Agri-IoT: A Semantic Framework for Internet of Things-enabled Smart Farming Applications

Andreas Kamilaris*, Feng Gao†, Francesc X. Prenafeta-Boldú* and Muhammad Intizar Ali†
*GIRO Joint Research Unit IRTA-UPC, Barcelona, Spain
Email: firstname.lastname@irta.cat
†Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
Email: firstname.lastname@insight-centre.org

***Abstract*—With the recent advancement of the Internet of Things (IoT), it is now possible to process a large number of sensor data streams using different large-scale IoT platforms. These IoT frameworks are used to collect, process and analyse data streams in real-time and facilitate provision of smart solutions designed to provide decision support. Existing IoT-based solutions are mainly domain-dependent, providing stream processing and analytics focusing on specific areas (smart cities, healthcare etc.). In the context of agri-food industry, a variety of external parameters belonging to different domains (e.g. weather conditions, regulations etc.) have a major influence over the food supply chain, while flexible and adaptive IoT frameworks, essential to truly realize the concept of smart farming, are currently inexistent. In this paper, we propose Agri-IoT, a semantic framework for IoT-based smart farming applications, which supports reasoning over various heterogeneous sensor data streams in real-time. Agri-IoT can integrate multiple cross-domain data streams, providing a complete semantic processing pipeline, offering a common framework for smart farming applications. Agri-IoT supports large-scale data analytics and event detection, ensuring seamless interoperability among sensors, services, processes, operations, farmers and other relevant actors, including online information sources and linked open datasets and streams available on the Web.**

***Keywords*-Internet of Things; Smart Farming; Agriculture; Framework; Semantic Web.**

I. Introduction

The development of highly accurate embedded sensors measuring the environmental context inside farms has led to the enablement of *precision agriculture* [1], for improving productivity and increasing yields and profitability, reducing the environmental footprint, by techniques such as more efficient irrigation, targeted, more precise use of fertilizers and pesticides for crops, as well as food and antibiotics for animals. Precision agriculture enables the vision of *smart farming*, which is about real-time data gathering, processing and analysis, as well as automation technologies on the farming procedures, allowing improvement of the overall farming operations and management, and more informed decision-making by the farmers. Farming is highly unpredictable, due to its large dependency on weather and environmental conditions (e.g. rain, temperature, humidity, hail), unpredictable events (e.g. animal diseases, pests), as well as price volatility in agricultural markets. This implies the need for large-scale agriculture-related frameworks that harness sensor/automation technologies and data analytics, in order to help the farmers become informed about their farms' conditions and risks early enough, in order to take proper counter-measures and protect their crops/livestock and overall production.

The Internet of Things (IoT) [2] is a perfect match for smart farming due to its highly interoperable, scalable, pervasive and open nature. Realizing this enormous potential of IoT technologies for smart farming, IoT is now gaining momentum in the agricultural sector. For example, nowadays it is mandatory for all Australian farmers to affix passive RFID ear tags to their cattle and to report movements between farms to an online national database [3]. The IoT-based environment is a leap forward for the smart farming and has a great potential to revolutionize the traditional farming techniques. This can be achieved through IoT frameworks targeting smart farming, as such frameworks have been successfully applied in other, relevant domains such as smart cities, healthcare and smart homes [4]–[8], bringing seamless connectivity and advanced interoperability - by means of open standards (e.g. 6LoWPAN, ZigBee, CoAP, ISOBUS, SigFox, REST, MQTT, XMPP) and semantics (e.g. RDF, OWL, SWE, SensorML) - among actors, components, devices, processes and platforms.

IoT frameworks for smart farming can bring various benefits, including reducing the risk of vendor lock-in, adopting machinery and sensing/automation systems from different companies, as these could become easily interoperable in the overall farm's smart system, easier data exchange among different, heterogeneous components, increased automation with less effort by employing internet standards etc.

Being motivated by the above benefits and potential of IoT applied in smart farming, considering the non-existence of complete, reliable, well-established and bulletproof solutions and frameworks yet in this area, we propose Agri-IoT as a highly customizable online platform for IoT-based innovative data analytic solutions, enabling large-scale data processing, analysis and automatic reasoning based on real-time streams of data coming from a variety of sources, such as sensory systems, surveillance cameras, hyperspectral images from drones [9], online weather forecasting services, social media streams for fast identification of events (e.g. hazards, earthquakes, floods), information, warning and alerts from governmental organizations, such as the Ministry of Agriculture etc.

