Semantic Matching for Adapting Heterogeneous IoT Services.

Jairo Ariza
Systems and Computing Engineering Department Universidad de los Andes, Colombia
ja.arizac1@uniandes.edu.co

Nicolás Cardozo
Systems and Computing Engineering Department Universidad de los Andes, Colombia
n.cardozo@uniandes.edu.co

Kelly Garcés
Systems and Computing Engineering Department Universidad de los Andes, Colombia
kj.garces971@uniandes.edu.co

ABSTRACT

Due to a continuously dynamic and changing environment, adaptation to these situation is an important concern in Internet of Things (IoT) systems. Our main concern are the inclusion of IoT devices on the system, automatic matching based on properties comparison and adaptation when previously existing connection are not longer available. In this paper, we propose an solution based on IoT and web Services ontologies, ontology instance matching and guidepost registries. Also an architecture is conceived to provide the required adaptations. It is based on the IoT reference architecture with new additional components. The future work will cover development of the proposed solution and experiments to measure its performance.

KEYWORDS

IoT adaptation, Ontology instance matching, IoT ontology matching.

ACM Reference Format:
Jairo Ariza, Nicolás Cardozo, and Kelly Garcés. 2018. Semantic Matching for Adapting Heterogeneous IoT Services.. In *Proceedings of SELF'18 (Seminario de tesis MISO).* ACM, New York, NY, USA, 7 pages. https://doi.org/-

1 INTRODUCTION

With the advent of IoT devices, new environments emerge that provide more means for users' interactions. Moreover, software systems gain more meaningful descriptions of the physical surroundings, using the data provided by sensors. By exploiting this raw data, software systems can interact with the physical environment and the users through services, or other IoT devices.

Due to the high amount of IoT devices and the dynamic nature of the environment, the description of, and discovery between IoT devices and web services needs to be changed to accommodate for autonomous interactions. Moreover, the middleware must be able to adapt transient IoT device or web service connections, in order to assure continuous communication and fault tolerance in the system (Section 2).

To address the fist problem, we propose a semantic representation based on ontologies. Such representation is based on a combination of the Service Integration Ontology (SIO) [16] and W3C Semantic Sensor Network (SSN) ontologies [4]. The intent of combining these ontologies is to provide a description of the physical and virtual properties of devices and web services.

In the second case, a combination of syntactic and semantic service matching techniques is considered as a solution. Connection between devices and services is enabled by matching algorithms based on semantic, syntactic, data type and value matching. The matching algorithms compare the functional and non-functional properties, and weight personalizing to determine a matching degree between services and IoT devices.

The adaption of the third problem, involves the use of the matching results, organized on a registry that relates web services, IoT devices, and their health status.

This paper proposes an adaptive architecture for IoT systems (Section 4). The main concern addressed in such architecture is resiliency and means of adaptation to the changing environment. The architecture consists of an integration of classical components needed in an IoT architecture [1], and new components to complement the required adaptation [3].

We propose an experiment for validation of the proposed architecture (Section 5), to demonstrate the feasibility of our approach to adapt and communicate entities autonomously in an IoT environment. In Section 6, we conclude the paper documenting the state of the art describing different approaches that help us to direct our research.

2 PROBLEM DESCRIPTION AND REQUIREMENTS

IoT systems have a highly dynamic nature due to the great amount of devices that can be simultaneously connected and the fact that these devices can appear or disappear from the system, in a quick, continuous and unannounced way. Without description of functional and non functional properties of connected IoT devices, external web services are unable to discover suitable IoT devices to establish a connection with their data stream. Even with the proper IoT devices description, it can be problematic for web services to find the right device for their requirements, forbidding a seamless and efficient way to connect web services to IoT devices.

Moreover, if we assume a successful connection, the dynamic nature will continue to pose a challenge with intermittent connections or total disconnections, interrupting the flow of data from the IoT devices to the web services.

The following research questions are formulated to address the defined problem:

(1) How to described IoT devices to allow their discovery external web services, through the middleware?

