User-centric Smart Services

Karuna P Joshi[1], Yelena Yesha[1], A. Ant Ozok[2] and Yaacov Yesha[1]

[1]Computer Science and Electrical Engineering Department
[2]Information Systems Department
University of Maryland, Baltimore County, MD, USA

Abstract

In this position paper we describe our vision for the next generation of IT services. Services will be automatically discovered, procured and integrated with the service consumer's technical environment. This whole process will be determined by the policies defined by the consumer and will be transparent to the consumer.

1 Introduction

It is increasingly being recognized that the Internet in its current state is server-centric where the content, control and performance lies with the servers running the websites. The onus is on the consumer to search for the required data/information across various websites and combine it as needed. The next generation of the Internet, termed Smart Internet [2], advocates a user-centric model for the web instead of the server-centric model that exists today. The Smart Internet addresses the technological gaps due to lack of integration, individualization, user control, collaborative services that exist in the current rendition of the Internet.

In this paper, we articulate a similar vision for services, specially virtualized services on the cloud. Virtualized service models are emerging and IT development and maintenance which was previously either in-house or outsourced is being replaced by this new delivery model where businesses purchase IT components, like software, hardware or network bandwidth, as services from providers distributed globally. In such scenarios, multiple providers often collaborate to create a single service for an organization. This model has also been termed Service Virtualization [1]. Virtualization implies that the service with which an end customer interacts may be composed of many others, and each service in turn could depend on backend applications (database, web server) and resources (storage, bandwidth, CPU). It could be delivered remotely to the consumer via a computing grid or cloud. The service, in effect, is *virtualized on the cloud*. This virtualized model of service delivery potentially allows easier service customization, better resource utilization and greater responsiveness on part of the service providers. In this model, the service is acquired through the "on demand" pull technology. This is true of services that are purely IT in nature (e.g. Software as a Service (SaaS) or Infrastructure as a Service (IaaS)), as well as services that are IT enabled (ITeS) but involve human contact (e.g. those provided via contact centers). It is possible in these scenarios that neither the hardware infrastructure, nor the software; and not even the people running these services belong to the organization that uses the service.

Creation of a user-centric model for virtualized services presents new challenges. While the concept of 'Service on demand', i.e. a user requesting service or its components when needed, is promising; there appears to be no rush from organizations to adopt this new model. One of the key barriers is the lack of infrastructure to enable automatic management, procurement and integration of services. Smart internet is essential to realize the full benefit of the virtualized delivery model. In this paper we describe our vision of how services will be more automatically managed in a Smart Internet environment. We review related work in this area in section 2 and detail the proposed service flow in section 3.

2 Related Work

A user-centric model of the Internet mandates the capability of individualizing or personalizing websites for each user based on their preferences, traversal patterns, functional domain, services consumed etc. Personalization of web has been extensively researched. Researchers have tried to enhance it by improving upon Internet's searching capabilities. Joshi and Jiang [13] had proposed an algorithm to cluster search results to provide users with a personalized view of their web queries. Stanford NLP group [14] have developed PageRank algorithm to decide the ranking of web pages returned after a search. More recently, Balke and Wagner [14] have proposed an algorithm featuring the expansion of service requests by user-specific demands to enable personalized selection of web services. However, most of these approaches have been based on the current server-centric model on the Internet.

Researchers have also concentrated on developing methodologies for virtualized services. Papazoglou and Van Den Heuvel [3] have proposed a methodology for developing and deploying web services using service oriented architectures. Their approach, however, is limited to the creation and deployment of web services and does not account for virtualized environment where services are composed on demand. Providers may need to combine their services with other resources or providers' services to meet consumer needs. Other methodologies, like that proposed by Bianchini et al. [4], do not provide this flexibility and are limited to cases where a single service provider provides one service. Zeng et al. [5] address the quality based selection of composite services via a global planning approach but do not cover the human factors in quality metrics used for selecting the components. Maximilien and Singh [6] propose an ontology to capture quality of a web service so that quality attributes can be used while selecting a service. Their ontology is limited by the fact that it considers single web services, rather than service compositions. Black et al. [7] have proposed an integrated model for IT service management. Their model is limited to managing the service from the service provider's perspective.

