputation Assessment of Web Services," Proc. Eighth Int'l Conf. Web Information Systems Eng., pp. 38-49, 2007.

[18] S. Park, L. Liu, C. Pu, M. Srivatsa, and J. Zhang, "Resilient TrustManagement for Web Service Integration," Proc. Third IEEE Int'l Conf. Web Services, pp. 499-506, 2005.

[19] L. Xiong and L. Liu, "Peertrust: Supporting Reputation-BasedTrust for Peer-to-Peer Electronic Communities," IEEE Trans.Knowledge and Data Eng., vol. 16, no. 7, pp. 843-857, July 2004

[20] S. Nepal, Z. Malik, and A. Bouguettaya, "Reputation Propagation in Composite Services," Proc. Seventh IEEE Int'l Conf. Web Services, 2009.

[21] L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam,and H. Chang, "QoS-Aware Middleware for Web ServicesComposition," IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311-327, May 2004.

[22] Bandura, A. Self-efficacy: Toward a unifying theory ofbehavioral change. *Psychol. Rev.* 8(2):191–215, 1977.

[23] Beckwith, L., Burnett, M., Grigoreanu, V., and Wiedenbeck,S. Gender HCI: What about the software? *Computer*39(11):83–87, 2006.

[24] Cao, J., Riche, Y., Wiedenbeck, S., Burnett, M. andGrigoreanu, V. End-user mashup programming:Through the design lens, In *Proc. CHI*, 1009–1018,2010.

[25] Compeau, D. and Higgins, C. Computer self-efficacy:Development of a measure and initial test. *MIS Quarterly*19(2), 1995.

[26] Cross, N. *Designerly Ways of Knowing*, Springer, 2006.

[27] Grigoreanu, V., Cao, J., Kulesza, T., Bogart, C., Rector,K., Burnett, M., and Wiedenbeck, S. Can feature designreduce the gender gap in end-user software developmentenvironments? In *Proc. VL/HCC*, 149–156, 2008.

[28] Grigoreanu, V., Brundage, J., Bahna, E., Burnett, M.,ElRif, P., and Snover, J. Males' and females' script debuggingstrategies. In *Proc. EUD*, 205–224, 2009.