978-1-5090-4130-5/16/$31.00 ©2016 European Union;

Agri-IoT has the capabilities of combining and analyzing data streams such as the aforementioned ones, helping the farmers in more informed decision-making in near real-time and fast reaction to changes and unpredictable events. For example, combining sensory data about soil fertility (see Section IV-B), together with web services for weather forecasting, then better decisions could be made about more precise irrigation and fertilization of the crops. In this paper, we explain the operation of Agri-IoT (Section III), evaluating its performance by means of two demanding smart farming scenarios (Section IV), discussing how IoT frameworks could offer new possibilities in the farming industry (Section V).

II. Related Work

Since IoT frameworks and platforms are still inexistent or immature for agriculture, related work involves IoT frameworks in relevant domains, especially smart cities. We consider mostly smart city-based IoT frameworks (and not frameworks in other domains such as healthcare, smart homes, sports, participatory sensing etc.), because the requirements of smart farming are similar to those of urban environments: scalability, support of heterogeneous data streams, multiple actors/users, real-time analysis and reasoning, decision support and real-world services from embedded sensors.

Regarding IoT frameworks in the farming sector, the work in [10] proposes an architecture of a management information system for intelligent agriculture, based on IoT. Although this paper describes an information system for smart farming, it does not provide details on integrating with IoT. In order to design a truly interoperable IoT-based smart farming ecosystem, semantics of information and ontologies describing the relationship between data is required. Towards this goal, Taylor et al. [3] use semantic web technologies to define alert conditions in the farm, and to publish data in a stream management system. The Global Sensor Network (GSN) [11] middleware is employed, while summaries of sensor data are translated to RDF, based on the Semantic Sensor Network (SSN) ontology [12]. The aforementioned works, although promising, are still immature and mostly abstract.

Focusing on smart cities, we can identify more complete, powerful and scalable frameworks, providing better performance and scalability. Such frameworks include the ODAA platform [4], providing open data access to IoT data collected from the City of Aarhus, and Spitfire [5], which uses semantic technologies to provide a uniform way to search, interpret and transform sensory data. IBM Star City [6] supports real-time IoT data analytics for event detection pertaining to the traffic domain while CityPulse [7] combines a service-oriented approach and complex event processing for on-demand discovery and integration of urban data streams. Moreover, OpenIoT [13] and IoT-A [14] are software platforms for discovering and semantically annotating sensory data streams, and PLAY [8] provides an event-driven middleware able to process complex event detection in large heterogeneous systems. Although the aforementioned frameworks do provide numerous features and services for semantic-based sensory data processing and analysis, they are dedicated for urban environments, designed for city-specific event and notification patterns, device and service discovery, fault tolerance and reusability. Finally, it is worth mentioning the FIWARE platform [15], which provides a powerful set of APIs that ease development of smart applications in multiple vertical sectors.

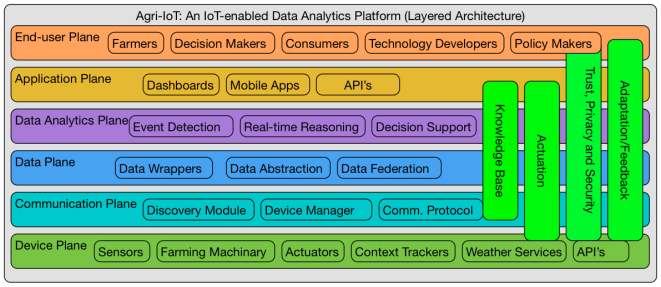

Fig. 1. The Agri-IoT ecosystem layered architecture.

Our work is one of the first efforts related to effectively bringing the IoT in the agricultural sector, in the same direction as [3], [10], considering advancements in smart city frameworks [7], [13]–[17], by reusing some of their well-defined software components, proposing Agri-IoT as a complete, interoperable, flexible and adaptable, large-scale data analytics framework for smart farming based on IoT standards and semantic web technologies.

