Cloud Package Selection for Academic Requirements using Multi Criteria Decision Making based Modified Ant Colony Optimization Technique

C.Madhumathi [1], Gopinath Ganapathy[2]
School of Computer Science, Engineering and Applications
Bharathidasan University, Tiruchirapalli, TamilNadu 620 023, India.
[1]madhu.csbdu54@gmail.com
[2]gganapathy@gmail.com

Abstract— Quality of Service (QoS) and user satisfaction are two of the major requirements considered by the current cloud service providers. In-order to incorporate these qualities in the cloud resource selection framework, user's requirements must be clearly known. This paper presents an effective cloud package allocation technique that utilizes the user's logs and fuzzy user inputs to identify the user requirements to perform optimal allocations. Since cloud packages are predefined and do not correspond to the direct user requirements, optimal package allocation is the only option. This process is carried out by Ant Colony Optimization (ACO). Due to the metaheuristic nature of ACO, the results obtained from this selection technique was found to be optimal and the results were obtained faster even with the usage of a large number of agents (ants). Experiments show that ACO provides optimal and fast allocations.

Keyword-Cloud Package Selection, Resource Provisioning, MCDM, ACO, Optimization

I. INTRODUCTION

Cloud computing has its basic operations on identifying the best resource suiting the needs of the user and paying the service provider as per the usage levels. The services that are provided by cloud platforms include Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [1]. Different providers offer these services at different levels and the same provider might also offer services with varied configurations described by the Service Level Agreements [2]. Hence it becomes the responsibility of the user to identify the appropriate service suiting their needs [3]. Optimal service selections are of paramount importance in a cloud platform. Though the infinite scalability provided by the cloud service providers is an advantage, this pattern of operation tends to increase the cost for the user. It is always more costly to use a service on-demand rather than using it on dedicated basis. Hence the concept of quality awareness during the selection of services is mandatory. Resource allocation in cloud environments are challenging due to two major reasons [4,5,6]; dynamic resources extended over a wide geographical region and on demand scalability. Though some of these are not applicable in the academic scenario, several other challenges exists in the process of provisioning resources. This paper discusses the issues and presents a framework to perform effective optimization of resource allocation in cloud environments.

II. RELATED WORKS

Optimal resource identification in cloud environment is of major importance and hence several techniques have been proposed in-order to perform effective utilization of resources. A resource provisioning technique based on the perspective of QoS was presented by Singh et al. in [7]. This technique operated by clustering the resources on the basis of workloads and then re-clustering them on the basis of QoS requirements. Scheduling is performed based on several scheduling policies and the final resource allocation is performed. A distributed resource provisioning technique was proposed by Andres et al. in [8], that identifies over utilization of resources and allocates appropriate resources. A deadline based resource allocation technique that can be used for dynamic reallocation of existing workloads is presented by Christian et al. in [9]. A decision tree based provisioning technique was proposed by Nikolas et al in [10]. A large scale high throughput computing based cloud provisioning technique was presented by Kim et al. in [11]. This technique uses statistical methods to operate on job traces to identify appropriate workloads. The most influential features are picked and these features are utilized to improve the performance of the virtual machines during the allocation phases. Similar scientific applications based cloud provisioning techniques include [12-15]. A preference based service selection technique for cloud brokers is presented by Patiniotakis et al. in [16]. This technique uses a holistic multi criteria decision making approach to identify services and also uses precise and imprecise metrics to deal with the fuzziness. User based cloud service selection techniques [17,18,19] are also on the raise due to the increase in the user preference incorporation for effective and optimal resource allocations.

III. Cloud Package Selection for Academic Requirements using Multi Criteria Decision Making based Modified Ant Colony Optimization Technique

Cloud package selection is one of the major optimization scenarios faced by cloud users. The process of identifying the requirements is usually difficult, however, in the academic scenario, this becomes an easy process due to the predefined timings and days of operation. Having clear requirement parameters will only solve half of the issue. Cloud services are not fine tunable. The service providers usually provide service packages from which the user is required to select the optimal package that best suits the needs of the user. An academic requirement based package selection method using Multi Criteria Decision Making (MCDM) based Modified Ant Colony Optimization (ACO) is presented in this paper. The framework of operations of the proposed method is presented in Figure.

