Object (B)logging: a Decentralized Cognitive Paradigm for the Industrial Internet of Things

Giovanna Capurso[1], Michele Ruta[1], Eugenio Di Sciascio[1]

***Abstract*— This paper proposes *object (b)logging*, a general framework for managing object networks in the Semantic Web of Things. By applying standard supervised machine learning algorithms on raw sensor data along with non-standard semantic-based reasoning services, this approach enables resource-constrained components to produce rich and meaningful representations of real-world environments and events as in a *log*. The acquired knowledge is progressively enriched during object's lifetime and exposed to the outside world as in a *blog*, in order to trigger complex interactions through an advanced resource discovery. A case study of predictive maintenance and early results support the feasibility of the proposal to Industrial Internet of Things settings.**

I. Introduction

The Industrial Internet of Things (IIoT) evolution is reshaping industry control toward smart industry featured with data-driven decision making [1]. Digital information and communication technologies are integrated with conventional control networks in production plants and supply chains. The increased information processing capabilities enable fine-grained real-time visibility and control of industrial processes, products and equipment. In IIoT, a network of intelligent devices is able to process fast, thick data streams and generate thin, high-level significant information to share in interoperable formats. This means evolving the IoT to a large-scale agent-based paradigm, which tends to produce novel interactions via several relatively simple local interchanges excluding centralized decision-making [2].

To accomplish such vision, this paper proposes *object (b)logging* as a novel general framework for smart objects following the *Semantic Web of Things* (SWoT) [3] paradigm. The effectiveness and relevance of IIoT can be further enhanced by associating semantically rich, compact descriptions to both devices and data to enable novel classes of smart solutions. Semantic-based software agents running on devices equipped with sensors, actuators, communication ports and (even limited) computation and storage facilities, define a new generation of *smart objects* [4].

Object (b)logging relies on both ideas and technologies of distributed knowledge-based systems [3], whose individuals (assertional knowledge) are physically tied to objects in a given environment, not requiring hotspot coordination. The proposed approach integrates standard supervised machine learning with non-standard inferences on Description Logics (DL) expressions. The aim is to enable smart objects producing and annotating high-level descriptions about themselves and the environment they are located in (as in a micro-*log*). Each annotation refers to an ontology providing the conceptualization for the particular domain. According to this vision, smart objects are able to continuously enrich an early logical descriptive core, which models ground knowledge about their own features and capabilities as well as about concepts and relationships with general validity in the domain. Annotations evolve during the object's lifetime and are exposed toward external devices and applications in a self-contained fashion like in a micro-*blog*.

In order to evaluate the usefulness of the proposed approach, it has been applied to an IIoT scenario concerning *predictive maintenance*. Furthermore, feasibility and sustainability have been assessed by means of an preliminary experimental evaluation of the most performance-critical elements in the framework and a comparison with the state of the art.

The remainder of the paper is as follows. Section II discusses related work. Section III describes the proposed framework in detail, while the predictive maintenance case study is in Section IV. Section V reports on evaluations, before conclusion.

II. Related work

In literature, several approaches exist for interpreting raw heterogeneous data to extract knowledge about relevant conditions, events and features of the current context and to support decision-making in dynamic IIoT situations. The context-aware adaptive data fusion system in [5] is able to reconfigure a sensor infrastructure dynamically: heterogeneous data collected by sensors and contextual information are combined by a Bayesian network in order to grant the best trade-off between energy consumption and sensing accuracy. The solution in [6] exploits neural networks in cloud-assisted processing of manufacturing equipment data for an active preventive maintenance. Real-time scheduling of maintenance resources is carried out according to fixed logic: this may be a sub-optimal choice in complex and unpredictable environments.

