Journal of Physics: Conference Series

PAPER • OPEN ACCESS

A recursive pattern recognition approach to selection web services in cloud environment

A recursive pattern recognition approach to selection web services in cloud environment

M A Adarme[1], M A Jimeno[2], and E G Puerto[1]
[1] Universidad Francisco de Paula Santander, San José de Cúcuta, Colombia
[2] Universidad del Norte, Barranquilla, Colombia

E-mail: madarme@ufps.edu.co, majimeno@uninorte.edu.co

Abstract. There are optimization problems in the Cloud for the selection of web services, due to the large number of services available by different cloud providers and the diversity of quality of service parameters of each of them. This work proposes the adaptation of a pattern recognition model based on the systematic functioning of the brain called Ar2p for the selection of web services in composition activities in Cloud environments. The web serice are represented as patterns to be recognized by Ar2p, which determines the necessary and sufficient web services that constitute the composition of services that meet its functional and non-functional requirements. The services composition and activity selection have been formalized through a logical-mathematical model of web service recognition mechanisms in two steps, one that describes the syntactic search of the service and the second, which offers filtering through quality of service parameters. An adaptive implementation of the final model allows its recognition modules to be provided with any desired optimization strategy.

1. Introduction

Systems based on web services composition (WSC) rules were originally built to be used within a single organization, for a single domain, and a specific application, such as commercial transactional systems. Nowadays, they have been extended to be used in multi-organizational environments (heterogeneous and large-scale domains), this has been possible thanks, in part, to the fact that web services, whose applications are based on the service oriented architecture (SOA) paradigm and they are able to use Cloud computing resources.

The dynamical environment of cloud resources means that the task of composing services is not carried out in a static manner as is usually the case in conventional SOA, since in the cloud the composition depends on parameters for the use of computational resources that, although they are by nature elastic, there are considerations of use [1] that involve hardware as a service (I/O, memory and CPU), in this respect some services will consume more resources than others, generating an inconvenience of monetary costs (monetary increase) and hardware disposition [2]. Tasks such as the search, negotiation and monitoring of a web service (WS) are very important, and consequently there may be non-functional requirements that must be taken into account and that must support cloud computing environments with quality parameters that ensure compliance with the requirements of applicants and suppliers, as well as the ability to establish agreements on the provision and acquisition of computing resources through static or dynamic negotiations if the composition requires it.

Although customer growth has been exponential in this model, the problem lies in the amount of WS that can be composed to meet a particular requirement [3], it is difficult for users to analyze the options that suit their needs, even more so when implementing software under this model.

Problems with the selecting and searching for WS have been widely studied in several investigations [3-5] some of them are: High algorithmic complexity in search processes due to the considerable number of SWs that have similar functionalities, different WS selection heuristics that must infer the functionality of a service when ambiguity exists in the description and handling of incomplete information in the web context, different forms of solution integration that allow giving the same result for different models of WSC and selection processes that must take into account non-functional requirements in dynamic environments.

The objective of this work is to present a strategy for the selection of SW based on recursive pattern recognition algorithm called Ar2p [6], a model is established that allows: recognizing and filtering those WS with ambiguity and similarity in the information of their descriptors, and selecting through a heuristic pattern recognition the WS that satisfy functional and non-functional requirements of the WSC.

2. Compositing web services on cloud

The WS selection in cloud computing is denoted as a nondeterministic polynomial time (NP) [7] problem since the WS are highly configurable components, from the definition phase to the execution stage, becoming dynamic scenarios par excellence,this means that the composition of an application can be modeled from the static or the dynamic, understanding each level from the perspective of the creation of the WSC model (workflows) with WS already known or selected with reflective processes or semantic processes in execution time and managed automatically [8].

WS are described by providers through service level agreement (SLA), it is there where the provider defines the characteristics of use of the service, its quality parameters and its rules of negotiation (benefit agreements when one of the parties does not fulfill its role). In [9] it is highlighted how in Cloud environments there is a problem of standardization of the SLA and therefore their difficulty when negotiating and defining characteristics necessary for the correct execution of a WSC.

Composition then requires that the services involved in it is under an appropriate quality of service (QoS) seen from the WS and WSC level, with the appropriate management of the SLAs containing the set of non-functional requirements that each service would need for its composition becoming relevant. At the current time the solution heuristics and optimization of composition services have been studied and converge in the efficient administration of the SLAs [9-10], however there are inconveniences to guarantee QoS in the life cycle of the composition, a resource for example may not adapt to a particular service making its execution environment change and therefore affect QoS parameters and see its committed workflow.