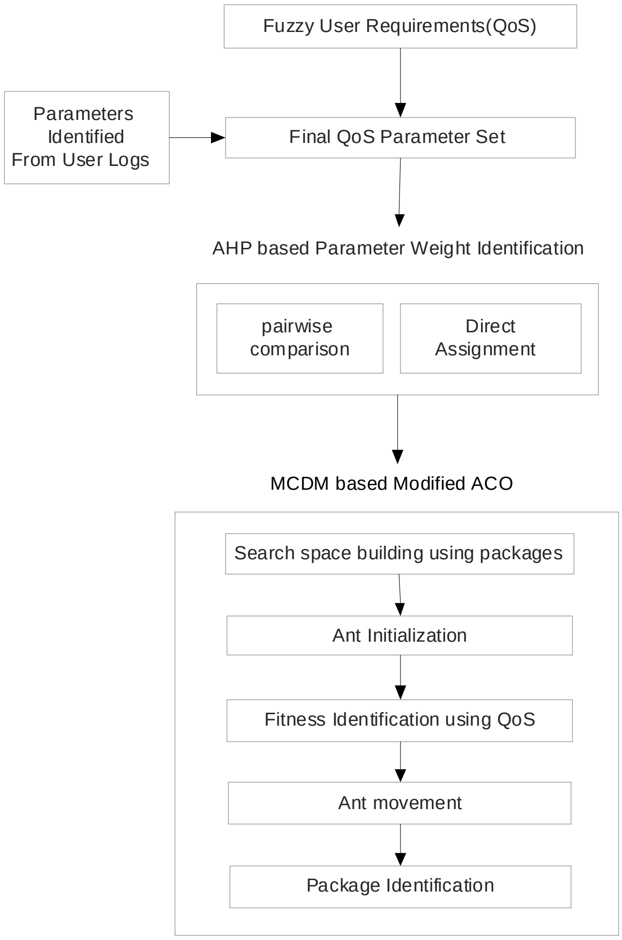

Fig. 1 Cloud Package Selection for Academic Requirements using Multi Criteria Decision Making based Modified Ant Colony Optimization – Framework

The package selection process is divided into three distinct phases, namely; parameter set identification, parameter ordering and Modified ACO based package selection.

A. Parameter Set Identification

The process of identifying the required parameters plays a major role in the package selection process. This approach uses 14 QoS parameters for the process of package selection (Table 1). It is assumed that the packages and the requirement specifications contain data for all the 14 parameters.

TABLE. 1. QoS Parameters Considered for Evaluation

QoS Parameters Considered for Evaluation	Obtained From
Bandwidth (Bw)	**Web Logs**
Computation Capability (CC)	
Availability (Av)	
Correctness (Cr)	
Usability (Us)	
Reliability (Re)	
Variable computation load (Vc)	
Serviceability (Se)	
Latency (l)	
Security (S)	**User Input**
Portability (P)	
Reliable storage (Rs)	
Data Backup (Db)	
Customization (Cu)	

Our previous contribution dealt with the process of identifying SMI parameters, and in turn the QoS requirements of an academic organization using their web logs. It should be noted that not all the parameters can be obtained from the web logs. Several parameters cannot be identified, and hence should be obtained as input from the administrator of the organization. Identifying the actual values is complicated, as the administrator might be unaware of such detailed specifications. Hence fuzzified requirements are obtained from the organization. The user provides values between 1 and 5, where 1 being the lowest and 5 the highest. This provides a complete list of QoS requirements for a single organization.

B. Parameter Ordering

In ACO, parameter importance is of vital importance in identifying the fitness of nodes. Not all users exhibit the same level of importance for the QoS parameters, hence the importance provided to the QoS parameters is obtained from the user using Analytic Hierarchy Processing (AHP).

AHP is a technique that can be utilized to make complex decisions on the basis of mathematics and user preferences. It can be utilized in a variety of situations requiring the user to make decisions on the basis of several dependent or independent attributes. Two methods of identifying weights using AHP [20,21] includes; pairwise comparison and direct user assigned weights.

