Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 700 – 706

Procedia Computer Science

The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)

Decentralised Service Composition using Potential Fields in Internet of Things Applications

Elli Rapti[a,]*, Anthony Karageorgos[b], Vassilis C. Gerogiannis[b]

[a]*Institute of Research and Technology, Thessaly, CERTH, Volos, Greece*
[b]*Technological Educational Institute of Thessaly, Larissa, Greece*

Abstract

Traditional service composition approaches rely mostly on centralised architectures, which have been proven inadequate in pervasive Internet of Things (IoT) environments. In such settings, where decentralisation of decision-making is mandatory, nature-inspired computing paradigms have emerged due to their inherent capability to accommodate spatiality, self-adaptivity, and evolvability. In this paper, taking inspiration from natural metaphors we propose a decentralised service composition model which is based on artificial potential fields. In the proposed approach, artificial potential fields (APFs) lead the service composition process through the balance of forces applied between service requests and service nodes. APFs are formed considering the percentage of user requested services that can be offered by service provision nodes, as well as service node availability. The applicability of the proposed approach is discussed in an exemplar scenario concerning dynamic and personalised composition of an audio-visual virtual guide service in an IoT network of a trade show venue.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Decentralised Service Composition; Dynamic Services; Internet of Things; Nature-inspired Computing; Artificial Potential Fields

1. Introduction

The term "Internet of Things" refers to "a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols"[1]. Unlike traditional Internet, IoT networks have an additional sensing

* Corresponding author. Tel.: +30-24410-64715.
E-mail address: erapti@ireteth.certh.gr

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.05.079

layer, which enables the integration of non-intelligent or weakly-intelligent devices[2] and raises issues of heterogeneity and interoperability. To this end, service-oriented computing has emerged as a promising solution as each thing provides its functionality through standard services that could be composed to offer enhanced functionality[3]. In such pervasive environments decentralised approaches for service discovery and composition[4,5,6] are promising as they offer flexibility, robustness, scalability and fault-tolerance to the system[7].

In this paper, taking advantage of the inherent capabilities of nature-inspired computing to accommodate spatiality, self-organisation, self-adaptivity, and evolvability[8], we propose a decentralized service composition model which performs emergent selection of atomic services based on artificial potential fields (APFs). Taking inspiration from[9], APFs are generated by service provision nodes and user requests and apply forces between them. The APFs generated are based on the compatibility between user service requests and services offered by service provision nodes, as well as on service provision node availability. The remainder of this paper is structured as follows: Section 3 introduces the proposed service composition approach, while Section 4 presents an exemplar case study scenario. Finally, Section 5 concludes this paper.

2. Approach

2.1. Problem Modelling

An IoT network can be defined as a directed graph $G(V, E)$, in which V denotes the sets of all nodes v and E denotes the set of wireless links among nodes which can communicate directly. We consider $V = V^P \cup V^S$, where V^P represents the set of all high level devices (e.g. tablets, smartphones, RFID readers), and V^S represents the set of low level devices (e.g. RFIDs, sensors). Services run on high level devices, while low level devices are only able to provide information to the network (e.g. current temperature). Links between nodes, which represent their capability of direct communication, are defined as $E = \{e_{ij} = (v_i, v_j) | i, j \in \mathbb{N}, i \neq j, v_i, v_j \in V^P, d(v_i, v_j) \leq R\} \cup \{(v_i, v_j) | i, j \in \mathbb{N},\ i \neq j, v_i \in V^P, v_j \in V^S, d(v_i, v_j) \leq R\}$, where $d(v_i, v_j)$ denotes the distance between nodes v_i and v_j and R denotes the maximum communication distance among two nodes.

To better represent the lack of centralised control we consider that each node $v \in V^P$ is driven by a software agent which we term service agent $\mathrm{SA}(v)$, according to a technique known as service "agentification"[10]. While services contribute to overcoming interoperability issues, the use of agent technology enables building of high-level models with flexible interaction patterns[11]. Service agents can interact with the respective device's infrastructure to be able to sense user requirements and to provide their functionality. This "servitisation" of real world object functionality allows utilisation of their capabilities through established methods for service management, discovery and composition.

