

Cloudysme: An Ontological Framework for Aiding SMEs Adoption of SaaS in a Cloud Environment

Richard Ikechukwu Otuka, Abdel-Rahman Tawil, Ameer Al-Nemrat

School of Architecture Computing & Engineering, University of East London, London, UK

Email: u0749744@uel.ac.uk, a.r.tawil@uel.ac.uk, a.al-nemrat@uel.ac.uk

How to cite this paper: Otuka, R.I., Tawil, A.-R. and Al-Nemrat, A. (2017) Cloudysme: An Ontological Framework for Aiding SMEs Adoption of SaaS in a Cloud Environment. *Journal of Computer and Communications*, 5, 86-112.

<https://doi.org/10.4236/jcc.2017.514008>

Received: November 5, 2017

Accepted: December 26, 2017

Published: December 29, 2017

Copyright © 2017 by authors and

Scientific Research Publishing Inc.

This work is licensed under the Creative

Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Adoption techniques are widely applied in and for cloud service usage to improve the slow acceptance rate of cloud services by SMEs. In such context, a well-understood problem is finding a suitable service from the vast number of services offering similar packages to satisfy user requirements such as security, cost, trust and operating systems compatibility has become a big challenge. However, a major drawback of existing techniques such as frameworks, web search, decision support tools, management models, ontology models and agent technology is that they are restricted to a specific task or they replicate service provider offerings. In this paper, we present Cloudysme a cloud service adoption solution, a middleware that is capable of aiding the decision making process for SMEs adoption of cloud services. Using a case study of SaaS storage services offerings by cloud providers, we introduce a new formalism for judging the superiority of one service attribute over another, we propose an extended version of pairwise comparison and Analytical hierarchical Process (AHP) which is a traditional multi-criteria decision method (MCDM) in solving complex comparisons. We solve the issue of service recommendation by introducing an acceptable standard for each service attribute and propose a protocol using rational relationships for aiding cloud service ranking process. We tackle the issue of specific tasking by using a set of concepts and associated semantic rules to rank and retrieve user requirements. We promote a knowledge engineering approach for natural language processing by using terms and conditions in translating human sentences to machine readable language. Finally, we implement our system using 30 SMEs as a pivotal study. We prove that the use of semantic rules within an ontology can tackle the issue of specific tasking.

Keywords

SaaS, Ontology, AHP, DSS, SME, Service Ranking, Knowledge Management

1. Introduction

Cloud computing provides a mix of technologies that enhance resource utilization at reduced cost. However, despite the efforts made to enable easy accessibility to this technology, the presence of different service providers rendering similar services at different cost, performance level and different features has made it complex for SME owners to adopt cloud services to meet their business needs. Furthermore, given the diversity of cloud computing services offerings, an important challenge for customers is the complexity in discovering the “Right” cloud service that meets their requirement. Although there may be variances between different functional and non-functional requirements offered by different service providers, this makes it difficult to evaluate cloud service offerings of different providers in a way such that reliability, quality and user satisfaction is ensured. Therefore, it is insufficient to just adopt any cloud service as it is more important to evaluate which cloud service is most suitable for a company business needs.

Nevertheless, existing adoption techniques and middleware are restricted to a specific task or they replicate service provider offerings without any form of evaluation and this may not be the best solution to match user requirements [1]. In their approach, a cloud service ranking technique was proposed for IaaS services based on service provider offerings using AHP method. [2] proposed an ontology of cloud service governance. [3] [4] [5] proposed an ontology of cloud service knowledge management. [6] proposed a cloud service recommendation ontology. [7] proposed an ontology based on service discovery in different service layers. While the work of [8] proposed a cloud service provider resource ontology. The functional and non-functional properties of cloud services make it difficult for some cloud service attributes to be measured. Therefore deciding which service best meets a user requirement is a decision problem. Such problem is referred to as a Multi-Criteria Decision Making (MCDM) [9]. Whereby each individual attributes affects the cloud service selection process and its overall impact in adoption process.

This paper presents cloud service concepts modelled as part of a semantically engineered middleware which can be used for cloud service ranking, cloud services discovery, knowledge management and service recommendation to aid service adoption decision. The developed semantic model (ontology) is built based on a set of semantic rules to aid SMEs in cloud service adoption process thereby selecting the most relevant service that meets their business process requirement. Our main contributions are as follows:

- Firstly, we observe that there is a general lack of intelligent systems that help SMEs in the adoption decision process particularly for SaaS storage services. As present works are either paper based or specific to other aspects of cloud computing. Therefore, we call on a novel approach in the cloud service adoption technique by integrating the use of standard benchmarks to determine the acceptability of cloud service attribute to meet user requirement in a ser-

vice adoption process.

- Secondly, as the issue of cloud service comparison is complex because of the presence of various service providers rendering similar services. This sort of complexity is termed as a multi-criteria decision problem. We propose a new formalism in our judgement of cloud service attributes, we adopt pairwise comparison to quantify our judgement. We adopt Analytical Hierarchical Process for the assigning of weights in order to prioritize cloud service attributes. We also extend this process by introducing a set of procedural formalization and protocol in a bid to rank cloud services not only based on highest attained accumulated weight but also on their ability to meet certain consumer standards. As users satisfaction is the main focus of our ranking process.
- Thirdly, we propose a layered architecture of our approach which comprises of middleware to realize our contribution. The Middleware comprises of a semantic model which gives us the ability to transform human language to machine readable language using an OWL ontology supported by description logic reasoning engine such as (pellet or hermiT) which has reasoning capabilities to infer knowledge based concepts and relationships in view of retrieving accurate and timely information to aid decision making. We introduce a set of concepts and associated semantic rules to tackle the issue of specific tasking and presenting our service ranking in machine readable form.

The resulting middleware represents a cloud service semantic-based approach which can be used for cloud service ranking, service discovery, cloud service knowledge management and service recommendation towards cloud service adoption decision making for SMEs with specific focus on SaaS as a storage service. The ability to perform the above tasks shows the multi-decision aiding nature of our system and overcoming the traditional specific tasking nature of other systems. To the best of our knowledge we are the first to present an MCDM approach of cloud service adoption technique in an intelligent system as most researchers who compare cloud service offerings present them in a tabular form which is still a complex procedure for potential cloud service adopters. A case study of advertised SaaS storage offering is used to demonstrate how our framework works.

The rest of this paper is organized as follows: Section 2 shows our system architecture; Section 3: MCDM approach; Section 4: Case Study; Section 5: Ontology Build-up; Section 6: Implementation and Service Ranking; Section 7: Implementation; Section 8: Related works; Section 9: Discussion; Section 10: Conclusion and future works.

2. Cloudysme System Architecture

We propose a decision support framework which is equipped with a semantic model to aid SME owners/managers in the decision-making process for SaaS cloud services adoption. The proposed framework comprises of four different

phases. The first phase comprises of information gathering of advertised cloud service catalogue by service providers. This shows the characteristics of advertised SaaS storage services based on how they meet users' requirement, the functional and non-functional properties of SaaS storage services of four major service providers are being considered in this phase based on their level of user friendliness in pricing, compatibility, trust, security, storage allowance, operating systems supported, availability of Office via web, Free storage. The second phase is the Filtering phase: In this phase, a set of formalism is used to compare two similar attributes from different cloud services, Pairwise comparison is used to judge the superiority of one cloud service offerings over another in quantifiable values. Weights are assigned to cloud service attributes [10] in applying AHP as a Multi-Criteria Decision Making method. A set benchmark is determined by the priority value of each KPI after comparing and normalizing the importance of one KPI over another based from a user perspective. We introduce a set of protocol for ranking each cloud service based on a set of protocol. The third phase is development of an ontology of cloud service which is developed as a middleware which helps us translate human language to machine readable language in order to achieve our system goals. The fourth phase is the implementation phase where we use a set of concepts and associated semantic rules to rank and retrieve user requirements towards cloud service adoption and this is demonstrated using a case study. The final stage is the decision making which comprises of knowledge management, service recommendation, service ranking, service discovery and adoption decision.

The architecture in **Figure 1** shows the sequence of events that takes place within CLOUDYSME framework in view of aiding an SME owner/manager in the decision making process towards adopting a cloud service for his business. The next section explains the multi-criteria decision making method.