In a virtualized service-oriented environment, consumers and providers need to be able to exchange information, queries, and requests pertaining to the data and policies with some assurance that they share a common meaning. One possible approach to this issue is to employ Semantic Web techniques for modeling and reasoning about services related information. *Semantic Web* is an enhancement of the World Wide Web that deals primarily with data instead of documents. It enables data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as Resource Description Framework (RDF) [10] and Web Ontology Language (OWL) [9] for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions. Web Services Description Language (WSDL) [8] is an XML based language that provides a model for describing web services. It defines services as collections of network endpoints, or ports operating on messages containing either document-oriented or procedure-oriented information. Business Process Execution Language (BPEL) [11] is a standard executable language for specifying business process behavior based on web services. Its messaging facilities depend on WSDL. Semantic web technologies can be used to provide common semantics of service information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each other's data and services effectively.

3 Service on Demand

Existing methodologies for designing and deploying web based services put the control in the hands of the service providers who plan and design the service, decide its composition, delivery mode and then wait for consumer request of the service. Services are mainly developed by using the semantic web technologies. While semantic web languages are easier to read compared to computer languages, like C or Java; they still require the consumers to possess technical expertise in semantic web languages to be able to consume the services.

We envision that with the development of the Smart Internet and further adoption of virtualized services and cloud computing models, the next generation of service lifecycle will be fully automated. It will be transparent to the consumer and services needed will be automatically identified by the service environment, discovered and procured in the cloud and seamlessly integrated back with the consumer's service environment. Figure 1 illustrates the flow of information in automated services. We further detail the automated service environment in the next sections.

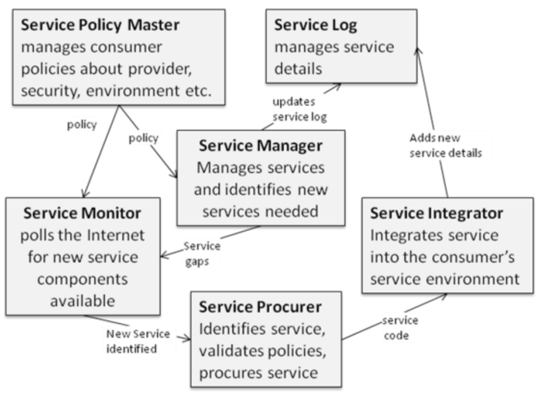

Figure 1: Smart services will consist of service manager, service monitor, service procurer and service integrator applications built into the service infrastructure.

3.1 Service Manager

Current design of the Internet requires the consumers to manually search/discover and procure a service available on the web and then integrate it within their IT environment. This approach is cumbersome for the consumers and also prone to guess-work on the user's part as to which service is needed. Smart Internet in the future would enable automation of this step by dynamically identifying the services that a user needs and automatically procuring them. This would be similar to the existing upgrade utility on computers that automatically checks the software versions installed and then runs the update in the background after confirming with the user. For instance, if a user is consuming a service that provides financial information, then additional services that are either missing or could enhance the information displayed would be automatically identified and procured for the consumer depending on the service procurement policies that the consumer has defined. Smart services infrastructure running on the consumer's IT environment will include the following applications:

Service Manager manages the services by tracking their quality and performance. It also identifies gaps in the services and inform the Service Monitor of them so that the Monitor could search for services that can fill the gap. If a service does not meet the desired performance levels, the service manager will send alerts to humans or automatically terminate the service and send instructions to the service monitor to poll for another service.

Many software providers like Microsoft [19], IBM [20], Hewlett-Packard [22], Oracle [21] etc. have released basic versions of service managers in the market. However these products are platform specific. For instance, the Microsoft service manager manages only Windows specific services and Oracle server manages only Oracle services. Due to this the consumer has to either manually integrate service management across various platforms or write a proprietary program to manage the service flow across platforms. We envision that in Smart Services, the Service Manager application will be platform independent and will provide a user-centric aggregated view of all the services in the consumer's IT environment .

Service Monitor program regularly polls the cloud to identify any new services or service upgrades that have been released. It refers to the Service policy master to determine the domains and constraints that it should filter off.

When the service monitor identifies a service that needs to be procured, it sends the list of requirements along with the associated policies determining the service constraints to the Service Procurer.

Service Policy master manages the policies decided by the consumer on service discovery and procurement. For instance, the consumer may specify a limited list of providers for service discovery; or may want to limit service discovery to a geographical location. Similarly the consumer may specify the budgetary, security or language constraints as part of their procurement policy. The Service Policy Master would ideally contain all the business rules and IT policies of the organization that should be considered by the consumer before procuring a service.

The collection of consumer's policies is translated to a description using semantic web languages such as RDF and OWL. We plan to use the Rei policy framework [12] (http://rei.umbc.edu) to develop the policies. This will ensure that the Service Policy Master uses a standard interface, and replacing it does not require reentering the policies.

