Towards Cloud Services Marketplaces

Rahul Akolkar, Tom Chefalas, Jim Laredo, Chang-Shing Perng,
Anca Sailer, Frank Schaffa, Ignacio Silva-Lepe, Tao Tao
IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne NY 10532, USA
{akolkar,chefalas,laredoj,perng,ancas,schaffa,isilval,ttao}@us.ibm.com

***Abstract*—When we think of Marketplaces we think of places where third parties bring their goods for consumers to find them and transact in some convenient way. When we take this idea to the cloud services space we think of third party services providers that bring their cloud services, and services consumers that expect to match the capabilities that best suit their needs with a service offering. Services as opposed to goods vary in the way they function, in the way that they get activated, and vary in the way they surface the requirements they fulfill. For the consumer the task does not get easier, and to find a solution tailored to their solution may be a trying task. In this paper we explore the landscape of cloud services marketplaces, where we are, a perspective of an architecture, and in particular some of the enablers that would help consumers engage with the marketplace in an easy to use and successful manner.**

Keywords: *semantic service descriptions; service commerce tools; service deployment; service management infrastructure; service consumability*

I. Introduction

For as long as there have been services, computational or otherwise, there has been a desire to have a convenient medium to expose and discover service offerings. Since the early days of DCE and CORBA, we have had directory servers and trading services, the latter being an attempt at mimicking a sort of yellow pages for distributed objects, the yellow pages being the archetypal medium for finding services by various criteria when a direct or symbolic reference to the service does not exist. More recently we have seen efforts, such as JNDI for Java remote objects, and UDDI for Web services, aimed at defining a standard approach for the exchange of computational services. At the same time, numerous commercial Web services directories and registries have been created in an effort to capitalize on the potential of a services marketplace. In [10], Legner provides an analysis of Web services directories prompted by the question of whether there is a market for Web services. Some of her conclusions are that, on one hand the investigated Web services intermediaries use rather simplistic search and categorization algorithms (despite the more advanced features in specifications like UDDI), and on the other hand classification schemes that reflect the vocabulary of the target customers are a prerequisite for the discovery of suitable Web services.

We believe that a services marketplace should fulfill the promise of an electronic emporium where third party service providers are able to offer their services in a ubiquitous ecosystem, and where service consumers are able to acquire service solutions that are tailored to their requirements. Amazon has established itself as the go-to electronic commerce site for (1) buyers looking for a wide variety of products and the ability to compare and contrast products from a wide variety of sellers, and (2) for sellers to have direct access to the Amazon catalog and to take advantage of brand name, one-stop shopping and reputation systems to target a buyer base that they might not otherwise have access to. Similarly, we envision a services marketplace as (a) providing consumers with service solutions that are custom-fit to their requirements, with the system implicitly selecting from a variety of service offerings, and (b) enabling providers to offer their services via an intelligent service store that has deep knowledge of services and consumers, and that is able to respond to consumer requests and learn from its experience.

To a service consumer this means that requirements do not need to be stated in terms of service offering features or the a priori existence of an off-the-shelf solution, the system will be able to compose a solution. In addition, service consumers can take advantage of the social nature of the marketplace to rate and recommend services and providers, and to let the marketplace use other consumers' ratings and recommendations factor into the fulfillment of a service requirement. To a service provider, a services marketplace is not only a means to expose service offerings, its existing content and features are also a means to enrich new offerings via, for instance, custom adaptation and incorporation of such customizations back into the service store. The social nature of the marketplace also allows providers to reach an otherwise not available service consumer base, and to benefit from the competition and collaboration with other providers.

The remainder of this paper focuses on key enablers that are needed for services marketplaces, in particular, from a services consumer point of view. We also present a prospective architecture that lays a foundation for realizing our services marketplaces vision.

II. Key enablers to get where we want to go

If we look back at what types of service have been successful, we see standalone services that enhance offerings, and platforms that have evolved into ecosystems. These platforms offer full life-cycle processes for their data artifacts, and as a result they become the system of record for those key artifacts, such as saleforce.com. These elements create a so-to-speak stickiness, and are able to build an ecosystem that keeps participants engaged. These platforms are in essence

978-3-901882-48-7

marketplaces where a consumer can come and pay for the base services and then get add-ons that enhance the value of those base services. To build a successful services marketplace as we envision it, it is necessary to bring together number of necessary, although not necessarily sufficient, enablers. We elaborate on a subset of these enablers in the following subsections, namely consumability and intelligence.