In addition, due to the fact that devices are wirelessly interconnected, service agents can establish communication relationships and invoke services from different devices in their proximity, which is defined as its neighbourhood. Thus, considering a service agent $\mathrm{SA}(v_i)$ in node v_i ($i \in \mathbb{N}$), its neighbourhood is defined as a set $N(\mathrm{SA}(v_i))$ which contains all service agents in nodes $v_j \in V, j \in \mathbb{N}$, with $d(v_i, v_j) \leq R$, where R is the maximum radius of communication between any two nodes. That is: $N(\mathrm{SA}(v_i)) = \{\mathrm{SA}(v_j) | j = 1, .., N_N, v_j \in V, d(v_i, v_j) \leq R\}$, where N_N is the total number of neighbours of $\mathrm{SA}(v_i)$. In the general case the neighbourhood of a service agent changes dynamically, for example when visitor mobile devices appear and disappear as in pervasive computing settings.

Moreover, each service agent $SA(v)$ has a service set $S(SA(v))$ that represents the atomic encapsulated functionality it can provide to the users in the form of software services. That is: $S(SA(v)) = \{s_i | i = 1, 2, \dots, N_S\}$, where each $s_i \in S(SA(v))$ is a set $s_i = \{a_j^i | j = 1, \dots, N_A\}$ of attributes ($a_j^i \in s_i$) that represent the constituents of the respected service, for example the output parameters that are provided after service execution, and N_s is the total number of services provided by $SA(v)$.

Furthermore, services are provided to users as response to personalised requests for particular functionalities. Users provide each request to a requester agent $RA(v)$ through a node (i.e. a device) $v \in V^P$. A service request handled by a requester agent $RA(v)$ is denoted as $\mathrm{Req}(RA(v))$ and represents a query for services providing the requested functionality. It is formally defined as a triple $\langle I, \mathrm{Con}(RA(v)), \mathrm{Path} \rangle$, where I is the initiating node, $\mathrm{Con}(RA(v)) = \{a_j | j = 1..N_C\}$ is the request context provided by the user with N_C being the total number of service attributes requested, and Path is a sequence (i.e. ordered list) containing all nodes that participate in the composition of the final service which will serve this particular request, that is which they will provide part of the functionality requested. That is: Path $= (v_i | i \in \mathbb{N}, v_i \in V^P)$. While service agents are considered to always reside at a particular service node, requester agents can migrate between service nodes aiming to maximise satisfaction of the service attributes they carry by most suitable service agents. Requester agents that migrate to service agents are placed in a request queue waiting to be served. A request queue of a service agent $SA(v)$ is defined as a sequence $Q\big(SA(v)\big) = \big(RA(v_i) | v_i \in V^P, j = 1 \dots N_R\big)$, where N_R is the total number of requests in the queue. It is considered that each service agent can serve only one request at a time and that service provision time can generally vary depending on the attributes included in the submitted request.

2.2. From Internet of Things to Artificial Potential Fields

Potential fields are a commonly used and well understood method to achieve coordination in open and dynamic scenarios since it naturally promotes self-organization and self-adaptation[8]. Exploiting the benefits arising from field-based approaches, we adopt potential fields to achieve decentralised and dynamic service composition in the pervasive environment of IoT networks.

Services in an IoT network are composed aiming to provide functionality that will be able to satisfy user requirements. User requirements are expressed through generation of service requests in a node (i.e. a device) of the network, which in turn are represented by requester agents. Requester agents are able to communicate with service agents in their neighbourhood and vice versa. Both requester and service agents are considered to create dynamic artificial potential fields in their neighbourhood like electrically charged particles. The generated forces drive requester agents to migrate and be served at the most promising (i.e. one that is able to provide most of the requested attributes on time) service agent. Migration is considered to be taking place only along the edges of the communication graph of service nodes in V^P. The potential fields generated by requester and service agents are described in detail in the following subsections.

