



# Towards avatar-based discovery for IoT services using social networking and clustering mechanisms

Karima Khadir, Nawal Guermouche, Thierry Monteil, Amal Guittoum

## ► To cite this version:

Karima Khadir, Nawal Guermouche, Thierry Monteil, Amal Guittoum. Towards avatar-based discovery for IoT services using social networking and clustering mechanisms. 16th International Conference on Network and Service Management (CNSM 2020), Nov 2020, Izmir (Virtual), Turkey. hal-03009203

**HAL Id: hal-03009203**

**<https://laas.hal.science/hal-03009203>**

Submitted on 17 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Towards avatar-based discovery for IoT services using social networking and clustering mechanisms

Karima KHADIR  
LAAS-CNRS, Université  
de Toulouse, INSA  
Toulouse, France  
kkhadir@laas.fr

Nawal GUERMOUCHE  
LAAS-CNRS, Université  
de Toulouse, INSA  
Toulouse, France  
nguermou@laas.fr

Thierry MONTEIL  
LAAS-CNRS, Université  
de Toulouse, INSA  
Toulouse, France  
monteil@laas.fr

Amal GUITTOUM  
LAAS-CNRS, Université  
de Toulouse, INSA  
Toulouse, France  
aguittoum@laas.fr

**Abstract**—The Internet of Things (IoT) paradigm is defined as a complex large scale and distributed, and dynamic infrastructure composed of a huge number of heterogeneous devices. Identifying particular services provided by a massive number of IoT devices remains a challenging problem. The classical centralized discovery approaches are no more suitable. In our previous work, we have proposed an avatar-based Fog-Cloud architecture to support IoT object management. The avatars are defined as virtual entities of heterogeneous IoT objects. They are endowed with reasoning capabilities that make them able to coordinate with each other to accomplish an IoT application. Through this paper, we propose to extend our previous work by a new distributed mechanism for efficient discovery of IoT services relying on Social Networking (SN) and clustering methods. This is particularly interesting in large scale IoT systems since it allows to reduce the search space so that only the neighboring social avatars most apt to participate in the collaboration to accomplish an IoT application are considered. The proposed solution has been evaluated in connected vehicles context.

**Index Terms**—Avatars, Social Web of Things, Fuzzy c-means, IoT Services, Distributed discovery, Semantic.

## I. INTRODUCTION

The Internet of Things (IoT) infrastructure interconnects a multitude of heterogeneous objects, manages their interactions, generates huge amounts of data, and provides billions of services. Over time, the IoT has suffered from a vertical fragmentation of approaches used to cover the needs of different application areas. To overcome this problem, an abstract layer, called *Web of Things (WoT)* [1], has been designed on top of IoT. This layer aims to hide IoT heterogeneity problems, and therefore reduces the complexity of the IoT applications development process. In this context, we have already proposed an Avatar-based Fog-Cloud architecture [2]. An avatar is an artifact entity that represents a heterogeneous IoT device in a uniform description on the Web. It gives to its corresponding IoT device the ability to reason on its capabilities and to make intelligent decisions using semantic Web features, called *Semantic Web of Things SWoT* [3].

With the proliferation of IoT objects, new opportunities have been provided to build new innovative applications and scenarios. However, the IoT services discovery remains one of the challenging problems. Indeed, identifying particular IoT service among a huge number of IoT objects, required to achieve complex applications can suffer from scalability issues. In this

context, using centralized approaches where the discovery is fulfilled by one entity is limited [4].

In this paper, we propose a new distributed IoT services discovery approach that relies on Social Networking (SN) and clustering mechanisms. Indeed, the integration of SN in the WoT domain, which gives rise to the *Social Web of Things (SoWoT)* [5], seems promising. The goal is to transform smart IoT objects into smart social IoT objects and to build a set of intelligent avatars able to organize themselves to form social communities. The social avatars are linked to each other according to commitments and similarity criteria. Social Networking allows to bring out the collective intelligence of an avatars group that can provide much more interesting and precise answers to complex problems than a single avatar [6]. More specifically, in the proposed work, when a given avatar fails to achieve a complex objective by itself, based on its cooperative intelligence capabilities, it can proceed to autonomously build its social network and requests its relations to find the underlying needed IoT services. The use of SN concepts improve the effectiveness and the scalability of the discovery process. It makes it possible to accelerate and facilitate navigability in a dynamic IoT network made up of a large number of objects. To make the discovery more efficient and to reduce the search space, a Fuzzy C-means based clustering process [7] is proposed to classify avatars in the social network according to their functionalities.

This article is organized as follows. Section II analyses the related works around the existing discovery methods based mainly on SNs. Then, an overtaking scenario in the context of connected vehicles is introduced in Section III. Section IV presents the proposed SN-based discovery approach. Then, the implementation and experimental results are provided in Section V. Finally, Section VI concludes the paper.

## II. RELATED WORKS

The discovery problem remains one of the challenging problems of IoT based systems. If traditional discovery approaches include a large panel of scientific researches such as [18] and [19], the social-based approaches are relatively new.