III. Agri-IoT Framework

The Agri-IoT data analytics platform is illustrated in Figure 1. It is composed of multiple layers, both lower level (device, communication planes), intermediate layers (data, data analytics) and higher layers (application, end user planes). At each layer, various software components perform specific operations, related to data acquisition, modeling, analysis or visualization. Since well-defined relevant software components already exist, developed in IoT and smart city-related projects [7], [13]–[15], [18]–[21], we have focused on reusing those components, instead of re-inventing the wheel, according to the particular needs of smart farming, to cover most if not all of the smart farming requirements and scenarios as described in introduction. Each software component acts as a single entity, with its own open API, which allows to provide a flexible distributed architecture, where applications can integrate components from different layers based on their specific needs. In this way, components become plug and play and can be selectively used according to particular agricultural applications' requirements. Thus, based on the proposed ecosystem depicted in Figure 1, we have selected and combined various software components, appropriate for the domain of smart agriculture, which constitute the Agri-IoT framework, presented in Figure 2. Agri-IoT can integrate, manipulate and process a huge variety of cross-domain streaming data sources in a flexible and extensible way, using standardized methods for data acquisition following IoT principles, employing semantics.

A. Main Components

The main components of the Agri-IoT framework are described below.

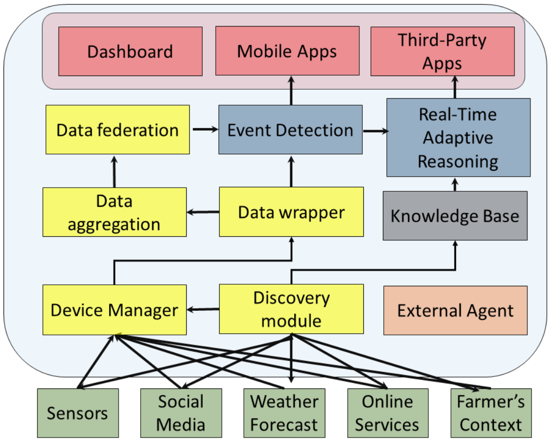

Fig. 2. Agri-IoT Architecture.

*a) **Data wrapper**:* (source: CityPulse) offers a generic way to describe characteristics of sensors using sensory metadata, containing general information about the data stream. A semantic annotation module enables to annotate the parsed sensory data (see Section III-B).

*b) **Device manager**:* (source: FIWARE IoT Back-end) automatically manages IoT devices, removing the need for human operators, providing the necessary tools for autonomic management processes to enforce decisions at a later stage. Manages device identity and authorization, considers reliability of data streams (e.g. real-time checking if values fall into specific limits) and fault recovery.

*c) **Discovery module**:* (source: FIWARE IoT Edge) ensures scalable registration and discovery of IoT devices and services in real-time, in a plug and play way. These devices can be either located at the same physical space (e.g. inside the farm) or remotely, accessed through the internet/web.

*d) **Data aggregation**:* (source: IoT-A) deals with large volumes of data using time series analysis and data compression techniques to reduce the size of raw sensory observations delivered by the data wrappers.

*e) **Data federation**:* (source: CityPulse) answers users' queries, e.g. the amount of fertilizer needed to apply over some area. This component first finds relevant streams according to the requirements specified in the request. Then, it translates users' requests into RDF Stream Processing (RSP) queries and evaluates the queries to obtain results. As IoT-based smart farming involves fast changing data from real-world sensors and online services, as well as real-time processing and analytics based on semantics, we argue that RSP is an appropriate technology to be employed. RDF-based reasoning is supported through CSPARQL [22] and CQELS [23], which are RDF query languages managing continuous data streams.

*f) **Event detection**:* (source: CityPulse) provides tools for processing annotated and aggregated data streams to obtain farm events, such as need for irrigation, sick animals or pest identification in crops.

*g) **Real-time adaptive reasoning**:* (source: CityPulse) takes into account farmer's preferences and dynamic contextual farm-related information (represented by real-time events), in order to provide optimal decision support in real-time. Provides reliable, accurate and fast decision support to the farmer, based on farm's conditions as measured by the smart sensing technology available.

