Web of Things Semantic Functionality Distance

Maria Ines Robles
Aalto University. Espoo, Finland
maria.robles@aalto.fi

Bilhanan Silverajan
Tampere University. Tampere, Finland
bilhanan.silverajan@tuni.fi

Nanjangud C. Narendra
Ericsson Research. Bangalore, India
nanjangud.narendra@ericsson.com

***Abstract*—The Web of Things is an architectural initiative proposed by the World Wide Web Consortium, to enable Internet of Things devices to interact through Web standards. One of the cornerstones of the architecture is a Thing Description, which is an object model that exposes devices to the Internet through a common interface composed by properties, actions and events. In this paper, we evaluate the similarity level on capabilities calculated for Web of Things objects. We developed, based on the Thing Description, a metric called Web of Things Semantic Functionality Distance (WoTSFD). The semantic functionality distance is a measure of the device ability to perform the same function in a specific application context. We evaluate this metric in a smart home environment. The results show that different devices can be detected to be similar, thus suitable to collaborate or be replaced by each other to perform a specific task in a determined use case.**

***Index Terms*—Web of Things, Semantic Distance, IoT, Management.**

I. Introduction

The Web of Things architecture [1] aims to empower interoperability between Internet of Things (IoT) devices, independently of their underlying platform, to facilitate the implementation of IoT behavior through web standards. One of the cornerstones of the architecture is the WoT Thing Description (TD). The TD is a structural representation composed of Properties, Events and Actions.

One of the greatest challenges in networks composed of heterogeneous resource constrained IoT devices, is the large scale coordination and collaboration amongst these devices [2]. While there is existing work in defining IoT devices so that they are discoverable in terms of their names or properties, such as the CoRE Resource Directory [1], allowing the discovery of semantically related devices so that they can be adapted to a primary or secondary context, is not well explored. In this paper, we tackle this issue, by proposing and evaluating a metric called the Web of Thing Semantic Functionality Distance (WoTSFD), We denote semantic functionality distance as a measure of *the capability of two Web of Things devices to perform the same function*. In our previous work [3], we proposed a metric that calculates the semantic distance between devices in a context of IoT device management. The work considered only the device properties, but did not take into account device actions, events nor their Web context. WoTSFD, proposed here, overcomes this. In addition this paper proposes a mechanism to detect dissimilar devices independently of the application context, when, for example, we want to identify critical devices from the non-critical ones.

The rest of the paper is organized as follows: Section II describes the related work. Section III describes the WoTSFD metric. Section IV describes the evaluation of WoTSFD. Section V depicts the methodology to calculate the dissimilarities between devices, and Section VI comprises the Conclusion and Future Work.

II. Related Work

Semantic similarities is applied extensively in different contexts, for example between Web resources for Linked Data in [5], [6] and [7]. Semantic distance applied to Information Retrieval are described in [13] and in [14]. In [12] the semantics is applied to IoT management, and for pervasive computing in [15]. Improvements of the precision ratio of the web service discovery process is described in [8]. A web service discovery scheme based on semantic similarity using Hungarian algorithm is applied in [9]. In [10], a multiple semantic fusion model, for service matching, getting high correlation with end-user requirements is evaluated. A web service discovery on goal-based queries that optimizes the performance in the discovery process is described in [11]. In our paper, we provide relevance level to the device features (properties, actions and events) and based on this we calculate the semantic similarities. Our methodology assigns relevance level to the device features calculated based on context requirements.

III. Web of Things Semantic Functionality Distance

We define the Web of Things Semantic Functionality Distance (WoTSFD) to be a metric, that detects functional similarities in two or more disparate devices, in order that these devices can be used to collaborate in a specific task or be grouped to be managed together. Because WoTSFD is based on the WoT Thing Description, it enables interactions among devices from different vendors without any modifications of the underlying device implementations. The WoTSFD contains a calculated value in the range [0,1]. Its exact value is based on the following features: 1) Devices with semantically equal functionality have distance of 0. 2) Devices that are semantically close in functionality have distance close to 0. 3) Devices that are semantically far in functionality have a distance close to 1. 4) Devices with semantically unrelated functionality have a distance of 1. 5) WoTSFD is symmetric:

[1]https://tools.ietf.org/html/rfc6690

The distance between device A and device B is the same as that between device B and device A. and 6) The method should be able to cope in any type of environment and with all devices. To fulfill the previous requirements we describe the following concepts:

