Academia Journal of Scientific Research 8(3): 095-103, March 2020
DOI: 10.15413/ajsr.2019.0182
ISSN 2315-7712

Research Paper

Design qualitative metrics for semantic web of things

Accepted 25th November 2019

Rabia Afzaal* and Muhammad Shoaib

CS&E Department, UET, Lahore, Pakistan

*Corresponding author. E-mail: ram3109@gmail.com[2]shoaib@uet.edu.pk

ABSTRACT

Semantic web of things (SWoT) is an extension of IoT. SWoT makes it possible for devices to behave intelligently and hence, delivery of information and communication is achieved. It also makes it possible to elaborate the quality of text analysis process and thus provides the mean for semantic search. Design of SWoT is crucial in the delivery of services. The quality of design depends upon the measurement of defining parameters. Metrics are the tools through which this objective can be achieved. To propose design quality metrics for the semantic web of things is the major objective of this research. A suite of quality metrics will also be proposed. Metrics will be evaluated by mathematically and illustrated through case study. Security of different devices is another important aspect. Security metrics can cover this aspect.

Key words: Semantic web of things, metrics, qualitative analysis, sensors, layered architecture, measurement.

INTRODUCTION

Semantic Web of Things(SWoT) architecture was introduced by Amelie and her team (Gyrard, 2015; Gyrard et al., 2015a). This SWoT architecture is under consideration of W3C. Domain experts and Federated Interoperable Semantic IoT Testbeds and Applications (FIESTA) are working on SWoT. FIESTA-IoT[1] actually provides tools and techniques for supporting IoT based testbed/platforms operators to communicate with their services in an interoperable way based upon advanced semantics-based solutions (Gyrard and Serrano, 2018).

In recent years, the numbers of applications embedded with sensors are suitable and progressively more established. On the other hand, applications which already exist are largely specific to a domain and they are deficient in cross-domain applications. Some examples of cross-domain SWoT applications (Gyrard, 2015) include recommended food according to the weather predictions, home remedies according to health measurements, and safety apparatus in a smart car according to the weather. SWoT generators are built and perform their tasks under FIESTA-IoT[1].

Metrics are valuable object in the whole life cycle of the development. Finest metrics should be simple, specific, easily obtainable, valid and robust (Singh et al., 2012). Traditional metrics for the web as well as for the software cannot be used for the Semantic web of things (Sobral et al., 2016). There is a conventional set of qualitative and quantitative metrics that allow measuring the model's outcome (Garbarino and Holland, 2009). The web and software metrics are based on the syntactic aspect, whereas SWoT are working on the semantic aspect (Treiblmaier and Pinterits, 2010). Software metrics depend on syntactical source code and ignore the semantic aspect. Semantic metrics are more precise than syntactic ones (Helali, 2015).

Metrics are important at each layer of the end product. Quality is an uncompromised factor in any process (Ince, 1984). For each phase of development, quality metrics are significant. There are several design quality metrics that are

[1] **FIESTA-IoT[1]** (http://fiesta-iot.eu/) is funded by the European Horizon 2020 Programme with the Grant Agreement No. CNECT-ICT-643943. Thanks to the FIESTAIoT consortium for brainstorming and fruitful discussions
https://searchcio.techtarget.com/definition/999992

active to make sure that the standard of the ultimate product is at a design level. Implementation of these metrics ensures the best quality product, customer satisfaction and improved business (Ince, 1984). These existing quality metrics are syntactically analyzed, whereas quality metrics for SWoT are demand for semantically analyzes. In literature, metrics for e-health, weather, smart homes, transportation and lots of others have been proposed, but quality metrics for SWoT at the design level is still missing. Metrics for websites and softwares are evaluated on syntactical basis, but for SWoT there is a need for semantic based evaluation. Sobral et al. (2016) pointed out that a general architecture of SWoT and a complete suite of design quality metrics is required. Design quality metrics are presented in this study. Metrics are evaluated through a case study.

