A Knowledge-Based Approach for Resource Discovery and Allotment in Swarm Middleware

Michele Ruta / Floriano Scioscia / Eliana Bove / Annarita Cinquepalmi / Eugenio Di Sciascio
Dept. of Electrical and Information Engineering – Politecnico di Bari
Via E. Orabona 4 Bari, I-70125
ITALY

email: michele.ruta@poliba.it / floriano.scioscia@poliba.it / eliana.bove@poliba.it /
annarita.cinquepalmi@poliba.it / eugenio.disciascio@poliba.it

ABSTRACT

This paper proposes a semantic-based solution to resource discovery, allotment and sharing in swarm intelligence scenarios. The envisioned framework allows a novel and advanced retrieval of resources in highly dense contexts, based on semantics of the annotation they convey. In fact, logic-based inferences permit to determine the compatibility between a request and available resources so enhancing the quality of discovery. In order to do that, a vocabulary (ontology) sharing is unavoidable to properly annotate resources. The approach proposed here bypass centralized and encumbering Knowledge Bases (KBs) through an ontology scattering and rebuilding when needed. On the other hand, factual knowledge follows the material resources and is distributed by nature. The proposed framework has been implemented in a prototype to prove correctness of the approach and obtain an early performance evaluation. The adopted communication solution is represented by Bee Data Distribution System (Bee-DDS), a message-oriented middleware exploiting a publish-subscribe model. It provides affordable interaction among loosely-coupled resources in the scenario to support advanced discovery and sharing functionalities.

1 INTRODUCTION

The Internet of Things (IoT) vision theorizes information could be scattered at the field layer of an ad-hoc network populated by non-obtrusive small devices capable of some processing and storage and also equipped with lightweight wireless communication facilities. Early studies in such contexts and scenarios envisioned IoT as a small reproduction of Internet of Personal Computers and, as a consequence, tended to reuse service-oriented approaches and Web-derived data management. Unfortunately, such attempts revealed immediately their limits and problems in unpredictable IoT scenarios featured by high volatility and resource paucity. The peculiarities of involved actors and possible case studies called for more flexible and interoperable schemes for information discovery and sharing. Micro-devices were required to autonomously interact among them and with the user.

Just for the above reasons, artificial intelligence has been largely investigated for searching and implementing innovative solutions to resource discovery and sharing. In particular, Knowledge Representation (KR) techniques and technologies have been employed for resource representation and processing providing machine-understandability to languages for service/resource annotation. In more detail, formalisms based on semantics of Description Logics (DLs) have been considered, given the good trade-off they provided between expressiveness and computational complexity. General knowledge on a domain is encapsulated in an ontology, *i.e.,* the thesaurus of concepts and relationships to be used in the annotation of data, events and entities involved in a particular scenario. The logic-based descriptions, along with their reference ontology, compose a Knowledge Base (KB). Software tools named reasoners can execute automated inferences to extract implicit knowledge from what is explicitly stated in a KB.

This paper proposes a knowledge-based approach for resource discovery, allocation and sharing in distributed scenarios like the IoT ones, exploiting a publish/subscribe (pub/sub) Message-Oriented

Middleware (MOM) as reference communication layer. The resulting framework is a layered one. The semantic-based resource/service discovery acts as interface towards users and their applications. It recalls requests and provides a ranking of available resources scored according to their semantic compatibility with the request. A DL mobile reasoner [9] is integrated for this purpose.

Under the topmost discovery level, there is a layer materializing the minimal conceptualization needed for supporting inferences on a given set of semantic descriptions. Ontology fragments (chunks) disseminated on the devices in the environment are collected in a collaborative fashion and a slight least ontology is rebuilt on-the-fly: both the size of individual ontology chunks (compressed before) and the number of message interactions required for reassembling the TBox are taken into account to determine the rebuilding policies. The above opportunistic model complies with the ubiquitous KB (u-KB) [19] paradigm, where a node in a larger distributed system hosting a reasoner fetches at runtime all and only the KB parts required for the current inference problem.

Finally, the lowest layer is a pub/sub MOM. It operates an affordable communication among the loosely-coupled objects populating the environment and enables services allowed by the higher-level layers.

The theoretical framework described above has been implemented in a working prototype whose pub/sub communication layer is represented by Bee Data Distribution System (Bee-DDS) http://sine.ni.com/nips/cds/view/p/lang/it/nid/211025/. The resulting system has been used to set-up early verifications devoted to ensure correctness of the approach as well as to perform a preliminary evaluation of performance. It must be considered that the integration of a support for semantic-based resource discovery in swarm intelligence impact on data exchange and on-node computation. First outcomes indicate the theoretical approach is feasible and performance indexes witnesses data distribution work well. Anyway, it is also evident there is the need of further improvements in the framework implementation towards a more advanced optimization of component layers.