One of the first main contributions in this field is the work of Kranz and al. [8]. The authors propose to use human social networks as an infrastructure to allow users to share the services

TABLE I  
A SYNTHESIS OF THE STUDIED WORKS

Work	Architecture	Social Metrics	Topology	Scalability	Semantic	End-to-end process
[8]	Publish/subscribe architecture	Devices owners relationships	Decentralized	Undefined	No	No
[9]	UDDI architecture	WS interaction degree	Centralized	Undefined	No	Yes
[10]	UDDI architecture	QoS	Centralized	No	No	No
[11]	Server/client architecture	C-WOR, SOR, C-LOR, OOR, POR	Centralized	Undefined	Yes	No
[12]	Undefined	Node centrality degree	Decentralized	Yes	No	Yes
[13] [14]	Server/ objects architecture	personality, proactivity, locality, trust	Centralized	Undefined	Yes	No
[15] [16]	RESTful architecture	C-WOR, SOR, C-LOR, OOR, POR	Centralized	Undefined	No	No
[17]	PaaS architecture	C-WOR, SOR, C-LOR, OOR, POR	Decentralized	Undefined	No	No
Proposal	Fog/Cloud architecture	C-WOR, C-LOR, OOR	Decentralized	Yes	Yes	Yes

provided by their objects with their friends. However, this solution implies a strong human intervention and opposes the connectivity principle on which the IoT paradigm is based. In [9], the social networking metaphor in the Web services discovery is explored. The authors were inspired by conventional human social networks such as Facebook to propose a new framework allowing the distributed discovery of Web services called LinkedWS (Linked Web Services). This model relies on capturing the different interactions that can occur between Web services to create social relationships between them accordingly. Two types of relationships are defined: collaboration or substitution. This method is interesting, however, the social relationships created between Web services are based only on the interaction degree that does not reflect the social aspects as location or interests for example. The integration of social networking in Web service discovery has been considered in [10]. This work focused only on QoS attributes and does not take into account some important social features such as the collaboration relationship between services. Atzori and al. have contributed to the use of social networking in IoT [11]. They have proposed five types of social relationships which are widely used in several works such as [20]. These relationships include: Co-Work Object Relationship (C-WOR), Co-Location Objects Relationship (C-LOR), Ownership Object Relationship (OOR), Parental Object Relationship (POR), and Social Object Relationship (SOR). Similarly in [12], a decentralized system based on IoT node centrality degree (number of object connections) is proposed. This method makes it possible to reduce the search space but using only the centrality degree criterion. However, using only this criterion is limited to build interesting social relations. In [13], the authors have proposed a new system called DSSoT (Dynamic Social Structure of Things). DSSoT is designed to strengthen sociability and to reduce the complexity of the context detection process of a given situation for heterogeneous systems. A dynamic service discovery approach, which integrates spatio-temporal contextual information and social aspects using a cognitive reasoning mechanism, is proposed in [14]. This work illustrates the implementation of the dynamic discovery of IoT services in a real-life scenario, taking as an example an instance of an airport terminal. However, their discovery approach are carried out via a single manager which makes the approach limited, non-scalable and suffers from the single failure point problem.

Several SWoT platforms have been proposed on this issue, such as ThingSpeak and Lysis. ThingSpeak platform [15] [16] has been used in several domains. It provides methods for creating and managing social relationships between IoT objects embedded in a centralized server based on object virtualization. However, the use of virtualization is limited to remote server registrations and does not provide any autonomous capabilities in terms of creating and managing social relationships that are managed at the central server level. Concerning Lysis, it provides a SWoT platform based on the Cloud infrastructure and it is deployed as a Platform-as-a-Service (PaaS) [17]. However, the end-to-end mechanisms for IoT services discovery are not addressed at all.

Table I summarizes the studied works and compares them according to six properties: **Architecture**: defines the system deployment architecture, **Social Metrics**: designates the social used metrics, **Topology**: indicates the nature of the systems management, **Scalability**: indicates whether the approach is applicable for large-scale systems, **Semantic**: refers to the use of rich semantic features, **End-to-end process**: specifies if the social network building process and the services discovery methods are supported.

Through this study, we have concluded that in the most of existing contributions that integrate social concepts into IoT, few details about architectures and social models are provided. Besides, very few approaches have focused on the methods of discovery and composition of services provided by IoT objects. Mainly, the proposed approaches ignore important features when building social networks such as interests and collaboration capabilities. Moreover, the discovery is not distributed over a set of social networks which can be highly useful in large scale infrastructures. In this context, we propose through this paper a distributed approach that integrates social networking and clustering mechanism for efficient discovery of IoT services based on autonomous avatars.

### III. MOTIVATING USE CASE

In this section, a use case of vehicles overtaking assistance system is provided. The aim is to find the appropriate IoT objects which can collaborate with the target vehicle that is not equipped with all the required devices that allow it to perform the overtaking alone in an efficient manner. The corresponding process is shown in Fig.1.

