International Journal Of Computer Science And Applications Vol. 1, No. 1, June 2008
ISSN 0974-1003

Service Mining Architecture for Web-based Services using Agents

M.MuthuMeena
Department of Computer Science,
Pondicherry University,
Pondicherry, India
Ph: 04132240707
mmuthumeena@yahoo.com

S.K.V.Jayakumar,
Department of Computer Science,
Pondicherry University,
Pondicherry, India.
Cell: 09442253228
skvjey@yahoo.com

P. Dhavachelvan,
Department of Computer Science,
Pondicherry University,
Pondicherry, India.
Cell: 09843293561
pd_chelvoume@yahoo.co.in

ABSTRACT
In the current IT scenario, web service is the fast emerging technology for integrating disparate IT systems and assets. Both academic and industrial organization in business process environment faced a thorny problem, to flexibly modify or update existing components and to connect heterogeneous application to meet the specific requirements of business. There are number of services published and available as runtime usable services in Internet. Lack of semantic base in search mechanisms of UDDI make it difficult for users to find a required web services. In this paper, we are extending web mining concept to service mining having knowledge of service deployed in the registry to automatically discover an efficient service which meets users specific requirements. Agents are used to discover an accurate service which meets specific requirements of user more efficiently using agents than the existing methods in service mining. This work also discussed about the quality attributes with organized set of design related questions which helps an evaluator to analyze the ability of the architecture to meet quality requirements, and provides a brief sample evaluation. ATAM method of software architecture evaluation is used to evaluate the proposed model. The assessment justifies the proposal in terms of the performance attributes such as reliability, availability, modifiability, security and interoperability etc.

Categories and Subject Descriptors:
H.2.8 [Database Management]: Data Mining, **H.3.5 [Information storage and retrieval]:** Online Information Services - Web-based services, **D.2.11 [Software Engineering]:** Software Architecture - Domain Specific architecture, **H.3.3 [Information storage and retrieval]:** Information Search and Retrieval, **D.2.11 [Software Engineering]:** Metrics – Performance Metrics.

General Terms
Design, Performance, Measurement.

Published by Research Publications, Chikhli, India

Keywords
Web services, service mining, agents, software architecture, ATAM

1 INTRODUCTION
Web Services are bringing revolution in World Wide Web by making use of loosely coupled architectures and reusable software components i.e. it can be published, located, and invoked any where from remote location through internet. Web Services are described in XML and communicated using Simple Object Access Protocol (SOAP) over existing HTTP infrastructure and its I/O signatures are provided by means of Web Service Description Languages (WSDL), Universal Description, Discovery and Integration provides a way to publish services and advertise them in it. The increase in web services in registry and its lack of semantics in UDDI make it difficult for users to find a required service to his actual need.

There are many methods has been proposed and implemented service composition. All these techniques use either syntax or semantic matches of service which is stored in repository but, updation of services in UDDI is dynamic in nature. The major challenges in existing technologies addressing for automatic service composition is whether it is possible to search correct service which meets actual need of user's query for composition. Due to the above stated problem, result of service discovery for composite service often fails to satisfy the user. Moreover UDDI is not only intended for accepting service, posting information and responding queries of requesters. It has to take decisions in case of integration with other services.

By considering all these above stated facts, Service Mining concepts is proposed to automatically compose existing atomic services, each meets specific requirements of user. Service mining is similar to web mining concept which is extended from data mining technology.

Web mining is classified into three types namely web content mining, web structure mining, web usage mining [11]. In our concept, we can relate service mining with both web content mining and web usage mining. Unlike web mining, no cleansing of data is required since already services are available in registry as catalog. Web content mining concept is applied to service description file but it is not necessary to create or build index for searching. Service mining mainly deals with service attributes and its properties to discover a right service and additionally our

output is to be a service. Hence there is no need for text or document mining in web mining concept.
In section 2, describes proposed system, agents in service mining with proposed architecture with each component explanation and in section 3, we evaluated the proposed architecture service mining for service composition and in Section 4 analysis the architecture with scenarios and its attributes. Section 5 states the conclusions and Section 6 lists the references.