*h) **External agent**:* (source: Agri-IoT in-house developed) addresses interoperability, device heterogeneity, data handling and protocol adaptation. Plays an important role for virtualising objects, services, methods and processes, considering user's identity and authorization.

*i) **Dashboard**:* (source: ThingSpeak, freeboard) provides immediate and intuitive visual access to the results of processing and analysis of data and events.

*j) **Mobile Apps**:* (source: MapYourMeal, FoodLoop) are built on top of the other components, similarly to the dashboard, and use their APIs to offer various services to their mobile users, either to the farmers for real-time information and fast decision making, or to the consumers and transport agents at the sales points for more transparency.

*k) **Knowledge base**:* (source: OpenIoT) provides service metadata for sensor/data stream discovery.

B. Semantic Annotation

To semantically annotate data streams, Agri-IoT uses lightweight information models developed on top of well-known models, such as the SSN [12] and OWL-S [24] ontologies. Based on these, we describe streams coming from sensors deployed inside the farm using the Stream Annotation Ontology[1] and events detected relating to the farm using the Complex Event Ontology[2]. Related to agriculture, AGROVOC [25] and the Agricultural Ontology Service (AOS) [26] are popular vocabularies and ontologies. AgOnt [27] is another ontology targeting farming, built from agriculture terminologies and lifecycles including seeds, grains, transportation, storage and consumption. Through the aforementioned ontologies (using AgOnt for the agricultural domain), we semantically describe the sensors' real-time data streams, their metadata, the agricultural products they are referring to, and possible events occurring at the farm. When the real-time data streams are semantically annotated, we can use RDF Stream Processing (RSP) techniques (e.g. CSPARQL [22] and CQELS [23]) to easily process heterogeneous data streams. Using RSP, one can use SPARQL-like query languages to evaluate query patterns over static/dynamic knowledge, facilitating on-demand stream discovery (see Section IV-B). Figure 3 shows abstractly the Agri-IoT information model, combining the aforementioned ontologies together.

IV. Evaluation

To evaluate Agri-IoT, we focused on the feasibility of using RSP in agricultural applications. This is the most demanding operation on the framework, towards the goal of real-time processing, analysis and event identification (see components *Data federation* and *Event detection* in Section III-A). We used a machine running Debian GNU/Linux 6.0.10, with 8-cores of 2.13 GHz processor and 64 GB RAM. Two realistic scenarios

[1] SAO. http://iot.ee.surrey.ac.uk/citypulse/ontologies/sao/sao

[2] CEO. http://citypulse.insight-centre.org/ontology/ces/

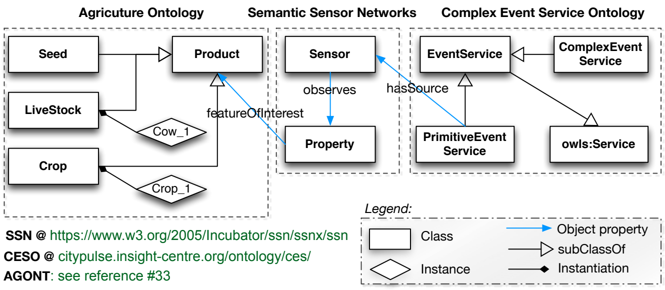

Fig. 3. Agri-IoT information model.

were considered, using CityBench [28] to evaluate the queries' performance. We note that the real potential of IoT is revealed when sensing-based data sources become combined with relevant online services, such as weather forecasting, news feeds etc. The following scenarios do not involve online services though, as the focus is mostly on performance and scalability of the framework in terms of managing and analyzing large numbers of data streams coming from any source.

A. *Scenario A: Fertility management of dairy cows*

Heat measurement is used in fertility management as an indication of insemination timing. Automatic heat detection is a tool to assist producers in better managing fertility.