1) WoT Application Context Requirements (WACoRq): These requirements have the function to fulfill one or more goals, e.g., *I have a goal to group devices that are able to alert about a fire*. Namely, these requirements help to identify the relevance of a property, action or event to fulfill the goal, such as a device with a dimmable feature allows to alert in case of fire changing the color of the device. In this scenario, a property that indicates if the device has a *dimmable* feature has more relevance than the property that defines the *name* of the device. We define a WACoRq as a *Thing Description*, thus as a set of *property* (pr), *action*(ac) and *event*(ev). $WACoRq = \{\{pr_1, ..., pr_n\}, \{ac_1, ..., ac_m\}, \{ev_1, ..., ev_z\}\}$. We denote the total number of properties with *n*, the total number of actions with *m* and the total number of events with *z*.

2) Weights assigned to properties (w_{pr}), actions (w_{ac}) and events (w_{ev}) of a Thing Description: The weight indicates the relevance of the property, action or event to fulfill a goal, namely the WACoRq. The values close to 1 indicate that the feature (property, action or event) is relevant to fulfill the contextual requirement and values close to 0 indicate that the relevance is minimal. For example, a *dimmable* property is going to have w_{pr} = 0.95, since it is relevant to alert in case of fire. On the other hand, the *name* of the device does not provide useful information since it can be an arbitrary name. Thus the weight should not be high, and it could be assigned a value of w_{pr} = 0.10.

3) Direct Links between properties, actions or events that exists between two Thing Descriptions: If two Thing Descriptions present the same property, it implies that a Direct Link (DL) exists between them. If a smart TV object and a light bulb both have a *dimmable* property, it means that a DL exists between the smart TV and the light bulb. Thus, the value of the DL is the weight assigned to a property (w_{pr}). The DL values are also in the interval between [0,1]. If two Thing Descriptions present the same action, it implies that a Direct Link (DL) exists between them. If a smart TV and a smart alarm clock both have an action of *increase volume level*, it means that a DL exists between the smart TV and the light bulb. Thus, the value of the DL is the weight assigned to the action (w_{ac}). If two Thing Descriptions present the same event, it implies that a Direct Link (DL) exists between them. If a smart cup object and a thermostat both have a *battery low* event, it means that a DL exists between the smart cup and the thermostat. Thus, the value of the DL is the weight assigned to the event (w_{ev}). The DL values are also in the interval between [0,1].

4) The Semantic Distance between properties, actions or events of two Thing Descriptions (SD_{pr}): The SD_{pr} method calculates the relatedness between properties of two Thing Descriptions and it is defined as follows: $SD_{pr}(TD_a, TD_b) = \frac{n - dl_{pr}}{(1+SDL_{pr}(TD_a,TD_b))\times n}$, where dl_{pr} indicates the total number of direct link between properties, and $SDL_{pr}(TD_a, TD_b)$ indicates the Sum of all Direct Links between properties of two Thing Descriptions (TD_a and TD_b).

Analogously for actions, the SD_{ac} method calculates the relatedness between actions of two Thing Descriptions, and it is defined as follows: $SD_{ac}(TD_a, TD_b) = \frac{m - dl_{ac}}{(1+SDL_{ac}(TD_a,TD_b))\times m}$, where dl_{ac} indicates the total number of direct link between actions, and $SDL_{ac}(TD_a, TD_b)$ indicates the Sum of all Direct Links between actions of two Thing Descriptions (TD_a and TD_b).

Analogously for events, the SD_{ev} method calculates the relatedness between events of two Thing Descriptions, and it is defined as follows: $SD_{ev}(TD_a, TD_b) = \frac{z - dl_{ev}}{(1+SDL_{ev}(TD_a,TD_b))\times z}$, where dl_{ev} indicates the total number of direct link between events, and $SDL_{ev}(TD_a, TD_b)$ indicates the Sum of all Direct Links between events of two Thing Descriptions (TD_a and TD_b).

5) The Web of Things Semantic Functionality Distance (WoTSFD) between two Thing Descriptions: The WoTSFD is the sum of the Semantic Distance evaluated for properties (SD_{pr}), actions (SD_{ac}) and events (SD_{ev}) divided by 3.

It is also worthwhile to highlight two points: 1) Related to the relevance to accomplish the WACoRq: the weights for properties (w_{pr}), actions (w_{ac}) and events (w_{ev}) take values between [0,1]. Values close to 0 indicate that the properties/actions/events do not sufficiently achieve WACoRq goals, Conversely, values close to 1 indicate that the properties/actions/events fulfill the WACoRq goals quite well. 2) Related to the functionality distance (WoTSFD) between WoT objects: *WoTSFD* takes also values between [0,1]. However, here values close to 0 indicate that the WoT objects are quite similar, while values close to 1 imply that they do not exhibit any considerable relatedness.

