Object (b)logging: semantically rich context mining and annotation in pervasive environments

Eliana Bove
DEI - Politecnico di Bari
Via E. Orabona 4
Email: eliana.bove@poliba.it

***Abstract*—*Object (b)logging* is proposed as a novel general framework for the Semantic Web of Things, based on an evolution of conventional Web of Things paradigms. The advanced performance and the miniaturization of sensors allow to acquire several environmental parameters for event and phenomenon detection in many operational contexts. 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, continuously enriching their descriptive core based on events they detect. Finally they expose them to the outside world as in a blog. The feasibility of the proposed framework is supported by a case study and an early experimental campaign.**

I. Introduction

The *Web of Things* (WoT) vision [1] raises the pervasive computing paradigm to a world-wide scale. Information is retrieved and/or carried by micro-devices equipping everyday items or deployed in given environments and interconnected wirelessly. Nowadays, a wide variety of existing sensor devices are available for environmental features monitoring (*e.g.* humidity, temperature, pressure sensors) and people/object identification (*e.g.* RFID readers and tags) in order to detect events of interest in observed areas. Because of power and size constraints, such *things* have little processing capabilities, very small storage and low-throughput, short-range wireless links. Their activity is slow but continuous, so collectively producing large amounts of raw data to be processed by agents on mobile computing devices, through ad-hoc networks.

Nowadays, the relevance of the Web of Things 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. Semantic-based automatic inferences could be exploited to derive implicit information starting from the explicit event and context detection. The *Semantic Web of Things* (SWoT) improves intelligence of embedded objects and autonomic information management in pervasive contexts, in order to better support user activities. A *smart object* [2] is a software agent acting on behalf of an intelligent device, equipped with embedded sensors, actuators, communication ports as well as computation and storage facilities. Each smart object processes locally the retrieved information in order to describe itself and the context where it operates toward a variety of external devices and applications. The existing approaches defined for ”smart objects”, however, adopt architectures designed for single applications and use low-level data mining methods. In order to improve flexibility and interoperability, Semantic Web standard technologies can be adopted for rich and unambiguous semantic-based information exchange. Furthermore, frameworks should be extensible depending on the availability of different sensor and data source types.

This paper introduces a novel general framework for smart objects in the Semantic Web of Things. By leveraging the integration of standard supervised Machine Learning (ML) techniques with non-standard semantic-based inference services in [3] on annotations in Semantic Web languages, smart objects become able to annotate in a fully automatic way the environment they are in, continuously enriching their basic descriptive core according to events and phenomena they detect, in order to model context in an increasingly accurate way. Identification and sensing information are expressed in OWL-DL (Web Ontology Language based on Description Logics) annotations [4] via a semantic-based evolution of standard *k Nearest Neighbors* (k-NN) ML algorithm. The proposed approach relies on both ideas and technologies of distributed knowledge-based systems [5], whose individuals (assertional knowledge) are physically tied to objects disseminated in a given environment, without centralized coordination. Each annotation refers to an ontology providing the conceptualization for the particular domain. This model is supported by an advanced matchmaking carried out using metadata stored in the micro-devices, lacking fixed knowledge bases, and inference tasks distributed among mobile computing devices which provide minimal computational capabilities. This enables objects to annotate the context as in a log file and to describe themselves toward the rest of the world in a self-contained fashion like in a blog. In order to evaluate the usefulness of the proposed theoretical approach in a real scenario, the framework is being implemented in a prototypical smart robot butler, able to summarize the information gathered via its sensing interfaces in a semantically annotated description of the environment and relevant entities in it. Furthermore, it can communicate with a semantic-enhanced home/building automation infrastructure [6] to schedule other appliances, create greater comfort and well-being for users, detect possible issues and interact with people via ”blog” entries. The proposed framework has been tested in an early simulation campaign, basically devoted to assess its feasibility and sustainability.

The remainder of the paper is as follows. Section II provides a survey of related work. Section III discusses the proposed framework in detail. Section IV reports the early experimental evaluation of the proposal, while the next section describes a real case study. Conclusion closes the paper.