3. Multi-Criteria Decision Method

The cloud service ranking which is vital for the system development is based on the quality of services advertised by service providers and how they meet user requirements. As mentioned earlier, the task of ranking attributes of cloud services is very complex. This challenge is described as a multi-criteria decision making (MCDM) problem [9]. To address this challenge we adopt an extended version of Analytical Hierarchical Process (AHP) which is one of the most widely used method for tackling problems related to MCDM. To justify the adoption of AHP as our decision making method, we compare its advantage over other MCDM approach. Comparing AHP with multi-attribute value (MAV) [11] confirmed that both methods are widely acceptable and successful means of decision-making but MAV fails to incorporate systematic checks on the consistency of judgement compared to AHP. Belton further concluded that the large evaluations and judgements required by AHP can be somewhat complex. Comparing AHP with outranking method, the outranking method is applied when measurement units are unmeasurable and aggregation of criteria

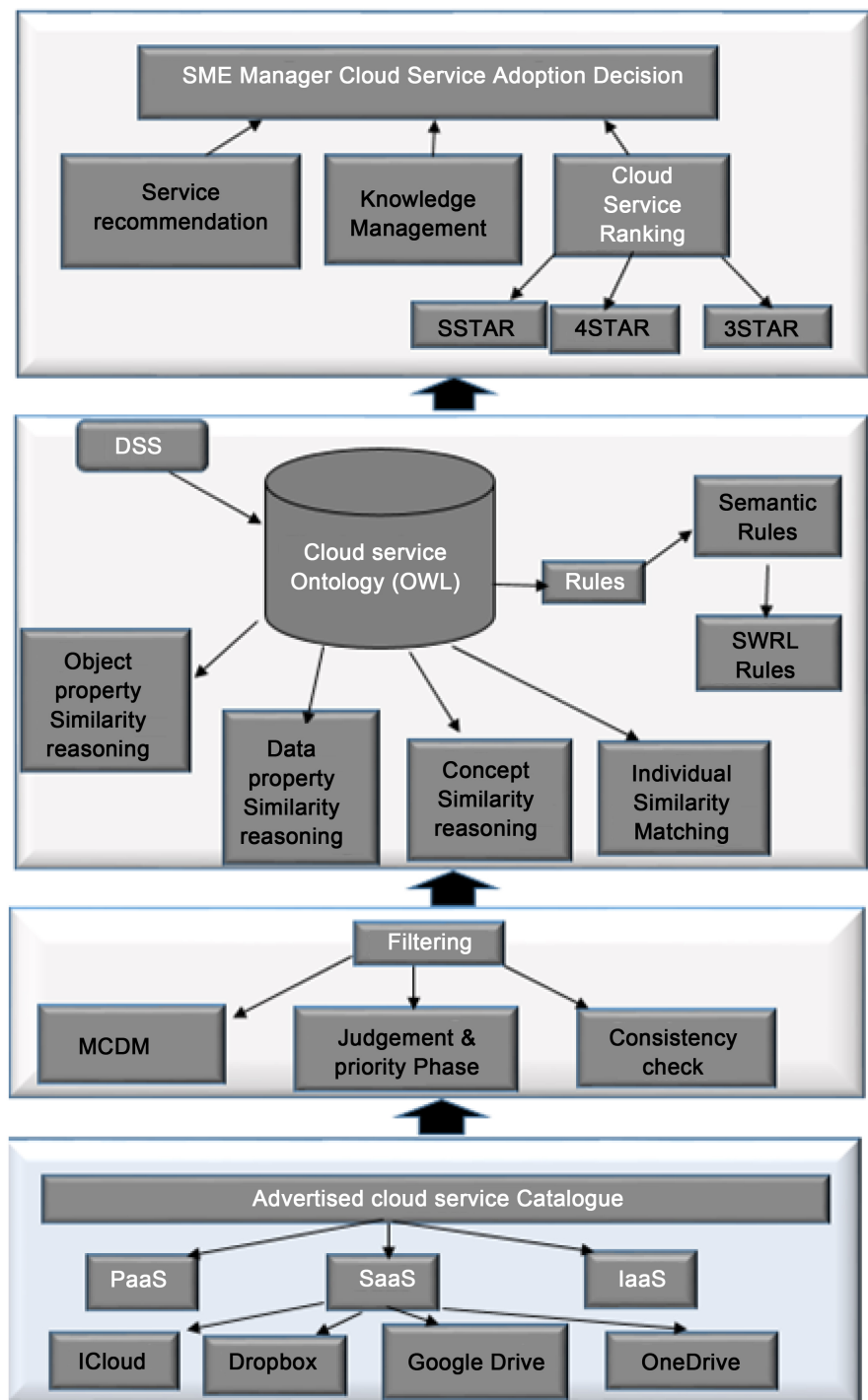


Figure 1. Cloudysme framework.

metrics is complex. The limitation of this approach is that in most cases it is inconclusive in decision making compare to other MCDM approaches. Although AHP is widely adopted among the MCDM methods, some researchers criticize its approach. [12] [13] are in the view that the method can suffer from rank reversal. [14] criticized the AHP method on the grounds that it lacks a firm theoretical basis. On the contrary, [15] [16] discussed these major criticisms and us-

ing theoretical work and examples proved they are invalid. They concluded that AHP approach is based on a firm theoretical foundation, it is a viable and useable decision-making tool.

3.1. Extended AHP Approach for Service Ranking

Based on the above advantages of AHP method over other MCDM problem, we are proposing an extended version of AHP in addressing the issue of comparing and assigning of quantifiable weights, setting of acceptable benchmarks for each KPI, we are introducing a different ranking approach compared to the traditional AHP ranking method. Generally, The AHP method tends to simplify complex and unstructured problems by arranging the decision factor in hierarchical structure. It is based on pairwise comparison which allows decision makers to determine the balance among criteria. Also its ability to check consistency, flexibility and intuitive appeal to decision makers gives it an advantage over other multi-criteria methods. In addition, AHP captures both subjective and objective evaluation methods while offering a powerful mechanism that checks the consistency of the evaluation measures and alternatives [17]. There for the ability to reduce bias in decision making.

There are three major phases within the AHP mechanism which are: problem decomposition, judgement of priorities and aggregation of priorities.

Phase 1: Problem decomposition

In the phase, a hierarchy structural representation of cloud services that shows the interrelationship among the overall goal, the attributes and the alternative services is shown. This layer tends to analyze the goals and how each cloud service attribute tends to satisfy the essential requirement of the user. This is represented in our case study of cloud service offerings by service providers and how they meet user requirements. Further explanation is shown in Section 5.1.

Phase 2: Judgement and priority phase

This phase deals with the assigning of weights to each attribute which is essential in comparing two cloud services in order to ascertain their relative importance. To address this issue we consider the user assigned weighting method which is a standard for using AHP. In this regard, we assign weights using pairwise comparison scale of [1]-[9] as shown in **Table 1** and recommended in the AHP method to judge the importance of one attribute over another. This allows the quantifying of both functional and non-functional cloud service preference of a certain attribute over another. By using the AHP method, the sum of all weights must be equal to 1 [17].

Phase 3: Aggregation of priority phase

The task of assigning weights is not as easy as it seems because some of the attributes are not quantifiable. To tackle this issue we consider the following formalism:

Let us assume A_q be the weight assigned by the user for the attribute q . let P_i and P_j be the values of attribute q for cloud services i and j . If b_i and b_j

Table 1. Pairwise comparison scale for AHP preference.

Intensity of importance	Definition	Explanation
1	Equal importance	Two elements contribute equally to the objective
3	Moderate importance	Somewhat more important/better
5	Strong importance	Definitely more important/better
7	Very strong important	Much more important/better
9	Extreme importance	Highest possible order of affirmation of one element over another

are the cloud services, then $\frac{b_i}{b_j}$ represents the relative comparison of b_i and b_j . Then we represent the value required by the user as t_i . In order to compare the values P_i and P_j for cloud services b_i and b_j we need to confirm that conventional unit for both values are the same. In an instance where we want to compare the cost of two advertised cloud service data storage, they must have the same currency notation (USD) as well as price per 1 GB for us to be able to have a perfect comparison. To contain the versatility of cloud service attribute especially the non-measurable characteristics of some attributes, a different nature of comparison is proposed for each type. Furthermore, for 2 types of cloud services b_i and b_j with numeric attributes we can compare them using two different criteria either higher is better therefore a higher intensity of importance is assigned or higher is lower therefore a lower intensity of importance is assigned to it. If we consider higher to be better then $\frac{P_i}{P_j}$ is the value of $\frac{b_i}{b_j}$ and if lower is better then $\frac{P_j}{P_i}$ is the value of $\frac{b_i}{b_j}$. In addition, we consider two attributes that maybe defined based on the number of platforms supported. To assign weights to cloud services for such attributes we declare size P_i and P_j to be the number of platforms supported by services i and j respectively. Let t_i be the size of the user requirement value for quantity of service attribute q . In such scenario the cloud service with the largest number of elements is declared better and higher weights are assigned to it. Furthermore, in a scenario where A_q be the weight assigned by the user for the attribute q . let P_i and P_j be the values of attribute q for cloud services i and j . If P_i and P_j values are equal, then there intensity of importance is equal to 1.

The above comparison matrices enables us obtain a one-on-one comparison of each cloud service for a particular attribute. This will further translate in a one-to-on metrix for a size $N \times N$ if there are a total of N services. The relative ranking of a particular attribute for all the cloud services is given by the eigenvector ($\hat{\lambda}$) of the matrix.

Finally, each attribute are aggregated with their relative weights assigned in

phase 2. The process of aggregation is repeated for all attributes in the service hierarchy.

3.2. Assigning of KPI Standards

The next stage after the aggregation for all attribute hierarchy is obtained is to determine a head to head comparison of all KPIs based on their level of importance to user requirement and adoption challenge. We attain an acceptable standard for each KPI based on the resultant priority value for each KPI after normalization.

3.3. Consistency Check

When pairwise comparison is performed, some inconsistencies may arise. In a situation where we consider 3 criteria and the first criterion is judged to be slightly more important than the second criterion while the second criterion is also judged slightly more important than the third. It is obvious that an inconsistency will arise if the third criterion is judged to be equal or more important than the first criterion. Also an inconsistency will arise if the first criterion is judged to be slightly more important than the third criterion. This is because from the above scenario a consistency evaluation should judge the first criterion more important than the third.