The remainder of the paper is organized as follows. Section 2 reports on most relevant related work, while Section 3 outlines the proposed framework. A clarifying case study is presented in Section 4 to allow a deep understanding of the approach. Finally, experimental results are in Section 5 and Section 6 closes the paper.

2 RELATED WORK

Service-Oriented Architecture (SOA) is a scalable paradigm for service interoperability, flexibility and reusability both in military and civil domains. NATO research task group IST-118 has focused a strong interest about SOA, providing best practices to make it applicable at the operational and tactical level for the area of NATO Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) [1]. Standard SOA middleware is only partially applicable in tactical environments [2] [3], due to challenges for SOA implementation in highly dynamic, unpredictable network-centric environments, such as military ones [4]. The Internet of Things (IoT) has been having a strong effect in the military field. A Military IoT (MIOT) architecture has been proposed in [5] to realize “anytime, anyplace connectivity for anything, ubiquitous network with ubiquitous computing” in the military domain.

Basically, in the above scenarios data management has a central role. Both dissemination protocols and resource discovery mechanisms determine the concrete adoptability of possible middleware solutions in highly intensive decentralized systems [6]. Traditional middleware infrastructures for ubiquitous and pervasive computing fall into the node-centric paradigm for information distribution, sharing and for resource/service discovery. Nevertheless, the strong volatility and unpredictability of ubiquitous computing contexts make the most common approaches unfeasible or even not performant. Hence, more and more studies tended to support decentralization and not contrast it, promoting the opportunistic nature of communication, discovery and sharing of information. From this standpoint, Artificial Intelligence (AI) has been increasingly seen as a key technology enabling the so-called swarm intelligence. In particular, knowledge-based approaches have been envisioned as a possible mean to empower autonomic capabilities of nodes and develop a collective intelligence. Possible solutions proposed so far maintained their rudimentary nature due to very basic reasoning capabilities put in place by middleware components [7] [8]. On the contrary, distributed resource discovery exploiting automated reasoning algorithms [9] increases the node consciousness of their neighborhood. It allows managing approximate matches and returning an explanation of outcomes based on semantics of resource descriptions: a big leap forward in mobile and pervasive contexts.

An in-depth survey on existing middleware solutions can be found in [10]. In [11] a cross-layer middleware named MiMiNet (Middleware for Military Networks) has been developed to coordinate service infrastructures with the tactical router. In order to fulfill critical military scenarios such as battlefield surveillance, disaster recovery and emergency response, lightweight middleware for dynamic service/resource discovery and allocation are essential. In order to accomplish the three essential peculiarity of a multi-robot system -robustness, adaptability and scalability- [12] autonomous discovery is required. As such, resources like mobile robots have also to be autonomous without announcing their presence in the environment. Based on the concept of swarm intelligence [13] which mimics the cooperation action of socializing animals -*e.g.*, bees, birds, ants- the proposed approach exploits Bee Data Distribution System (Bee-DDS), a message-oriented middleware based on the publish-subscribe model. The proposal outlined in this paper aims to enhance the middleware capability with respect to existing publish/subscribe competitor systems. The vast majority of them are topic-based, where each event is labelled with one (in a set of predefined) topic, usually described via a static configuration [14]. Such systems have limits regarding a customized discovery, lacking the possibility of managing structured information about node characteristics. Additionally, the actual resource discovery mechanisms are often designed for static environments, fixed networks, therefore they are inadequate for mobile and pervasive computing. The latter suffer from limited computational resources and service volatility due to unpredictable device mobility and network link unreliability [15]. Semantically rich node annotations add machine-understandable meaning to resource characterizations. Furthermore, semantic-based discovery automatically matches requester's needs to the publishers' annotated resource descriptions.

3 SEMANTIC SWARM MIDDLEWARE ARCHITECTURE

The proposed framework introduces a knowledge-based approach for resource discovery, allocation and sharing in distributed swarm scenarios. In these contexts, large numbers of nodes are typically interconnected through publish/subscribe Message Oriented Middleware (pub/sub MOM) infrastructures. Exchanged messages are labeled with *topic* string specifying the type, structure and purpose of the message payload. Each node can act as a *publisher* to emit messages with a specific topic and/or as a *subscriber* to receive all messages related to a subscribed topic. The pub/sub MOM architecture grants robust deferred message broadcasting/multicasting in unreliable networks. Nevertheless, existing middleware solutions exploit trivial syntactic match of topics for service discovery. Conversely, the proposed framework allows the support for a dynamic semantic-based resource retrieval. This is achieved through the integration of additional functional layers to the standard MOM paradigm. As shown in Figure 1, the proposed framework comprises three layers: (i) *Bee-DDS*, an off-the-shelf pub/sub MOM; (ii) a *ubiquitous Knowledge Base*, a distributed model to collect ontology fragments disseminated among the devices deployed in the environment; (iii) *Resource/Service Discovery*, a decentralized collaborative resource/service discovery protocol exploiting non-standard inference services to enable a fine-grained categorization and ranking of resources matching a request. Details are described in what follows.