II. Related Work

Smart objects are able to retrieve a set of external data and to adapt themselves accordingly and/or act on the surrounding environment in order to modify it. Basically three kinds of information are managed:
– **Environmental data**: a description of the context where the object operates, according to data collected through a sensor set;
– **Capabilities**: smart object sensing and actuation features;
– **Constraints**: limits and features imposed by other entities and influencing the smart object's behavior.

Data sources can be either on-board sensors or queried through short-range wireless communication protocols. In the latter case, some reference technologies include sensor-enabled RFID (Radio Frequency IDentification) tags [7] and Semantic Sensor Networks (SSNs) [8]. Anyway, environmental information include the most important facts a smart object can learn about the context where it is located. Different works exist in literature about interpretation of raw, low-level gathered data and behavior adaptation. Current event recognition approaches are mainly based on threshold detectors or standard ML techniques. In [9] the authors proposed a Bayesian approach using a networked infrastructure for combining information from different sources and deriving an estimation of the user's situation in the workplace. The higher-level information is automatically derived from labeled sensor data. Raw sensor data –not accompanied by descriptive metadata– have a difficult exploitation, as they are hard to be interpreted or integrated in case of resource dearth. The low-level data analysis is integrated with the high-level context interpretation to derive implications and adapt the system behavior accordingly. Although probabilistic learning models are capable of handling noisy, uncertain and incomplete sensor data, they are characterized by several limitations such as scalability, ad-hoc static models and data scarcity. To overcome the low level of model accuracy using crisp threshold values, [10] presented a fuzzy logic approach for event detection based on a rule-base with reduced size. However, smart objects not only require high accuracy, but also computational efficiency for working on pervasive computing platforms. In [11] and [12], ontology based approaches are proposed for office and home activity recognition, respectively. They enable intelligent processing at a high level of automation and exploit reasoning, yet both support only full matches, which seldom occur in pervasive scenarios characterized by many heterogeneous information sources.

The proposed approach has been conceived and designed to describe and detect more complex context state, merging the strengths of Machine Learning and high-level semantic interpretation. Various ML techniques have different model accuracy, storage and computational requirements, hence it is quite impractical to identify the one universally suitable for smart objects purposes. Anyway, useful classification surveys can be found in literature [13]. Particularly, *incremental learning* algorithms, *e.g.* k-Nearest Neighbors (k-NN), appear to be very useful for intelligent entities which should evolve their understanding about the surrounding environment as new data are collected and processed. k-NN is endowed with good accuracy, is insensitive to outliers, works well with both nominal and numerical features and can be used as incremental learner, a crucial requirement for smart objects.

The task of describing sensor features and retrieved data through ontology-based formalisms has been addressed by the Semantic Web research community in recent years. *OntoSensor* [14] and *SSN-XG* [15] are among the most relevant and widely used ontologies. They are general enough to adapt to different applications and they are compatible with the OGC SWE (Open Geospatial Consortium Sensor Web Enablement) standards at the sensor and observation levels [16]. Such ontologies have been used in many projects, *e.g.* SPITFIRE [17], which combine semantic and networking technologies to build full frameworks. [18] aimed to improve the integration and of observation data for the Digital Earth effort, through the exploitation of a transparent RESTful proxy for OGCs Sensor Observation Service (SOS) to serve Linked Sensor Data on-the-fly. OGC SWE standards and other communication technologies are also integrated in Cloud4Sens [19], a cloud architecture for risk management which provides services based on both data-centric and device-centric models. To overcome QoS-aware smart object orchestration issues in such environments, [20] proposed two heuristic methods (top-down and bottom-up) based on the direction of the information flow during a service composition process. The problem related to the semantic data flow compression in limited resource spaces was faced in [21] by developing a scalable middleware platform to publish semantically-annotated data streams on the Web through HTTP.