Service Log manages a list of all services in the consumer's service environment. It tracks the service by its functionality, provider, dependent services/components and technical features. The service log also acts as an audit for each service by logging the number of times it was used, the user name and cost to the organization.

3.2 Service Procurer

Service procurer searches for services on the cloud that match the specified service. The discovery is constrained by functional and technical attributes identified by the Service monitor, and also by the budgetary, security, data quality and agent policies of the consumer defined in the policy master. One approach that the service procurer could take for discovering services would be to run a query against the services registered with a central registry or governing body. Alternatively, it could limit the service search to a set of providers specified in the consumer's provider policy. While discovering the service, the procurer will need references for providers that match the requirements, especially if the provider is not a part of the preferred vendor list defined in the policy. Service procurer will be able to get certification for the service provider by contacting a central registry, like UDDI [18].

If the service procurer identifies the exact service matching the specifications and constraints then it automatically procures it and transfers the control to the service integrator. If the procurer discovers multiple providers for the service, each meeting the consumer's policies, then it creates an aggregated view of the search results providing instantaneous comparison of the services available, their cost, their constraints and the service gap. The control is then transferred to the consumer who finalizes the service provider. While the existing Internet infrastructure does not allow aggregation of website from multiple servers or transfer of control to the user, Smart Internet has been envisioned to enable this. After the consumer finalizes the service provider, service procurer begins contract negotiation with the provider. Service Level Agreements (SLA) are finalized during this negotiation. SLAs define the service data, delivery mode, agent details, compliance policy, quality and cost of the service.
After the service procurer acquires the service, it transfers control to the service integrator.

3.3 Service Integrator

The Service Integrator program will automatically integrate the new service procured into the consumer's existing service infrastructure. Often different services procured could be combined to form a single service. In other instances, the service procured could be integrated into multiple services. For instance, a currency converter service procured could be integrated with multiple financial application services. Either way, this integration would be transparent and seamless to the end-user. Service integrator will also update the service manager, service monitor and the service log with the details of the new service procured.

3.4 Usability and User Issues

As the "on demand" service systems deal with users, human-computer interaction issues need to be taken into consideration in design, implementation, and maintenance phases of the system and beyond. In designing such software systems with a service focus, emphasis on usability and user issues may be critical for maintaining optimal user and satisfaction. Variations of usability design guidelines initially created by Shneirderman [16] and Nielsen [17] can be adopted for service-oriented systems and its components. Time-critical tasks can be run more effectively on systems with optimally usable interfaces that present the relevant content in a user-friendly design.

4 Conclusion

User-centric smart services will create a paradigm shift in the way management perceives IT in the organization. In this paper we have detailed our vision for services lifecycle in the smart internet. We have described the service manager, monitor, procurer and integrator applications that will need to be developed. To the best of our knowledge, this is the first such effort, and it provides a description of the new applications needed to automate discovery, acquisition and deployment of services.

About the Authors

Karuna P. Joshi is an IT Program/Project Manager with over 15 years of industrial experience. She worked at the International Monetary Fund for over nine years. She has managed projects across various domains including Web Content Management, Document Management, Helpdesk Applications, etc. She is currently pursuing PhD in Computer Science in the field of Services Sciences Management and Engineering and Distributed Web Systems from University of Maryland, Baltimore County (UMBC). She completed her MS in Computer Science from UMBC in 1999 and her Bachelors in Computer Engineering from University of Mumbai, India in 1993. Apart from SSME, her research interests also include Databases, Web Technologies and Data mining. She can be contacted at kjoshi1@cs.umbc.edu

Yelena Yesha received the B.Sc. degree in Computer Science from York University, Toronto, Canada in 1984, and the M.Sc. and Ph.D degrees in Computer and Information Science from The Ohio State University in 1986 and 1989, respectively. Since 1989 she has been with the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County, where she is presently a Professor and Associate Director for the Multicore Computational Center. In addition, since December, 1994 till august 1999 Dr. Yesha served as the Director of the Center of Excellence in Space Data and Information Sciences at NASA. Her research interests are in the areas of distributed databases, distributed systems, digital libraries, electronic commerce, and trusted information systems. She coauthored 14 books and authored over 180 refereed articles in these areas. Dr. Yesha was a program chair and general co-chair of the ACM International Conference on Information and Knowledge Management, General Chair of ACM Sigmod 2005 and member of the program committees of many prestigious conferences. During 1994, Dr. Yesha was the Director of the Center for Applied Information Technology at the National Institute of Standards and Technology. She can be contacted at yeyesha@umbc.edu.