The above works rely on a centralized architecture, where collected raw data are sent by mobile agents to a single –often cloud-based– hotspot for further processing. In practical implementations, the communication overhead with remote hosts can be very resource-intensive for both the infrastructure and participant devices. Additionally, higher latency and jitter values due to cloud communication are ill-suited to real-time monitoring needed to detect events and trigger actions on-the-fly. To overcome these issues, *edge*

[1] Authors are with Polytechnic University of Bari, via E. Orabona 4, I-70125 Bari, Italy name.surname@poliba.it

computing [7] approaches require to process data locally on each smart object and reach conclusions/decisions by interacting autonomously with other entities through Mobile Ad-hoc NETworks (MANETs). This is adopted *e.g.*, in [8] and [2], which unfortunately either require direct user intervention or support only elementary agent behaviours and basic interactions. Automated real-time event detection as well as triggering complex objects interactions is prevented by the limited expressiveness and the lack of explicit semantics in device and service profiles.

In order to surmount these limits, the integration of wireless sensor networks (WSNs) and pervasive micro-devices with ontologies and Semantic Web technologies has given rise to the SWoT vision [3]. Starting from knowledge representation models grounded on formal logic-based semantics, reasoning engines can automatically infer knowledge entailed by a given *Knowledge Base* (KB) to provide adaptive applications which can take smart decisions according to data streams gathered from sensors [9]. Several proposals have injected semantic in communication technologies. For example, [10] exploited Web technologies in building automation to dynamically coordinate devices/services in accordance with the context. A multi-layer cross-domain reasoning framework has been proposed in [11], able to identify relevant events, suggest potential actions and/or enable dynamic service discovery and automatic composition.

A not negligible issue of current approaches, however, is that triggering rules requires the system state to fully match the rule pattern; this is quite unlikely in real-world scenarios. Semantic similarity measures, coping with approximate matches, can be more flexible. The sensing-as-a-service discovery platform in [12] exploits context information related to sensors in order to search, rank and select the most relevant ones w.r.t. a user's request. Also [13] presented an approach for a semantic-based discovery and combination of services, exploiting proper similarity metrics and a heuristic algorithm to search the best candidates for a given service request.

The logic-based framework proposed here enhance information gained through machine learning, annotating them for a further inferential stage [14]. This is devoted to gather implicit notions referred to context state and enabling cooperation among autonomous objects.

III. Proposed approach

Hereafter it will be described the proposed framework: a general overview of the approach precedes most relevant algorithmic details.

A. The object (b)logging vision

A *smart object* [4] is recognizable as an intelligent agent running on a device equipped with embedded sensors, actuators, communication ports as well as (usually constrained) computation, storage and energy resources. Smart objects are able to gather data streams for internal and external parameters, to adapt themselves to the environment and/or act in order to (possibly) modify it. The proposed approach envisions an object *log* as the collection of all the information a smart object has learned first-hand or received from peers. Exploiting the log, a smart object is able to adapt and coordinate in order to achieve a goal. More formally, a log is composed by the semantic descriptions referred to a domain ontology progressively enriched during the objects' lifetime. It is based on a logical descriptive core and is used for intelligent interpretation of retrieved information. Basically, each object dipped in a given environment collects data from sensors and processes them in order to produce a high-level annotation of detected events and conditions. Particularly, this information is expressed in the *Attributive Language with unqualified Number restrictions* ($\mathcal{ALN}$) DL, which is a subset of the *Web Ontology Language* (OWL) 2 [15] standard (adopted for modelling ontologies and annotating resources in the Semantic Web). Each object is also equipped with a micro-reasoning engine performing automated inferences to derive implicit knowledge out of semantic-based information gathered from the environment. In this way, an object becomes able to identify on-the-fly the task(s) needed to change its own configuration or to act on the environment also exposing information possibly useful to nearby objects.

The object (b)logging vision aims to enable activity monitoring and recognition without requiring large computational resources. For this reason, the paradigm focuses on mobile and pervasive scenarios, affected by severe resource limitations in processing, memory, storage and energy consumption. Machine learning combined with non-standard reasoning allow managing approximate matches in order to compensate for possible anomalies in data gathering and communication, so increasing robustness and flexibility of sensing, interpreting and interacting.

B. Semantic-enhanced machine learning for context annotation

In order to generate semantically rich context descriptions in a fully automatic way, each smart object continuously performs the tasks depicted in Figure 1 and described hereafter.