Unfortunately, the above solutions only allow elementary queries in SPARQL fragments on RDF annotations. More effective techniques such as ontology-based *complex event processing* [22] exploit a shared domain conceptualization to define and specify complex events and actions that run on an event processing engine. This is similar to the approach proposed here, which is characterized by a rigorous integration of logic-based reasoning into a classic ML algorithm. A semantic-based framework for resource annotation and discovery in WoT contexts was proposed in [23]: the devised evolution of the Constrained Application Protocol (CoAP) resulted in a more effective information discovery and exchange, but data gathering and annotation were performed via simplistic threshold-based classification. The current proposal aims to a more principled and general solution.

III. Object (B)logging: Framework and Approach

In what follows, information about the proposed framework is provided. A general overview of the approach precedes the most relevant algorithmic details.

A. Framework overview

The envisioned framework aims to characterize the descriptive core of a smart object in a fully automatic way starting from sensed data. Throughout the object's lifetime, this semantic endowment is progressively enriched and completed so that it could be exposed to the outside world as in a blog. Basically, a smart object classifies sensory data and further combines classifications to identify patterns, situations and events. This high-level knowledge is useful to trigger actions, assume decisions or make interventions on the environment. Raw data are annotated in a semantically rich formalism grounded on the Attributive Language with unqualified Number restrictions ($\mathcal{ALN}$) Description Logics [24]. A properly devised ontology provides the needed conceptualization for the particular domain and a logic-based matchmaking algorithm allows to derive implicit knowledge starting from the one explicitly retrieved from sensor set and further annotated. This enables a fully automatic decisional capability about the actions to undertake by the object considering the context where it is. To achieve this purpose, each smart object continuously performs three steps, shown in Figure 1 and described hereafter:

1. Clustering: unsupervised clustering is adopted to pre-process input data: an unknown input instance is associated to the nearest cluster description [25]. By applying *k-Means* algorithm [26], the system cleans data from noise and outliers, replaces missing values, identifies possible issues in clustering and makes a preliminary coarse data classification. Each cluster description includes two components: *geometry* and *context*. Geometry describes data through statistical parameters. The context component annotates data w.r.t. the adopted reference ontology.

2. Advanced k Nearest Neighbors: an enhanced version of the *k-NN* algorithm is exploited to provide high-level data representation. It is based on a utility function combining semantic-based similarity measures (context distance) with partial scores deriving from geometry, *i.e.* quantitative statistical attributes.

3. Semantic-based matchmaking: the matchmaking task is performed on the data produced after the above steps in a given time span (observation window). The semantic annotation of the context is compared with descriptions of instances in the object Knowledge Base (KB) via inferences in [27] in order to find possible novel environmental characterizations so giving a semantic-based interpretation to the raw data collected by sensing devices. Each new data point or batch acquired in the 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 t_p are removed.

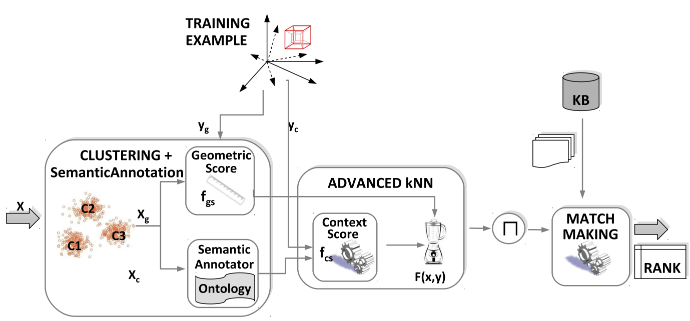

Figure 1. Sketch of the (b)log information flow

B. Mining Approach

The integration of a classic k-NN supervised machine learning technique with semantic-based matchmaking is one of the notable aspects of the proposed approach. In standard k-NN [28], a data point is classified by assigning the most frequent label among the k training samples nearest to it, and a common metric is used for measuring the distance between the data point itself and each training instance. Here a more complex distance metric is adopted, which integrates a geometric measure f_{gs} with a contextual semantic-based one f_{cs}, merged through a score combination function F. Here in what follows, input arguments of F are purposely named x and y, *i.e.* the instance to be analyzed and each element of the training set, respectively. In turn, both are described by geometric (x_g, y_g) and contextual (x_c, y_c) components.