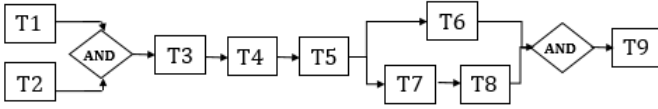


Fig. 1. The process of overtaking use case

The overtaking process starts when the driver puts on the left-blinking of his vehicle. The first task to do is to check if the rear vehicle (T1) or the front vehicle are overtaking (T2). Then, and if those two conditions are checked, the distance from the front vehicle is measured (T3). If this distance is greater than 30m, the overtaking is possible, therefore, the speed of the front vehicle is also measured (T4). The speed difference with the front vehicle is calculated (T5). If this difference is less than 20km/h, accelerated overtaking is performed. For that, the possible and available distance to accomplish the overtaking is measured (T6) and it is compared (T9) to the minimum safety distance required are calculated according to [21] (T8). To calculate this distance, the speed of the incoming vehicle is measured (T7). We suppose that this process is carried out in the situation shown in Fig.2:

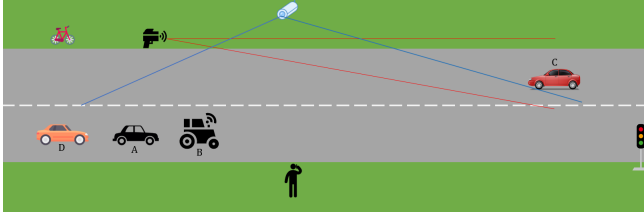


Fig. 2. Overtaking use case situation

In this scenario, we assume that the target vehicle A, which wants to overtake the tractor B which rolling at low speed, doesn't have the perception mechanisms as cameras to perceive its environment. So, to perform the overtaking safely, vehicle A needs to collaborate with the devices of its environment. In such a scenario, we suppose that there are: a radar and a camera installed by the roadside, a bicycle rolling beside, the pedestrian who uses a smartphone, a traffic light a few meters from host vehicle, the drivers of A, B and C have smartphones. A Global Navigation Satellite System (GNSS) is also available for vehicles tracking and an RSU (Road-Side Unit) which covers the region in which the host vehicle is located. We suppose also that the travel speed of the host, front, rear and oncoming vehicles is constant.

#### IV. AVATARS DISCOVERY PROCESS

The avatars discovery problem handled is described as follows: given an abstract process which models an IoT application that an avatar, which we call initiator, cannot perform alone, and a set of available avatars, the goal is to enable the initiator to discover in a distributed way a coalition of avatars that can collaborate with it to resolve the process. This approach is summarized in five steps (Fig.3): 1) handling the abstract

definition of avatars and the process to achieve, 2) building the social network of avatars, 3) social network clustering to classify its avatars according to their functionalities, 4) propagating and managing requests, and 5) composition plan construction. Hereafter, we detail each step.

##### A. Avatar and process definition

This section gives the definition of an avatar and the process that characterizes the objectives to realize.

1) *Avatar definition*: an avatar is modeled by a set of functionalities  $f_i = (f_i^1, f_i^2, \dots, f_i^p)$  and services  $S_i = (s_i^1, s_i^2, \dots, s_i^q)$  that it provides. These functionalities represent a generic description of what the services can achieve. It also characterized by a social context which is described by a set of semantic metadata published in its corresponding regional repository to allow it to build its social links. The social descriptions of an avatar  $A_i$  include: 1) a set of interest quantified with an interest degrees that are modeled with a vector  $I_i = (I_i^1, I_i^2, \dots, I_i^m)$  of  $m$  dimensions ( $0 < I_p < 1$ ,  $p$  in  $[0, m]$ ), 2) a location defined with the longitude and latitude of the represented object ( $LongA_i, LatA_i$ ), and 3) the identifier of the owner.

2) *Process definition*: the process to be resolved is made up of several composite or/and atomic tasks. It is designed as a workflow. It is modeled by a vector  $T_{req} = \{T_1, T_2, \dots, T_n\}$ , such as  $T_k$  is its  $k^{th}$  task. The tasks are interconnected with the sequential, parallel, loop, and Conditional operators.

*Example*: Table II gives an overview of the set of candidate avatars  $A_i$ , their services  $S_{ij}$  for each abstract task of the overtaking process  $T_j$ , such as for each service  $S_{ij}$ , the avatar  $A_i$  can perform the abstract task  $T_j$ . The set of interests depicted in the table are:  $I_1$  is Transport,  $I_2$  is Weather,  $I_3$  is Perception,  $I_4$  is Obstacle,  $I_5$  Communication,  $I_6$  is Luminosity,  $I_7$  is Geolocalization and  $I_8$  is Circulation. The set of functionalities are:  $f_1$  is the environment perception,  $f_2$  is the distance measure,  $f_3$  is the speed measure,  $f_4$  is the computing capacity,  $f_5$  is the moving and  $f_6$  represents the communication.

