Andreea Buga, Bernhard Freudenthaler, Jorge Martínez Gil, Sorana Tania Nemes, Alejandra Lorena Paoletti (2017)
Management of Accurate Profile Matching using Multi-cloud Service Interaction
In: Proceedings of the 19th International Conference on Information Integration and Web-based Applications and Services, iiWAS 2017, Salzburg, December 4-6, 2017 161-165.

# Management of Accurate Profile Matching using Multi-cloud Service Interaction

Andreaa Buga
Christian Doppler Laboratory for Client-Centric Cloud Computing
Hagenberg, Austria
a.buga@cdcc.faw.jku.at

Bernhard Freudenthaler
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
bernhard.freudenthaler@scch.at

Jorge Martinez-Gil
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
jorge.martinez-gil@scch.at

Sorana Tania Nemeș
Christian Doppler Laboratory for Client-Centric Cloud Computing
Hagenberg, Austria
t.nemes@cdcc.faw.jku.at

Lorena Paoletti
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
lorena.paoletti@scch.at

## ABSTRACT

The current paper describes our research towards a cloud infrastructure for the universal access and interaction with a number of services implementing methods for enriching, matching and querying information about job offers and applicant profiles in the cloud. These methods exploit well-known recruitment knowledge bases in order to deliver valuable information to such organizations as public and private employment agencies that we assume to be geographically distributed. The rationale behind our approach is to offer an universal, yet inexpensive, distribution model able to reduce the cost of installing and maintaining the recruitment technology within the client's businesses.

## CCS CONCEPTS

• **Computer systems organization → Architectures**; • **Information systems →** *World Wide Web*; Information retrieval;

## KEYWORDS

Cloud Computing, Profile Matching, e-Recruitment

**ACM Reference Format:**
Andreaa Buga, Bernhard Freudenthaler, Jorge Martinez-Gil, Sorana Tania Nemeș, and Lorena Paoletti. 2017. Management of Accurate Profile Matching using Multi-cloud Service Interaction. In *Proceedings of The 19th International Conference on Information Integration and Web-based Applications & Services, Salzburg, Austria, December 4–6, 2017 (iiWAS '17),* 6 pages.
https://doi.org/10.1145/3151759.3151831

## 1 INTRODUCTION

In the Human Resources (HR) domain, the accurate matching of job applicants to position descriptions and vice versa is of great importance for both employers looking for filling open positions and job seekers looking for a suitable job [39]. However, there is a number of problems to be addressed in order to build such a solution, some of them related to the scientific challenge consisting of going further beyond traditional matching techniques, and some of them related to the deployment of an infrastructure for the universal access to that technology.

To face the problem of the first kind, we need external sources of background knowledge supporting the domain knowledge to be exploited. However, sophisticated knowledge bases in the HR domain are still rare, as building up a good, large knowledge base is a complex and time-consuming task, though in principle this can be done as proven by experiences in many other application domains. Nevertheless, there already exist several terminological frameworks that have been developed to capture relevant concepts in the recruitment field. The most common frameworks in the area are DISCO [14], ESCO [23] and ISCED [22].

There are also other approaches towards matching algorithms such the one proposed by Stantchev et al. [48], which proposes study recommendation based on the social network profiles of students. Or proposals based on the trofile matching between GitHub accounts and job ads has been introduced by Hauff et al. in [20] in a three steps procedure: concept extraction, weighting and matching. Our approach enhances the previous work by allowing extension of concepts, learning matching measures and performing various matching queries. The contribution of this work is the proposal of a cloud infrastructure for the universal access and interaction with a number of cloud services related to the HR domain. The potential beneficiaries of such an infrastructure are intended to be public and private employment agencies that we assume to be geographically distributed. As shown throughout the paper, the computing capabilities, the scalability and the heterogeneous deployment models of multi-clouds serve as a robust infrastructure for an HR profile matching application. Enhanced with robust monitoring and adaptation, the multi-clouds can improve the reliability of the system.

The rest of this paper is organized as follows, Section 2 details the architecture of the multi-cloud, while Section 3 explains the technical details of our contribution. Advantages and limitations of our approach are discussed in Section 4. Finally, we present the


*iiWAS '17, December 4–6, 2017, Salzburg, Austria*

ACM ISBN 978-1-4503-5299-4/17/12…$15.00
https://doi.org/10.1145/3151759.3151831

Jorge Martinez Gil

concluding remarks and the possible future lines of research in Section 5.

