Technology Management, Economics, and Policy Discussion Paper Series

SEOUL NATIONAL UNIVERSITY
College of Engineering
Technology Management, Economics, and Policy Program

Achieving Market Liquidity through Autonomic Cloud Market Management

Ivan Breskovic, Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic, Jörn Altmann

TEMEP Discussion Paper No. 2011:80

서 울 대 학 교
공 과 대 학, 기술경영경제정책 대학원과정
151-742 서울시 관악구 관악로 599

Technology Management, Economics, and Policy Program
College of Engineering, Seoul National University
599 Gwanak-Ro, Gwanak-Gu, Seoul 151-742, South-Korea
Phone: ++82-2-880-9140, Fax: ++82-2-880-8389

The *Technology Management, Economics, and Policy Program (TEMEP)* is a graduate program at Seoul National University. It includes both M.Sc. and Ph.D. programs, which are integrated into the graduate programs at the College of Engineering.

The *TEMEP Discussion Paper Series* is intended to serve as an outlet for publishing research about theoretical, methodological, and applied aspects of industrial economics, especially those related to the institute's areas of specialization, namely management of technology, information and communication technology, telecommunication, services, health industries, energy industries, as well as infrastructures for development. In particular, paper submissions are welcome, which analyze current technology and industry related issues, discuss their implications, and suggest possible alternative policies. The objective is to gain insights into important policy issues and to acquire a balanced viewpoint of policy making, technology management, and economics. It will enable us to identify the problems in industries accurately and to come up with optimal and effective guidelines. In addition, another important aim of this series is to facilitate communication with external research institutes, individual researchers, and policy makers worldwide.

Research results disseminated by TEMEP may include views on policies for information and communication technologies, technology management, telecommunication, energy, health, and development economics, but the institute itself takes no institutional policy position. If appropriate, the policy views of the institute are disseminated in separate policy briefs. Thus, any opinion expressed in the TEMEP Discussion Paper Series is those of the author(s) and not necessarily the opinion of the institute.

Finally, the current editor of this series would like to thank Prof. Dr. Almas Heshmati for establishing this working paper series in January 2009 and for guiding its development during the course of the first 55 issues. In particular, Prof. Heshmati provided invaluable contributions for setting up the local IT infrastructure, registering the series with RePEc, disseminating the working papers, and setting the quality requirements.

Jörn Altmann, Editor

Office: 37-305
Technology Management, Economics, and Policy Program
College of Engineering
Seoul National University
599 Gwanak-Ro, Gwanak-Gu
Seoul 151-742
South-Korea

Phone: +82-70-7678-6676
Fax: +1-501-641-5384
E-mail: jorn.altmann@acm.org

Achieving Market Liquidity through Autonomic Cloud Market Management

Ivan Breskovic[1], Michael Maurer[1], Vincent C. Emeakaroha[1], Ivona Brandic[1], Jörn Altmann[2]

[1] Distributed Systems Group
Vienna University of Technology
Vienna, Austria
{breskovic, maurer, vincent, ivona}@infosys.tuwien.ac.at

[2] Department of Industrial Engineering
Technology Management, Economics, and Policy Program
College of Engineering
Seoul National University
Seoul, South-Korea
jorn.altmann@acm.org,

August 2011

Abstract: Due to the large variety in computing resources, Cloud markets often suffer from a low probability of finding matches between consumers' bids and providers' asks, resulting in low market liquidity. The approach of SLA templates (i.e., templates for electronic contracts) is a means to reduce this variety as it channels the demand and supply. However, until now, the SLA templates used were static, not able to reflect changes in users' requirements. To address this shortcoming, we introduce an adaptive approach for automatically deriving public SLA templates based on the requirements of market participants. To achieve this goal, we utilize clustering algorithms for grouping similar requirements and learning methods for adapting the public SLA templates to observed changes of market conditions. To assess the benefits of the approach, we conduct a simulation-based evaluation and formalize a utility and cost model. Our results show that the use of clustering algorithms and learning algorithms improves the performance of the adaptive SLA template approach.

Keywords: Service level agreement, SLA management, electronic markets, autonomic computing, marketplace, market environments.

JEL Classification Numbers: C61, C63, L14, L15, L86, M15.

1. Introduction

Cloud computing is the result of the convergence of several concepts, ranging from virtualization, distributed application design, Grid computing, and enterprise IT management. It enables a flexible approach for deploying and using applications [3]. In such a paradigm, hardware resources, software resources, and data are offered as on-demand services to the user in a pay-as-you go model. The functional and non-functional requirements of those services, termed Quality of Service (QoS) requirements, are expressed and negotiated by means of Service Level Agreements (SLAs). Those are contracts in Cloud resource markets and are signed between a customer and a service provider.

For the successful implementation of the Cloud computing paradigm, well-developed hardware and software resource markets are fundamental [15]. One essential feature of a well-developed market is liquidity of its goods. Illiquid goods have the risk of not being sellable or purchasable when needed, driving potential market participants away. Therefore, well-developed resource markets must enable immediate execution of standard orders, which is possible through a sufficiently large number of participants trading [16].

Many electronic marketplaces suffer from challenging situations, such as dynamic evolution of user requirements for goods. Due to the large resource variability in Cloud markets and still low number of market participants, a small difference in a single user requirement might prevent a match. This can result in a very low percentage of matches of consumers' requirements and providers' offers, making the market unattractive to both consumers and providers.

To counteract this problem, SLA templates have been introduced. They are documents comprising all required elements of an SLA but without QoS values assigned [1]. By creating publicly available SLA templates that stored in searchable registries, finding an appropriate trade partner is simplified. However, since the generation of SLA templates has been static until now (i.e., any market changes that require the adaptation of resource attributes within the SLA template cannot be considered), the solution of SLA templates becomes unattractive.

To confront this challenge, we investigate autonomic creation and management of SLA templates with the goal of attracting participants to trade and, therefore, increasing the liquidity of goods. In detail, the approach presented in this paper is based on SLA mapping. SLA mappings are XSLT documents, which bridge the differences between two SLA templates by defining mappings between the non-matching parts of the two SLAs. Using this approach, a service consumer can define mappings between his private SLA template and the publicly available SLA template.

In this paper, in particular, we derive new public SLA templates by adapting public SLA templates to changes in demand in the market through the use of clustering methods and economic mechanisms. This approach leads to the automation of resource markets, as it allows automatically adapting to changes of market conditions. With the help of a simulation environment, we demonstrate the benefits of our approach. In particular, we show that this SLA management approach brings three additional benefits for market participants: (1) It allows market participants to keep their existing private SLA templates that may already be used in other business processes; (2) It enables participants to impact the evolution of Cloud resource markets; (3) It delivers better results for the Cloud market as a whole. We show this benefit by formalizing and applying a measure for assessing publicly available SLA templates within our simulation.

