Semantic-Based Context Mining and Sharing in Smart Object Networks

Eliana Bove

DEI - Politecnico di Bari
Via E. Orabona 4
Bari, Italy I-70125
Email: eliana.bove@poliba.it

***Abstract*—The Semantic Web of Things is the evolution of Internet of Things paradigms introducing novel Knowledge Base models, in order to associate semantic representation to real-world objects and events. The paper proposes a semantic-based approach for high-level information representation, knowledge discovery, allotment and sharing in distributed scenarios populated by smart objects. By leveraging the integration of standard supervised machine learning techniques with non-standard semantic-based reasoning services, smart objects annotate in a fully automatic way the context they are in and expose their acquired knowledge to the outside world as in a blog, exploiting a layered architecture built on a publish/subscribe Message Oriented Middleware. The feasibility of the envisioned framework is supported by a case study and an early experimental campaign.**

***Keywords*–Logic-based matchmaking; Pervasive Computing; Machine Learning; Ubiquitous Knowledge Base; Mobile Resource Discovery.**

I. Introduction

Increasingly available *Internet of Things* (IoT) technologies are enabling the pervasive computing paradigm, where information is really scattered in a given environment in the form of atoms which deeply permeate the context [1]. Heterogeneous data streams must be continuously retrieved and locally processed by mobile ad-hoc networks of *smart objects* dipped in the environment in order to detect events of interest in observed areas. A smart object [2] is an intelligent software agent acting on a mobile device, equipped with embedded sensors, actuators, communication ports as well as limited computation and storage facilities. Each smart object describes itself and the context where it operates toward a variety of external devices and IoT applications. The interoperability and the relevance of the IoT could be further enhanced by associating semantically rich (compact) descriptions to real-world objects and to data they retrieve, so featuring novel classes of smart applications. This is the so-called *Semantic Web of Things* evolution of classic Internet of Things paradigms. This paper proposes a novel semantic-based framework for knowledge high-level representation, discovery and sharing within smart object networks in the Semantic Web of Things. By leveraging the integration of standard supervised Machine Learning (ML) techniques with non-standard semantic-based inference services [3] on annotations in Semantic Web languages, smart objects become able to annotate in a fully automatic way the context they are in, continuously enriching their basic descriptive core according to events they detect and exposing them to the outside world as in a blog. Identification and sensing information are expressed in OWL 2 (Web Ontology Language) annotations [4] via a semantic-based evolution of standard *k Nearest Neighbors* (k-NN) ML algorithm. For knowledge sharing in smart object networks, the proposed framework interconnects distributed components by exploiting the publish/subscribe (pub/sub) Message-Oriented Middleware (MOM) architectural model. In detail, the topmost layer provides resource/service discovery based on standard and non-standard inference services for semantic matchmaking. It enables a fine-grained categorization and ranking of resources matching a request. An optimized Description Logic (DL) reasoner for mobile and embedded devices [5] is integrated for this purpose. The middle layer is a distributed collaborative protocol to collect ontology fragments (*chunks*) disseminated among the devices in an environment, in order to rebuild a minimal ontology core needed for supporting inference procedures on a particular set of semantic annotations. As ontologies can be large and Semantic Web languages use the verbose XML syntax, both compression and ontology partitioning are needed. The proposal adopts a novel scheme for rebuilding partitioned ontologies. It seeks a practical trade-off between the size of individual ontology chunks managed by devices –also exploiting compressed encoding– and the number of message exchanges required for on-the-fly reassembly. This layer implements a *ubiquitous KB* (u-KB) [6] model, where a node of the distributed system endowed with a reasoner fetches on the fly all and only the KB parts required for the current inference problem. Finally, the lowest layer is a message-oriented middleware based on the publish-subscribe model. It provides reliable communication among loosely-coupled components to support functionalities of the higher-level layers. The proposed approach results as a general-purpose, cross-domain semantic-based context mining, knowledge discovery and sharing facilitator among pervasive smart devices. In order to evaluate the usefulness of the proposed theoretical approach in a real scenario, the framework has been implemented in a prototypical smart farmer robot team. The proposal makes every entity involved in the scenario able to summarize the information gathered via its sensing interfaces into a semantically annotated description of the environment and relevant objects in it. Furthermore, robots can interact and communicate with each other by leveraging the proposed knowledge sharing approach which integrates a scalable off-the-shelf middleware as pub/sub communication layer, namely Bee Data Distribution System (Bee-DDS) [7]. A prototype was implemented and tested in experimental evaluations, to ensure correctness of the approach and perform a preliminary performance evaluation. The remainder of the paper is organized as follows. Section II provides a survey of related work. Section III discusses the proposed framework in detail. An illustrative case study is described in Section IV to allow a better understanding of the proposal. Finally, experimental results are in Section V and Section VI closes the paper.