# 2 STRUCTURE OF THE MULTI-CLOUD SERVICES

Distributed computing has favored the development of the cloud computing business model. Most of the existing services rely on single cloud providers. However, this dependency leads sometimes to vendor lock-in and to an inefficient usage of resources as while one provider might be overloaded, others might have idle components. The current focus in the area of cloud computing is on achieving multi-cloud solutions, which coordinate services from different providers and are accessed by the user through simple requests. While such an implementation relies on complex mechanisms, the user must be able to access the services through simple interfaces.

The multi-cloud computing model fits the profile matching application as it supports a separation of the services for different institutions. The interaction of both public and private cloud is possible with the aid of mediator components as the middleware. Therefore, sensitive data for a company can be stored on a private cloud, while the knowledge base can be saved on a public cloud and enriched by different clients, benefiting in this way different institutions. Also, for the job candidate an interface supporting profile editing, update and job search operations can be provided.

## 2.1 Client-Cloud Interaction Middleware

Bósa et al. describe a formal approach for a robust middleware, which encompasses also identity management and content adaptivity services [3]. The model was enriched also with a security module focusing on intrusion detection and with a Service-Level Agreement (SLA) component addressing the needs of clients [27]. The middleware handles the requests of the user and manages the intricacies of the additional above mentioned services. Its formal specification relies on ambient Abstract State Machines (ASM), which permit modeling algorithms and methods using ASM and describing service communication topology, location and mobility using ambient calculus [11].

The proposed middleware can be distributed along a multi-cloud environment and permit three different modes of interaction as described by [7]. We will briefly describe the roles of the three interaction modes for the job profile matching application as depicted by Fig. 1. First supported interaction is between the user and the middleware. The user accesses the cloud services through a Software as a Service (SaaS) application and requests to carry out an operation. For instance, a job candidate updates its profile which can lead to a modification of the knowledge base stored in the cloud. The middleware takes the request with its corresponding data and submits it to the responsible service. In case an operation requires the coordination of several services, different middleware instances communicate between each other, this representing the second type of interaction. For instance, a representative of a company might request a list of candidates matching a set of criteria. Its request is recorded by a middleware component and split in different services as follows. Authentication and authorization is verified in the cloud with a more robust security where the accounts are saved. The list of candidates is retrieved by applying similarity measures by the query engine belonging to a cloud with higher processing capabilities. If the authentication and authorization middleware validates the credentials, the querying middleware can further submit its response to the user. The communication between the authorization middleware and the authorization method is regarded as the third type of interaction, namely between a middleware and a service belonging to a cloud it is responsible for.

Multi-clouds are characterized by a high complexity and heterogeneity, usually reflected in faults as random behavior or operation inconsistencies. For the case of the job profile matching, these faults can cause inaccurate results that would affect the selection process. Another undesired effect might be the incorrect alteration of the data from the knowledge base. In order to avoid such situations, we address the aspect of robust monitoring and adaptation of services in case of abnormal execution.

In order to provide guarantees for the functioning of such profiling systems as a whole, the middleware complements the execution layer with monitoring and adaptation processes. The monitoring and adaptation layers are cooperating for the delivery of expected Quality of Service (QoS), each fulfilling clearly defined tasks, but closely collaborating.

## 2.2 Monitoring

*2.2.1 Background.* Monitoring refers to the process of collecting measurements about a system or a service in order to assess their behavior and verify the expected properties. Active and passive monitoring techniques can be applied for these purposes. While active techniques usually generate artificial (controlled) load on a real system with the only objective of monitoring, passive techniques collect measurements on a system while it is operating, that is, under its actual load [9, 10]. Passive monitoring techniques include network sniffing, code profiling and application service logging. Measurements are usually stored into trace logs, that is, collections of time stamped recordings of various types of information, which can be processed to either on the fly or postmortem.