3.1 BEE Data Distribution Service

The basic inter-node communication primitives are implemented by *Bee-Data Distribution Service*[1] middleware. This software infrastructure provides services for real-time data distribution by adopting the publish/subscribe model in order to guarantee the delivery among the nodes of a network. Bee-DDS is suitable for a wide range of applications including surveillance, crisis management, ground segments, logistics, smart cities control, air traffic management and many more. As shown in Figure 1, the middleware includes **D**ata **L**ocal **R**econstruction **L**ayer (DLRL) and **D**ata **C**entric **P**ublish/**S**ubscribe (DCPS). The former comprises the standard interaction API between application objects and data coming from the DCPS level. DCPS defines entities, roles, interfaces and QoS policies for the publish/subscribe platform, as well as discovery techniques of communicating parties. In the off-the-shelf Bee-DDS platform, service discovery is carried out via basic string-matching of topics, whilst the implementation of additional layers proposed in this paper aims to make the middleware semantic-enabled, in order to increase effectiveness and flexibility.

3.2 Ubiquitous Knowledge Base model

The *ubiquitous Knowledge Base* (*u-KB*) layer grants transparent access to information embedded in semantic-enabled devices of the swarm. It derives from the classic Knowledge Base model in Knowledge Representation Systems based on Description Logics (DLs), which is denoted as a pair $\mathcal{K}=<\mathcal{T},\mathcal{A}>$. $\mathcal{T}$ is the Terminological Box (TBox or *ontology*), *i.e.,* the formal representation of the conceptual model of a fragment of reality through a hierarchy of *classes* (a.k.a. *concepts, i.e.,* sets of objects) and *properties* (a.k.a. *roles*) relating them with each other [16]. $\mathcal{A}$ is the Assertion Box (ABox) specifying the factual knowledge concerning a specific problem, with *individuals* as instances of classes. Software tools called *reasoners* a.k.a. *inference engine*s provide deduction services to derive implicit knowledge from what explicitly stated in a KB. Ontologies have been successfully used as part of expert and multiagent systems, as well as a core element in the Semantic Web, which proposes to extend the current Web to give information a well-defined and machine-interpretable meaning. In a u-KB, individuals are physically associated to distinct devices.

Even when using compression techniques, the size of typical ontologies may exceed the memory availability of mobile devices. Therefore, the ontology is fragmented in one or more *chunks* scattered across the network nodes. Different u-KBs can be managed by nodes in the same domain. The unambiguous association of every individual to its reference ontology is guaranteed by means of unique ontology URIs (Uniform Resource Identifiers). For a given reasoning task, the whole ontology is not typically required: in a u-KB, before starting inferences, the reasoner-equipped node retrieves only the ontology fragments needed for reasoning, based on classes and properties referenced by the involved resources.

In order to enable dissemination and on-the-fly ontology reconstruction, the class hierarchy is translated in a nested numbered list, where each class has a unique associated ID. The root of the ontology tree is represented by the *Thing* (a.k.a. *Top*) concept, having ID 1. Subclasses are sorted alphabetically and IDs differ from each other for the last number, which is incremented accordingly. For each nesting level an additional number is added to the superclass ID, separated by a “.” (dot). An example is sketched in Figure 2. The OWL[2] (Web Ontology Language) ontology file includes IDs as class annotation properties. The ontology partitioning starts from the *Upper Ontology* (UO) chunk, comprising the topmost levels in the class hierarchy. The UO depth level can be set based on size and complexity of the ontology itself.