As mentioned above, one distinctive advantage of the AHP above other MCMD is its incorporation of an effective technique that aids in checking the consistency of the judgements made when building each of the pairwise comparison matrices involved in the process. This technique depends on the computation of suitable consistency index. To determine the consistency index (CI) the scalar λ is first computed as the average of the elements of the vector whose j th element is a ratio of the j th element of the vector Aw to a corresponding element of the vector w (where A is a matrix and w is a weight vector). Then:

$$CI = \frac{\lambda - M}{M - 1} \quad (1)$$

A perfect judgement is should always obtain $CI = 0$, but small inconsistency values are accepted when:

$$\frac{CI}{RI} < 0.1 \quad (2)$$

From the equation above, RI is the random index which is the consistency index when the entries of the matrix of A are completely random. The values of RI range from ($m \leq 10$) as shown in **Table 2**.

Table 2. Average random consistency (RI).

Size of matrix	1	2	3	4	5	6	7	8	9	10
Random consistency	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

3.4. Ranking Protocol

For our service ranking we deviate from the traditional summing up of weight used in the AHP method by proposing a protocol for ranking our cloud services as follows:

- Declare the acceptable standard for each KPI;
- obtain the acceptable KPI standards from the highest to lowest;
- The service ranking is done from 5 star to 1 star where five star are the services with the highest ranking and 1 star are services with the lowest rank.

For our service ranking we make the following declarations:

Let M -Be the cloud service;

p_1 -The weight value of the attribute with the highest KPI priority;

$p_1.x_1$ -The acceptable standard for the attribute p_1 ;

q_2 -The weight value of the attribute with the 2nd KPI priority weight;

$q_2.x_2$ -The acceptable standard for attribute q_2 ;

r_3 -The weight value of the attribute with the 3rd priority weight;

$r_3.x_3$ -The acceptable standard for attribute r_3 ;

k_4 -The weight value of the \geq attribute of any of the remaining KPI priority;

$k_4.x_4$ -The acceptable standard for any of the attribute of k_4 .

1) 5 star service ranking is characterized based on the following formation **when**

$$M \rightarrow (p_1 \geq p_1.x_1) \wedge (q_2 \geq q_2.x_2) \vee (r_3 \geq r_3.x_3) \wedge (k_4 \geq k_4.x_4) \quad (3)$$

2) 4 star service ranking is characterized by the following formation **when**

$$M \rightarrow (p_1 \leq p_1.x_1) \wedge (q_2 \geq q_2.x_2) \vee (r_3 \geq r_3.x_3) \wedge (k_4 \geq k_4.x_4) \quad (4)$$

3) 3 star service ranking is characterized by the following formation **when**

$$M \rightarrow (p_1 \leq p_1.x_1) \wedge (q_2 \leq q_2.x_2) \wedge (r_3 \geq r_3.x_3) \wedge (k_4 \geq k_4.x_4) \quad (5)$$

4) 2 star service ranking is characterized by the following formation **when**

$$M \rightarrow (p_1 \leq p_1.x_1) \wedge (q_2 \leq q_2.x_2) \wedge (r_3 \leq r_3.x_3) \wedge (k_4 \geq k_4.x_4) \quad (6)$$

5) 1 star ranking is characterized by the following formation **when**

$$M \rightarrow (p_1 \leq p_1.x_1) \wedge (q_2 \leq q_2.x_2) \wedge (r_3 \leq r_3.x_3) \wedge (k_4 \leq k_4.x_4) \quad (7)$$

We have described the steps taken in our approach. In the next section we will demonstrate our approach using a case study of advertised SaaS storage cloud services.

4. Case Study of Advertised Cloud Storage Services

The selection of SaaS storage product is based on experience and focus group sessions with SME owners and service information is gotten from advertised

cloud service offerings of four major cloud service providers. The four top SaaS cloud services used in this case study are OneDrive, Dropbox, Google Drive and iCloud. Although we use Service_A, Service_B, Service_C and Service_D randomly without using their real names.

In the following (Table 3), we compare each cloud service based on perceived adoption possibilities by SME owners, service provider's level of user consideration and flexibility of usage. The relative weight method helps us to determine the relative priority of each attribute for each SaaS cloud service in our scenario as advertised in service provider's website.

Proposed SaaS Storage KPIs Used in Our Case Study Are Described in This Section as Follows

- **Security**

Security is considered as one of the biggest fears among cloud service adoption challenges. This is because cloud computing represents a relatively new technological model and hence a lot of uncertainty on security issues at all levels. Possible security issues that may occur in a SaaS environment are application security which is caused due to the applications delivered over the internet. Therefore, some web application flaws may occur leading to SaaS applications being vulnerable. Other security issues that may arise are multi-Tenancy issue whereby users share the same database and this may lead to high risk of data leakage. In a bid to address this issue, most service providers have adopted different security measures. Dropbox uses AES-256 bit encryption to protect customer files at rest, Google Drive uses 128-bits AES and HTTP, OneDrive users

Table 3. Case study example.

Judgement criteria	Service_A (SaaS) (0.2620)	Service_B (SaaS) (0.4835)	Service_C (SaaS) (0.2429)	Service_D (SaaS) (0.1904)	Value type	Criteria weight
File size restriction	2 GB (0.0580)	- (0.5542)	10 GB (0.1462)	15 GB (0.2340)	Int	0.0721
Free storage	5 GB (0.2517)	2 GB (0.0967)	15 GB (0.5551)	2 GB (0.0967)	Int	0.0247
On-going payment plan	\$83.88/1TB/1Year \$2/month/100GB (0.5142)	\$99/1TB/1Year \$10/month/1TB (0.1899)	\$99/1TB/1Year \$2/month/100GB (0.1899)	\$99.99/1TB/Year \$20/month/1TB (0.108)	int	0.1640
Operating system supported	Windows, Mac, Android and iOS (0.1093)	Windows ,Mac, Linux, Android, iOS (0.5717)	Windows, Mac, Android and iOS (0.2090)	Windows, Mac, Android and iOS (0.1093)	String	0.1170
Trust access rights	0.125	0.625	0.125	0.125	string	0.2259
Security	0.1714	0.4708	0.0734	0.2839	string	0.3247
Office via web	0.5502	0.2519	0.1553	0.0423	string	0.0463
Bandwidth adjustment	0.069	0.5750	0.2867	0.069	string	0.2259

can encrypt on per file basis although they do not get at rest encryption on default, while iCloud uses a minimum of 128-bits-AES. Furthermore, the most insecure cloud storage is one that shares its login details with other products and services. Google Drive users are most likely to be vulnerable since all its offerings use the same login details, next is OneDrive and the iCloud. The users of Dropbox are least vulnerable due to Dropbox having very few services.

- **Accessibility and Trust**

Cloud service applications are easily accessed over the internet via a web browser and this makes accessibility via any computer device easy. This leads to easy access to information stealing through mobile malware, insecure market place as well as unauthorized hacking. Amongst the four compare cloud services, Dropbox is least likely to be accessed without authorization because compared to other SaaS services it has only on point access except in situations where a user decides to reuse the same password across different platforms. In addition, all four services are backed by SLAs that guarantee a 99% uptime per month, however there is still a little possibility of outage sometimes.

- **Platform Compatibility**

All four compared cloud services are compatible via multi platforms. Google Drive is compatible with windows, Mac, Android and iOS although it is not compatible with windows mobile. OneDrive which is a natural windows product is compatible with windows mobile as well as the Mac, Android and iOS. Dropbox is compatible windows mobile, Android, Mac, iOS and Linux while iCloud is compatible with windows, Mac, iOS and Android.

- **Free Storage**

The four SaaS cloud services compared offer free storage to its users although it is limited. The free storage allowance varies among the service providers. Google Drive offers the highest free storage allowance with a space of as much as 15 GB, OneDrive offers free allowance of 5 GB while Dropbox and iCloud offer 2 GB each.

- **Office via Web**

The ability for cloud services to access office applications is vital to some SMEs. OneDrive which is part of Microsoft products naturally links with office 365 and this gives its users the opportunity to edit their office documents directly within the web app. Likewise, users of Dropbox can also edit their office documents on office apps and save directly to Dropbox if using the office app online. Furthermore, Google Drive users can also edit, save and share documents via office apps.

- **Bandwidth Adjustment**

Among the four SaaS services compared, only Dropbox and Google Drive allows bandwidth adjustment. Dropbox and Google Drive allow users upload data from smartphones and from a web client. While Google Drive uses 100% of the bandwidth by default, Dropbox uses only 75% but they can both be increased or decreased depending on the user's choice. While Google controls the amount of

bandwidth that should be used while uploading or downloading on desktops and smartphones, Dropbox offers unlimited upload without any form of restrictions.

- **Pricing**

This is one of the most important factors when considering adoption of cloud services. The cost of cloud services varies among service providers at different subscription rate and price packages. OneDrive offers \$83.88/1TB/Year, Dropbox and Google Drive offers \$99/1TB/Year, while iCloud offers \$99.9/1TB/Year. However each of the service providers still provide monthly subscription in order to accommodate users who prefer to pay on monthly subscription. OneDrive \$2/month/100GB, Dropbox \$10/month/1TB, Google Drive \$2/month/100GB and \$10/month/1TB while iCloud offers \$20/month/1TB.

- **File size Restriction**

SaaS cloud services regulate the amount of files that can be shared at any given time. This restriction varies among the service providers. OneDrive allows for 2 GB, Google Drive allows 10 GB and iCloud allows 15 GB. Dropbox is the only cloud service that does not restrict the amount of file that can be shared.

Furthermore, using the information from **Table 3**, we show the procedural steps leading to ranking of cloud services. The relative weighting method as described in the above section is used to assign weights for each functional and non-functional property of the cloud services. A relative ranking matrix is constructed for each attribute we have shown the matrix of payment (for illustration) while we outline the results for the other attributes.