II. Related Work

A smart object is a software agent able to process and analyze sensory raw data and further combine data classifications to identify patterns, situations and events. It can collect data sources either by exploiting on-board sensors or querying short-range wireless communication protocols. The interpretation of raw, low-level gathered data and the behavior adaptation characterize different works existing in literature. Threshold detectors or standard ML techniques are exploited by current event classification approaches [8]. This paper is based on the integration between low-level data analysis and the high-level context interpretation to trigger actions, assume decisions or make interventions on the environment. Although noisy, uncertain and incomplete sensor data are well handled by probabilistic learning models, several limitations such as scalability, ad-hoc static models and data scarcity characterize these models. An event detection fuzzy logic approach based on a rule-base was presented in [9] in order to overtake the low level of model accuracy leveraging crisp threshold values. However, in addition to high accuracy, another important requirement of a smart object is the computational efficiency for working on pervasive computing platforms. Ontology based reasoning approaches for home and office activity recognition were presented in [10] and [11], respectively. They support only full matches and this is a limit in pervasive scenarios featured by several heterogeneous information sources. This paper merges the strengths of Machine Learning and high-level semantic interpretation in order to depict and detect more complex context state. Useful classification surveys about different particularities of ML techniques are in [12]. Particularly, good accuracy, insensitivity to outliers, high performance with both nominal and numerical features and incremental learner characteristics make the k-NN algorithm very useful for smart objects. In recent years, the Semantic Web research community dealt with the task of describing sensor features and recover data through ontologies. The most relevant and widely used vocabularies are *OntoSensor* [13] and *SSN-XG* [14]. Both are general enough to cover different application domains, unfortunately they are too large and complex to be processed by a single node in pervasive computing contexts where semantic-based knowledge sharing among different smart objects is required for auto-coordination and collaboration. Therefore, strategies for modularizing terminologies are necessary. Solutions in literature are strongly influenced by the specific applications. In general, the issue faced on this paper is somewhat different from the above classical ontology modularization, as it requires a dynamic, problem-oriented approach compatible with resource-constrained devices. [15] presents a relevant work enabling ontology decomposition and run-time rebuilding based on service instance descriptions, albeit giving slightly less flexibility in run-time ontology distribution. Furthermore, by construction the framework supports only semantic matchmaking based on Subsumption, preventing inference services which evaluate non-full matches. In order to derive implicit information starting from explicit event and context detection, supporting approximate matches and service ranking metrics is very important. That is why ubiquitous logic-based matchmakers implementing non-standard inference services [5] appear as enabling technologies in mobile and pervasive contexts. The knowledge-based information sharing needed for smart object cooperation is achieved by exploiting middleware software infrastructure. Different semantically enriched middleware platforms exist in literature [16][17], however, they take into account only full matches, which are quite rare in complex pervasive domains. The current proposal aims to a more principled and general solution supporting distributed knowledge representation, management, sharing and discovery in pervasive context where mobile computing devices provide minimal computational capabilities and where the exploitation of logic-based and approximate discovery strategies manage non-full matching results, typical in these scenarios.

III. Proposed Approach

The proposed framework introduces a semantic-based approach for knowledge high-level representation, discovery and sharing in distributed smart object networks. The approach aims to: (i) characterize the descriptive core of each smart object in a fully automatic way starting from sensed context data; (ii) share the learned semantic-based knowledge within the network and (iii) achieve objects cooperation for triggering actions, taking decisions or making interventions on the environment.