A. Consumability

One of the key barriers to service use is how easy it is for a service consumer to fulfill a business need, even when assuming that a service solution may exist, either readily available, or via adaptation or composition. In fact, in many cases a service consumer's needs are expressed in general or vague terms that are not directly translated into the corresponding service solution. For example, when a service consumer states a "need for a payroll service solution", the fact that a particular realization of, say an ADP solution, may fit the consumer's needs is not obvious.

Traditional approaches to customizing a service solution are limited by how well the marketplace can understand the consumer's requirement, what it knows about the solution, how well it can correlate the two, and how it can adapt the solution to fit the consumer's needs. For instance, understanding a need for a payroll service solution can amount to checking off a multiple choice box. But when the number of choices grows beyond the grasp of the consumer, or when presenting each choice cannot be reasonably done with a few words, alternatives to the familiar form-based and menu-based interfaces are needed. Similarly, narrowing down possible payroll solutions can amount to presenting SQL query results in a table, which the consumer would try to understand and select from. Alternatively, the marketplace could present the consumer with the next level of choices based on its understanding of results to service knowledge queries.

In order to substantially improve the consumability of service use it is necessary to go beyond traditional approaches. In particular, a consumer should be able to express their needs and requirements in as flexible a manner as possible. The marketplace should be able to understand a consumer's request in as many modalities as possible (including, for instance natural language requests), it should turn vague or ambiguous requests into service knowledge queries and their results, and it should engage in a conversation with the consumer to further elucidate details of their requirements. In addition, a consumer should be able to regard the marketplace as custom-fitting a service solution to their needs. The marketplace should be able to not only determine a candidate service solution, but it should also customize such a solution to fit the consumer's needs, possibly by composing more than one candidate solutions. For example, after a consumer states a need for a payroll service solution, the marketplace would be able to determine that a few candidate solutions may fit but such candidates may vary depending on the size of the business and whether payroll data could be maintained in the cloud or they need to stay on premise. After engaging in a conversation with the consumer, the marketplace would determine that an ADP solution could fit their needs. If payroll data need to be maintained on premise and a third party solution could be used, the marketplace would also perform a (perhaps semi-automated) adaptation of ADP with, for instance, SAP, provision an instance of the composition, and return a URL for the consumer to start using.

B. Intelligence

A system that attempts to custom-fit a service solution to a consumer's business needs is only as good as its solution repository, its deep question answering (DeepQA [1]) ability and its ability to internalize its previous experience. We refer to these features as the intelligence of a services marketplace. These features are also beneficial to a service provider, who is interested in incorporating service knowledge from service characterizations.

In [10], an analysis of directories for web services suggests that service descriptions that go beyond those found in marketplaces such as Universal Business Registry (UBR), StrikeIron and RemoteMethods, for instance, are a prerequisite for service consumers to discover and use suitable service offerings; in particular, such descriptions should include more sophisticated classification schemes that reflect the vocabulary of the target service consumer.

More recent service repository products have started to incorporate semantic metadata into service descriptions. For instance, WebSphere Service Registry and Repository (WSRR) is able to load and parse OWL files, but currently does not exploit any of the richer semantics of the OWL language. Instead, the taxonomy part of the OWL file (i.e., the owl:Class and rdfs:subClassOf elements) is made available to "classify" artifacts in WSRR (in WSRR, each taxonomy is referred to as a "classification system").

An intelligent services marketplace knows about a lot of service offerings (orders of magnitude more than existing web service marketplaces, for instance), and it knows a lot about those service offerings. It knows about service offerings in a great many domains – insurance, payroll, health care, accounting, to name a few – and it knows variations of those services, for instance a payroll service customized to fit payroll data on-premise and another one using an ad-hoc payroll data format. It knows about simple service offerings, such as a stock quote service, and about composite service offerings, such as a financial services application, that includes a stock quote service as well as a trading service. But an intelligent services marketplace also knows a lot about each service offering. It knows about its service interfaces, of course, but it also has semantic knowledge about what the service does, its application domain, its points of variability, for instance.

