International Innovative Research Journal of Engineering and Technology
ISSN NO: 2456-1983

International Conference on Emerging Innovation in Engineering and Technology

ICEIET-2017

A Dynamic And Reconfigurable Collaborative Filtering Approach For Qos-Aware Web Service Orchestration

Dr. A.Meiappane [1], G.Bhavani [2], M.Monica [3], A.Aruna [4]

[1] Associate Professor, [2] Students
Department of Information Technology, Manakula Vinayagar Institute of Technology, Puducherry.

ABSTRACT

The Service-Oriented Architecture (SOA) promises a new generation of information systems applications based on a new set of standards for enabling self-describing interoperable Web services. Web service orchestration and choreography are both concerned with the composition of Web services to meet the needs of business processes. There are two important standards for modeling and implementing work-flows and business processes based on Web services: BPEL follows the orchestration paradigm, and WS-CDL covers the choreography. This paper gives a formal methods focused survey of BPEL and WS-CDL languages. Basic language constructs are presented as core building blocks for business processes. From the perspective of composing Web services to execute business processes, the orchestration is a more flexible approach compared to the choreography. Existing work concentrated on web service orchestration aspects using formal models. But it has not considered significant QOS parameter like overall web service turnaround time which is a key parameter to evaluate the quality of web service orchestration. In this paper we are proposing QOS-aware Web service orchestration that translates the QOS requirements of the customers into those of its component Web services using the Collaborative Filtering Approach.

KeyWords: SOA,BPEL,QoS,WebserviceOrchestration

1. INTRODUCTION

Web services are open standard based web applications that interact with other web applications for the purpose of exchanging data. Web services can convert existing applications into web applications. The w3c defines a web service generally as: Web service is a software system designed to support interoperable machine-to-machine interaction over a network. Web services as an interface described in a machine-process format. Web services are services that are made available form a business web server for web user or other web connected program. Provider of web services generally known as Application service providers.

Web services (sometimes called application services) are services (usually including some combination of programming and data, but possibly including human resources as well) that are made available from a business's Web server for Web users or other Web-connected programs. Providers of Web services are generally known as application service provider s. Web services range from such major services as storage management and Customer Relationship Management (CRM) down to much more limited services such as the furnishing of a stock quote and the checking of bids for an auction item. The accelerating creation and availability of these services is a major Web trend.

Web service orchestration is the direction of specific web service business processes by a central controller. The controller, which can also be a web service, coordinates asynchronous interaction, flow control.

Web Services Orchestration has been introduced to address composition and coordination of Web Services. Several languages to describe orchestration for business processes have been presented and many of them use concepts such as long-running transactions and compensations to cope with error handling. WS-BPEL is currently the best suited in this field [6, 7]. However, its complexity hinders rigorous treatment. In this paper we address the notion of orchestration from a formal point of view, with particular attention to transactions and compensations.

The aim of Web Services is to ease and to automate business process collaborations across enterprise boundaries. The core Web Services standards, WSDL and UDDI, cover calling services over the Internet and finding them, but they are not enough. Creating collaborative processes requires an additional layer on top of the Web Services protocol stack: this way we can achieve Web Services composition and orchestration. In particular, orchestration is the description of interactions and messages flow between services in the context of a business process. Orchestration is not a new concept; in the past it has been called workflow.

2. BACKGROUND

BPEL (Business Process Execution Language) is an XML-based language that allows Web services in a service-oriented architecture (SOA) to interconnect and share data [13]. BPEL messages are typically used to invoke remote services, orchestrate process execution and manage events and exceptions. BPEL is often associated with Business Process Management Notation (BPMN), a standard for representing business processes graphically. In many organizations, analysts use BPMN to visualize business processes and developers transform the visualizations to BPEL for execution.

Business Process Execution Language (BPEL) defines a notation for specifying business process behavior based on Web Services [8]. Business processes can be described in two ways: Executable business processes model actual behavior of a participant in a business interaction. Business protocols, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior. The process descriptions for business protocols are called abstract processes. BPEL is used to model the behavior of both executable and abstract processes.

Service-oriented architecture (SOA) is an approach used to create an architecture based upon the use of services [12]. Services (such as restful Web services) carry out some small function, such as producing data, validating a customer, or providing simple analytical services. In addition to building and exposing services, SOA has the ability to leverage these services over and over again within applications (known as composite applications). SOA binds these services to orchestration, or individually leverages these services [9]. Thus, SOA is really about fixing existing architectures by addressing most of the major systems as services, and abstracting those services into a single domain where they are formed into solutions. One of the keys to SOA architecture is that interactions occur with loosely coupled services that operate independently. SOA architecture allows for service reuse, making it unnecessary to start from scratch when upgrades and other modifications are needed. This is a benefit to businesses that seek ways to save time and money.

3. METHODOLOGY

Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future [10]. A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others.

Collaborative Filtering (CF) is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behavior of other users in the system. The fundamental assumption behind this method is that other users' opinions can be selected and aggregated in such a way as to provide a reasonable prediction of the active user's preference. Intuitively, they assume that, if users agree about the quality or relevance of some items, then they will likely agree about other items.

