Towards a Constraint-based Approach for Service Aggregation and Selection in Cloud E-Marketplaces

Azubuike Ezenwoke, Olawande Daramola, Matthew Adigun
Covenant University
{azu.ezenwoke, olawande.daramola}@covenantuniversity.edu.ng, m.adigun@pan.uzulu.za

***Abstract*—Service providers leverage cloud ecosystems and cloud e-marketplaces to increase the business value of their services to reach a wider range of service users. The operations of commercial e-marketplaces can be further enhanced by enabling service composition mechanisms that allow automatic aggregation of atomic services into composite offerings that meets complex user requirements. Existing approaches of cloud service selection are yet to achieve this. Currently, users are constrained to make choices only from a set of predefined atomic services, or at best, manually configure their desirable features and QoS requirements in order to realize their complex requirements given that they have deep knowledge of the service domain. In this paper, a constraint-based approach for service composition and selection to address this problem was proposed. The proposed approach applies constraint-based automated reasoning on feature models to formally guide the aggregation of atomic services to offer composite services in order to satisfy complex requirements with minimal user involvement. The plausibility of the proposed approach is demonstrated via an illustrative customer relationship management (CRM) service ecosystem. The study offers a credible way to replicate the kind of user experience that is currently available on e-commerce platforms in cloud service e-marketplaces.**

Keywords—cloud computing; ecosystem; e-marketplace; feature model; constraint programming

I. INTRODUCTION

In addition to basic cloud service models like IaaS, PaaS, and SaaS, more complex models that support the notion of anything or everything-as-a-service (XaaS) are also possible. However, the current monolithic model imposes vendor lock-ins, such that services cannot be dynamically combined with other third party services to offer more value-adding functionalities to the users [1]. The maturity of cloud computing will be fast-tracked by the ability to commoditize services in e-marketplace environment enabled by cloud ecosystems [2]-[5].

An important enabler for the realization of a true cloud e-marketplace is the possibility of formal and/or incidental service composition to derive complex business solutions [6]. Formal composition refers to the combination of one or more services into composite services before-hand, while incidental composition is described as 'on the spot' service composition based on specific user request [6]. A review of existing e-marketplaces, such as, Saasmax (saasmax.com) and AppExchange (appexchange.com) etc. shows the existence of basic features like product search and product directory, but lacks sophistication that can enable dynamic service composition in order to support the realization of complex business processes [6]. A cloud e-marketplace can benefit from an ecosystem, such that atomic services can be aggregated into composite offerings to be listed in the e-marketplace directory [4], [6]. As a result, multiple service providers can then participate in service provisioning, thus increasing revenue for stakeholders, while users can better access a variety of value-adding service offerings that try to satisfy their requirements [3].

According to [7], ad hoc service composition by the e-marketplace provider is quite impractical and error prone, particularly considering the multiplicity of constraints that might limit the composition of two or more services. Hence the need for an approach that enables automatic service composition in way that obscures the user from the underlying complexities of service provisioning [5]. But existing proposals for a cloud service e-marketplace, like [3], [5], [6], did not specify particular methodology of realizing service composition, but rather presented architectural blueprints of possibilities. In addition, the plethora of services in an e-marketplace necessitates some type of decision support that can assist users to make selection decisions.

Unfortunately, many proposed cloud service selection methods (e.g. [8]-[10]) only enables a user to make selections from a list of predefined atomic services, which cannot address more complex situations where a user's requirements extend beyond the limit of individual atomic services [11]. But [11]-[14] have attempted to address these kinds of complex scenario, by enabling prospective users to select desirables features that are available in specific atomic services in order to realize their complex set of requirements. This usually includes specifying both the QoS requirements and selecting features of the services. Still, the drawback of these attempts is that it is cognitively demanding because the user is expected to have deep knowledge of the domain in order to make useful selections. The proposed approach in this paper bridges this gap; by first organizing and formally guiding the automatic aggregation of atomic services to in way that satisfies complex user requirements; also it reduces the cognitive load in the process of cloud service selection by affording minimal user involvement.