---

[1] BEE Data Distribution System, http://sine.ni.com/nips/cds/view/p/lang/it/nid/211025/

[2] OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11 December 2012, https://www.w3.org/TR/owl2-overview/

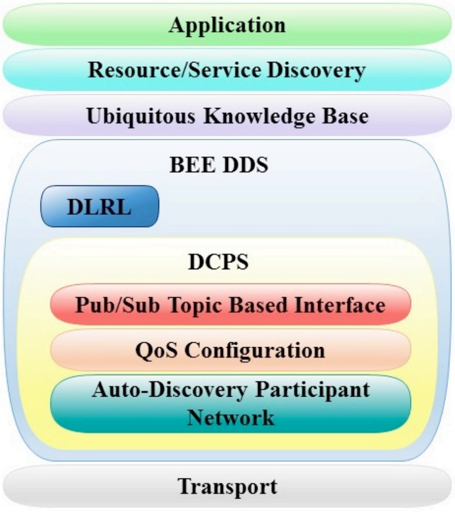


Figure 1: DDS Layered Architecture

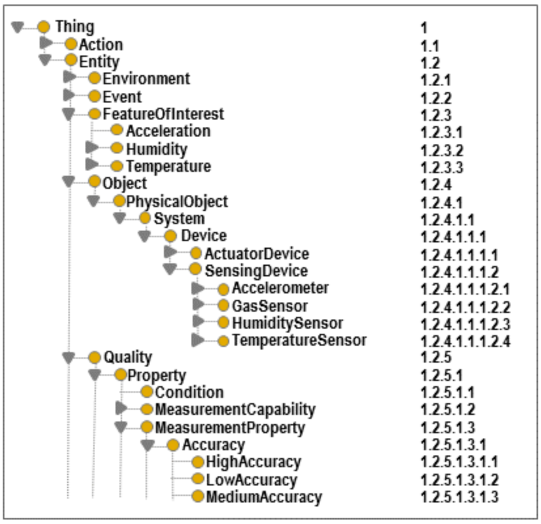


Figure 2: Ontology snippet with related class IDs

Each of the nodes scattered in the environment manages a cache of ontology chunks. Every cache contains the UO as well as the chunk(s) required by detained semantic resource annotations. Before the discovery phase, a requester node must recompose a subset of the whole ontology and intercept all the classes associated with the proper ancestors engaged in the semantic descriptions in order to enable the matchmaking process. Hence, it sends a message with the *BuildTBox* topic containing: (i) the ontology URI identifier, (ii) the list of needed class IDs, and (iii) the topic name to subscribe for reply (*e.g.*, *MergeOnto*_NodeID). All nodes should be subscribed to *BuildTBox* topic. If a node has one or more requested class IDs in its cache, it will publish on the above topic the compressed ontology chunk containing those classes. Requester node is subscribed to topic *MergeOnto*_NodeID to receive the ontology chunks and merge them.

3.3 Semantic resource/service discovery

A semantic-based service/resource request is a logic-based annotation expressed w.r.t. a reference ontology. The ontology URI implicitly defines the domain of the request. Resource/service discovery exploits the *Discovery* topic, which all semantic-enabled nodes should subscribe to. The requester starts inquiry by sending a *Discovery* message containing: (i) the reference ontology URI, (ii) the topic *SemAnn*_NodeID to be used in reply messages. Nodes subscribed to *Discovery* receive the request and check whether they own services related to the same domain. In that case, they reply with compressed service annotations on *SemAnn*_NodeID. Each annotation is further associated with a service-specific topic. The requester collects all service descriptions and compares them with the request through a semantic matchmaking process. The outcome consists of a ranked list of the best services. Finally, the requester uses the topic(s) associated to the selected service(s) in order to start fruition. In case of data gathering services, such as from sensors, the requester will act as a topic subscriber to receive information; on the other hand, controllable resources require the service user to be also a publisher on the service-specific topic in order to send commands and data.

While ontology and service collection phases are based on string matching of ontology URIs and class IDs, semantic-based matchmaking for service ranking and selection exploits non-standard inferences on an OWL 2 KB. Consider an ontology $\mathcal{T}$ and two concept expressions R (the request) and S (a service/resource description). Standard reasoning services for matchmaking allow only to detect *full matches* and *incompatibility* between resources. This usually achieves poor precision and recall, because full matches are rare and incompatibility is frequent when dealing with complex descriptions. The approach adopted here exploits non-standard inferences to enable a finer matchmaking, with support for approximate matches, resource ranking based on semantic affinity with request and formal explanation of outcomes. The matchmaking process was adapted from e-commerce scenarios [17]. If the conjunction R and S are incompatible (*i.e.,* a *partial match* occurs [17]), *Concept Contraction* determines what features G (for *Give up*) make R incompatible with S; if the requester accepts to retract G from R, a contracted version of the request K (for *Keep*) is obtained, which is compatible with S. Moreover, if S does not completely satisfy constraints in R (*i.e.,* a *potential match* occurs [17]), the *Concept Abduction* non-standard inference computes a concept H (for *Hypothesis*) such that the conjunction of S and H is a full match for R. In this way, retracting incompatible constraints can switch a partial match to a potential one, then hypothesizing underspecified characteristics can switch further to a full match. For both Concept Contraction and Abduction, minimality criteria for solutions are defined, which induce a *penalty* metric useful to measure the semantic distance of a concept toward another one. In the proposed framework, the overall relevance score of a service S w.r.t. a request R is computed as:

$$d(S,R) = 100\ [\ 1 - (\ penalty_c(S,R)+penalty_a(S,R)\ )\ /\ max\_penalty_a(R)\ ]$$

where $penalty_c$ and $penalty_a$ are the penalties induced by Contraction and Abduction, respectively. Penalty is normalized w.r.t. the maximum possible semantic distance from the request R, which is the one of the *Thing* concept and depends only on assertions in the reference ontology. Finally, score is converted to an ascending percentage scale for expressing semantic affinity rather than distance. The requester selects the available service with the highest score.

A flexible semantic-based approach improves standard service discovery and allotment [17]. In the proposed framework, the above non-standard inference services are implemented in the *Mini-ME*[3] matchmaker [9], which was designed for computationally constrained nodes. It works on KBs in the $\mathcal{ALN}$ (*Attributive Language with unqualified Number restrictions*) DL, a language offering moderate expressiveness and tractable complexity for both standard and non-standard inferences.

4 CASE STUDY: RESOURCE DISCOVERY IN SEMANTIC SWARM-BASED SENSOR NETWORKS

Functional and non-functional peculiarities of the proposal are clarified by means of a small illustrative case study. *A surveillance mission requires environmental temperature monitoring. The system consists of five nodes connected in a swarm-based sensor network: two temperature sensors (N1 and N4), a humidity sensor (N2), a floodlight drive (N3) and a mission controller (N5).*

The interactions in the system are based on the semantic-enhanced publish/subscribe middleware framework described in Section 4. As depicted in Figure 3, each node comprises a publisher for data dissemination through one or more Data Writer (DW) objects, allowing data to be published under a given topic, as well as a Subscriber for data gathering through one or more Data Reader (DR) objects, each associated to one topic subscription. In our case study, N1, N2, N3, N4 play the role of resource/service providers. In the initial system state, they all subscribe to general topics *BuildTBox* and *Discovery*; furthermore, each provided service has a specific topic associated via the respective node's Publisher (*Temp_N1*, *Hum_N2*, *Drive_N3* and *Temp_N4*). N5 acts as service requester, exploiting an on-board semantic matchmaker for logic-based service ranking. The semantic service discovery process is composed of the following interaction steps, marked with numbers referenced also in Figure 3.

1. **Ontology reconstruction**. N5 requires a temperature service whose semantic description is reported in the first data row of Table 1. Before starting service discovery, N5 sends its request on the *BuildTBox*

---

[3] http://sisinflab.poliba.it/swottools/minime/

topic according to the message request structure described in Section 3.2: `http://www.example.com/onto.owl{1.2.5.1.3.7.1;1.2.4.1.1.1.2.4;1.2.3.3;1.2.5.1.3.6.2;1.2.5.1.3.9.1;1.2.5.1.3.4.2;1.2.5.1.3.1.1}MergeOnto_N5`

2. Through the DR on the *BuildTBox* topic, N1, N2, N3 and N4 receive the metadata and check whether the URI in the request refers to some chunks of an ontology they own. If it does, they determine whether at least one item is in the list defined in their ontology chunk(s). In that case, a DW on *MergeOnto_N5* is created on-the-fly (dynamically created DWs and DRs are shown with a dashed outline in Figure 3) for sending selected chunk(s). In the example, N1, N2 and N4 reply, whereas N3 does not manage the requested ontology.

3. Through the DR on *MergeOnto_N5*, N5 receives the needed ontology chunks and merges them in order to rebuild a minimal self-contained subset of terminology for matchmaking.

4. **Semantic-based service discovery**. Now N5 can forward its service request on the *Discovery* topic according to the message request structure described in Section 3.3: `uri=http://www.example.com/onto.owl;semanticTopic=SemAnn_N5`

5. Through the *Discovery* DR, N1, N2, N3 and N4 receive the metadata and check whether the URI specified in the request is the same of their service description(s). N3 has no service described by the specified vocabulary, while the check succeeds for N1, N2 and N4 and they become publishers on the *SemAnn_N5* topic. For example, N1 replies: `topic=Temp_N1;semanticAnnotation=[...]`

6. N5 gets the messages of N1, N2 and N4 and executes the matchmaking process between the annotated request and the semantic descriptions of discovered services. Table 1 shows descriptions (in classical DL notation) and matchmaking scores. The best match (*i.e.,* lowest semantic distance) is achieved by N1, so N5 becomes subscriber on *Temp_N1* topic for receiving temperature data from the sensor exposed by N1.