Pair wise Comparison

Every pair of attributes are ranked with respect to each other. This is helpful if the ranking parameters contain too many attributes or if the user does not possess detailed knowledge about the problem at hand. Since the process of cloud package selection satisfies both these properties, AHP is the best candidate for operating on the QoS requirements in-order to rank them. Comparisons are made on a 5 point scale and the comparison matrix depicting the priority set P is as follows

$$P = \begin{bmatrix} w_1/w_1 & w_1/w_2 & \dots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{bmatrix} = \begin{bmatrix} 1 & s_{12} & \cdots & s_{1n} \\ s_{21} & 1 & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & s_{n2} & \cdots & 1 \end{bmatrix}$$

Where S_{xy} represents the comparison score of x when compared with y and W_x/W_y compares the attribute x against y. An attribute when compared to itself will return a value of 1. These values are then integrated to provide the final attribute weights (WA_i).

Direct User Assigned Weights

Several parameters might be of understandable importance, hence the user might be capable of providing the weight directly rather than using the pairwise comparison technique. These correspond to the direct user assigned weights and are not considered for pairwise comparison.

C. Modified ACO based Package Selection

The final phase is the identification of packages using modified ACO. Ant Colony Optimization [22,23] is a metaheuristic technique used to solve optimization problems. They work on the basis of ant movement on a search space built by the probable solutions in hand. Search space in the proposed approach is created by considering each package provided by the cloud provider as the node. Fitness of a node is identified from the probability of its selection level. The probability is identified by

$$p_{ij}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}.\left[\eta_{ij}\right]^{\beta}}{\sum_{j=1}^{n}\left[\tau_{ij}(t)\right]^{\alpha}.\left[\eta_{ij}\right]^{\beta}}$$

where m and n are the number of ants and the number of probable neighbor nodes, τ_{ij} is the pheromone intensity in the edge *ij* and η_{ij} represents the visibility range of the edge *ij*. α and β represents the importance of the pheromone trial and the visibility respectively.

The ACO proposed by Dorigo utilizes two parameters namely the trail intensity and the visibility. However, the current application requires several parameters, hence the fitness function of ACO is modified as,

$$p_{ij}(t) = 1 - \left(\frac{[Bw]^{\omega 1}.[cc]^{\omega 2}.[Av]^{\omega 3}.[Cr]^{\omega 4}.[Us]^{\omega 5}.[R]^{\omega 6}.(1/[Vc]^{\omega 7})}{\sum_{j=1}^{n}[Bw]^{\omega 1}.[cc]^{\omega 2}.[Av]^{\omega 3}.[Cr]^{\omega 4}.[Us]^{\omega 5}.[R]^{\omega 6}.[Vc]^{\omega 7}} \times \frac{[Sr]^{\omega 8}.(1/[l]^{\omega 9}).[S]^{\omega 10}.[P]^{\omega 11}.[Rs]^{\omega 12}.[Db]^{\omega 13}.[Cu]^{\omega 14}}{\sum_{j=1}^{n}[Sr]^{\omega 8}.[l]^{\omega 9}.[S]^{\omega 10}.[P]^{\omega 11}.[Rs]^{\omega 12}.[Db]^{\omega 13}.[Cu]^{\omega 14}}\right)$$

Where *Bw* is the bandwidth provided by the package, *cc* is the computational capability provided by the package, *Av* is the availability provided by the package, *Cr* is the correctness provided by the package, *Us* is the usability provided by the package, *R* is the reliability provided by the package, *Vc* is the variable computing load provided by the package, *Sr* is the serviceability provided by the package, *l* is the latency provided by the package, *S* is the security provided by the package, *P* is the portability provided by the package, *Rs* is the reliable storage levels provided by the package, *Db* is the data backup provided by the package and *Cu* is the customization provided by the package. Weights corresponding to the properties are represented from ω_1 to ω_{10}.