The starting point is represented by raw data collected by object sensors. Each object processes data and produces an annotation in a semantically rich formalism grounded on the Attributive Language with unqualified Number restrictions ($\mathcal{ALN}$) Description Logics [18], which is a subset of OWL 2 language. In order to guarantee the delivery of this information within the network, the proposed approach includes a layered architecture where a semantic-based knowledge discovery supports resource allotment in scenarios populated by a large number of resource-constrained nodes. The envisioned framework exploits a pub/sub MOM for inter-node communication. In what follows, proposed framework details are provided.

A. Data Mining and Semantic Annotation

Each intelligent entities continuously executes three steps, as shown in Fig. 1.

1. Clustering: adopts an unsupervised clustering approach to

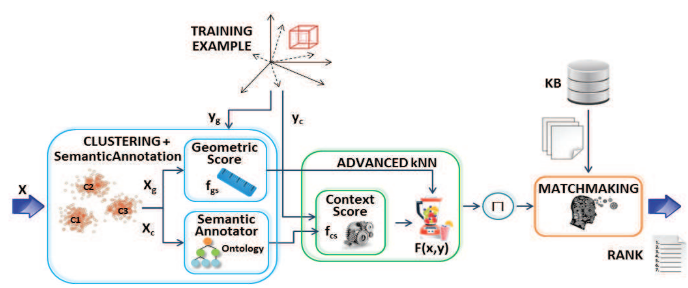


Figure 1. Sketch of the log information flow

pre-process input data. The previous knowledge of the object is represented by a training data set grouped into clusters. Each cluster is characterized by two components: geometry and context. *Geometry* describes data through statistical parameters. *Context* annotates data w.r.t. an OWL 2 reference ontology, which is different for each application domain. An unknown input instance is linked with the description of the nearest cluster [19].

2. Advanced k Nearest Neighbors: exploits an enhanced version of the *k-NN* algorithm to give high-level data representation. A semantic-based similarity measure f_{cs} (context distance) and a partial scores f_{gs} deriving from geometry (i.e., quantitative statistical attributes) are combined by a utility function F. A peculiar aspect of the approach proposed here consists in the integration of classic k-NN supervised machine learning with semantic-based matchmaking. In what follows, x and y are input arguments of F: the first is the instance to be examined and the latter represents each element of the training set. Both are described by two components: the geometric (x_g, y_g) and the contextual (x_c, y_c) ones.
The *geometric score* $f_{gs}(x_g, y_g)$ numerically expresses the similarity between y_g and x_g, as proposed in [20]. This numerical assessment is referred to the statistical distribution parameters featuring the data point and the training examples. Since x_g is the value to be matched, only the k dimensions describing x must be taken into account. Therefore, a basis vector $B(x_g) = \langle b_1, b_2, \ldots, b_k\rangle$ is defined, where $b_i \in [0, 1]$ and $b_i = 0 \Leftrightarrow x_{g_i} = \emptyset$. The matching value on a single dimension is computed as:

$$dmatch(x_{g_i}, y_{g_{ji}}) = \begin{cases} \frac{|x_{g_i} \cap y_{g_{ji}}|}{|x_{g_i}|} & if\ B(x_{g_i}) = 1 \land \\ & B(y_{g_{ji}}) = 1 \\ 0 & else \end{cases} \tag{1}$$

According to (1), the value $dmatch(x_{g_i}, y_{g_{ji}})$ is computed by determining the overlap between x_{g_i} and $y_{g_{ji}}$ (the i-th dimension of the j-th training example) divided by the length of x_{g_i}. The overall matching score is defined as:

$$f_{gs}(x_g, y_g) = 1 - \frac{\sum_{i=1}^{k} dmatch(x_{g_i}, y_{g_{ji}})}{k} \tag{2}$$