The case of e-health system is selected and its results are showing improvement and accuracy of the system. Metrics of each layer is revealing its importance. It has been observed that the metrics are very essential for SWoT at the design level. These metrics are performing well. Though it is a never-ending process of improvement, but more knowledge will be required about the use of these metrics because the number of devices and the use of domains can vary.

The rest of the study is structured as follows: presentation of literature survey and explanations of the different domain metrics, their limitations and shortcomings. This is followed by description of the M3 layered architecture and the metrics for each layer with their mathematical calculations. Results illustrated by the given case study are then presented. The conclusion and future work are shown in the end of this study.

LITERATURE SURVEY

A huge number of constructs and items have been used in domains such as IoT, WoT, Semantic web and SWoT (Singh et al., 2014; Achary and Shaileshbhai, 2017; AGyrard et al., 2017; Bukhari, 2017) for scholarly literature to analyse the different dimensions. Some of these constructs have been newly developed by the authors while working on SWoT but for specific domains like e-health, smart homes or e-education. Amelie worked on cross domain Semantic web of Things applications and proposed the Machine to Machine Measurement M3 architecture (Gyrard, 2015; Gyrard et al., 2015a). Many authors have worked on different metrics for different domains such as Coral worked on Classifying web metrics (Calero et al., 2005). Similarly, metrics for the websites were defined by Horst (Treiblmaier and Pinterits, 2010). Some authors worked on quality metrics for Web ontologies (Farooq et al., 2010; Vrandečić and Sure, 2007; Alani and Brewster, 2006; Steinberg et al., 2017; García et al., 2010). Metrics for Quality Assurance of Web based Applications were also defined (Zia, 2015). Sobral et al. (2016) suggested a concept of Ambient Assisted Living (AAL) which they seek more appropriate, secure for elderly people. This study is about a new routing metric that considers not only nodes and its connections, but also enhance the network performance for AAL-IOT. Savola et al. (2015) discussed in their study that their principal contribution in it was analysing the security and privacy risk for an e-Health self-care system. He proposed initial heuristics for security metrics via decomposition of security objectives. According to Misra et al. (2015), currently, different IoT architectures being used are based on the development and deployment of IoT technologies. These technologies may in the future not be met and barrier could be created for them. In this research report, the writers sought to integrate these essential problems; and kept a trail with a set of design blueprints and a possible architecture for a new IoT ecosystem (Barnaghi et al., 2012). Some authors such as Gyrard et al. (2014 and 2015b) described that its applications are becoming more popular day by day but they are not interoperable with each other. The writers suggested a framework named as Machine to Machine Measurement (M3). This M3 framework is produced along the bases of semantic web technologies to clearly identify the meaning of sensor measurements in an interconnected manner to facilitate the interpretation of sensor data and to combine domains.

Studies have mentioned that the semantic web recommendations are commonly not acknowledged by the IOT community which delays automation or reuse of domain knowledge (ontologies, datasets and rules); however initially, an ontology was considered to be easily shared and reused. According to the writers, there is a factual requirement to spread semantic web best practices in the community of IoT which can easily share and reuse domain knowledge to later interlink them in building favorable cross-domain Semantic Web of Things applications.

Julio believed that up till now there is no study as well as no formal characterization methods for the perceived quality of network interconnected actuators. They meet the qualitative user opinions against quantitative measured performance. The authors suggested the first framework in this arena, which is overlooked (Paschou et al., 2013). Some metrics and methods for efficient data transfer in IOT with specifically health domain have been addressed. The health domain is really vital and crucial. The results prove that the given methodology is efficient for different cases of Health issues. Besides, the given methodology and metrics can be easily adopted by other IOT domains easily.