IV. Evaluation

The evaluation of WoTSFD is performed using a smart home scenario employing the continuous transmission and playback of a multimedia message, such as a simple audiovisual clip, via different smart home devices as the user moves from room to room in an apartment. We term the activity of such a continuous audio visual message, playing and resuming in different devices, as *message reproduction*.

As a concrete smart home use case scenario, we have several smart devices located in different rooms. These devices are: 1) In Room 1: one smart camera (C), one smart treadmill (T) and one smart Air Purifier (P). 2) In Room2: one smart TV (TV), one smart smoke detector (SD), one smart cap (Ca). 3) In Room3: one smart thermostat (Te), one smart fridge (F), one smart cup (Cu) and one smart light bulb (LB). For simplicity, we assume every room is equipped with a motion sensor which detects the presence of a person and communicates it to the rest of the devices located in the same room.

Each of the smart devices in the smart home can be perceived to be a WoT object or Thing. They behave as servers and expose their *Thing Descriptions* (properties, actions and

events) through a REST-based API. The client, in this case, is a management entity that requests device-specific TDs, to calculate the WoTSFD and determine which devices are close enough to fulfill a specific task. The client can be installed in any device in charge of device management, such as a gateway.

A sequence interaction between the Things (server) and a management entity (client) comprises the follows: The user provides the context requirements (as an OWL file) to the management entity. A client armed with this information then calculates which properties, actions and events are relevant for this use case. At some point, when a device joins the network and exposes its TD over HTTP, the client retrieves the TD using a GET request. Then, the client with that information, calculates the WoTSFD. Subsequently when a required task needs to be performed, the client ascertains which devices are recommended for that task and selects the most appropriate one.

We implemented WoTSFD using a Web Thing Server[2] python implementation, having *HTTP* as the web transfer protocol. Each device is described as a *Thing* composed by a *Thing Description* (set of *properties*, *actions* and *events*). The *Thing* (device) is added to a server, thus we have a *Web Thing Server* where the device exposes its *properties, actions and events* to the environment. In our implementation, the Thing Description is serialized in JSON format. Then, we have a *Client*, e.g. a manager entity implemented in a gateway, that consumes the available *Thing Description* to calculate the WoTSFD.

The TD (Properties (P), Actions (A) and Events (E)) of the smart objects are: C={P={frequency, ImgMem, imgsensor, detectRange}, A={takePic, startRecord, stopRecord}, E={overheat, memFull, battLow}}; T={P={txtDisplay, clip, video}, A={reproVid, reproAud, displayTxt}, E={videoNotAvail, memFull, audioNotAvail}}; P={P={filter, quality, fanSpeed}, A={actFil, setAir, SetFan}, E={filterObstr, fanBrkn, battLow}}; TV={P={video, dimmer, audio, txtDisplay}, A={reprodVid,increAud, pauseV, stopV}, E={videoNotAvail, audioNotAvail}}; SD={P={Co2Det, clip, heatDet}, A={detCo2, triggerAlrm}, E={reset, battLow}}; Ca={P={clip, buzzer}, A={reprAud, pauseAud, stopAud}, E={buzzertrigurd, audioNotAvail}}; Te={P={temp, txtDisplay}, A={takeTmp, displayTmp}, , E={overheat, battLow, rest}}; F={P={video,audio, txtDisplay}, A={reprodAudio, pauseAud, stopVd}, E={videoNotAvail, audioNotAvail}}; Cu={P={temp, onOff}, A={setTmp, setSleep}, E={overheat, battLow}} and LB={P={dimmer, onOff, colour, unit}, A={toggle, turnOn,turnOff}, E={overheat, dimmerfailed}}. In this use case, we suppose also that the devices with a WoTSFD value lower or equal to 0.5 are appropriate to perform the task of message reproduction.

A. Application Context

To determine which device features are relevant to fulfill a goal in a specific use case, we use the SENACT [4] methodology, which objectively determines the relevance level assigned to the properties, actions and events to calculate the semantic distance metric. SENACT is implemented in python3 and located in the Manager Entity (Client). SENACT methodology utilizes a set of assumptions, that are translated into hierarchies of features. Namely, SENACT consists of the following steps: 1) An user describes the requirements of an application context (use case) through a set of assumptions based on sensing and actuation concepts. Then, the user represents the assumptions into a set of classes and subclasses (hierarchies) in RDF/OWL data model format. 2) SENACT, with this information assigns values to the features hierarchies. 3) SENACT maps those values to the Thing Description of an application context.