1. Clustering: unsupervised clustering is adopted to preprocess input data for each feature of interest. By applying the *k-Means* algorithm [16], data are cleaned from noise and outliers replacing missing values. Each cluster is characterized by *geometry* and *context*. Geometry describes data through statistical parameters, while context annotates them w.r.t. a reference domain ontology. An unknown input sample is associated with the description of the nearest cluster, realizing a preliminary coarse data classification.
2. Advanced k-Nearest Neighbors: an enhanced version of the k-NN algorithm [17] grounded on a composite distance metric is exploited to provide high-level feature representation. The adopted distance metric integrates a *geometric* measure f_{gs} and a *contextual* semantic similarity degree f_{cs}, merged through a *score combination function* F. The integration of classic k-NN supervised machine learning with semantic-based matchmaking is a peculiar aspect of the proposed approach. In what follows, $x = \langle x_g, x_c \rangle$ and $y_j = \langle y_{g_j}, y_{c_j} \rangle$ are input arguments of F: they represent

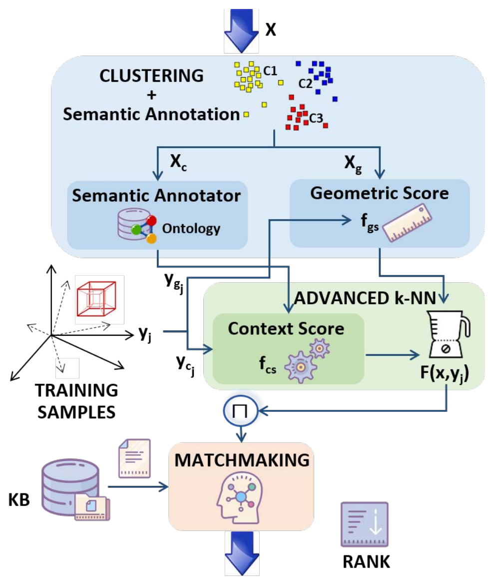


Fig. 1. Block diagram of semantic-enhanced machine learning

the sample to be examined and the j-th element of the training set, respectively. Both are described by geometric and contextual components.

– *Geometric score*: $f_{gs}(x_g, y_{g_j})$ expresses numerically the similarity between x_g and y_{g_j}, as proposed in [18]. Since x_g is the value to be matched, only the q dimensions describing x must be taken into account. Therefore, a ground vector $B(x_g) = \langle b_1, b_2, ..., b_q \rangle$ is defined, where $b_i \in (0, 1)$ and $b_i = 0 \iff x_{g_i} = \emptyset$. The matching value on a single dimension is computed as:

$$m(x_{g_i}, y_{g_{ji}}) = \begin{cases} \frac{|x_{g_i} \cap y_{g_{ji}}|}{|x_{g_i}|}, & B(x_{g_i}) = B(y_{g_{ji}}) = 1 \\ 0, & otherwise \end{cases} \quad (1)$$

The value of $m(x_{g_i}, y_{g_{ji}})$ is computed by determining the overlap between x_{g_i} and $y_{g_{ji}}$ (*i.e.*, the i-th dimension of the j-th training example) normalized by the length of x_{g_i}; if x_{g_i} is fully contained in $y_{g_{ji}}$, then $m(x_{g_i}, y_{g_{ji}}) = 1$. The overall geometric matching score is defined as:

$$f_{gs}(x_g, y_{g_j}) = 1 - \frac{\sum_{i=1}^{q} m(x_{g_i}, y_{g_{ji}})}{q} \quad (2)$$

Division by q produces normalization w.r.t. the highest cardinality of x_g.

