

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Survey of Service Description Languages and Their Issues in Cloud Computing

Le Sun¹, Hai Dong², Jamshaid Ashraf³

School of Information Systems, Curtin Business School, Curtin University of Technology, Perth, WA, Australia

¹le.sun1@postgrad.curtin.edu.au

²hai.dong@cbs.curtin.edu.au

³jamshaid.ashraf@gmail.com

Abstract—Along with the growing popularity of cloud computing technology, the amount of available cloud services and their usage frequency are increasing. In order to provide a mechanism for the efficient enforcement of service-relevant operations in cloud environment, such as service discovery, service provision, and service management, a completed and precise service specification model is highly required. In this paper, we conducted a survey on existing service description languages applied in three different domains—general services, Web/SOA services, and cloud services. We discussed and compared the past literature from seven major aspects, which are: (1) domain, (2) coverage, (3) purpose, (4) representation, (5) semantic expressivity, (6) intended users, and (7) features. Additionally, two core dimensions – semantic expressivity and coverage – are employed to categorize and analyse the key service description languages by using Magic Quadrant methodology. These two dimensions are regarded as the most essential factors for the evaluation of a service description model. Based on this analysis, we concluded that Unified Service Description Language (USDL) is the language with the widest coverage from business, technical and operational aspects, while OWL-S is the one that has the highest semantic expressivity. At last, critical research issues on cloud service description languages are identified and analysed. The solution of these issues requires more research efforts on the standardization of cloud service specification, which will eventually enhance the development of cloud industry.

Keywords—Cloud computing, Service description language, Semantic expressivity, Coverage

I. INTRODUCTION

Along with the growing popularity of the cloud computing technology, the amount of available cloud services and their usage frequency are increasing. Compared with other types of services provided via the Internet, cloud computing service has its specific features, thereby presenting several new challenges on service delivery, consumption and management.

One of the main challenges is there is a need of a comprehensive and precise specification language with the purposes of (a) covering unique features of cloud services, (b) advancing the ability of discovering and selecting cloud services, and (c) ensuring a consistent and reliable service provision.

To improve the understanding of this need, the specific features of cloud computing services were summarized as follows [1]:

1) *Five essential characteristics of cloud computing services*: (a) **on-demand self-service** – the ability to provide computing capabilities on-demand and to render service-evolving without human interaction; (b) **broad network access** – the cloud resources can be accessed through standard mechanisms and over the network; (c) **location independent resource pooling** – the inability of the service users to be aware of the exact location or to control the provided resources; (d) **rapid elasticity** – cloud resources can be provided and scaled as needed by service users; and (e) **measured service** – cloud resources are usually paid for according to the consumption of service users.

2) *Three basic service delivery models*: (a) **Infrastructure as a service (IaaS)** – The basic computing capability, e.g., storage, processing, network, is delivered as the standardized services over the network; (b) **Platform as a service (PaaS)** – services at this layer refer to a development environment where developers can build and run an applications by using prebuilt components and interfaces that particular platform provides as a service; and (c) **Software as a service (SaaS)** – software applications are delivered as services at this layer.

3) *Four major cloud deployment models*: (a) **Public cloud** – The cloud infrastructure is made available to the general public or a large industry group, (b) **Private cloud** – The cloud infrastructure is operated solely for an organization, (c) **Hybrid cloud** – The cloud infrastructure is a composition of two or more clouds that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability, and (d) **Community cloud** – The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns.

Generally, it is believed that a specification language of cloud services should be able to express the implications of cloud computing and to cover all these aspects.

In terms of service discovery and selection, a great amount of work currently is being carried out on Internet

service description and standardization, especially in the area of Service-oriented Architecture (SOA) and Web services. One of the most prominent languages is WSDL. It defines a Web service from a technical perspective, including the aspects of service, interface, operations, endpoint, binding, and type definition [2]. In spite of its effectiveness and popularity, two weaknesses are still identified by Dong et al. [3]: (a) a lack of semantic supports to indicate the meaning and semantic constraints of data involved in Web services, which may generate ambiguities during the service discovery process; and (b) it does not cover the description of the capabilities of a Web service, and therefore it is not able to recognize the similarity between the capabilities of a Web service being provided and the functionalities of a Web service being requested in the matchmaking process. They also highlighted the usefulness of semantic modelling technologies on Web service description and matchmaking, and did a rounded review on main semantic Web service description languages, including the introduction and comparison of DAML-S, OWL-S, WSMO, WSDL-S and SAWSDL etc.