Composition based systems therefore require the ability to adapt to the conditions that a QoS parameter requires. Cloud composition platforms and providers are varied, where it is necessary that a proposed solution for composition challenges be as transparent as possible without affecting the performance of them whose searching and selecting problems are addressed so that the execution engines increase their performance when creating the WSC.

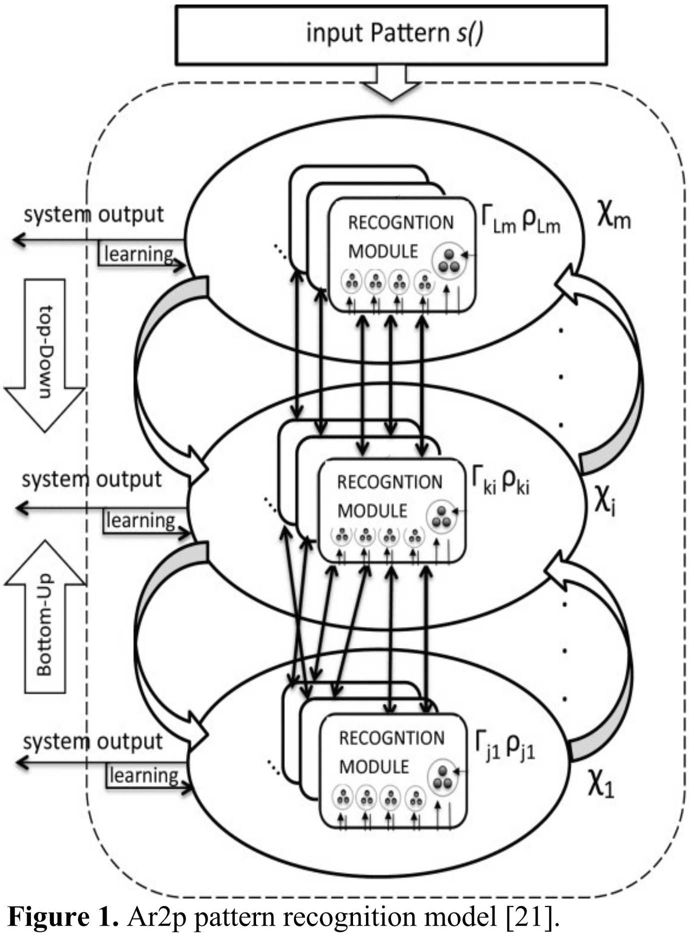
3. Related works

Research in the field of web services compositions has grown considerably, a sample of which can be seen in the works [8,11] where they are related in three large study groups: middleware solutions, frameworks and groups concerning compositions made through frontend abstract models that interact with composition engines, these are mainly based on pure SOA environments. However, for the case study for this research, a searching range is taken to include only studies dealing with composition and selection in cloud environments. In the work presented in [12] develops a heuristic of service selection based on multicloud, based on a combinatorial optimization process carried out through the Greedy-WSC algorithm and an algorithm based on ant colony optimization applied to web service compositions in cloud computing. In [13] they developed a model taking into account dynamic QoS where they monitor the quality of service values of each SW present in the composition, as well as using a strategy using genetic algorithms to search and select the SW, their results are compared with proposals of exhaustive search algorithms and random selection algorithms. A work to be highlighted by the use of a data model for the representation of clouds, providers and services, can be found in [14], it establishes the representation of clouds and SW to perform a preliminary search according to the QoS parameter that must meet both the service and the composition, five QoS metrics are established and with these the system solves its optimization problem through entire programming.

Qingtao, *et al.* [15] proposes a selection method in WSC based on the hidden markov model, whose function is to predict the necessary services on a composition and where they are located in a multinube environment, its importance resides in the fact of being able to make compositions not only with atomic services but also compounds, its system has the function of predicting the total quality of the composition and balancing it in such a way as to ensure that the services are optimal for the different composition models. Wang, *et al.* [16] implements a selection and composition strategy based on the skyline operator for the SW filtering process and the composition is done through a particle optimization algorithm (swarp) that allows to have a finite set of solutions for which a WS is a candidate on a composition.

4. Pattern recognition

Computational pattern recognition is referred to as a set of algorithms and techniques whose purpose is to discover and recognize an entity (physical or its representation in a digital model) through elementary characteristics that allow them to be classified for later processing with recognized entities [17,18]. Recognition patterns have been widely studied [19,20], mainly the papers point to selection methods based on predefined characteristics over a data set that is usually of considerable size.