Seminario de tesis MISO, 201820, Bogotá, Colombia

ACM ISBN -.
https://doi.org/-

(2) How can web services establish an automatic connection with an specific IoT device, that partially or fully comply with the specified requirements?
(3) How the middleware can adapt previously existing connections between IoT devices and web services, when one of the endpoints is unavailable?

The specific requirements to solved the three defined problems are the following and are listed accordingly:

(1) A way to describe the functional and non functional properties of IoT devices and web services connected to the middleware.
(2) An algorithm to perform matching between web services and IoT devices (or vice versa).
(3) An algorithm to adapt existing connections between IoT devices and web services, when one of them is unavailable.

3 SOLUTION

This section presents our proposed solution to the problems described in Section 2.

To solve the first problem, the description of the different entities in the environment,we investigate several methods to describe domain specific knowledge. In our state of the art development, we found three main approaches:

(1) The Domain Specific Language (DSL) [5].
(2) The Web Services Description Language (WSDL) format in conjunction with the Universal Description, Discovery, and Integration (UDDI) directory and the Simple Object Access Protocol (SOAP) protocol[6].
(3) Semantic ontologies [4].

Our chosen solution is to describe IoT devices, by using semantic ontologies. The WSDL approach is discarded because the lack of semantic capabilities of this description. Compared with the DSL approach, the semantic ontologies provide a richer semantic meaning of the attributes of a domain specific language.

In addition to the selected description method, there are another two important aspects to describe an IoT device, its functional and non-functional attributes. The functional attributes describe aspects related to the behavior of the device. The non-functional attributes specify criteria used to judge the performance of a system or device.

We define an ontology based on a combination of the SSN ontology [4], the SIO ontology [16] and a Quality of Service (QoS) ontology[22]. The SSN ontology was implemented to describe the functional and non-functional attributes of the IoT device. The SIO ontology is used to describe web services. In our context, we integrate this ontology to describe the data stream provided by the IoT device as a web service and the functional attributes of the entity representation. To complete the proposed ontology, the QoS ontology was integrated to describe the non-functional properties of the web services.

Figure 1 shows the non-functional attributes represented on the SSN ontology, and Figure 2 illustrates an excerpt from the modified proposed ontology.

The second problem, the autonomous connect between entities in the environment, is addressed by using an ontology instance matching. Shvaiko and Euzenat [18] document several matching methods in their book, which includes semantic, syntactic, taxonomy based, model-based, and graph-based methods. Shvaiko and Euzenat [18] also states that the best option is to combine different techniques in order to improve the matching results. In our case, we combine semantic, syntactic, data type and value matching. Using these methods, we compare the functional and non-functional properties between IoT devices and web services to find the most appropriate match that fulfills the specified requirements. The system also establish a subscription between a web service and the stream of data provide by the most suitable IoT device to provide the required information.

When comparing functional and non-functional properties, priority definition is needed to establish the relevance of a property for a specific requirement. Weight adjustments is a common approach to solve this need [8][23][25]. The numeric values of the weight can be provided by the user or by an algorithm. In our research, we determine the weight adjustments based on the specified application intention, and applicable heuristics of the control systems domain. The intended objective of a service can range from monitoring, provide non-critical alarms, operations control or critical alarm annunciation.

The final problem, adaptation of existing interactions, is addressed combining two approaches. The first approach is the ontology instance matching [18]. The second approach is the use of an execution guidepost [3]. When the first matching between a service and an IoT device is executed, more than one suitable match can be found. At this point, we create a key-value pair containing the web service as key and the list of suitable IoT devices as value. This key-value pair will be persisted in the execution guidepost to enable the required adaptation. If the previously established subscription between the web service and the most suitable IoT device is lost, our system will consult the guidepost for the next suitable IoT device on the list to execute a new subscription between the two endpoints.