An intelligent services marketplace is also able to perform DeepQA over a vast knowledge base of service offering metadata, including syntactic and semantic knowledge. That is, it is able to "complement classic knowledge-based approaches with recent advances in NLP, Information Retrieval, and Machine Learning to interpret and reason over huge volumes" [1] of service offering metadata to find precise results (as opposed to unmanageable numbers of hits that a traditional search engine would return) with reasonably accurate confidence (as opposed to the logical proof of correctness that a classic QA system would attempt).

III. Prospective architecture

Based on our vision of a cloud services marketplace, we have started developing a concrete realization. As a first step, we have defined a high-level architecture that brings together the key enablers we have enumerated into functional modules that (1) support service providers and consumers, and (2) embody a marketplace platform and cloud abstraction layer. The diagram in Fig. 1 illustrates this high-level architecture.

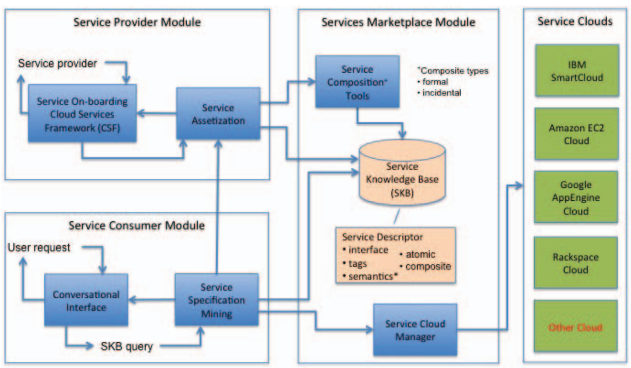

Fig. 1. High-level architecture

An important architectural property of such a cloud services marketplace is the feedback loop that allows for new service compositions to be assetized and for these assets to contribute back to the service knowledge base, thus enabling the evolution of the service knowledge base and making subsequent comparable cloud solution delivery easier.

Another key architectural property is the ease with which the marketplace enables stakeholders to change roles. It is clearly simpler for an average service provider or developer to be a service consumer of other services within the marketplace. In addition, by virtue of the conversational interface and provided service composition tools, which we describe below, such a cloud services marketplace also enables the average service consumer to become a service provider by assetizing and contributing a custom-fit, composed services solution that may solve a recurring need, back into the marketplace.

The architecture includes the following modules:

A. Services Marketplace Module

This module serves as the main platform for the services marketplace. It contains various kinds of metadata in a service knowledge base, as well as a number of marketplace support services, including service composition tools and a service cloud manager, that are not further elaborated on for space reasons.

1) Service Knowledge Base: Also known as the service store, or service knowledge network, it is the main repository of knowledge that is obtained from any characterization of a service. Service knowledge comprises both features inherent to a service as well as cross-service relationships. It is encoded using various types of machine-processable representations (formats and languages), each representation type having its own processing agent. The service knowledge base also contains background knowledge about context in which services operate and it provides various levels of query and update interfaces; in addition, the service knowledge base maintains knowledge about consumers and providers in one or more social networks.

B. Service Consumer Module

Two main functions of the service consumer module, in support of service consumers, are to provide a conversational interface to consumers and to perform service specification mining based on service knowledge queries.

1) Conversational Interface: Given a request from a consumer (in modalities such as natural language but not excluding generated choices), this interface engages in a conversation with the consumer. It generates a service knowledge query from the initial request, that it forwards downstream for processing. This processing may yield choices for the user to make in order to narrow down the customization process, which the conversational interface uses to generate questions for the consumer. Aspects of the conversational interface are detailed in Fig. 2.