χ_m: recognition space of more higher-level.
Γ_{Lm}: is the L-module of m-level which recognize ρ_{Lm}.
ρ_{Lm}: It is the recognized pattern by the module L at level m.

χ_i: recognition space of i-level.
Γ_{ki}: is the k-module of i-level which recognize ρ_{ki}.
ρ_{ki}: It is the recognized pattern by the module k at level i.

χ_1: recognition space of 1-level.
Γ_{j1}: is the j-module of 1-level which recognize ρ_{j1}.
ρ_{j1}: It is the recognized pattern by the module j at level 1.

Figure 1. Ar2p pattern recognition model [21].

Ar2p [21] is a recognition model based on the systematic functioning of the brain, this work proposes its adaptation of a recognition model for the selection of WS in composition activities on cloud environments. Ar2p is developed from the mind theory of pattern recognition through which memory handles a hierarchy of patterns and these are perceived through our senses conceptually called" sensors", when one of our senses is activated, the pattern is recognized.

The Ar2p general model is shown in Figure 1, each oval represents a recognition module, this way there are several levels that can recognize atomic patterns (characteristics with only one attribute/value) in the X_i levels, likewise, the X_m levels would have the functionality of recognizing complex patterns derived from the learning of the sub-levels. Each recognition module aims to recognize its corresponding pattern; that is, a copy of the pattern in the real world in such a way that recognition can occur at any level. Ar2p has two strategies to recognize a pattern, one by key signals and the second by partial recognition of signals. The first strategy uses the weight of importance of the input signals identified as keys, and the second uses the partial or total presence of the signals, in this way, for example, a pattern can be recognized by a single key signal or by the sum of the partial signals, if and only if they pass an observation space. Ar2p's advantage is its level of uniformity as it offers that each function the underlying recognition modules is the same and this process runs recursive.

5. Recursive pattern recognition to selection web services

Service-composition is based on the atomic WS involved in the composition model, and algorithms are needed to ensure the correct coupling and operation of the WS, each of which has different quality-conditions, so the selection of services that meet the non-functional requirements demanded by the entities is the objective for the selection of WS.Taking Ar2p's provided base, the following three theorems are defined:

Theorem 1. A WSC is defined as an abstract model on a directed acyclic graph, where each node represents a WS, the relationship of each node is made by its edges that represent input and return parameters. Each node has a set of QoS parameters, it must be established that WS are the necessary ones to satisfy the abstract model of composition.

Theorem 2. Recognition through key signals. A key signal activates a recognizing module when a value is greater than or equal to the value defined in a desired QoS parameter in each of the nodes.

Theorem 3. Recognition through partial signals. A pattern is recognized if the QoS value lies within the range of the quality acceptance threshold.

Formal specification of the operating activities data is established for the selection:

- WS Providers: A provider defines its services in form of a service specification file (WSF): WSF= {WS_1, WS_2, WS_3, ..., WS_n} where each WS_i corresponds to the WS deployed in that provider.
- Cloud (C): A Cloud may contain a finite number of providers that expose their WSFs, in a form: C = {WSF_1, WSF_2, WSF_3, ..., WSF_n}, where "n" is the number of WS providers.
- WS: WS is defined as a set of input (I), output (O) data and their non-functional requirements (RNF). The specification of this data is given according to WS standards and they are easily identifiable in the registration directories of each provider. Therefore, a WS is a 3-tuple < I, O, |RNF| > and |RNF| can be an RNF vector.
- SLA: it defines agreements to use the services between the client and the supplier, its function is to specify and clarify the performance expectations, establish responsibility, and specify options and outcomes if the performance or quality of service are not agreed by both parties.

- Therefore, an SLA should be complied with for each WSC and these SLAs are defined as: SLA={RNF_1, RNF_2, RNF_3, RNF_4, RNF_5}, where RNF are the nonfunctional requirements of each service and of the WSF and m the cardinality of the set of RNFs for the WSF whose maximum value will be five (5). These RNF are contained in the 3-tuple of the WS definition. Quality parameters used are taken from the research presented in [14].