Monitoring involves examining the execution traces of a system in order to verify expected properties. Monitoring is based on passive testing, i.e. the observation of the system traces without interfering with the system's normal operation [1, 2]. Monitoring is also closely related to run-time verification [28]. Many established techniques are available addressing either the monitoring of network communication between services and systems (black-box monitoring) or the monitoring of application execution (white-box monitoring). State-of-the-art techniques for network monitoring exploit SNMP [47, 52], Deep Packet Inspection (DPI) [13], and invariants [35]. DPI is a technique that is used for completely analyzing communication packets (both headers and payloads) and has been employed for security analysis of network traffic and for detecting and preventing intrusions (Intrusion Detection and Prevention Systems (IDPS) [46]). Most techniques depend on pattern matching, e.g. [50], but a few use correlation of events, e.g. [49]. In white-box monitoring an application is analyzed during its execution. State-of-the-art techniques exploit just-in-time compilation (e.g. [38, 44, 51]) or debugging tools.

*2.2.2 Framework.* Monitors run continuously in the background of the service execution and check if components face abnormal

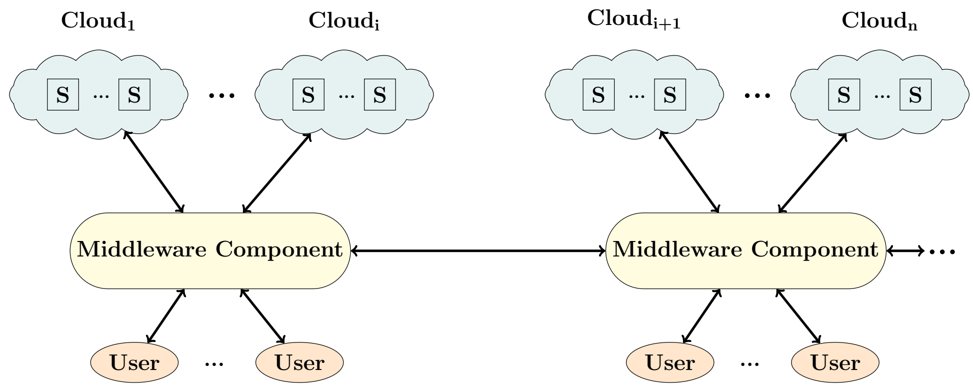


**Figure 1: General structure and interaction of the middleware**

situations. The monitoring layer has been firstly sketched in [4]. Each node of the multi-cloud system is assigned a set of monitors, whose redundancy aims to remove the problem of a single-point of failure. Adaptation actions are triggered whenever an issue is reported by the monitors and are costly for the providers. It is, therefore, essential to have an accurate evaluation of the system. For this reason, we previously proposed a measure of the confidence of a monitor that reflects the correctness of its assessment of the node [8]. In order to avoid a high communication overhead, we assigned a leader for each set of monitors, which is in charge of executing a collaborative diagnosis whenever a problem is reported by a monitor. This implies that each monitor submits its last evaluation to the leader, which in the end chooses the most voted diagnosis.

The correctness of the monitoring processes is fundamental for the evaluation of the system and its reliability. We, thus, integrated the ASM method and build a model of the monitors, which we validated [6] and verified against a set of desired properties [5]. The resulted model can be further integrated with the ambient ASM specification of the middleware.

## 2.3 Adaptation

*2.3.1 Background.* Adaptivity in general refers to systems that can change themselves at run-time via learning, evolution, development, or more subtle kinds of interaction [29]. These systems might be evolving populations, developing or interacting individuals, colonies or swarms with division of labor, assistive robots, interactive interfaces, or less circumscribed agent-environment systems, living or artificial systems, etc. Dynamic adaptation is becoming a key element in software engineering for a growing range of domains, such as automotive systems, web services, networks, pervasive systems, etc. (see e.g. [12, 17, 21, 45]). A general architectural framework has been proposed within IBM's *Autonomic Computing* initiative, based on a vision of creating computer systems with so-called *self-* properties* (self-healing, self-stabilizing, self-organizing, etc.) [21]. Adaptation decisions may involve the evaluation of new alternatives by exploring the adaptation space. Some preliminary work for tackling these design/adaptation decisions already exist in the context of software architectures and service-oriented applications (see e.g. [24, 33]). One promising approach, in particular, is the identification and application of architectural *design patterns* and *tactics* [15, 18, 19, 25, 26, 32, 34, 43]. Architectural patterns are chosen in response to early design decisions, and provide the major structures in which multiple design decisions are realized. A *tactic* is a design decision whose goal is the improvement of one specific design concern regarding a quality attribute.