4 PROTOTYPE

This section describes our proposed architecture, shown in Figure 3, to deal with adaptation of IoT environments. Several architectural components are implemented by the reference IoT architecture [1]. To address the problems described in Section 2, we integrate seven new components as part of the adaptive IoT architecture. The components are a Server gateway, resource discovery, instance matching, execution guidepost, and healthcheck controller, which are described in the following.

4.1 OM2M server/gateway

We support our proposed architecture on top of the OM2M platform [10], this platform is used for its multiple communications. This component interacts directly with the physical IoT devices. Its primary function is to provide a protocol abstraction of the IoT devices, allowing the services to connect to the IoT devices' data streams (exposed as URLs), using the HTTP protocol. Additionally, OM2M can handle a distributed architecture using gateways located on different locations, managed by a server.

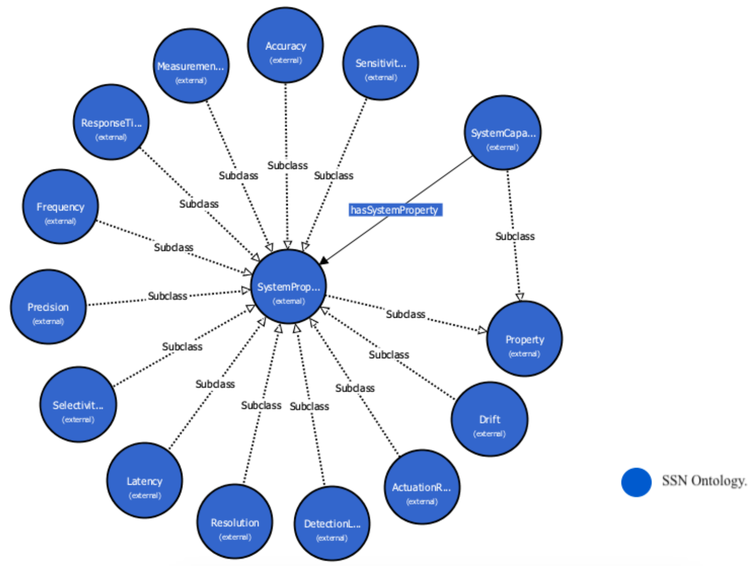

Figure 1: Non-functional attributes (System Property) of the SSN ontology [24].

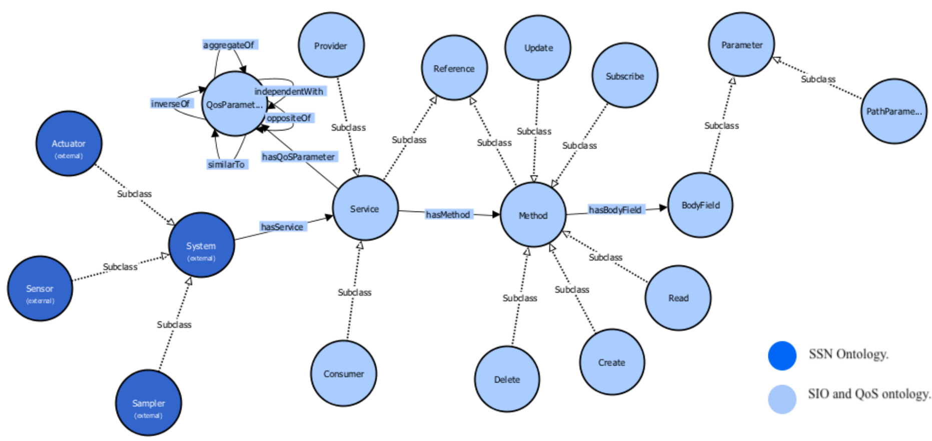

Figure 2: Proposed ontology.

4.2 Resource discovery

The resource discovery component automatically search the OM2M network for all the data streams provided by IoT devices. For every data stream, the resource discovery collect metadata that describes the functional and non-functional properties of the entities. The metadata is advertised by the OM2M platform using the Extensive Markup Language (XML), or JavaScript Object Notation (JSON) formats.