The application of the semantic technology does improve the efficiency and accuracy of service discovery due to the decreased semantic ambiguities of the matching between service description and service requirements. However, the main goal of the Web services description languages, no matter they are semantic or non-semantic, is to provide a distributed computing infrastructure for both intra and cross-enterprise application integration and collaboration [4]. They describe services from pure technical point of view, which is not sufficient for the specification of business services delivered via the Internet. A business service is concerned with the end-to-end delivery that is delivered by a provider to a consumer possibly over a specified period of time, a payment structure, a service level agreement, and related legal obligations of the consumer and the provider [5]. Cloud computing service is exactly a type of business services as we can see from its definition and features. Like the statement in [6], a business service is not a function alone. It is a function performed on customers' behalf at a cost that is not only a monetary cost, but a whole collection of limitations. In order to close the gap existed in Web service description, O'Sullivan [6] focused on the definition of non-functional properties of electronic services with the purpose of improving the technology of automated service discovery, comparison, selection and substitution. Based on this work, Cardoso et al. [5] proposed a Unified Service Description Language (USDL) aiming to describe Internet services from business, operational and technical perspectives. By capturing the business and operational nature of services and aligning them with the technical perspective, USDL has more significant applicability on Internet service discovery, comparison, evaluation and management, enabling more rigorous decision-making of service requestors.

When it comes to cloud service specification, the existing research more concentrates on the consistent and reliable

service provision like flexible VM migration or service integration. JUSSI et al. [7] analysed the cloud service domain languages and divided them into three groups based on their purposes and functionalities, namely cloud service declarative definition language (CSDL), cloud service constraint language (CSCL), and cloud service manipulation language (CSML). Specifically, CSDL provides the necessary abstraction constructs to describe the operational, performance, and capacity requirements of infrastructure, platform, and application third-party services. A typical example is the open virtualization format (OVF) [8]. It describes a specification of software packaging and distribution to be run in VMs, to address low-level interoperability problems. CSCL specifies any explicitly stated rule or regulation that prescribes any aspect of cloud service defined in CSDL and verifies the validity of rule combinations. Logic programming is a convenient and concise notation to exemplify the aims of a CSCL, based on which a wealth of existing languages could potentially be used, such as RuleML [9], SWRL [10], and WS-BPEL [11], etc. CSML provides a set of operators for manipulating, comparing, and redefining CSDL blueprints, without compromising backward compatibility. Examples include W3C's XQuery that defines a query language to extract information matching with specific criteria, and QVT [12] proposed by OMG for the operations of query, view and transformations. Nonetheless, these cloud specification languages are merely from technical or operational perspectives, focusing on service configuration, service manipulation, or service provision processes. They lack the ability to support business-level operations like service discovery, selection, or assessment based on business policies or strategies.

This research concentrates on the understanding, comparison, and evaluation of previous service description languages, making a rigorous analysis and discussion by taking into account different phases of a service lifecycle (e.g. service discovery, service configuration, service consumption, and service evaluation), and from diverse perspectives (e.g. business, technical and operational). The main purpose is to identify the benefits, usefulness, as well as the gaps, issues of the existed languages, making an in-depth thinking and exploring for cloud service specification and standardization.

This paper is structured as follows: In Section 2, we give an overview of the contemporary service description languages. Section 3 analyses the benefits and gaps of the existing service description languages. The identified research issues are presented in Section 4. Section 5 summarizes the job in this paper.

II. SURVEY OF CONTEMPORARY SERVICE DESCRIPTION LANGUAGES

Two of the previous surveys on service description have been introduced in the above section. The first is a review of Semantic Web Service description languages conducted by

Dong et al. [3]. The second is JUSSI et al.'s work [7], focusing on the cloud service domain languages. As an extension and complement of the previous work, this review presents a broader vision on the studies of Internet service specification in the recent five years. It falls into three areas: (1) general Internet services, (2) Web/SOA services, and (3) services in cloud computing environment.

A. General Internet service description

Several efforts have been made devoting to a standard format of the description of services delivered and consumed via the Internet. E3service ontology [13] was proposed to build a customer-oriented catalogue of customer needs for e-services. Its purpose is to automatically elicit the customers' needs for an e-service bundle. Another important domain independent taxonomy for service description [6] is towards both conventional and electronic services, which is capable of representing the non-functional properties of services. The concrete representation of this taxonomy provides the ability to communicate non-functional properties of services as part of a service description, increasing the efficiency of the service discovery processes.

Moerschel et al. [14] illustrated an essential structure of service description in the procurement stage. The procurement stage was divided to 14 steps covering the before-, during-, and after- service supply phases. In particular, 16 attributes were defined in before-service-supply phase of the procurement process, which complemented the work of O'Sullivan from both business and technical perspectives. Based on a study about electronics market places and industry workshops, this work aims at the advancement of the service trade, the industrialization of the service sector, and the transparency of the service sector.