This paper proposes a constraint-based approach that employed techniques from the domain of product configuration and product line engineering, by adopting feature models to model the inter-relationships and constraints among atomic services. The applicability of the proposed approach is demonstrated by an example of customer relationship Management (CRM) as a service ecosystem. The proposed approach in this paper contrasts previous efforts in that it demonstrates a plausible way to facilitate automatic aggregation of atomic services in order to realize complex user requirements in a manner that improves user experience by supporting minimal user involvement.

The rest of this paper is structured as follows: Section II presents the background of this paper discussing relevant concepts. In Section III, related work is presented, while an exemplar problem scenario is described in Section IV. The proposed solution approach is presented in Section V. Section VI discusses the implication of the proposed approach, while this paper concludes in Section VII with future works.

II. Background

This section presents a background on key concepts in the context of this paper. The details are described in the sequel sections.

A. *Cloud Service e-Marketplace*

An e-marketplace is a platform where the demand and supply for certain products or services are fulfilled using information and communication technologies [3], [15], [16]. The cloud e-marketplace extends the concept of an e-marketplace, and is an online platform that manages the distribution and trading of cloud services. On this platform, service providers enlist services with the purpose of integration with other services to form composite services for users to purchase [1], [3], [5], [6], [17], [18]. Typical examples of cloud e-marketplaces include SaaSMax, AppExchange, Oracle marketplace, Google play store.

The maturity of cloud computing would be fast-tracked by successful partnerships and collaborations among multiple service providers to tie services together [19], [20]. The collaborations leverage integrators enabled by Service Oriented Architecture and provide an environment where XaaS are delivered to meet business needs [20].

B. *Cloud Service Ecosystem*

A cloud service ecosystem is an environment that host heterogeneous cloud service offerings from different providers and affords the opportunity for collaborations. The structure of cloud ecosystem is analogous to software product line engineering (SPLE) and product configuration (PC) domains [21], [22]; which enables mass-customization of concrete or insubstantial products targeted at specific requests. In the same vein, a cloud ecosystem affords customized selection and composition of services either formally or on the fly in order to respond top concrete user requirements.

C. *Constraint—Based Reasoning and Feature Modelling*

1) *Feature Modelling*

A feature model is a hierarchically arranged collection of features and consists of the inter-relationships between a parent feature and its child features, and a set of cross–tree constraints that define the criteria for feature inclusion or exclusion [23], [24].

Benavides et al. [23] identified three main types of feature-based models: basic, cardinality-based and extended feature models (EFM). Basic feature model was introduced by Kang et al. [24] and it describes three feature types (Mandatory, Optional, and Alternative) and two cross-tree constraints (Requires and Excludes). A mandatory feature is a feature that must be included in a product, while an optional feature is a feature that may or may not be included in a feature. An alternative feature is one that is selectable from a set of possible features that can be included in a product. Required and Excludes cross–tree constraints in basic feature model are defined as follows: given features X and Y; X requires Y is defined as if X is included in a product, then Y should also be included, but not vice-versa; while X excludes Y means that if X is included then Y should not be included, and vice-versa.

EFMs [23] are annotated with quality information (such as non-functional attributes), and analysis could use these qualities as a basis for specifying valid combination. In this paper, the EFM notation was adopted because of its ability to model the cloud ecosystem, by capturing cloud services, their QoS attributes, inter-relationship and constraints, which is vital to the generation of valid combinations.

2) *Constraint-based Feature Model Analysis*

Automated analysis of feature models uses computer-aided tools to extract important information from feature models [23]. This process entails transforming the feature models into a logic-based representation, which becomes inputs to solvers e.g. Choco constraint solver; and analysis operations are performed to obtain useful information about the model. Approaches that can transform the feature model into formal representations have been classified into Description Logic, Propositional Logic, and Constraint Programming [23]. In this paper, the constraint approach was employed because the process is more straightforward compared to others, and not solver-dependent [23]. Formally, constraint satisfaction problem (CSP) is fined as:

Definition 1 (CSP): A CSP is defined as a finite set of variables, each of which is associated with a finite domain, and a set of constraints that restrict the values the variables can simultaneously take.