2.2.1. Service Satisfaction Potential Field

The service satisfaction potential field aims to ensure that the request context provided in a service request will be satisfied. When a requester agent $\mathrm{RA}(v_i)$ settles at node $v_i \in V^P$, $i \in \mathbb{N}$, it has to determine the number of attributes service agents residing in the current node and neighbouring nodes, $v_j \in V^P$ and $v_j \in N\big(\mathrm{SA}(v_i)\big)$, $j \in \mathbb{N}$, they are able to satisfy according to their available services.

Considering requester agent $\mathrm{RA}(v_i)$ representing request $\mathrm{Req}\big(\mathrm{RA}(v_i)\big)$ in node v_i ($i \in \mathbb{N}, v_i \in V^P$), we define its service satisfaction potential field $P_S(\mathrm{RA}(v_i))$ as the request context $\mathrm{Con}\big(\mathrm{RA}(v_i)\big)$. That is:

$$P_S(\mathrm{RA}(v_i)) = \mathrm{Con}\big(\mathrm{RA}(v_i)\big)$$

Accordingly, considering service agent $\mathrm{SA}(v_j)$ in node v_j ($j \in \mathbb{N}, v_j \in V^P$) with a service set $S\Big(SA\big(v_j\big)\Big) = \{s_i | i = 1, 2, \dots, N_S\}$, we define its service satisfaction potential field $P_S\big(\mathrm{SA}(v_j)\big)$ as the union of its services, that is:

$$P_S\big(\mathrm{SA}(v_j)\big) = s_1 \cup \dots \cup s_{N_S}.$$

The service satisfaction force applied between $\mathrm{RA}(v_i)$ and $\mathrm{SA}(v_j)$ is given as follows:

$$\mathrm{SF}\Big(\mathrm{RA}(v_i), \mathrm{SA}\big(v_j\big)\Big) = \frac{\big|P_S(\mathrm{RA}(v_i)) \cap P_S\big(\mathrm{SA}(v_j)\big)\big|}{\big|\mathrm{Con}\big(\mathrm{RA}(v_i)\big)\big|}$$

where $|\cdot|$ denotes the cardinality of the respective sets and $0 \leq \mathrm{SF}\Big(\mathrm{RA}(v_i), \mathrm{SA}\big(v_j\big)\Big) \leq 1$.

2.2.2. Availability Potential Field

Considering only service satisfaction could lead to congestion problems in nodes that would attract many requests for example due to high number of different services provided. To avoid congestion of service requests and decrease waiting time for requests to be served we introduce the availability potential field. The availability potential field aims to ensure that the waiting time experienced at a node by a requester agent will be kept at a minimum, as this solves the dual purpose of congestion avoidance and fast application completion[12].

Each service agent $\mathrm{SA}(v)$ in node $v \in V^P$ has a request queue $\mathrm{Q}(\mathrm{SA}(v))$ where each requester agent migrating in node v is placed waiting to be served. Considering requester agent $\mathrm{RA}(v_i)$ in node v_i ($i \in \mathbb{N}, v_i \in V^P$), waiting to be served in request queue $Q(\mathrm{SA}(v_i))$ of service agent $\mathrm{SA}(v_i)$. We define its availability potential field $P_A(\mathrm{RA}(v_i))$ as $\mathrm{RA}(v_i)$'s position in the queue, denoted as $L(\mathrm{RA}(v_i))$, that is:

$$P_A(\mathrm{RA}(v_i)) = L(\mathrm{RA}(v_i)).$$

Accordingly, considering a service agent $\mathrm{SA}(v_j)$ in node v_j ($j \in \mathbb{N}, v_j \in V^P$) with a request queue $Q\left(\mathrm{SA}(v_j)\right)$, we define its availability potential field $P_A(\mathrm{SA}(v_j))$ as the prospective position of $\mathrm{RA}(v_i)$ in $\mathrm{SA}(v_j)$'s queue, that is:

$$P_A(\mathrm{SA}(v_j)) = \left|Q\left(\mathrm{SA}(v_j)\right)\right| + 1$$

where $|\cdot|$ denotes the cardinality of the respective set.