*2.3.2 Framework.* The adaptation framework reacts to and evaluates the data collected and assessed by the monitors and deals with recovering from anomalous situations, logging them, and finding the best remedy to restore the system to normal running mode, under presumably optimal performance. On the level of adaptation, a general approach is developed that interrupts parts of the running system, rolls back to a consistent state, execute adaptation algorithms that mitigate or repair critical situations and restart after adaptation. For this we exploit a recently developed behavioral theory of reflective algorithms, which has been developed in connection with reflective ASMs [16].

The flaws and the solutions chosen for their resolution are stored in a case base repository which is continuously accessed and improved by the adaptation component [36]. Each case is defined as a collection of description features subject to a common pattern recognition mechanism (the problem) and a finite set of repair actions also known as the adaptation schema (the solution). Any repair/adaptation action is an activity that uses a set of inputs to produce a set of outputs relevant to solve the problem the adaptation solution was designed for. Repair actions can be the replacement of a component service by an equivalent one exploiting dynamic service deployment, the change of location for a service, or the replacement of larger parts of the multi-clouds system, i.e. a set of services involved, by a completely different, alternative solution. The action deployment component is used to drive and monitor the configured implementation of any adaptation solution, once it was deemed most similar to the registered problem. The core of this component lies in the action deployment registry and review capabilities that compile the action workflow schema and load all the relevant data to handle the execution of the adaptation [37].

Once the solution is carried out according to its specification, the monitors are requested to qualitatively characterize the status of the system post reconfiguration, in other words apply the workflow analysis and performance, accuracy and output evaluation to specific threshold values. In correlation with previous recording of the monitors, the system can detect if the adaptation plan was efficient or not. The analysis is further sent to the adapter, which will either mark the solution as successful and index and retain it in the case repository for future problem reference, or unsuccessful which will lead to further revision and optimization of the aggregating features, making it a better fit for the given problem.

## 3 SERVICES IN A PROFILE MATCHING SYSTEM

In previous works, we presented an approach to eliminate the need for job recruiters to have deep and specialized knowledge within a professional domain [31]. In this work, we propose a way to model domain knowledge from a lot of different professional sectors supported by an universally accessible infrastructure. In addition, this knowledge could be used as a support when performing matching process so that the results can be very similar to those produced by an expert from the field of interest. Our previous approach was designed to work as a standalone system that could be deployed in a number of organizations with great recruitment needs. However, each of these organizations was responsible to set up, enrich, train and maintain the system leading to operations, which imply certain expenses. For this reason, we propose to work towards a cloud solution that can facilitate universal access to these services.

In this context, one of the scientific goals that our research tries to pursue is to develop new methods and tools for achieving a much more realistic matching between job offers and applicant profiles. This realistic matching goes further beyond traditional matching techniques that merely rely on the syntactic overlapping of offers and profiles. Semantic matching aims to limit this short-sighted strategy by exploiting knowledge bases that are a valuable source of background knowledge concerning the different recruiting domains.

It is envisioned to make use of a number of taxonomies available in the HR area for education, skills, competences such as DISCO, ESCO, ISCED, as they provide the set of concepts necessary to construct the initial knowledge base for this approach. Additionally, with social skills, preferences, interests, etc., they comprise the complete body of Competences required for that purpose. In this regard, the representation of knowledge can be covered by Description Logic.

Profile matching reflects how well a given profile fits a required profile. In order to achieve profile matching in knowledge bases, the main point is how to represent profiles in knowledge bases and, to do so, the approach is to represent skills by sets, the sets of skills a person may posses, for instance: "knowledge of Bionformatics", "good English speaker", "proactive profile".

In order to match the similarities between the skills of two profiles, the idea is to measure the similarity of the skill sets. Thus, it seems reasonable to exploit partially-ordered sets and lattices to capture the hierarchical dependencies of concepts and, represent profiles by filters in lattices. Such that, a required profile $R$ and a given profile $G$ can be respectively represented by filters $\mathcal{F}_r$ and $\mathcal{F}_g$ in the lattice, and their *matching value* $\mu(\mathcal{F}_g, \mathcal{F}_r)$ can be calculated by their hierarchical dependencies, as introduced in [41]. With this approach, Popov and Jebelean introduced the first attempt to use filters in lattices. In fact, our approach is based on this work, although we have further investigated the filter-based matching.