– *Contextual score*: $f_{cs}(x_c, y_{c_j})$ is calculated on features annotated in OWL 2 language according to the data mining methodology proposed in [19] with the domain ontology as reference vocabulary. Moreover *Concept Contraction* and *Concept Abduction* non-standard inferences are exploited to provide fine-grained semantic distance metrics [14]. Given an ontology $\mathcal{T}$ and two concept expressions R and S (describing features related to the data point and the training sample, respectively), if the conjunction $R \sqcap S$ is unsatisfiable w.r.t. $\mathcal{T}$ (*i.e.*, R and S are in logical conflict with each other), *Concept Contraction* determines which part G (for *Give up*) should be retracted from R to obtain a contracted version K (for *Keep*) such that $K \sqcap S$ is satisfiable in $\mathcal{T}$. An associated $penalty_{(c)}$ value is also computed, representing the semantic distance of R from S. Conversely, if R and S are compatible, but S does not satisfy R completely, then *Concept Abduction* determines what should be hypothesized in S in order to obtain a full match, *i.e.*, to make the subsumption relation $S \sqsubseteq R$ true w.r.t. $\mathcal{T}$. The solution H (for *Hypothesis*) to Abduction can be interpreted as what is requested in R and not specified in S; also in this case a related distance metric $penalty_{(a)}$ is computed.

The overall contextual score is calculated $\forall j$ as:

$$f_{cs}(x_c, y_{c_j}) = \frac{\omega_j \cdot penalty_{(c)} + (1 - \omega_j) \cdot penalty_{(a)}}{penalty_{(a),max}} \quad (3)$$

The adopted normalizing factor $penalty_{(a),max}$ is the maximum semantic distance computed between x_c and the most generic concept $\top$, and depends only on axioms in the reference ontology. The scoring mechanism is tuned by the weight ω_j, defined as:

$$\omega_j = \delta \cdot \ f_{gs}(x_g, y_{g_j}) \quad (4)$$

The proportional factor δ ranges in $[0.8, 1]$; the rationale is to assign greater weight to Contraction penalty when the geometric distance is large.

– *Overall score*: F is defined as:

$$F(x, y_j) = (f_{gs}(x_g, y_{g_j}) + \epsilon)^{\alpha} \cdot (f_{cs}(x_c, y_{c_j}) + \gamma)^{1-\alpha} \quad (5)$$

It is a monotonic function $\in [0, 1]$ and ranks input training samples in a consistent way. It basically adopts a scale distance, where lower outcomes are better results. The $\alpha \in [0, 1]$ factor determines the relative weight of contextual and geometric scores. In case of contextual or geometric full match, the score is tuned by means of $\epsilon \in [0, 1]$ or $\gamma \in [0, 1]$, respectively. Values for α, γ and ϵ are determined heuristically according to the particular scenario.

3. Semantic-based matchmaking: the matchmaking task is performed on the semantic characterization of the heterogeneous sensory data collected in a given observation window. The annotation of the detected context is built as the logical conjunction of each preliminary classification of individual features. By comparing this description with instances in the KB via the above non-standard inferences, it is possible to (i) derive implicit knowledge about current events and conditions, giving a semantic-based interpretation to the raw data collected by sensing devices, and (ii) trigger actions, take decisions or make interventions on the environment at runtime.

Each new data point or batch acquired in the observation window undergoes this process: finally the latest data points are integrated in the training set –while data older than a purging threshold are removed– in order to keep it updated. The smart object is than ready for a new iteration.

IV. Case Study: Predictive Maintenance in Oil Plants

In order to motivate the proposed approach and show its possible benefits, a short case study about failure prediction on drill rigs in the petroleum industry is reported hereafter.

An oil extraction plant operates in a desertic area. The bearings conditions and the surrounding environment must be continuously monitored, carrying out autonomous predictive maintenance to maximize equipment life.

A monitoring solution, based on the object (b)logging paradigm, uses miniaturised sensors and intelligent wireless components embedded in the bearing. Inspection features include bearing volume and type, temperature and vibration revealing unusual behavior when bearings are closer to fail. The proposed approach exploits the high-level annotation framework described in Section III-B on raw data collected by equipped sensors, combined with context-awareness related to both equipment configuration and working conditions. Smart objects cooperate to autonomously take corrective actions and avoid damaging factors as much as possible, enabling a longer machine operating life. Each measure provided by sensors connected to a smart bearing is characterized by a semantic annotation defined w.r.t. the domain ontology[1].

