ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 8, No. 6, 37-45 (2023)

www.astesj.com

Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

Infrastructure-as-a-Service Ontology for Consumer-Centric Assessment

Thepparit Banditwattanawong[1], Masawee Masdisornchote[*,2]

[1]*Department of Computer Science, Kasetsart University, Krung Thep Maha Nakhon, 10900, Thailand*

[2]*School of Information Technology, Sripatum University, Krung Thep Maha Nakhon, 10900, Thailand*

ARTICLE INFO

Article history:
Received:07 September, 2023
Accepted:09 November, 2023
Online: 30 November, 2023

Keywords:
Cloud computing
Infrastructure-as-a-Service (IaaS)
Knowledge engineering
Ontology
Taxonomy

ABSTRACT

In the context of adopting cloud Infrastructure-as-a-Service (IaaS), prospective consumers need to consider a wide array of both business and technical factors associated with the service. The development of an intelligent tool to aid in the assessment of IaaS offerings is highly desirable. However, the creation of such a tool requires a robust foundation of domain knowledge. Thus, the focus of this paper is to introduce an ontology specifically designed to characterize IaaSs from the consumer's perspective, enabling informed decision-making. The ontology additionally serves two purposes of other relevant parties besides the consumers. Firstly, it empowers IaaS providers to better tailor their services to align with consumer expectations, thereby enhancing their competitiveness. Additionally, IaaS partners can play a pivotal role in supporting both consumers and providers by understanding the protocol outlined in the ontology that governs interactions between the two parties. By applying principles of ontological engineering, this study meticulously examined the various topics related to IaaS as delineated in existing cloud taxonomies. These topics were subsequently transformed into a standardized representation and seamlessly integrated through a binary integration approach. This process resulted in the creation of a comprehensive and cohesive ontology that maintains semantic consistency. Leveraging Protégé, this study successfully constructed the resultant ontology, comprising a total of 340 distinct classes. The study evaluated the syntactic, semantic, and practical aspects of the ontology against a worldwide prominent IaaS. The results showed that the proposed ontology was syntactically and semantically consistent. Furthermore, the ontology successfully enabled not only the assessment of a real leading IaaS but also queries to support automation tool development.

1. Introduction

In the process of cloud adoption, there are three parties involved [1]. First, cloud service customers need to assess potential services to determine the best fit for their business and technical requirements. Presently there are numerous Infrastructure as a Service (IaaS) options available across multiple providers. This makes it challenging for potential IaaS customers to assess the different options and make optimal decisions about cloud adoption and/or migration. Second, cloud service providers strive to deliver cloud services that fully satisfy customers' expectations. However, these expectations remain partially unknown due to the lack of comprehensive checklists. This hinders the readiness improvement of offered IaaSs. Finally, cloud service partners facilitate the activities of both customers and providers based on mutually agreed-upon protocols between both IaaS customers and providers. Unfortunately, such protocols are currently lacking.

One promising solution to address these problems is to enhance the comprehension of customers' concerns and service judgment criteria via the introduction of an IaaS knowledge base that is understandable by all parties. As such, the contributions of this paper are twofold.

- Firstly, this paper proposes a semantic model, an ontology, that facilitates the decision making of IaaS adoption. The ontology is constructed from a consumer perspective and potentially serves as a foundational knowledge base for the development of various IaaS assessment tools, such as score-based comparing systems, recommendation systems, and expert systems.

[*] Corresponding Author: Masawee Masdisornchote, masawee.ma@spu.ac.th

https://dx.doi.org/10.25046/aj080605

- Secondly, this paper demonstrates an innovative approach to constructing this ontology by leveraging existing IaaS taxonomies. This approach not only extends benefits to researchers but also to industry practitioners. For instance, it can serve as a launchpad for further research and development concerning Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) ontologies, thus fostering advancement in the field as a whole.

The paper is organized as follows. Next section reviews related ontologies and their limitations in addressing our research problem. Section 3 describes the systematic formulation of a proposed ontology using the ontology engineering principle. Section 4 evaluates the practicality of the ontology with real-world IaaSs. The paper concludes in Section 5, summarizing our key findings.

2. Related Work

Aiming to facilitate cloud customers as our work, the author in [2] recently proposed a KPI-based framework for evaluating and ranking IaaSs, PaaSs, and SaaSs to help cloud users identify cloud services that best suit their needs. However, the framework consists of only 41 KPIs, lacking several practical details for effective decision making. The framework has not been evaluated using real cloud service data. In [3], the authors proposed a recommender system using quality of cloud services for selecting cloud services that satisfied end user requirements. However, the quality of service (QoS) attributes were rather limited: response time, availability, throughput, dependability, reliability, price, and reputation. The authors in [4] proposed a signature-based QoS performance discovery algorithm to select IaaS. The algorithm leveraged the combination of service trial experiences and IaaS signatures. Each signature represented a provider's long-term performance behavior for a service over a fixed period. Similarities between users' service trial experiences and IaaS signatures were measured to select proper IaaSs. Nevertheless, the algorithm merely focused on a performance aspect. In [5], the authors proposed an IaaS selection algorithm based on utility functions and deployment knowledge base, which stored application execution histories. Unfortunately, the authors provided no details about the knowledge base's abstraction structure to be evaluated for its IaaS aspect coverage.