Concluding, the main contributions of this paper are: (1) introduction of an approach for applying clustering algorithms to group similar SLA mappings, enabling autonomic market management; (2) formalization of a measure for evaluating the approach by determining the utility and the cost to users; and (3) evaluation of the proposed mapping approach using an experimental testbed.

The remainder of the paper is organized as follows. Section 2 describes related work. Section 3 defines the SLA mapping approach and introduces the idea of applying clustering algorithms to group similar SLA mappings. Section 4 introduces in detail clustering algorithms and adaptation methods used for generating new public SLA templates. The formalization of the utility and cost model to assess the benefits of the approach is given in Section 5. A description of the simulation environment and the evaluation results are presented and discussed in Section 6. Section 7 concludes the paper.

2. Related Work

Over the past several years, many large IT companies have entered the utility computing market by offering different types of services. We divide them in three groups, based on the types of resources offered: (1) group of service providers, who offer their computing resources on a pay-per-use basis in the infrastructure-as-a-service model, such as Amazon EC2[1] and S3[2], (2) providers, who offer their computing resources in combination with their own software components, such as Google Apps[3] and Salesforce.com[4], and (3) group of providers offering not only a deployment platform but an application development platform as well, such as Microsoft Azure[5] and Salesforce.com. However, although these providers offer a large variety of services, they usually sell a single type of resources and do not cover flexible SLAs.

In addition to this, several research projects investigate SLA template generation [14], [4], [6]. As reported in [8], most projects use SLA specifications, based on WSLA and WS-Agreement, which lack support for economic attributes and negotiation protocols. To compensate these shortcomings, [14] introduce utilization of semantic Web technologies based on WSDL-S and OWL for enhancement of WS-Agreement specifications to achieve autonomic SLA matching. Similar to that, [6] introduces an ontology-based SLA formalization where OWL and SWRL are chosen to express the ontologies. [4] suggest

---

[1] http://aws.amazon.com/ec2/
[2] http://aws.amazon.com/s3/
[3] http://www.google.com/apps/
[4] http://www.salesforce.com/
[5] http://www.microsoft.com/windowsazure/

producing a unified QoS ontology applicable to the main scenarios such as QoS-based Web Services selection, QoS monitoring, and QoS adaptation.

Several research projects have also discussed the implementation of system resource markets [2], [13], [12]. GRACE [2] developed a market architecture for Grid markets and outlined a market mechanism, while the good itself (i.e., computing resource) has not been defined. Moreover, the process of creating agreements between consumers and providers has not been addressed. The SORMA project [13] has also considered open Grid markets. They identified several market requirements, such as allocative efficiency, budget-balance, truthfulness, and individual rationality [12]. However, they have not considered that a market can only function efficiently, if there is sufficiently large liquidity of its goods.

Overall, most of the computing resource market research did not define the tradable good, or they worked only with simplified definitions of goods. Furthermore, most of the existing approaches in SLA management use static SLA templates and, therefore, cannot consider market changes. This means that the issue of maintaining sufficiently high liquidity in an electronic market has not been addressed yet.

3. Towards Autonomic Cloud Markets

In this section, we present the SLA mapping approach and introduce the idea of applying clustering algorithms to group SLA mappings.

3.1. The SLA Mapping Approach

The SLA mapping process is presented in Figure 1. In step 1, a service provider assigns his service (e.g., infrastructure resources) to a particular public SLA template. Since it is sometimes not possible to find a perfect match between a private SLA template and a public SLA template, service provider can define mappings to bridge the differences between the two templates (step 2).

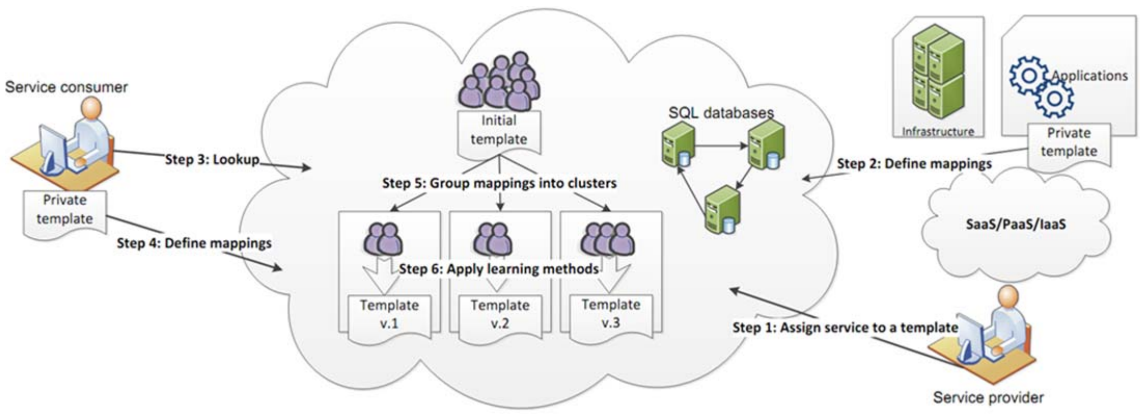


Figure 1. The SLA mapping process.

In the next step, a service consumer looks for an appropriate Cloud service. Since it might happen that a consumer cannot change his private template due to existing business processes, legal issues, or other enterprise-internal reasons, he can define SLA mappings to bridge the differences between his private template and the public template (step 4).

Both providers and consumers can submit two types of SLA mappings:

1. **Ad-hoc SLA mapping type** defines translation between a parameter existing in the user's private SLA template and the public SLA template. We distinguish simple ad-hoc mappings, i.e., mapping of different values for an SLA element (e.g., a mapping between the names "CPUCores" and "NumberOfCores" of an SLA parameter, or a mapping between two different values of a service level objective), and complex ad-hoc mappings. The later one maps between different functions used for calculating a value of an SLA parameter. For example, complex ad-hoc mappings can define a mapping for a function, which is used for calculating the SLA parameter "Price" in the metric unit "EUR", to a function, which is used for calculating the SLA parameter "Price" in "USD".
2. **Future SLA mapping type** defines a wish, which is supported by the private SLA template, for adding a new SLA parameter to the public template or deleting an SLA parameter from the public SLA template. Unlike ad-hoc mapping, future mappings cannot be applied immediately but possibly in the future if a sufficiently large number of private templates supports the action.