Based on the case study data, the relative service ranking matrix (RSRM) for payment (Pa) will be: $RSRM_{PA}$ =

	Service_B	Service_C	Service_A	Service_D
Service_B	0.1818	0.182	0.174	0.222
Service_C	0.1818	0.182	0.174	0.222
Service_A	0.5454	0.5454	0.522	0.444
Service_D	0.091	0.101	0.130	0.111
total	1	1	1	1

Computing the relative ranking vector for payment from the $RSRM_{PA}$ above is:

$$RSRV_{PA} = (0.1899 \ 0.1899 \ 0.5142 \ 0.108)$$

Similarly, we determine the Relative ranking vector for Operating System supported (OPS) as

$$RSRV_{OPS} = (0.5717 \ 0.2090 \ 0.1093 \ 0.1093)$$

Next we determine the relative ranking vector for file size restriction (FSR) as

$$RSRV_{FSR} = (0.5542 \ 0.1462 \ 0.058 \ 0.234)$$

Next we determine the relative ranking vector for free storage (FS) as:

$$RSRV_{FS} = (0.0967 \ 0.5551 \ 0.2517 \ 0.0967)$$

Next we determine the relative ranking vector for security (SE) as

$$\text{RSRV}_{\text{SE}} = (0.47085 \ 0.0732 \ 0.1714 \ 0.2839)$$

Next we determine the relative ranking vector for Trust (TS) as

$$\text{RSRV}_{\text{TS}} = (0.625 \ 0.125 \ 0.125 \ 0.125)$$

Next we determine the relative ranking vector for Bandwidth Adjustment (BA) as

$$\text{RSRV}_{\text{BA}} = (0.5750 \ 0.2867 \ 0.069 \ 0.069)$$

Next we determine the relative ranking vector for Office via Web (OVW) as

$$\text{RSRV}_{\text{OVW}} = (0.5542 \ 0.1462 \ 0.058 \ 0.234)$$

Furthermore, we determine the combined RSRV for all the KPI in the case study. The resultant RSRV for each attribute is set as an acceptable standard each attribute must attain for it to be acceptable and this is demonstrated within our ontology.

	PA	OPS	SE	FSR	BA	FS	OVW	TS
PA	0.1377	0.1828	0.1267	0.2249	0.1739	0.1666	0.1880	0.1121
OPS	0.0688	0.0914	0.0951	0.1687	0.1449	0.1428	0.1504	0.0746
SE	0.4132	0.3656	0.3805	0.2811	0.2318	0.2142	0.2633	0.4484
FSR	0.0344	0.0304	0.0633	0.0562	0.1159	0.1190	0.1128	0.0448
BA	0.0229	0.0182	0.0475	0.0140	0.0289	0.0476	0.0125	0.0324
FS	0.0196	0.0152	0.0761	0.0112	0.0144	0.0238	0.0094	0.0280
OVW	0.0275	0.0130	0.0543	0.0187	0.0869	0.0952	0.0376	0.0373
TS	0.2755	0.2742	0.1902	0.2249	0.2028	0.1904	0.2257	0.2242

$$\text{RSRV}_{\text{STANDARD}} = [\text{Pa}(0.1640), \text{Ops}(0.1170), \text{SE}(0.3247), \text{FSR}(0.0721), \\ \text{BA}(0.028), \text{Fs}(0.0247), \text{OVW}(0.0463), \text{TS}(0.2259)]$$

Finally, to determine our cloud service attribute combined weighting, we combine the RSRVs for all the attribute and the RSRV of the four service providers and then we multiply the result RSRV with the RSRV standard for each of the KPI.

$$\text{RSRV}_{\text{Overall weighting}} = \begin{pmatrix} 0.1899 & 0.5717 & 0.4708 & 0.5542 & 0.5750 & 0.0967 & 0.2519 & 0.625 \\ 0.1899 & 0.2090 & 0.0734 & 0.1462 & 0.2867 & 0.5551 & 0.1553 & 0.125 \\ 0.5142 & 0.1093 & 0.1714 & 0.0580 & 0.069 & 0.2517 & 0.5502 & 0.125 \\ 0.108 & 0.1093 & 0.2839 & 0.2340 & 0.069 & 0.0967 & 0.0423 & 0.125 \end{pmatrix} \begin{pmatrix} 0.1640 \\ 0.1170 \\ 0.3247 \\ 0.0721 \\ 0.0280 \\ 0.0247 \\ 0.0463 \\ 0.2259 \end{pmatrix}$$

Finally, we determine the SaaS cloud services attribute with overall highest weight based on the findings above as follows: Service B has the highest overall weight value of 0.4672 based on findings. The findings obtained in this section will be used to determine our service ranking protocol in Section 6.

We further depict a visual representation of our cloud service KPI's comparison using kivaith graphs in **Figure 2**.

5. Decision Support System

In the sections above we have shown the sequence of activities that takes place within the Framework in order to determine the superiority of one cloud service attribute over another as well as determine the acceptable benchmark of each cloud service attribute as shown in $RSRV_{STANDARD}$ above. To complete our



Figure 2. A visual cloud service comparison based on Cloudysme KPIs.

framework, we need to develop a decision support system which is equipped with an ontology of cloud services. This ontology holds all the information in our case study and also the findings of our comparison which is used to determine our SaaS storage cloud service ranking protocol within the DSS. The ontology which is built based on a set of concepts and associated semantic rules is used to retrieve information based on SME owner's requirements towards cloud service adoption. Furthermore, the ontology language helps us translate human language to machine readable language using terms and conditions. The design of the cloud service ontology which has already been developed and tested on protégé software to check consistencies is discussed in the next section.

5.1. Cloudysme Decision Support System Architecture

The decision support system architecture shows the sequence of activities that takes place within the DSS when a user requirement is queried. In order to complete this framework we have developed an ontology of SaaS storage cloud services. This ontology holds information about cloud services advertised by service providers and is used by our system algorithms to retrieve user requirements. The developed cloud service ontology has been tested on protégé software which is an ontology editor to check consistencies as explained below.

5.2. Characteristics of Each Component of Cloud Service Architecture

The CLOUDYSME DSS architecture is shown in **Figure 3**. The architecture consists of the following component: Graphical User Interface, query processor, Similarity reasoning, cloud service knowledge management (ontology) and service ranking. Firstly, The SME owner/manager sends their requirements to CLOUDYSME from the graphical user interface. The decision support system carries out the following functions depending user request: 1) Query processing; 2) Similarity reasoning; 3) Similarity matching; 4) Cloud service ranking.

5.2.1. Query Processing

When a user requirement is sent from the SME owner/manager via the graphical user interface, the query processor initiates query processing and converts the query to machine readable format. Then the processor sends the processed query to the similarity reasoning component for further processing based on the required information.

5.2.2. Similarity Reasoning

The processed query initiates the similarity reasoning process. This is done by consulting the cloud service ontology. Similarity decision is based on the type of information the query processed seeks to fetch. As mentioned above the main similarity reasoning types as shown in the CLOUDYSME system architecture are as follows:

- **Concept Similarity Reasoning:** This is based on the conceptual modelling of

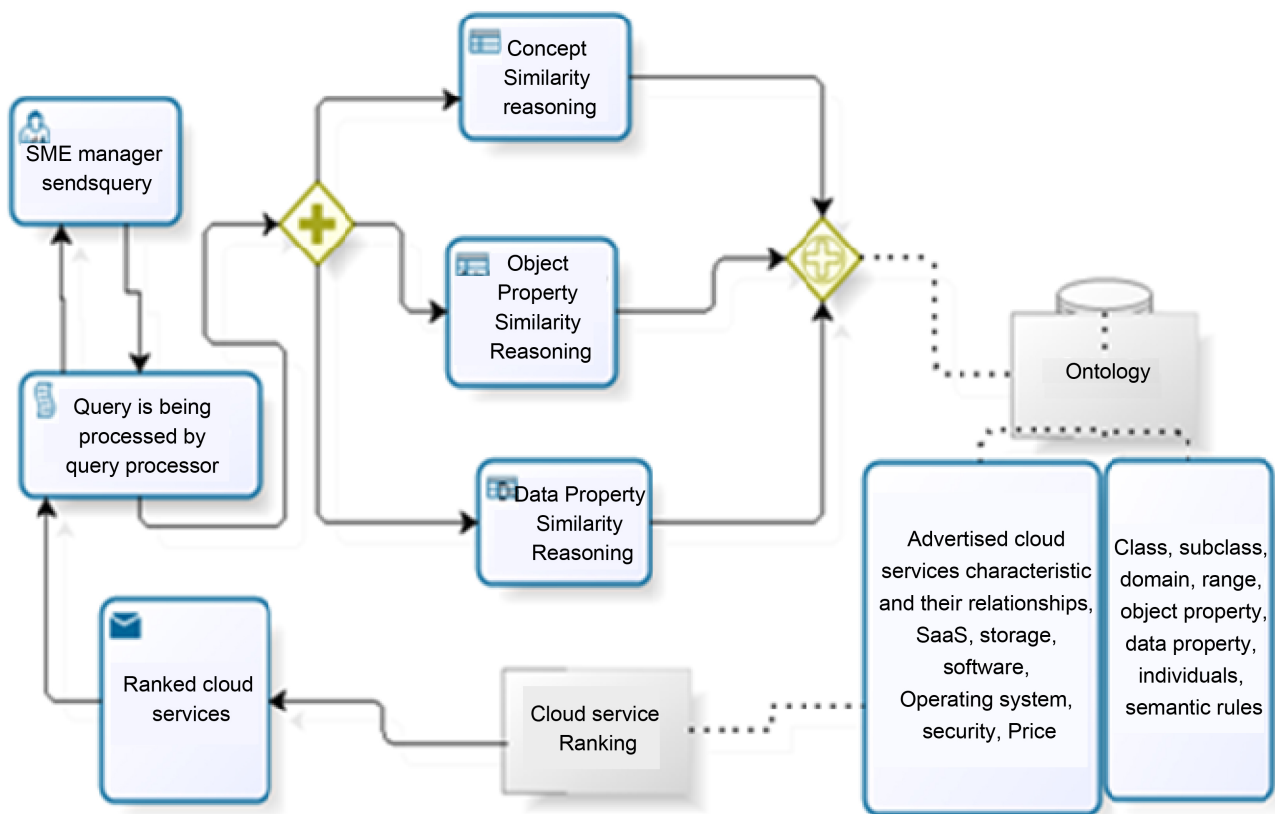


Figure 3. Showing cloudysme DSS architecture.