The steps and rules for encoding feature models as CSP and the analysis operations that can be performed on the model are described in [23].

3) *QoS Aggregation Functions*

At least more than one service is composed in a valid combination. The QoS attributes of the constituent atomic services are cumulated using aggregation functions so as to determine the overall QoS attributes for the composite service. Aggregation functions are mathematical operators employed to cumulate atomic values based on the composition patterns and the nature of the QoS attribute [25]. Four basic composition patterns inform the arrangement of constituent services in a business process; they include Sequential; Parallel; Conditional (or branch); Loop [25]. Three classes of QoS aggregation functions exist: summation, multiplication and min-max; and are used if a QoS attribute of the composite service is a sum, product or minimum/maximum of QoS components respectively. In this paper, the sequential pattern is assumed because it is the fundamental pattern, and other patterns can be reduced or converted to it [26]. Also, based on the QoS attribute considered, applied summation and multiplication aggregation functions (see Table 1). The multiplication function can be converted to summation by a logarithmic function [27].

TABLE I. AGGREGATION FUNCTIONS

Aggregation Type	QoS Attribute	Aggregation Function
Summation	Cost	$q_i(s) = \sum_{j=1}^{t} q_i(Z_j)$
	Response Time	
Multiplication	Availability	$q_i(s) = \prod_{j=1}^{t} q_i(Z_j)$
	Reliability	

III. RELATED WORKS

The popularity of cloud services and the rise of marketplaces to trade services require means to harness the multiplicity of services via aggregation and/or customization to meet user's need. Previous works have proposed the use of feature models to capture the variabilities of cloud services and applied automated means to generate valid cloud service offerings. An SPL-based approach for cloud service selection that employs feature models, extended with cardinalities and attributes, to describe the variability in cloud environments has been proposed in [12], [13]. The approach utilizes a domain model to support the consistent configuration of the complete stack of cloud services that comply with user's requirements. In the same line, an approach was presented in [11] to harness cloud service capabilities using variability model. The variability models serve as representation mechanisms and are called Cloud Feature Models (CFMs). CFMs are used to elicit requirements and to perform the filtering operation. To manage the variability among cloud-based applications with support for multiple stakeholders, authors in [14] applied extended feature modeling to configure cloud-based multi-tenant aware applications, by using the model to express the variability in functionality and QoS attributes. The proposed approach manages dynamic configuration in an adaptive staged configuration process capable of adding/removing providers or users from the cloud platform and allows for reconfiguration of variant services as requirements changes.

In these approaches, users are expected to painstakingly configure cloud services, with the assumption that all users are full domain experts. However, a cloud service e-marketplace should among others, provide a real online shopping experience similar to existing ecommerce platforms [3], [6]; where available service offerings are indexed in the e-marketplace service directory, more like a catalogue, and seamlessly updated in a manner completely transparent to the users. The user is shielded from the underlying complexity of performing service configuration, and since all possible alternatives are composed formally [6]. With the proposed approach users would be able to explore other alternatives with respect to their requirements like on ecommerce platforms.

IV. PROBLEM SCENARIO

For the purpose of the paper, the scenario of a Customer Relationship Management as a Service (CRMaaS) ecosystem was envisioned. The envisioned CRMaaS ecosystem involves multiple atomic service providers who collaborate to provision CRM solutions, while prospective small businesses can purchase CRM solutions from the e-marketplace. The components that make up the CRMaaS ecosystem include Contact Management, Database, Marketing, and Social media analysis. The description of each module is as follows: **Contact Management Service:** This is a tool to manage user contacts and communication; including appointment management, task management and scheduling, communication (SMS, email); **Cloud Database:** The cloud-based database system will store user information including user personal data, purchase history, preferences etc.; **Marketing Service:** These are tools for communicating with users; including email marketing, text message marketing, social media marketing, etc.; **Social Media Analytics:** This is a tool that monitors conversations on social media and analyses feedbacks, capturing user sentiments; **Cloud Platform:** The derived valid compositions would require a cloud platform on which to run.