The ants are distributed in a single requirement node and they traverse through the search space in accordance with the probability exhibited by the packages. The proposed approach uses the 3-Opt variant of the Ant System [24]. This approach uses a transition state *S* which facilitates both exploration and exploitation of the search

$$S = \begin{cases} argmax\{p_{ij}(t)\} & if\ q\ \leq qo\ (exploitation) \\ P & otherwise\ (biased\ exploration) \end{cases}$$

Where q is a random number distributed in [0,1], q_0 is a parameter ($0 \leqslant q_0 \leqslant 1$) that determines if the system should be explored or it should follow the already explored areas (exploitation) and *P* is the resultant value obtained by using the probability function. All the ants are made to traverse in the search space and the package containing the maximum number of ants is selected as the optimal package for the current requirement.

D. Results and discussion

The process of package selection using MCDM based Modified ACO was implemented in C#.NET. Standard packages satisfying various requirements were used as the offered services and the requirements are obtained from the user. Both the packages and the requirements contain values for all the QoS parameters discussed in the previous section. Each QoS parameter is provided with a standard weight depicting its importance in the current requirement. These weights are obtained from the user using pairwise comparisons or direct weight assignments in AHP. This data is utilized in the Modified ACO to identify the fitness values of each of the packages. Search space was created using the predefined packages and ant movement is orchestrated from the requirement node. Modified ACO uses 100 ants to traverse the search space. 100 requirement scenarios were presented to the Modified ACO and the results were recorded.

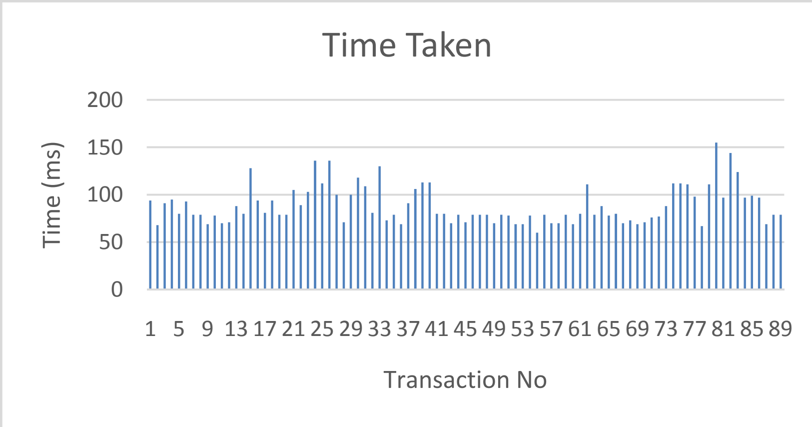

Fig. 2. Time taken

Fig. 2 shows the time taken for identifying the package that best suits the requirement specification presented to the algorithm. A maximum of 160ms and minimum of 60ms were observed from the experiments. Hence it could be concluded that the algorithm operates effectively with an average allocation rate < 0.1sec.

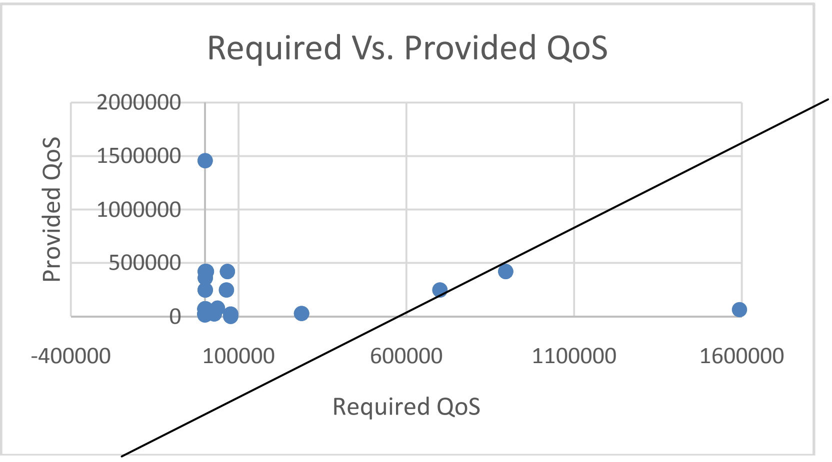

Fig. 3. Required Vs. provided Qos

Fig. 3 presents a plot depicting the required QoS and the provided QoS. Each point in the plot represents a single transaction requirement. The required QoS is plotted in the x-axis and the QoS provided by the algorithm is plotted in the y-axis. The diagonal line represents perfect match, the points above the diagonal represents over assignment while the points under the diagonal represents under assignment. Perfect packages matches cannot be identified, as packages are predefined and user's requirements can vary. Hence points closest to the diagonal line (both top and bottom) represents effective allocations. It could be observed in the plot that most of the points are clustered near the diagonal. A very few diversions could be observed in the plot. This depicts the high accuracy provided by the algorithm.