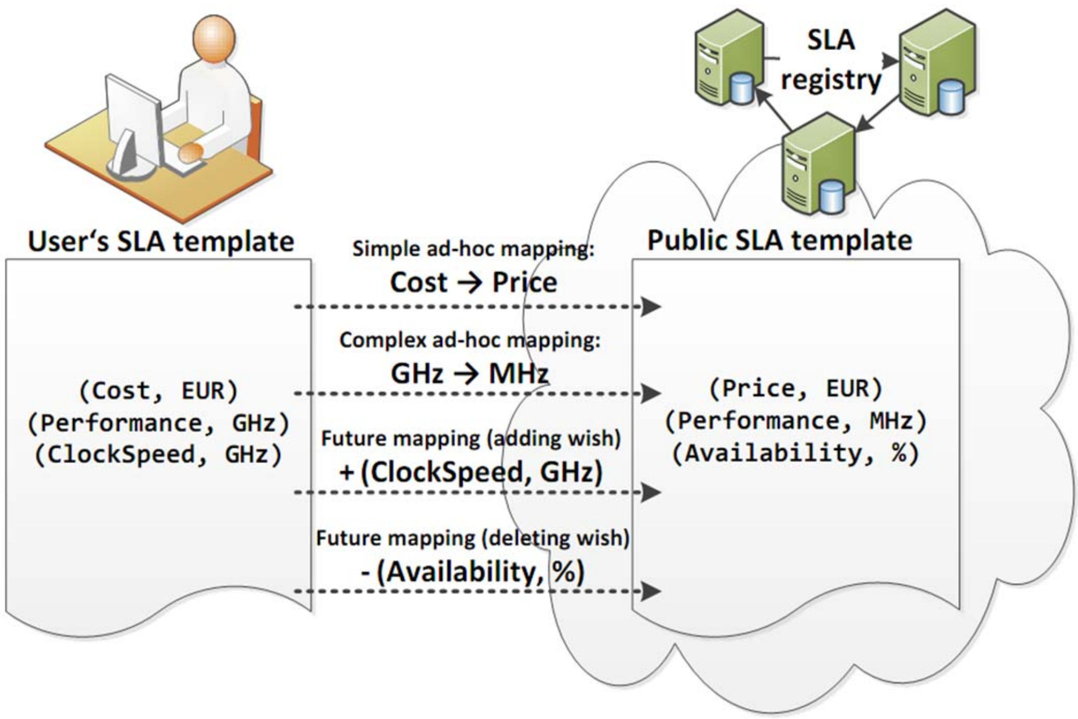


Figure 2. An example of the SLA mapping process.

Figure 2 represents a sample public SLA template and a service consumer's private SLA template. In this example, the consumer's private template differs from the public template in four ways: (1) the name of the SLA parameter “Price”, (2) the unit of a value of the parameter “Performance” (i.e., the parameter metric), (3) the parameter “Availability”, which does not exist in the consumer's private SLA template, and (4) the SLA parameter “ClockSpeed”, which exists in the consumer's private template but not in the public SLA template. Therefore, the consumer will create four SLA mappings to bridge these differences, namely (i) a simple ad-hoc mapping to map the difference of the parameter name, (ii) a complex ad-hoc mapping for defining a mapping function for measuring performance, (iii) a deleting wish (i.e., a future mapping) for wishing for the deletion of the “Availability” parameter from the public SLA template, and (iv) an adding wish (i.e., a future mapping) for adding a new parameter “ClockSpeed" into the public SLA template.

3.2. Evolution of Public SLA Templates

After a certain period of time, SLA mappings are evaluated in order to adapt public SLA templates to the current market trends. By utilizing adaptation methods, we enable the creation of new branches of public templates that reflect consumers' needs more precisely. For example, a general template for internal medicine applications can be substituted by more specialized templates on cardiology and oncology. The adaptation process is executed in two steps. First, by applying clustering algorithms, similar SLA mappings are grouped (step 5 in Figure 1). For example, one group of SLA mappings could be for those templates that name their parameters in a certain language. In the second step (i.e., step 6 in Figure 1), adaptation methods are applied to each of the groups. Those methods analyze the SLA mappings and decide about the content of new public SLA templates. At the end of the process, new public templates are generated. With this approach, public SLA templates are adapted according to the need of market participants. This increases the likelihood that new requests of consumers will need less SLA mappings, reducing the cost of consumers and making the market more attractive.

After new public SLA templates have been generated, users can decide if they want to use the newly generated templates or if they prefer keeping the old ones. Note, although new templates reflect their requirements, users have to define new SLA mappings, which incur cost.

4. Methods for Autonomic Market

Adaptation In this section, we discuss the algorithms and methods implemented for grouping similar SLA mappings and for generating new public SLA templates.

4.1. Clustering Algorithms for Grouping SLA Mappings

For the purpose of grouping similar sets of SLA mappings, we apply two clustering algorithms: DBSCAN and k-means. In our clustering algorithms, an clustering item (i.e., data point)is a set of consumer's SLA mappings, and a cluster centroid is a newly generated public SLA template for a group of consumers (i.e., private SLA templates).

DBSCAN is a data clustering algorithm that is based on the density distribution of nodes and finds an appropriate number of clusters [5]. The ϵ-neighborhood of a point p is defined as a set of data points that are not farther away from p than a given distance ϵ. A data point q is directly density-reachable from the point p, if q is in the ϵ-neighborhood of p, and the number of points in the ϵ-neighborhood of q is bigger than *MinPts*. A cluster satisfies two properties: (1) each two points of a cluster are mutually density-reachable, and (2) if a point is mutually density-reachable to any point of the cluster, the point is part of the cluster as well.

K-means is a clustering algorithm that, given an initial set of k means, assigns each data point to a cluster with the closest mean. It then calculates new means to be centroids of data points in the clusters and stops when the assignments no longer change [9]. A frequent problem in k-means algorithm is the estimation of the number k. Within this paper, we implemented two approaches for estimating k:

1. *Rule-of-Thumb* is a simple but very effective method, in which k is set to $\sqrt{N/2}$, where N is the number of entities [10].
2. *Hartigan's Index* is an internal index introduced in [7]. Let $W(k)$ represents the sum of squared distances between cluster members and cluster centroid for k clusters. When grouping n items, the optimal number k is chosen so that the relative change of $W(k)$ multiplied with the correction index $\gamma(k) = n - k - 1$ does not significantly change for $k + 1$, i.e.,

$$H(k) = \gamma(k)\frac{W(k)-W(k+1)}{W(k+1)} < 10 \tag{1}$$

The threshold *10* shown in Hartigan's index is also used in our simulations. It is "a crude rule of thumb" suggested by Hartigan [7].