Samples of the constituent services that can fulfill each module, together with the values of the QoS attributes are shown in Table 2, while samples of required and exclude constraints are presented in Table 3. Although other QoS attributes are also important, four QoS attributed was chosen for this scenario for illustrative purposes. The QoS attributes considered includes: availability and reliability, measured in percentages (%); response time measured in milliseconds (ms), while cost is measured in Dollars/month ($/Month). An instance of the CRMaaS offering is a combination of any/all of the atomic services to create a complete CRM solution. On the e-marketplace, multiple variants of CRMaaS solutions exist and are differentiated by QoS factors. A small business can then search for and purchase CRM solution that satisfies their preferences.

The e-marketplace service directory should contain a set of *m* CRM solutions with *n* QoS criteria. With the user's requirements converted into a search query, the e-marketplace is expected to generate results that show the ranking of the composite CRM solutions in relation to user requirements. This arrangement will make it easy for users to find satisfactory service.

TABLE II. SAMPLE CONSTRAINTS ON SERVICE COMPOSITION

CM2	**Excludes**	M1
CD2	**Excludes**	P2
SMA3	**Excludes**	CD2
CM1	**Requires**	P1
CM1	**Requires**	CD1
SMA1	**Requires**	CD2
SMA2	**Requires**	M1

TABLE III. SAMPLE ATOMIC SERVICES TO REALIZE CRMAAS MODULES

CRMaaS Components	**Candidate Services**	**QoS Values**			
		Avail.	***Resp. Time***	***Reliability***	***Cost***
Contact Management	CM1	90	--	90	30.50
	CM2	95	--	67	29.99
	CM3	70	--	40	25.50
	CM4	99	--	79	34.99
Cloud Database	CD1	89	100.22	60	13.50
	CD2	79	50.54	75	20.50
	CD3	97	120.34	80	50.00
Marketing	M1	99	--		55.50
	M2	91	--		59.99
Social Media Analysis	SMA1	90	200.45	88	49.99
	SMA2	95	138.56	90	50.00
	SMA3	85	125.45	79	45.67
Platform	P1	99	300.45	70	199.99
	P2	99	423.10	75	149.99

V. PROPOSED SOLUTION APPROACH

In response to the CRMaaS problem scenario, the proposed solution approach is conceptualized in the steps shown in Fig. 1. First, the atomic services are modeled using extended feature models notations to produce the ecosystem feature model which is transformed into a CSP. Automated reasoning is performed to determine valid compositions aggregated into composite services which are listed on the e-marketplace service directory. Users can find services that match their preferences using service selection mechanisms. The following subsections discussed how the proposed approach addresses the CRMaaS problem scenario.

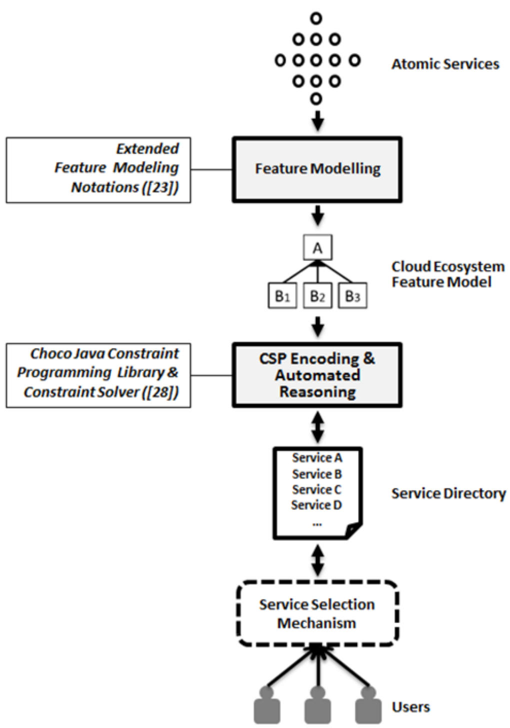

Fig. 1. Process flow of proposed approach.