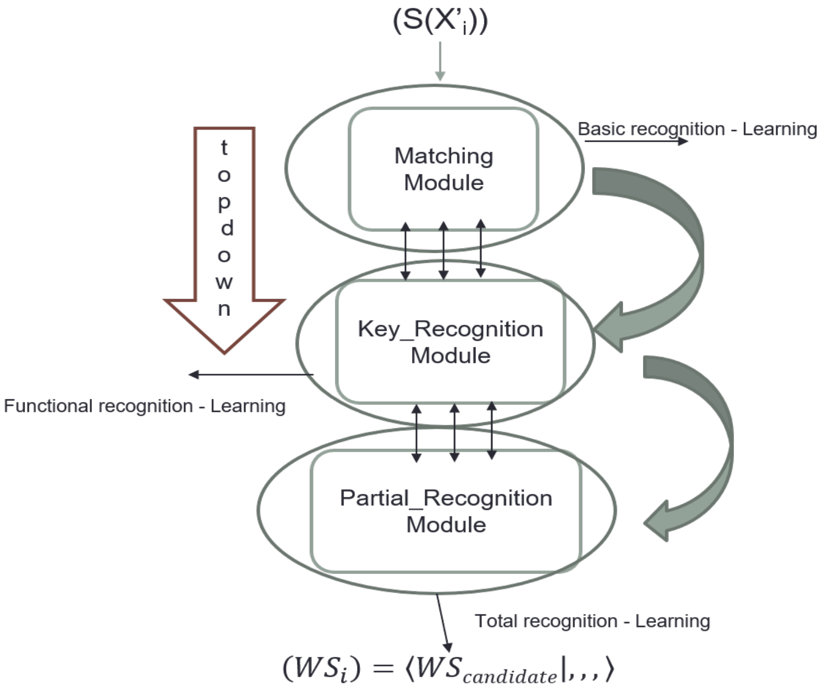

$(S(X'_i))$
Basic recognition - Learning
t o p d o w n
Matching Module
Key_Recognition Module
Functional recognition - Learning
Partial_Recognition Module
Total recognition - Learning
$(WS_i) = \langle WS_{candidate}|,,,\rangle$

Figure 2. Pattern recognition system for WSC.

Figure 2 shows the system of pattern recognition for web services. System is top-down, and its way of learning is by reinforcement, *i.e.* each WS must achieve the objectives or rewards of each recognizing module. However, the system can recognize in three levels, being the functional and the total those used in an abstract model of composition. Targets can be achieved in three ways: first, basic recognition consisting only of those WS whose description of services corresponds syntactically to a WS of some repository defined in group C; second, functional recognition, which corresponds to WSs that are recognized by key signals and can be used directly in the composition; and third, total recognition, at this level the composition model will achieve to satisfaction all WSs with their corresponding quality parameters. Modules architecture allows to operate with different strategies and optimization heuristics, that is, for the syntactic recognition module, any string recognition algorithm can be implemented. It should be noted that the key and partial recognition modules can be interchanged in their selection process. A general recognition algorithm (Algorithm 1) is described to follow.

Algorithm 1 works in the following way, the system receives a pattern with the registration of the data described in Figure 2, line 2 loads in memory the information of the WS and the abstract composition model, and each node will have its own recognition process; the process starts with the syntactic recognition of the service to filter and optimize the data that will be processed by the following recognition modules (line 4). Results obtained by this first filter will be processed by the key recognition module (line 7); however, it may be the case that the system only has partial recognition in which case the process would move to line 15. The recognition modules process is simile (8 line up to 14 line), for each WS incident information it is calculated if it complies with the optimal quality parameter, in order for the WS to be recognized it must pass through all the key signals that are defined for that particular node. Partial recognition selection process only changes that for its recognition only needs to activate signals that pass at a pre-defined quality threshold.

Algorithm 1. General recognition algorithm.

```
Input: S(Xi)
load_data S(Xi)
L=generate_list_WS(y)
if Recognize_syntactically(L)
    create WS candidates on LC
    Update pattern information -Partial learning
        if |key sign| >0
            for key_sign to
                if Recognizing_key_signal (i, LC)
                    Create WS candidates on LC - Updating Partial
                    Pattern Information
            end_for
            if |LC'| == |key sign|
                Create WS Applicants on LC_keysii. Update pattern information -Learning with
                functional recognition.
        else
            LC_keys=LC
        if |partial_signal| >0
            for key_partial to
            if Recognizing_key_signal(i, LC_keys)
                Create WS Applicants on LC and Update partial pattern information
                Update pattern information -Learning with functional
Create WS_list with LC_keys
Sending recognition to a higher level
end
```

6. Conclusion

WS selection strategy presented offers a dynamic model for the recognition of services over service involved in the composition. Ar2p as the basis of the solution offers learning by reinforcement in three basic levels, whose partial results could be used in a system with non-restrictive QoS parameters. Recursive behavior of the model makes the implementation of the modules at the software development level fast and easily scalable with the use of any optimization strategy that could be applied on each recognizing module. Ar2p's characteristic functioning as a biological model of the neocortex offers a way of dividing the problem of composition selection into functional and descriptive tasks, whose conception of solution can be adapted to other search models.