2 PROPOSED SYSTEM

We have proposed architecture for Service mining for web service composition which allows for exact service discovery for composite process and satisfies accurately user's specific requirements. External user interacts with the user agent by submitting their requests through communication component

This component then submits the request to the message handler, where the request is parsed with respect to the type of the user. If the user interacting the communication component is an agent, then the request are parsed with the help of the ACL/KQML handler, if it is other web service then the SOAP handler takes the responsibility of parsing and if it is a normal user program, then the Natural language (NL) interaction handler is used for converting the request to the format used for discovering the service by the match maker component.

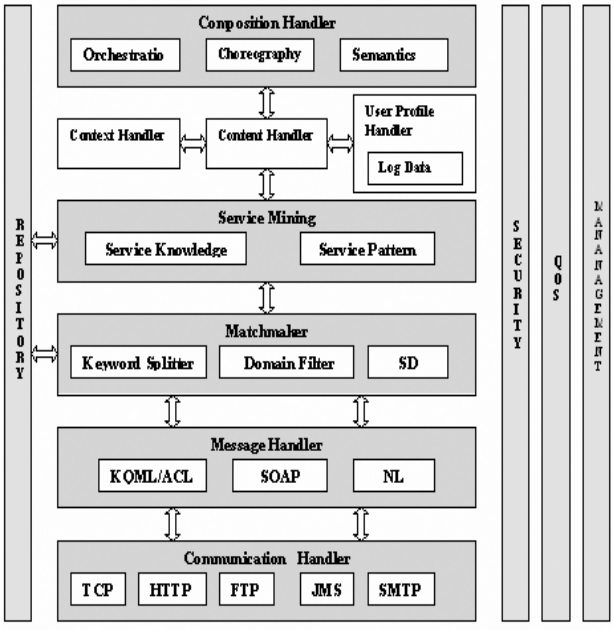

Figure 1. Service Mining Architecture for Web Service Composition.

The message handler also uses the security component for decrypting the input messages form the external user. After processing the request it then passes it to the match maker component.

The match maker agent after receiving the parsed request from the message handler, checks it with the repository which contains services in UDDI. First it will extract the keywords from the requests and then it matches with domain ontology and filter services based on domain. For this, it requests the information from domain ontology. The service discovery (SD) component then selects the best matched results and submits them to service mining layer. Subsequent to services obtained from match maker, service mining agent tends to ensure the accuracy of selected service based on knowledge of service in it, for this it request the information from context and content handler which uses again domain ontology as source for it. User profile handler maintains the service usage preference of different types of users. Service mining contains successful composition patterns and it uses service usage of users from user profile handler. The user preference handler is also updated with the preferences of the recently accessed request.

If the repository does not contain the request made by the user, then the match maker component uses its domain filter component for discovering the service from the UDDI registry. The whole process as described above is repeated. The message handler after receiving the results of the selection process from the match maker converts the results into a format as required by the user. It uses the security component for encrypting the results if necessary and passes them to the communication component which then sends back to the user. They can search the services based on the available contexts. They will obtain context details from the user information (i.e. user profile) or by using other services and moreover, agents will adjust according to user preferences based on the historical tasks.

3 EVALUATION OF PROPOSED ARCHITECTURE TYPESET TEXT

The proposed architecture is evaluated by the Architecture Tradeoff Analysis Method (ATAM) [12], [13], [14]. All the scenarios corresponding to each applications of the service mining are listed and evaluated based on quality attributes. Before evaluation, we have to identify the different stakeholders involved in the system. The Stakeholders are developers, maintainers, evaluation team, customer representative, architecture team, business analysts, end-user, operator, tester, system administrator. The evaluation team presents ATAM to above stated stakeholders with brief explanation of steps and techniques followed for analyzing and eliciting utility tree generations, architectural approaches and scenario mapping and result of evaluation identified with stakeholders prioritize, risks, tradeoff, response, response measure.
Next, goal of architecture is identified and based on the analysis of stakeholders need and present the business goals. The utility tree provides a mapping between the quality attributes that the architecture to meet discussed in business driver to the corresponding scenarios. In this tree, root node is "utility" and second level node are various quality attributes of architecture and third level follows attribute concerns and in final fourth level represents scenario with pair of ranking to represent the prioritize of nodes in leaves. The scenarios are prioritized relative to each other using ranking pairs of (High, Medium, and Low). It would be (H, H) (H, M) (H, L) (M, H) (M, M) (M, L) (L, H) (L, M) (L, L). The first letter denotes degree of importance to system and second letter denotes degree of difficulty in achieving it. The scenario prioritization is included in table 2.
According to SEI [12], the suitability of the architecture proposed is determined by quality attribute requirements that are important to stakeholders of system. The ATAM relies on elicitation of