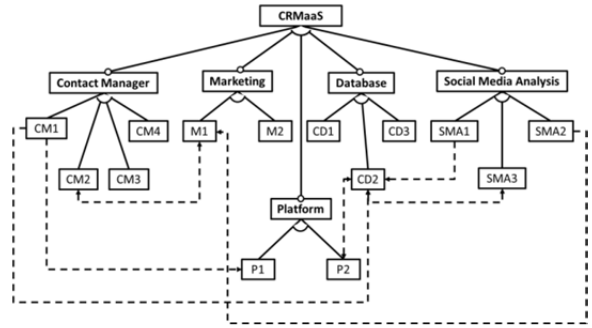

Fig. 2. High-level view of the CRMaaS cloud ecosystem feature model.

A. Modelling the Cloud Service Ecosystem

We modeled the cloud service ecosystem by adopting extended feature models (see Fig. 2), which was christened called Cloud Ecosystem Feature Model (CEFM). All CRMaaS components are mandatory; however, each candidate service is an alternative to other candidate services within the same module group. The CEFM was transformed into CSP based on the rules outlined in [23].

B. Reasoning Engine

We used Choco constraint solver as the constraint-based reasoning engine to perform automated analysis of the CEFM. The solver determines the satisfiability of the CSP, and also searches for a solution in a CSP, using inbuilt search algorithms to generate all the possible combinations of values for each variable in the CSP. Choco solver employs, by default, a backtracking approach to finding solutions. The search is ordered as an enumeration tree and traversed using a Depth-First Search (DFS) algorithm augmented with variable and value selection heuristics [28]. The corresponding CSP representation of the CEFM is read by the reasoning engine and performs automated analysis on the CSP representation to generate all valid service compositions. The overall QoS attribute of all valid combinations was determined by considering the QoS factors of constituent services. This was performed by aggregating the QoS of each atomic service using QoS aggregation functions in Table 1.

C. Service Directory

The service directory indexes all the QoS information about the collection of valid combination services generated by all products operations on the CEFM. The encoding of the CEFM as CSP, together with the aggregation functions were implemented using Java in NetBeans 8.1 based on the constraints provided in the Choco library [28]; the analysis operation performed to generate all products from the CEFM yielded a total 38 valid combinations (see, Table 4), including the constituent atomic services, and the aggregated values for each QoS attributes. The generated composite services are then indexed as the services contained in the cloud service e-marketplace service directory. The indexed list becomes the catalogue, from which users are served recommendations with respect to their QoS requirements.

D. Cloud Service Selection

The requirements for cloud service selection are similar across all approaches as they include: a finite or infinite set of alternatives, at least two evaluation criteria, and a decision maker, in this case, a user; and the goals, include choosing, ranking, or sorting alternatives [29]. Decision making requires that many alternatives be evaluated along some criteria, in order to arrive at the best choice. Selecting a service(s) from a cloud e-marketplace can be regarded as a multi-criteria decision analysis (MCDA) problem [9], [10]. The services listed in the directory share similar functionalities with varied QoS attributes, and user's preference on these attributes defines the utility functions by which each service alternative is evaluated. The decision matrix is defined as:

TABLE IV. VALID COMPOSITIONS GENERATED BY REASONING ENGINE

ID	Constituents Services	Aggregate QoS Values			
		Avail.	*Res. Time*	*Relia.*	*Cost*
1	CM4; CD3; SMA3; M2; P2	98.68	668.89	75.73	340.64
2	CM3; CD3; SMA3; M2; P2	97.16	668.89	72.78	331.15
3	CM4; CD3; SMA3; M2; P1	98.67	546.24	75.43	390.64
4	CM3; CD3; SMA3; M2; P1	97.16	546.24	72.48	381.15
5	CM4; CD1; SMA3; M2; P2	98.29	648.77	74.48	304.14
6	CM3; CD1; SMA3; M2; P2	96.79	648.77	71.53	294.65
7	CM4; CD1; SMA3; M2; P1	98.29	526.12	74.19	354.14
8	CM3; CD1; SMA3; M2; P1	96.79	526.12	71.23	344.65
9	CM2; CD3; SMA3; M2; P2	98.49	668.89	75.02	335.64
10	CM2; CD3; SMA3; M2; P1	98.49	546.24	74.72	385.64
11	CM2; CD1; SMA3; M2; P2	98.11	648.77	73.77	299.14
12	CM2; CD1; SMA3; M2; P1	98.11	526.12	73.47	349.14
13	CM4; CD3; SMA3; M1; P2	99.03	668.89	75.73	336.15
14	CM3; CD3; SMA3; M1; P2	97.53	668.89	72.78	326.66
15	CM4; CD3; SMA2; M1; P2	99.51	682	76.3	340.48
16	CM3; CD3; SMA2; M1; P2	98.01	682	73.34	330.99
17	CM4; CD3; SMA3; M1; P1	99.03	546.24	75.43	386.15
18	CM3; CD3; SMA3; M1; P1	97.53	546.24	72.48	376.66
19	CM4; CD3; SMA2; M1; P1	99.51	559.35	76	390.48
20	CM3; CD3; SMA2; M1; P1	98.01	559.35	73.04	380.99
21	CM4; CD1; SMA3; M1; P2	98.66	648.77	74.48	299.65
22	CM3; CD1; SMA3; M1; P2	97.15	648.77	71.53	290.16
23	CM4; CD1; SMA2; M1; P2	99.14	661.88	75.05	303.98
24	CM3; CD1; SMA2; M1; P2	97.63	661.88	72.1	294.49
25	CM4; CD1; SMA3; M1; P1	98.66	526.12	74.19	349.65
26	CM3; CD1; SMA3; M1; P1	97.15	526.12	71.23	340.16
27	CM4; CD1; SMA2; M1; P1	99.14	539.23	74.75	353.98
28	CM3; CD1; SMA2; M1; P1	97.63	539.23	71.8	344.49
29	CM1; CD1; SMA3; M2; P1	97.88	526.12	74.75	349.65
30	CM1; CD1; SMA3; M1; P1	98.24	526.12	74.75	345.16
31	CM1; CD1; SMA2; M1; P1	98.73	539.23	75.32	349.49
32	CM4; CD2; SMA1; M2; P1	98.02	551.35	75.62	360.46
33	CM3; CD2; SMA1; M2; P1	96.52	551.35	72.67	350.97
34	CM2; CD2; SMA1; M2; P1	97.84	551.35	74.91	355.46
35	CM4; CD2; SMA2; M1; P1	98.62	489.46	75.72	360.98
36	CM3; CD2; SMA2; M1; P1	97.12	489.46	72.76	351.49
37	CM4; CD2; SMA1; M1; P1	98.39	551.35	75.62	355.97
38	CM3; CD2; SMA1; M1; P1	96.88	551.35	72.67	346.48

Definition 2 (Decision Matrix): Let A be m × n Matrix that contain the QoS information of all service $s_i \in S$ where each element $a_{i,j}$ represents the j[th] QoS value of the i[th] service, while $i, j > 2$.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{n2} & \cdots & a_{mn} \end{pmatrix}$$

From A, a row vector would describe a service $s_i \in S$ with QoS attributes where each element represents the QoS attribute of service s_i. Given the decision matrix A, a cloud service selection problem is to determine the optimal service option, A^+ with the highest degree of desirability with respect to all relevant criteria [30]. Therefore, service selection is effectively enabled by matching the representations of user's requirements to the properties of the service offerings; this can be achieved using existing cloud service selection methods such as those proposed in [8]-[10], [30], [31].

VI. DISCUSSION

With other cloud service selection approaches, the user can only select services from a predefined list of atomic services. There are cases in atomic services can alone not satisfy the user's requirement. In such cases, manual configuration of atomic will increases the cognitive demand on the user. However, the proposed approach aggregates atomic services in a manner that satisfies the constraints between these services and generates a list of potential composite offerings, which forms the basis of cloud service selection. While satisfying complex user requirements, the proposed approach reduces the cognitive load in the process of cloud service selection by affording minimal user involvement; thereby replicating the kind of user experience that is currently available on ecommerce platforms in cloud service e-marketplaces.

Furthermore, the automated analysis reveals a number of useful information about the ecosystem. For example, the e-marketplace provider may be interested in the number of composite services that can be offered based on the number of participating atomic services. The provider can also determine those atomic services that will not fully benefit from the value-chain of the ecosystem (partly or fully due to their presence in a few or none of the likely compositions), and advise accordingly.