Compared with the above mentioned service description languages, the USDL [5] provides a more comprehensive view towards the description of business services in the context of IoS (Internet of Services). Rather than describing services only from technical or functional perspectives, it provides a solution of service description by combing three kinds of properties of services: business, operational and technical (BOT). It complements the current functional- or technical- based service description languages. As a result, it enables service providers publish their services by using a more comprehensive means, and enable users discover and select services in any business situations.

B. Web service/SOA description

SOA is changing the way of service trade in the context of e-business. To support all steps of the service trade and the process of service orientation, the research on the description language of SOA services is conducted for a long time. The most widely accepted one is the W3C standard Web Service Description Language (WSDL) [2] which is towards the description of Web services. It mainly focuses on technical aspects like the interaction interface and protocols, the interacted messages and the message exchange patterns. The location of the service is also presented. Although Non-functional attributes are not defined in

WSDL, it continues to rapidly evolve and to gain wide industry support.

Based on the WSDL-based Web service description, the Universal Description Discovery & Integration (UDDI) [15] is devised to establish a service directory by defining a set of service meta-information. This service directory supports the description and discovery of (1) businesses, organizations, and other Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access those services.

Another service specification mechanism that has got a wide range of concerns is the one proposed by CDBI [16]. It defines and describes SOA services from three dimensions—service specification, service implementation and service deployment. These three dimensions are respectively corresponding to three contracts --service specification, Automation Unit Specification and service level agreement documents. The relationships between these different types of contract are also illustrated in detail to ensure a coordinated contractual position between provider and consumers. Another prominent feature of this service specification architecture is it adopts a dynamic, graduated and tired-specification approach, from abstract to specified service description. The level of detail is determined by the following factors: the type of service, the applicable process or SDLC stage, the service lifecycle stage and the context in which the specification is to be used. In addition, based on the determination of suitable service levels, the participants of the service delivery are too clarified and defined.

By using Unified Modelling Language (UML), service oriented architecture Modelling Language (SoaML) specification [17] presents a meta model for the specification and design of services within a service-oriented architecture. It supports a standard way to architect and model SOA solutions, and enables business-oriented and system-oriented service architectures to mutually and collaboratively support the enterprise mission. Additionally, Scheithauer et al. [18] proposed a comprehensive service description framework for the service ecosystem that is an evolution of service orientation, by considering both functional and non-functional properties of e-services.

Apart from the above syntactic-based languages, there is a research trend towards the semantic representation of web services contributing to the semantic knowledge discovery and management. Dong et al. [3] conducted a rigorous survey on semantic web service description languages. Based on these techniques, Zhuge et al [19-21] proposed a semantic model, titled the resource space model (RSM) for specifying, organizing, and retrieving versatile resources. Also, they investigated the mappings between three typical semantic models: the Web ontology language (OWL), relational database model, and resource space model, and suggested integrating them to form a powerful semantic platform that enables different semantic models to enhance each other [22].

C. Cloud service description

Nguyen et.al [23] investigated previous work on cloud service description languages and summarized that much recent work mostly aims to propose standards for only certain aspects and thus fails to cover the full picture of cloud computing. For example, in order to solve the vendor lock-in problem, [24] and [25] concentrated on the infrastructure level standardizations, targeting at the solutions of interoperability and portability among federated clouds. Model Driven Engineering (MDE) related techniques [26, 27] are also explored in some work consensus on the models, languages, model transformations and software processes for the model-driven development of cloud-based SaaS. After the analysis of the previous work, they proposed a uniform representation, namely blueprint template, to capture the comprehensive knowledge of a cloud service offering. The proposed method can support SBA (Service-based application) developers during the various development phases. This work can (1) assist application developers to pick and choose offerings from multiple software, platform, and infrastructure service providers and configure them dynamically and in an optimal fashion to address their application requirements, and (2) combine different independent cloud-based services necessitates a uniform description format that facilitates their design, customization, and composition.

The current challenges for service providers and users entering into the cloud computing environment is discussed in [7], within which the most prominent one is the ability to automatically provide services, effectively manage workload segmentation and portability, and manage virtual service instances, all while optimizing the use of cloud resources and accelerating the deployment of new services. Against these challenges, the authors analysed the deficiency of the previous work on service delivery languages including the aspects of the cloud service definition, constraint specification, and manipulation. Also, a cloud blueprinting approach is presented, which transforms the fabric of the current inflexible service delivery models by making heavy use of knowledge-intensive techniques that rely on the use of cloud service definition, constraint specification, and manipulation languages.