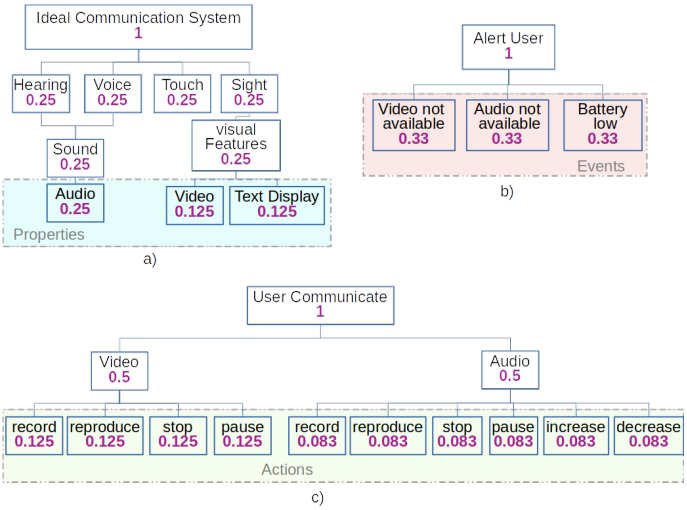

Fig. 1: Hierarchies that represent the application context of the message reproduction use case.

The set of assumptions (a) for this evaluation are: a1) An ideal communication system reaches the senses of hearing, sight, voice and touch of a human being. a2) The sense of hearing is reached by the human being through the sound. a3) The sense of sight is reached by the human being through the visual features. a4) The voice is covered by the human being through speaking features. a5) The touch sense is reached by the human being through contact with materials. a6) Sound is produced by audio actuator. a7) Visual features includes light and text display actuators. a8) An user communicate mostly through audio and video. a9) A video can be recorded, reproduced, paused or stopped. a10) An audio can be increased, decreased, reproduced, paused or stopped. a11) In case of audio, video not available or battery low alert to the user. Then, the assumptions are represented into hierarchies (Fig. 1). For this use case, the assumptions [a1-a7] are represented in Fig. 1.a, the assumptions [a8-a10] in Fig. 1.b and in Fig. 1.c the assumption [a11]. The hierarchies are given by the user in the RDF/OWL format.

The next step is to assign the relevance level to the hierarchies features. SENACT assigns the 100% of the relevance to the top, which is represented by 1 and then, SENACT divides it

[2]https://github.com/mozilla-iot/webthing-python

TABLE I: The application context for the message reproduction use case represented by a Thing Description with the relevance value (Rel. Level) for the properties, actions and events.

Thing Description of the Application Context (WACoRq)			
Properties	Rel. Level w_{pr}	Events	Rel. Level w_{ev}
audio	0.75	video-not-avail	0.33
video	0.625	audio-not-avail	0.33
text display	0.125	battery low	0.33
Actions			
	Rel. Level w_{ac}		Rel. Level w_{ac}
reproduce audio	0.083	stop video	0.125
reproduce video	0.125	pause audio	0.083
record audio	0.083	pause video	0.125
record video	0.125	increase audio	0.083
stop audio	0.083	decrease audio	0.083

proportionally downwards. The relevance level for each feature is the sum of the value for that feature in all the hierarchies. e.g. the relevance level for *Audio* is 0.25+0.5=0.75. SENACT calculates the relevance level for the features that represent *properties, actions or events*. The result is showed in Table I. We observe that the application context (WACoRq) is represented by a Thing Description, with the properties, actions and events that are relevant in the message reproduction use case.

In this evaluation, we are calculating the WoTSFD between two TDs, the TD of the device and the TD of the context (WACoRq), since we want to know how similar are the devices to the context of message reproduction. The WoTSFD values for all the devices in the use case is shown in the figure 2. We observe that the devices suitable for reproducing a message have a WoTSFD value lower than 0.5 (0.5 is indicated as a horizontal dashed line in the figure). They are the T, TV, Ca and the F. Thus, for example, if the user is in room 1, the treadmill is the most appropriate device to reproduce the message. If the user moves to room 2 and the action is detected by the motion sensor, a notification to the management entity can subsequently result in the selection of the TV to reproduce the message, as it has the lowest WoTSFD value in that room. Subsequently when the user moves to room 3, the management system can select the fridge as the appropriate device to reproduce a message to the user.