In this way, the initial approach was introduced in [39] where weighting on concepts for matching measure was introduced as well as the *over-qualification measures* where the inverted measures are taken into account, and also the blow-up operators, as described earlier in this work.

*Learning of the matching measures* is another feature included in Fig. 2, initially introduced in [30] and further investigated in [31, 42]. The goal is to determine the matching measures between profiles that comply with the matching measures given by a human expert. The idea is to start from a set of filters and matching values determined by a human expert and from there derive plausibility constraints that should be satisfied to exclude unjustified bias, based on facts not present in the knowledge base.

In this way, the problem of how human-made matching could be exploited to learn a suitable weighting function is thought to be also included as part of the approach presented in this work. This was recently introduced in [30] where we come to the conclusion that under some assumptions, the ranking preserving matching measure exists.

In order to make the described architecture in Fig. 2 available to end users we have investigated an efficient methodology to implement queries to the profile database. This implies the investigation of *matching queries* in particular, *top-$k$ queries* [40] and *gap-queries*.

As for top-$k$ queries the approach is to allow the users to query profiles and define themselves the weights for concepts relevance within the ontology structure. This presumes a pre-computation of matching measures, which is an efficient formulation provided that updates to the TBox are assumed to be infrequent. The use of relational databases are also taken into account for the storage of profiles, their selected weights and the pre-computed matching values. The rings and spiders storage architecture designed for the top-$k$ queries facilitates an accurate and efficient retrieval of the top-$k$ matching profiles.

As for the gap queries, the basic idea is to benefit the job seekers by providing them with suggestions on further training/education in order to increase their chances to be selected. In today's competitive job market it seems reasonable to provide advise on profiles improvement in relation to a particular job offer in contrast with the rest of the job seekers. The approach is to compute an extension of a given profile that will appear in the result of a top-$k$ matching query for at least $l$ requested profiles. And then, compute the skill difference between profiles, the original given profile and the extended result set of profiles.

## 4 RESULTS

The area of profile matching is essential for the improving the recruitment procedure and ensuring that candidates keep up with the requirements of the market. Our approach distinguishes itself from

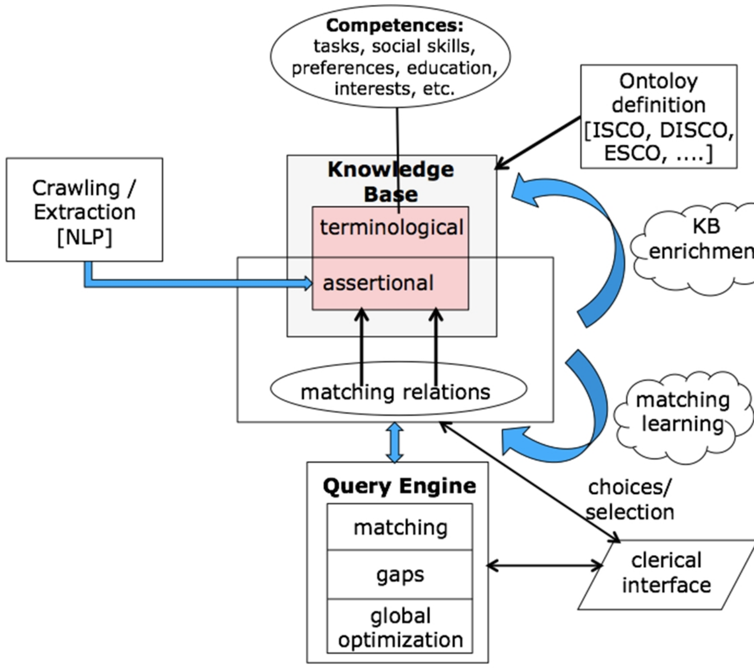


**Figure 2: General overview of the envisioned enveloping system**

the existent solutions by the following advantages. Firstly, our cloud platform offers the capability to access the recruitment services that are delivered on demand over the Web, without the need to store them. This means that it is possible to obtain recruitment information from any device and at any time from every place around the world. Secondly, it is possible to work with multiples information sources of candidates and potential employers. Our cloud approach for the HR solution means combining many promising sources of employment into a single source for tracking, measuring and reporting available for candidates. Thirdly, our cloud approach is designed to augment, rather than replace, existing HR solutions. This means that implementing our cloud solution does not disrupt business as usual, and partly eliminates the cost commonly associated with the replacement of old systems. Lastly, our cloud platform could offer an API to automate the communication between different systems and platforms. This would allow companies to customize their own solutions in order to meet their specific needs.