Sun et al. [28] proposed a model for discovering Cloud resources in a multi-provider environment. This model aims at the overcome of a main challenge faced by cloud application developers: cloud service providers have different means of describing resources and presents different application programming interfaces for the acquiring of Cloud resources. As a result, developers have to make complex decisions involving multiple Cloud products, different Cloud implementations, deployment options, and programming approaches. The proposed model takes into account the constraints of cloud resources mapping to the development of applications, enabling cross-Cloud implementations and avoid a low level technical restriction to a single Cloud provider.

Another problem existed in the current cloud computing market is that the movement of applications to cloud does not happen quickly, in particular for applications running business-critical processes and containing sensitive data [29].

One of the reasons is a lack of service delivery transparency, i.e. clients are afraid of losing the control of their applications and data since the cloud infrastructure is owned and managed by service providers. To improve such transparency thereby enhance the trust of cloud service users to service providers, a specific language cloud# was proposed in [30] to model the internal organisation of cloud. It enables the cloud users understanding more on how services are delivered inside cloud. By this way, service users have more confidence to move their business-critical applications to cloud.

The above literatures devoted to the description of cloud services with the bias towards engineering applications and technical attributes. Meanwhile, there is also a great deal of work tends to the definitions of cloud services from the business perspective. Service Measurement Index (SMI) [31] is a set of business-relevant Key Performance Indicators (KPI's) that provide a standardized method for measuring and comparing a business service. It is expected to become a standard method to help organizations measure business services based on their specific business and technology requirements, and enable individual preferences to be the basis for what defines a good service. From procurement and ongoing service levels, to business viability and security, the SMI Framework provides a holistic view into the entire customer experience for cloud service providers in six primary areas: Quality, Agility, Risk, Capability, Cost and Security. Users of the SMI Framework can not only compare cloud service vendors based on their specific business and technology requirements, they can also make dynamic, real-time decisions on where to best migrate or deploy an application. The Framework provides a single, standard way to evaluate, monitor and implement services demanded by the business.

By extending the SMI with type hierarchies, aspects and goal patterns, [32] describes a model of cloud service characteristics, focusing on non-functional qualities associated with the service and the important associations between service characteristics necessary to generate claims about the goodness or otherwise of a service. It presents several examples of measures, metrics and indicators specified to describe different types of cloud service characteristics included in the model. Meanwhile, to determine a more complete set of relevant cloud characteristics, the authors reviewed previous work from three aspects: quality ontologies from Web service research and practice[33] [34], taxonomies from requirements [35, 36] engineering research and practice and published system quality models [37, 38].

III. DISCUSSION AND ANALYSIS

Several key specification models are compared and analysed from seven aspects (Table 1): (1) domain – the specification models are categorized into three groups, i.e. specification for general services, Web/SOA services, and cloud services; (2) coverage – business or technical. Software engineering attributes are regarded as a sub-class of the technical attributes; (3) purpose – the main purposes of

service specifications, e.g., resource composition, deployment, discovery, selection, etc.; (4) representation – the modelling tools or representation forms used by specifications – XML, UML, plain text, etc.; (5) semantics – the semantic expressivity needed to represent the requirements and capabilities of services; (6) intended users – the type of the intended users, e.g., service requestor, application developers, service providers, etc. Service users refer to the general users; (7) features – additional comments or special emphasis. Research gaps will be identified that highlight the challenges that will be addressed by this paper.

As shown in Table I, work in the area of the Internet service specification has been carried out for different purposes. The specification models of the general services that include conventional services and electronic services are mainly focused on business perspectives, like service matching, discovery, and selection. Their main purpose is to support a more efficient and consistent platform for service trade between service providers and service users. Typically, their intended users/readers are service requestors or service matchmakers by which suitable services can be compared and selected according to their descriptions. The specification models for this kind of services are presented as abstracted taxonomies of the general attributes. They are domain-independent and can be extended and specified by certain domains.

As to services in SOA, service description techniques are reaching maturity, especially that of Web services. WSDL and UDDI, as a standard service description mechanism, has been widely accepted and used by industries and academic circles. Because of the agent-based characteristics of Web services, WSDL is presented by XML documents with the purpose of machine-readable and process-able. Against the one-sidedness of WSDL, i.e., merely specify technical features of services, CBDI proposed a rich service specification language from both business and technical perspectives, supporting different service-related objectives by taking into account more comprehensive factors. It can be used as brochures to assist end users in selecting services. Service providers or brokers can also locate a suitable service portfolio as recommendations to service users based on these detailed service specifications. In addition, this specification can also be regarded as an implementation specification for the developers to deploy their applications in cloud environment. SoaML is a UML-based service modelling language that is mainly used to model the design process of services in SOA environment. As the authors stated, it is an abstracted model language, only defining several core attributes. By defining detailed attributes, SoaML can be extended to describe different kinds of SOA services and even services in cloud environment.