In this scenario, we assume the farm has 1,000 cows, each with a wearable sensor attached, measuring periodically its heat status, sending the observations (annotated with SSN) to a dedicated sensor data stream every second. Listing 1 shows an example of the semantic annotations for cows and sensors using the Agri-IoT information model (see Section III-B).

```
@prefix st: <http://agri-IoT.sensorTaxonomyNamespace>.
@prefix ssn: <http://agri-IoT.ssnNamespace>.
@prefix ag: <http://agri-IoT.agricuturalOntology> .
@prefix : <http://agri-IoT.sensorRepositoryNamespace>.
:sampleSensor a st:TempSensor; ssn:observes st:SurfTemp_1.
st:SurfTemp_1 a ssn:Property; ssn:isPropertyOf :cow_1.
:cow_1 a ag:Cow; ag:age "3.5"^^xsd:double;
ag:weight "xyz"^^xsd:double.
ag:Cow rdfs:subClassOf ssn:FeatureOfInterest.
```

Listing 1. Semantic annotation for a temperature sensor attached to a cow

```
SELECT ?sensorId WHERE{ ?sensorId ssn:observes ?property.
?property ssn:isPropertyOf ?cow. ?cow ag:age ?age.}
FILTER(?age > 1.5 && ?age < 2.5)
```

Listing 2. Static SPARQL query for relevant sensors

To monitor cow fertility, farmers need to be notified about the heat of cows which are 18 to 30 months old. Listing 2 shows a static SPARQL query to find these cows and identify relevant sensor streams. Using the results from Listing 2, farmers may deploy dynamic CQELS/CSPARQL queries similar to the query shown in Listing 3 to consider in real-time the cows' fertility potential. To evaluate the performance of RSP engines, we deploy different numbers of simultaneous queries similar to the one in Listing 3 over CQELS and CSPARQL engines, and observe the query latency and memory consumption. Notice that under this experiment setting, n queries are deployed over n separate sensor streams. The solid lines in Figures 4 and 5 show the performance of CQELS and CSPARQL in this scenario using separated sensor streams, respectively.

```
SELECT ?cowId ?obId ?value WHERE{
?property a sr:SurfaceTemperature.
?property ssn:isPropertyOf ?cowId.
### each sensor report to a separate stream
STREAM <http://agri-IoT.sensorRep#sensorId1> [range 10s]{
  ?obId a ssn:Observation.
  ?obId ssn:observedProperty ?property.
  ?obId ssn:observationSampleTime ?time.
  ?obId ssn:observationResult ?result.
  ?result muo:numericalValue ?value.} }
FILTER (?value>30.0)
```

Listing 3. CQELS query for temp. monitoring over sensor streams

The results show that, although CQELS query performance has dropped significantly when handling over 700 queries/sensor streams, it is in general better at handling multiple queries than CSPARQL, in terms of query latency and memory consumption. In our experiment, CSPARQL failed to handle more than 500 queries stably, i.e., it may stop producing results when handling a larger number of queries.

In the experiment above, fertility detection task is actually carried out in two steps: a static discovery step to identify relevant streams and then a dynamic query step. The use of ontologies improves the interoperability for both static and dynamic information. Moreover, by leveraging RSP the static discovery process, which is a query on stream metadata, can be integrated within the dynamic stream query. Hence, the first step can be skipped and we can evaluate the queries directly over a merged sensor stream (i.e. instead of subscribing to individual sensor streams, we listen to all sensor updates and merge them into a single stream). The merged query is shown in Listing 4, and the latency and memory consumption for merged queries are shown as dashed lines in Figures 4 and 5.

```
SELECT ?cowId ?obId ?value WHERE{
  ?property ssn:isPropertyOf ?cowId. ?cow ag:age ?age.
FILTER(?age > 1.5 && ?age < 2.5)
 #### all sensor readings in a single stream
STREAM <http://agri-IoT.sensorRep#allsensors> [range 10s]{
  ?obId a ssn:Observation.
  ?obId ssn:observedProperty ?property.
  ?obId ssn:observedBy ?sensorId.
  ?ob ssn:observationResult ?result.
  ?result muo:numericalValue ?value. }
FILTER (?value>30.0)}
```

Listing 4. Merged CQELS query for temperature monitoring

Using a merged query reduces system complexity by combining static discovery with dynamic monitoring. However, an overhead is introduced by subscribing to all, including irrelevant, sensor streams. Moreover, the results in Figure 4 show that CQELS has a much higher latency when handling more frequent stream updates for the merged queries, and the memory consumption of CQELS is higher for the merged queries when handling more than 700 sensors. A possible reason for the decline in performance is the synchronization issue of the streaming function in CQELS. On the other hand, CSPARQL copes very well with the merged queries. The query