Other objects dipped in the context provide actuators to perform maintenance intervention: as shown in Figure 2, the annotation describing each actuator consists of the set of early warning conditions when intervention is needed, including bearing behaviour, as well as environmental status. In the proposed scenario, lubricant injector, air cooling and compressed air cleaner systems are considered.

```
LI1 EquivalentTo: LubricantInjector and
(hasBrgStatus only ((hasBrgSoundLoud only HighLoud)
and (hasBrgSoundType only MetallicSound) and
(hasBrgTemp only HighTemp) and (hasBrgVibration
only LowVibration)))

AC1 EquivalentTo: AirCooling and (hasBrgStatus
only ((hasBrgSoundLoud only LowLoud) and
(hasBrgSoundType only RegularSound) and (hasBrgTemp
only HighTemp) and (hasBrgVibration only
LowVibration))) and (hasEnvStatus only (hasEnvTemp
only HighTemp))

CAC1 EquivalentTo: CompressedAirCleaner and
(hasBrgStatus only ((hasBrgSoundLoud only HighLoud)
and (hasBrgSoundType only IrregularSound) and
(hasBrgTemp only MediumTemp) and (hasBrgVibration
only HighVibration))) and (hasEnvStatus only
(hasEnvDust only HighDusty))
```

Fig. 2. Annotations of the available maintenance services

Following up with the proposed case study, *an instrumented bearing is fit for operating under well-known standard conditions defined by system designer. Predicted Remaining Useful Life (RUL) amounts to 40000 hours.*

Figure 3 shows the standard operating conditions (OC) in the form $\langle OC, RUL\rangle$ where OC is a semantic annotation w.r.t. the domain ontology and RUL is the estimation of the remaining useful life under the current OC.

```
<OC_std, RUL_std> = <BrgStd, 40000>

BrgStd EquivalentTo: (hasBrgDesign only
BrgStdDesign) and (hasBrgStatus only BrgStdStatus)
and (hasEnvStatus only StdEnv)

BrgStdDesign EquivalentTo: (hasBrgLoad only
MediumLoad) and (hasBrgRpm only MediumRpm) and
(hasBrgShaft only VerticalShaft) and (hasBrgType
only SphericalRollerBrg)

BrgStdStatus EquivalentTo: (hasBrgSoundVolume only
LowVolume) and (hasBrgSoundType only RegularSound)
and (hasBrgTemp only LowTemp) and (hasBrgVibration
only LowVibration)

StdEnv EquivalentTo: (hasEnvHum only LowHum) and
(hasEnvTemp only LowTemp) and (hasEnvDust only
LowDust)
```

Fig. 3. Annotations of the standard operating conditions

Operating conditions deviating from OC_{std} are associated with a RUL $< RUL_{std}$. The service life hours under unforeseen cases at time t_i is estimated by means of the formula:

$$RUL_{t_i} = [1-p] \cdot RUL_{std} \tag{6}$$

with the semantic penalty function p computed as:

$$p = \frac{w \cdot penalty_{(c)} + (1-w) \cdot penalty_{(a)}}{penalty_{(a),max}} \tag{7}$$

where $penalty_{(c)}$ is the penalty calculated by Concept Contraction between OC_{std} and OC_{t_i}, while $penalty_{(a)}$ is the penalty value of the Concept Abduction between the consistent part determined by Concept Contraction and OC_{t_i}. The value of p is normalized w.r.t. the maximum possible semantic distance $penalty_{(a),max}$ of OC_{std}, as explained in Section III-B. In the ongoing example, the parameter w is set to 0.7, to more penalize conflicting features.

```
<OC_t1, RUL_t1> = <BrgT1, 31320>

BrgT1 EquivalentTo: (hasBrgDesign only
BrgStdDesign) and (hasBrgStatus only BrgStatusT1)
and (hasEnvStatus only EnvStatusT1)

BrgStatusT1 EquivalentTo: (hasBrgSoundVolume
only HighVolume) and (hasBrgSoundType only
MetallicSound) and (hasBrgTemp only VeryHighTemp)
and (hasBrgVibration only LowVibration)

EnvStatusT1 EquivalentTo: (hasEnvHum only
MediumHum) and (hasEnvDust only HighDust) and
(hasEnvTemp only HighTemp)
```