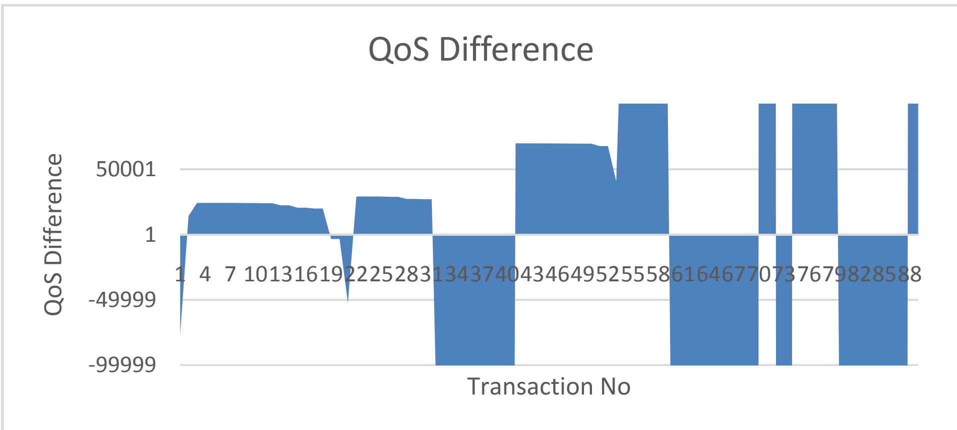

Fig. 4. Qos Difference

Fig. 4 presents the difference in QoS observed in the package allocation process. Transaction numbers are presented in the x-axis and the difference in the QoS values is presented in the y-axis. It could be observed that positive allocations and most appropriate allocations account for more than 70% of the total transactions. This proves the efficiency of the algorithm in terms of effective allocations.

IV. CONCLUSION

In order to ease the process of optimal resource selection, cloud service providers provide packages that can be utilized by the users. However identifying the best packages that suit the needs of the academic organizations is a challenge. This paper presents an effective package identification technique that can be used by the academic organizations to effectively identify the requirements to perform optimal selections. The user requirements are identified from the web logs and as fuzzy requirements from the user. The identified requirements are in the form of QoS parameters. The parameters are ranked using AHP in accordance with the user's requirements. The packages are used to build the search space for 3-Opt modified ACO. Experimental results show that the modified ACO operates effectively on the search space to provide optimal allocations within required time limits.

REFERENCES

[1] S. Singh, I, Chana, "QoS-aware autonomic resource management in cloud computing: a systematic review." ACM Comput Surv 48(3):1–46, 2015.

[2] S. Singh, I. Chana, " Cloud resource provisioning: survey, status and future research directions." Knowl Inform Syst 44:1–50 , 2016.

[3] S. Singh, I. Chana, "A survey on resource scheduling in cloud computing issues and challenges." J Grid Comput 14:1–50, 2016.

[4] Y.S. Lee, A.Y. Zomaya , "Energy efficient utilization of resources in cloud computing systems." J Supercomput 60(2):268–280,2012.

[5] D. Kliazovich, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers." J Supercomput 62(3):1263–1283 ,2012.

[6] W.J. Wang, Y.S. Chang, W.T. Lo, Y.K. Lee "Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments." J Supercomput 66(2):783–811, 2013.

[7] S. Singh, and I. Chana, " Resource provisioning and scheduling in clouds: QoS perspective." The Journal of Supercomputing, pp.1-35, 2016.

[8] Quiroz, H. Kim, M. Parashar, N. Gnanasambandam, N. Sharma, " Towards autonomic workload provisioning for enterprise grids and clouds." In: Grid Computing, 10th IEEE/ACM International Conference on. IEEE, pp 50–57,2009.