Computing Distance Between SLA Templates In order to utilize clustering algorithms, the measure of distance between two clustering items must be defined, as well as the distance between a clustering item and a cluster centroid. As already mentioned in the previous section, an item for clustering is a set of consumer's SLA mappings, while a cluster centroid is an SLA template for the given group of consumers. Since having knowledge about the content of a public SLA template and about the SLA mappings a consumer created, we can exactly reconstruct consumer's private SLA template. Therefore, both the distance between two clustering items as well as between an item and a cluster centroid is equivalent to computing distances between two SLA templates.

An SLA template is defined through definitions of SLA parameters (i.e., their basic properties as, e.g., parameter name) and parameter metrics (i.e., methods for computing values of the parameters). The distance between two SLA templates is calculated by iterating through all SLA parameters contained in at least one of the SLA templates and calculating the distance between the parameter. The distance d_p between templates T_1 and T_2 in respect to an SLA parameter p and its properties (definition and metric) is defined as:

$$d_p(T_1, T_2) = \begin{cases} 0, \text{ if name and metric of } p \text{ are the same in } T_1 \text{ and } T_2 \\ 1, \text{ if } T_1 \text{ or } T_2 \text{ does not contain } p \\ 1, \text{if only name or metric of } p \text{ differs in } T_1 \text{ and } T_2 \\ 2, \text{if both name and mectic of } p \text{ differ in } T_1 \text{ and } T_2 \end{cases} \quad (2)$$

The total distance between two SLA templates is calculated as the sum of distances between all SLA parameters contained in at least one of the templates.

4.2. Adaptation Methods for Evolving Public SLA Templates

For enabling public SLA templates to react to market changes, we utilize three adaptation methods. These methods analyze the submitted SLA mappings and, for each SLA parameter, determine whether the current parameter name and parameter metric should be changed, or whether a new parameter should be added or an existing one deleted.

Table 1. Distribution of mappings for the SLA parameter *π*.

a) Name of the SLA parameter π

Name	*Cost*	*Charge*	*Rate*	*(Price)*
Mappings	*15%*	*15%*	*40%*	*(30%)*

b) Unit of the SLA parameter π

Name	*USD*	*GBP*	*YPI*	*(EUR)*
Mappings	*38%*	*2%*	*40%*	*(20%)*

Table 2. Distribution of mappings for the SLA parameter *μ*.

a) Name of the SLA parameter μ

Name	*MemConsumption*	*Consumption*
Mappings	*70%*	*30%*

b) Unit of the SLA parameter μ.

Name	*Mbit*	*Gbit*	*Tbit*
Mappings	*5%*	*55%*	*40%*

To demonstrate the methods, we use the following example. We consider the evolution of an SLA parameter π, which occurs in an initial public SLA template T_{init}, when adapting the initial public SLA template to the SLA mappings of 100 service consumers. The parameter π in the initial template is (*Price; EUR*), where the first member of the tuple represents the parameter name, and the second member represents the metric unit for expressing the parameter value (Table 1). The example also considers an SLA parameter μ which has not been defined in the initial public SLA template. Let *20* consumers have expressed a wish for deleting the parameter π from T_{init}, and the rest of them define mappings according to the distribution presented in Table 1. Note that the last column of the table represents the number of non-mapped values, i.e., the number of private SLA templates containing the same value as in the public SLA template. Let 75 consumers have expressed a wish for adding the parameter μ to the public template, with a name and unit according to the distribution depicted in Table 2.

The maximum method selects the option which has the highest number of SLA mappings (i.e., the maximum candidate). This candidate will then be used in the new SLA template. If there is more than one maximum candidate with the same number of SLA mappings, one of them is chosen randomly. In the given example, only *20%* of consumers wished for deleting the parameter π, which is not sufficiently high. Therefore, the parameter stays in the template with "Rate" as the new parameter name. The parameter name "Rate" got the highest number of mappings (*40%* in comparison to *30%* who did not create mappings, i.e., who want to keep the name "Price"). The price will be expressed in Japanese Yens due to the highest number of mappings. The parameter μ will also be in the new template, since there are more than *50* wishes for this parameter. Its parameter name and unit will be (*MemConsumption, Gbit*).

In order to lower the requirements for selecting the maximum candidate, **the threshold method** introduces a threshold value. In this method, the maximum candidate gets chosen only if it is used more than the given threshold. If more values satisfy the conditions, one of them is chosen randomly. Throughout the evaluation in this paper, we fix the threshold to *60%*. In the given example, neither the mappings for the name nor for the unit of the parameter π exceeds the threshold value. Therefore, no changes will happen. As for the parameter μ, it will be introduced in the new public template with the properties chosen according to the maximum method.

The significant-change method is a modification of the maximum-percentage-change method introduced in [11]. In this method, an SLA parameter property is only changed, if the percentage difference between the highest count value and the current public SLA template value exceeds a given threshold, which we assume to be significant at $\sigma_T > 15\%$. In the given example, *40* consumers have the name "Rate" for the "Price" parameter, while *30* consumers use the same name as in the public SLA template. Since the percentage difference of *33%* is higher than the given threshold, "Rate" will be chosen as the new name for the parameter. As the new unit, "YPI" will be chosen. Since "MemoryConsumption" does not exist in the old public SLA template, the decision about this SLA parameter is made as described for the maximum method.

5. Formalization of the Utility and Cost Model

As already mentioned in Section 1, the approach presented in this paper brings many benefits to market participants. However, it also incurs cost. Namely, whenever a public SLA template has been adapted, the users of the template have to define new SLA mappings. Within this section, we investigate these costs and assess the benefits of our approach. For this purpose, we introduce the utility and cost model.

In the following, we observe a set of SLA parameters $P_{N,F}$, where possible names of a parameter $\alpha \in P_{N,F}$ are $N(\alpha) = \{A, A', A'', \dots\}$, and the set of possible metrics is $F(\alpha) = \{f_\alpha, f'_\alpha, f''_\alpha, \dots\}$. Let there be a set of consumers' private SLA templates $C = \{c_1, c_2, \dots, c_n\}$ and two public SLA templates: the initial public SLA template T_{init}, and the newly generated public SLA template (for the given set of consumers) T_{new}.