1) Geometric Score: As inspired by [29], the geometric score $f_{gs}(x_g, y_g)$ is a numerical assessment of how much y_g is similar to x_g. Similarity is referred to the statistical distribution parameters that describe the data point and the training examples. Since x_g is the value to be matched, only the k dimensions featuring x must be considered. 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:

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

The above value $dmatch(x_{g_i}, y_{g_{ji}})$ is determined by calculating the overlap between the two dimension values x_{g_i} and $y_{g_{ji}}$, *i.e.* the i-th dimension of the j-th training example, divided by the size of x_{g_i}. This formula summarizes the notion that if $y_{g_{ji}}$ is fully contained in x_{g_i}, then $dmatch(x_{g_i}, y_{g_{ji}}) = 1$. If $B(x_{g_i}) = 0$ or $B(y_{g_{ji}}) = 0$, of course $dmatch(x_{g_i}, y_{g_{ji}}) = 0$. The overall geometric score is computed as:

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

Note that dividing by k normalizes w.r.t. the maximum cardinality of x_g dimensions.

2) Contextual Score: The contextual metric $f_{cs}(x_c, y_c)$ is computed on features annotated in *OWL 2* language [4] according to the reference terminology and leverages $\mathcal{ALN}$-based non-standard inference services in [3]. Description Logics (DLs) allow to represent knowledge by means of: *concepts a.k.a. classes*, representing sets of objects; *properties a.k.a. roles*, representing relationships between concepts; *individuals*, *i.e.* named instances of classes. These elements can be combined in expressions via *constructors*, which distinguish each language of the DL family and also determine its computational complexity characteristics. An ontology is composed by two types of assertions: *inclusion*, which allows to define *is-a* relationships between classes; *equivalence*, which allows to give a name to a particular concept expression. Table I summarizes syntax and semantics of constructors and assertions in $\mathcal{ALN}$. DL reasoners usually provide two standard inferences on concept expressions, *i.e. Satisfiability* and *Subsumption*. In particular, in a resource discovery setting, Subsumption returns `true` iff all requested features are provided. As evident, the output is boolean and a positive outcome is quite rare in realistic scenarios. In order to take into account non-full matches, Concept Abduction non-standard inference should be used alternatively. It yields a graded similarity between request and provided resources by calculating a semantic distance which enables in turn the resource ranking. In more detail, given an ontology $\mathcal{T}$ and two concept expressions A and B, if $\mathcal{T} \models A \sqsubseteq B$ is false then Concept Abduction computes a concept H (for *Hypothesis*) such that $\mathcal{T} \models A \sqcap H \sqsubseteq B$ is true. H represents missing features in the resource A, able to completely satisfy B w.r.t. the information modeled in $\mathcal{T}$.

TABLE I. SYNTAX AND SEMANTICS OF $\mathcal{ALN}$ CONSTRUCTORS

Name	Syntax	Semantics
Top	$\top$	$\Delta^{\mathcal{I}}$
Bottom	$\bot$	$\emptyset$
Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Atomic negation	$\neg A$	$\Delta^{\mathcal{I}} \backslash A^{\mathcal{I}}$
Universal quantification	$\forall R.C$	$\{d_1 \mid \forall d_2 : (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$
Number restriction	$\geq nR$	$\{d_1 \mid \sharp\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\} \geq n\}$
	$\leq nR$	$\{d_1 \mid \sharp\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\} \leq n\}$
Inclusion	$A \sqsubseteq D$	$A^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Equivalence	$A \equiv D$	$A^{\mathcal{I}} = D^{\mathcal{I}}$

On the contrary, 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 Abduction cannot be used and the Concept Contraction non-standard inference should be adopted. It is able to determine 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 $\mathcal{T}$. If nothing can be kept in B during the contraction process, one gets the worst solution $\langle G, K\rangle = \langle B, \top\rangle$, that is give up everything of B. On the opposite, if $A \sqcap B$ is satisfiable in $\mathcal{T}$ nothing has to be given up and the solution is $\langle \top, B\rangle$, *i.e.* give up nothing. Hence, a Concept Contraction problem amounts to an extension of Satisfiability. For both Concept Contraction and Concept Abduction minimality criteria for solutions are needed, which induce a distance metric (penalty) useful to measure the grade of match of a concept w.r.t. another one. The reader is referred to [3] for algorithms and wider argumentation.