Although currently most public cloud services are

TABLE I. A COMPARISON OF SERVICE DESCRIPTION LANGUAGES

	Domain	Coverage	Purpose	Representation	Semantics	Intended user	Feature
O'Sullivan [6]	General	Business / Non-functional	Service discovery / comparison / selection / substitution	Attribute taxonomy	No	Service requestor	Domain independent
USDL [5]	General	Business & technical	Service matching & discovery	MOF-based meta-model	Low	Service requestor	Domain independent
WSDL / UDDI	SOA	technical	service communication / discovery	XML	Low	Service requestor	Web service
Rich service specification [16]	SOA	Business & technical	Description, implementation, deployment	Document	No	Requestor, provider, & developer	Timing, level of detail, role-related
SoaML [17]	SOA & cloud	Software engineering	Service modelling & design	UML	No	Software engineering practitioner	Model-driven
OWL-S [39]	Semantic Web Service	Technical / Functional	Ditto	OWL/RDF	High	Web service requestor	
WSDL-S [40]	Semantic Web Service	Technical / Functional	Ditto	WSDL/XML	High	Web service requestor	Aligning with Web service standards, language interoperability
SAWSDL [41]	Semantic Web Service	Technical / Functional	Ditto	WSDL / XML / RDF	High	Web service requestor	Lightweight SWS description language
Sun [28]	Cloud	Technical / Software engineering	cloud resource discovery & integration	Programming model	No	Application developers	Constraints-based
SMI [31]	Cloud	Business	Service selection / comparison	Attributes taxonomy	No	Service requestor	Users satisfaction / preference
Blueprint [23]	Cloud	Software engineering	Selection, customization, & composition	Blueprint template	No	Application developers	Inter-changeability interoperability
Cloud# [30]	Cloud	Technical	Transparency / enhance trust	Specification model	No	Service requestor	Internal organisation

delivered via the interface of Web services, they are more complicated and their specifications are expected to cover very different aspects. For example, the NIST defines three delivery models for cloud services, including IaaS, PaaS and SaaS [1]. Hence, the level of detail, or the number of tiers used in a given case, should be specified respectively and appropriate for these models. Current specification methods of cloud services cover different purposes towards different users. For application developers, cloud service specifications are defined to implement accurate resource selection and application deployment. A programming model [28] and a blueprint template [23] are proposed for the achievement of this objective. The interoperability and interchangeability of different cloud services/resources are also the considerations of these models. In terms of the business aspect, CloudCommons [31] defines a set of SMIs to measure the business properties of cloud services. They entirely focused on commercial operations of services, taking into account the measurements of service-users preferences and satisfactions. Cloud# [30] is also a user-oriented service modelling tool, but its main goal is to provide transparency and enhance trust of service users towards the service delivery process by describing the internal organization of cloud services.

Semantic expressivity is becoming one of the most significant considerations when comes to Internet service description. With an original motivation of Web service description, semantics is able to reduce the ambiguities in the process of matching between service requirements and functionalities, improve software reuse and discovery, significantly facilitate service composition and enable integration of legacy applications as part of business process integration [40]. Table I shows that currently the major application of semantic technology still falls in the domain of Web services. Few studies are conducted with an eye to the semantic expressivity of cloud service specification.

Based on the above analysis, we identify two dimensions, i.e., semantic expressivity and coverage, to categorize and evaluate the key service description languages. In order to provide a more intuitive and reasonable view on the features and purposes of these description models, assisting investigators in capturing their main contributions on cloud service specification, Magic Quadrant [42] research methodology is adopted (Fig. 1). These two dimensions are regarded as the most essential characteristics that a cloud service description language should possess. Specifically, “semantic expressivity” is a measure of the ability to support semantic representation and expression. “High” means that the language is able to express service information with strong and precise semantic support, while “low” indicates a lower semantic expression ability towards none at all. The higher the semantic expressivity is, the more accurate the results of service matchmaking and discovery are. “Coverage” defines the aspects covered by the description language, such as the business-relevant attributes or functional descriptions. “Narrow” means that a language defines attributes merely from one or two dimensions, while “wide” implicates a more comprehensive view of service specification.