The availability force applied between $\mathrm{RA}(v_i)$ and $\mathrm{SA}(v_j)$ is given as follows:

$$\mathrm{AF}\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right) = \frac{P_A(\mathrm{RA}(v_i)) - P_A(\mathrm{SA}(v_j))}{\max\left(P_A(\mathrm{RA}(v_i)), P_A(\mathrm{SA}(v_j))\right) - 1}$$

where $\max\left(P_A(\mathrm{RA}(v_i)), P_A(\mathrm{SA}(v_j))\right)$ is the maximum place in a queue in which $\mathrm{RA}(v_i)$ could be placed. That is $-1 \leq \mathrm{AF}\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right) \leq 1$. If $i = j$, then the availability force is $\mathrm{AF}\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right) = 0$ since current position and prospective position are equal ($P_A(\mathrm{RA}(v_i) = P_A(\mathrm{SA}(v_j)))$.

2.2.3. Resultant Force

To ensure that a requester agent $\mathrm{RA}(v_i)$ in node v_i ($i \in \mathbb{N}, v_i \in V^P$) will migrate towards the most promising node v_j ($j \in \mathbb{N}, v_j \in V^P$) that achieves the best balance between service satisfaction and availability, we linearly combine the service satisfaction force and the availability force defined above. The resultant force between requester agent $\mathrm{RA}(v_i)$ and service agent $\mathrm{SA}(v_j)$ is then given by the following formula:

$$F\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right) = a \times \mathrm{SF}\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right) + (1-a) \times \mathrm{AF}\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right)$$

where, a is a parameter in the range of [0,1] representing the degree of influence of each field, and $-0.5 \leq F\left(\mathrm{RA}(v_i), \mathrm{SA}(v_j)\right) \leq 1$. Finally, requester agent $\mathrm{RA}(v_i)$ will migrate towards the node with the highest resultant force.

2.2.4. Service Composition

In each execution step the resultant forces applied between a requester agent and neighbouring service agents are calculated, including the service agent on the node that the service requester is currently residing. Accordingly, the requester agent selects the service agent with the highest resultant force to migrate to. To achieve service composition, that is to gather a sequence of service nodes in its Path that can provide the requested attributes, a requester agent $\mathrm{RA}(v_i)$, with $v_i \in V^P$ and $i \in \mathbb{N}$, at each step checks if the following requirements are satisfied:

- The time t requester agent $\mathrm{RA}(v_i)$ has remained at the request queue of node v_i exceeds a maximum pre-defined time threshold t_{thres}
- The request context has not been fully satisfied, that is:

  $$\mathrm{Con}(\mathrm{RA}(v_i)) - (s_1 \cup s_2 \cup \ldots \cup s_{N_S}) \neq \emptyset$$

  where $\mathrm{Con}(\mathrm{RA}(v_i))$ is the request context of request $\mathrm{Req}(\mathrm{RA}(v_i))$ and $(s_1 \cup s_2 \cup \ldots \cup s_{N_S})$ is the union of the service sets provided by service agent $\mathrm{SA}(v_i)$ in node $v_i \in V^P$.

If both the requirements are satisfied then requester agent $\mathrm{RA}(v_i)$ generates a new requester agent $\mathrm{RA}'(v_i)$, whose request context is defined as follows:

$$\mathrm{Con}(\mathrm{RA}'(v_i)) = \mathrm{Con}(\mathrm{RA}(v_i)) - (s_1 \cup s_2 \cup \ldots \cup s_{N_S})$$

where $(s_1 \cup s_2 \cup \ldots \cup s_{N_S})$ is the union of the service sets provided by service agent $\mathrm{SA}(v_i)$ in node $v_i \in V^P$. Thus, $\mathrm{RA}'(v_i)$ considering the new service request $\mathrm{Req}(\mathrm{RA}'(v_i)) = \langle I, \mathrm{Con}(RA'(v_i)), \mathrm{Path}\rangle$, where I is node v_i of service requester $\mathrm{RA}(v_i)$, calculates the resultant forces with neighbouring service agents to select the most promising node to migrate to as described above.