## 5 CONCLUSION

The current paper has presented our research work towards a cloud infrastructure for the universal access and interaction with a number of novel e-Recruitment services. The goal of this approach is to automatize the matching process and enhance the performance of the matching algorithm through learning approaches. The proposed deployment model of the solution is a multi-cloud infrastructure that can ensure different access models to the application and also scalability. Multi-clouds also fulfill the requirements of a profile matching system that needs to compose information from different sources and expert systems, which can be stored at different providers. By thoroughly monitoring the execution and reacting to issues with appropriate adaptation measures, the system becomes more reliable.

Furthermore, such a system can be considered an important tool for educating and training institutions, which could perform gap analysis to determine the most needed qualification offers. This measure would improve the skills of job seekers with respect to the available positions and would balance the job market. We propose as a future work a formal analysis of the profile matching algorithm, which can be further included in the formal model. Additionally, we aim to enrich the specification with the design of a prototype of the system.

## ACKNOWLEDGMENTS

The research reported in this paper has been supported by the Christian-Doppler Society in the frame of the Christian-Doppler Laboratory for Client-Centric Cloud Computing, and further by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

## REFERENCES

[1] C. Andrés, M.-E. Cambronero, and M. Nuñez. Formal passive testing of service-oriented systems. In *IEEE SCC 2010*, pages 610–613, 2010.
[2] E. Bayse, A. Cavalli, M. Nuñez, and F. Zaidi. A passive testing approach based on invariants: Application to the WAP. *Computer Networks*, 48:247–266, 2005.
[3] K. Bósa, R.-M. Holom, and M. B. Vleju. *Correct Software in Web Applications and Web Services*, chapter A Formal Model of Client-Cloud Interaction, pages 83–144. Springer International Publishing, Cham, 2015.
[4] A. Buga. A scalable monitoring solution for large-scale distributed systems. In R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, editors, *Computer Aided*

*Systems Theory Ű EUROCAST 2015*, volume 9520 of *Lecture Notes in Computer Science*, pages 219–227. Springer International Publishing, 2015.

[5] A. Buga and S. T. Nemeş. Formalizing monitoring processes for large-scale distributed systems using abstract state machines. In P. Arcaini, M. Mongiello, E. Riccobene, and P. Scandurra, editors, *Formal Approaches for Advanced Computing Systems – 1st International Workshop (FAACS 2017)*, LNCS. Springer, 2017. (to appear).

[6] A. Buga and S. T. Nemeş. Towards modeling monitoring services for large-scale distributed systems with abstract state machines. In *Proceedings of the 22nd International Working Conference on Evaluation and Modeling Methods for Systems Analysis and Development (EMMSAD)*, volume 1859 of *CEUR Workshop Proceedings*, pages 103–112. CEUR-WS.org 2017, June 2017.

[7] A. Buga, S. T. Nemeş, and K.-D. Schewe. Conceptual modelling of autonomous multi-cloud interaction with reflective semantics. In G. Guizzardi, H. Ma, and H. C. Mayr, editors, *Conceptual Modeling – 36th International Conference (ER 2017)*, LNCS. Springer, 2017. (to appear).

[8] A. Buga and S. T. Nemes. Towards modeling monitoring of smart traffic services in a large-scale distributed system. In *CLOSER 2017 - Proceedings of the 7th International Conference on Cloud Computing and Services Science, Porto, Portugal, April 24-26, 2017.*, pages 455–462, 2017.

[9] M. Calzarossa, M. Della Vedova, L. Massari, D. Petcu, M. Tabash, and D. Tessera. Workloads in the clouds. In L. Fiondella and A. Puliafito, editors, *Principles of Performance and Reliability Modeling and Evaluation*, Springer Series in Reliability Engineering. Springer, 2016.

[10] M. Calzarossa, L. Massari, and D. Tessera. Workload characterization: A survey revisited. *ACM Compututing Surveys*, 48(3):48:1–48:43, 2016.