For each SLA template $p \in C \cup \{T_{init}, T_{new}\}$, we define the relationship between a certain SLA parameter with its name and metric as:

$$SLA_p: P_{N,F} \rightarrow \bigcup_{\alpha \in P_{N,F}} (N(\alpha) \cup \emptyset) \times \bigcup_{\alpha \in P_{N,F}} (F(\alpha) \cup \emptyset) \quad (3)$$

where a parameter of a template is represented as a tuple (*name, metric*), i.e., for an SLA parameter in an SLA template p it holds $SLA_p\ (\alpha) \rightarrow (N(\alpha), F(\alpha))$. Assuming that $SLA_p(\alpha) = (A, f_\alpha)$, we can define the projection function Π as:

$$N_p(\alpha) = \Pi_N\left(SLA_p(\alpha)\right) = A, \quad F_p(\alpha) = \Pi_F\left(SLA_p(\alpha)\right) = f_\alpha \tag{4}$$

Additionally, we define the function to determine if an SLA template p contains an SLA parameter α:

$$X_p(\alpha) = \begin{cases} 0, & \text{if } p \text{ does not contain } \alpha \\ 1, & \text{if } p \text{ contain } \alpha \end{cases}$$

Finally, we can define the utility function for a consumer c and for an SLA α as follows:

$$u_c^+(\alpha) = \begin{cases} 0, & X_c(\alpha) \neq X_{T_{new}}(\alpha) \vee X_c(\alpha)X_{T_{new}}(\alpha) = 0 \\ A, & X_c(\alpha) = X_{T_{new}}(\alpha) = 1 \wedge N_c(\alpha) \neq N_{T_{new}}(\alpha) \wedge F_c(\alpha) \neq F_{T_{new}}(\alpha) \\ B, & X_c(\alpha) = X_{T_{new}}(\alpha) = 1 \wedge ((N_c(\alpha) = N_{T_{new}}(\alpha) \wedge F_c(\alpha) \neq F_{T_{new}}(\alpha)) \vee \\ & (N_c(\alpha) \neq N_{T_{new}}(\alpha) \wedge F_c(\alpha) = F_{T_{new}}(\alpha))) \\ C, & X_c(\alpha) = X_{T_{new}}(\alpha) = 1 \wedge SLA_c(\alpha) = SLA_{T_{new}}(\alpha) \end{cases}$$

As stated in the definition of the utility function, the consumer gains no utility for an SLA parameter, if it does not exist in one of the SLA templates (consumer's private template or the public template). If it does exist, the utility can be *A* (if both name and metric for an SLA parameter differ in user's private SLA template and the public SLA template), *B* (if only name or metric differ), or *C* (if templates do not differ with respect to the parameter).

The cost function for a consumer c and an SLA parameter α is defined as follows:

$$u_c^-(\alpha) = \begin{cases} 0, & SLA_{T_{new}}(\alpha) = SLA_{T_{init}}(\alpha) \vee SLA_c(\alpha) = SLA_{T_{new}}(\alpha) \vee \\ & (X_c(\alpha) = X_{T_{new}}(\alpha) = X_{T_{init}}(\alpha) = 1 \wedge \\ & ((N_c(\alpha) = N_{T_{new}}(\alpha) \wedge F_{T_{new}}(\alpha) = F_{T_{init}}(\alpha)) \vee \\ & (N_{T_{new}}(\alpha) = N_{T_{init}}(\alpha) \wedge F_c(\alpha) = F_{T_{new}}(\alpha)))) \\ E, & X_{T_{init}}(\alpha) \neq X_{T_{new}}(\alpha) \wedge X_c(\alpha) \neq X_{T_{new}}(\alpha) \\ D, & \text{otherwise} \end{cases}$$

The cost function states that a user has no cost, if he does not have to create new SLA mappings, i.e., if the parameter properties are the same in his private template and in the newly generated public template, or the new template did not change with respect to the parameter. If a user has to create a future mapping, the cost is equal to *E*. Finally, if a user has to create ad-hoc mappings for name, metric, or both, the cost is equal to *D*.

At the end, we define the overall net utility as

$$U^O = \sum_{c\in C}\sum_{\alpha\in P_{N,F}} u_c^+(\alpha) - \sum_{c\in C}\sum_{\alpha\in P_{N,F}} u_c^-(\alpha) \tag{5}$$

Note that the more similar SLA templates are, the more a user is attracted to the market and gets higher utility. Considering the cost, we assume that creating a future SLA mapping incurs more cost than creating an ad-hoc SLA mapping, due to the mapping complexity. Therefore, although the values of the utility and the cost functions are not strictly defined, considering the effort or benefit, the following relation should hold $0 \leq A \leq B \leq C$ and $0 \leq D \leq E$.

6. Evaluation

In this section, we measure and investigate the benefits and costs of defining SLA mappings and adapting public SLA templates to the current needs of market participants, using the utility and cost model presented in Section 5. Moreover, we assess the scalability of the approach and discuss the quality of newly generated public SLA templates.

6.1. Simulation Environment

For the simulation of our approach, we designed a testbed within the VieSLAF environment. VieSLAF is a framework for semi-automated management of SLA mappings and for generating public SLA templates [1]. The major VieSLAF components are the knowledge base and the SLA mapping middleware.

1. The knowledge base is used for storing and managing public SLA templates. Browsing and creating SLA templates is enabled over searchable repositories, which are implemented in Microsoft SQL 2008 DBMS with a Web service front-end to provide the interface for the SLA mapping management.
2. The SLA mapping middleware is based on several Windows Communication Foundation services. The middleware comprises administration (e.g., creation of SLA templates), accounting (e.g., creation of consumer accounts), querying (e.g., finding an appropriate SLA template), management of SLA mappings (e.g., defining mappings for a user), and adaptation of public SLA templates.

Both VieSLAF components can be accessed only with appropriate access rights depending on three access roles: service consumer, service provider, and registry administrator.

The simulation is conducted following the scenario described in Section 3. The process is evaluated by investigating the overall net utility (Equation 5). For evaluation purposes, all of above-mentioned clustering algorithms and adaptation methods for creating new public SLA templates are used and compared. The number of service consumers creating SLA mappings varies from *100* to *10000*. The initial public SLA template and contains *8* SLA parameters, while the consumers' private SLA templates are generated randomly at the beginning of the evaluation process and can contain up to *11* SLA parameters, where, for each SLA parameter, consumers need to choose between *5* names and *2* metrics.

6.2. Evaluation Results

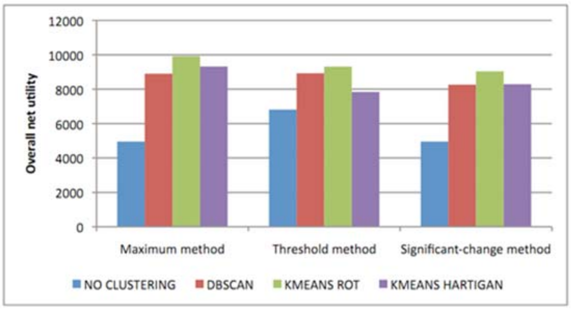
Overall Net Utility To assess the benefits and costs of the SLA mapping approach, we use the overall net utility defined by Equation (5), where the values of the constants of the utility and cost model are fixed as depicted in Table 3. For the evaluation, we simulate *500* service consumers creating SLA mappings to a public SLA template. In order to compare results achieved with the different methods, all of the above-mentioned clustering algorithms and adaptation methods have been used and compared. After the new SLA templates are generated, the overall net utility is calculated, representing the difference between gains and costs of the SLA mapping approach. The simulation results are depicted in Figure 3(a).