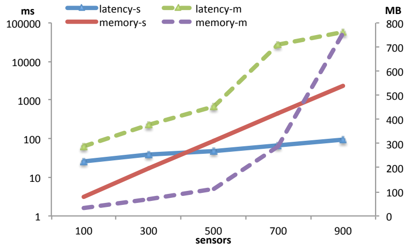

Fig. 4. Scenario A: CQELS performance

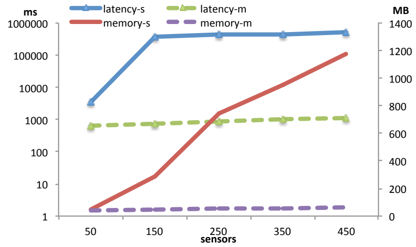

Fig. 5. Scenario A: CSPARQL performance

latency and memory consumption in CSPARQL are almost flat with the increasing number of sensors involved.

B. Scenario B: Soil fertility for crop cultivation

Soil fertility is important in order to cultivate vegetables and fruits. Soil composition (phosphorus, potassium and magnesium), salinity, and moisture levels are indicators for measuring quality and conditions of the soil [29]. Soil fertility is an outcome of measuring those parameters[3] and it is important in order to know when to cultivate, when to irrigate or fertilize as well as how to improve the yields. In this scenario, we considered a set of soil sensors measuring soil index (for soil composition), salinity and moisture. Let us now assume that the farmers are using Agri-IoT to know when it is a good time to cultivate[4]. The query for soil fertility created by Agri-IoT may look like the one shown in Listing 5. This query is more complicated and expensive than the previous ones, as each sensor produces not only one but three different observations each time. As a result, more query patterns are required. Also, there are more filters and aggregation functions used. This query counts the number of sensors reporting values indicating whenever conditions are appropriate for cultivation. If the number is above a threshold, e.g., half of the total sensors deployed in the same land (assuming a heterogeneous land in edaphological terms), then a notification is delivered to the farmers. We deploy queries similar to the one in Listing 5 with different number of sensors and different sensor update frequencies and record the query latency and memory consumption in Figures 6 and 7, respectively.

In this experiment, we tested merged queries because of the need of aggregating functions, only showing results for CSPARQL, since CQELS was not able to stably handle queries after 10-20 minutes involving more than 150 sensors. For the same reason, some lines in Figures 6 and 7 terminate before others, when the engine becomes unstable. In Scenario A, we used a time window of 10 seconds and a sensor update frequency of 1.0 Hz. Since each sensor update in this scenario contains three observations, we need to slow down the update frequency to 1/3 Hz, if we want to keep same number of triples in the time window as the first scenario. Considering the fact that soil cultivation activity is much less time-sensitive[5] than animal fertility management, we tested the query performance with sensor frequency from 0.1 Hz to 0.3 Hz. In reality, a much lower frequency is likely to be used, for managing cost in energy, network etc. The results in Figures 6 and 7 show that higher sensor frequency introduces more latency and likelihood of causing unstable states for the CSPARQL engine, without significant effect on memory consumption.

```
SELECT (count(distinct ?soilsensorId) as ?cnt) WHERE{
?salinity a ag:Salinity. ?salinity ssn:isPropertyOf ?soil.
?moisture a ag:Moisture. ?moisture ssn:isPropertyOf ?soil.
?soilComposition a ag:SoilComposition.
?soilComposition ssn:isPropertyOf ?soil.
?soil ag:hasLocation ?loc. ?loc ag:hasLatitude ?lat.
?loc ag:hasLongitude ?long.
STREAM <http://agri-IoT.sensorRepy#allSensors> [range 60s]{
?obIdsalinity ssn:observedProperty ?salinity.
?obIdsalinity ssn:observedBy ?soilsensorId.
?obIdsalinity ssn:observationResult ?salinityResult.
?obIdmoisture ssn:observedProperty ?moisture.
?obIdsoilComposition ssn:observedProperty ?soilComposition.
?salinityResult ssn:hasValue [ muo:numericalValue ?value1].
 ### similar patterns omitted for brevity
FILTER (?value1 < 6.0 && ?value2 >= 60 && ?value2 <= 80
      && (?value3 = 3 || ?value3 = 4) )}
FILTER (?lat = "53.289894"^^xsd:string
      && ?long = "-9.0742363"^^xsd:string) }
GROUP BY ?soil HAVING (?cnt>=75)
 ### threshold varies based on total number of sensors
```