[11] L. Cardelli and A. D. Gordon. Formal methods for distributed processing. chapter Mobile Ambients, pages 198–229. Cambridge University Press, New York, NY, USA, 2001.

[12] B. Cheng, R. de Lemor, H. Giese, P. Inverardi, and J. Magee, editors. *Software Engineering for Self-Adaptive Systems*, volume 5525 of *LNCS*. Springer, 2009.

[13] E. H. Corwin. Deep packet inspection: Shaping the internet and the implications on privacy and security. *Information Security Journal: A Global Perspective*, 20(6):311–316, 2011.

[14] European distionary of skills and competences. http://www.disco-tools.eu.

[15] N. Esfahani and S. Malek. On the role of architectural styles in improving the adaptation support of middleware platforms. In *Proceedings of the 4th European conference on Software architecture (ECSA'10)*, pages 433–440. Springer, 2010.

[16] F. Ferrarotti, L. Tec, and J. Turull Torres. Towards an asm thesis for reflective sequential algorithms. In M. Butler et al., editors, *ABZ 2016, Proceedings*. 2016. to appear.

[17] H. Giese and B. H. C. Cheng, editors. *2011 ICSE Symposium on Software Engineering for Adative and Self-Managing Systems (SEAMS 2011), Waikiki, Honolulu, HI, USA, May 23-24, 2011*. ACM, 2011.

[18] N. Harrison and P. Avgeriou. How do architecture patterns and tactics interact? a model and annotation. *Journal of Systems and Software*, 83(10):1735–1758, 2010.

[19] N. Harrison and P. Avgeriou. Implementing reliability: The interaction of requirements, tactics and architecture patterns. In *Architecting Dependable Systems VII*, volume 6420 of *LNCS*, pages 97–122. Springer, 2010.

[20] C. Hauff and G. Gousios. Matching github developer profiles to job advertisements. In *12th IEEE/ACM Working Conference on Mining Software Repositories, MSR 2015, Florence, Italy, May 16-17, 2015*, pages 362–366, 2015.

[21] M. Huebscher and J. McCann. A survey of autonomic computing - degrees, models, and applications. *ACM Comp. Surv.*, 40(3), 2008. Article No. 7.

[22] International standard classification of education. http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx.

[23] International standard classification of occupations. http://www.ilo.org/public/english/bureau/stat/isco/isco08/, 2008.

[24] R. Kazman, L. Bass, G. Abowd, and M. Webb. A method for analyzing the properties of software architectures. In *Proceedings of the 16th International Conference on Software Engineering (ICSE'94)*, pages 81–90, 1994.

[25] S. Kim, D. Kim, L. Lu, and S. Park. Quality-driven architecture development using architectural tactics. *Journal of Systems and Software*, 82(2):1211–1231, 2009.

[26] A. Koziolek, H. Koziolek, and R. Reussner. Peropteryx: Automated application of tactics in multi-objective software architecture oprimization. In *Proceedings of the joint ACM SIGSOFT conference – QoSA and ACM SIGSOFT symposium – ISARCS on Quality of software architectures – QoSA and architecting critical systems – ISARCS (QoSA-ISARCS'11)*, pages 33–42, New York, 2011. ACM.

[27] H. Lampesberger and M. Rady. *Correct Software in Web Applications and Web Services*, chapter Monitoring of Client-Cloud Interaction, pages 177–228. Springer International Publishing, Cham, 2015.

[28] M. Leucker and C. Schallhart. A brief account of runtime verification. *Journal of Logic and Algebraic Programming*, 78(5):293–303, 2009.

[29] M. Macy and R. Willer. From factors to actors: Computational sociology and agent based modelling. *Annual Review of Sociology*, 28:143–166, 2002.

[30] J. Martinez-Gil, A. L. Paoletti, G. Rácz, A. Sali, and K. Schewe. Maintenance of profile matchings in knowledge bases. In *Model and Data Engineering - 6th International Conference, MEDI 2016, Almería, Spain, September 21-23, 2016, Proceedings*, pages 132–141, 2016.

[31] J. Martinez-Gil, A. L. Paoletti, and K. Schewe. A smart approach for matching, learning and querying information from the human resources domain. In *New Trends in Databases and Information Systems - ADBIS 2016 Short Papers and Workshops, BigDap, DCSA, DC, Prague, Czech Republic, August 28-31, 2016, Proceedings*, pages 157–167, 2016.