Fig. 4. Annotations of the operating conditions at time t_1

At a given time t_1 a smart bearing will summarise the information gathered via its sensing interface in a semantic annotation of itself and the context it is in. The acquired knowledge leads to a RUL estimate according to equation 6, as depicted in Figure 4. OC_{t_1} has non-negligible differences w.r.t. OC_{std}, in both operating environment and bearing status as highlighted by Concept Contraction and Abduction results in Figure 5. The annotation is saved in the smart

[1]For the sake of compactness and readability, reported DL concept expressions are simplified w.r.t. the ones actually used for the case study. In particular, for each universal restriction of the form `role only concept`, the reader should assume a corresponding `role some owl:Thing` existential restriction is present in conjunction. Furthermore, due to space limitations, only a relevant subset of classes is illustrated within figures in this section, adopting OWL 2 *Manchester* syntax. Reported matchmaking results refer to the complete expressions.

```
<BrgStd_Keep> EquivalentTo: (hasBrgDesign
only BrgStdDesign) and (hasBrgStatus only
((hasBrgVibration only LowVibration))) and
(hasEnvStatus only ((hasEnvHum only MediumHum)))

<BrgStd_GiveUp> EquivalentTo: (hasBrgStatus
only ((hasBrgSoundVolume only LowVolume) and
(hasBrgSoundType only RegularSound) and (hasBrgTemp
only LowTemp))) and (hasEnvStatus only ((hasEnvTemp
only LowTemp) and (hasEnvDust only LowDust)))

<BrgT1_Hypothesis> EquivalentTo: (hasBrgStatus
only (hasBrgSoundVolume only HighVolume) and
(hasBrgSoundType only MetallicSound) and
(hasBrgTemp only VeryHighTemp)) and (hasEnvStatus
only EnvStatusT1)
```

Fig. 5. Concept Contraction and Concept Abduction between OC_{std}/$OC_{std\ keep}$ and OC_{t_1}

bearing's log and published on the mobile ad-hoc network of nearby devices in the oil plant. Each device starts a semantic-based matchmaking between OC_{t_i} and its own capabilities, inferring what useful services it can provide to nearby nodes. Re-lubrication, provided by *LI1*, is the most suitable maintenance service, but it is temporary unavailable. $AC1$ is therefore selected and activated, in order to prevent further overheating and mitigate the effects of the detected anomaly. Figure 6 shows the updated operating conditions shared by smart bearing at time t_2. Thanks to maintenance action, bearing temperature improves from $VeryHighTemp$ to $HighTemp$. RUL_{t_2} increases slightly, but it is not optimal yet. After a new data gathering and mining round, a new semantic-based service discovery task will be triggered to restore normal working conditions.

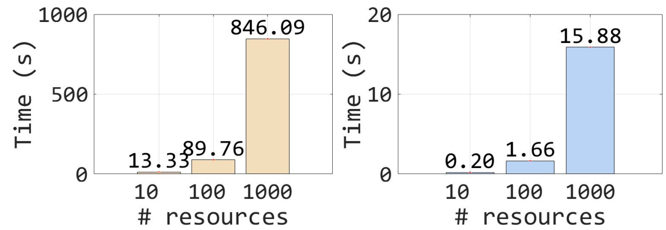
```
<OC_t2, RUL_t2> = <BrgT2, 31480>

BrgT2 EquivalentTo: and (hasBrgDesign only
BrgStdDesign) and (hasBrgStatus only BrgStatusT2)
and (hasEnvStatus only EnvStatusT1)

BrgStatusT2 EquivalentTo: (hasBrgSoundVolume
only HighVolume) and (hasBrgSoundType only
MetallicSound) and (hasBrgTemp only HighTemp) and
(hasBrgVibration only LowVibration)
```

Fig. 6. Annotations of the operating conditions at time t_2

The above toy example aims to highlight the flexibility of object (b)logging applied to challenging industrial IoT scenarios. Since the application-specific elements are encapsulated in the KB only, the semantic-based framework results as a general-purpose, cross-domain solution. Semantic descriptions in the examples were kept short for easier understanding, but the adopted inferences allow managing more detailed specification.