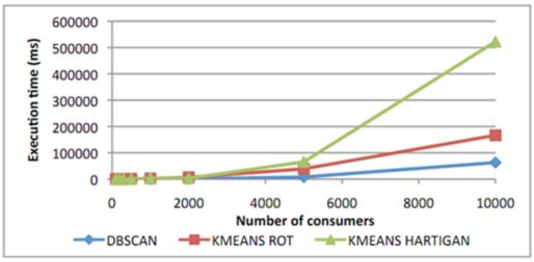
Table 3. Distribution of mappings for the SLA parameter π.

Variable	*A*	*B*	*C*	*D*	*E*
Value	*1*	*2*	*3*	*1*	*2*

Using the results depicted in the graph, we can compare the approach described in this paper with the work on SLA mapping described in [11], where clustering algorithms were not utilized. In [11], the adaptation methods have been applied to create a single new public SLA template for all market participants. The first column of the graph presents the overall net utility achieved when creating a single new public template, while the rest of the columns represent results achieved by utilizing different clustering algorithms. As it can be seen from the graph, by generating only one public SLA template for all service consumers, the overall net utility is significantly lower than in the other cases. This is due to creating more new public SLA templates that reflect consumers' needs more precisely.

When comparing the overall net utility gained by utilizing different adaptation methods, we can conclude that the best results are achieved by using the maximum method. Although the threshold method causes significantly lower number of changes (due to the high threshold), and therefore very low cost, the maximum method adapts public templates more dynamically and achieves higher utility rates. The highest overall net utility is achieved by the k-means clustering algorithm with the rule-of-thumb method, due to creating large number of very specific clusters.





(a) Overal net utility (b) Simulation time

Figure 3. Evaluation results.

The utility rate highly depends on the number of generated clusters. The more clusters are created, the higher the utility will be. This relation can be seen when combining the results depicted in Figure 3(a) with Table 4, representing the number of clusters per clustering algorithm and per number of service consumers creating SLA mappings. The maximum overall net utility for n service consumers can be achieved by creating n clusters, i.e., by generating one public SLA template per consumer. In this case, the public SLA templates would be equal to consumers' private templates. Moreover, the utility rate would be maximum, and the cost would be equal to 0. However, by raising the number of new public SLA templates in the market, the liquidity of goods decreases. Therefore, we must find a balance between achieving high net utility and ensuring high liquidity of market goods. The problem of finding the optimal number of new public SLA templates to be introduced to the market will be addressed in our future work.

Table 4. Number of generated clusters per algorithm and number of service consumers.

	DBSCAN	*k*-means (rule-of-thumb)	*k*-means (Hartigan's index)
100	5	7	3
200	5	10	3
300	6	12	3
500	6	15	5
1000	7	22	6
2000	7	31	6
5000	8	50	6
10000	10	69	10

Since the number of generated clusters influences the overall net utility, we conclude that both the overall net utility and the dataset properties determine the quality of newly generated public SLA templates. Therefore, it is not sufficient to investigate only the utility gained, but also the properties of the simulation dataset that may have influenced the results of the utility measurements. For this purpose, we introduce measures of compactness and isolation of clusters. In our discussion of dataset properties, *C* represents a set of clusters, where $C_j \in C$ is a cluster, and C_{jc} its centroid. R_i is a clustering item, and $D(R_i,R_k)$ a distance between two items. $|C|$ represents the total number of clusters, and $|C_i|$ represents a number of items in a cluster C_i.

Compactness is reciprocal of average distance between consumers' private SLA templates within clusters, and is formally defined as follows:

$$\left(\frac{1}{|C|}\sum_{C_j \in C}\left(\frac{1}{|C_j|^2}\sum_{R_i \in C_j}\sum_{R_k \in C_j} D(R_i, R_k)\right)\right)^{-1} \tag{6}$$

Since a cluster represents a group of consumers with similar requirements, high compactness implies a well-formed market. Figure 4(a) depicts the compactness rates achieved by the three clustering algorithms. For this purpose, we have simulated SLA mappings specified by different number of service consumers, varying from *100* to *10000*. The adaptation method used for the evaluation is the maximum method.

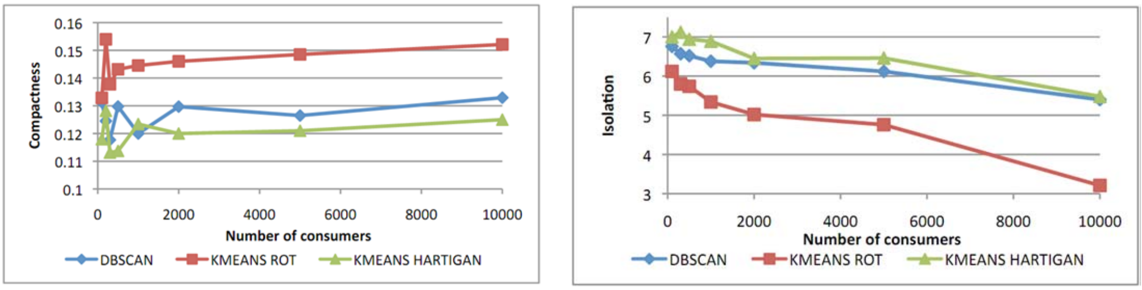
High utility is common for compact clusters, since all members of the cluster have similar properties, and the cluster centroid, i.e., the newly generated SLA template, reflects those properties more precisely. On the other side, items of a less compact cluster differ to a larger extent, and it is probable that the utility achieved by creating a new public SLA template will be lower. High compactness can be achieved by creating large number of smaller clusters and is maximized, as mentioned before, when there is one cluster per item.

As depicted in Figure 4(a), the k-means algorithm with the Hartigan's index achieves lowest rate of compactness. This is due to creating small number of general clusters, where items mutually differ to a large extent. Therefore, this algorithm results in low overall net utility. The k-means algorithm with the-rule-of-thumb achieves the largest rate of compactness, and consequently a larger overall net utility, due to a large number of clusters.