Cloud computing ontologies that encompass IaaS concepts, which particularly represented decision-making factors in cloud service adoption, are limited as follows. As many clouds vendors and standards employed inconsistent terminology to define their services, a common ontology in [6] was proposed to provide an approach to discover and use services in cloud federations. The ontology enables applications to negotiate cloud services as requested by users. Although the ontology included IaaS related terms such as some resources and specific services, it fell short of encompassing the complete spectrum of IaaS-related aspects essential for a comprehensive customer-oriented IaaS ontology. An ontology in [7] was proposed to implement intelligent service discovery and management systems for searching and retrieving appropriate cloud services in an accurate and quick manner. The ontology consisted of cloud computing concepts in general including some IaaS related ones such as compute, network, and storage services. Nevertheless, it remained incomplete as an IaaS ontology in its entirety. The authors in [8] proposed an ontology to define functional and non-functional concepts, attributes and relations of infrastructure services and used it to implement a cloud recommendation system. The ontology was composed of detailed services, compute, network, and storage, totally 26 classes. So, the ontology left several additional IaaS facets unaddressed. Several years later, the authors in [9] improved the ontology in [8] by adding 11 classes of price and quality-of-service concepts. The ontology was evaluated by being deployed in the development of semantic data sets sourced from Azure and Google Cloud services.

In summary, the IaaS concepts in aforementioned ontologies are only parts of our proposed ontology, which is more comprehensive by incorporating numerous customer-perspective IaaS taxonomies. Our preliminary work [10] is significantly extended in this paper to incorporate more recent taxonomies as follows. The authors in [11] extended [8] to derive the taxonomy of interoperability in IaaS cloud. The taxonomy's main topics were access mechanism, virtual appliance, network, and service level agreement (SLA). In [12], the author proposed the taxonomy of IaaS services where its main topics were Hardware as a Service (HaaS), which provisions hardware resources, and Infrastructure Services as a Service (ISaaS), which provided a set of auxiliary services to enable successful HaaS provision. The authors in [13] proposed a cloud computing services taxonomy including main topics: main service category, license type, intended user group, payment system, formal agreement, security measures, and standardization efforts. In [14], the authors proposed the taxonomy of fundamental IaaS components consisting of main topics: support layer, management layer, security layer, and control layer. In [15], the authors proposed a comprehensive taxonomy of cloud pricing consisting of three pricing strategies (i.e., value-based pricing, market-based pricing, and cost-based pricing) and nine pricing categories (e.g., retail-based pricing and utility-based pricing). The authors in [16] proposed a SLA taxonomy for PaaS and SaaS besides IaaS.

Furthermore, this paper resolves all conflicts and redundancy among the taxonomies and merges them by using a binary approach, which is more natural but less automated than an identified tabular list previously introduced in [17]. In addition, this paper transforms the integrated taxonomies into a unified ontology in order to allow wider applicability.

3. Ontology Formulation

This section presents the formulation of our proposed ontology based on the principle of ontology engineering [18]. The formulation process entails four steps: taxonomy reuse, refinement, formalization, and evaluation.

3.1. Taxonomy Reuse

A taxonomy serves as a structured arrangement of pertinent topics and subtopics designed for classification purposes. This study conducted a thorough examination of existing IaaS-related taxonomies. The period from 2009 to 2011 marked a significant juncture for cloud computing when Gartner Inc. positioned it at the zenith of the emerging technology hype cycle, garnering worldwide attention. Consequently, our focus was directed exclusively towards taxonomies created after 2009, as they began to gain widespread conceptual clarity and recognition during this