V. Calculating the Semantic dissimilarities

In some scenarios is useful to identify dissimilar devices, such as instances in which we have critical devices and we want to identify non-critical ones e.g. to perform a firmware update in a specific time frame. To identify dissimilar devices we can mention two approaches: one approach takes the application context in consideration, while the other approach

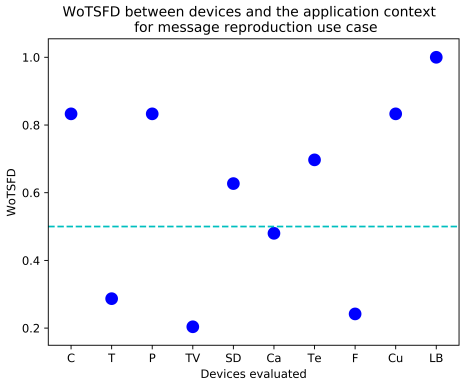

Fig. 2: WoTSFD evaluated between the devices and the application context. The horizontal dashed line represents WoTSFD=0.5.

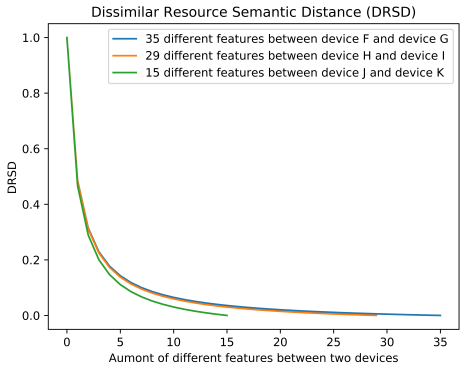

Fig. 3: Dissimilar Resources Semantic Distance evaluated between two devices with different amount of distinct features.

is independent of the context: 1) Application context related: The semantic distance method, WoTSFD, detects those devices that are dissimilar in the scope of one specific application context. e.g. devices with a value of WoTSFD close to 1 for the message reproduction use case. 2) Application context independent: We propose an algorithm to calculate the level of dissimilarities between devices, analyzing their resources independently of a context. We call this method Dissimilar Resources Semantic Distance (DRSD).

The Dissimilar Resources Semantic Distance (DRSD) metric between two devices A and B is defined as follows: $DRSD_{ab} = \frac{tt_r - m_{ab}}{(1+m_{ab}) \times tt_r}$, where, tt_r is the total of number of features (properties, actions or events) presents in both devices, and m_{ab} is the amount of features that are different between both devices. This metric takes values between [0,1], where 0 means that the devices are quite dissimilar and if the metrics takes value close to 1 means that the devices are identical.

As graphic exemplification, the figure 3 shows DRSD between two devices with different amounts of distinct features. We suppose that we have devices F and G with 35 different features between them, devices H and I with 29 distinct features and devices J and K with 15 different features. We

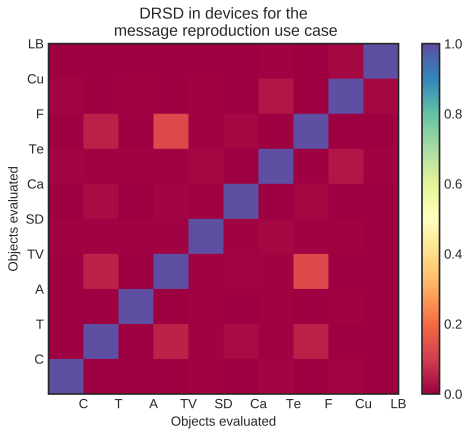

Fig. 4: DRSD for the use case under evaluation.

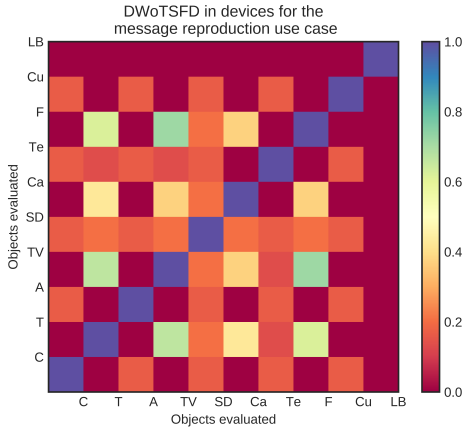

Fig. 5: DWoTSFD for the use case under evaluation.