Isolation is an average distance between newly generated public SLA templates. When using the formulations given in the previous section, isolation can be defined as follows.

$$\frac{1}{|C|^2}\sum_{C_i \in C}\sum_{C_k \in C} D(R_{ic}, R_{kc}) \tag{7}$$

Isolation represents the measure of how different newly generated public SLA templates are. The larger the distance, the more distinct the templates are. We imply that higher isolation of public templates ensures more distinct products, and therefore provides better chances for creating product niches. Figure 4(b) presents the isolation rates achieved.



(a) Compactness (b) Isolation

Figure 4. Simulation: dataset properties.

High isolation can be achieved by creating a small number of very distinct clusters. However, by doing so, we are reducing the compactness of the clusters, since they will contain larger number of items with broader variety of preferences. Therefore, we must find a balance between creating more general clusters with high isolation (i.e., distinct market products) and well-formed clusters with respect to their specificity and compactness.

As it can be seen in Figure 4(b), k-means algorithm with the Hartigan's index achieves a high rate of isolation. This result is not surprising, since this algorithm creates a small number of clusters and, as presented in Figure 4(a), achieves the lowest rate of compactness. Additional to that, this algorithm achieves the smallest rate of the overall net utility. DBSCAN also creates relatively small number of new public templates and creates well-isolated clusters. Finally, the *k*-means with the rule-of-thumb creates clusters with low isolation, due to creating a large number of clusters.

Scalability For Cloud environments, it is of great importance that resource markets are highly scalable. In order to assess the scalability of our approach, we measure the execution time of the clustering process for different number of users creating SLA mappings, varying from *100* to *10000*. Results are depicted in Figure 3(b).

As shown in the figure, the *k*-means algorithm with the Hartigan's index takes most time to compute groups of SLA mappings. This is due to the nature of the Hartigan's index method, in which the clustering algorithm runs in several iterations and computes different number of clusters before it finds the optimal number *k*. The best performance is achieved by the DBSCAN algorithm. This result is also not surprising, since unlike the *k*-means algorithms, it creates a low number of clusters and does not compute cluster centroids, which incurs cost.

The results show that the approach presented in this paper is highly scalable. Moreover, the execution time of *k*-means algorithm with the Hartigan's index, which is far the highest, can be improved by introducing heuristics for determining the initial number *k*, which we will consider in our future work.

7. Conclusion and Future Work

In this paper, we have extended the SLA mapping approach by introducing the idea of applying clustering algorithms to group users' SLA mappings. This enables the automatic creation of new public SLA templates, paving the way for autonomic market management. In particular, we utilized two clustering algorithms for grouping SLA mappings and used three adaptation methods for generating new SLA templates. We have investigated the utility and cost to users when applying the adaptive SLA mapping approach. Our simulation results show that our approach facilitates continuous market evolution and adaptation of public SLA templates. It allows responding to market trends and changes.

In the future, we will consider heuristic methods for improving the efficiency of the clustering algorithms for grouping SLA mappings. Besides, we will address the problem of finding the optimal number of new public SLA templates to be introduced to the market, in order to ensure liquidity on the market while keeping a high net utility rate.

Acknowledgement

This work is supported by the Vienna Science and Technology Fund under the grant agreement ICT08-018 Foundations of Self-Governing ICT Infrastructures (FoSII), and by the National Research Foundation of the Ministry of Education, Science and Technology of Korea under the grant K21001001625-10B1300-03310.

References

[1] Brandic, I., Music, D., Dustdar, S.: VieSLAF framework: Facilitating negotiations in clouds by applying service mediation and negotiation bootstraping (2010).

[2] Buyya, R., Abramson, D., Giddy, J.: A case for economy grid architecture for service oriented grid computing. Parallel and Distributed Processing Symposium 2 (2001).

[3] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25, 599 { 616 (2009).

[4] Dobson, G., Sanchez-Macian, A.: Towards unified QoS/SLA ontologies. In: Services Computing Workshops, 2006. SCW '06. IEEE. pp. 169 {174 (2006).

[5] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. pp. 226{231. AAAI Press (1996).

[6] Green, L.: Service level agreements: an ontological approach. In: 8th international conference on Electronic commerce. pp. 185{194. ICEC '06, ACM (2006).

[7] Hartigan, J.A.: Clustering Algorithms. John Wiley & Sons Inc (1975).

[8] Karänke, P., Kirn, S.: Service level agreements: An evaluation from a business application perspective. In: eChallenges e-2007. pp. 104{111. IEEE-CS Press (2007).

[9] MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 281{297. University of California Press (1967).

[10] Mardia, K.V., Kent, J.T.: Multivariate Analysis. Academic Press (1980).

[11] Maurer, M., Emeakaroha, V.C., Risch, M., Brandic, I., Altmann, J.: Cost and benefit of the SLA mapping approach for defining standardized goods in cloud computing markets. In: International Conference on Utility and Cloud Computing (UCC 2010) in conjunction with the International Conference on Advanced Computing (ICoAC 2010) (2010).

[12] Neumann, D., Stösser, J., Weinhardt, C.: Bridging the adoption gap-developing a roadmap for trading in grids. Electronic Markets 18, 65{74 (February 2008).

[13] Nimis, J., Anandasivam, A., Borissov, N., Smith, G., Neumann, D., Wirstrm, N., Rosenberg, E., Villa, M.: SORMA: Business cases for an open grid market: Concept and implementation. In: Grid Economics and Business Models, Lecture Notes in Computer Science, vol. 5206, pp. 173{184. Springer Berlin / Heidelberg (2008).

[14] Oldham, N., Verma, K.: Semantic WS-agreement partner selection. In: 15th international conference on World Wide Web. WWW '06. pp. 697{706. ACM Press (2006).

[15] Risch, M., Brandic, I., Altmann, J.: Using SLA mapping to increase market liquidity. In: Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops, Lecture Notes in Computer Science, vol. 6275, pp. 238{247. Springer (2010).

[16] Samuelson, P., Nordhaus, W.: Economics. McGraw-Hill/Irwin, 18. ed., internat. ed. edn. (2005).