KE4WoT methodology is given by authors based on k-means and word2vec based on two machine techniques. The author claims that there is no study on the significance of automatically classifying the appropriate topics for iot.schema.org. KE4WoT specifically deals with only three domains; smart homes, smart cities and it has been

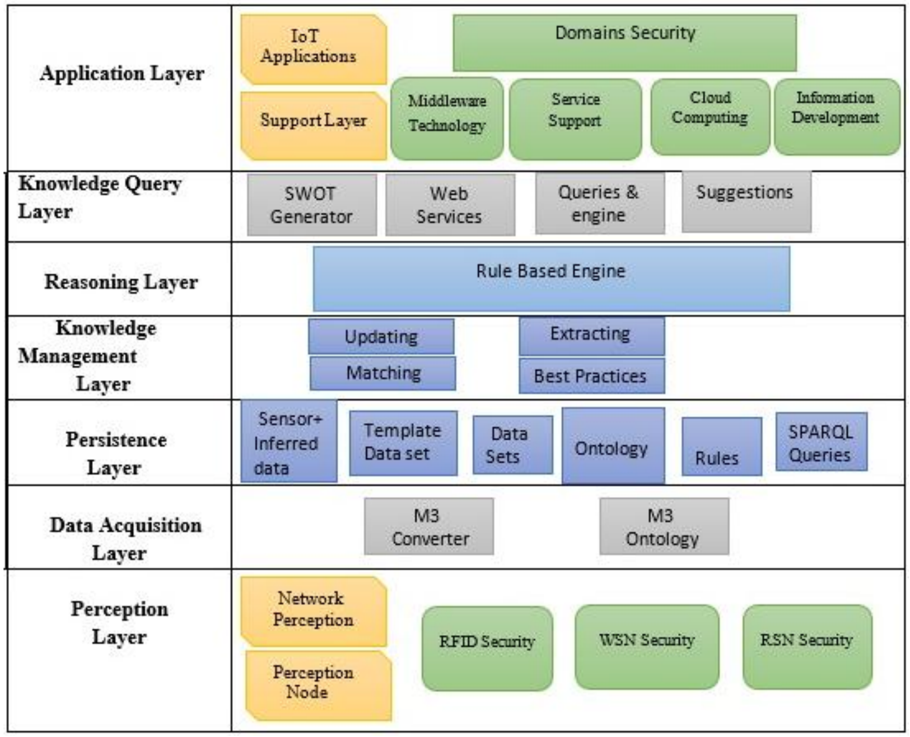

Figure 1: Layered M3 architecture.

proposed that it can be used for any other important domains such transportation etc. (Noura et al., 2019).

MATERIALS AND METHODS

M3 layered architecture as shown in Figure 1 is built in layers which are split as follows:

I. Perception Layer
II. Data acquisition Layer
III. Persistence Layer
IV. Knowledge management Layer
V. Reasoning Layer
VI. Knowledge query Layer
VII. Application Layer

Perception-layer

In the perception layer, the system aims to obtain, collect and process the data from the physical world. All the information is perceived and collected in this layer. This layer performs two tasks:

a) Collecting Data
b) Collected information converted to a digital signal for easy transmission to the next layer.

Perception layer consists of two types of devices: Sensors and actuators. The metrics that perception layer should be noted are as follow:

Recoverability

All the devices, sensors and tags used in this perception layer must be re-coverable. It should have proper back up of data but if the data are lost due to any reason, they must be regained at any cost. Recoverability depends upon the backups of data (in required time) and predefined plans and strategies.
Calculate the recoverability time by the Fault Time and Actual Time. From the above given data, it is obvious that:

$FT|AT * RE * BS$Shown in table 1

Changing this sign into equality, use a constant K, This K is the recoverability time. So, our equation will become:

$$\mathrm{RE} = \mathrm{k} * \mathrm{BS}$$

Where, BS is backup and predefined plan and strategies. For the term BS in the formula, we will use 0 and 1. If a sensor contains any predefined plan or strategy then we will use 1 otherwise 0. If there is no predefined plan, means we will not get any back up of data as there is no technique used to prevent damage of data. And recoverability will lead to zero, RE = 0. But for our given

Table 1: Abbreviations.