quality attribute scenarios. The Scenarios chosen for evaluation of the architecture are given below in the table 1:

4 ARCHITECTURE ANALYSIS

Architecture Analysis is to reveal enough information about the proposed architecture to identify it risks, non-risks, tradeoffs, sensitive points in the design phase itself rather than later phase. This method is not meant with precise and detailed evaluation of architecture quality attributes with its numerical value[12]. In the evaluation phase, we identified risks associated with architectural decisions and their effects on quality attributes. Table 2 shows with defined scenarios and their risks, tradeoff, scenario prioritization. The Architecture Tradeoff Analysis Method (ATAM) reveals that proposed architecture for service mining agent with information's like risk, tradeoff points, response measure, and priority in design phase. The most important tradeoff and sensitive points identified in proposed system is highly sensitive to performance of the system and in next phase of implementation care should be taken to control sensitiveness of system performance. In addition to the tradeoffs and sensitive points, several risks associated with architecture are also identified using ATAM.

5 CONCLUSION

In this paper we addressed actual need of web service mining in registry where vast amount of services are publishing and updation of services are likely to be happen in it. This paper is proposed to develop a service mining agent for web service composition. The various steps in the service mining architecture development phases are explained in this paper. The proposed architecture is evaluated using Architecture Tradeoff Analysis Method (ATAM) which allows identifying risks, non-risks, sensitive points, tradeoffs, priority of each scenario to system previously in design phase.

6 REFERENCES

[1] Liang, Q.A., Miller, S., Chung, J.-Y., 2005. "Service mining for Web service composition", IEEE International Conference on. Information Reuse and Integration2005, Volume, Issue, 15-17 Page(s): 470 – 475, Aug. 2005.

[2] Qianhui Liang, Jen-Yao Chung, Steven Miller, Ouyang Yang, 2006." Service Pattern Discovery of Web Service Mining in Web Service Registry-Repository", IEEE International Conference on E-Business Engineering, Pages: 286-293, 2006.

[3] Qianhui Liang J. Y. Chung, 2007 "Analyzing Service Usage Patterns: Methodology and Simulation" IEEE International Conference on e-Business Engineering (ICEBE'07) pp. 359-362.

[4] Zheng, George, Bouguettaya, Athman, July 2007. "A Web Service Mining Framework", IEEE International Conference on Web Services, Volume, Issue, 9-13 Page(s):1096 – 1103.

[5] Mohsen Rouached , Walid Gaaloul , Wil M. P. van der Aalst , Sami Bhiri and Claude Godart,"Web Service Mining and Verification of Properties: An Approach Based on Event Calculus", Cooperative Information Systems (CoopIS) 2006 International Conference, Springer Berlin / Heidelberg, Volume 4275/2006, Pages 408-425,2006.

[6] Walid Gaaloul, Sami Bhiri, Claude Godart, 2006. "Research Challenges and Opportunities in Web Services Mining" ICWS'06 September 2006.

[7] Schahram Dustdar, Robert Gombotz,2006." Discovering web service workflows using web services interaction

[8] mining", International Journal of Business Process Integration and Management ,Vol. 1, No.4 pp. 256 - 266.

[9] Chhabra, Manish Lu, Hongen La Trobe University, Australia, 2007."Towards Agent Based Web Service ", International Conference on Computer and Information Science, On page(s): 93-99.

