Peer-to-Peer Networks Approach for Discovery of Decentralized Service

Mr. Kishor Janu Pawar[1], Prof.Ankita Gautam[2]
[1]Dept. of Computer Science, Shreeyash College of Engineering &Technology, Aurangabad, DBAMU
[2]Dept. of Computer Science, Shreeyash College of Engineering &Technology, Aurangabad, DBAMU

Index Terms **About Peer-to-Peer Networks, Discovery of Decentralized Service, Web Service Discovery with Ranking, Owl Web Otology Language Reference Web Services Dynamic Discovery.**

I. INTRODUCTION

With the increasing need for networked applications and distributed resource sharing, there is a strong incentive for an open large-scale service infrastructure operating over multidomain and multi-technology networks. Service discovery, as an essential support function of such an infrastructure, is a crucial current research challenge. Although a few survey papers have been published on this subject, our contribution focuses on comparing and analyzing key discovery approaches in the context of large-scale and multidomain networks. The comparison is conducted based on a set of well-defined criteria, leading to a selection of few approaches that can serve as the guide in designing a global service discovery system for large-scale and multi-technology networks.

In this article, we have provided an in-depth analysis of various service and resource discovery approaches. In order to provide a formal structure to our discussion and guide the reader through this document, we first defined a set of evaluation criteria that are important to any large-scale, multidomain discovery approach and outlined their related aspects. By analyzing each approach against these criteria, we brought out the strengths and weaknesses of each approach against the goal of an Internet-scale multi-domain service discovery architecture. Based on this analysis, a comparison study of these approaches is conducted and a brief description of our findings is presented in the Technology Comparison Section. Since this survey has revealed that no single approach meets all of our defined objectives, we have focused instead on choosing those approaches that could be combined to build the desired service-discovery architecture. In our technology selection, we have focused on three particular approaches: Web Services, INS/Twine and structured P2P systems as potential components of such an architecture. They are chosen for their scalability, standardization, performance, effectiveness, security aspects, system independence and implementation support. In the time since it was originally conducted, this survey has served as the groundwork in designing a new large-scale discovery mechanism that is multi-domain, multi-technology and aims to unite multiple diverse service-discovery platforms [79]. This platform-independent, extensible approach enables cross-technology and cross-domain service discovery, supports domain-level access control for discovery operations, and has bounded query lookup time. During the design process, the set of criteria and requirements, as well as the variety of innovative approaches that we have encountered in our survey has helped us immensely in all aspects of design and implementation.

II. CRAWLING MULTIPLE UDDI BUSINESS REGISTRIES

V UBRs for finding services of interest. As the number of Web services increase, the success of businesses will depend on service discovery and performance time when searching multiple UBRs. Our experiments demonstrate that building a crawler and a centralized repository for Web services is inevitable. For future work, we plan to extend our current framework to include a ranking mechanism that outputs desired services of interest within top results and therefore, rendering the discovery process to become more efficient.

III. AN ANALYSIS OF THE SKYPE PEER-TO-PEER INTERNET TELEPHONY PROTOCOL

Skype is a peer-to-peer VoIP client developed by KaZaa in 2003. Skype claims that it can work almost seamlessly across NATs and firewalls and has better voice quality than the MSN and Yahoo IM applications. It encrypts calls end-to-end, and stores user information in a decentralized fashion. Skype also supports instant messaging and conferencing. This report analyzes key Skype functions such as login, NAT and firewall traversal, call establishment, media transfer, codecs, and conferencing under three different network setups. Analysis is Skype is the first VoIP client based on peer-to-peer technology. We think that three factors are responsible for its increasing popularity. First, it provides better voice quality than MSN and Yahoo IM clients; second, it can work almost seamlessly behind NATs and firewalls; and third, it is extremely easy to install and use. We believe that Skype client uses its version of STUN [1] protocol to determine the type of NAT or firewall it is behind. The NAT and firewall traversal techniques of Skype are similar to many existing applications such as network games. It is by the random selection of sender and listener ports, the use of TCP as voice streaming protocol, and the peer-to-peer nature of the Skype network, that not only a SC traverses NATs and firewalls but it does so without any explicit NAT or firewall traversal server. Skype uses TCP for signaling. It uses wide band codecs and has probably licensed them from GlobalIPSound [10]. Skype communication is encrypted. The underlying search technique that Skype uses for user search is still not clear. Our guess is that it uses a combination of hashing and periodic controlled flooding to gain information about the online Skype users. Skype has a central login server which stores the login name and password of each user. Since Skype packets are encrypted, it is not possible to say with certainty what other information is stored on the login server. However, during our experiments we did not observe any subsequent exchange of information with the login server after a user logged onto the Skype network.