S/N	Symbols	Description
1.	FT	Fault Time
2.	AT	Actual Time
3.	RE	Recoverability
4.	BS	Backup System
5.	DLP	Data loss prevention
6.	In.D	Incoming Data
7.	Ls.D	Lost Data
8.	RA	Retainability
9.	Ax	Availability of the system
10.	n.s	Number of sensors attached in a system
11.	AT	Accessibility time
12.	T	Time when data was being stored (in years)
13.	V	Volume of data being stored (in GB)
14.	K	current availability of system (%availability)
15.	R.P	Reuse Percentage
16.	R.I	Total reused information or knowledge in the database
17.	T.I	Total information or knowledge saved in the database
18.	OPD	output data
19.	mOPD	mean of output data
20.	RB	Reliability
21.	E.T	Execution Time
22.	N	no. of records retrieved
23.	M	no. of RDF tables joined
24.	Ap	percentage of time that the application is operational
25.	Up	Uptime-Application working time
26.	Dw	Downtime-Application down time(not working)

scenario we have DLP technique which is based on predefined plan so we will put 1.

$$\mathrm{RE} = \mathrm{k}\,(1)$$

Now, to find recovery percentage:

$$\boldsymbol{RE = \left({}^{FT}/_{AT}\right) * 100} \quad (1)$$

Data-acquisition-layer

The layer where the source data first lands is the data acquisition layer. It represents the data in its rawest state. This layer converts the acquired data in an integrated way (RDF/XML) combined with the M3 ontology. Retain ability metrics should be on the data acquisition layer. This layer must be able to absorb all the data it acquires. Data must not be lost at this point. Layer must allow all data to be recorded, and then, using the M3 and RDF ontologies, converts and decodes the data according to its need.

Retain-ability

First, calculate the Lose data, abbreviation is shown in Table 1:'

$$\boldsymbol{T.D = In.D + Ls.D}$$
$$\boldsymbol{Ls.D = T.D - In.D}$$

Incoming data are inversely proportional to the lost data

$$\boldsymbol{In.D \propto {}^{1}/_{Ls.D}}$$

Retainability can be calculated using Ls.D:

$$\boldsymbol{RA = In.D - Ls.D} \quad (2)$$

Availability

The layer must be readily available to note the changes as measured by the e-Health sensor. Now, by calculating the availability of the system, then it can easily be measure by number of sensors. The availability is directly proportional to number of sensors, that is, increased number of sensors will enhance the availability. Availability of the system should be 100%. However, at least any 2 sensors out of the 9 available sensors from scenario should be working or cooperating with each other, abbreviations are shown in Table 1:

$$\boldsymbol{A\% = Ax - {}^{n.s}/_{2}} \quad (3)$$

Persistence-layer

The persistence layer contracts with persevering (storing and retrieving) data from a data stockpile (such as a database). Accessibility metric should be on this layer. The reason for this is to access the data easily, accurately and within less time.

Accessibility

Now, calculating the accessibility time taken by a system to retrieve the data is shown in Table 1:

$$AT = {}^{T}/_{V}$$

$$\boldsymbol{AT = {}^{T}/_{V} * log(K)} \quad (4)$$

Taking, log here for making our answers reasonable or in limits.

Knowledge-management-layer

It deals with defining, indexing, scheming, reclaiming and combining domain specific knowledge (e.g. smart home, intelligent transport system etc.). Reusability metric should be on this layer. This metric will calculate the data or already available knowledge. Reusability saves time and cost of the work being done. Now, it depends on the company as to what percent of existing data do they want to reuse.

Reusability: Abbreviations are shown in Table 1.

$$R.P \propto R.I$$

Where as $R.P = {}^{1}/_{T.I}$

$$R.P = \left({}^{R.I}/_{T.I}\right)$$

$$R.P = \left({}^{R.I}/_{T.I}\right) * 100 \quad (5)$$

Reasoning-layer

High-level knowledge is inferred in reasoning layer using reasoning engines and M3 rules which are extracted from Sensor-based Linked Open Rules. Reliability metric should be at this layer. Reliability is an important metric for this layer to show how logically correct all the work that is being done.