[10] Bin Li; Xiao-Yan Tang; Jian Lv, 2005 ."The Research and Implematation of Services Discovery Agent in Web Services Composition Framework " Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Volume 1, Issue , 18-21 Page(s): 78 – 84, Aug. 2005.

[11] Vinoski, S.2003." Service discovery 101", IEEE Internet Computing Volume 7, Issue 1, Page(s): 69 – 71 Jan/Feb 2003.

[12] Cooley, R. Mobasher. B, Srivastava, J. 1997. "Web mining: information and pattern discovery on the World Wide Web" Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence, Volume, Issue, 3-8 Page(s):558 – 567.

[13] Bass, Len; Klein, Mark; & Moreno, Gabriel.2001. "Applicability of General Scenarios to the Architecture Tradeoff Analysis Method". Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, http://www.sei.cmu.edu/publications/documents/01.reports/01tr014.html.

[14] Bass, L.; Clements, P.; & Kazman, R. Software Architecture in Practice. Reading, MA: Addison-Wesley, 1998.

[15] Muhammad Ali Babar, Liming Zhu, Ross Jeffery, 2004" A Framework for Classifying and Comparing Software Architecture Evaluation Methods", Australian Software Engineering Conference (ASWEC'04), p. 309-315.

[16] Web Services Architecture, http:// www.w3.org/TR/ws-arch/.

ISSN 0974-1003

Table 3. Scenario Identification

SCENARIO NUMBER	SCENARIOS	QUALITY ATTRIBUTES
Scenario 1	User request is processed double (including duplicate request)	Reliability
Scenario 2	Response time of the system should be less	Performance
Scenario 3	New service provider with own service interface is added.	Interoperability
Scenario 4	New architecture component / business partner is added.	Modifiability
Scenario 5	Ability to accommodate new requirements	Adaptability
Scenario 6	Provider fails after advertise service.	Availability
Scenario 7	Increase number of users for a service.	Scalability
Scenario 8	Ability to accommodate new technology.	Maintainability
Scenario 9	Communicate execution status / Common look and feel GUI.	Usability
Scenario 10	Policies of services changes or modified.	Flexibility
Scenario 11	Payment processing functions should be secure.	Security
Scenario 12	User request is satisfied by discovering a correct service.	Correctness

Table 2. Analysis of Scenarios in Architecture

Quality Attributes	Scenarios	Goal	Risks	Tradeoffs	Response Measure	Priority
Reliability	S 1	User satisfaction	-	Provides Idempotent endpoints- performance overhead and complexity in implementation	100% transaction	(H,M)
Performance	S2	User satisfaction	No control over execution of external service providers.	Provide interoperability & loose coupling- latency of request increases	System replies to user within few seconds.	(H,H)
Interoperability	S3	Ease of Integration	-	Interact with external service interface- extensive XML processing – performance overhead.	Not more than 10 person per day	(H,H)
Modifiability	S4	Ease of up gradation	-	-	Not more than 1 person per day	(M,L)
Adaptability	S5	User satisfaction	No assurance of communication	Wireless communication - may affect reliability.	Not more than 5 person per day	(L,M)
Availability	S6	Provide user reliable service	No control to prevent authorized provider to advertise service	Monitoring providers and checks for consumer time's out- performance and reliability overhead.	Not more than 10 person per day	(H,H)
Scalability	S7	User satisfaction	-	Centralized DB to handle multiple instances – poor response time.	Not more than 1 person per day	(H,H)
Maintainability	S8	Ease of up gradation	Less impact in system	-	1 spiral	(L,H)
Usability	S9	User satisfaction	-	Increases users learn ability of system- attackers enters and security of system is in critical.	Not more than 3 person per day	(H,M)
Flexibility	S10	Provider satisfaction	-	Monitoring changes in providers functionality and updates in DB - performance overhead	Not more than 1 person per day	(H,H)
Security	S11	Assure security.	Digital Certificate exchange is done with care.	Adds performance overhead and adds complexity in implementing digital certificate authentication.	Not more than 15 person per day	(H,M)
Correctness	S12	User satisfaction	Interfaces are too fine grained.	Adds performance overhead in transmission & processing of many small messages to complete a task.	Not more than 3 person per day	(H,M)