A. Ant Ozok is an Associate Professor of Information Technology and Engineering at UMBC. He has published close to twenty journal articles in the areas of human factors and human-computer interaction in journals including Behavior and Information Technology, Computers in Human Behavior, Ergonomics, Industrial Management and Data Systems, International Journal of Human Computer Interaction, Journal of Organizational Computing and E-Commerce, and more. His research has been supported by the National Science Foundation, United States Department of Education and other entities. He is associate editor of two journals and chaired the Third International Conference on Online Communities and Social Computing in 2009. He can be contacted at ozok@umbc.edu.

Yaacov Yesha is a Professor at the Department of Computer Science and Electrical Engineering of the University of Maryland, Baltimore County. He received the Ph.D. degree in computer science in 1979 from the Weizmann Institute of Science. His interests include Web services, mobile computing, wireless networks, software testing and SSME. He has many publications in refereed journals and proceedings, and has received substantial external research funding from NSF, NSA, IBM, SAP, LTS, Northrop Grumman and Aether Systems Inc. Yaacov Yesha was a program committee member of several conferences, a program vice chair for the Seventh International Conference on Parallel and Distributed Computing Systems, 1994, a chair of the workshop Towards a Research Tradition in Services Science, Management, and Engineering at IBM CASCON 2007, October, 2007, and a chair of the workshop Addressing the Services Science, Management, and Engineering Curriculum: Now and Future at IBM CASCON 2007. He can be contacted at yayesha@umbc.edu.

References

[1] M Xu, Z Hu, W Long, W Liu, Service virtualization: Infrastructure and applications - The Grid: Blueprint for a New Computing Infrastructure By Ian Foster, Carl Kesselman, Morgan Kaufman, 2004

[2] J. Ng, M. Chignell, J. Cordy, The Smart Internet Transforming the Web for the User, Sep 2009, Manuscript under preparation

[3] M. Papazoglou and W. Van Den Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering and Technology, Volume 2, Number 4, 2006, pp. 412 – 442

[4] D. Bianchini, V. De Antonellis, B. Pernici, P. Plebani, Ontology-based methodology for e-service discovery, International Journal of Information Systems, The Semantic Web and Web Services, Volume 31, Issues 4-5, June-July 2006, pp 361-380

[5] L Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q. Sheng, Quality driven web services composition, Proceedings of the 12th international conference on World Wide Web, 2003, pp 411 - 421

[6] E. M. Maximilien, M.Singh, A Framework and Ontology for Dynamic Web Services Selection, IEEE Internet Computing, vol. 8, no. 5, pp. 84-93, Sep./Oct. 2004

[7] J. Black et al, An integration model for organizing IT service Management, IBM Systems Journal, VOL 46, NO 3, 2007

[8] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl, March 2001

[9] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C recommendation, World Wide Web Consortium, 2004.

[10] O. Lassila, R. Swick and others, Resource Description Framework (RDF) Model and Syntax Specification, World Wide Web Consortium, 1999

[11] Web services Business Execution Language (WS-BPEL) 2.0, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf

[12] L. Kagal, T. Finin, A. Joshi, A Policy Based Approach to Security for the Semantic Web, In proceedings of 2nd International Semantic Web Conference (ISWC2003), September 2003

[13] Anupam Joshi, Z. Jiang, Retriever: Improving Web Search Engine Results Using Clustering, in Managing Business with Electronic Commerce: Issues and Trends, (A. Gangopadhyay, Editor), Idea Press, 2001.

[14] W.T. Balke, M. Wagner, Towards personalized selection of web services, (WWW 2003) Alternate Track on Web Services, 2003

[15] Stanford Personalized PageRank Project, http://nlp.stanford.edu/projects/pagerank.shtml, last retrieved on October 12 2009

[16] Shneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction. New York: Addison-Wesley, 1992

[17] J. Nielsen, Usability Engineering, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993

[18] S Ran, A model for web services discovery with QoS, ACM SIGecom Exchanges, Vol 4, Issue 1, 2003, pp 1-10, 2003

[19] Microsoft System Center: Service Manager, http://www.microsoft.com/systemcenter/en/us/service-manager.aspx, retrieved October 14 2009

[20] IBM Service Management, http://www-01.ibm.com/software/tivoli/solutions/it-service-management/, retrieved October 14 2009

[21] Oracle Web Services Manager, http://www.oracle.com/appserver/web-services-manager.html, retrieved October 14 2009

[22] Hewlett-Packard Service Manager software, https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-85^12473_4000_100__&jumpid=reg_R1002_USEN, retrieved October 14 2009