The contextual score is defined as:

$$f_{cs}(x_c, y_c) = \frac{\omega \cdot penalty^{(c)} + (1-\omega) \cdot penalty^{(a)}}{penalty^{(a)}_{max}} \quad (3)$$

where $penalty^{(c)}$ is the penalty induced by Concept Contraction between the semantic descriptions of data point and training example, x_c and y_c respectively. On the contrary, $penalty^{(a)}$ is the penalty value of Concept Abduction between (the consistent part K of) the data point x_c and the training example y_c. This value is normalized w.r.t. $penalty^{(a)}_{max}$ *i.e.* the maximum semantic distance computed between a data point and the most generic $\top$ concept, which depends only on axioms in the reference domain ontology. The scoring mechanism is regulated by ω, which determines the relative weight of explicitly conflicting elements in y_c w.r.t. x_c. It depends on the calculated geometric score and is computed as:

$$\omega = \delta \cdot f_{gs}(x_g, y_g) \quad (4)$$

where $\delta \in [0.8, 1]$ is a proportional factor. ω assigns greater weight to Contraction result when the geometric distance is larger.

3) Overall Score: The overall distance F is defined as:

$$F(x, y) = (f_{gs}(x_g, y_g) + \epsilon)^{\alpha} \cdot (f_{cs}(x_c, y_c) + \gamma)^{1-\alpha} \quad (5)$$

It is a monotonic function ranging between 0 and 1 providing a consistent ranking of input training examples. Lower outcomes represent better results. A tuning phase can be performed to determine the values of parameters α, γ, ϵ following requirements of a specific discovery application. In detail, $\alpha \in [0, 1]$ weighs the relevance of contextual and geometric factors, respectively. Parameters $\epsilon \in [0, 1]$ and $\gamma \in [0, 1]$ control the outcome in case of either context or geometric full matches. Geometric full matches occur when the statistical data input distribution is equal to the one of considered training example, while contextual full matches arise when all features of the data point measurement x_c are satisfied by the training example y_c.

IV. EXPERIMENTS AND EVALUATION

The proposed framework was implemented in a Java-based system prototype to early evaluate its feasibility. The resulting architecture consists of three basic modules:
– *Clustering*: input data are clustered invoking the *k-Means* algorithm provided by the *Weka 3.7* library[1].
– *Advanced k-NN*: this step performs a semantic-enhanced classification of the contextual property observed by the smart object. Inference services are provided by the embedded *Mini-ME 2.0.0* matchmaking and reasoning engine[2].
– *Semantic-based matchmaking*: also this module exploits Mini-ME to infer the environmental state from the semantic-based context description.

Performance evaluation was carried out on a mobile host to simulate a real smart object with limited resources. The proposed framework was executed on a Raspberry Pi[3] single-board computer equipped with Broadcom BCM2835 system on a chip, with an ARM1176JZF-S CPU at 700 MHz, 512 MB RAM and an SD card for booting and long-term storage. It was configured with Wheezy Debian GNU/Linux 7.8 operating system, Wi-Pi WLAN USB Module[4] and 32-bit Java 7 SE Runtime Environment (build 1.8.0-b132). Tests were conducted on a dataset of 400 real instances of weather sensor data (temperature and humidity, collected from Weather Underground[5] Web service) 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 as performance metrics for each module of the proposed framework. Time was measured through timestamping instructions embedded in the source code. Each test was repeated 10 times and the average values were taken. Figure 2 reports turnaround time results for the analysis of only one and both properties, with and without cross validation. As expected, turnaround time increases significantly when the system performs 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, *jvmtop* tool[6] was used to profile memory usage at runtime. In this case, reported results are the peaks of 10 runs. They are shown in Figure 3. RAM occupancy is always below 17 MB. Memory peaks correspond to the most data intensive tasks, *i.e.* cross validation and matchmaking. The latter works with an ontology whose size is 221 KB. These preliminary tests evidence the feasibility of the proposed