IV. WEB SERVICES DYNAMIC DISCOVERY (WS-DISCOVERY)

This specification defines a multicast discovery protocol to locate services. By default, probes are sent to a multicast group, and target services that match return a response directly to the requester. To scale to a large number of endpoints, the protocol defines the multicast suppression behavior if a discovery proxy is available on the network. To minimize the need for polling, target services that wish to be discovered send an announcement when they join and leave the network.

V. OWL WEB ONTOLOGY LANGUAGE REFERENCE

Current service description and composition approaches consider simplistic method invocation. They do not accommodate ongoing interactions between service Providers and consumers, nor do they support descriptions of legal protocols of interactions among them. We propose richer representations which enable us to capture more of the semantics of services than current approaches. Further, we develop algorithms by which potential problems in service compositions can be detected when services are configured, thereby leading to superior execution of composed services.

Besides Web services, the work described here touches upon extensive bodies of research on the semantic Web, workflow modeling, protocols, and agent-based techniques. We lack the space to review these in detail (but some were cited in the above discussion), but mention representative work here. Semantic Web. DARPA Agent Markup Language (DAML) [9] enables the creation of ontologies in the description of specific Web sites. DAML-S [2] is a Web service ontology from the semantic Web community [4]. DAML-S provides a core set of markup language constructs for describing the properties and capabilities of Web services. Some emerging approaches add structure to service descriptions through the use of ontologies, e.g., [17]. Workflow and process modeling. Klein & Bernstein develop a richer approach for describing and indexing services based on process models [12]. Verharen develops a contract specification language, CoLa, to specify transactions and contracts [19]. Verharen's approach captures obligations involving actions, but does not allow the obligations to be manipulated dynamically. This is a possible line of extension for the present work. Protocols. Barbuceanu and Fox [3] develop a language, COOL, for describing coordination among agents. Their approach is based on modeling conversations through FSMs, where the states denote the possible states a conversation can be in, and the transitions represent the flow of the conversation through message exchange. They try to handle exceptions through error recovery rules. HP's Conversation Definition Language [8] has similar goals to ASDL. CDL provides an XML schema for defining valid sequences of documents exchanged between Web services. Like ASDL, it uses conversations to model the externally visible interaction model of the Web service.≤ Agent-based exception handling. Klein & Dellarocas exploit a knowledge base of generic exception detection, diagnosis, and resolution expertise [13]. Specialized agents are dedicated to exception handling. Our approach is complementary, since it applies at design time and does not require extensive intelligence. 5.3 Directions This work opens up several interesting directions for research, some of which we are pursuing actively. On the practical side, we are working on our prototype to enhance its representational capabilities for services. On the theoretical side, it will be helpful to explicitly incorporate extended transactions in our models to capture richer constraints on service behavior.

VI. A PEER-TO-PEER FRAMEWORK FOR WEB SERVICE DISCOVERY WITH RANKING

Current Web service discovery methods are based on centralized approaches where Web services are identified based on service functionality. Examples of service func-tionality include car rental, hotel booking and book selling. Since higher level Web services are increasingly composed in terms of lower level Web services, it is important that service discovery not only be based on service functionality but also be based on process behavior, i.e., how a service functionality is served. Furthermore, centralized approaches to service discovery suffer from problems such as high operational and maintenance cost, single point of failure, and scalability. Another issue that has not been considered in current Web service discovery paradigms is the issue of trust and quality of service of the service provider.

We, therefore, propose a structured peer-to-peer framework for Web service discovery in which Web services are located based on both service functionality and process behavior. In addition, we integrate a scalable reputation model in this distributed peer-to-peer framework to rank Web services based on both trust and service quality.

In this paper, we propose a structured peer-to-peer framework for Web service discovery in which Web services are located based on both service functionality and process behavior. We represent the process behavior of the Web services with finite automata and use these automata for publishing and querying the Web services within the system. Our model is scalable and robust due to the underlying peer-to-peer architecture. Web services can join and leave the system dynamically. We also propose an efficient and scalable reputation model based on sketch theory. Thus the returned services are ranked based on the trust and quality ratings of the services using the proposed reputation model. As a future work, we plan to use Petri Nets for represent-ing the process behavior as they are more expressive than finite automata. Our goal is to locate the services based on their Petri Nets, while pushing some simulation and veri- fication tasks to the discovery phase, such as locating the Web services which do not result in a deadlock.

VII. MODEL-BASED VERIFICATION OF WEB SERVICE COMPOSITIONS

In this paper, we discuss a model-based approach to verifying Web service compositions for Web service implementations. The approach supports verification against specification models and assigns semantics to the behavior of implementation model so as to confirm expected results for both the designer and implementer. Specifications of the design are modeled in UML (Unified Modeling Language), in the form of message sequence charts (MSC), and mechanically compiled into the finite state process notation (FSP) to concisely describe and reason about the concurrent programs. Implementations are mechanically translated to FSP to allow a trace equivalence verification process to be performed. By providing early design verification, the implementation, testing, and deployment of Web service compositions can be eased through the understanding of the differences, limitations and undesirable traces allowed by the composition. The approach is supported by a suite of cooperating tools for specification, formal modeling and trace animation of the composition workflow.