As shown in the figure, WSDL, SoaML, and the work of [6] show a very low level of semantic expressivity and present a narrow coverage on service attributes definition. Particularly, WSDL describes the technical aspect of Web services. It mainly focuses on what the service is and how it can be used. O’Sullivan highlighted the business-level consideration of the trade of electronic services, while SoaML presents a dynamicity of the service-provision process. USDL also has low level semantic expressivity. However, the mechanism namely “universal description” in USDL allows adding domain specific semantics and provides an advanced and fairly complete solution to describe each USDL entity as precisely as possible [5]. Hence, it presents a relatively higher degree of semantic expressivity. In addition, USDL has the widest coverage of service attributes. It specifies Internet services by taking into account business, technical and operational perspectives, covering kinds of objectives of the service usage.

As to the space of “high semantic expressivity”, three semantic Web service languages that are most widely-accepted are shown in this figure. Specifically, WSDL-S and SAWSDL are both based on the semantic annotation techniques of WSDL, whilst OWL-S, built upon Web Ontology Language (OWL), has richer semantic expression abilities.

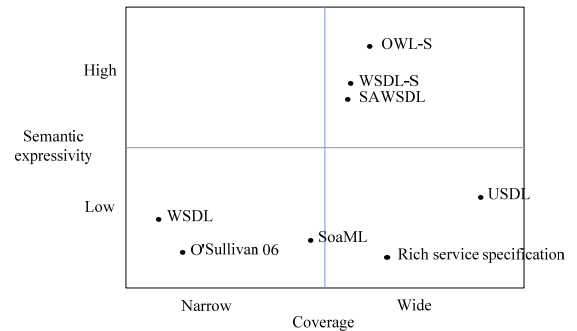


Figure 1. A space of service description languages.

IV. ISSUES IN SERVICE DESCRIPTION LANGUAGES

Based on the above discussion, it is concluded that there is a lack of comprehensive cloud-service specification language that can specify cloud services from different perspectives, against different service delivery models, towards different service users, and for different usage purposes. The details of such issue are presented as follows.

1) *There is a lack of a comprehensive specification model for cloud services covering multiple perspectives: business, operational, and technical.*

Current cloud service specification language is not able to distinguish the characteristics of different cloud service delivery models. Services deployed on distinct cloud delivery models are aimed at separate user-groups, relying on disparate cloud resources, and presenting specific interaction interfaces.

2) *Different cloud deployment models should be considered when it comes to cloud service specification.*

In order to establish cost-effective cloud solutions for various cloud users, four types of cloud deployment models were identified by NIST [43]: public, private, hybrid and community. They have distinct requirements on the deployment of cloud services, and respective implications for specific service users, including network dependency, subscribers still need IT skills, risk from multi-tenancy, data import/export and performance limitations, workloads locations. Therefore, the knowledge relating to cloud deployment models needs to be specified and standardized in the specification of cloud services.

3) *There is a need for more granular specification model to make it user-role centric.*

Specification models need to be designed in a way catering to kinds of audiences, like service users, service providers, application developers and brokers. For example, the commercial details, quality levels and business levels are all likely to have different meanings for separate audiences. In addition, it is desirable that the service specification can be broken down into more manageable chunks towards different audiences [16].

4) *Existing cloud service specification models do not have the ability to model the major participating actors and their relationships and interactions in the process of service delivery.*

The NIST Cloud Computing Reference Architecture (RA) and Taxonomy (Tax) [44] defines five major participating actors: cloud consumer, cloud provider, cloud broker, cloud auditor and cloud carrier. A successful service delivery requires efficient interactions between these participants. An illustration of the participating roles, their relationships and interactions contributes to a clear understanding of the service delivery process, thereby enable service users selecting capable services and managing their applications or data efficiently. There is a need to detail the service specification step by step during the lifecycle of cloud services.

As the service passes through its lifecycle states, it is expected to see an increasing detail of the artefacts of a service specification. For example, CBDI starts their service description model by an abstract service definition, followed by a specifying process towards the detailed service specification[16].

5) *Previous specification models of cloud services do not provide measurement mechanisms for service attributes and do not present the association\relationship and re-usability of the attributes.*

It is expected to define a measurement mechanism in service specification models as it provides references for both service providers and service users on the judgement of the service-contract violations. Furthermore, the specification should be able to record common service attributes that can be shared or reused for the same kinds of services.

6) *There is no research focusing on the semantic representation of cloud services, although Semantic Web technology has been emerged and developed for a long time.*

It is noted that semantic technology is giving a huge boost to the development of electronic service marketplaces, and will be a main research direction in the area of service matchmaking and service transaction. Hence, semantic expressivity should be considered as one of the most important dimensions when developing specification models for cloud services.