4.3 Ontology and instance manager

The resource discovery results are parsed as individuals of the proposed ontology by the Ontology and instance manager component. When a new IoT device is discovered, this component also examines the non-functional properties of the new device and compares it with the existing ones. If no identical device exists, a new individual is created on the ontology. In the opposite case, the existing identical individual is reused, optimizing the use of the ontology.

4.4 Instance matching

As described in Section 2, our second concern is to provide matching between services and data stream provided by the devices. The Instance matching will execute a combination of the matching algorithms (syntactic, semantic, data type, and value matching). Using these methods, it compares the functional and non-functional properties between IoT devices and web services to find the most appropriate match that fulfills the specified requirements. The instance matching also establish a subscription between a web service and the stream of data provide by the most suitable IoT device.

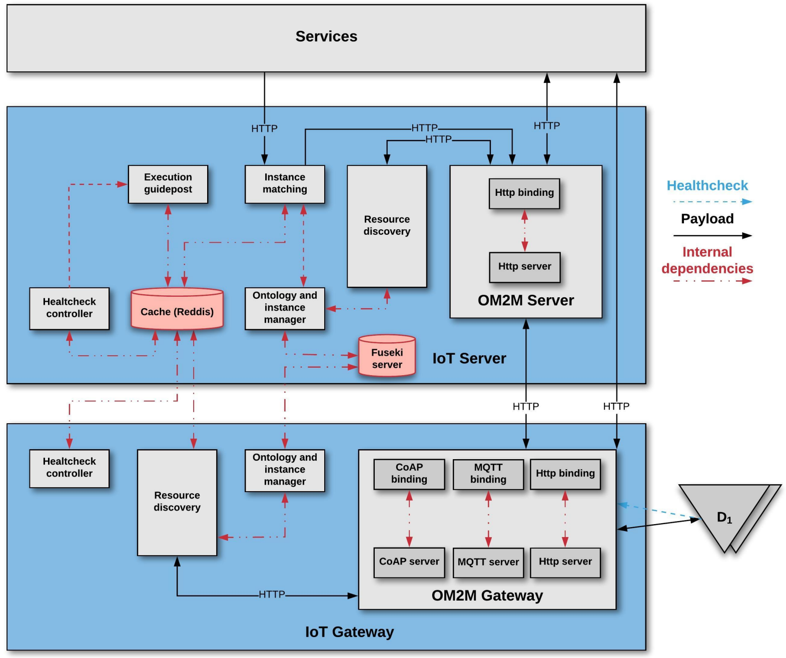

Figure 3: IoT architecture for adaptation.

4.5 Execution guidepost

As stated en Section 3, the execution guidepost [3] is a registry used to keep track of the subscription relationship between services and the most suitable IoT devices. In addition to a list of suitable devices, the Execution guidepost also contain the availability of the data provided by each device. This information will be used as a means of adaptation. When the connection with the device is lost, the system will search for the next most suitable an available device to establish a new subscription between the parties. Device availability will be determined by the Healthcheck Controller component.

This registry is persisted on a in-memory key-value database used as cache, to provide fast access and response. The cache technology selected by our research is Reddis.

4.6 Healtcheck Controller

As described in the previous section, the function of the Healthcheck controller is to determine the availability status of each available data stream. This component will subscribe to every data stream to monitor a periodic healthcheck signal. If a determined number of healthcheck signals are not received, the Controller will mark the data stream as unavailable and update the availability state of the corresponding data stream on the Execution guidepost.

4.7 Fuseki server

This component manages access to the proposed ontology described in Section 3. It stores the ontology together with its instances describing the particular devices and services of our IoT domain.

5 VALIDATION

This section presents the planning for feasibility and testing of our proposed IoT architecture.