Division by k produces normalization w.r.t. the highest cardinality of x_g.
The *contextual metric* $f_{cs}(x_c, y_c)$ is calculated on features annotated in *OWL 2* language [4] according to the reference terminology and exploits non-standard inference services presented in [3]. Concept Abduction and Concept Contraction non-standard inferences are used in order to consider non-full matches. Given an ontology $\mathcal{T}$ and two concept expressions A and B (acting as resource and request descriptions, respectively), if the conjunction $A \sqcap B$ is unsatisfiable w.r.t. the ontology $\mathcal{T}$, i.e., A, B are not compatible with each other, Concept Contraction determines what features G (for *Give up*) can be retracted from B to obtain a subset K (for *Keep*) such that $K \sqcap A$ is satisfiable in $\mathcal{T}$, and returns a value $penalty_{(c)}$ representing the associated semantic distance. Futhermore, if $\mathcal{T} \not\models A \sqsubseteq B$ then Concept Abduction computes a concept H (for *Hypothesis*) such that $\mathcal{T} \models A \sqcap H \sqsubseteq B$. That is, H represents what should be hypothesized (i.e., is underspecified) in A in order to completely satisfy B w.r.t. the information modeled in $\mathcal{T}$. Concept Abduction provides a related distance metric named $penalty_{(a)}$. Given these premises, the contextual score is calculated as:

$$f_{cs}(x_c, y_c) = \frac{\omega \cdot penalty_{(c)} + (1-\omega) \cdot penalty_{(a)}}{max\ penalty_{(a)}} \tag{3}$$

using as normalizing factor the maximum possible semantic distance, which is the one between x_c and the most generic $\top$ concept. The scoring mechanism is tuned by ω, which depends on the geometric score and is computed as $\omega = \delta \cdot f_{gs}(x_g, y_g)$ with the proportional factor $\delta \in [0.8, 1]$ and the weight ω, which emphasizes explicit incompatibility measured by Contraction as geometric distance increases.
3. Semantic-based matchmaking: leverages inference services in [3] to compare the semantic characterization of the context with descriptions of instances in the object Knowledge Base (KB), so giving a semantic interpretation to the raw data. The *overall distance* F is computed as:

$$F(x, y) = (f_{gs}(x_g, y_g) + \epsilon)^{\alpha} \cdot (f_{cs}(x_c, y_c) + \gamma)^{1-\alpha} \tag{4}$$

It is a monotonic function in $[0, 1]$ and ranks input training examples in a consistent way. It basically adopts a user-friendly scale distance, where lower outcomes represent better results. In a great detail, $\alpha \in [0, 1]$ factor determines the relative weight of contextual and geometric scores. In case of contextual or geometric full matches, score is tuned by means of $\epsilon \in [0, 1]$ and $\gamma \in [0, 1]$, respectively. Each new data point or series acquired in the same observation window undergoes this process which terminates when the latest data points are integrated in the training set, while data older than a purging threshold are removed.

B. Knowledge Discovery

Throughout the objects lifetime, the semantic endowment is progressively enriched and completed so that it could be exposed to the outside world as in a blog. To achieve this, the proposed system exploits a pub/sub MOM. In this paradigm, topics of exchanged messages specify 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. In conventional pub/sub MOM architectures, resource discovery occurs through syntactic match of topics. Conversely, the proposed framework allows the support for a dynamic semantic-based resource retrieval. This is realized through the integration of additional functional layers to the standard MOM paradigm. As shown in Fig. 2, the proposed approach includes three layers: (i) Bee-DDS, an off-the-shelf pub/sub MOM; (ii) a ubiquitous Knowledge Base, a distributed model for information partitioning and on-the-fly materialization; (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.
1. Bee Data Distribution Service: it provides services for real-time data distribution by adopting the publish/subscribe model in order to guarantee the basic inter-node communication. Its software infrastructure comprises Data Local Reconstruction Layer (DLRL) and Data Centric Publish/Subscribe (DCPS).
2. Ubiquitous Knowledge Base Model: transparent access to information embedded in semantic-enabled devices of the network is granted by the *ubiquitous KB* (u-KB) layer. KB is partitioned in a decentralized way and scattered across multiple nodes. Specifically, the Terminological Box $\mathcal{T}$ (i.e., the ontology) is fragmented in one or more *chunks* managed by multiple distributed nodes. Individuals in the Assertion Box $\mathcal{A}$ are not centrally stored, but disseminated in the environment as they make part of the endowment of each node. Due to the generality of the proposed approach, all nodes within the same network can manage any domain ontology, even using multiple