Listing 5. CQELS query for temp. monitoring over sensor streams

V. Discussion

Findings indicate that Agri-IoT offers satisfactory performance even in demanding scenarios with large numbers of simultaneous data streams and sensors, and can be used in medium-to-large farms (100-300 sensors deployed at the field) for performing real-time stream processing and reasoning, based on IoT and semantic web technologies, helping the farmers in decision making and fast reactions to events happening. CQELS has performed better in terms of query latency and memory consumption, but CSPARQL is more scalable.

Some limitations of Agri-IoT include dynamicity, autonomy and full adaptability to heterogeneity. On-demand discovery, integration and complex event processing could become optimized to provide robust real-time analytic solutions over heterogeneous data streams originating from agricultural sensors. As future work, we aim to address these limitations, making Agri-IoT flexible and adaptable to every farming scenario,

[3] Other parameters not considered in our evaluation, might include trace elements (Fe, Mg, Mn, Cu, Zn, S, etc.), soil pH and electrical conductivity.

[4] Soil fertility differs according to the particular soil, weather conditions, crops used etc. In our experimentation, we used as general values as possible.

[5] It is noticeable that experimented sensor frequencies are relatively high, to put the system under stress. In a realistic scenario of monitoring soil fertility, frequencies could be in the range of few hours up to some days.

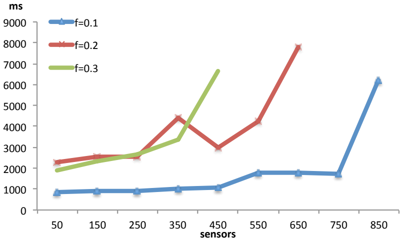

Fig. 6. Scenario B: CSPARQL latency

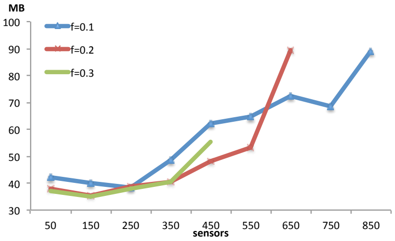

Fig. 7. Scenario B: CSPARQL memory

with true plug and play support of heterogeneous sensors, information and solutions. We plan to fully involve semantic web technologies for data integration, stream processing and reasoning, including mechanisms of complex event service discovery/reusability. We will work on improving RSP performance for concurrent queries, high stream rates and large triple window sizes (e.g. by realizing a distributed RSP engine), using the experimental results of this paper as baseline. Finally, we will focus on improving existing agricultural ontologies and semantic models [25]–[27], developing a complete knowledge representation framework in the form of linked data for IoT streams, enabling linking of multiple streams (e.g. sensors, social networking feeds, weather forecasting etc.), associating their descriptions to the domain knowledge.

VI. Conclusion

In this paper, we have described Agri-IoT, an IoT-based framework applying real-time stream processing, analysis and reasoning in the domain of agriculture, based on semantic web technologies, facilitating more informed and accurate decision making by farmers and event detection. We have investigated the introduction of IoT in smart farming and its opportunities, through the seamless combination of heterogeneous technologies, as well as the semantic integration of information from various sources (sensors, social media, connected farms, governmental alerts, regulations etc.), ensuring increase of production and productivity, better products' quality, protection of the environment, less use of resources (e.g. water, fertilizers), faster reaction to unpredictable events and more transparency to the consumer. Agri-IoT achieves this by offering interoperability between sensors, processes, data streams, farms as entities and web-based services, exploiting open data, making use of semantic technologies and linked web data. Our evaluation efforts focusing on two realistic and demanding farming scenarios indicate the good performance of the proposed framework in medium-to-large farms, while our discussion reveals the large opportunities arising in farming by introducing open standards and semantics based on IoT.