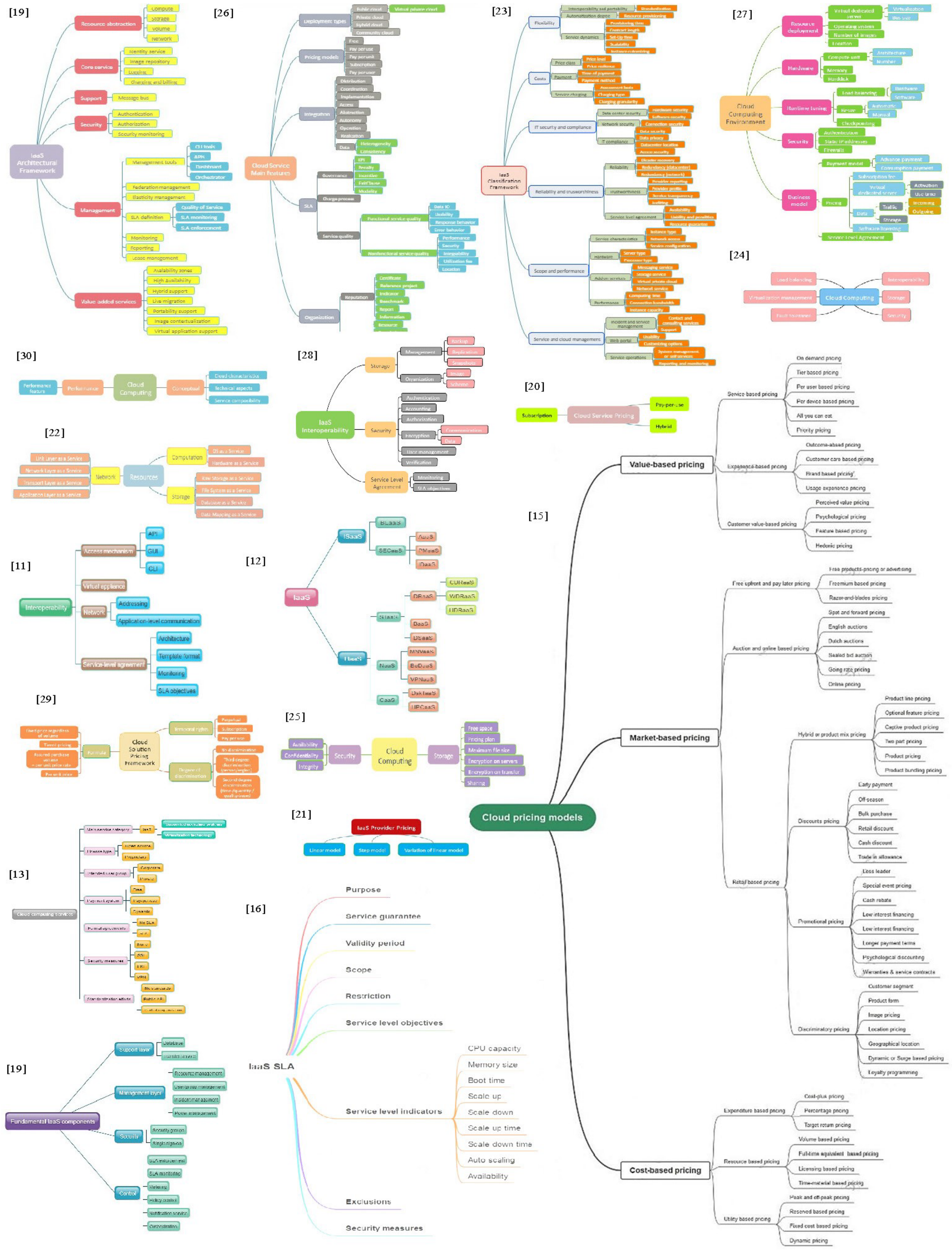


Figure 1: Tenant-centric IaaS excerpted and transformed taxonomies

period. While these taxonomies encompassed concepts found in all the ontologies examined in the preceding section, they were not primarily geared towards IaaS. Therefore, a careful analysis was required to identify IaaS-specific topics that are directly relevant to customers. Table 1 provides an overview of the taxonomies employed in this paper, along with the proportion of tenant-centric IaaS topics extracted from each taxonomy.

Table 1: Customer-centric topic excerption from IaaS taxonomies

Taxonomy	Excerption	Taxonomy	Excerption
[19]	100%	[24]	67%
[20]	100%	[25]	65%
[21]	100%	[26]	61%
[15]	100%	[27]	60%
[22]	98%	[11]	54%
[12]	96%	[28]	49%
[23]	95%	[29]	41%
[19]	75%	[16]	41%
[13]	73%	[30]	33%

Reusing these taxonomies in their original formats presents a challenge, given their diverse presentation styles, which encompass textual descriptions and various graphical models such as mind maps, feature models, decision trees, layered block diagrams, SBIFT models, and textual lists. To create a cohesive and consistent representation while eliminating conflicts and redundancies, this study chose to employ a standardized mind map approach. This allowed us to amalgamate all the extracted taxonomies into a unified model. Figure 1 visually depicts the consumer-centric IaaS excerpts that have been individually transformed into mind maps.

3.2. Refinement

This step analyzed semantic consistencies among the taxonomies from the previous step and subsequently merged them into a unified taxonomy by employing a binary integration approach. The algorithm of the binary integration is as follows.

- Step 1: A pair of taxonomies from Figure 1 that have some common topic(s) (which will be used as a merging point) is selected each time. For example, taxonomies in [20] and [26] have cloud service pricing and pricing models, respectively, as a common topic.
- Step 2: Any redundant and inconsistent topics between both taxonomies from step 1 were identified. For example, subscription in [20] is redundant with subscription in [26].
- Step 3: All redundant topics if there is any, except the one to be used as merging point(s), in both taxonomies are removed to retain the topic's uniqueness. For example, only subscription in [26] is removed.
- Step 4: Any synonymous topics are resolved by choosing the most appropriate topic and renamed the others to be the chosen one. For example, since cloud service pricing in [20] is synonymous with pricing models in [26], cloud service pricing in [20] is renamed to pricing models as that of [26].
- Step 5: Any homonymous topics are resolved by renaming each of them to a distinct term. For example, since availability in [25] and availability in [23] refer to the availability aspect of security and SLA, respectively, availability in [23] is renamed to availability aspect instead.
- Step 6: Merge both taxonomies into a single one by using the merging points. For example, taxonomy in [20] and taxonomy in [26] are merged by using the same topic pricing models.
- Step 7: Repeat step 1 to step 6 for the remaining pairs of taxonomies, including the merged one resulting from step 6, until a unified and consistent taxonomy is achieved.

As a result of refinement, the algorithm resolved 121 (sub)topics out of the total (sub)topics that were redundant and 67 inconsistent (sub)topics that held synonyms and homonyms to obtain a unified taxonomy at last.

3.3. Formalization

All of nonredundant topics and subtopics in the unified taxonomy were converted into classes and subclasses that were connected to other classes and subclasses based on semantic domains. The root class is denoted as "owl: Thing". Consequently, the proposed ontology is composed of 15 direct subclasses, as illustrated in Figure 2: Performance, Resource abstraction, Core service, Support, Costs, Security, Management, Value-added services, Resource deployment, Control, Standardization efforts, Flexibility, Deployment types, Integration, and Organization. The total number of subclasses is 340, such as Performance feature, Computing time, and Connection bandwidth, that are arranged hierarchically in a tree structure with a height of 7, as depicted in Figure 3.

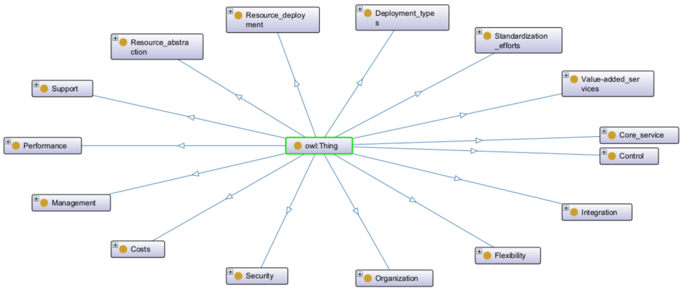


Figure 2: Proposed ontology and its direct subclasses

To provide a comprehensive understanding of the proposed ontology, each of direct subclasses is essentially described one by one, including its nested subclasses enclosed within curly brackets. Furthermore, supplementary explanations are provided within parentheses immediately following respective (sub)classes.