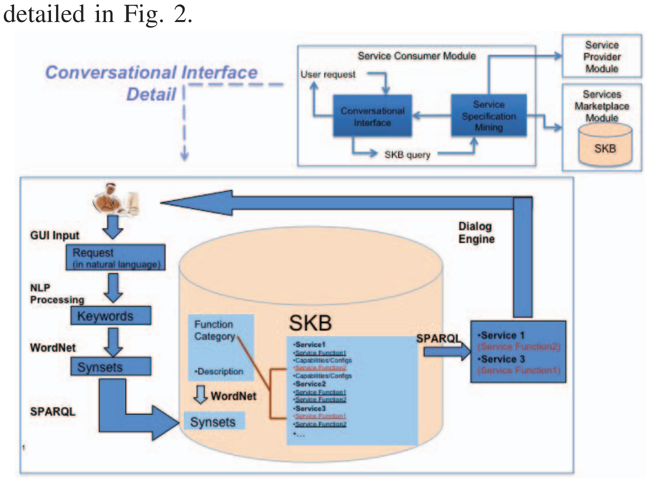

Fig. 2. Conversational Interface Details

In particular, a consumer's request can be analyzed into key words. These key words can be used to create SPARQL queries to look for service functions defined in the service knowledge base. In addition, to improve the ability to find a matching service function, words that are related to the consumer's key words may also be used to create additional SPARQL queries. Specifically, given a dictionary such as WordNet, derivative forms of a key word, such as hypernyms, hyponyms, holonyms and meronyms could be used. For instance, take the word payroll from a consumer's request such as "need for payroll service solution". This word's corresponding derivative forms include register (hypernym), and register, pay, earnings and salary (meronyms). Assuming that a service function has been annotated with one of these words, generating queries from all of them will allow finding such service functions.

2) Service Specification Mining: The main task of this component is to mine the service knowledge base for service solutions based on input service knowledge queries. Once a result set is obtained, it is processed with the goal of zeroing in on one, or at most a handful of solutions for the consumer. In order to do this, the elements of the result are analyzed to determine criteria on which to continue asking questions. As this iterative process is performed, this component also maintains conversation state as questions are asked and answered.

IV. Related Work

Existing literature related to IT Cloud ecosystems can be grouped in three main classes: (1) service search and composition, (2) handling of service quality and price, (3) service deployment workflow automation. The field of service matching and composition is traditionally a technical domain that is dominated by computer science research. For determining a suitable service a wide range of methods have been proposed. Traditional services catalogs or registries, such as Universal Description Discovery and Integration (UDDI), are intended to store metadata for SOA services retrieval. The Web Service Modeling Ontology (WSMO) [14] is an effort to fill the semantic gap between different services metadata by adding semantic capabilities to the Web services description which was initially based on a syntactic interchange approach only. Thus, WSMO, OWL-S and other Semantic Web services based frameworks and languages [2] [8] attempt to alleviate the search and composition of services.

Other marketplace service matching approaches rely on AI planning algorithms and apply backward chaining to derive suitable matching and composition from a certain goal. Such an approach is, for example, presented in [11] for stateful and in [16] for stateless services. Often such approaches leverage formal descriptions of service functionality as proposed by the W3C OWL-S or WSMO. Composition that also considers information about the (temporal) behavior of a service is presented in [3]. A graph service representation is used in [15] for service search, composition, and mapping to an operational infrastructure in data centers in view of optimized deployment. Typically, such approaches support matching and composition using service functionality criteria only and disregard other non-functional (e.g., business-related) service properties used for service differentiation (e.g. quality of service or prices).

QoS-based matching and composition algorithms are introduced in [17]. However, these approaches do not support declarative representation for service descriptions and queries as required for service retrieval. An approach for a compact representation of service descriptions and queries with complex pricing and preference functions is presented in [9]. Although this approach supports multiple service configurations and features a service selection algorithm, the approach is restricted to the selection of single services and not applicable for service composition. Typically, the presence of asymmetric information and rational opportunistic behavior of service providers and customers is always ignored in the body of work that deals with service composition from a purely technical perspective.

In the economics and marketing science literature, prior contributions typically focus on the trade-off between extra marketplace coverage and lost revenue, building on the rational opportunistic behavior of service providers and customers. They differ with respect to the customer segments considered and the number of service versions. While the more applied articles [6], [7], [12] consider discrete service quality offerings for a discrete number of customer segments, [13] develop a model for continuous service qualities offered to a continuum of customers.

By design, cloud service products are characterized by an inherent product complexity. Hence, the description of the cloud service and the cloud service customer preferences need to be driven by a combination of functional and non-functional attributes. In the economic literature, typically, a cloud service utility is modeled as linear multi-attributive utility functions over non-functional quality attributes which yield a user's willingness to select and pay for the product [9], [4], [5] and can lay the basis for price negotiation.