[1]http://www.cs.waikato.ac.nz/ml/weka/
[2]http://sisinflab.poliba.it/swottools/minime/
[3]https://www.raspberrypi.org/about/
[4]http://it.farnell.com/element14/wipi/dongle-wifi-usb-for-raspberry-pi/dp/2133900
[5]http://www.wunderground.com/
[6]https://code.google.com/p/jvmtop/

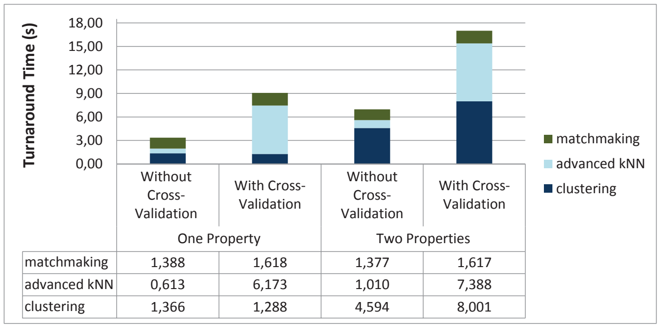

Figure 2. Turnaround time results

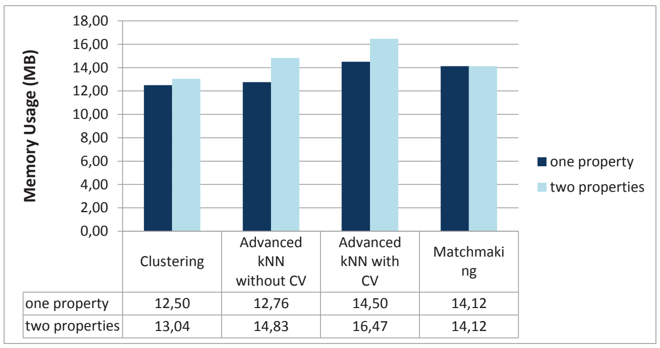

Figure 3. Memory usage

framework, even though optimizations will be required to run in realistic scenarios.

V. Case study: Smart Butler

The *Object (b)logging* paradigm has a strong potential impact in supporting humans in disaster scenarios as well as in everyday life. Intelligent autonomous robots are increasingly adopted in the *Urban Search And Rescue* (USAR) field, facing operations which would be too hazardous for human agents (*e.g.* search and rescue survivors trapped under collapsed buildings after earthquakes). Auto-coordination of different smart entities in a team allows to detect the context state, formulate plans to reach the mission goals and act accordingly. Furthermore, smart objects can be very useful to simplify the current frantic and busy everyday life. The case study proposed here is related to the latter field, *i.e.* a *home/office smart butler*. It is a robotic agent which exploits embedded as well as external sensors to achieve situation awareness while moving along rooms and corridors. For example, it can interact wirelessly with RFID readers placed at the entryways to detect tagged people who enter/exit. It is then capable of summarizing the gathered information in a semantically annotated description of the environment and its occupants, and publish it to a semantic-enhanced home/building automation infrastructure [6] to schedule other appliances and detect possible issues. In order to evaluate the usefulness of the framework in a real smart butler scenario, a prototypical testbed is under development, exploiting an *iRobot Create 2* programmable robot[7]. The device is being enriched with additional sensors and peripherals so that it can take on the duties of the home/office butler shortly described above. In what follows, an

[7]http://www.irobot.com/About-iRobot/STEM/Create-2.aspx/

TABLE II. Geometric and Contextual Features for Temperature property

Geometric features	Contextual features
– Mean – Variance – Kurtosis – Skewness	– FoodLocation: (*Fridge, KitchenCabinetNearOven, KitchenCabinetUnderSink, etc*) – FoodContainer: (*HermeticContainer, AluminumBox, GlassJar, etc*) – StorageLocationUsage: (*WidelyUsed, RarelyUsed*)