The above process terminates when one of the following requirements is satisfied:

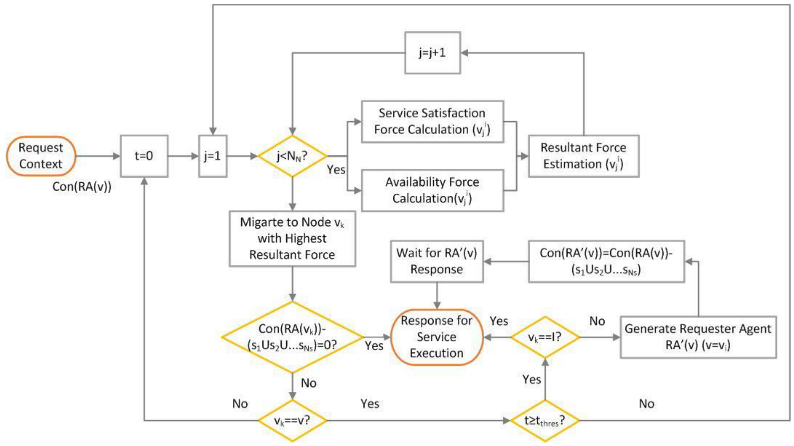


Fig. 1. Migration and service composition process of a requester agent RA(v)

- The request context has been fully satisfied, that is:

$$\mathrm{Con}\left(\mathrm{RA}'(v_j)\right) - \left(s_1 \cup s_2 \cup \ldots \cup s_{N_S^j}\right) = \emptyset$$

  where $\left(s_1 \cup s_2 \cup \ldots \cup s_{N_S^j}\right)$ is the union of the service sets provided by service agent $\mathrm{SA}(v_j)$ in node $v_j \in V^P$ that $\mathrm{RA}'(v_j)$ has migrated to.

or

- The time t a requester agent $\mathrm{RA}'(v_j)$ has remained at the request queue of node v_j exceeds a pre-defined time threshold t_{thres} and v_j is the initiating node, that is $I = v_j$ where $I \in \mathrm{Req}\left(\mathrm{RA}'(v_j)\right)$.

If one of the above requirements is satisfied, the requester agent, following the nodes included in its path, is able to reach the initiating node that generated the request, and inform the service agents participating in the service composition for execution.

The migration and service composition process described in the previous section which is adopted by a service requester agent is depicted in the flowchart of Fig. 1.

3. Exemplar Scenario - Discussion

To provide a better explanation of the proposed approach we continue by providing an exemplar scenario of a personalised audio-visual virtual guide service in a trade show venue. Trade shows can attract a significant number of different exhibitors and venues usually span thousands of square meters where booths belonging to different exhibitors are placed.

Table 1. Services provided, initial request queue and neighbourhood of each node v

Node (v)	Services (s)	Attributes (a)	Request Queue	Neighbourhood ($N(v)$)	
				High Level Nodes	Low Level Nodes
v_1	s_1 s_2	video_streaming audio_streaming	2	v_2, v_3	v_5, v_6
v_2	s_3	audio_to_text	1	v_1, v_4	--
v_3	s_4 s_5	video_HD_streaming text_tour	3	v_1	v_7, v_8
v_4	s_6	text_translation	2	v_2	--

To demonstrate the applicability of the proposed approach, we consider a furniture trade show venue that has two booths and two visitors. Each booth is equipped with one RFID reader-enable device that can extract information from RFIDs placed on exhibits. RFID reader-enabled devices, RFIDs and visitor mobile phones represent the nodes of the IoT network. The services provided, the initial request queue and the neighbourhood of each node are shown in Table 1. It should be noted that only high level devices, such as smartphones and RFID reader-enabled devices, can provide services to the network. The topology of the network is provided in Fig. 2.