The basic interaction sequence and mechanisms of the proposed framework were described with a tiny example, but in complex real scenarios –where nodes expose more services described through several domain vocabularies– the resource/service discovery carries out in the same way.

Table 1: Matchmaking outcome

Semantic Description	**Score**
N5 (request): *TemperatureSensor ⊓ ∀observes.Temperature ⊓ ∀hasMeasurementProperty. (HighAccuracy ⊓ LowMeasurementRange ⊓ LowFrequency ⊓ HighPrecision ⊓ HighResponseTime)*	N.A.
N1: *TemperatureSensor ⊓ ∀observes.Temperature ⊓ ∀hasMeasurementProperty. (HighAccuracy ⊓ LowFrequency ⊓ MediumMeasurementRange ⊓ HighPrecision ⊓ MediumResponseTime ⊓ MediumResolution ⊓ LowLatency)*	92,6
N4: *TemperatureSensor ⊓ ∀observes.Temperature ⊓ ∀hasMeasurementPropert. (LowAccuracy ⊓ LowFrequency ⊓ LowMeasurementRange ⊓ LowPrecision ⊓ MediumResponseTime ⊓ LowResolution ⊓ LowLatency)*	85,2
N2: *HumiditySensor ⊓ ∀observes:Humidity ⊓ ∀hasMeasurementProperty.(LowAccuracy ⊓ LowFrequency ⊓ LowMeasurementRange ⊓ MediumPrecision ⊓ MediumResponseTime ⊓ LowResolution ⊓ LowLatency)*	71,4

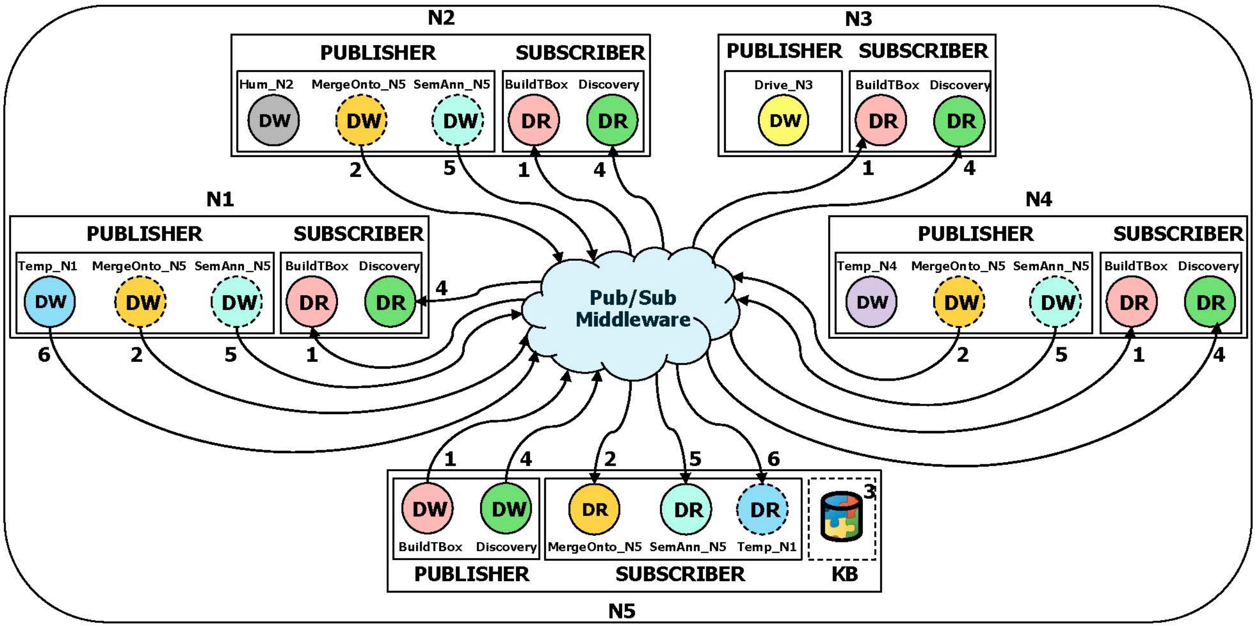


Figure 3: Case study - Temperature service discovery

5 EXPERIMENTS

The proposed framework was implemented in Java language as a set of integrated modules of the Bee-DDS software to evaluate the feasibility of the approach. Preliminary tests were executed in a small scenario with a single service requester node and 50 resource/service providers connected through the Bee-DDS middleware. These tests (results not reported) were performed in order to tune two system configuration variables: (i) the taxonomic depth of the shared upper ontology chunk and (ii) the algorithm for the compression of ontology chunks and semantic service annotations exchanged among system nodes. For the UO, 2, 3 and 4 were considered as possible depth values, while *COX* [18] and *EXI*[4] were tested as encoding algorithms.