In addition, the proposed approach encourages variability in the ecosystem as multiple functionally equivalent atomic services can collaborate in service provisioning. Small-scale service providers will benefit from the proposed approach, by participating in the global market of e-services ecosystem; thus promoting their profitability by multiplying their revenue and impact. The use of a structured model and automated analysis will also enable these providers to estimate the profitability of their services by the number of compositions their services are a part of, and use this discovery to position their offerings for better competitiveness in the ecosystem.

Besides, the proposed approach makes it easier to accommodate new services in a manner that is seamless and natural to an ecommerce platform, with little or no disruption to e-marketplace operations. The proposed approach can effectively handle the scenarios of new entrants and exits of services into and from the ecosystem. With each case of entrants or exits based on the stated entrance and exit policies of the e-marketplace, such that if the feature model is altered; a seamless automated update of the e-marketplace service directory can be still achieved. This presupposes that service registration and disengagement from the ecosystem is performed offline, not at request time, giving the proposed approach the scalability advantages in the event of multiple concurrents exists and entrants.

VII. Conclusion

A cloud e-marketplace is an ecosystem of atomic services from multiple sources. The different ways in which these services are aggregated creates a plethora of potential offerings that can meet diverse business needs of users. In this paper, a constraint-based approach was proposed to perform formal composition of atomic services to satisfy complex business process in an e-marketplace. Extended feature model was employed to explicitly capture the services, their QoS attributes, and the cross-service relationships and constraints in a logical and structural manner as part of an ecosystem. This model was used to determine blueprints to consistently generate valid compositions that are available on the e-marketplace for users to purchase. With the aid of an example, the applicability of the proposed approach was demonstrated by showing how atomic services in a CRM service ecosystem are aggregated and made available to users via the e-marketplace platform. The main contribution of this paper is that it presents a plausible approach for enabling formal composition of atomic services in order to respond to complex user requirements. By so doing it presents an opportunity to replicate the kind of user experience that is currently available in the ecommerce platforms, which existing cloud service e-marketplaces do not yet offer.

Since constraint solvers have the ability to analyze numeric or text-like attributes, the proposed approach will be improved to cater for qualitative QoS attributes like security, user-friendliness and eco-friendliness etc. whose values are qualifier tags. The proposed approach will be integrated into a service selection framework to improve the user experience of the cloud service e-marketplace environment in the near future.

References

[1] M. Papazoglou and W. van den Heuvel, "Blueprinting the cloud," IEEE Internet Computing, pp. 74-79, 2011.

[2] R. Buyya, C. Yeo, and S. Venugopal, "Market-oriented cloud computing," in Proc. of the 10th IEEE Int. Conf. on High Performance Computing and Communications (HPCC'08), 2008, pp. 5-13.

[3] A. Menychtas et al., "4CaaSt marketplace: An advanced business environment for trading cloud services," Future Generation Computer Systems, pp. 104–120, 2014.

[4] A. Barros and M. Dumas, "The rise of web service ecosystem," IT Professional, vol. 8, no. 5, pp. 31-37, 2006.

[5] A. Gatzioura, A. Menychtas, V. Moulos, and T. Varvarigou, "Incorporating business intelligence in cloud marketplaces," 10th Int. Symp. Parallel and Distributed Processing with Applications, 2012, pp. 466-472.

[6] R. Akolkar et al., "The future of service marketplaces in the cloud," in IEEE 8th World Congr. on Services, 2012, pp. 262-269.

[7] Y. Syu, S. Ma, J. Kuo, and Y. FanJiang, "A survey on automated service composition methods and related techniques ," in IEEE 9th Int. Conf. on Services Computing, 2012 , pp. 290-297.

[8] Y. Syu, S. Ma, J. Kuo, and Y. FanJiang, "A survey on automated service composition methods and related techniques ," in IEEE 9th Int. Conf. on Services Computing, 2012 , pp. 290-297.

[9] Y. Syu, S. Ma, J. Kuo, and Y. FanJiang, "A survey on automated service composition methods and related techniques ," in IEEE 9th Int. Conf. on Services Computing, 2012 , pp. 290-297.