##### B. Social network building

A tree of connected regional repositories in which the avatars publish their semantic descriptions allows to manage the system scalability. So, the avatar begins by recovering the avatars that have at least one common interest by executing the following SPARQL query on the corresponding regional repository to which it is connected.

```

SELECT *
WHERE {
  ?resource avatarOnt:hasInterest ?Interest
  FILTER (?Interest IN
    (avatarOnt:I1 , ... , avatarOnt:Im))
}
  
```

Once this list is recovered, the avatar calculates social relationships with each avatar on this list using three relationships:

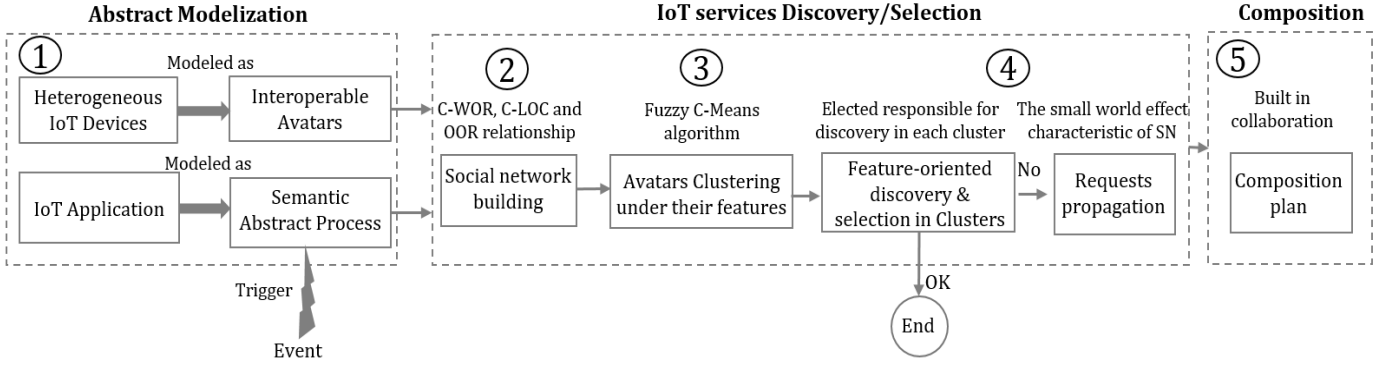


Fig. 3. An overview of the proposed approach

TABLE II  
CANDIDATE AVATARS SERVICES OF EACH ABSTRACT TASK

Avatar	Services	Interests	Location	Owner	Features
A <sub>1</sub> : Vehicle A	S <sub>13</sub> , S <sub>14</sub> , S <sub>15</sub> , S <sub>18</sub> , S <sub>19</sub>	I <sub>1</sub> (0.6), I <sub>2</sub> (0.2), I <sub>4</sub> (0.6)	43.57834 1.441185	User <sub>1</sub>	f <sub>2</sub> , f <sub>3</sub> , f <sub>4</sub> , f <sub>5</sub> , f <sub>6</sub>
A <sub>2</sub> : Vehicle B	S <sub>22</sub> , S <sub>23</sub> , S <sub>24</sub>	I <sub>1</sub> (0.9), I <sub>2</sub> (0.3), I <sub>3</sub> (0.5), I <sub>4</sub> (0.6)	43.578291 1.441146	User <sub>2</sub>	f <sub>2</sub> , f <sub>3</sub> , f <sub>4</sub> , f <sub>5</sub> , f <sub>6</sub>
A <sub>3</sub> : Vehicle C	S <sub>37</sub>	I <sub>1</sub> (0.9), I <sub>2</sub> (0.3), I <sub>3</sub> (0.4), I <sub>4</sub> (0.5)	43.578074 1.441052	User <sub>3</sub>	f <sub>2</sub> , f <sub>3</sub> , f <sub>4</sub> , f <sub>5</sub>
A <sub>4</sub> : Vehicle D	S <sub>41</sub>	I <sub>1</sub> (0.9), I <sub>2</sub> (0.3), I <sub>3</sub> (0.4), I <sub>4</sub> (0.5)	43.578074 1.441052	User <sub>4</sub>	f <sub>2</sub> , f <sub>3</sub> , f <sub>4</sub> , f <sub>5</sub> , f <sub>6</sub>
A <sub>5</sub> : Bicycle	/	I <sub>1</sub> (0.9)	43.578315 1.441277	User <sub>5</sub>	f <sub>5</sub> , f <sub>6</sub>
A <sub>6</sub> : RSU	/	I <sub>1</sub> (0.7), I <sub>5</sub> (0.7), I <sub>7</sub> (0.5)	43.577163 1.440768	User <sub>6</sub>	f <sub>5</sub> , f <sub>6</sub>
A <sub>7</sub> : Radar	S <sub>77</sub>	I <sub>1</sub> (0.7), I <sub>4</sub> (0.9)	43.57827 1.4412062	User <sub>6</sub>	f <sub>2</sub> , f <sub>3</sub> , f <sub>5</sub>
A <sub>8</sub> : Camera	S <sub>81</sub> , S <sub>82</sub>	I <sub>1</sub> (0.6), I <sub>2</sub> (0.4), I <sub>3</sub> (0.9), I <sub>4</sub> (0.7)	43.578405 1.441300	User <sub>6</sub>	f <sub>1</sub> , f <sub>2</sub> , f <sub>5</sub>
A <sub>9</sub> : Traffic light	/	I <sub>1</sub> (0.7), I <sub>8</sub> (0.9)	43.577899 1.440873	User <sub>6</sub>	f <sub>5</sub>
A <sub>10</sub> : GNSS	S <sub>10,1</sub> , S <sub>10,2</sub> , S <sub>10,3</sub> , S <sub>10,4</sub> , S <sub>10,6</sub> , S <sub>10,7</sub>	I <sub>1</sub> (0.8), I <sub>4</sub> (0.7), I <sub>7</sub> (0.9)	Server	User <sub>7</sub>	f <sub>1</sub> , f <sub>2</sub> , f <sub>3</sub> , f <sub>5</sub>
A <sub>11</sub> : Road lighting	/	I <sub>1</sub> (0.2), I <sub>4</sub> (0.3), I <sub>6</sub> (0.9), I <sub>8</sub> (0.4)	43.57822 1.441251	User <sub>6</sub>	f <sub>5</sub>
A <sub>12</sub> : Smartphone A	/	I <sub>1</sub> (0.4), I <sub>5</sub> (0.9), I <sub>7</sub> (0.7)	43.57834 1.441185	User <sub>1</sub>	f <sub>6</sub>
A <sub>13</sub> : Smartphone B	/	I <sub>1</sub> (0.4), I <sub>5</sub> (0.9), I <sub>7</sub> (0.7)	43.578291 1.441146	User <sub>2</sub>	f <sub>6</sub>
A <sub>14</sub> : Smartphone C	/	I <sub>1</sub> (0.4), I <sub>5</sub> (0.9), I <sub>7</sub> (0.7)	43.578074 1.441052	User <sub>3</sub>	f <sub>6</sub>
A <sub>15</sub> : Smartphone X	/	I <sub>1</sub> (0.4), I <sub>5</sub> (0.9), I <sub>7</sub> (0.7)	43.578332 1.441082	User <sub>5</sub>	f <sub>6</sub>