Reliability

$$RB = \left(\left(\sqrt{OPD} - mOPD\right)\right)^{2}/_{n.s - 1} \quad (6)$$

Values of the RB will vary from 1-9. Value from 1 is for best, while from 9 is for worst. As the value move away from 1, it means reliability of the system will go down.

Knowledge-query-layer

SPARQL is a query language that is used to retrieve and manipulate data conforming to the RDF data model. Efficiency metric of this layer will be high. This is because efficiency should be checked as SPARQL queries are checked and maintained their execution, abbreviations are shown in Table 1.

Efficiency

Time of a SPARQL query takes to execute is:

$$E.T = (n * m)$$

Big Oh (O) will be taken so that the worst execution can be calculated:

$$E.T = O(n * m) \quad (7)$$

Application-layer

The application layer services (running on smart devices) which analyses and displays the outcomes to end users. This layer should have obtainability metrics. Because obtainability means system is in a functioning condition. In application layer as discussed earlier all the services of an application will be analyzed and displayed to the users.

Obtainability

The obtainability of an application will be high if it has the percentage of time that it works (99% or above).

$$Ap = \left(Up \ / \left(Up \pm Dw\right)\right) * 100 \quad (8)$$

RESULTS AND DISCUSSION

Software architecture evaluation has many methods. By using these methods, the quality of the software can be measured. By enhancing the quality of the software, it identifies upcoming potential risks and assures the developers that their chosen architecture is behaving correct. Moreover, it also assures them that it meets both functional and non-functional quality requirements (Barnaghi et al., 2012). Several methods and techniques have been proposed for software architectural evaluation. Evaluation methods are applicable on the different stages of the software. Mainly two types of evaluation methods are applied. One is considered before implementation and the other is considered after implementation of software methods. For "before implementation software" early architecture evaluation methods are applied. Similarly, for "after implementation software" late evaluation methods are applied. M3 architecture is under consideration of W3C for Semantic web of things.

Perception-layer

Recoverability: Recoverability is important for the data. Data lost will occur at any stage. So it must be recoverable and in our given case study it will be high. So, the fault time is half of actual time. Data can be recoverable otherwise the data will be lost and have to do work again.

Data-acquisition-layer

Retain ability: Out of the total data acquired, the lost data are only 1%, hence, the percentage retain ability is 99%. Therefore, we must keep a check on the lost data in order to enhance our retain ability factor as the data acquisition layer works best in high retainable environment. The data are going to lose again and again. The working will stop at that point, so the data will recover. If it is not like that, data recovery will create problems.

Availability: So, availability of the system is directly proportional to number of sensors. The number of sensors will increase availability of the system.

Persistence-layer

Accessibility: The accessibility of the system is dependent on the availability of the system. The range of the systems is of three types as follows:

- Special system
- Shared component system
- Clustered system

In our scenario, system is of special type and very sensitive in its use, so the availability time in our metrics will give us the good accessibility time. It will satisfy our metrics and enhance its performance.

Knowledge-management-layer

Reusability: Our metrics gives its percentage according to the company requirements. It varies with the working of the different domains and their collaborations. Different domains and their data can vary according to the system's need. Similarly, data to be used in different domains are dependent on the requirements.

Reasoning-layer

Reliability

Values of the R will vary from 1-9
1 is for best and 9 is for worst
As the value move away from 1, it means reliability of the system will go down.

Knowledge-query-layer

Efficiency: The efficiency of the system depends upon the rows, columns and their joints. Our efficiency metric will have higher efficacy if there will be less joints in rows and columns and vice versa.

Application-layer

Obtainability: As discussed earlier, obtainability of our system will be high. In this way, we may say that our system performance will enhance and work smoothly.

Case study is defined as below to show its overall evaluation of those metrics.