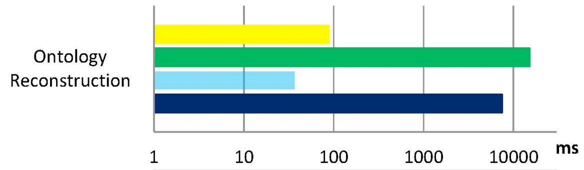
Subsequent experiments were conducted with EXI as data compression algorithm and a UO depth level of 4, in accordance with the best trade-off between the desired UO detail level and the time required for message encoding. For these tests, the considered scenario was composed of 10 service requesters and 500 resource providers. Requester nodes were distributed on 3 virtual machines with dual-core CPU, 2 GB RAM, 32-bit Ubuntu 14.04 LTS operating system and 32-bit Java 8 SE Runtime Environment (build 1.8.0 72-b15). Provider nodes were equally deployed among 50 virtual machines with dual-core CPU, 800 MB RAM, same operating system and Java runtime environment. All host machines were connected to a 100 Mb/s IEEE 802.3 network. This scenario was designed in order to simulate a realistic workload characterized by several loosely-coupled components distributed on different machines and interconnected through the middleware. The performance metrics analyzed for the 500:10 test were turnaround time and RAM usage for both ontology reconstruction and resource discovery/allotment phases.

Turnaround time results related to the ontology rebuilding phase are reported in Figure 4: the system took about 22.67 seconds on average for the reconstruction of an ontology subset containing the classes used for reasoning. Turnaround time results for resource allotment are shown in Figure 5: the system took 20.20 seconds for this stage. For both phases, the longest sub-task is message transmission and this may depend on the communication network physical properties and the middleware configuration. Results for RAM usage

---

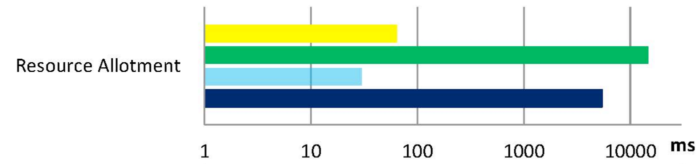
[4] Efficient XML Interchange (EXI) Format 1.0 (Second Edition), W3C Recommendation 11 February 2014, https://www.w3.org/TR/exi/

for KB creation and discovery are illustrated in Figure 6: memory usage peak for requesters was always below 30MB for KB creation and approximately 43 MB for the discovery phase, while for providers RAM occupancy peak was about 23 MB in both phases. The higher RAM occupancy for requester than provider nodes is due to the reasoner, which enables requesters to execute the matchmaking process.



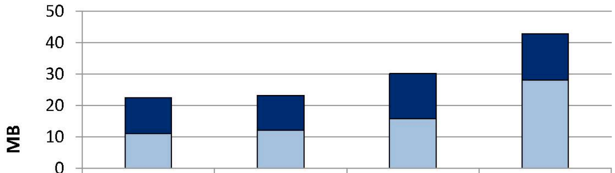
	Ontology Reconstruction
🟨 Compression	87.84
🟩 Transmission	15066.79
🟦 Decompression	36.78
■ Chunk Merging	7471.45

Figure 4: Time for ontology reconstruction (500:10 test)



	Resource Allotment
🟨 Compression	63.18
🟩 Transmission	14714.92
🟦 Decompression	29.88
■ Semantic Matchmaking	5397.01

Figure 5: Time for resource allotment (500:10 test)



	KB Creation	Discovery	KB Creation	Discovery
	Providers		Requesters	
■ Non-heap Memory	11.36	11.07	14.42	14.79
□ Heap Memory	11.12	12.17	15.74	28.05

Figure 6: Memory usage peak (500:10 test)

Tests evidenced the feasibility of the proposed framework in a realistic scenario revealing an impact basically manageable of semantics on communication and computation. On the other hand results highlighted the relevance of endowing the middleware with support for dynamic semantic-based service discovery, given the reached flexibility not comparable with standard approaches.

6 CONCLUSION AND FUTURE WORK

The paper proposed a layered architecture where a semantic-based discovery enriched resource allotment in swarm systems, *i.e.,* scenarios populated by a large number of resource-constrained nodes. The envisioned framework leveraged an off-the-shelf publish/subscribe middleware for inter-node communication. A logic-based annotation of both available resources and node needs enabled approximate, deductive matchmaking as well as ranking of results by semantic compliance. The proposal includes an opportunistic technique for terminology decomposition and rebuilding across multiple nodes just for reasoning on a given set of annotations. Both ideas and approach have been implemented in an early testbed to prove both correctness and feasibility of the proposal in a practical case study.