1) *Co-work Object Relationship (C-WOR)*: it binds the avatars that can collaborate with each other to accomplish a given work. The C-WOR between two avatars  $A_i$  and  $A_j$ , denoted  $CS(A_i, A_j)$ , is calculated using their respective interests vectors  $I_i$  and  $I_j$  and based on cosine similarity metric [22] as given in Equation 1:

$$CS(A_i, A_j) = \cos(I_i, I_j) = \frac{I_i \cdot I_j}{|I_i| \cdot |I_j|} \quad (1)$$

Where  $|I_i|$  and  $|I_j|$  are the norms of  $I_i$  and  $I_j$  respectively and  $0 < CS(A_i, A_j) < 1$  such that the value 0 denotes that  $A_i$  and  $A_j$  are independent and 1 indicates that they are similar.

2) *Co-Location Objects Relationship (C-LOR)*: it groups the avatars whose physical objects are located in the same

location. The C-LOR between two avatars  $A_i$  and  $A_j$ , denoted  $LS(A_i, A_j)$ , is calculated relying on their GPS (Global Positioning System) coordinates using the distance as-the-crow-flies as given in Equation 2:

$$LS(A_i, A_j) = \frac{1}{D(A_i, A_j) + 1} \quad (2)$$

$$D(A_i, A_j) = R * \arccos[\cos(LatA_i) * \cos(LatA_j) * \cos(LongA_j - LongA_i) \sin(LatA_i) * \sin(LatA_j)] \quad (3)$$

Where:  $R = 6371Km$  is the Earth radius,  $(LatA_i)$  and  $(LatA_j)$  are the decimal latitudes of  $A_i$  and  $A_j$ , and  $(LongA_i)$  and  $(LongA_j)$  their longitudes.

3) *Ownership Object Relationship (OOR)*: it groups the avatars that belong to the same owners. It is calculated between  $A_i$  and  $A_j$  as given in Equation 4:

$$OS(A_i, A_j) = \begin{cases} 1 & \text{if } A_i \text{ and } A_j \text{ have the same owner} \\ 0 & \text{else} \end{cases} \quad (4)$$

4) *Synthesis*: based on the above relationships, the social distance between two avatars  $A_i$  and  $A_j$  is modeled by a function  $SD(A_i, A_j)$  in  $[0, 1]$  as given in Equation 5:

$$SD(A_i, A_j) = \alpha * CS(A_i, A_j) + \beta * LS(A_i, A_j) + \gamma * OS(A_i, A_j) \quad (5)$$

Weighting coefficients  $\alpha, \beta, \gamma \in [0, 1]$  with  $\alpha + \beta + \gamma = 1$  represent the preferences and the Weight given for each distance. They are defined by the experts for each type of application.

### C. Neighboring avatars clustering

After the construction of its social network, the initiator avatar proceeds to the clustering of its neighboring avatars according to their functionalities to make the discovery more efficient by transmitting the requests to the appropriate avatars as well as possible. The Clustering [23] is defined as a process of partitioning a set of data points into homogeneous clusters. In our work, we rely on the Fuzzy C-means algorithm because

an avatar can provide various functionalities and therefore it can belong to several clusters. The main idea of this algorithm is to use membership coefficients that designate the belonging degree of each avatar to each cluster. It is carried out using an iterative optimization of an objective function, with the updating of the membership coefficients and cluster centroids which are closely related. Algorithm 1 presents the Fuzzy C-means clustering procedure where  $N$  is the number of avatars and  $X$  is the data matrix. Each line of  $X$  represents the functionality vector of an avatar in a binary form: the component is worth 1 if the avatar can perform the corresponding functionality and 0 otherwise.  $m$  represents the fuzzification parameter of the algorithm and  $Dist$  is the distance to use to measure the similarity between centroids and avatars as the euclidean, standard Euclidean, manhattan, and camberra distance.