[10] Z. Rehman, F. Hussain, and O. Hussainz, "Towards multi-criteria cloud service selection," in International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011, pp. 44-48.

[11] E. Wittern, J. Kuhlenkamp, and M. Menzel, "Cloud service selection based on variability modeling," Service-Oriented Computing, pp. 127-141, 2012.

[12] C. Quinton, D. Romero, and L. Duchien, "Automated selection and configuration of cloud environments using software product lines principles," in IEEE 7th Int. Conf. on Cloud Computing, 2014, pp. 144-151.

[13] C. Quinton, N. Haderer, R. Rouvoy, and L. Duchien, "Towards multi-cloud configurations using feature models and ontologies," in Proc. Int. workshop on Multi-cloud applications and federated clouds, 2013, pp. 21-26.

[14] J. Schroeter, P. Mucha, M. Muth, K. Jugel, and M. Lochau, "Dynamic configuration management of cloud-based applications," in Proc. 16th Int. SPLC -Volume 2, 2012, pp. 171-178.

[15] Y. Bakos, "The emerging role of electronic marketplaces on the Internet," Communications of the ACM, vol. 41, no. 8, pp. 35-42, 1998.

[16] A. Akingbesote, M. Adigun, E. Jembere, M. Othman, and I. Ajayi, "Determination of optimal service level in cloud e-marketplaces based on service offering delay," in Int. Conf. Computer, Communications, and Control Technology, Malaysia, 2014, pp. 283-288.

[17] R. Vigne, W. Mach, and E. Schikuta, "Towards a smart webservice marketplace," IEEE 15th Conf. Business Informatics, 2013, pp. 208-215.

[18] J. Schulz-Hofen, "Web Service Middleware - an infrastructure for near future real life web service ecosystems," in Int. Conf. Service-Oriented Computing and Applications, 2007.

[19] E. Cavalcante et al., "Optimizing services selection in a cloud multiplatform scenario," in IEEE Latin America Conf. on Cloud Computing and Communications, 2012, pp. 31-36.

[20] S. Baek, K. Kim, and J. Altmann, "Role of platform providers in service networks: The case of salesforce.com App Exchange," in IEEE 16th Conf. Business Informatics, 2014, pp. 39-45.

[21] A. Hubaux et al., "Unifying software and product configuration: A research roadmap," in Proc. Workshop on Configuration, France, 2012, pp. 31-35.

[22] T. Berger et al., "Variability mechanisms in software ecosystems," Information and Software Technology, vol. 56, pp. 1520-1535, 2014.

[23] D. Benavides, S. Segura, and A. Ruiz-Cortes, "Automated analysis of feature models 20 years later: A literature review," Information Systems, pp. 615-636, 2010.

[24] K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson, “Feature–Oriented Domain Analysis (FODA) Feasibility”, November 1990.

[25] B. Mohabbati et al., "A quality aggregation model for service-oriented software product lines based on variability and composition patterns," Service-Oriented Computing, pp. 436-451, 2011.

[26] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for QoS-based web service composition," in Proc. 19th Int. conf. World wide web, 2010, pp. 11-20.

[27] J. Li, X. Zheng, S. Chen, W. Song, and D. Chen, "An efficient and reliable approach for quality-of-service-aware service composition," Information Sciences, vol. 269, pp. 238-254, 2014.

[28] N. Jussien, G. Rochart, and X. Lorca, "Choco: an open source java constraint programming library," in Workshop on Open-Source Software for Integer and Contraint Programming, 2008, pp. 1-10.

[29] M. Whaiduzzaman et al., "Cloud service selection using multicriteria decision analysis," The Scientific World Journal, 2014.

[30] Z. Rehman, O. Hussain, and F. Hussain, "IaaS cloud selection using MCDM methods," in IEEE 9th Int. Conf. e-Business Engineering, 2012, pp. 246-251.

[31] R. Mirmotalebi, C. Ding, and C. Chi, "Modeling user's non-functional preferences for personalized service ranking," in Service-Oriented Computing, 2012, pp. 359-373.