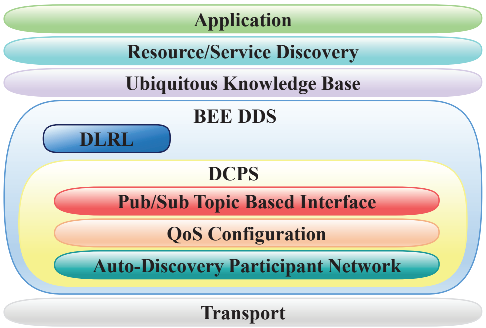


Figure 2. Bee DDS Layered Architecture

vocabularies in order to cover different application domains. Furthermore, the use of unique ontology Uniform Resource Identifiers (URI) ensures that all objects working with the same reference ontology can share parts of the u-KB dynamically without requiring preliminary agreement among them. In order to enable dissemination and on-the-fly ontology reconstruction, the ontology partitioning is based on associating each class with a unique ID, computed from its position in the taxonomy. The most generic class, named *Thing* in OWL 2 (*a.k.a. Top* or $\top$ in DL notation), takes ID 1. Each nesting level adds a further numerical suffix, separated by a **.** (dot). The ontology partitioning starts from an 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. Every node cache contains the UO as well as the chunk(s) required by detained semantic resource annotations. Before the discovery phase, a requester node must rebuild a subset of the ontology containing the classes used in the logical expressions of the involved annotations.To do so, it publishes a message with the *BuildTBox* middleware topic, which all semantic-enabled nodes must be subscribed to. The message contains: (i) the unique ontology URI, (ii) the list of requested class IDs, and (iii) the topic name (e.g., *MergeOnto_NodeID*) to be used in reply messages. 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. Semantic Resource Discovery: discovery is based on a semantic resource request which consists of a logic-based annotation expressed w.r.t. a reference ontology. The requester starts inquiry by sending a *Discovery* message containing: (i) the reference ontology URI which implicitly defines the request domain, (ii) the topic *SemAnn_NodeID* to be used in reply messages. Through the *Discovery* topic, other nodes receive the request and check whether they own resources related to the same domain. Only in this case, nodes become publishers on the reply topic and send back the related compressed annotations; each annotation is associated with a resource-specific topic. The requester collects all descriptions and compares them with its request through the semantic matchmaking process described in [3] and recalled hereafter. The outcome of the match determines a ranked list of resources which best satisfy the request. Finally, the requester uses the topic(s) associated to the selected resource(s) in order to start fruition. In case of data gathering resources, such as from sensors, the requester will act as a subscriber to receive information; on the other hand, controllable resources require the service user to be a publisher on the topic to send commands and data. As for the above mining approach, this layer exploits non-standard inferences for semantic matchmaking implemented in the *Mini-ME* reasoning engine [5], which is suitable for computationally constrained nodes.

IV. CASE STUDY: SAVE THE GRAPES

The proposed approach has a strong potential impact in supporting a wide range of applications including urban search and rescue, personal assistance, industrial maintenance, home automation, smart agriculture and many more. The case study proposed here is related to the smart agriculture field, where objects can share information in order to monitor crops by means of appropriate sensors or fulfill a product tracking system able to follow them from the farm to seller shelves. In order to evaluate the usefulness of the framework, a prototypical testbed is under development, exploiting a semantic distributed sensor network and a *3D Robotics Iris* drone [21] equipped with additional sensors and peripherals. The cooperation of these entities allows to detect the specific agricultural context state, formulate plans to reach the mission goals and act accordingly. In what follows, an illustrative example is presented to clarify functional and non-functional aspects of the proposal.

Downy mildew is a serious fungal disease of grapevine which can result in severe crop loss. It is caused by the fungus Plasmopara viticola. *The pathogen attacks all green parts of the vine, especially the leaves. In order to eliminate the fungus, a smart vine monitoring is realized by analyzing environmental parameters collected by a sensor network. According with this monitoring, a smart farming drone is able to automatically infer when, where and how spraying fungicides on susceptible cultivars.*