TEMEP Discussion Papers

2009-36: Jörn Altmann and Juthasit Rohitratana, "Software Resource Management Considering the Interrelation between Explicit Cost, Energy Consumption, and Implicit Cost: A Decision Support Model for IT Managers"

2009-37: Marcel Risch, Ivona Brandic and Jörn Altmann, "Using SLA Mapping to Increase Market Liquidity"

2009-38: Amit K. Bhandari and Almas Heshmati, "Willingness to Pay for Biodiversity Conservation"

2010-39: Marcel Risch, Jörn Altmann, Li Guo, Alan Fleming and Costas Courcoubetis, "The GridEcon Platform: A Business Scenario Testbed for Commercial Cloud Services"

2010-40: Junseok Hwang, Jörn Altmann and Ashraf Bany Mohammed, "Determinants of Participation in Global Volunteer Grids: A Cross-Country Analysis"

2010-41: Ruzana Davoyan, Jörn Altmann and Wolfgang Effelsberg, "Intercarrier Compensation in Unilateral and Bilateral Arrangements"

2010-42: Ruzana Davoyan, Jörn Altmann and Wolfgang Effelsberg, "Exploring the Effect of Traffic Differentiation on Interconnection Cost Sharing"

2010-43: Ruzana Davoyan and Jörn Altmann, "Investigating the Role of a Transmission Initiator in Private Peering Arrangements"

2010-44: Ruzana Davoyan, Jörn Altmann and Wolfgang Effelsberg, "A New Bilateral Arrangement between Interconnected Providers"

2010-45: Marcel Risch and Jörn Altmann, "Capacity Planning in Economic Grid Markets"

2010-46: Dang Minh Quan and Jörn Altmann, "Grid Business Models for Brokers Executing SLA-Based Workflows"

2010-47: Dang Minh Quan, Jörn Altmann and Laurence T. Yang, "Error Recovery for SLA-Based Workflows within the Business Grid"

2010-48: Jörn Altmann, Alireza Abbasi and Junseok Hwang, "Evaluating the Productivity of Researchers and their Communities: The RP-Index and the CP-Index"

2010-49: Jörn Altmann and Zelalem Berhanu Bedane, "A P2P File Sharing Network Topology Formation Algorithm Based on Social Network Information"

2010-50: Tai-Yoo Kim, Seunghyun Kim and Jongsu Lee, "The Gene of an Accelerating Industrial Society: Expansive Reproduction"

2010-51: Almas Heshmati and Sangchoon Lee, "The Relationship between Globalization, Economic Growth and Income Inequality"

2010-52: Flávio Lenz-Cesar and Almas Heshmati, “Agent-based Simulation of Cooperative Innovation”

2010-53: Erkhemchimeg Byambasuren and Almas Heshmati, “Economic Development in Mongolia”

2010-54: Almas Heshmati and Subal C. Kumbhakar, “Technical Change and Total Factor Productivity Growth: The Case of Chinese Provinces”

2010-55: Bory Seng and Almas Heshmati, “Digital Divide and Its Variations Amongst OECD, NIE and ASEAN Countries”

2010-56: Kibae Kim, Jörn Altmann and Junseok Hwang, “The Impact of the Subgroup Structure on the Evolution of Networks: An Economic Model of Network Evolution”

2010-57: Kibae Kim, Jörn Altmann and Junseok Hwang, “Measuring and Analyzing the Openness of the Web2.0 Service Network for Improving the Innovation Capacity of the Web2.0 System through Collective Intelligence”

2010-58: Alireza Abbasi and Jörn Altmann, “A Social Network System for Analyzing Publication Activities of Researchers”

2010-59: Jörn Altmann, Costas Courcoubetis and Marcel Risch, “A Marketplace and its Market Mechanism for Trading Commoditized Computing Resources”

2010-60: Fatmawati Zifa and Jörn Altmann, “A empirically validated framework for limiting free-riding in P2P network through the use of social network”

2010-61: Ashraf Bany Mohammed, Jörn Altmann and Junseok Hwang, “Cloud Computing Value Chains-Understanding Business and Value Creation in the Cloud”

2010-62: Ashraf Bany Mohammed and Jörn Altmann, “A Funding and Governing Model for Achieving Sustainable Growth of Computing e-Infrastructures”

2010-63: Junseok Hwang, Jihyoun Park and Jörn Altmann, “Two Risk-Aware Resource Brokering Strategies in Grid Computing: Broker-driven vs. User-driven Methods”

2010-64: Juthasit Rohitratana and Jörn Altmann, “Agent-Based Simulations of the Software Market under Different Pricing Schemes for Software-as-a-Service and Perpetual Software”

2010-65: Radoslaw R. Okulski and Almas Heshmati, “Time Series Analysis of Global Airline Passengers Transportation Industry”

2010-66: Alireza Abbasi and Jörn Altmann, “On the Correlation between Research Performance and Social Network Analysis Measures Applied to Research Collaboration Networks”

2010-67: Kibae Kim, Jörn Altmann and Junseok Hwang, "An Analysis of the Openness of the Web2.0 Service Network Based on Two Sets of Indices for Measuring the Impact of Service Ownership"

2010-68: Bory Seng, "The Driving Forces Underlying the Growth of Total Factor Productivity in Cambodia, Quantitative and Qualitative Studies"

2010-69: Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic, and Jörn Altmann, "Cost and Benefit of the SLA Mapping Approach for Defining Standardized Goods in Cloud Computing Markets"

2010-70: Khin Swe Latt and Jörn Altmann, "A Cost-Benefit-Based Analytical Model for Finding the Optimal Offering of Software Services"

2010-71: Jung Eun Lee, Younghoon Kim, Yeonbae Kim and Donghyuk Choi, "The Impact of Technology Licensing Payment Mechanisms on Firms' Innovative Performance"

2010-72: Dongook Choi and Yeonbae Kim, "Effects of Piracy and Digital Rights Management on the Online Music Market in Korea"

2011-73: Tai-Yoo Kim, Jihyoun Park, Eungdo Kim and Junseok Hwang, "The Faster-Accelerating Digital Economy"

2011-74: Ivan Breskovic, Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic and Jörn Altmann, "Towards Autonomic Market Management in Cloud Computing Infrastructures"

2011-75: Kiran Rupakhetee and Almas Heshmati, "Rhetorics vs. Realities in Implementation of e-Government Master Plan in Nepal"

2011-76: Alireza Abbasi, Jörn Altmann and Liaquat Hossain, "Identifying the Effects of Co-Authorship Networks on the Performance of Scholars: A Correlation and Regression Analysis of Performance Measures and Social Network Analysis Measures"

2011-77: Ivona Brandic, Michael Maurer and Vincent C. Emeakaroha, Jörn Altmann, "Cost-Benefit Analysis of the SLA Mapping Approach for Defining Standardized Cloud Computing Goods"

2011-78: Kibae Kim and Jörn Altmann, "A Complex Network Analysis of the Weighted Graph of the Web2.0 Service Network"

2011-79: Jörn Altmann, Matthias Hovestadt and Odej Kao, "Business Support Service Platform for Providers in Open Cloud Computing Markets"

2011-80: Ivan Breskovic, Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic and Jörn Altmann, "Achieving Market Liquidity through Autonomic Cloud Market Management"