---

**Algorithm 1** Avatars Fuzzy C-means Algorithm

---

**Inputs:** number of clusters  $C$ , avatars data matrix  $X=[x_{ij}]$   
 1. Initialize randomly the membership coefficients matrix  $U=[u_{ij}]$ ,  $U^{(0)}$  Such as:  $\sum_{j=1}^C u_{ij} = 1$   
 2. Calculate the centroid  $C_j$  of each cluster  $j$

$$C_j = \frac{\sum_{i=1}^N [u_{ij}]^m x_{ij}}{\sum_{i=1}^N [u_{ij}]^m} \quad (6)$$

3. Update the current membership matrix  $U^{(k)}$ ,  $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{z=1}^C \left( \frac{Dist(X_i, C_j)}{Dist(X_i, C_z)} \right)^{\frac{2}{m-1}}} \quad (7)$$

**If**  $(\sum_{i=1}^N \sum_{j=1}^C [u_{ij}]^m Dist(X_i, C_j)^2 < \epsilon)$  **Then Stop**  
**Else Return** to step 2.

---

After obtaining the final membership matrix, a membership threshold is used to define the final clusters, i.e. an avatar belongs to a given cluster only if its membership degree is greater than this threshold which is generally between 0.4 and 0.7 to obtain good results according to our experiments.

#### D. Discovery, selection and propagation

Once the social network clusters are built, an elected avatar is designated in each cluster. The delegate allows to handle requests sent to its cluster to discover the appropriate avatars. The aim is to propagate requests over different networks to enable distributed discovery while minimizing the redundancy of messages exchanged between avatars. After that, the initiator proceeds to the discovery of the required services for each task of the abstract process. For this, it sends a request to the elected avatar responsible for the functionality expressed in the task. The elected will take charge of searching for the requested service in its cluster. As soon as a service is found, it will be selected to accelerate the discovery. However, if no service is found at the cluster level, the elected avatar propagates the request by creating its own social network and designates an elected avatar in the new social network to perform the research. The propagation can continue up to sixth levels, inspiring from

the effect of the small world of social networks in sociology [24].

#### E. Execution plan generation

Whenever a service is selected, two new triplets are added to the process description: a triplet to indicate the chosen avatar for the task  $[task:T_k]$   $[isRealizedBy]$   $[Avatar:A_i]$ , and another to indicate the service to invoke  $[task:T_k][isExecutedBy][Service:S_k]$  to build the final composition execution plan. This plan will be shared by the initiator with the avatars involved in the composition.

*Example:* For the example, the vehicles, the RSU, the GNSS, and the different devices are associated with their avatars in the regional-edge server. Once the left-blinking is put ON, the avatar  $A_1$  starts by reasoning on its knowledge base. It finds that it can perform the tasks:  $T_3, T_4, T_5, T_8$  and  $T_9$ , but it can't perform the tasks  $T_1, T_2, T_6, T_7$  (Perception, Distance Measure, and Speed Measure). Consequently,  $A_1$  starts with discovering the avatars that have at least one interest in common with it from the server. After that, it builds its social network. For this, we set the application preferences to  $\alpha=0.4$ ,  $\beta=0.5$ , and  $\gamma=0.1$  given the importance of interest and localization in such a scenario of connected vehicles. Taking a social network membership threshold equal to 0.6 and after calculating social relationships, we find that the avatars who have relationships with  $A_1$  are:  $A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9$  and  $A_{12}$ . The result of clustering of these avatars according to their functionalities is: Cluster 1 that corresponds to Perception contains  $A_4, A_2, A_8, A_5$ , Cluster 2 that represents Distance Measure contains  $A_7, A_{12}, A_9, A_6$  and Cluster 3 that represents Speed Measure contains  $A_3, A_7, A_4, A_2$ . The avatar which has the closest distance to the centroid of its cluster is elected. Therefore,  $A_4$  is the elected of Cluster 1,  $A_7$  is the elected of Cluster 2 and  $A_3$  represents the elected of Cluster 3. So, for  $T_1$  and  $T_2$  (which implies the perception functionality),  $A_1$  sends a request to the elected  $A_4$  which takes care of finding the services responding to  $T_1$  and  $T_2$  in Cluster 1. It finds that  $T_1$  can be performed by itself and  $T_2$  by  $A_2$ . Likewise,  $A_1$  sends a request for  $T_6, T_7$  to the elected  $A_7$ , and  $A_3$  respectively. For  $T_6$  no avatar of Cluster 2 can perform it and for  $T_7$ , it can be performed directly by the elected  $A_3$ . Therefore, the only task that cannot be solved at the first social level is  $T_6$ .  $A_7$  is responsible for researching this task by creating its social network which will contain  $A_6, A_{10}, A_8$ , and  $A_3$ .  $A_6$  will be the elected of this social network to seek to resolve  $T_6$ . This avatar finds that  $A_{10}$  can perform this task. From our simulations, the response time for this scenario is 30ms.