5.1 Evaluation Scenario

To obtain appropriate results, our case study must contain the presence of several different and identical IoT devices, which advertise their data on different URLs. Due to the absence of a significant amount of physical devices, we will emulate the devices, but we will take into account the inherent resource limitations of the devices. By experimentation, we have determined that the main constraint which concern us is the size of the payload send by the device. In our experiments we found that the maximum allowable payload is 256 characters for the NodeMCU IoT device. Taking this constraint into consideration, we will restraint the size of the payload in our emulated IoT devices.

The scenario will evaluate scalability, by progressively increasing the number of IoT devices from 100 up to 1000. Each device will be emulated using Python code, simulating inclusion of IoT devices in the middleware, data streaming from each simulated device (reading of the physical variable), malfunction or intermittent behavior of devices, and discovery/subscription to IoT data streams by web services.

The IoT server, the IoT gateway, the Fuseki server and Cache server, will be implemented on virtual machines in a local area network located at the university campus.

5.2 Methodology

The first part of our experiment will evaluate the inclusion of new simulated IoT devices. Each device will advertise its functional and non-functional properties. We will configure the devices' emulation to randomly create a new device with identical system properties, but with different web services properties.

The first metric we will measure is the amount of time taken to match the new IoT device with the existing individuals on the ontology. The second metric of interest is the time consumed to create the new individual on the ontology. After a new individual is added, the reasoner must be executed to verify if the logical of the ontology and its individuals is decidable.

The performance test will evaluate the variation of the previously described times, when the number of existing individuals increases.

The second part of the experiment will measure the matching time between web services requirements and existing IoT devices data streams. The performance test will also quantify the time variations for the matching process as the number of existing individuals in the ontology increases.

6 RELATED WORK

This section puts existing approaches that address ontology matching application and domain specific representations in the IoT systems perspective. Here we discuss the related work in these areas.

6.1 Systematic literature review

To develop the related work, we queried the Scopus database. We used several keywords related to the IoT domain, ontologies representation and matching, web services and DSL. The relevance of the founded documents was determined by the following criteria: (1) Documents published in the last 5 years or less, (2) most cited documents, and (3) document relevance to the defined keywords.

The keywords used on the queries and the number of documents founded are the following:

(1) "IoT ontology matching", 8 relevant over 19 documents founded.
(2) "IoT SSN", 3 relevant over 32 documents founded.
(3) "Ontology instance matching", 2 relevant over 3141 documents founded.
(4) "QoS ontology matching", 2 relevant over 108 documents founded.
(5) "UDDI", 3 relevant over 20 documents founded.
(6) "UDDI matching web services" 2 relevant over 173 documents founded.
(7) "DSL Dustdar", 3 relevant over 64 documents founded.

6.2 Knowledge representation

Domain specific language description has extensive research with different approaches, like DSL, the UDDI and ontologies. From this approaches, Ontologies gather more attention from researchers, specially to describe IoT systems and its properties.

Ganzha et al. [7] identify the interoperability nature of heterogeneous IoT platforms and the need of a common method to represent the attributes of those platforms. In many cases the devices are represented using a format like the JSON or the XML. However the authors propose ontologies as a more universal representation enriched by the semantic meanings it provides. Ganzha et al. [7] recognize ontology matching as a suitable method to align concepts provided by IoT systems.

DSLs as used by Copil et al. [5] and Oberortner et al. [12] describe domains' specific knowledge. DSLs also define applicable rules with constraints to trigger specific actions when these are violated. Copil et al. [5] defines a constraint directive that stars with the keyword CONSTRAINT, and uses mathematical comparison signs (<,>,>=,<=,!=,==) to establish which values are acceptable. Moreover, Copil et al. [5] use DSL to control elasticity on cloud applications based on performance requirements like cost, resource performance and QoS. Oberortner et al. [12] also apply a QoS DSL to trigger alarms when web services' Service Level Agreements (SLA) gets violated.