[9] C. Vecchiola, R.N. Calheiros, D. Karunamoorthy, R. Buyya. "Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka." Future Gener Comput Syst 28(1):58–65, 2012.

[10] N.R. Herbst, N. Huber, S. Kounev, E. Amrehn, "Self-adaptiveworkload classification and forecasting for proactive resource provisioning." Concurrency Comput Pract Exp 26(12):2053–2078,2014.

[11] S. Kim, J.S. Kim, S. Hwang, and Y. Kim, "Towards effective science cloud provisioning for a large-scale high-throughput computing." Cluster Computing, 17(4), pp.1157-1169, 2014.

[12] L. Wang, J. Zhan, W. Shi, "can scientific communities benefit from the economies of scale? : In cloud, TPDS 99, 1 (2011).

[13] X. Y. Wang, et al. "Appliance-based autonomic provisioning framework for virtualized outsourcing data center." In: Proceedings of the Fourth International Conference on Autonomic Computing, p. 29 (2007).

[14] H, Li, D. Groep, L. Wolters, "Efficient response time predictions by exploiting application and resource state similarities," In Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing. IEEE Computer Society, pp. 234–241 (2005).

[15] B. Urgaonkar, P. Shenoy, and T. Roscoe, "Resource overbooking and application profiling in a shared Internet hosting platform." ACM Trans. Internet Technol. 9, 1, Article 1 (February 2009), pp. 45. 2009.

[16] I, Patiniotakis, Y. Verginadis, and G. Mentzas, "PuLSaR: preference-based cloud service selection for cloud service brokers." Journal of Internet Services and Applications, 6(1), pp.1-14,2015.

[17] H. Ma, and Z.G. Hu, "User preferences-aware recommendation for trustworthy cloud services based on fuzzy clustering." Journal of Central South University, 22, pp.3495-3505,2015.

[18] H. Ma, and Z. Hu, "Recommend trustworthy services using interval numbers of four parameters via cloud model for potential users." Frontiers of Computer Science, 9(6), pp.887-903, 2015.

[19] K. Jayapriya, N.A.B. Mary, and R.S. Rajesh, "Cloud Service Recommendation Based on a Correlated QoS Ranking Prediction." Journal of Network and Systems Management, pp.1-28.

[20] T.L.Saaty, "Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process." RACSAM-Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas, 102(2), pp.251-318, 2008.

[21] T. L. Saaty, and K. Peniwati, "Group decision making: drawing out and reconciling differences." RWS publications, 2013.

[22] A. Colorni, M. Dorigo et V. Maniezzo, "Distributed Optimization by Ant Colonies," actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[23] M. Dorigo, "Optimization, Learning and Natural Algorithms," PhD thesis, Politecnico di Milano, Italy, 1992.

[24] A. Prakasam, and N. Savarimuthu, "Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants." Artificial Intelligence Review, 45(1), pp.97-130, 2016.

AUTHOR PROFILE

Ms.C.Madhumathi., MCA.,M.Phil.,SET., She did her under graducation , post graduation and Master of Philosophy in 2008, 2011 and 2012 respectively from Bharthidasan university,India. She has received MCA degree from School of Computer Science, Engineering and applications from Bharathidasan University, Tiruchirappalli and was a University First Rank holder (Gold Medalist) in her MCA. She has one year teaching experience and her area of research is Cloud Computing.

Gopinath Ganapathy PhD is the Professor and Head of the Dept of Computer Science and Engineering, Bharathidasan University, India. He did his under graduation and post graduation in 1986 and 1988 respectively from Bharathidasan University, India. He obtained his PhD degree, in Computer Science in 1996, from Madurai Kamaraj University, India. Received Young Scientist Fellow Award for the year 1994 and eventually did the research work at IIT Madras. He published around 60 papers. He is a member of IEEE, ACM, CSI, and ISTE. He was a Consultant for 10 years in the international firms in the USA and the UK, including IBM, Lucent Technologies (Bell Labs) and Toyota. His research interests include Modeling, Patterns, NLP, Web Engineering, and Text Mining.