V. CONCLUSION

Cloud computing is receiving a significant amount of attention from both academia and industry. In order to provide a mechanism for the consistent and efficient enforcement of service-relevant operations, such as service discovery, service provision, and service management, a completed and precise service specification model is highly required. In this paper, we conducted a survey on service description languages in three different domains—general service description, SOA service description, and cloud service description. We discussed and compared the past literature from seven major aspects, which are: (1) domain, (2) coverage, (3) purpose, (4) representation, (5) semantics, (6) intended users, and (7) features. Additionally, two core dimensions (semantic expressivity and coverage) are captured to categorize and analyse the key service description languages by using Magic Quadrant. This Magic Quadrant indicates different levels of semantic expressivity and various aspects covered by several popular service description models. These two dimensions are regarded as the most essential factors for the evaluation of a service description model. Based on this, we concluded that USDL is the language with the widest coverage that encompasses business, technical and operational aspects, while OWL-S is the one having the highest semantic expressivity. In summary, USDL is able to present a comprehensive view for service requestors when conduct service evaluation and selection, while OWL-S is more beneficial to the decrease of ambiguity of the description languages.

Subsequently critical research issues on cloud service description are identified and analysed. Through further studies on these issues, additional fertile avenues for the research that has been formally described in this paper would be provided. Against these issues, our further research will focus on the standardization of cloud service specification with the final purpose of the improvement of the cloud industry.

REFERENCES

- [1] P. Mell, and T. Grance, "The NIST definition of cloud computing," *NIST SP 800-145*, Information Technology Laboratory, National Institute of Standards and Technology (NIST), 2011.