CASE STUDY

The e-Health device Platform provides the most effective example of the uses of devices and sensors once they are engaged. Before starting our work to outline and make a case for the metrics of the mentioned sensors, let us have a bird's eye view on the general working of system. Take the e-Health sensor platform's scenario for our metrics discussion. e-Health Sensor guard/shield allows Arduino and Raspberry Pi users to achieve biometric and medical requests. The body intensive care is monitored by nine different sensors:

Pulse, element of oxygen in blood (SPO2)
Flow of air (Breathing)
Body temperature
Electrocardiogram (ECG)
Blood sugar level (Glucometer)
Galvanic skin response (GSR - sweating)
Blood pressure level (Sphygmomanometer)
Patient position (Accelerometer)
Electromyography Sensor (EMG)

For instance, taking any one sensor, such as Accelerometer into consideration: This sensor is used to indicate patient's body position. Patient's body position indicates many diseases which need to be treated timely. Now if we talk about perception layer in this sensor, it is the collection of physical data from the environment. The core purpose of this layer is to acquire, collect and process data. So, this layer must hold the metrics of recoverability. All the devices, sensors and tags used in this layer must be recoverable. It should have proper back up of our data. If due to any reason our data are damaged, it must be able to recover that data. It should have proper voltage supply, for this purpose it should contain amplifiers because amplifiers are good at managing voltages. Our data should be retained at any cost.

The layer where the source data first lands is the data acquisition layer. It represents the data in its rawest state. They all either monitor the state of a patient or extract sensitive data to be subsequently analysed for medical diagnosis. All the Nine sensors in the e-health platform holds the data on data acquisition layer. These sensors sense the change in health and record the data to the platform as soon as it is read by them. Readings from our health measuring devices and acquired data are stored in the computing devices. These health measuring devices perform the basic task of data acquisition in our scenario. From the knowledge of the working of data acquisition layer, it is inferred that the layer must possess the retain ability metrics under all conditions. The layer must be able to absorb all the data it acquires. It must not lose the data. Layer must allow all data to be recorded to it, and then, using the M3 and RDF ontologies, converts and decodes the data according to its need. Here, it must note all readings. The layer must keep a track on the previous records as well. All data, blood pressure, heartbeat etc. should be acquired at regular intervals, and cannot miss any reading as this could make our e-health sensor work inefficiently. Availability metric is measured at this layer. The layer must be readily available to note the changes as measured by the e-Health sensor. When faced with any delays while acquiring the readings, this may badly affect the patient's health. So, all readings must be spontaneously acquired as soon as they are read by the sensors. It must be available all the time to respond to the environment.

The persistence layer deals with uninterrupted (storing and retrieving) information from a knowledge store (such as a database). Accessibility metric should have been measured at this layer. In a hospital, there is a system that stores all the database of the patients for past 50 years. The triple story database stores all the ontologies in the persistence layer. Now a patient with a past medical history in the same hospital visits. But he visited 45 years back. The doctors want an immediate report of his past medical history to further treat him.

Knowledge management layer is liable for result, indexing, designing, reusing and combining domain specific knowledge (e.g. smart home, intelligent transport system etc.). Reusability metric is to be measured for this layer. It is the most important metric in knowledge management. It means to reuse the already available knowledge and data. Reusability saves time and cost of the work being done. But not everything is reusable, so special care is required to decide which knowledge is further usable and which is not. Suppose a hospital has a database in which all the information about its patients and employees, its payments, induction dates, departments, orders and all of its products are saved. With the passage of time, the hospital feels the need to make a new database which is more advanced. Now, the hospital starts gathering all the data from scratch for the new database which will probably take ages to complete. Instead reusing the already available information for the database will not only save time, but also it will reduce the budget because gathering information itself is a costly process.

High-level knowledge is inferred in reasoning layer using reasoning engines and M3 rules that are extracted from Sensor-based Linked Open Rules. Unit of measurement M3 rules, a group of rules, will work with the M3 philosophy to infer new data on device information. Sensor-based Linked Open Rules (S-LOR) have been designed to help build cross domain IoT applications based on Linked Open Data and Linked Open Vocabularies. Correctness metric should be measured at this stage. As in Facebook, if a person has liked pages of hospitals and diseases, we can deduce that this person is concerned with health topics. Now, what Facebook does is that it starts giving suggestions to that person to like other pages regarding health care in the column of "suggested pages". In this case, Facebook has logically concluded that the person is a fan and he may be interested in other pages related to health care.