VIII. RESULT

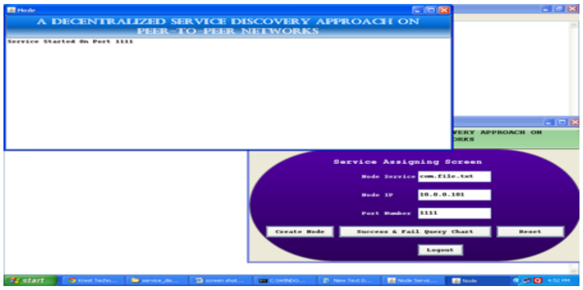

Figure-a: *Admin creating a node (server) for downloading txt files with ip 10.0.0.101 and port as 1111*

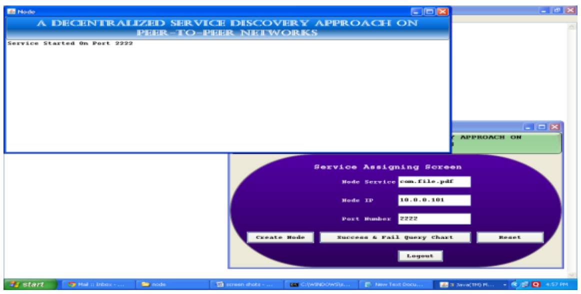

Figure-b: *Admin creating a node (server) for downloading pdf files with ip 10.0.0.101 and port as 2222*

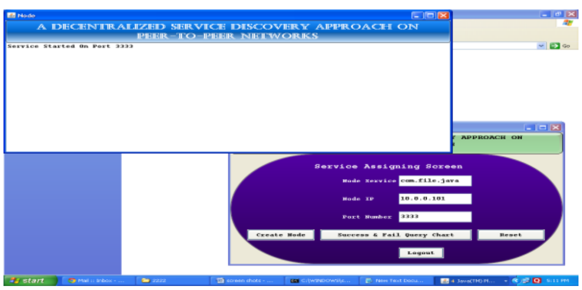

Figure-c: *Admin creating a node (server) for downloading java files with ip 10.0.0.101 and port as 3333*

IX. CONCLUSION

Service discovery is a critical component in service-oriented computing. Over recent years, peer-to-peer-based service discovery has attracted researchers' attention after the deficiencies of centralized service discovery are identified. This paper has proposed Chord4S, a peer-to-peer-based approach for decentralized service discovery. To improve data availability, Chord4S distributes the descriptions of functionally equivalent services. Chord4S supports QoSaware service discovery and service discovery with wildcard(s). An efficient routing algorithm is developed to facilitate queries of multiple functionally equivalent services. Chord4S is scalable, reliable, and robust due to the enhanced peer-to-peer architecture. Experimental results demonstrate that Chord4S can achieve high data availability and efficient query of multiple functionally equivalent services with reasonable overhead.

In the future, integration of semantic information of services into Chord4S using popular tools, such as Petri Net and WSMO, will be investigated in order to increase the flexibility and accuracy of the service discovery.

REFERENCES

[1] American Industrial Classification Scheme (NAICS) codes,‖ http://www.naics.com/.
[2] Universal Standard Products and Services Classificatio(UNSPSC), ‖http://www.unspsc.org/, 2012.
[3] R. Ahmed, N. Limam, J. Xiao, Y. Iraqi, and R. Boutaba, "Resource and Service Discovery in Large-Scale Multi-Domain Networks,‖ IEEE Comm. Surveys and Tutorials, vol. 9, no. 4, pp. 2-30, Oct.-Dec. 2007.
[4] E. Al-Masri and Q.H. Mahmoud, "Crawling Multiple UDDI Business Registries,‖ Proc. 16th Int'l Conf. World Wide Web (WWW '07), pp. 1255-1256, 2007.
[5] D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, and P. Plebani, "PAWS: A Framework for Executing Adaptive Web-Service Processes,‖ IEEE Software, vol. 24, no. 6, pp. 39-46, Nov./Dec.2007.
[6] S. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol,‖ Proc. IEEE INFOCOM, pp. 1-11, 2006.
[7] J. Beatty, G. Kakivaya, D. Kemp, T. Kuehnel, B. Lovering, B. Roe, C. St.John, J. Schlimmer, G. Simonet, D. Walter, J. Weast, Y. Yarmosh, and P. Yendluri, "Web Services Dynamic Discovery (WS-Discovery),‖ http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf, 2005.