Environmental factors that influence development of

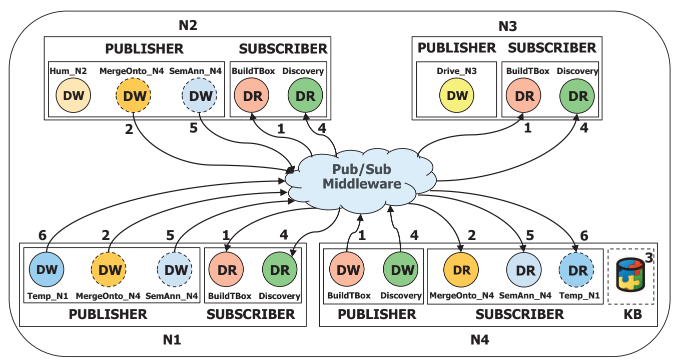


Figure 3. Temperature service discovery

Plasmopara viticola include relative humidity, atmospheric pressure, soil moisture, leaf wetness, rugged soil temperature, sun calibration quantum, meteorological data. Raw data are processed on the fly by the smart farmer drone leveraging the proposed semantic-based approach. As shown in Fig. 3, each node includes a Publisher for data dissemination, through one or more Data Writer (DW) objects and a Subscriber for data gathering through one or more Data Reader (DR) objects, each associated to one Topic subscription. N4 represents the drone that acts as a requester of knowledge about the environment in order to act on it. N1 is a temperature sensor, N2 is a humidity sensor and N3 is a cutter drive. They are distributed in the monitored area and play the role of resource 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 Publisher (*Temp_N1, Hum_N2, Drive_N3*). The knowledge discovery process is composed by the following interaction steps, also marked in Fig. 3.

1. N4 requires a soil temperature service, with high accuracy and precision, low measurement range and frequency, and high response time.

$SoilTempSensor \sqcap \forall observes.SoilTemp \sqcap \forall hasMeasuremProp.(HighAccuracy \sqcap HighPrecision \sqcap LowFrequency \sqcap LowMeasuremRange \sqcap HighResponseTime)$

Before starting service discovery, N4 sends its request on the BuildTBox topic.

2. Through the DR on the BuildTBox topic, N1, N2 and N3 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_N4 is created on-the-fly (dynamically created DWs and DRs are shown with a dashed outline in Fig. 3) for sending selected chunk(s). In the example, N1 and N2 reply, whereas N3 does not manage the requested ontology.

3. Through the DR on MergeOnto_N4, N4 receives the needed ontology chunks and merges them to rebuild a minimal self-contained terminology subset for matchmaking.

4. N4 forwards its service request on the Discovery topic.

5. N1, N2 and N3 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 and N2 and they become publishers on the SemAnn_N4 topic.

N1: $SoilTempSensor \sqcap \forall observes.SoilTemp \sqcap \forall hasMeasuremProp.(HighAccuracy \sqcap LowFrequency \sqcap MediumMeasuremRange \sqcap HighPrecision \sqcap MediumResponseTime \sqcap MediumResolution \sqcap LowLatency)$

N2: $HumSensor \sqcap \forall observes.Humidity \sqcap \forall hasMeasuremProp.(LowAccuracy \sqcap LowFrequency \sqcap LowMeasuremRange \sqcap MediumPrecision \sqcap MediumResponseTime \sqcap LowResolution \sqcap LowLatency)$

6. N4 gets the messages of N1 and N2 and executes the matchmaking process between the annotated request and the semantic descriptions of discovered services. The best match (i.e., lowest semantic distance) is achieved by N1, while N2 is less relevant as a sensor, because its observed quantity is incompatible. So, N4 becomes subscriber on Temp_N1 topic for receiving temperature data from the sensor exposed by N1.

N4 executes these steps for all environmental data needed to detect the monitored area state. Furthermore, the drone exploits the proposed mining approach to analyze data collected by each sensor and to determine high-level feature values of the monitored factors for the whole observation window.

Soil temperature semantic-based classification value is calculated considering not only quantitative statistical parameters (mean, variance, kurtosis, skewness), but also relevant context features ($Altitude$, $Latitude$, $Season$; $PartOfDay$). Clusters are $VeryLowSoilTemp$, $LowSoilTemp$, $MediumSoilTemp$, $HighSoilTemp$ and $VeryHighSoilTemp$. By replicating this process for each sensed parameter, the smart object (e.g., the drone) creates a high-level representation of the considered grapevine status. A semantic description detected by the system follows as an example.