- Performance of IaaS has the following subclasses : Performance feature (identifying the atomic elements of cloud-service performance evaluation), Computing time, Connection bandwidth, Instance capacity, and Load balancing of Hardware or software type.
- Resource abstraction is various resources offered as services. Its subclass and nested subclasses are as follows. Compute = {OS as a Service = {Software licensing = {Open-source,Proprietary}}, Desktop as a Service (DskTaaS), High Performance Computing as a Service (HPCaaS)}, Storage = {Raw Storage as a Service (i.e., block level storage), File

System as a Service, Storage management = {Free space, Maximum file size, Encryption on servers, Encryption on transfer, Sharing, Backup, Replication, Snapshots}, Storage organization = {Image, Scheme (i.e., block, file, or object storage)}, Data Storage as a Service (DRaaS) = {Cold-site DRaaS (CDRaaS), Warm-site DRaaS (WDRaaS), Hot-site DRaaS (HDRaaS)}, Backup as a Service (BaaS) = { Data Storage as a Service (DSaaS)}}. Network = {Link Layer as a Service, Network Layer as a Service, Transport Layer as a Service, Application Layer as a Service, Traffic = {Incoming, Outgoing}, Addressing (providing accessibility to applications and underlying virtual machines once moved to new networks), Application-level communication (specifying API to be RESTful to decouple client from server components and advocate interoperable IaaSs via standard interfaces), Mobile Network Virtualization as a Service (MNVaaS), Bandwidth on Demand as a Service (BoDaaS), Virtual Private Network as a Service (VPNaaS)}, Infrastructure Services as a Service (ISaaS) (which provides a set of auxiliary services) = {Billing as a Service (BLaaS), Security as a Service (SECaaS) (encompassing Identity as a Service (IDaaS) for managing authentication and authorization), Auditing as a Service (AaaS) (for checking providers for standard compliance), Policy Management as a Service (PMaaS) (for handling all access policies across multiple providers)}}.

- Core service is fundamental to all other services. Its subclasses are {Image repository, Charging and billing, SECaaS = {Auditing as a Service (AaaS) (i.e., Logging), Policy Management as a Service (PMaaS), Identity as a Service (IDaaS)}}.

- Support is services used to operate some other services. It has three subclasses: Message bus (providing a means for passing messages between different cloud services), Database, and Transfer service (for other layers to communicate and interact).

- Costs = {Price class = {Price level (all factors affecting resulting cost directly), Price resilience (price options for flexibility purpose)}, Payment = {Time of payment, Payment method, Payment model = {Advance payment, Consumption payment}}, Service charging = {Assessment basis (how regular billing occurs such as hourly or monthly), Charging granularity}, Pricing models = {Value-based pricing (estimating customers' satisfaction) = {Service-based pricing (focusing on service content) = {On-demand pricing, Tier-based pricing, Per-user-based pricing, Per-device-based pricing, All-you-can eat (buffet pricing), Priority pricing}, Experience-based pricing (based on performance) = {Outcome-based pricing, Customer-care-based pricing, Brand-based pricing, Usage-experience pricing}, Customer-value-based pricing (a price from a subjective view of a customer) = {Perceived-value pricing, Psychological pricing, Feature-based pricing, Hedonic pricing}}, Market-based pricing (equilibrium of customers and providers) = {Free-upfront-and-pay-later pricing = {Free products-pricing-on-advertising, Freemium-based pricing, Razor-and-blades pricing (giving away nonconsumable element and charging consumable replacement element)}, Auction-and-online-based pricing = {Spot-and-forward pricing (i.e., current and future prices), English auctions (Open Ascending), Dutch auctions (Open Descending), Sealed-bid auction, Going-rate pricing, Online pricing}, Retail-based pricing (for small quantity purchase) = {Hybrid-or-product-mix pricing (combining different pricing models) = {Product-line pricing, Optional-feature pricing, Captive-product pricing (i.e., cheap core part with costly accessory), Two-part pricing, By-product pricing, Product-bundling pricing}, Discounts pricing = {Early payment, Off-season, Bulk purchase, Retail discount, Cash discount, Trade in allowance (discount in exchange of buyer's asset)}, Promotional pricing = {Loss leader (selling below market price), Special event pricing, Cash rebate, Low interest financing, Longer payment terms, Psychological discounting, Warranties & service contracts}, Discriminatory pricing (charging different prices to different customers) = {Customer segment, Product form (different prices for different versions of a product), Image pricing, Location pricing, Geographical location, Dynamic or surge-based pricing (based on current market demands), Loyalty programming (rewarding customers to continue buying from the brand}}}, Cost-based pricing (covering Capex and Opex) = {Expenditure-based pricing = {Cost-plus pricing (cost plus margin), Percentage pricing, Target-return pricing}, Resource-based pricing = {Volume-based pricing, FTE (full-time equivalent)-based pricing, Licensing-based pricing, Time-material-based pricin}, Utility-based pricing = {Peak-and-off-peak pricing, Reserved-based pricing (e.g., Subscription or no up-front, partial up-front, and all up-front), Fixed-cost-based pricing, Dynamic pricing}}}}, Formula = {Linear model, Step model, Variation of linear model}, Degree of discrimination (how a service is offered for different buyers for different prices) = {No discrimination, Second degree discrimination (when providers sell different units for different prices where customers must do self-selection to choose from the offers), Third degree discrimination (vendor identifies different customer groups based on their willingness-to-pay and can be personal (e.g., student discounts) or regional (e.g., different prices for developing countries))}}}.