our ontology to meet user requirements. The presence of pellet reasoner within the ontology editor (protégé) aids our DSS to undergo conceptual reasoning by consulting the ontology to retrieve accurate information using system algorithms to meet user requirements. To show that our DSS equipped with a semantically designed ontology of cloud services can undergo conceptual similarity reasoning in an attempt to answer user requirements. In a scenario whereby an SME owner intends to adopt a cloud service for his data storage with a budget of 2 Dollars per month for 100 GBs of storage and consults CLOUDYSME for decision making. Based on the SME user requirement, the query sent by the user will trigger the system to use the ontology to do concept similarity reasoning. This is based on the conceptual modelling of advertised cloud services from service providers as shown in **Table 3**. The user requirement can be summarized as follows: The cloud service required by the user is a SaaS cloud service with a storage of 100 GB and a price value of 2 Dollars. The conceptual modelling is designed within the system following the RDF format of Subject, Predicate and Object statements with Subject and Object representing the domain and range of the predicate which helps us to translate the user requirement into machine language as follows: (DOMAIN: SaaS, Data Property: has Payment plan 1 price, Range: Integer). To get our user requirement, the following query is processed in machine readable format as (SaaS and has Payment plan 1 price value 2 and

has payment Plan 1 GB value of 100) as represented in **Figure 4**. The query is translated to lay terms as follows (Software as a service with a payment plan of 2 USD for 100 Gigabyte of data per month) Please note that the price value is in USD/Month and the GB (Gigabyte).

Based on the requirement by the SME user and the conceptual modelling of cloud services advertised by service providers as shown in our case study, the conceptual reasoning recommends that the cloud services that best meets the requirements in the above scenario are *Service C* and *Service A*. By adopting our system recommendations The SME owner can now narrow his decision on these two SaaS services. The SME owner may also decide to query other requirement such as Operating System, File size Restriction etc.

- **Object Property Similarity Reasoning:** This can be referred to as a condition where two or more cloud services have a common object property of the same instance. To demonstrate that our decision support system equipped with an ontology when queried can undergo object property similarity reasoning. We demonstrate this as follows: In a scenario whereby the SME owner above also needs a cloud service for his company which is compatible with android operating system. Based on the user requirement, the ontology will determine which cloud services meet the following RDF condition (Subject: (SaaS) Predicate (hasOS name3) Object (Android)) with the subject and object representing the domain and range respectively. For the ontology to understand the user requirement in machine readable language the following query is sent (SaaS and hasOS name3 value “Android”) as represented in **Figure 5**. The query can be translated in lay terms as follows (Software as a Service that is compatible with Android operating system). Please note that OS stands for operating system.

The screenshot shows a software interface for a DL query. At the top, there is a yellow header bar with the text "DL query:" and some icons. Below this, the interface is divided into two main sections: "Query (class expression)" and "Query results".

In the "Query (class expression)" section, there is a text input field containing the query: "SaaS and hasPaymentPlan1price value 2 and hasPaymentPlan1sizeGB value 100". Below the input field are two buttons: "Execute" and "Add to ontology".

In the "Query results" section, there is a table-like structure. On the left, there is a list of results. Under the heading "Sub classes (0)", there are no results. Under the heading "Instances (2)", there are two results: "ServiceC" and "ServiceA", each preceded by a diamond icon and followed by a question mark icon. On the right side of the "Query results" section, there is a list of checkboxes for filtering the results: "Super classes", "Ancestor classes", "Equivalent classes", "Subclasses" (which is checked), "Descendant classes", and "Individuals" (which is checked).

Figure 4. Example of conceptual similarity reasoning.

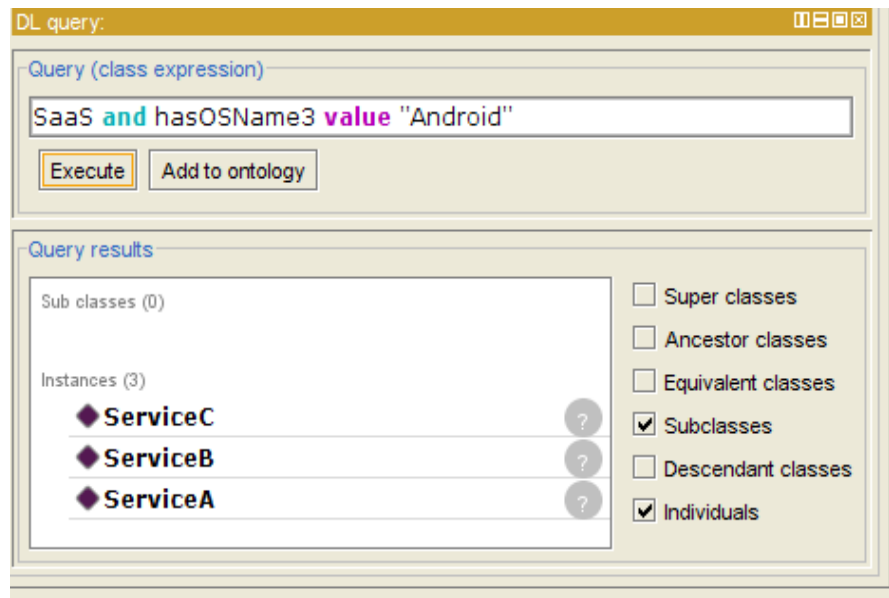


Figure 5. Example of object property similarity reasoning.

Based on the modelled ontology of advertised cloud services of service providers as shown in our case study, Our DSS recommends *Service C*, *Service B* and *Service A* are the SaaS services that are android compatible and recommended to the user by our system.

- Data Property Similarity Reasoning:** This is a type of reasoning that occurs within the ontology when a user queries the DSS Cloudysme for two or more cloud services with same datatype properties for a range of data. To demonstrate that our Decision System which includes a semantically designed ontology of cloud services can perform datatype similarity reasoning. In a scenario whereby the same SME owner in the two scenarios above also considers a SaaS cloud service with a file size restriction of data between 2 and 14 Gigabytes (GB). Before we show our DSS recommendation we will first show the RDF format of the above requirement within the ontology (Subject: (Cloud Service) Predicate: (has Filesize Restriction GB) Object: Int [$> X <$]). We query our ontology to meet user requirement by using machine readable language as The follows query is sent (SaaS and has File Size Restriction GB some int [$>1, <15$]) We translate the query in lay terms as follows: (Software as a service with a file size restriction between 1 and 14 gigabytes).

From the above datatype property reasoning, our system recommends *Service C* and *Service A* as the cloud services that meet the user requirement in **Figure 6** above.

At the stage we have proven that our system can undergo conceptual similarity reasoning, Object property similarity reasoning and Data property similarity reasoning in answering user requirements in view of aiding SME owners in cloud service adoption process. The next section will be demonstrating the use individual matching by our system to meet user requirements.

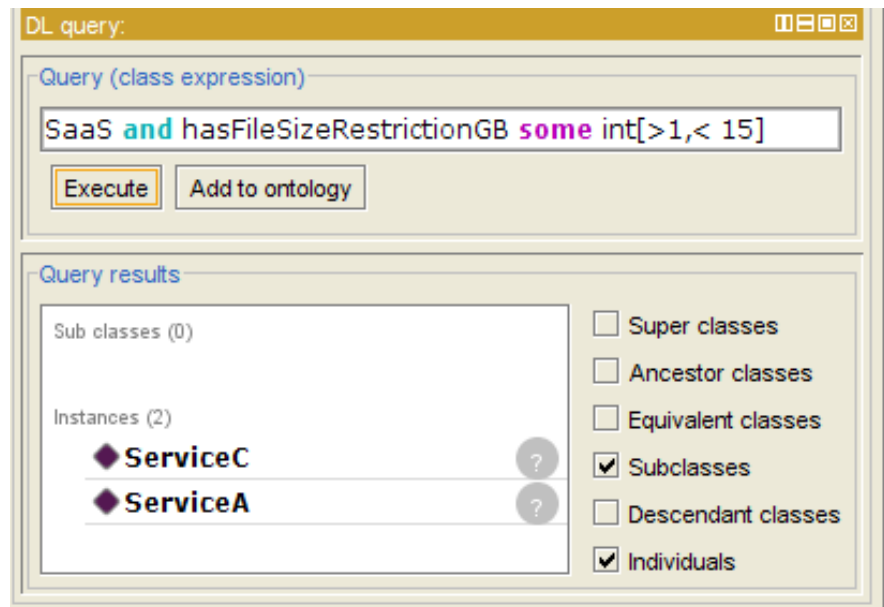


Figure 6. Data property similarity reasoning.