Web services discovery (Curbera et al. [6]) is an existing technology, based on the UDDI directory, the WSDL format and SOAP. This technology allows registry, description and discovery of web services using syntactic matching. Due to the lack of expressivity, Paolucci et al. [13] and Purohit and Kumar [14] proposed a semantic extension of the WSDL and UDDI to provide semantic matching of the web services, to improve the matching results. Ran [15] and Seghir and Kazar [17] also modify the WSDL and UDDI with QoS attributes extend the discovery functionality to the non-functional domain.

In IoT systems, ontologies are nowadays, the most used method to describe this domain. The two most recognized IoT ontologies are the SSN [4] and the ETSI Smart Appliances REFerence (SAREF) [11]. Both of them provide functional properties description, but only SSN provides the required non-functional description required by our research.

Ontology matching and Ontology instance matching has raised the attention of many researchers and there is a great amount of research on this topic. Recently the raise of IoT systems originated an additional interest on applying the ontology knowledge in different contexts and applications of the IoT domain.

Castano et al. [2] describe the main ontology and instance matching techniques independent of the knowledge domain, which helps to align the functional attributes of the represented objects. The techniques include similarity-based methods, and reasoning-based methods. The similarity-based methods includes the syntactic approach using string matching algorithms and the semantic technique involves the use of a thesaurus or a lexical system (e.g., Wordnet). The reasoning-based approach embraces deductive (i.e., description logics) and probability methods (i.e., machine learning) to achieve ontology instance matching.

Junhao et al. [8] propose an algorithm for web services selection based on a ontology that describes nonfunctional parameters (i.e, QoS). The paper states the upper and middle QoS ontology for web services and how it can be used to execute web services matching bases on specific QoS requirements. Tran [22] also developes a QoS ontology for web services, which describes the QoS information and the related attributes in detail. Junhao et al. [8] use an algorithm that receives as input the QoS requirements from the user and performs a semantic matching, data verification, value matching and a personalized adjustment (i.e, weights adjustment) to determine the degree of similarity between web services.

Tang and Meersman [20] implement an ontology based discovery and a recommender system for IoT projects, where the user specifies its requirements and the recommender systems suggests similar existing solutions that satisfy users' needs. The ontology matching solution consists on a string matching algorithm (syntactic similarity), followed by a lexical matching algorithm supported by Wordnet (semantic similarity), and a graph matching algorithm that uses a lexons (sentences represented as vectors) table. The objective is to determine suitable recommendation according to the user's needs.

Tao et al. [21] describe IoT devices data, provided on different formats, as individuals of an ontology. The individuals represent historic data of the same Smart Home environment and the authors use ontology instance matching to adapt the Smart Home to changes on the user requirements and behavior. The ontology instance matching algorithm uses semantic similarity calculation of the individuals and their parent, children and siblings' nodes.

6.3 Ontology matching

Wang et al. [23] use ontology matching to align heterogeneous IoT systems. The proposed algorithm executes a semantic similarity matching, a syntactic similarity matching and weight adjustment calculation in parallel. The weight adjustment matrix is determined by calculating the similarity degree of the ontologies hierarchy. The calculated matrices are aggregated in a combined similarity matrix, which determines the final degree of alignment.

Zhou and Ma [25] present a service discovery algorithm for the IoT domain. The algorithm combines semantic similarity and semantic relativity to calculate the similarity of different concepts represented on the ontology. Zhou and Ma [25] also implement weights to adjust the parameters compared on the matching algorithm.

Kibria et al. [9] use semantic ontologies to represents IoT devices as Virtual Objects (VO) and Composite Virtual Objects (CVO), and enable the reuse of existing objects to render multiple heterogeneous IoT devices. When a new device is registered, semantic similarity between the attributes of the new IoT device and the existing VO is calculated using a synsets dictionary. If there is a match with an existing VO or with parts of different VO, the algorithm instantiates the new device on the existing objects. In case there is no match, a new VO is created on the ontology.