- Security = {Availability = {Confidentiality, Integrity, Fault tolerance, Authentication, Authorization, Accounting, Security monitoring, Static IP address, Firewalls, Data center security = {Hardware security, Software security, Data center redundancy}, Network security = {Connection security, Network redundancy}, IT compliance = {Data security, Data privacy, Datacenter location, Access security, Disaster recovery}, User management, Verification, Security groups, Single sign-on, Security measures = {None, SSL, PKI, VPN}}.

- Management = {Management tools = {CLI tools, APIs, Dashboard, Orchestrator, Federation management, Elasticity management}, Formal agreement = {No SLA, SLA = {SLA objectives (defining quality-of-service measurement in SLAs), Scope, Quality of Service, Functional service quality = {Data IO, Usability, Response behavior, Error behavior}, Nonfunctional service quality = {Availability aspect, Performance aspect, Security aspect, Integrability, Utilization fee, Location}, SLA monitoring, SLA enforcement, Governance = {KPI, Penalty, Incentive, Exit clause, Modality}, Charge process, Liability and penalties, Validity period (of negotiated SLA), Resource guarantee, Architecture

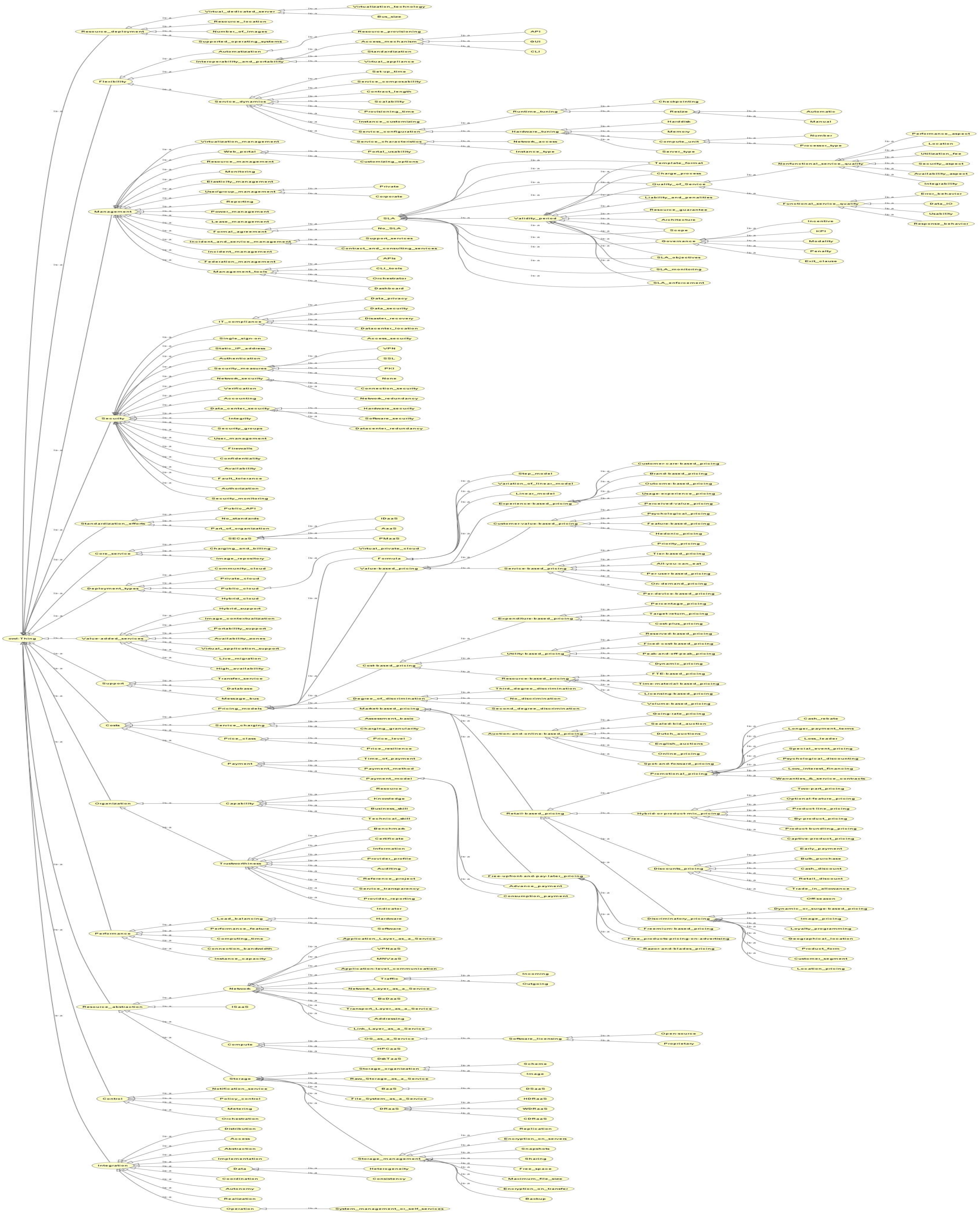


Figure 3: Proposed ontology's complete class hierarchy

(of SLA measures and SLA requirement management for different IaaSs; For example, Web Service Agreement Specification (WSA) is the standard for the SLA management architecture in Web service environments), Template format (used to electronically represent SLA for automated management)}, Monitoring, Reporting, Lease management, Incident and service management = {Contract and consulting services, Support services}, Web portal = {Portal_usability, Customizing options}, Virtualization management, Resource management, User/group management = {Corporate, Private}, Incident management, Power management}.