$Grapevine \sqcap \forall hasSoilTemp.LowSoilTemp \sqcap \forall hasAtmosphPressure.(VeryLowAtmosphPressure \sqcap \neg HighAtmosphPressure) \sqcap \forall hasHumidity.HighHumidity \sqcap \forall hasSoilMoisture.HighSoilMoisture \sqcap \forall hasLeafWetness.(VeryHighLeafWetness \sqcap \neg LowLeafWetness) \sqcap \forall hasSunCalibQuantum.HighSunCalibnQuantum \sqcap \forall hasRuggedSoilTemp.\ MediumRuggedSoilTemp$

The coordinator (drone) knows the influential environmental factors that directly interacting in the onset and evolution of vite disease states, hence it performs a second-level matchmaking process to detect whether the grapes is likely attacked by the pathogen. According to this detection, the smart farmer acts on the surrounding monitored environment spraying fungicides (if necessary).

V. Experiments

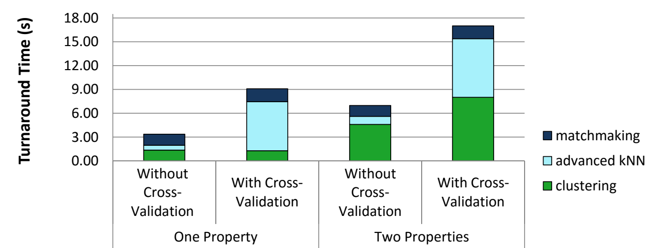
The proposed framework was implemented in a Java-based software prototype to early evaluate its feasibility. The semantic layer defined in this paper for the knowledge discovery phase was implemented by extending BEE DDS middleware [7]. The resulting architecture for each smart object consists of three basic modules:

– *Clustering*: performed with the *k-Means* algorithm provided by *Weka 3.7* library [22].

– *Advanced k-NN*: inference services for semantic-enhanced classification are provided by the embedded *Mini-ME 2.0.0* matchmaking and reasoning engine [23].

– *Semantic-based matchmaking*: also this module exploits Mini-ME to infer the environmental state from the semantic-based context description.

Object network performance tests were performed on 50 nodes that provide services and one requester node. They were connected through BEE DDS middleware enriched with the proposed semantic layer. The tests were conducted considering different system configuration variables: (i)annotation compression type: COX [24] or EXI [25]; (ii) upper ontology nesting level: 2, 3 or 4. Compressed size of messages exchanged between nodes, turnaround time and RAM usage were considered as performance metrics for both the ontology distribiution phase and the resource allotment step. Time was measured through timestamping instructions embedded in the source code. The system took less than 3 seconds for the first phase and less than 2 seconds for the second ones with EXI compression and considering the maximum nesting level of the upper ontology. With COX compression, the system performs both phases in a little more than 2.5 seconds. For memory usage analysis, an embedded thread was used to profile memory usage at runtime for both phases. RAM occupancy is always under 90 MB. It is important to note that the system appears to be stable and predictable. Intra-node performance evaluation was carried out on a Raspberry Pi [26] mobile host to simulate a real smart object with limited resources. Tests were conducted on a dataset of 400 real



	Without Cross-Validation (One Property)	With Cross-Validation (One Property)	Without Cross-Validation (Two Properties)	With Cross-Validation (Two Properties)
matchmaking	1.388	1.618	1.377	1.617
advanced kNN	0.613	6.173	1.010	7.388
clustering	1.366	1.288	4.594	8.001

Figure 4. Turnaround time results

instances of weather sensor data (temperature and humidity, collected from Weather Underground Web Service [27]) to simulate sensor data gathering by a smart object. The tests were performed in two different conditions: with static value of *k* for the *advanced k-NN* phase and with cross validation (useful to set *k* dynamically). Turnaround time of data point processing and RAM usage were considered for each module of the mining proposed framework. Fig. 4 reports turnaround time results for the analysis of only one and both properties, with and without cross validation. As expected, turnaround time increased significantly when the system performed cross validation to set the best *k* value for k-NN. The most significant differences between results for one and two properties are in the clustering and matchmaking phases, but the time increase is less than linear. For memory usage analysis, RAM occupancy is always below 17 MB. Memory peaks correspond to the most data intensive tasks, i.e., cross validation and matchmaking. These preliminary results evidence the feasibility of the proposed framework, even though optimizations will be required.