5.2.3. Individual Similarity Matching

This is a condition whereby cloud services are recommended based on the ability of each of their attributes to meet certain acceptable standards as shown in section 4 and this is achieved with the use of semantic rules as shown in Figure 7. From the above case scenarios and recommendations, our system has proven that it has enough knowledge to aid SME owners towards adopting cloud services for their businesses. In addition, our system has also shown that different cloud services meet different user requirements, furthermore, we have adopted the minimum acceptable standard for each criteria as represented above within our DSS. To prove that our system can undergo individual similarity matching, we demonstrate this by showing SaaS cloud services that meet the acceptable standard for File Size Restriction with an acceptable standard of $(0.0721 \geq 721)$ as seen in $RSRV_{STANDARD}$ above. To achieve this requirement, Semantic rules have been established within our ontology below. The following rule is being queried:

$(SaaS \text{ and } (hasFSRPriority \text{ some } int[\geq 721]))(?x) \rightarrow FSRacceptable (?x)$

From the above Individual similarity matching our system recommends, *Service D*, *Service C* and *Service B* as the cloud services that meet the acceptable standard for File Size Restriction. Our above system recommendation will enable the user narrow his choice to either of the recommended cloud services in his adoption decision making. The next section will be on cloud service ranking.

6. Cloud Service Ranking

Cloudysme service ranking is done using the 5 Stars, 4 Stars, 3 Stars, 2 Stars, 1 as discussed in Section 3.5. In this section we show the use of semantic rules (Figure 8) within the system in machine readable form. While Figure 9 shows a

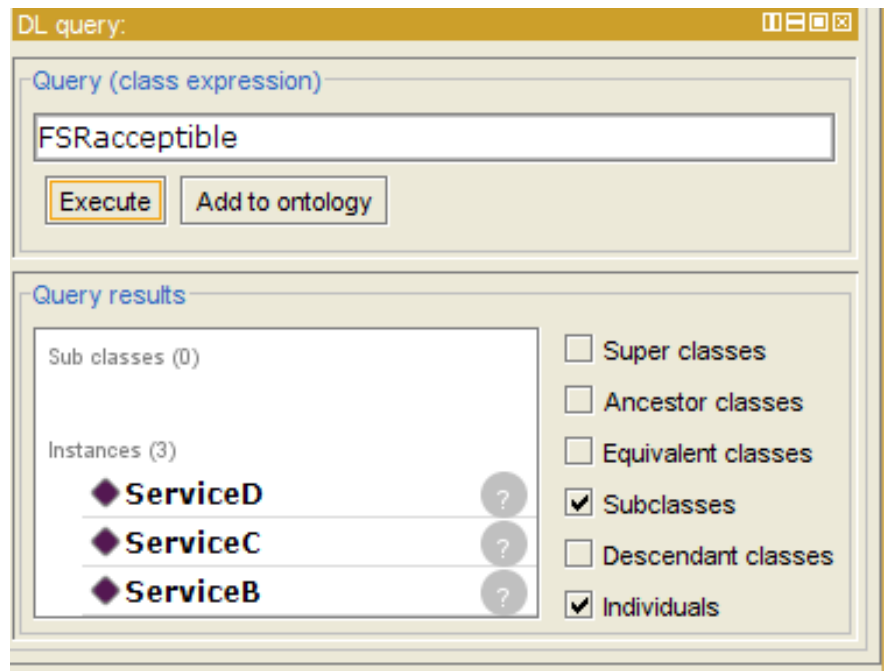


Figure 7. Individual similarity matching.

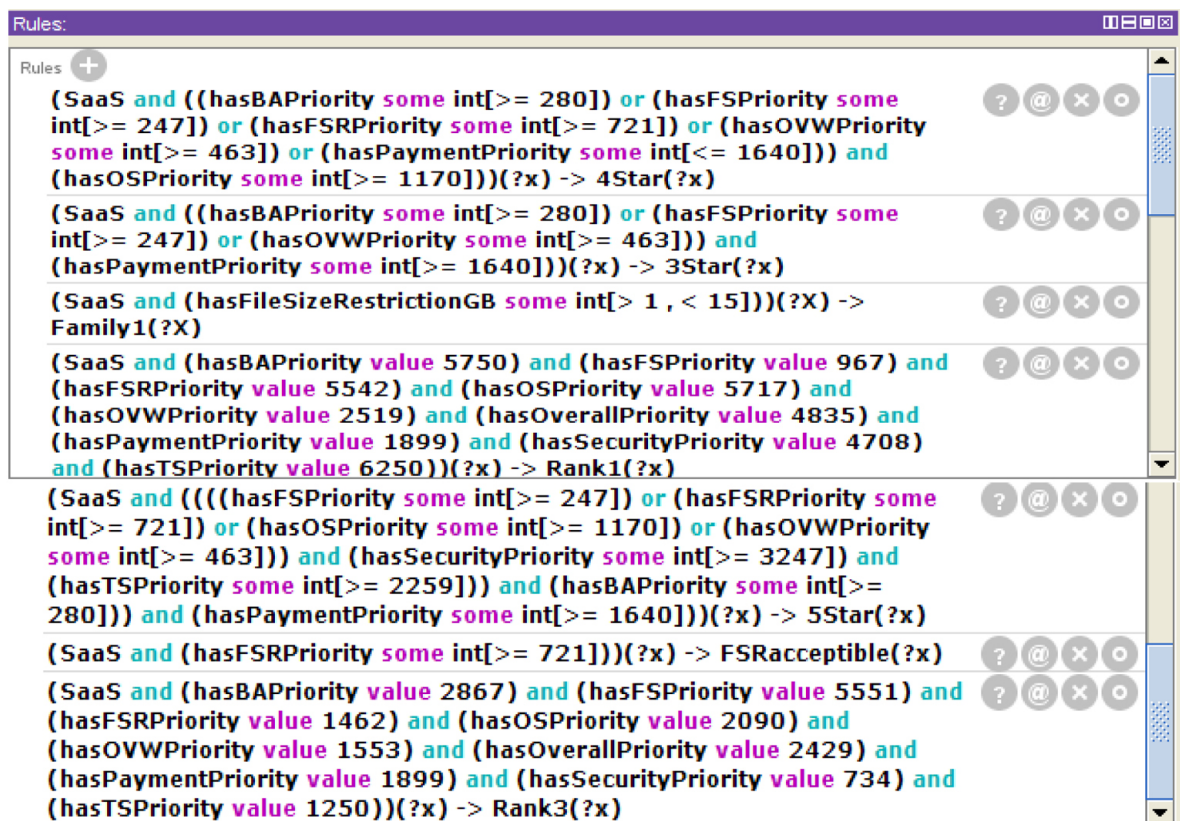


Figure 8. Showing rules within the system.

query execution listing Service B as the only SaaS storage service to meet the 5 star service ranking.

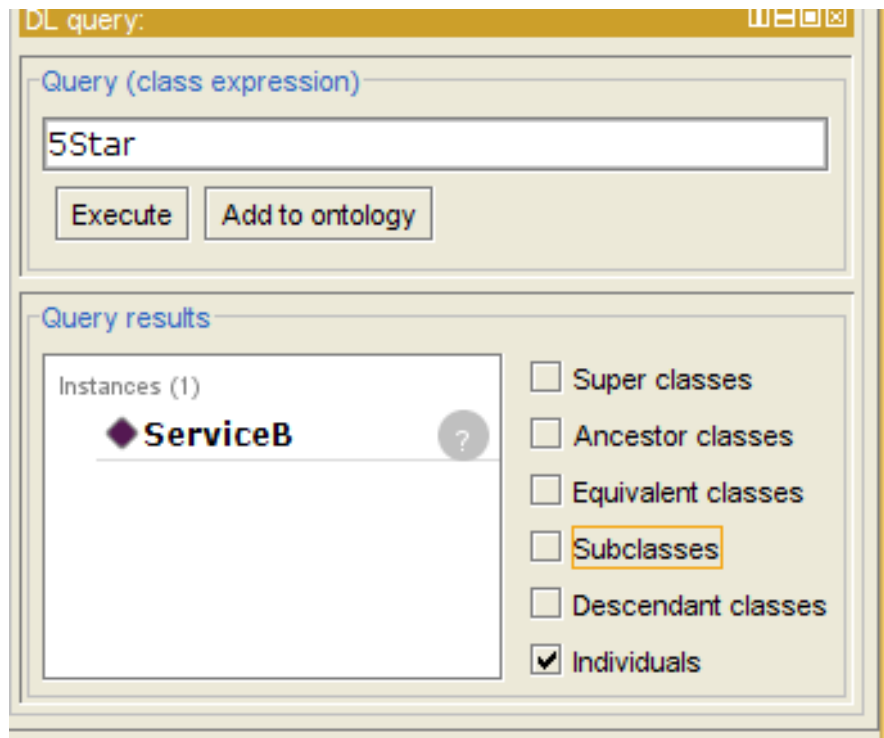


Figure 9. Showing Service B as the only SaaS to meets the 5 star ranking.

Below is a system query executed to show the service that meets the 5 star SaaS service ranking.

7. Implementation

In order to determine if our approach can aid SMEs in decision making towards adoption of SaaS cloud service for their business, we use 30 SMEs from various business sectors as pivotal test and we try to observe how many times our system recommended a particular cloud service for adoption based on the SME owner requirement. **Figure 10** shows a graphical representation of the number of times a particular cloud service was recommended.