- Value-added services = {Availability zones, High availability, Hybrid support (facilitating the implementation of hybrid cloud by resource extension to external), Live migration, Portability support, Image contextualization (enabling virtual machine (VM) instance to be deployed in the form of a shared customized image for specific context such as VM with a turnkey database), Virtual application support (i.e., containers consisting of several VMs and allowing design and configuration of multi-tier applications)}.
- Resource deployment = {Virtual dedicated server = {Virtualization technology, Bus size (or processor register size e.g. 64 bits)}, Number of images, Resource location, Supported operating systems (OS supported by providers)}.
- Control provides cloud systems with basic control features. Its subclasses are Metering, Policy control, Notification service, and Orchestration.
- Standardization efforts = {No standards, Public API (i.e., common API enabling interoperability and customization), Part of organization (i.e. organization involves in public standardization)}.
- Flexibility = {Interoperability and portability = {Standardization, Access mechanism = {API, GUI, CLI}, Virtual appliance (delivering a service as a complete software stack installed on a VM)}, Automatization = {Resource provisioning}, Service dynamics = {Provisioning time, Contract length, Set-up time, Scalability, Instance customizing, Service composability, Service characteristics = {Instance type, Network access}, Service configuration = {Hardware tuning = {Compute unit = {Number, Processor type}, Server type, Memory, Harddisk}, Runtime tuning = {Resize = {Automatic, Manual}, Checkpointing}}.
- Deployment types = {Public cloud = {Virtual private cloud}, Private cloud, Hybrid cloud, Community cloud}.
- Integration = {Distribution, Coordination, Implementation, Access, Abstraction, Autonomy, Operation = {System management or self-services}, Realization, Data = {Heterogeneity, Consistency}}.
- Organization = {Trustworthiness = {Certificate, Reference project, Indicator, Benchmark, Provider reporting, Information, Provider profile, Service transparency, Auditing}, Capability = {Resource, Knowledge, Technical skill, Business skill}}.

The properties of the classes are hasDirectCost, hasMgmtAspect, hasSecurityAspect, and useTechnology as depicted in Figure 4.

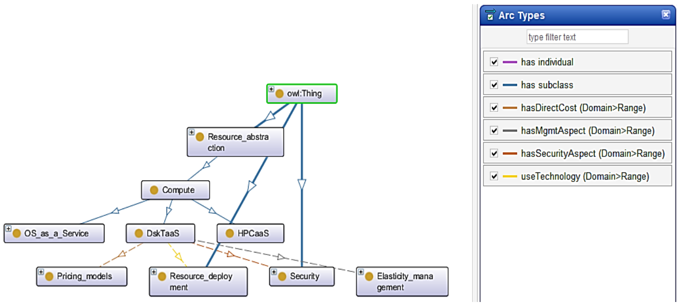
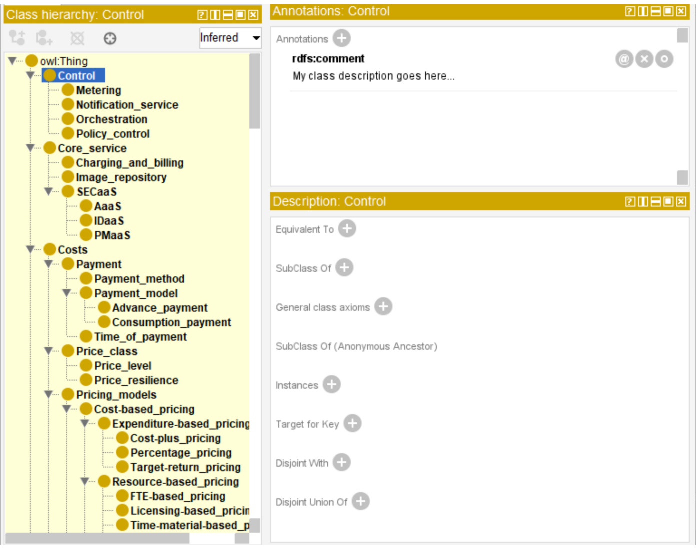


Figure 4: Ontology's class properties

4. Evaluation

The proposed ontology is evaluated into two crucial parts, a technology-focused evaluation and a user-focused evaluation, as follows.

For technology-focused evaluation, this paper employed a HermiT reasoner to rigorously determine whether the proposed ontology is syntactically and semantically consistent and identify subsumption relationships between classes. The result of this evaluation is an inferred ontology that is prominently displayed in yellow background in Figure 5 without any error (which will be indicated by red text if there is any).



rdfs:comment
My class description goes here...

Figure 5: Inferred ontology by using HermiT

Complementing the technological assessment, the study conducted a comprehensive user-focused evaluation. This evaluation entailed applying our ontology to assess a worldwide recognized Infrastructure as a Service (IaaS) platform, namely, AWS EC2. To accomplish this, the study carefully generated ontology individuals by drawing from publicly available information pertaining to AWS EC2, as exemplified in Figure 6, where these individuals are denoted by prefix violet diamond symbols. All pieces of information regarding AWS EC2 offerings for consumers can be seamlessly mapped into the ontology's existing classes (denoted by prefix orange oval symbols) to facilitate comprehensive IaaS selection.