observe that in all cases, when there are no different features, then DRSD equals 1, indicating that the devices are quite similar. Otherwise, when all features are different, then DRSD equals 0, indicating that the appliances are quite different. We evaluate DRSD for the message reproduction use case. To be able to compare WoTSFD with DRSD with the same meaning of values range, we denote: $DWoTSFD_{AB} = 1 - WoTSFD$, as a metric that show the dissimilarities between two devices A and B. As DRSD, DWoTSFD takes values between [0,1], where 0 means that the devices are quite dissimilar, and values close to 1 indicate that the devices are quite similar. Figure 4 shows the DRSD for the message reproduction use case. We observe that all the appliances are quite dissimilar (DRSD values close to zero). On the other hand, when we take in consideration the application context (Table I), we get more accuracy results as showed in Fig. 5.

VI. Conclusion and Future Work

Through the proposed metric called the Web of Things Semantic Functionality distance WoTSFD, we calculate the usefulness of a device to support functions of a given use case, with a proposed web standard interface. In the calculation, the number of features (properties, actions and events) plays a fundamental role to indicate the relevance level of the device for a specific context. If we want to emphasize more relevance e.g. to actions rather than to properties and events, the user should provide a higher amount of assumptions related to it. We propose a methodology to calculate when devices are dissimilar, and from the results, we observe that the evaluation is more accurate when the metric is calculated considering the application context (DWoTSFD). DRSD is less exact than DWoTSFD, but can be used when the application context is not known. As future work, we are working on evaluating WoTSFD in large scale networks.

Acknowledgment

This work is funded by a Tekniikan Edistämissäätiön (TES) Grant and the DIMECC Design4Value Programme. The authors wish to thank Jörg Ott for his feedback.

References

[1] Web of Things (WoT) Architecture, https://w3c.github.io/wot-architecture/. Last accessed 7 August 2018.
[2] Brush, A. J., et al. "Home automation in the wild: challenges and opportunities." proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, (2011).
[3] Robles, Maria Ines, et al."Measuring Semantic Distance between LWM2M Resources." Internet of Things (iThings), 2017 IEEE International Conference, (2017).
[4] Robles, Maria Ines, et al. "Calculating LWM2M resource semantic distance through SENACT ontology." Proceedings of the Seventh International Conference on the Internet of Things. ACM, (2017).
[5] Passant, Alexandre. "Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations." AAAI spring symposium: linked data meets artificial intelligence. Vol. 77. 2010.
[6] Piao, Guangyuan, and John G. Breslin. "Measuring semantic distance for linked open data-enabled recommender systems." Proceedings of the 31st Annual ACM Symposium on Applied Computing. ACM, 2016.
[7] Meymandpour, Rouzbeh, and Joseph G. Davis. "A semantic similarity measure for linked data: An information content-based approach." Knowledge-Based Systems 109 (2016): 276-293.
[8] Wang, Gongzhen, et al. "A semantic match algorithm for web services based on improved semantic distance." Next Generation Web Services Practices, . NWESP'08. 4th International Conference on IEEE.(2008)
[9] Khanam, Shirin Akther, and Hee Yong Youn. "A Web Service Discovery Scheme Based on Structural and Semantic Similarity." J. Inf. Sci. Eng. 32.1: 153-176.(2016)
[10] Lu, Wei, et al. "Joint semantic similarity assessment with raw corpus and structured ontology for semantic-oriented service discovery." Personal and Ubiquitous Computing 20.3: 311-323.(2016)
[11] Zhang, Neng, et al. "Web service discovery based on goal-oriented query expansion." Journal of Systems and Software 142 (2018): 73-91.(2018)
[12] Thoma, Matthias, et al. "Managing things and services with semantics: A survey." Network Operations and Management Symposium (NOMS), 2014 IEEE. IEEE, (2014).
[13] Kathuria, Mamta, et al. "A survey of semantic similarity measuring techniques for information retrieval." Computing for Sustainable Global Development (INDIACom), 3rd International Conference on. IEEE.(2016).
[14] Saruladha, K., G. Aghila. "A survey of semantic similarity methods for ontology based information retrieval." 2nd International Conference on Machine Learning and Computing (ICMLC 2010). IEEE, (2010).
[15] Guessoum, Djamel, et al. "Survey on Semantic Similarity Measure in Pervasive Computing." International Journal on Smart Sensing & Intelligent Systems 8.1 (2015).