V. Evaluation

Capabilities of the proposed approach have been compared with the state-of-the-art works discussed in Section II. Table I highlights six characteristics, deemed as key dimensions of intelligent IIoT frameworks. Features focus on context awareness and decentralized decision, as well as on the adoption of semantic technologies to describe entities and phenomena with explicit formal descriptions, allowing high-level information fusion, advanced resource/service discovery and ranking.



(a) Init task (b) Matchmaking task

Fig. 7. Time results depending on the number of resources

To prove the feasibility of the proposed framework, an experimental evaluation has been carried out by implementing the case study in Section IV on a resource-constrained computing platform with simulated sensor data. Tests focused on the most performance-critical element in the framework, *i.e.*, reasoning and query answering. Tests have been executed on Raspberry Pi Model B[2]. The *Mini-ME* reasoning engine –designed for computationally constrained nodes– [14] executed semantic matchmaking as described in Section III on $\mathcal{ALN}$ DL annotations grounding knowledge extraction. Designed experiment aims to evaluate turnaround time and memory usage. Particularly, turnaround is composed by (i) *init time* to load and initialize the KB and (ii) *matchmaking time* to produce similarity measures between query and resources. A growing number of resources involved in semantic matchmaking has been considered. The examined orders of magnitude are 10, 100 and 1000. The referenced request is a simplified version of *BrgT1* (in Section IV).

Time. As expected, init time increased significantly with higher resources number, due to the growing amount of instances in the KB; Figure 7(a) reflects this behaviour. Growth exhibited a slightly lower than linear trend, suggesting good scalability. Absolute times are high but this is not worrisome, since loading occurs only once per session, when a smart object starts. Furthermore in real-world scenarios such a large number of resources is rarely added at the start; typically the KBs will be augmented with new instances progressively during devices' lifetime. A similar trend is in Figure 7(b) concerning matchmaking time. The inference engine provides really acceptable performance, with the worst stress test involving 1000 resources taking nearly 16 s, while the most realistic simulations (10 and 100 resources) required 0.2s and less than 2s, respectively.

Memory. Memory usage is shown in Figure 8. Also in this case, the number of involved resources influenced memory consumption. In experiments involving 10 and 100 resources memory peak was always below 17 MB. The scenario with 1000 resources had a slightly higher memory consumption, with a peak lower than 23 MB.

Globally, experimental outcomes show the approach is computationally sustainable on computing devices for IIoT. Time performance is well suited to real-time monitoring for detecting events and triggering actions on-the-fly. Memory load fits properly the strict constraints of pervasive objects.

[2]Equipped with a single-core ARM11 CPU at 700 MHz, 512 MB RAM (shared with GPU), 8 GB storage memory on SD card, Raspbian Wheezy OS and 32-bit Java 8 SE Runtime Environment (JRE, build 1.8.0-b132)

TABLE I

COMPARISON OF THE PROPOSED APPROACH WITH THE STATE OF THE ART

Approach	Year	Decentralized decision	Semantic-enhanced	Context-aware	Discovery match type	Service ranking
Han *et al.* [10]	2014	No	Yes (RDF)	Yes	Exact only	No
Perera *et al.* [12]	2014	No	Yes (RDF)	Yes	Exact and approximated	Yes (Top-K on weighted attributes)
Liu *et al.* [13]	2016	Yes (multi-agent)	Yes	Yes	Exact and approximated (keyword-based)	Yes (information retrieval formulas)
Santos *et al.* [2]	2016	Yes (consensus)	No	No	n.a. (classifier)	No
Ali *et al.* [11]	2017	Yes (multilayer)	Yes (RDF)	Yes	Exact only (rule-based)	No
De Paola *et al.* [5]	2017	No	No	Yes	n.a. (classifier)	Yes (energy-based)
O'Toole *et al.* [8]	2017	Yes (consensus)	No	Yes	n.a. (threshold detector)	No
Wan *et al.* [6]	2017	No	No	Yes	n.a. (classifier)	No
Proposed approach		Yes (multi-agent)	Yes (OWL)	Yes	Exact and approximated (logic-based)	Yes (semantic distance)