[32] D. Menasce, J. Sousa, S. Malek, and H. Gomaa. QoS architectural patterns for self-architecting software systems. In *Proceedings of the 7th International Conference on Autonomic Computing (ICAC'10)*, pages 195–204. ACM, 2010.

[33] R. Mirandola and P. Potena. Self-adaptation of service based systems based on cost/quality attributes tradeoffs. In *Proceedings of the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNACS 2010)*, pages 493–501. IEEE, 2010.

[34] R. Mirandola, P. Potena, and P. Scandurra. An optimization process for adaptation space exploration of service-oriented applications. In *Proceedings of the IEEE 2011 6th International Symposium on Service Oriented System Engineering (SOSE), 12-14 Dec. 2011*, pages 146–151. IEEE, 2011.

[35] G. Morales, S. Maag, A. R. Cavalli, W. Mallouli, E. Montes de Oca, and B. Wehbi. Timed extended invariants for the passive testing of web services. In *ICWS 2010*, pages 592–599, 2010.

[36] S. T. Nemes and A. Buga. Towards a case-based reasoning approach to dynamic adaptation for large-scale distributed systems. In D. W. Aha and J. Lieber, editors, *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*, LNCS, pages 257–271. Springer, 2017.

[37] S. T. Nemes and A. Buga. Towards modeling adaptation services for large-scale distributed systems with abstract state machines. In B. Shishkov, editor, *Business Modeling and Software Design - 7th International Simposium, BMSD 2017, Barcelona, Spain, July 3-5, 2017, Proceedings*, pages 193–198. Springer, 2017.

[38] A. Nusayr and J. Cook. Extending AOP to support broad runtime monitoring needs. *Software Engineering and Knowledge Engineering*, pages 438–441, 2009.

[39] A. L. Paoletti, J. Martinez-Gil, and K.-D. Schewe. Extending knowledge-based profile matching in the human resources domain. In Q. Chen et al., editors, *Database and Expert Systems Applications (DEXA 2015) Part II*, volume 9262 of *LNCS*, pages 21–35. Springer, 2015.

[40] A. L. Paoletti, J. Martinez-Gil, and K.-D. Schewe. Top-k matching queries for filter-based profile matching in knowledge bases. In H. Ma and S. Hartmann, editors, *Database and Expert Systems Applications (DEXA 2016)*, LNCS. Springer, 2016. to appear.

[41] N. Popov and T. Jebelean. Semantic matching for job search engines – a logical approach. Technical Report 13-02, Research Institute for Symbolic Computation, JKU Linz, 2013.

[42] G. Rácz, A. Sali, and K.-D. Schewe. Semantic matching strategies for job recruitment: A comparison of new and known approaches. In M. Gyssens and G. R. Simari, editors, *Foundations of Information and Knowledge Systems (FoIKS 2016)*, volume 9616 of *LNCS*, pages 149–168. Springer, 2016.

[43] A. J. Ramirez and B. H. Cheng. Applying adaptation design patterns. In *Proceedings of the 6th international conference on Autonomic computing (ICAC'09)*, pages 69–70, New York, 2009. ACM.

[44] M. Richters and M. Gogolla. Aspectoriented monitoring of UML and OCL constraints. In *AOSD Modeling – UML Workshop, 6th International Conference on the Unified Modeling Language (UML)*, 2003.

[45] M. Salchie and L. Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems*, 4(2), 2009. Article No.14.

[46] K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (IDPS). http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf, 2007. National Institute of Standards and Technology (NIST); accessed 2013-01-04.

[47] K.-S. Shin, J.-H. Jung, J. Y. Cheon, and S.-B. Choi. Real-time network monitoring scheme based on SNMP for dynamic information. *J. Network and Computer Applications*, 30(1):331–353, 2007.

[48] V. Stantchev, L. Prieto-GonzÃąlez, and G. Tamm. Cloud computing service for knowledge assessment and studies recommendation in crowdsourcing and collaborative learning environments based on social network analysis. *Computers in Human Behavior*, 51:762 – 770, 2015. Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.

[49] http://bro-ids.org.

[50] http://www.snort.org.

[51] http://valgrind.org.

[52] W. Zeng and Y. Wang. Design and implementation of server monitoring system based on SNMP. *JCAI*, pages 680–682, 2009.