## V. APPROACH EVALUATION

In this section, we first describe the setup environment and the main technical choices. After that, the performance and the scalability of the proposed approach are evaluated.

### A. Implementation technologies

The avatars implementation is done using microservices components with Spring boot, Apache HTTP Components for communication, and XML format for transporting exchanged messages. The semantic descriptions is based on OWL, SPARQL and Apache JENA.

### B. Experimental results

A study for the three principals stages of our approach is performed. Then a global performance study based on two criteria: response time and success rate is presented. The test execution environment is a Ubuntu Server with 251GB RAM and AMD Opteron multiprocessor.

1) *Avatar Semantic data generation*: the graph in Fig.4 represents the variation of the generation step time which includes semantic data distribution, semantic file creation, and metadata publication. This step takes a considerable response time given the use of rich semantic descriptions but it is done only once at the startup time of the system.

2) *Social network building*: the graph in Fig.5 represents the variation of social network building time. The results show that this stage, in addition to its impact on reducing the discovery search space, has a reduced response time even for a large scale of avatars (about 3s for 7000 avatars).

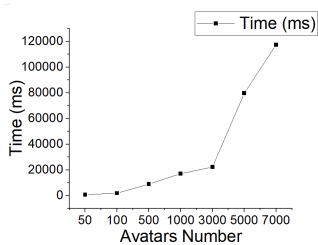


Fig. 4. Semantic data generation time according to the avatars number

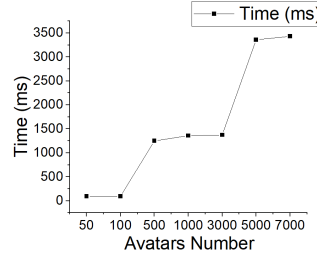


Fig. 5. Social network construction time according to the avatars number

3) *Clustering step*: to choose adequate fuzzy C-means parameters for the clustering algorithm, we have performed a set of experiments based on Partition Entropy PE, Partition Coefficient PC, and Xie & Beni index XB indices to measure clustering quality while considering the response time.

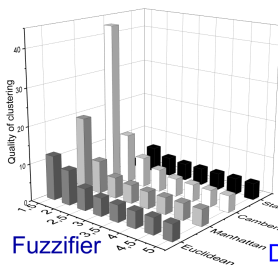


Fig. 6. Clustering quality variation as a function of both distance and fuzzifier

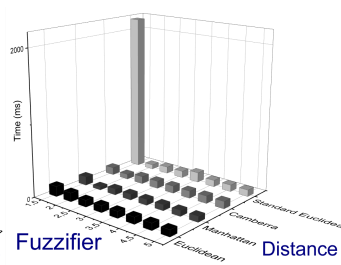


Fig. 7. Clustering time variation as a function of both distance and fuzzifier

According to graphs in Fig.6, Fig.7 and Fig.8, we can confirm that the distance which can offer good clustering quality, as well

as optimal clustering time, is Standard Euclidean distance, the adequate value of the fuzzifier is 2 and the good epsilon value is 0.1.

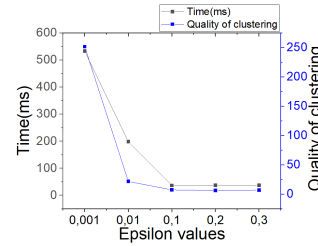


Fig. 8. Variation of clustering quality and time according to epsilon values

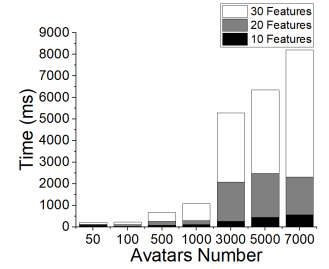


Fig. 9. Variation of fuzzy clustering time as a function of avatars number

In Fig.9, a performance evaluation of the chosen configuration using previous fixed Fuzzy C-means parameters shows that the clustering method has a reduced response time even with a large number of avatars (5s for 7000 avatars).

4) *Discovery process*: The graphs in Fig.10 and Fig.11 present a comparative study between our approach and the centralized approach which excludes clustering and social network construction. The considered parameters are: tasks number setting to 12, social network size setting to 40 avatars,  $\alpha=0.4, \beta=0.4, \delta=0.2$  and cluster size to 10 avatars.

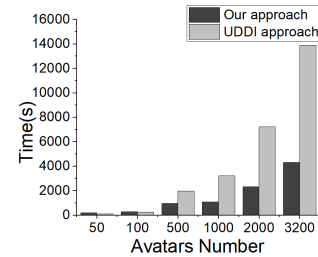


Fig. 10. Discovery time variation according to avatars number

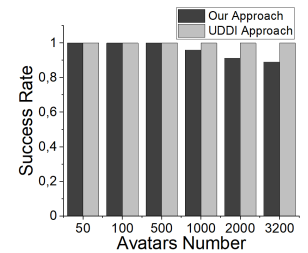


Fig. 11. Success rate variation according to avatars number

Fig.10 shows that the two methods give approximately the same results for small avatars networks (less than 500) while guaranteeing the same success rate as depicted in Fig.11. But for big size-network (greater than 500 avatars), our approach outperforms the centralized approach while guaranteeing a high success rate.