A visitor through his smartphone, representing node v_2 in Fig. 2, provides his/her request context $\mathrm{Con}(\mathrm{RA}(v_2)) = \{\text{audio\_streaming}, \text{audio\_to\_text}\}$ to requester agent $\mathrm{RA}(v_2)$. A service request is generated by requester agent, that is:

$$\mathrm{Req}(\mathrm{RA}(v_2)) = \langle v_2, \{\text{audio\_streaming}, \text{audio\_to\_text}\}, \{v_2\}\rangle$$

Then, $\mathrm{RA}(v_2)$ calculates the forces with service agent $\mathrm{SA}(v_2)$ and neighbouring service agents, that is $\mathrm{SA}(v_1)$ and $\mathrm{SA}(v_4)$. Considering equal influence by both fields ($a = 0.5$), the forces applied are calculated as follows:

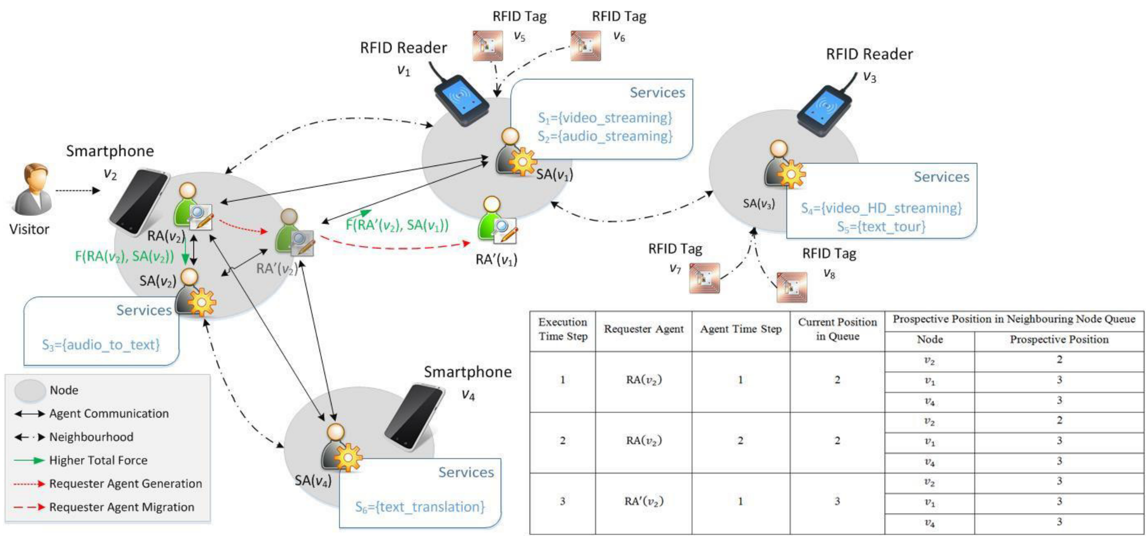
- $F(\mathrm{RA}(v_2), \mathrm{SA}(v_2)) = 0.5 \times \mathrm{SF}(\mathrm{RA}(v_2), \mathrm{SA}(v_2)) + 0.5 \times \mathrm{AF}(\mathrm{RA}(v_2), \mathrm{SA}(v_2)) \Rightarrow$
  $F(\mathrm{RA}(v_2), \mathrm{SA}(v_2)) = 0.5 \times \frac{|P_S(\mathrm{RA}(v_2)) \cap P_S(\mathrm{SA}(v_2))|}{|\mathrm{Con}(\mathrm{RA}(v_i))|} + 0.5 \times \frac{P_A(\mathrm{RA}(v_2)) - P_A(\mathrm{SA}(v_2))}{\max(P_A(\mathrm{RA}(v_2)), P_A(\mathrm{SA}(v_2))) - 1} = 0.5 \times \frac{1}{2} + 0 = 0.25.$