There are other methods for performing recommendation, such as finding items similar to the items liked by a user using textual similarity in metadata (content-based filtering or CBF) [11]. The focus of this survey is on collaborative filtering methods, although content-based filtering will enter our discussion at times when it is relevant to overcoming a particular recommender system difficulty.

A. *Neighbourhood-based approach:*

Most collaborative filtering systems apply the so called neighborhood-based technique. In the neighborhood-based approach a number of users is selected based on their similarity to the active user. A prediction for the active user is made by calculating a weighted average of the ratings of the selected users.

B. *Selecting neighbourhood:*

Many collaborative filtering systems have to be able to handle a large number of users. Making a prediction based on the ratings of thousands of people has serious implications for run-time performance. Therefore, when the number of users reaches a certain amount a selection of the best neighbors has to be made. Two techniques, correlation-thresholding and best-n-neighbor, can be used to determine which neighbors to select. The first technique selects only those neighbors who's correlation is greater than a given threshold. The second technique selects the best *n* neighbors with the highest correlation.

C. *Item-to-Item based approach:*

Another approach Shardanand and Maes consider for Ringo uses the correlation of artists and albums to generate predictions. This approach is simply an inversion of the neighborhood-based approach. Instead of measuring the similarities between people the ratings are used to measure the correlation between items. The Pearson correlation coefficient can again be used as a measure.

International Innovative Research Journal of Engineering and Technology
ISSN NO: 2456-1983

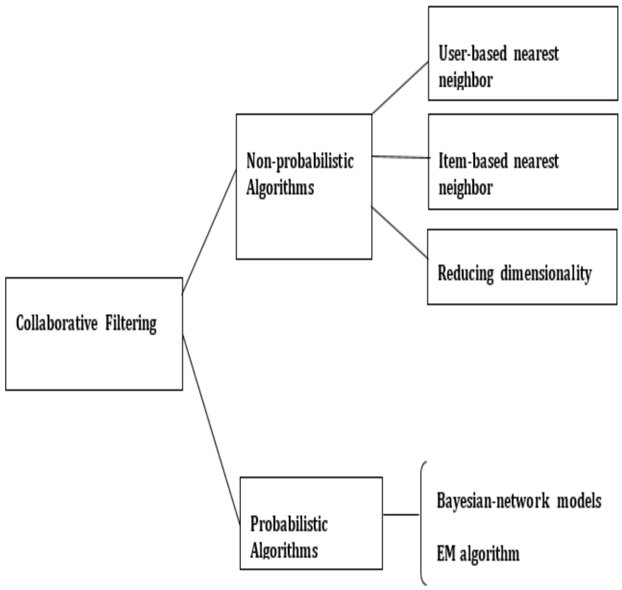

Fig-3.1 : Classification of Collaborative Filtering

D. Dimensionality Reduction:

Dimensionality reduction methods can be used to improve neighborhood-based methods both in terms of quality and in terms of efficiency. In particular, even though pairwise similarities are hard to robustly compute in sparse rating matrices, dimensionality reduction provides a dense low-dimensional representation in terms of latent factors [16]. Therefore, such models are also referred to as *latent factor models*. Even when two users have very few items rated in common, a distance can be computed between their low-dimensional latent vectors. Furthermore, it is more efficient to determine the peer groups with low-dimensional latent vectors.

4. PEARSON CORRELATION COEFFICIENT

In statistics, the Pearson product-moment correlation coefficient (sometimes referred to the PPMCC or PCC or Pearson's *r*) is a measure of the linear correlation between two variables *X* and *Y*, giving a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation. It is widely used in the sciences as a measure of the degree of linear dependence between two variables [15]. It was developed by Karl Pearson from a related idea introduced by Francis Galton in the 1880s.

Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations. The form of the definition involves a "product moment", that is, the mean (the first moment about the origin) of the product of the mean-adjusted random variables; hence the modifier *product-moment* in the name. The absolute values of both the sample and population Pearson correlation coefficients are less than or equal to 1. Correlations equal to 1 or −1 correspond to data points lying exactly on a line (in the case of the sample correlation), or to a bivariate distribution entirely supported on a line (in the case of the population correlation).

Employing PCC, the similarity between two users i and k can be computed based on their observed QoS [14] values on the commonly invoked web services with the following equation:

$$PCC\ (i,\ k) = \frac{\sum_{j\epsilon J}(R_{ij}-\overline{R_i})(R_{kj}-\overline{R_k})}{\sqrt{\sum_{j\epsilon J}(R_{ij}-\overline{R_i})^2}\sqrt{\sum_{j\epsilon J}(R_{kj}-\overline{R_k})^2}}$$

Where,

j = subset of web services that are invoked by both user i and k.

R_{ij} = QoS values of web services j observed by service user i.

R_i, R_j = average QoS values of different web services observed by i and k.