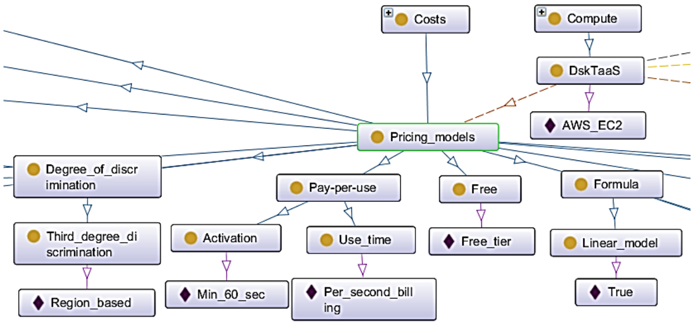


Figure 6: Ontology's instance portion for AWS EC2's offerings

Furthermore, in Figure 7, the study performs two distinct SPARQL queries against the ontology. The first query below

SELECT ?subject WHERE {

?subject rdf:type owl:Class.

}

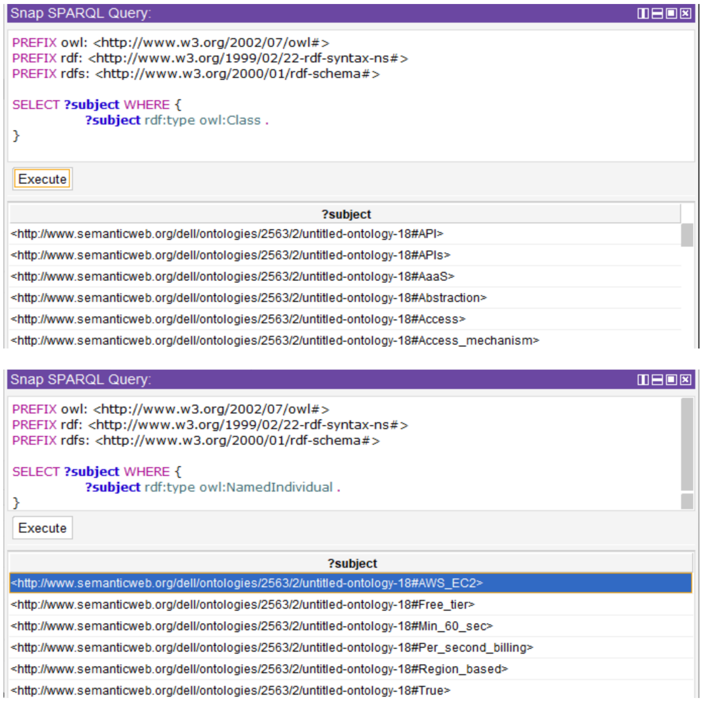
aims to list all (sub)classes. Part of the resulting classes of the query are listed in the bottom pane in the figure. The second query below

SELECT ?subject WHERE {

?subject rdf:type owl:NamedIndividual.

}

identifies and enumerates individual instances as partially displayed in the resulting pane in the figure. The successful execution of both queries substantiates the potential for automating IaaS assessment processes using our ontology.



```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?subject WHERE {
        ?subject rdf:type owl:Class .
}
```

```
?subject
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#API>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#APIs>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#AaaS>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Abstraction>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Access>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Access_mechanism>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?subject WHERE {
        ?subject rdf:type owl:NamedIndividual .
}
```

```
?subject
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#AWS_EC2>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Free_tier>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Min_60_sec>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Per_second_billing>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#Region_based>
<http://www.semanticweb.org/dell/ontologies/2563/2/untitled-ontology-18#True>
```

Figure 7: SPARQL queries for classes and individuals

5. Conclusion

This paper presents a novel customer-perspective IaaS ontology developed from various 18 IaaS taxonomies in present existence. This ontology stands out for its exceptional comprehensiveness, encompassing a total of 15 primary subclasses (e.g., performance, costs, and security) and 340 individual classes (e.g., instance capacity, availability, and price class). The evaluation shows that the proposed ontology is syntactically and semantically consistent. Furthermore, the ontology successfully enables not only the assessment of AWS EC2 IaaS but also SPARQL queries. This has affirmed that the ontology holds significant semantic value, offering utility not only to researchers but also to practitioners by leveraging it as a foundational component to develop a sophisticated assessment tools for facilitating effective IaaS adoption. The tool will be definitely helpful for IaaS customers, IaaS providers, and IaaS partners. The future work of this study is to develop an expert system in the form of SaaS to facilitate IaaS selection based on the proposed ontology.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This research was financially supported by Department of Computer Science, Kasetsart University, Krung Thep Maha Nakhon, Thailand.

References

[1] ISO/IEC, ISO/IEC 17789: Information technology — Cloud computing — Reference Architecture, ISO/IEC, 2014.

[2] F. Nadeem, "Evaluating and Ranking Cloud IaaS, PaaS and SaaS Models Based on Functional and Non-Functional Key Performance Indicators," IEEE Access, **10**, 63245-63257, 2022, doi: 10.1109/ACCESS.2022.3182688.

[3] E. Al-Masri, L. Meng, "A Quality-Driven Recommender System for IaaS Cloud Services," in 2018 IEEE International Conference on Big Data (Big Data), 5288-5290, 2018, doi: 10.1109/BigData.2018.8622017.

[4] S. M. M. Fattah, A. Bouguettaya, S. Mistry, "Signature-based Selection of IaaS Cloud Services," in 2020 IEEE International Conference on Web Services (ICWS), 50-57, 2020, doi: 10.1109/ICWS49710.2020.00014.

[5] K. Kritikos, K. Magoutis, D. Plexousakis, "Towards Knowledge-Based Assisted IaaS Selection," in 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 431-439, 2016, doi: 10.1109/CloudCom.2016.0073.

[6] F. Moscato, R. Aversa, B. Di Martino, T. -F. Fortiş, V. Munteanu, "An analysis of mOSAIC ontology for Cloud resources annotation," in 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), 973-980, 2011.