Accordingly for the neighbouring nodes:

- $F(\mathrm{RA}(v_2), \mathrm{SA}(v_1)) = 0.5 \times \mathrm{SF}(\mathrm{RA}(v_2), \mathrm{SA}(v_1)) + 0.5 \times \mathrm{AF}(\mathrm{RA}(v_2), \mathrm{SA}(v_1)) = 0.5 \times \frac{1}{2} + 0.5 \times \frac{2-3}{2} = 0.25 - 0.25 = 0$
- $F(\mathrm{RA}(v_2), \mathrm{SA}(v_4)) = 0.5 \times \mathrm{SF}(\mathrm{RA}(v_2), \mathrm{SA}(v_4)) + 0.5 \times \mathrm{AF}(\mathrm{RA}(v_2), \mathrm{SA}(v_4)) = 0 + 0.5 \times \frac{2-3}{1} = -0.5$

Considering the resultant forces, $\mathrm{RA}(v_2)$ selects the node of the service agent with which the highest force is applied to migrate to, that is v_2 as $F(\mathrm{RA}(v_2), \mathrm{SA}(v_2)) > F(\mathrm{RA}(v_2), \mathrm{SA}(v_1)) > F(\mathrm{RA}(v_2), \mathrm{SA}(v_4))$.

Furthermore, considering that forces between $\mathrm{RA}(v_2)$ and the service agents remain unaltered and $\mathrm{Con}(\mathrm{RA}(v_2)) - (s_3) = \{\text{audio\_streaming}\} \neq \emptyset$, $\mathrm{RA}(v_2)$ generates a new requester agent $\mathrm{RA}'(v_2)$ when $t \geq t_{thres}$. The request context of $\mathrm{RA}'(v_2)$ is $\mathrm{Con}(\mathrm{RA}'(v_2)) = \mathrm{Con}(\mathrm{RA}(v_2)) - (s_3) = \{\text{audio\_streaming}\}$ and its service request is $\mathrm{Req}(\mathrm{RA}'(v_2)) = \langle v_2, \{\text{audio\_streaming}\}, \{v_2\}\rangle$. Consequently, $\mathrm{RA}'(v_2)$ calculates the forces as above, that is $F(\mathrm{RA}'(v_2), \mathrm{SA}(v_2)) = 0$, $F(\mathrm{RA}'(v_2), \mathrm{SA}(v_1)) = 0.25$ and $F(\mathrm{RA}'(v_2), \mathrm{SA}(v_4)) = 0$. Thus, $\mathrm{RA}'(v_2)$ migrates to node v_1 as $F(\mathrm{RA}'(v_2), \mathrm{SA}(v_1)) > F(\mathrm{RA}'(v_2), \mathrm{SA}(v_2)) > F(\mathrm{RA}'(v_2), \mathrm{SA}(v_4))$. Since $\mathrm{Con}(\mathrm{RA}'(v_1)) - (s_1 \cup s_2) = \emptyset$, $\mathrm{RA}'(v_1)$ sends a response to $\mathrm{RA}(v_2)$ for service execution. The requester agents' state in each execution step is shown in Fig. 2.



Execution Time Step	Requester Agent	Agent Time Step	Current Position in Queue	Prospective Position in Neighbouring Node Queue	
				Node	Prospective Position
1	$\mathrm{RA}(v_2)$	1	2	v_2	2
				v_1	3
				v_4	3
2	$\mathrm{RA}(v_2)$	2	2	v_2	2
				v_1	3
				v_4	3
3	$\mathrm{RA}'(v_2)$	1	3	v_2	3
				v_1	3
				v_4	3

Fig. 2. Schematic representation of service composition based on forces applied on requester agent $\mathrm{RA}(v_2)$

Through this exemplar case study scenario we showed how we can achieve dynamic service composition in a decentralised way through the application of artificial potential fields. Thus, based only on local knowledge we are able to achieve maximum user requirements satisfaction while minimising waiting time at each service node by considering node availability. The approach can also achieve scalability as more nodes enter the system since it can provide balance between node congestion and service satisfaction. Nodes with high service request queue push prospective request towards less congested nodes and vice versa, nodes with low demand attract prospective requests from congested nodes.