## VI. CONCLUSION

In this paper, we have proposed a new efficient distributed approach based on avatars for IoT services discovery. This approach relies on social networking to reduce the research space and clustering mechanisms to enable an efficient distributed request discovery. The proposed solution is particularly interesting in the context of complex systems with a huge number of heterogeneous objects. The proposed work has been implemented into a microservice architecture to support a generic framework for interoperability and modularity. Using this architecture, several experimental studies have been

carried out on a large scale of avatars. The results show that our approach outperforms the traditional centralized approach, particularly when the number of devices is huge.

We plan to extend the proposed approach by integrating QoS (Quality of Service) parameters for distributed and parallel selection of the IoT services discovered in this stage.

## ACKNOWLEDGMENT

This work is funded by Continental Digital Service France (CDSF) in the framework of the eHorizon project.

## REFERENCES

- [1] D. Raggett, "The web of things: Challenges and opportunities," *Computer*, vol. 48, no. 5, pp. 26–32, 2015.
- [2] K. Khadir, N. Guermouche, and T. Monteil, "Autonomous avatar-based architecture for value-added services provision," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. IEEE, 2019, pp. 291–298.
- [3] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "Towards cooperative semantic computing: a distributed reasoning approach for fog-enabled swot," in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*. Springer, 2018, pp. 407–425.
- [4] B. Jia, W. Li, and T. Zhou, "A centralized service discovery algorithm via multi-stage semantic service matching in internet of things," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1. IEEE, 2017, pp. 422–427.
- [5] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)—when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.
- [6] E. A. Mennis, "The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations," *Business Economics*, vol. 41, no. 4, pp. 63–65, 2006.
- [7] W. Wang, Y. Zhang, Y. Li, and X. Zhang, "The global fuzzy c-means clustering algorithm," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1. IEEE, 2006, pp. 3604–3607.
- [8] M. Kranz, L. Roalter, and F. Michahelles, "Things that twitter: social networks and the internet of things," in *What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, 2010, pp. 1–10.
- [9] Z. Maamar, L. K. Wives, Y. Badr, S. Elnaffar, K. Boukadi, and N. Faci, "Linkedws: A novel web services discovery model based on the metaphor of "social networks"," *Simulation Modelling Practice and Theory*, vol. 19, no. 1, pp. 121–132, 2011.
- [10] W. Chen, I. Paik, and P. C. Hung, "Constructing a global social service network for better quality of web service discovery," *IEEE transactions on services computing*, vol. 8, no. 2, pp. 284–298, 2013.
- [11] L. Atzori, A. Iera, and G. Morabito, "Siot: Giving a social structure to the internet of things," *IEEE communications letters*, vol. 15, no. 11, pp. 1193–1195, 2011.
- [12] M. Nitti, L. Atzori, and I. P. Cvijikj, "Friendship selection in the social internet of things: challenges and possible strategies," *IEEE Internet of things journal*, vol. 2, no. 3, pp. 240–247, 2014.
- [13] D. Hussein, S. Park, S. N. Han, and N. Crespi, "Dynamic social structure of things: A contextual approach in cps," *IEEE Internet Computing*, vol. 19, no. 3, pp. 12–20, 2015.
- [14] D. Hussein, S. N. Han, G. M. Lee, N. Crespi, and E. Bertin, "Towards a dynamic discovery of smart services in the social internet of things," *Computers & Electrical Engineering*, vol. 58, pp. 429–443, 2017.
- [15] thingspeak. (2019) thingspeak. [Online]. Available: <https://www.thingspeak.com/>
- [16] R. Girau, M. Nitti, and L. Atzori, "Implementation of an experimental platform for the social internet of things," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2013, pp. 500–505.
- [17] R. Girau, S. Martis, and L. Atzori, "Lysis: A platform for iot distributed applications over socially connected objects," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 40–51, 2016.
- [18] D. Grigori, J. C. Corrales, M. Bouzeghoub, and A. Gater, "Ranking bpel processes for service discovery," *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 178–192, 2010.
- [19] B. Dong and J. Chen, "Architecting semantics-based publish/subscribe applications for the internet of things," in *2016 International Forum on Management, Education and Information Technology Application*. Atlantis Press, 2016.
- [20] K. M. Alam, M. Saini, and A. El Saddik, "tnote: A social network of vehicles under internet of things," in *International Conference on Internet of Vehicles*. Springer, 2014, pp. 227–236.
- [21] C. Mo, Y. Li, and L. Zheng, "Simulation and analysis on overtaking safety assistance system based on vehicle-to-vehicle communication," *Automotive Innovation*, vol. 1, no. 2, pp. 158–166, 2018.
- [22] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless forwarding in pocket switched networks," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011, pp. 251–255.
- [23] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," *A review of machine learning techniques for processing multimedia content*, vol. 1, pp. 9–16, 2004.
- [24] J. Travers and S. Milgram, "An experimental study of the small world problem," *The structure and dynamics of networks*, pp. 130–148, 2006.