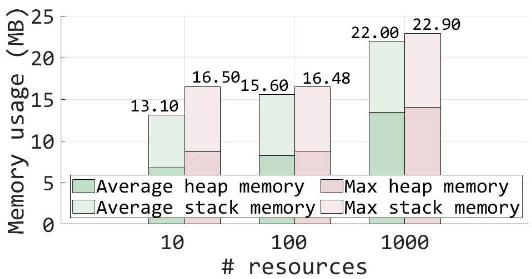


Fig. 8. Memory usage depending on the number of resources

VI. CONCLUSION

The paper proposed the object (b)logging framework as a general-purpose evolution of IoT. Smart objects combine classical machine learning algorithms and non-standard reasoning for feature and event detection from raw environmental data and ontology-based annotation. Compact, high-level logical descriptions progressively enrich a smart object's core knowledge in a micro-log and are published toward external devices and systems like in a blog through wireless ad-hoc links. Smart object networks become capable of decentralized coordination through semantic-based matchmaking to discover the most suitable tasks for the particular goal and context. The proposal has been validated in a case study concerning predictive maintenance, appearing as particularly suitable for Industrial IoT. Future work includes larger-scale implementation on a real testbed in order to perform further effectiveness evaluation. Improving the integration of machine learning and reasoning by means of more robust approaches is also under investigation. Finally, *stream publishing and reasoning* techniques will be integrated in order to increase scalability of knowledge sharing.

REFERENCES

[1] L. Da Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[2] I. L. Santos, L. Pirmez, L. R. Carmo, P. F. Pires, F. C. Delicato, S. U. Khan, and A. Y. Zomaya, "A decentralized damage detection system for wireless sensor and actuator networks," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1363–1376, 2016.

[3] M. Ruta, F. Scioscia, and E. Di Sciascio, "Enabling the Semantic Web of Things: Framework and Architecture," in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 345–347.

[4] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart Objects as Building Blocks for the Internet of Things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, 2010.

[5] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, and S. K. Das, "An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments," *IEEE Transactions on Mobile Computing*, vol. 16, no. 6, pp. 1502–1515, 2017.

[6] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, "A manufacturing big data solution for active preventive maintenancez," *IEEE Transactions on Industrial Informatics*, 2017.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] E. O'Toole, V. Nallur, and S. Clarke, "Decentralised detection of emergence in complex adaptive systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 12, no. 1, p. 4, 2017.

[9] R. De Virgilio, E. Di Sciascio, M. Ruta, F. Scioscia, and R. Torlone, "Semantic-based RFID Data Management," in *Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks*. Springer, 2011, pp. 111–142.

[10] S. N. Han, G. M. Lee, and N. Crespi, "Semantic context-aware service composition for building automation system," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 752–761, 2014.

[11] M. I. Ali, P. Patel, S. K. Datta, and A. Gyrard, "Multi-Layer Cross Domain Reasoning over Distributed Autonomous IoT Applications," *Open Journal of Internet Of Things (OJIOT)*, vol. 3, no. 1, pp. 75–90, 2017.

[12] C. Perera, A. Zaslavsky, C. Liu, M. Compton, P. Christen, and D. Georgakopoulos, "Sensor Search Techniques for Sensing as a Service Architecture for the Internet of Things," *Sensors Journal, IEEE*, vol. 14, no. 2, pp. 406–420, 2014.

[13] M. Liu, Y. Xu, H. Hu, and A.-W. Mohammed, "Semantic Agent-Based Service Middleware and Simulation for Smart Cities," *Sensors*, vol. 16, no. 12, p. 2200, 2016.

[14] 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 (IJSWIS)*, vol. 10, no. 4, pp. 77–100, 2014.

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

[16] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[17] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[18] 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.

[19] M. Ruta, F. Scioscia, A. Pinto, F. Gramegna, S. Ieva, G. Loseto, and E. Di Sciascio, "Coap-based collaborative sensor networks in the semantic web of things," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2018.