Likewise we try to identify which of the attributes was mostly identified by the SME owner as a priority when adopting SaaS cloud service for their business **Figure 11** shows a graphical representation of our findings.

8. Related Works

In recent times, many researchers have proposed, designed, developed and implemented cloud frameworks and systems that allow users find suitable services that meet their requirement. While some have developed algorithms for resource management, others use ontology models to represent cloud services and to perform process matching between object and data properties of cloud service attributes in a bid to meet users' requirements [4] [18]. Ontologies can be defined as a "formal explicit specification of a shared conceptualization" [19] [20] [21]. They are useful for information retrieval to deal with user queries as they

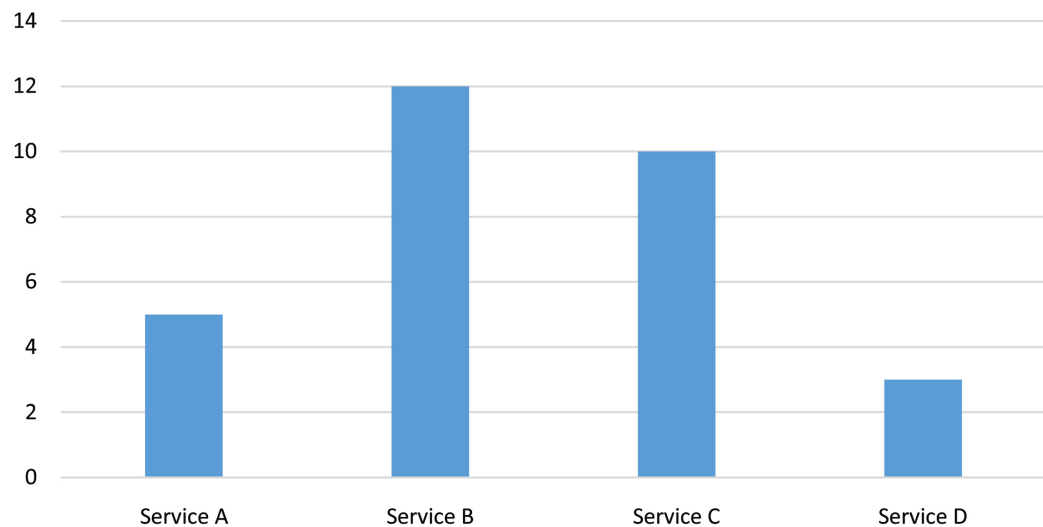


Figure 10. Showing number of times a service was recommended.

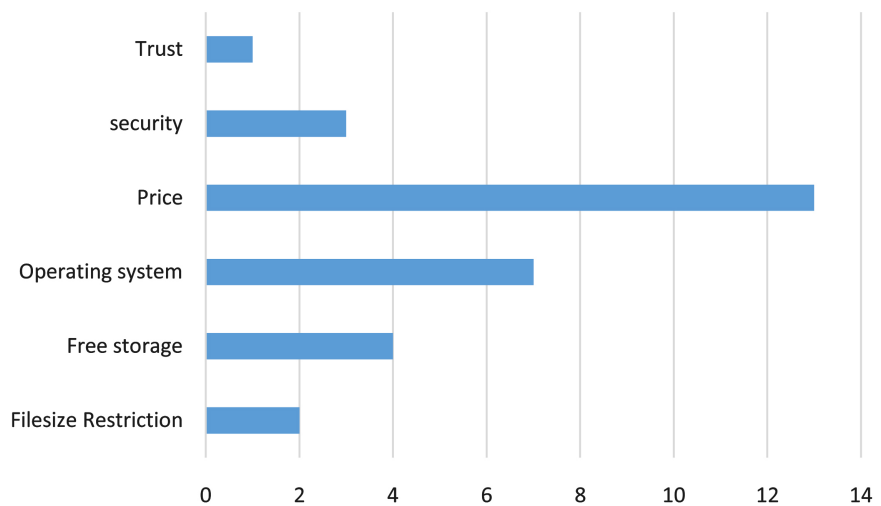


Figure 11. Showing SME owner priority towards cloud service adoption.

contain a set of concepts on the domain and the relationships between these concepts [22]. In addition, ontologies are known to have three important applications as follows: To enable the communication between software systems, to facilitate interoperability and to aid the communication among humans [23] [24]. Furthermore, the challenges associated with traditional search tools as well as matching between user requirement and advertised services by providers can be eliminated through ontologies and semantic technologies [25]. It is important in the area of information integration as seen in the work of [26], Knowledge management [3], Information retrieval and question answering [6] and Recommendation [27].

Presently, most cloud service ontologies are general with little or no detailed work on each cloud services [28]. Although the work of [5] presents a unified view of cloud computing representing its components and their relationships.

However, this cloud service ontology has a service orientation showing a clear distinction between different layers such as cloud application layer (SaaS), software environment (PaaS) and cloud Infrastructure (IaaS). Due to the rise in number of service providers rendering similar services, there has been a continuous research towards distinctively understanding the different layers of these services, their attributes, relationships, functional and non-functional properties. The work of [29] presents a tabular representation of survey findings of cloud services, thereby comparing the offerings of services provided by major service providers in each service layer IaaS (Amazon web service, GoGrid, Flexiscale, Mosso) PaaS and SaaS (GoogleApp Engine, Azure, force.com, GigaSpaces). Also [30] propose a taxonomy of comparison of cloud services providing detailed characteristics in a hierarchical form using common terminologies associated with each layer as a baseline for information and communication. The authors [31] classified cloud services based on pricing of complex services as well as security and reliability. While a framework for ranking cloud services by evaluating cloud offerings and ranking them based on their ability to meet users quality of service requirements is proposed by [1]. Our work compliments these previous works in the area of cloud service information gathering, classification and utilization of service ranking tools toward meeting user requirements.

An ontology enhanced cloud service discovery system is proposed by [32] the system enables users select cloud services providers based on the provided ontology. The work of [7] proposed an ontology based discovery of cloud providers with a range of querying possibilities on different cloud service layers. Cloud service provider resource management ontology is proposed by [8] while the work of [2] proposed an ontology that relates to service lifecycle and cloud governance. The work of Han *et al.* (2009) focuses on the ranking of available service providers using a statistical approach. The Authors, [4] claimed that their ontology is much better in the aspect of querying possibilities and more comprehensive in respect to other works where the three cloud service layers SaaS, PaaS, IaaS have been considered as it can be used in discovery of cloud services as well as resource management in more complex and comprehensive manner.

This study is complementary for existing cloud computing works as it proposes an ontology that can be used for cloud service discovery, knowledge management and service ranking. Our work goes a step further by comparing different advertised cloud services and their attributes using a multiple criteria decision making method by proposing an extended version of AHP towards cloud service adoption decision and the use of semantic rules within the ontology to aid cloud service information retrieval and decision making process to meet user requirement.

9. Discussion

One important decision in our approach is to equip our decision support system which is included in our framework with an ontology. This is because semantic

web supports descriptive logical reasoning using software's like pellet. In addition, it allows the use of semantic rules to give consistent and accurate feedback in a view to recommend solutions that meets user requirements, it allows the use of ontology language to translate human language to machine readable language. This makes it easy for SME owners to get real time solution and knowledge of cloud services rather than surfing the net for information or overlooking cloud services for their business due to lack of knowledge. The slow adoption rate associated with cloud service adoption by SMEs can be attributed to the number of service providers offering similar cloud service packages at different prices, lack of knowledge, Security issues, trust and compatibility as discussed in the above section.

Our proposed framework which tends to address the issue of slow adoption rate is divided into four major phases. The first phase comprises of advertised cloud services by services providers and this is represented in our case study. The attribute for each cloud service represented in our framework was obtained from the cloud service provider website. The second phase is the Filtering phase which an extended version of Analytical Hierarchical Process (AHP) which is multi criteria decision making method to determine the weight of each cloud service attribute, assign acceptable standard for each attribute as well as propose a protocol for cloud service ranking. To achieve this, we compare two similar cloud services attribute from different service providers head to head using pairwise comparison. This is followed by prioritizing the attributes based on superiority of one over another. The degree of consistency of the pairwise comparison is being measured by determining the consistency ratio (CR) of each comparison. In addition, we determine the acceptable benchmark for each cloud service criteria by performing a head to head comparison of all the criterion based on their level of importance to meet user requirement and we depict visual representation of our findings using a Kaviat graph. The third phase of our framework is composed of a Decision support system which holds a knowledge management of cloud services. The information obtained in the first and second phase of our framework are represented in this phase. Furthermore, in this phase, intelligent system reasoning takes place in a bid to meet user requirements based a set of semantic rules. The cloud service knowledge management is presented in our ontology using protégé and this gives the DSS the ability to fetch accurate and timely information by matching user requirement with the required information, to make accurate and timely recommendation towards cloud service adoption. The fourth phase of our framework is the cloud service ranking stage. This is achieved based on a set of rules within the system ontology and these rules are set according to how the cloud services meet the acceptable bench mark assigned to each judgement criteria. We adopt a set of protocols which must be met for each cloud service to be ranked either 5 Star, 4 Star, 3 Star, 2 Star or 1 Star. Finally, we use a case study scenarios to prove that the use of a semantic web ontology can aid in the decision making process of cloud ser-

vice adoption by SMEs when equipped with a knowledge of cloud services. Also, from our cloud service ranking, we conclude that Service B (not the real name) is the only cloud service to attain the 5 star ranking. The four major phases discussed above make up our framework in our bid to contribute to solving the slow adoption rate of cloud services by SMEs. Finally, from our implementation findings using 30 SMEs from various sectors, Service B was the most recommended service by our system based on user requirement while Price was the attribute the highest attribute considered by participating SMEs towards SaaS storage adoption.