[7] A. Abdullah, S.M. Shamsuddin, F.E. Eassa, Ontology-based Cloud Services Representation. Research Journal of Applied Sciences, Engineering and Technology. **8**(1), 83-94, 2014, doi:10.19026/rjaset.8.944.

[8] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, M. Menzel, S. Nepal, "An ontology-based system for Cloud infrastructure services' discovery," in 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 524-530, 2012.

[9] Q. Zhang, A. Haller, Q. Wang, CoCoOn: Cloud Computing Ontology for IaaS Price and Performance Comparison. in International Semantic Web Conference. 325-341, 2019, doi.org/10.1007/978-3-030-30796-7_21.

[10] T. Banditwattanawong, "The survey of infrastructure-as-a-service taxonomies from consumer perspective," in the 10th International Conference on e-Business, 2015.

[11] Z. Zhang, C. Wu, D.W.L. Cheung, "A survey on cloud interoperability: taxonomies, standards, and practice," ACM SIGMETRICS Performance Evaluation Review, **40**(4), 13-22, 2013, doi.org/10.1145/2479942.2479945.

[12] M. Firdhous, "A Comprehensive Taxonomy for the Infrastructure as a Service in Cloud Computing," in 2014 Fourth International Conference on Advances in Computing and Communications, 2014, 158-161, doi: 10.1109/ICACC.2014.45.

[13] C.N. Höfer, G. Karagiannis, "Cloud computing services: taxonomy and comparison," Journal of Internet Services and Applications, **2**, 81-94, 2011, doi.org/10.1007/s13174-011-0027-x.

[14] R. Dukarić, J. Matjaz, "A Taxonomy and Survey of Infrastructure-as-a-Service Systems," Lecture Notes on Information Theory, **1**, 29-33, 2013, doi: 10.12720/lnit.1.1.29-33.

[15] C. Wu, R. Buyya, K. Ramamohanarao, "Cloud Pricing Models: Taxonomy, Survey, and Interdisciplinary Challenges," ACM Computing Survey, 52(6), 1-36, 2019, doi.org/10.1145/3342103.

[16] Shivani and A. Singh, "Taxonomy of SLA violation minimization techniques in cloud computing," in Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 1845-1850, 2018, doi:10.1109/ICICCT.2018.8473230.

[17] T. Banditwattanawong, M. Masdisornchote, "Federated Taxonomies Toward Infrastructure-as-a-Service Evaluation," International Journal of Management and Applied Science, **2**(8), 33-38, 2016.

[18] S. Staab, R. Studer, Handbook on Ontologies (2nd. ed.). Springer Publishing Company, Inc., 2009.

[19] R. Dukaric, M.B. Juric, "Towards a unified taxonomy and architecture of cloud frameworks," Future Generation Computer System, **29**(5), 1196–1210, 2013, doi.org/10.1016/j.future.2012.09.006..

[20] S. Kansal, G. Singh, H. Kumar, S. Kaushal, "Pricing models in cloud computing," in International Conference on Information and Communication Technology for Competitive Strategies, 33:1–33:5, 2014, doi.org/10.1145/2677855.2677888.

[21] M.K.M. Murthy, H.A. Sanjay, J.P. Ashwini, "Pricing models and pricing schemes of iaas providers: A comparison study," in Proceedings of the International Conference on Advances in Computing, Communications and Informatics, 143–147, 2012, doi:10.1145/2345396.2345421.

[22] S. Kächele, C. Spann, F. J. Hauck, J. Domaschka, "Beyond IaaS and PaaS: An Extended Cloud Taxonomy for Computation, Storage and Networking," in IEEE/ACM 6th International Conference on Utility and Cloud Computing, 75-82, 2013, doi: 10.1109/UCC.2013.28.

[23] J. Repschlaeger, S. Wind, R. Zarnekow, K. Turowski, "A Reference Guide to Cloud Computing Dimensions: Infrastructure as a Service Classification Framework," in The 45th Hawaii International Conference on System Sciences, 2178-2188, 2012, doi: 10.1109/HICSS.2012.76.

[24] B.P. Rimal, E. Choi, I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," in The Fifth International Joint Conference on INC, IMS and IDC, 2009, 44-51, doi: 10.1109/NCM.2009.218.

[25] H. Kamal Idrissi, A. Kartit, M. El Marraki, "A taxonomy and survey of Cloud computing," in National Security Days (JNS3), 1-5, 2013, doi:10.1109/JNS3.2013.6595470.

[26] S. Gudenkauf, M. Josefiok, A. Göring, O. Norkus, "A Reference Architecture for Cloud Service Offers," in The 17th IEEE International Enterprise Distributed Object Computing Conference, 227-236, 2013, doi: 10.1109/EDOC.2013.33.

[27] R. Prodan, S. Ostermann, "A survey and taxonomy of infrastructure as a service and web hosting cloud providers," in The 10th IEEE/ACM International Conference on Grid Computing, 17-25, 2009, doi: 10.1109/GRID.2009.5353074.

[28] R. Teckelmann, C. Reich, A. Sulistio, "Mapping of Cloud Standards to the Taxonomy of Interoperability in IaaS," in IEEE Third International Conference on Cloud Computing Technology and Science, 522-526, 2011, doi:10.1109/CloudCom.2011.78.

[29] G. Laatikainen, A. Ojala, O. Mazhelis, "Cloud services pricing models," in Software Business. From Physical Products to Software Services and Solutions - 4th International Conference, 117–129, 2013, doi: 10.1007/978-3-642-39336-5_12.

[30] F. Polash, A. Abuhussein, S. Shiva, "A survey of cloud computing taxonomies: Rationale and overview," in The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014), 459-465, 2014, doi:10.1109/ICITST.2014.7038856.