References

[1] Sheng Q Z, Qiao X, Vasilakos A V, Szabo C, Bourne S, and Xu X 2014 Web services composition: A decade's overview *Information Science* **280** 218

[2] Goettelmann E, Fdhila W, and Godart C 2013 Partitioning and cloud deployment of composite web services under security constraints in Cloud Engineering *IEEE International Conference on Cloud Engineering (IC2E)* (Redwood: IEEE)

[3] Guzmán Luna J A and Ovalle Carranza D A 2008 Composición de servicios: Una aplicación de la web semántica y las técnicas de planificación automática *Ingeniería e Investigación* **28(3)** 145

[4] Wu C-S and Khoury I 2012 Tree-based search algorithm for web service composition in SaaS *Ninth International Conference on Information Technology - New Generations (ITNG)* (Las Vegas: IEEE)

[5] Gabrel V, Manouvrier M and Murat C 2014 Web services composition: Complexity and models *Discrete Applied Mathematics* **196** 1

[6] Puerto Cuadros E G and Aguilar Castro J L 2017 Un algoritmo recursivo de reconocimiento de patrones *Revista Técnica de la Facultad de Ingenieria Universidad del Zulia* **40(2)** 95

[7] Jula A, Sundararajan E, and Othman Z 2014 Cloud computing service composition: A systematic literature review *Expert Systems with Applications* **41(8)** 3809

[8] Zou G, Gan Y, Chen Y, and Zhang B 2014 Dynamic composition of web services using efficient planners in large-scale service repository *Knowledge-Based System* **62** 98

[9] Sundareswaran S, Squicciarini A, and Lin D 2012 A brokerage-based approach for cloud service selection in Cloud Computing *IEEE 5th International Conference on Cloud Computing (CLOUD 2012)* (Honolulu: IEEE)

[10] Garg S K, Versteeg S, and Buyya R 2011 Smicloud: A framework for comparing and ranking cloud services *Fourth IEEE International Conference on Utility and Cloud Computing (UCC)* (Victoria: IEEE)

[11] Cardinale Y and Rukoz M 2011 A framework for reliable execution of transactional composite web services *Proceedings of the International Conference on Management of Emergent Digital Ecosystems (MEDES '11)* (California: Association for Computing Machinery)

[12] Yu Q, Chen L, and Li B 2015 Ant colony optimization applied to web service compositions in cloud computing *Computers and Electrical Engineering* **41** 18

[13] Ye Z, Zhou X and Bouguettaya A 2011 Genetic algorithm based QoS-aware service compositions in cloud computing *International Conference on Database Systems for Advanced Applications* (Berlin: Springer)

[14] Wang S, Zheng Z, Sun Q, Zou H and Yang F 2011 Cloud model for service selection *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (Shanghai: IEEE)

[15] Wu Q, Zhang M, Zheng R, Lou Y and Wei W 2013 A QoS-satisfied prediction model for cloud-service composition based on a hidden Markov model *Mathematical Problems in Engineering* **2013** 387083

[16] Wang S, Sun Q, Zou H and Yang F 2013 Particle swarm optimization with skyline operator for fast cloud-based web service composition *Mobile Networks and Applications* **18(1)** 116

[17] Nilsson N J 1996 *Introduction to Machine Learning: An Early Draft of a Proposed Textbook* (Stanford: Stanford University)

[18] Duda R O, Hart P E and Stork D G 1997 *Pattern Classification* (California: Jhon Wiley & Sons)

[19] Chandrashekar G and Sahin F 2014 A survey on feature selection methods *Computers and Electrical Engineering* **40(1)** 16

[20] Khalid S, Khalil T and Nasreen S 2014 A survey of feature selection and feature extraction techniques in machine learning *Science and Information Conference (SIC)* (London: IEEE)

[21] Puerto E and Aguilar J 2016 Extended abstract: Formal description of a pattern for a recursive process of recognition *IEEE Latin American Conference on Computational Intelligence (LA-CCI)* (Cartagena: IEEE)