TABLE III. Temperature property classification with advanced k-NN

Property	TE	f_{gs}	f_{cs}	F	Result
	TE_1	0.2500	0.2583	62.82	
	TE_2	0.7084	0.1208	26.58	VeryLowTemperature
	TE_3	0.6667	0.3333	35.33	
	TE_4	0.5417	0.1708	39.80	
	TE_5	0.7084	0.0000	17.92	**LowTemperature**
	TE_6	0.8715	0.0719	13.49	
	TE_7	0.500	0.4833	51.66	
Temperature	TE_8	0.2917	0.3333	63.02	MediumTemperature
	TE_9	0.4271	0.1281	44.92	
	TE_{10}	0.3333	0.4332	63.19	
	TE_{11}	0.2917	0.5791	70.04	HighTemperature
	TE_{12}	0.5521	0.5010	47.82	
	TE_{13}	0.2917	0.4208	65.87	
	TE_{14}	0.5000	0.5166	52.33	VeryHighTemperature
	TE_{15}	0.2084	0.3958	71,01	

illustrative example is presented to better explain the flexibility of the approach.

Mr Brown's house is equipped with a smart refrigerator and a smart pantry, orchestrated by a smart butler which plays the role of coordinator. The information exchange about the environment state detected by these objects occurs through wireless communication protocols such as IEEE 802.11, Bluetooth or ZigBee. Both the refrigerator and the pantry are able to monitor food availability/consumption and control organoleptic qualities of products to detect expired food items. The sensor infrastructure used to fulfill these tasks consists of temperature sensor, ambient light sensor and relative humidity sensor [30]. Such sensor nodes are situated in kitchen cabinets and in the fridge. The smart butler exploits the proposed framework to analyze data collected by each sensor and to determine organoleptic property values of food items for the whole observation window. For example, geometric and contextual features that describe the temperature property are reported in Table II. Temperature semantic-based classification value is calculated considering not only quantitative statistical parameters, but also the context components which influence the variation of organoleptic quality. Possible class values are as *VeryLowTemperature*, *LowTemperature*, *MediumTemperature*, *HighTemperature* and *VeryHighTemperature*. For example, the temperature value in a kitchen cabinet is affected by its position: if it is near a switched-on oven, the temperature value within the cabinet may changes. Also, a low temperature value inferred for the fridge has not the same meaning of a low temperature value inferred to the refrigerator. Furthermore, if a food item is stored in a sealed container, the temperature value of the cabinet is less significant.

Table III shows the results for temperature property of *rice* stored in the pantry. In detail, advanced k-NN runs with k=7 and assigns the most frequent label among the 7 training samples (TE_i) nearest to the input instance. In particular, with a distance threshold of 50 only the items underlined in the table are considered near, therefore the selected class is

LowTemperature (3/7 *LowTemperature*, 2/7 *VeryLowTemperature*, 1/7 *MediumTemperature* and 1/7 *HighTemperature* nearby points). By replicating this process for each sensed parameter, the smart objects (*e.g.* pantry or refrigerator) create a high-level representation of the considered product conservation status. A food item semantic description detected by the system follows as an example.

$Rice \sqcap \forall has\_Storage\_Temperature.LowTemperature \sqcap$
$\exists\ has\_Storage\_Temperature\ \sqcap \forall has\_Storage\_Humidity.$
$(VeryLowHumidity\ \sqcap \neg HighHumidity) \sqcap$
$\exists\ has\_Storage\_Humidity\ \sqcap \forall has\_Storage\_Lighting.$
$(DimLight\ \sqcap\ \neg BrightLight)\ \sqcap\ \exists\ has\_Storage\_Lighting$
$\sqcap \forall has\_Storage\_Location.Pantry\ \sqcap\ \exists has\_Storage\_Location$

The coordinator butler knows recommended characteristics for the storage of a specific product, hence it performs a second-level matchmaking process to create a list of well-preserved food items. According to this list as well as to health concerns (*e.g.* food intolerances, retrieved from semantically annotated user profiles), the butler can suggest the most suitable recipes.