Song et al. [19] apply ontology matching to provide a recommendation system for task and work flow based service creations on a Smart building environment. When the user wants to create a new task, the recommendation system search for similar IoT devices based on user requirement for the task. The matching is executed calculating a semantic similarity between the requirements and the existing instances of the ontology, and comparing the hierarchical structure of the ontology using a semantic approach. Additionally, if there is an unavailable IoT device, the system will reconnect the service with a new similar suitable replacement.

Moreira et al. [11] explore a semantic translation to map two specific IoT ontologies, the SSN ontology and SAREF ontology. This article states that the translation between the two ontologies, cannot be fully performed due to the fact that the SAREF ontology does not provide a similar structure to the SSN Measurement Capability and System attributes classes to describe the non-functional attributes of an IoT device.

7 CURRENT STATE OF DEVELOPMENT

The current state of development of our research is the following:

(1) The state of the art is already extensive and have satisfied our expectations. The search for updated information will continue in the next step of the research.
(2) The ontology engineering is complete. All the required classes and properties are considered and several individuals were instantiated to verify if the ontology is decidable.
(3) Architectural components like Resource discovery, Ontology and instance manager and Healtchcheck controller are fully implemented.
(4) The instance manager and execution guidepost development is 70% completed.
(5) Both the IoT gateway and server are not deployed on the required virtual machines.
(6) Weight specifications based on the required control systems domain is not defined.
(7) The evaluation scenario and associated performance tests are not yet implemented.

REFERENCES

[1] Martin Bauer, Mathieu Boussard, Nicola Bui, Jourik De Loof, Carsten Magerkurth, Stefan Meissner, Andreas Nettsträter, Julinda Stefa, Matthias Thoma, and Joachim W. Walewski. 2013. *IoT Reference Architecture*. Springer Berlin Heidelberg, Berlin, Heidelberg, 163–211. https://doi.org/10.1007/978-3-642-40403-0_8
[2] Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Gaia Varese. 2011. Ontology and instance matching. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6050, Section 3 (2011), 167–195. https://doi.org/10.1007/978-3-642-20795-2_7
[3] Nanxi Chen, Nicolas Cardozo, and Siobhan Clarke. 2018. Goal-Driven Service Composition in Mobile and Pervasive Computing. *IEEE Transactions on Services Computing* 11, 1 (2018), 49–62. https://doi.org/10.1109/TSC.2016.2533348
[4] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web* 17 (12 2012), 25–32.
[5] Georgiana Copil, Daniel Moldovan, Hong Linh Truong, and Schahram Dustdar. 2013. SYBL: An extensible language for controlling elasticity in cloud applications. *Proceedings - 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013* (2013), 112–119. https://doi.org/10.1109/CCGrid.2013.42
[6] Francisco Curbera, Matthew Duftler, Rania Khalaf, and William Nagy. 2002. Unraveling the Communication : SOAP. *Ieee Internet Computing* April (2002), 86–93. https://doi.org/10.1109/4236.991449
[7] Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Katarzyna Szmeja Paweland Wasielewska, and Giancarlo Fortino. 2016. Tools for Ontology Matching—Practical Considerations from INTER-IoT Perspective. In *Internet and Distributed Computing Systems*, Wenfeng Li, Shawkat Ali, Gabriel Lodewijks, Giancarlo Fortino, Giuseppe Di Fatta, Zhouping Yin, Mukaddim Pathan, Antonio Guerrieri, and Qiang Wang (Eds.). Springer International Publishing, Cham, 296–307.
[8] Wen Junhao, Gu Jianan, Jiang Zhuo, and Zhu Yijiao. 2011. Semantic Web Service Selection Algorithm Based on QoS Ontology. *2011 International Joint Conference on Service Sciences* 1 (2011), 163–167. https://doi.org/10.1109/IJCSS.2011.39
[9] Muhammad Golam Kibria, Sajjad Ali, Muhammad Aslam Jarwar, Sunil Kumar, and Ilyoung Chong. 2017. Logistic model to support service modularity for the promotion of reusability in a Web Objects-Enabled IoT environment. *Sensors (Switzerland)* 17, 10 (2017). https://doi.org/10.3390/s17102180
[10] Laboratory for Analysis and Architecture of Systems (LAAS). 2015. The Eclipse OM2M project. URL: http://www.eclipse.org/om2m/.
[11] João Luiz Moreira, L.M. Daniele, Luis Ferreira Pires, Marten J. van Sinderen, Katarzyna Wasielewska, Pawel Szmeja, Wieslaw Pawlowski, Maria Ganzha, and