PCC (i,k) is in the interval of [-1,1], where a larger PCC value indicates higher user similarity. By using PCC correlation the similarities can be

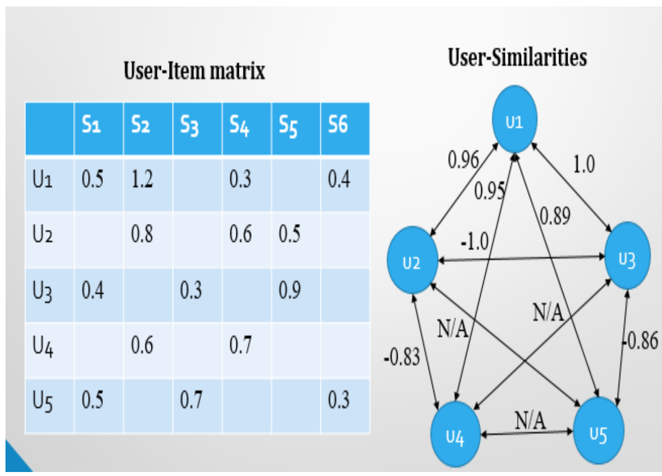

Fig- 4.1 : PCC Correlation with similarities

We compare our proposed approach with the Content-based Web service recommendation approach (rank Web services according to their historical user inter-est relevance and QoS utility), CF-based Web service rec-ommendation approach (rank Web services according to their potential user interest relevance and QoS utility), and Hybrid approach (rank Web services according to the combination of historical user interest relevance, potential user interest relevance, and QoS utility) under the diversity, score, and the ranking measurement defined before.

5. CONCLUSION

This paper has presented QOS-aware Web Service Orchestration that translates the QOS requirements of the customers into those of its component Web Services using the Collaborative Filtering Approach. This approach will help the service consumer to choose the best orchestration of business services by predicting the overall turnaround time required for service orchestration.

REFERENCES

[1] Yong Wang “A Formal Model of QoS-Aware Web Service Orchestration Engine”, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 13, NO. 1, MARCH 2016.

[2] David Jocha “Scalable Architecture for Service Function Chain Orchestration”, IEEE TRANSACTIONS ON NETWORK FUNCTION VIRTUALIZATION, VOL. 4, NO. 2, NOVEMBER 2015.

[3] Carlo Ghezzi “Dependability Assessment of Web Service Orchestrations”,IEEE TRANSACTIONS ON RELIABILITY,VOL. 63, NO.3,SEPTEMBER 2014.

[4] Michael N. Huhns “A Scalable Architecture for Automatic Service Composition”, IEEE TRANSACTIONS ON SERVICE COMPUTING,VOL.7,NO.1, JANUARY 2014.

[5] M.Kalyani “Personalized QOS-Aware Web Service Recommendation Via Collaborative Filtering”,IEEE TRANSACTIONS ON QOS AWARE WEB SERVICES, VOL. 3,DECEMBER 2015.

[6] Active BPEL [Online].http://www.activevos.com/

[7] M. Coabano, E. Denti, A. Ricci, and M. Viroli, “Designing a BPEL orchestration engine based on Respect tuple centres,” in Proc. 4th Int. Workshop Found. Coord. Lang. Softw. Archit. (FOCLASA), 2005, pp. 139–158.

[8] M. Viroli, E. Denti, and A. Ricci, “Engineering a BPEL orchestration engine as a multi-agent system,” in Proc. 4th Int. Workshop Found. Coord. Lang. Softw. Archit. (FOCLASA), 2005, pp. 226–245.

[9] W. Chen, J. Wei, G. Wu, and X. Qiao, “On the move to meaningful internet systems: OTM2008,” in Developing a concurrent service orchestration engine based on event-driven architecture. Berlin Heidelberg: Springer, 2008, vol. 5331, pp. 675–690.

[10] C. Mahmoudi and F. Mourlin, “Web service orchestration driven by formal specification,” in Proc. 7th Int. Conf. Syst. (ICONS’12), 2012, pp. 116–122.

[11] D. Kovaˇc and D. Trˇcek. (2013). A Survey of Web Services Orchestration and Choreography With Formal Models [Online]. http://www.softec.si/pdf/kovac-damjan.survey.pdf

[12] OASIS Committee Specification 01, Reference Architecture Foundation for Service Oriented Architecture Version 1.0 OASIS [Online]. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/ soa-ra-v1.0-cs01.html, December 2012

[13] A. Alves, A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, Sterling, D. König, V. Mehta, S. Thatte, D. van der Rijn, P. Yendluri, and A. Yiu, Web Services Business Process Execution Language Version 2.0 OASIS Committee Draft, May 2006.

[14] A Meiappane, B Chithra, P Venkataesan ” Evaluation of software architecture quality attribute for an internet banking system”, - arXiv preprint arXiv:1312.2342, 2013

[15] D. Wang and K. S. Trivedi, “Modeling user-perceived service availability,” in Proc. 2nd Int. Conf. Service Availability, 2005, ser. ISAS’05, pp. 107–122 [Online]. Available: http://dx.doi.org/10.1007/11560333_10, Berlin, Heidelberg: Springer-Verlag.

[16] A Meiappane, SS Murugan, A Arun, A Ramachandran “Latency of Web service in health care system using GSM networks”, - Machine Learning and Computing (ICMLC), 2010.