VI. Conclusion and Future Work

The paper proposed a novel knowledge-based framework enabling a smart object to collect and annotate sensor data in a fully automated fashion. k-NN machine learning algorithm was modified including non-standard semantic-based reasoning services in order to achieve this goal. The proposal also allows the knowledge sharing in distributed systems, particularly targeted toward scenarios including large numbers of resource-constrained nodes. The framework was devised as a semantic-enhancement layer to be added on top of an off-the-shelf publish/subscribe middleware. The approach was implemented in a working prototype, embedding a mobile semantic matchmaker. Correctness and feasibility of the proposal were evaluated in a reference case study. Future work concerns further performance evaluation comparison with state-of-the-art approaches and improvement, as well as enrichment of semantic-based capabilities for the data mining approach.

References

[1] L. Da Xu, W. He, and S. Li, "Internet of things in industries: a survey," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 4, pp. 2233–2243, 2014.

[2] L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": The next evolutionary step of the internet of things," *Communications Magazine, IEEE*, vol. 52, no. 1, pp. 97–105, 2014.

[3] M. Ruta, E. Di Sciascio, and F. Scioscia, "Concept abduction and contraction in semantic-based P2P environments," *Web Intelligence and Agent Systems*, vol. 9, no. 3, pp. 179–207, 2011.

[4] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation 11 December 2012. [Online]. Available: http://www.w3.org/TR/owl2-overview/ 2016.09.27

[5] F. Scioscia *et al.*, "A mobile matchmaker for the ubiquitous semantic web," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 10, no. 4, pp. 77–100, 2014.

[6] M. Ruta, F. Scioscia, and E. Di Sciascio, "Enabling the semantic web of things: framework and architecture," pp. 345–347, 2012, doi: 10.1109/ICSC.2012.42.

[7] BEE Data Distribuition System. [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/it/nid/211025 2016.09.22

[8] M. Martin *et al.*, "Learning to detect user activity and availability from a variety of sensor data," *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 13–13, 2004.

[9] K. Kapitanova, S. H. Son, and K.-D. Kang, "Using fuzzy logic for robust event detection in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 709–722, 2012.

[10] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 6, pp. 961–974, 2012.

[11] T. A. Nguyen, A. Raspitzu, and M. Aiello, "Ontology-based office activity recognition with applications for energy savings," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 5, pp. 667–681, 2014.

[12] S. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[13] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, "Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models," pp. 637–643, 2005.

[14] M. Compton *et al.*, "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, 2012.

[15] T. Rybicki, "Ontology recomposition," *Knowledge Engineering, Machine Learning and Lattice Computing with Applications*, pp. 119–132, 2012.

[16] S. B. Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers, "EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support," *Journal of Systems and Software*, vol. 81, no. 5, pp. 785–808, 2008.

[17] H. Li and G. Jiang, "Semantic message oriented middleware for publish/subscribe networks," *Defense and Security*, pp. 124–133, 2004.

[18] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook*. Cambridge University Press, 2002.

[19] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[20] K. Rasch, F. Li, S. Sehic, R. Ayani, and S. Dustdar, "Context-driven personalized service discovery in pervasive environments," *World Wide Web*, vol. 14, no. 4, pp. 295–319, 2011.

[21] 3D Robotics Iris. [Online]. Available: https://store.3dr.com/products/iris 2016.02.16

[22] Weka 3: Data Mining Software in Java. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/ 2016.05.27

[23] Mini-ME 2.0.0. [Online]. Available: http://sisinflab.poliba.it/swottools/minime/ 2016.07.14

[24] F. Scioscia and M. Ruta, "Building a semantic web of things: issues and perspectives in information compression," pp. 589–594, 2009.

[25] Efficient XML Interchange-EXI. [Online]. Available: https://www.w3.org/TR/exi/ 2016.06.30

[26] Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/about/ 2016.02.16

[27] Weather Underground Web Service. [Online]. Available: http://www.wunderground.com/ 2015.09.14