Results show interesting behaviour even if further performance evaluation and improvement are needed. Future work also concerns enrichment of semantic-based features to be added to the system, including resource composition, clustering, substitution and requester/provider negotiation.

REFERENCES

[1] NATO IST-118. 2013. “IST-118 SOA recommendations for Disadvantaged Grids: Tactical SOA Profile, Metrics and the Demonstrator Development Spiral”, *SCI-254 Symposium on “Architecture Assessment for NEC”*, Estonia, 14-15 May, 2013.

[2] K. Foster, “Middleware Interoperability in Tactical Defence Environments-Smartphone Integration in the Net Warrior Initiative”, *International Conference on Knowledge-based and Intelligent Information & Engineering Systems (KES)*, pp. 1646-4655, 2012.

[3] M. Manso, J. M. A. Calero, C. Barz, T. H. Bloebaum, K. Chan, N. Jansen, F. T. Johnsen, G. Markarian, P-P. Meiler, I. Owens, J. Sliwa, Q. Wang, “SOA and Wireless Mobile Networks in the tactical domain: Results from experiments”, *Military Communications Conference, MILCOM*. IEEE, pp. 593-598, 2015.

[4] J. Sliwa, M. Amanowicz, “Success factors for SOA implementation in network centric environment”, *Journal of Telecommunications and Information Technology*, pp 43-53, 2011.

[5] L. Yushi, J. Fei, Y. Hui, “Study on application modes of military internet of things (MIOT)”, *IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 3, pp. 630-634, 2012.

[6] F. T. Johnsen, T. Hafsøe, and M.Skjegstad, “Web services and service discovery in military networks”, *14th International Command and Control Research and Technology Symposium (ICCRTS)*, Washington D.C., US, June 2009.

[7] M. Franke, C. Seidl, T. Schlegel, “A seamless integration, semantic middleware for cyber-physical systems”, *10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 627-632, 2013.

[8] C. Yu, D. Yao, X. Li, Y. Zhang, L. T. Yang, N. Xiong, H. Jin, “Location-aware private service discovery in pervasive computing environment”, *Information Sciences*, vol. 230, pp. 78-93, Elsevier, 2013.

[9] F. Scioscia, M. Ruta, G. Loseto, F. Gramegna, S. Ieva, A. Pinto, E. Di Sciascio, “A mobile matchmaker for the Ubiquitous Semantic Web”, *International Journal on Semantic Web and Information Systems*, volume 10, number 4, pp. 77-100, 2014.

[10] V. Raychoudhury, J. Cao, M. Kumar, D. Zhang, “Middleware for pervasive computing: A survey”, *Pervasive and Mobile Computing*, vol. 9, no. 2, pp. 177-200, Elsevier, 2013.

[11] N. Jansen, D. Kramer, C. Barz, J. Niewiejska, M. Spielmann, “Middleware for coordinating a tactical router with SOA services”, *International Conference on Military Communications and Information Systems (ICMCIS)*, pp.1-7, 2015.

[12] R. H. Qadir, *Self-Sufficiency of an Autonomous Reconfigurable Modular Robotic Organism*, Springer, 2015.

[13] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*, John Wiley & Sons, 2006.

[14] G. Fortino, A. Guerrieri, M. Lacopo, M. Lucia, W. Russo, Wilma, “An agent-based middleware for cooperating smart objects”, *Highlights on Practical Applications of Agents and Multi-Agent Systems*,

pp. 387-398, Springer, 2013.

[15] N. A. Saadon, R. Mohamad, “Semantic-based Discovery Framework for Web Services in Mobile Computing Environment”, *Jurnal Teknologi*, vol. 77, no. 9, 2015.

[16] S. Staab, R. Studer, “Handbook on Ontologies”, *International Handbooks on Information Systems*, Springer, 2004.

[17] M. Ruta, E. Di Sciascio, F. Scioscia, “Concept Abduction and Contraction in Semantic-based P2P Environments”, *Web Intelligence and Agent Systems*, vol. 9, n. 3, pp. 179-207, 2011.

[18] F. Scioscia, M. Ruta, “Building a Semantic Web of Things: issues and perspectives in information compression”, in *Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009)*, pp. 589-594, 2009.

[19] M. Ruta, F. Scioscia, and E. Di Sciascio, “Enabling the semantic web of things: framework and architecture”, in *Sixth IEEE International Conference on Semantic Computing (ICSC 2012)*. IEEE, 2012, pp. 345-347, doi: 10.1109/ICSC.2012.42.