The efficiency of knowledge query layer is dependent on the type of SPARQL query being executed. The lesser the number of records and RDF tables used, the more would be the efficiency. This means if information is retrieved from one domain, efficiency will be much higher as compared with the case when information is retrieved from heterogeneous domains.

The application layer is employed on Nursing application (running on sensible devices) that parses the information and displays the results to end users. At this stage, the information will be provided in visual form for a user who will really perceive, instead of binary zeroes and ones. Obtainability is the proportion of time when a system is in a functioning condition. In e-health, availability is important for achieving continuity of electronic healthcare. To ensure the best healthcare services, it should be up all the time. If the network is down, the application cannot work properly. Due to this, the healthcare providers cannot proceed with the patient's data and hence, recommendations cannot be given.

CONCLUSION

The aim of this investigation is to identify quality metrics for the Semantic Web of Things (SWoT). These quality metrics are used to design Semantic Web of Things (SWoT) platform, to set-up and manipulate connections between physical devices and the web. This is also to produce metrics for each layer, such as recoverability, retainability, accessibility, reusability and so on, to satisfy the architecture. These metrics will not only enhance the architecture but also improve its quality. The architecture we have got is by no means a final stopping point. M3 architecture is still on-going procedure having vast ideas for SWoT. On the other hand, it will not be wrong to say that we have got an overall M3 architecture which might be offered by its developers.

Quality metrics are mathematically proved and evaluated on the basis of SAAM as M3 architecture is not implemented architecture. We believe that this is a step to move forward. These quality metrics actually enhance the performance and capacity of Semantic Web of Things (SWoT). These quality metrics can be applied to Semantic Web of Things in order to predict flaws and improve the quality of SWoT. Regarding these metrics, we are not claiming that this is the end of this study, work is still going on. Consequently, the results of the quality metrics can be seen as a chance for improvement in architecture. The provided case study shows better result for the architecture.

FUTURE WORK

Future study still need to be conducted on M3 design, and this might enhance its layers by users' or developer's perspective. M3 architecture is not an end of the story, a lot more study is yet to be conducted. In the future, researchers will make use of already given appropriate metrics for every layer. Scholars can improve the architecture and modify it in line with their research and need. Thus, additional study on architecture still needs to be performed in order to make it more accurate and precise.

REFERENCES

Achary R, Shaileshbhai J (2017). "Internet of Things: Essential Technology, Application Domain, Privacy and Security Challenges," Int. J. Comput. Appl. 157(6): 13–21.

Alani H, Brewster C (2006). "Metrics for Ranking Ontologies. and Information, Science; Lytras MD, Ordonez De Pablos P, Ziderman, A., Roulstone, A., Maurer, H., Imber, J.B., Eds.; Springer: Berlin/Heidelberg, Germany. 111.

Barnaghi P, Wang W, Henson C and Taylor K, "Semantics for the Internet of Things: early progress and back to the future," p. 22.

Calero C, Ruiz J, Piattini M (2005). "Classifying web metrics using the web quality model," Online Inf. Rev. 29(3): 227–248.

Demeyer S (2011). "Research methods in computer science," in 2011 27th IEEE International Conference on Software Maintenance (ICSM), Williamsburg, VA, USA. pp. 600–600.

Farooq A, Ahsan S, Shah A (2010). "Web-Ontology Design Quality Metrics, J. Am. Sci. 6(11): 52–58.

Garbarino S, Holland J (2009). University of Birmingham, and Governance and Social Development Resource Centre, Quantitative and qualitative methods in impact evaluation and measuring results. Birmingham? GSDRC.

García J, García-Peñalvo FJ, Therón R (2010). "A Survey on Ontology Metrics," in Knowledge Management, Information Systems, E-Learning, and Sustainability Research WSKS; Communications in Computer

Gyrard A "Designing cross-domain semantic Web of things applications," p. 232.

Gyrard A, Bonnet C, Boudaoud K (2014). "Enrich machine-to-machine data with semantic web technologies for cross-domain applications," in 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea (South). pp. 559–564.