Acknowledgment

This research has been supported by the P-SPHERE project, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 665919. It has also been partially supported by Science Foundation Ireland (SFI) under grant No. SFI/12/RC/2289 and EU FP7 CityPulse Project under grant No.603095. http://www.ict-citypulse.eu

References

[1] Rodolfo Bongiovanni and Jess Lowenberg-DeBoer. Precision agriculture and sustainability. *Precision agriculture*, 5(4):359–387, 2004.

[2] Neil Gershenfeld, Raffi Krikorian, and Danny Cohen. The Internet of Things. *Scientific American*, 291(4):76–81, 2004.

[3] Kerry Taylor et al. Farming the web of things. *Intelligent Systems*, 28(6):12–19, 2013.

[4] Open Data Aarhus, 2016. http://www.odaa.dk.

[5] Dennis Pfisterer et al. SPITFIRE: toward a semantic web of things. *IEEE Communications Magazine*, 49(11):40–48, 2011.

[6] Freddy Lécué et al. Smart traffic analytics in the semantic web with STAR-CITY: Scenarios, system and lessons learned in Dublin City. *Web Semantics: Science, Services and Agents on the World Wide Web*, 27:26–33, 2014.

[7] CityPulse EU FP7 Project, 2016. http://www.ict-citypulse.eu/page/.

[8] Roland Stühmer et al. PLAY: Semantics-based Event Marketplace. In *Collaborative Systems for Reindustrialization*, pages 699–707. 2013.

[9] senseFly SA, 2016. https://www.sensefly.com/.

[10] Duan Yan-e. Design of intelligent agriculture management information system based on IoT. In *Proc. of ICICTA*, pages 1045–1049, 2011.

[11] Karl Aberer, Manfred Hauswirth, and Ali Salehi. A middleware for fast and flexible sensor network deployment. In *Proc. of VLDB*, pages 1199–1202. VLDB Endowment, 2006.

[12] Michael 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*, 17(1):25–32, 2012.

[13] OpenIoT EU FP7 Project, 2016. https://github.com/OpenIotOrg.

[14] IoT-A EU FP7 Project, 2016. http://www.iot-a.eu/public.

[15] FIWARE. The FIWARE Catalogue, 2016. http://catalogue.fiware.org/.

[16] Andreas Kamilaris et al. Bridging the Mobile Web and the Web of Things in Urban Environments. In *First Workshop on the Urban Internet of Things, in Proc. of IoT 2010*, Tokyo, Japan, 2011.

[17] Andreas Kamilaris et al. Integrating Web-Enabled Energy-Aware Smart Homes to the Smart Grid. *International Journal On Advances in Intelligent Systems*, 5(1), July 2012.

[18] ThingSpeak, 2016. https://thingspeak.com/.

[19] freeBoard, 2016. https://freeboard.io/.

[20] Map your meal Europe Aid funded project, 2016. http://www.mapyourmeal.org/.

[21] FoodLoop GmbH. FoodLoop, 2016. https://www.foodloop.net/en/.

[22] Davide F. Barbieri et al. C-SPARQL: SPARQL for continuous querying. In *Proc. of WWW*, pages 1061–1062. ACM, 2009.

[23] Danh Le-Phuoc et al. A native and adaptive approach for unified processing of linked streams and linked data. In *Proc. of ISWC*, pages 370–388. Springer, 2011.

[24] W3C. OWL-S: Semantic Markup for Web Services, 2016. https://www.w3.org/Submission/OWL-S/.

[25] FAO. AGROVOC Thesaurus, 2009. http://www.taxobank.org/content/agrovoc-thesaurus.

[26] Boris Lauser et al. From AGROVOC to the Agricultural Ontology Service/Concept Server. An OWL model for creating ontologies in the agricultural domain. In *Dublin Core Conference*, 2006.

[27] Siquan Hu et al. AgOnt: ontology for agriculture internet of things. In *Computer and Computing Technologies in Agriculture IV*, pages 131–137. Springer, 2010.

[28] Muhammad Intizar Ali, Feng Gao, and Alessandra Mileo. CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets. In *Proc. of ISWC*, pages 374–389. Springer, 2015.

[29] John Havlin et al. *Soil fertility and fertilizers: An introduction to nutrient management*, volume 515. Pearson Prentice Hall, 2005.