VI. Conclusion and Future Work

The paper proposed a novel WoT framework enabling a smart object to collect and annotate sensor data in a fully automated fashion. k-NN machine learning algorithm was modified including semantic-based matchmaking in order to achieve this goal. Early experiments suggested the framework computational feasibility in terms of response time and memory usage. A prototypical testbed is under development to allow experimental evaluation in a real smart butler scenario and to assess effectiveness w.r.t. state-of-the-art approaches for pervasive contexts.

References

[1] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, "From the internet of things to the web of things: Resource-oriented architecture and best practices," in *Architecting the Internet of Things*. Springer, 2011, pp. 97–129.

[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/

[5] 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.

[6] M. Ruta, F. Scioscia, G. Loseto, and E. Di Sciascio, "Semantic-based resource discovery and orchestration in Home and Building Automation: a multi-agent approach," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 730–741, 2014.

[7] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*. IEEE, 2012, pp. 1282–1285.

[8] L. M. Ni, Y. Zhu, J. Ma, M. Li, Q. Luo, Y. Liu, S. Cheung, and Q. Yang, "Semantic sensor net: An extensible framework," in *Networking and Mobile Computing*. Springer, 2005, pp. 1144–1153.

[9] M. Martin, O. Brdiczka, D. Snowdon, J.-L. Meunier *et al.*, "Learning to detect user activity and availability from a variety of sensor data," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 2004, pp. 13–13.

[10] 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.

[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] 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.

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

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

[15] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *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.

[16] M. Botts, G. Percivall, C. Reed, and J. Davidson, "OGC® sensor web enablement: Overview and high level architecture," in *GeoSensor networks*. Springer, 2008, pp. 175–190.

[17] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kroller, M. Pagel, M. Hauswirth *et al.*, "Spitfire: toward a semantic web of things," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 40–48, 2011.

[18] K. Janowicz, A. Bröring, C. Stasch, S. Schade, T. Everding, and A. Llaves, "A restful proxy and data model for linked sensor data," *International Journal of Digital Earth*, vol. 6, no. 3, pp. 233–254, 2013.

[19] M. Fazio and A. Puliafito, "Cloud4sens: a cloud-based architecture for sensor controlling and monitoring," *Communications Magazine, IEEE*, vol. 53, no. 3, pp. 41–47, 2015.

[20] S. C. Geyik, B. K. Szymanski, and P. Zerfos, "Robust dynamic service composition in sensor networks," *Services Computing, IEEE Transactions on*, vol. 6, no. 4, pp. 560–572, 2013.

[21] J. A. Fisteus, N. F. García, L. S. Fernández, and D. Fuentes-Lorenzo, "Ztreamy: A middleware for publishing semantic streams on the web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 25, pp. 16–23, 2014.

[22] K. Cao, Y. Wang, and F. Wang, "Context-aware distributed complex event processing method for event cloud in internet of things." *Advances in Information Sciences & Service Sciences*, vol. 5, no. 8, 2013.

[23] F. Gramegna, S. Ieva, G. Loseto, and A. Pinto, "Semantic-enhanced resource discovery for CoAP-based sensor networks," in *5th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI 2013)*. IEEE, jun 2013, pp. 228–233.

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

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

[26] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[27] F. Scioscia, M. Ruta, G. Loseto, F. Gramegna, S. Ieva, A. Pinto, and E. Di Sciascio, "A mobile matchmaker for the Ubiquitous Semantic Web," *International Journal on Semantic Web and Information Systems*, vol. 10, no. 4, pp. 77–99, 2014.

[28] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[29] 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.

[30] V. Annese, G. Biccario, S. Cipriani, and D. De Venuto, "Organoleptic properties remote sensing and life-time prediction along the perishables goods supply-chain." Proceedings of the 8th International Conference on Sensing Technology, 2014.