Gyrard A, Datta SK, Bonnet C, Boudaoud K (2015). "Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation," in 2015 3rd International Conference on Future Internet of Things and Cloud, Rome, Italy. pp. 9–16.

Gyrard A, Datta SK, Bonnet C, Boudaoud K (2015b). "A Semantic Engine for Internet of Things: Cloud, Mobile Devices and Gateways," in 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Santa Cantarina, Brazil. pp. 336–341.

Gyrard A, Patel P, Datta SK, Ali MI (2017). "Semantic Web Meets Internet of Things and Web of Things: [2nd Edition]," in Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion, Perth, Australia. pp. 917–920.

Gyrard A, Serrano M "FIESTA-IoT: Federated Interoperable Semantic Internet of Things (IoT) Testbeds and Applications," p. 2.

Helali RGM (2015). "Software semantic metrics: A Survey," IJARCCE. pp. 380–383.

Ince DD (1984). "The evaluation of software design metrics," Des. Stud. 5(4): 239–247.

Metrics for Unit Ontologies," in OWL: Experiences and Directions – Reasoner Evaluation, vol. 10161, M. Dragoni, M. Poveda-Villalón, and E. Jimenez-Ruiz, Eds. Cham: Springer International Publishing, pp. 40–54.

Misra P, Simmhan Y and Warrior J (2015). "Towards a Practical Architecture for the Next Generation Internet of Things.

National Institute of Electronics and Information Technology(NIELIT), Srinagar, Jammu and Kashmir, India, S. N. Bukhari, J. A. Mir, CSE Department, SSM College of Engineering and Technology, Jammu and Kashmir, India, U. Ahmad, and CSE Department, SSM College of Engineering and Technology, Jammu and Kashmir, India, "Study and Review of Recent Trends in Semantic Web," Int. J. Adv. Res. Comput. Sci.

Softw. Eng. 7(6): 304–310.
Noura M, Gyrard A, Heil S and Gaedke M (2019). “Automatic Knowledge Extraction to build Semantic Web of Things Applications,” IEEE Internet Things J. pp. 1–1,
Paschou M, Sakkopoulos E, Sourla E and Tsakalidis A (2013). “Health Internet of Things: Metrics and methods for efficient data transfer,” Simul. Model. Pract. Theor. 34: 186–199.
Runeson P, Höst M (2009). “Guidelines for conducting and reporting case study research in software engineering,” Empir. Softw. Eng. 14(2): 131–164.
Savola RM, Savolainen P, Evesti A, Abie H and Sihvonen M (2015). “Risk-driven security metrics development for an e-health IoT application,” in 2015 Information Security for South Africa (ISSA), Johannesburg, South Africa. pp. 1–6.
Singh D, Tripathi G, Jara AJ (2014). “A survey of Internet-of-Things: Future vision, architecture, challenges and services,” in 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea (South). pp. 287–292.
Singh M, Mittal A, Kumar S (2012). “Survey on Impact of Software Metrics on Software Quality,” Int. J. Adv. Comput. Sci. Appl. 3:1.
Sobral JVV, Rodrigues JJPC, Saleem K, de Paz JF, Corchado JM (2016). “A composite routing metric for wireless sensor networks in AAL-IoT,” in 2016 9th IFIP Wireless and Mobile Networking Conference (WMNC), Colmar, France. pp. 168–173.
Steinberg MD, Schindler S, Keil JM (2017). “Use Cases and Suitability
Treiblmaier H, Pinterits.A (2010). “Developing Metrics for web sites,” J. Comput. Inf. Syst. p 10.
Vrandečić D, Sure Y (2007). “How to Design Better Ontology Metrics,” in The Semantic Web: Research and Applications, vol. 4519, E. Franconi, M. Kifer, and W. May, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 311–325.

Cite this article as:

Afzaal R, Shoaib M (2020). Design qualitative metrics for semantic web of things. Acad. J. Sci. Res. 8(3): 95-103.