4. Conclusion

This paper presents an approach for decentralised service composition in the pervasive environment of IoT networks that is based on the notion of Artificial Potential Fields. The paper has shown how artificial potential fields can be mapped to the service composition problem in IoT networks to achieve global equilibrium through local interactions. In particular, artificial potential fields created by service request satisfaction and node availability apply virtual forces on service requests, generated by user requirements, which in turn create service compositions as a result of balance in the applied forces.

As future work we plan to evaluate the proposed approach by simulation experiments using a purpose built simulator, comparing it with several centralized and decentralised approaches. Moreover, we will test its performance through an implementation in a real world IoT application. Further research efforts will include the extension of the proposed work so as to consider multiple variables in generating artificial potential fields, such as hop count. In addition, we will incorporate dynamic adaptation in the composition model due to changes in network topology.

Acknowledgements

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - in the context of the research project “SPRINT SMEs”, Act: “ARCHIMEDES III” in the Operational Program “Education and Lifelong Learning 2007-13”. Investing in knowledge society through the European Social Fund.

References

1. Christin D, Reinhardt A, Mogre PS, Steinmetz R. Wireless sensor networks and the internet of things: selected challenges. Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze. 2009:31-4.
2. Ma H-D. Internet of things: Objectives and scientific challenges. Journal of Computer science and Technology. 2011;26(6):919-24.
3. Spiess P, Karnouskos S, Guinard D, Savio D, Baecker O, Souza L, et al., editors. SOA-based integration of the internet of things in enterprise services. Web Services, 2009 ICWS 2009 IEEE International Conference on; 2009: IEEE.
4. Chakraborty D, Perich F, Joshi A, Finin T, Yesha Y. A reactive service composition architecture for pervasive computing environments. Mobile and Wireless Communications: Springer; 2003. p. 53-60.
5. Bakhouya M, Gaber J. Service composition approaches for ubiquitous and pervasive computing: A survey. Agent Systems in Electronic Business, IRMA International, Pensilvania, United States. 2008:323-50.
6. Gu X, Nahrstedt K, Yu B, editors. SpiderNet: An integrated peer-to-peer service composition framework. High performance Distributed Computing, 2004 Proceedings 13th IEEE International Symposium on; 2004: IEEE.
7. Bicchi A, Marigo A, Pappas G, Pardini M, Parlangeli G, Tomlin C, et al., editors. Decentralized air traffic management systems: performance and fault tolerance. the Proceedings of the IFAC International Workshop on Motion Control; 1998: Citeseer.
8. Mamei M, Zambonelli F. Field-based coordination for pervasive multiagent systems: Springer Science & Business Media; 2006.
9. Jiang H, Sun Y, Sun R, Chen W, Ma S, Gao J. A Distributed Energy Optimized Routing Using Virtual Potential Field in Wireless Sensor Networks. International Journal of Distributed Sensor Networks. 2014;2014.
10. Chen F, Yang H, Guo H, Xu B, editors. Agentification for web services. Computer Software and Applications Conference, 2004 COMPSAC 2004 Proceedings of the 28th Annual International; 2004: IEEE.
11. Shen W, Hao Q, Wang S, Li Y, Ghenniwa H. An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. Robotics and Computer-Integrated Manufacturing. 2007;23(3):315-25.
12. Ahmed T, Tripathi A, Srivastava A, editors. Rain4Service: An approach towards decentralized web service composition. Services Computing (SCC), 2014 IEEE International Conference on; 2014: IEEE.