- [2] R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana. (2007) Webservices description language (wsdl) version 2.0 part 1: Core language. [online]. Available: <http://www.w3.org/TR/wsdl20/>.
- [3] H. Dong, F. K. Hussain, and E. Chang, "Semantic Web Service matchmakers: State of the art and challenges," *Concurrency Computat.: Pract. Exper.*, 2012. doi: 10.1002/cpe.2886.
- [4] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: State of the art and research challenges," *Computer*, vol. 40, no. 11, Nov. 2007, pp. 38-45.
- [5] J. Cardoso, A. Barros, N. May, and U. Kylau, "Towards a unified service description language for the internet of services: Requirements and first developments," in *Proc. SCC2010*, 2010, pp. 602-609.
- [6] J. J. O'Sullivan, "Towards a precise understanding of service properties," QUT Thesis (PhD), Faculty of Information Technology, Queensland University of Technology, 2006.
- [7] K. Jussin, and K. Waldemar, *Knowledge service engineering handbook*, CRC Press, 2012
- [8] DMTF, *Open Virtualization Format Specification Version 1.0.0*, Specification DSP0243, Distributed Management Task Force, Inc., 2009.
- [9] H. Boley, S. Tabet, and G. Wagner, "Design rationale of RuleML: A markup language for semantic web rules," in *Proc. SWWS'01*, Stanford, 2001, pp. 381-402.
- [10] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean. (2004) SWRL: A semantic web rule language combining OWL and RuleML. [online]. Available: <http://www.w3.org/Submission/SWRL/>.
- [11] C. Ouyang, E. Verbeek, W. M. P. Van Der Aalst, S. Breutel, M. Dumas, and A.H.M. Ter Hofstede, "Formal semantics and analysis of control flow in WS-BPEL," *Science of Computer Programming*, vol. 67, no. 2, 2007, pp. 162-198.
- [12] I. Kurtev, "State of the art of QVT: A model transformation language standard," *Applications of Graph Transformations with Industrial Relevance*, 2008, pp. 377-393.
- [13] S. De Kinderen, and J. Gordijn, "e3Service-A model-based approach for generating needs-driven e-service bundles in a networked enterprise," in *Proc. ECIS 2008*, 2008.
- [14] I. C. Moerschel, and H. Hoek, "Grundstruktur für die Beschreibung von Dienstleistungen in der Ausschreibungsphase," *Beuth Verlag GmbH*, vol. Ref.Nr. PAS1018:2002-12, 2001.
- [15] OASIS (2002) UDDI Version 3.0.2. [online]. Available: http://uddi.org/pubs/uddi_v3.htm
- [16] P. Allen, *SOA Methodology Report: Service Contracts in the Service Oriented Process*, Everware-CBD Inc., 2008.
- [17] B. Elvesæter, A. J. Berre, and A. Sadovykh, "Specifying Services using the Service oriented architecture Modeling Language (SoaML): A baseline for Specification of Cloud-based Services," in *Proc. CLOSER 2011*, 2011, pp. 276-285.
- [18] G. Scheithauer, and M. Winkler, *A Service Description Framework for Service Ecosystems*, Nr.78, Otto-Friedrich-Univ. Bamberg, 2008.
- [19] H. Zhuge, *The Web Resource Space Model*, New York, USA: Springer, 2008.
- [20] H. Zhuge, *The Knowledge Grid*, 2012 (2nd edition) ed., Singapore: World Scientific Publishing Co., 2004.
- [21] H. Zhuge, "Probabilistic resource space model for managing resources in Cyber-Physical Society," *IEEE Transactions on Services Computing*, vol. PP, no. 99, 24 February, 2011.
- [22] H. Zhuge, Y. Xing, and P. Shi, "Resource space model, OWL and database: Mapping and integration," *ACM Transactions on Internet Technology*, vol. 8, no. 4, pp. 31, September 2008.
- [23] D. Nguyen, F. Lelli, Y. Taher, M. Parkin, M. Papazoglou, and W.J. van den Heuvel, "Blueprint template support for engineering cloud-based services," in *Proc. ServiceWave'11*, 2011, pp. 26-37.
- [24] F. Galán, A. Sampaio, L. Rodero-Merino, I. Loy, V. Gil, and L.M. Vaquero, "Service specification in cloud environments based on extensions to open standards," in *Proc. COMSWARE '09*, New York, USA, 2009, pp. 1, doi:10.1145/1621890.1621915.
- [25] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the intercloud-protocols and formats for cloud computing interoperability," in *Proc. the Fourth International Conference on Internet and Web Applications and Services (ICIW'09)*, 2009, pp. 328-336.
- [26] H. Bruneliere, J. Cabot, and F. Jouault, "Combining model-driven engineering and cloud computing," in *Proc. MDA4ServiceCloud'10*, Paris, France 2010.
- [27] M. Hamdaqa, T. Livogiannis, and L. Tahvildari, "A reference model for developing cloud applications," in *Proc. CLOSER 2011*, 2011, pp.98-103.
- [28] Y. Sun, T. Harmer, A. Stewart, and P. Wright, "Mapping application requirements to cloud resources," in *Proc. Euro-Par'11*, 2011, pp. 104-112.
- [29] C. I. D. Reporter. (2009) Survey: Cloud computing 'no hype', but fear of security and control slowing adoption. [Online]. Available: http://www.circleid.com/posts/20090226_cloud_computing_hype_security/
- [30] D. Liu, and J. Zic, "Cloud#: A Specification Language for Modeling Cloud," in *Proc. IEEE Cloud 2011*, 2011, pp. 533-540.
- [31] CloudCommons, *Introducing the Service Measurement Index*, Cloud Service Measurement Initiative Consortium, 2012.
- [32] K. Zachos, J. Lockerbie, B. Hughes, and P. Matthews, "Towards a framework for describing cloud service characteristics for use by chief information officers," in *Proc. RESS*, 2011, pp. 16-23.
- [33] G. Dobson, R. Lock, and I. Sommerville, "Qosont: a qos ontology for service-centric systems," in *Proc. EUROMICRO-SEEA 2005*, 2005, pp. 80-87.
- [34] A. Gehlert, and A. Metzger, *CD-JRA-1.3.2 Quality reference model for SBA*, S-Cube Consortium, 2008
- [35] B. Meyer, "On formalism in specifications," *IEEE Software*, vol. 2, no. 1, 1985, pp. 6-26.
- [36] S. Robertson, and J. Robertson, *Mastering the Requirements Process*, Boston, Massachusetts: Addison-Wesley Professional, 2006.
- [37] ISO-9126. (2010) ISO 9126 Software Quality Characteristics. [Online]. Available: <http://www.sqa.net/iso9126.html>
- [38] D. G. Firesmith, and B. Henderson-Sellers, *The OPEN process framework: An introduction*. Harlow, UK: Addison-Wesley Professional, 2002.
- [39] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, and T. Payne. (2004) OWL-S: Semantic markup for web services. [Online]. Available: <http://www.w3.org/Submission/OWL-S/>.
- [40] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.T. Schmidt, A. Sheth, and K. Verma. (2005) Web service semantics-WSDL-S. [Online]. Available: <http://www.w3.org/Submission/WSDL-S/>
- [41] J. Kopecký, T. Vitvar, C. Bournez, and J. Farrell, "SawSDL: Semantic annotations for wsdl and xml schema," *IEEE Internet Computing*, vol. 11 no. 6, 2007, pp. 60-67.
- [42] R. S. J. R. J. Hagerty, and B. Hostmann, *Magic Quadrant for Business Intelligence Platforms*, Gartner RAS Core Research Note G, 2012.
- [43] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, "Draft cloud computing synopsis and recommendations," *NIST SP 800-146*, Information Technology Laboratory, National Institute of Standards and Technology (NIST), 2011.
- [44] R. B. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "NIST cloud computing: reference architecture," *NIST SP 500-292*, Information Technology Laboratory, National Institute of Standards and Technology (NIST), 2011.