Our work is complementary to the above efforts in at least two aspects: (1) we target general IT Cloud services, which need a more comprehensive service descriptor than the web services (e.g., booking airline tickets) existing work handles today; (2) our services catalog is intended to cover a broader definition of API, going beyond SOAP into REST and DCOM, to name a few.

V. Conclusions

The next generation of service marketplaces will require capabilities to custom-fit service solutions for consumers, to allow providers to retail and differentiate their services, and to warehouse and deeply understand large amounts of service knowledge. In addition, such marketplaces will need to enable social networking of consumers and providers, and provide economics and operational capabilities that allow for price negotiation and differentiation. We have argued for a minimal number of enabling technologies to fulfill our next-generation marketplace vision, motivated by a number state-of-the-art themes, and we have incorporated these enablers into a prospective architecture to realize the vision. This architecture represents a starting point onto which additional enabling technologies can be integrated in the future.

REFERENCES

[1] DeepQA. http://www.research.ibm.com/deepqa/deepqa.shtml.

[2] Steve Battle, Abraham Bernstein, Harold Boley, Benjamin Grosof, Michael Gruninger, Richard Hull, Michael Kifer, David Martin, Sheila McIlraith, Deborah McGuinness, Jianwen Su, and Said Tabet. Semantic Web Services Language. http://www.w3.org/Submission/SWSF-SWSL/, September 2005.

[3] Daniela Berardi, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and MASSIMO MECELLA. Automatic service composition based on behavioral descriptions. *INTERNATIONAL JOURNAL OF COOPERATIVE INFORMATION SYSTEMS*, 14:2005, 2005.

[4] Rainer Berbner, Michael Spahn, Nicolas Repp, Oliver Heckmann, and Ralf Steinmetz. Heuristics for qos-aware web service composition. *Web Services, IEEE International Conference on*, 0:72–82, 2006.

[5] Benjamin Blau, Clemens van Dinther, Tobias Conte, Yongchun Xu, and Christof Weinhardt. How to Coordinate Value Generation in Service Networks–A Mechanism Design Approach. *Business and Information Systems Engineering (BISE)*, 1(5):343–356, October 2009.

[6] Gregory Dobson and Shlomo Kalish. Positioning and pricing a product line. *Marketing Science*, 7(2):107–125, 1988.

[7] Michael L Katz. Firm-specific differentiation and competition among multiproduct firms. *The Journal of Business*, 57(1):S149–66, January 1984.

[8] S Kona, A Bansal, G Gupta, and T D Hite. Web service discovery and composition using usdl. *The 8th IEEE International Conference on ECommerce Technology and The 3rd IEEE International Conference on Enterprise Computing ECommerce and EServices CECEEE06*, pages 65–65, 2006.

[9] Steffen Lamparter, Anupriya Ankolekar, and Rudi Studer. Preference-based selection of highly configurable web services. In *In Proc. of the 16th Int. World Wide Web Conference (WWW07*, pages 1013–1022. ACM Press, 2007.

[10] Christine Legner. Is There a Market for Web Services? In Elisabetta Nitto and Matei Ripeanu, editors, *Service-Oriented Computing - ICSOC 2007 Workshops*, pages 29–42. Springer-Verlag, Berlin, Heidelberg, 2009.

[11] Freddy Lcu and Alain Lger. A formal model for semantic web service composition. In *IN ISWC06*, pages 385–398, 2006.

[12] K. Sridhar Moorthy. Market segmentation, self-selection, and product line design. *Marketing Science*, 3(4):288–307, 1984.

[13] Michael Mussa and Sherwin Rosen. Monopoly and product quality. *Journal of Economic Theory*, 18(2):301–317, August 1978.

[14] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubn Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Christoph Bussler, and Dieter Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.

[15] Anca Sailer, Michael R. Head, Andrzej Kochut, and Hidayatullah Shaikh. Graph-based cloud service placement. *Services Computing, IEEE International Conference on*, 0:89–96, 2010.

[16] Evren Sirin, Bijan Parsia, Dan Wu, James Hendler, and Dana Nau. Htn planning for web service composition using shop2, 2004.

[17] Liangzhao Zeng, Boualem Benatallah, Anne H. H. Ngu, Marlon Dumas, Ieee Computer Society, Jayant Kalagnanam, and Henry Chang. Qos-aware middleware for web services composition. *IEEE Trans. Software Eng*, pages 311–327, 2004.