Marcin Paprzycki. 2017. Towards IoT platformsâĂŹ integration: Semantic Translations between W3C SSN and ETSI SAREF. SEMANTiCS conference 2017 ; Conference date: 11-09-2017 Through 14-09-2017.
[12] E. Oberortner, U. Zdun, and S. Dustdar. 2009. Tailoring a model-driven Quality-of-Service DSL for various stakeholders. In *2009 ICSE Workshop on Modeling in Software Engineering*. 20–25. https://doi.org/10.1109/MISE.2009.5069892
[13] Massimo Paolucci, Takahiro Kawamura, Terry R Payne, and Katia Sycara. 2002. Semantic Matching of Web Services Capabilities. In *The Semantic Web — ISWC 2002*, Ian Horrocks and James Hendler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 333–347.
[14] Lalit Purohit and Sandeep Kumar. 2016. Web Service Selection Using Semantic Matching. In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing (AICTC '16)*. ACM, New York, NY, USA, 16:1—-16:5. https://doi.org/10.1145/2979779.2979795
[15] Shuping Ran. 2003. A Model for Web Services Discovery with QoS. *SIGecom Exch.* 4, 1 (2003), 1–10. https://doi.org/10.1145/844357.844360
[16] Minwoo Ryu, Jaeho Kim, and Jaeseok Yun. 2015. Integrated semantics service platform for the Internet of Things: a case study of a smart office. *Sensors* 15, 1 (01 2015), 2137–2160.
[17] N Ben Seghir and O Kazar. 2017. A new framework for web service discovery in distributed environments. In *2017 First International Conference on Embedded Distributed Systems (EDiS)*. 1–6. https://doi.org/10.1109/EDIS.2017.8284046
[18] P. Shvaiko and J. Euzenat. 2013. *Ontology Matching*. http://www.ontologymatching.org/index.html
[19] Seheon Song, Sang Oh Park, Sang Il Lee, and Jae Hyun Park. 2017. Mission-oriented service development using capability-based semantic recommendation for the internet of things. *Multimedia Tools and Applications* (2017), 1–23. https://doi.org/10.1007/s11042-017-4889-1
[20] Yan Tang and Robert Meersman. 2012. DIY-CDR: an ontology-based, Do-It-Yourself component discoverer and recommender. *Personal and Ubiquitous Computing* 16, 5 (jun 2012), 581–595. https://doi.org/10.1007/s00779-011-0416-y
[21] Ming Tao, Kaoru Ota, and Mianxiong Dong. 2017. Ontology-based data semantic management and application in IoT- and cloud-enabled smart homes. *Future Generation Computer Systems* 76 (2017), 528–539. https://doi.org/10.1016/j.future.2016.11.012
[22] Vuong Xuan Tran. 2008. WS-QoSOnto: A QoS ontology for Web services. *Proceedings of the 4th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2008* (2008), 233–238. https://doi.org/10.1109/SOSE.2008.17
[23] Z Wang, R Bie, and M Zhou. 2012. Hybrid Ontology Matching for Solving the Heterogeneous Problem of the IoT. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. 1799–1804. https://doi.org/10.1109/TrustCom.2012.177
[24] World Wide Web Consortium (W3C). 2017. Semantic Sensor Network Ontology. URL: https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/.
[25] M Zhou and Y Ma. 2012. A web service discovery computational method for IOT system. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, Vol. 03. 1009–1012. https://doi.org/10.1109/CCIS.2012.6664533