10. Conclusions and Future Work

Cloud computing is seen as a technology that provides services over the internet just like public utilities. Many cloud service providers present cloud services in their own format as their no standardization for representing cloud services. Similarly, the presence of dominant players within the IT sector providing this technology shows that cloud computing is important in our world today. This has made it challenging for possible adopters especially in the SME sector to make a decision on which cloud service will best suit their company business process. Therefore, the need for a framework that will aid in cloud service adoption decision in terms of how service providers meet user requirement is vital.

The benefits of the proposed cloud service adoption framework are: The multi-tasking ability of our middleware tackles the issue of specific tasking associated with previous proposed systems, Our proposed ranking method can be adopted by other researchers in other fields, The proposed framework can also be used to compare other cloud service layers, it also has the ability to aid SME owners in cloud service adoption decision thereby tackling the issue of adoption complexity faced by prospective adopter. Furthermore, from our service ranking Service B (not real name) is the only cloud service to attain the 5 Star ranking which is the highest ranking within our framework. In addition, from our pivot studies, service was the most recommended service based on user requirement while price was the highest attribute required by users.

In future work, we intend to extend our work to the quantifiable quality of cloud services (IaaS & PaaS) as well as develop a system that can be used to aggregate the QOS configuration between cloud service layers in different applications.

References

- [1] Garg, S.K., Versteeg, S. and Buyya, R. (2013) A Framework for Ranking of Cloud Computing Services. *Future Generation Computer Systems*, **29**, 1012-1023. <https://doi.org/10.1016/j.future.2012.06.006>
- [2] Fortis, T.-F., Munteanu, V.I. and Negru, V. (2012) Towards an Ontology for Cloud Services. *6th International Conference on Complex, Intelligent and Software Intensive Systems*, Palermo, 4-6 July 2012, 787-792.
- [3] Colomo-Palacios, R., García-Crespo, Á., Soto-Acosta, P., Ruano-Mayoral, M. and

- Jiménez-López, D. (2010) A Case Analysis of Semantic Technologies for R&D Intermediation Information Management. *International Journal of Information Management*, **30**, 465-469. <https://doi.org/10.1016/j.ijinfomgt.2010.05.012>
- [4] Ali, A., Shamsuddin, S.M. and Eassa, F.E. (2014) Ontology-Based Cloud Services Representation. *Research Journal of Applied Sciences, Engineering and Technology*, **8**, 83-94. <https://doi.org/10.19026/rjaset.8.944>
- [5] Youseff, L., Butrico, M. and Da Silva, D. (2008) Toward a Unified Ontology of Cloud Computing. *Grid Computing Environments Workshop*, Austin, 16 November 2008, 1-10.
- [6] Valencia-García, R., Fernández-Breis, J.T., Ruiz-Martínez, J.M., García-Sánchez, F. and Martínez-Béjar, R. (2008) A Knowledge Acquisition Methodology to Ontology Construction for Information Retrieval from Medical Documents. *Expert Systems*, **25**, 314-334. <https://doi.org/10.1111/j.1468-0394.2008.00464.x>
- [7] Tahamtan, A., Beheshti, S.A., Anjomshoaa, A. and Tjoa, A.M. (2012) A Cloud Repository and Discovery Framework Based on a Unified Business and Cloud Service Ontology. *IEEE 8th World Congress on Services*, 16 July 2012, 203-210. <https://doi.org/10.1109/SERVICES.2012.42>
- [8] Ma, Y.B., Jang, S.H. and Lee, J.S. (2011) Ontology-Based Resource Management for Cloud Computing. *Asian Conference on Intelligent Information and Database Systems*, Daegu, 20-22 April 2011, 343-352. https://doi.org/10.1007/978-3-642-20042-7_35
- [9] Zeleny, M. and Cochrane, J.L. (1973) Multiple Criteria Decision Making. University of South Carolina Press, Columbia.
- [10] Saaty, T.L. (2008) Decision Making with the Analytic Hierarchy Process. *International Journal of Services Sciences*, **1**, 83-98. <https://doi.org/10.1504/IJSSCI.2008.017590>
- [11] Belton, V. (1986) A Comparison of the Analytic Hierarchy Process and a Simple Multi-Attribute Value Function. *European Journal of Operational Research*, **26**, 7-21. [https://doi.org/10.1016/0377-2217\(86\)90155-4](https://doi.org/10.1016/0377-2217(86)90155-4)
- [12] Belton, V. and Gear, T. (1983) On a Short-Coming of Saaty's Method of Analytic Hierarchies. *Omega*, **11**, 228-230. [https://doi.org/10.1016/0305-0483\(83\)90047-6](https://doi.org/10.1016/0305-0483(83)90047-6)
- [13] Dyer, J.S. (1990) Remarks on the Analytic Hierarchy Process. *Management Science*, **36**, 249-258. <https://doi.org/10.1287/mnsc.36.3.249>
- [14] Dyer, J.S. and Wendell, R.E. (1985) A Critique of the Analytic Hierarchy Process. Department of Management, College of Business Administration and Graduate School of Business, University of Texas, Austin.
- [15] Pérez, J. (1995) Some Comments on Saaty's AHP. *Management Science*, **41**, 1091-1095. <https://doi.org/10.1287/mnsc.41.6.1091>
- [16] Hacker, P. and Vargas, L. (1987) The Theory of Ratio Scale Estimation: Saaty's Algorithms, and How It Is.
- [17] Saaty, T.L. (2005) Theory and Applications of the Analytic Network Process: Decision Making with Benefits, Opportunities, Costs, and Risks. RWS Publications.
- [18] Chang, Y.-S., Juang, T.-Y., Chang, C.-H. and Yen, J.-S. (2012) Integrating Intelligent Agent and Ontology for Services Discovery on Cloud Environment. *IEEE International Conference on Systems, Man, and Cybernetics*, Seoul, 14-17 October 2012, 3215-3220. <https://doi.org/10.1109/ICSMC.2012.6378286>
- [19] Gruber, T.R. (1993) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, **5**, 199-220. <https://doi.org/10.1006/knac.1993.1008>

- [20] Studer, R., Benjamins, V.R. and Fensel, D. (1998) Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, **25**, 161-197.
[https://doi.org/10.1016/S0169-023X\(97\)00056-6](https://doi.org/10.1016/S0169-023X(97)00056-6)
- [21] Kang, J. and Sim, K.M. (2011) Towards Agents and Ontology for Cloud Service Discovery. *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Beijing, 10-12 October 2011, 483-490.
<https://doi.org/10.1109/CyberC.2011.84>
- [22] Reshma, V. and Balaji, B.S. (2012) Cloud Service Publication and Discovery using Ontology. *International Journal of Scientific & Engineering Research*, **3**, 1-5.
- [23] Jasper, R. and Uschold, M. (1999) A Framework for Understanding and Classifying Ontology Applications. *Proceedings 12th International Workshop on Knowledge Acquisition, Modelling, and Management KAW*, Banff, 16-22 October 1999, 16-21.
- [24] Maedche, A. (2012) Ontology Learning for the Semantic Web 665. Springer Science & Business Media, Berlin.
- [25] Lupiani-Ruiz, E., García-Manotas, I., Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J.T., et al. (2011) Financial News Semantic Search Engine. *Expert Systems with Applications*, **38**, 15565-15572.
- [26] Wang, C., Lu, J. and Zhang, G. (2006) Integration of Ontology Data through Learning Instance Matching. *IEEE/WIC/ACM International Conference on Web Intelligence*, Washington DC, 18-22 December 2006, 536-539.
- [27] García-Crespo, Á., López-Cuadrado, J.L., Colomo-Palacios, R., González-Carrasco, I. and Ruiz-Mezcua, B. (2011) Sem-Fit: A Semantic Based Expert System to Provide Recommendations in the Tourism Domain. *Expert Systems with Applications*, **38**, 13310-13319. <https://doi.org/10.1016/j.eswa.2011.04.152>
- [28] Androcec, D., Vreck, N. and Seva, J. (2012) Cloud Computing Ontologies: A Systematic Review. *Proceedings of the 3rd International Conference on Models and Ontology-Based Design of Protocols, Architectures and Services*, Mont Blanc, 29 April-4 May 2012, 9-14.
- [29] Rimal, B.P., Choi, E. and Lumb, I. (2009) A Taxonomy and Survey of Cloud Computing Systems. *INC, IMS and IDC*, Seoul, 25-27 August 2009, 44-51.
<https://doi.org/10.1109/NCM.2009.218>
- [30] Höfer, C. and Karagiannis, G. (2011) Cloud Computing Services: Taxonomy and Comparison. *Journal of Internet Services and Applications*, **2**, 81-94.
<https://doi.org/10.1007/s13174-011-0027-x>
- [31] Weinhardt, C., Anandasivam, D.-I.-W.A., Blau, B., Borissov, D.-I.N., Meinel, D.-M.T., Michalk, W., et al. (2009) Cloud Computing—A Classification, Business Models, and Research Directions. *Business & Information Systems Engineering*, **1**, 391-399. <https://doi.org/10.1007/s12599-009-0071-2>
- [32] Han, T. and Sim, K.M. (2010) An Ontology-Enhanced Cloud Service Discovery System. *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, Hong Kong, 17-19 March 2010, 17-19.