Semantic IoT Middleware-enabled Mobile Complex Event Processing for Integrated Pest Management

Francesco Nocera, Tommaso di Noia, Marina Mongiello and Eugenio Di Sciascio
Department of Electrical & Information Engineering, Polytechnic University of Bari, Via Orabona, 4, Bari, Italy

Keywords: Complex Event Processing, Cyber-physical Systems, Ontologies, IoT Middleware.

Abstract: Agricultural domain presents challenges typical of the Cyber-Physical Systems field and the hard-core of information technology industry of the new generation, such as Cloud computing and Internet of Things. In fact, modern agricultural management strongly relies on many different sensing methodologies to provide accurate information about crop, climate, and environmental conditions.
In this paper we propose an approach to model a mobile-driven and distributed Complex Event Processing solution which is enabled by an IoT middleware. The proposed framework is robust with reference to contextual and environmental changes also thanks to the exploitation of an ontological model.

1 INTRODUCTION AND MOTIVATION

More than one third of the world's labour force is employed in agriculture field, which is one of the most dangerous of all sectors and many agricultural workers suffer occupational accidents and ill health. Furthermore, informal employment, as a percentage of non-agricultural employment, exceeds 50 per cent in half of the countries with comparable data. In one-third of countries, it affects over 60 per cent of workers. (Horne et al., 2016)
As pointed out in the last Meridian Health Report, presented by The European House-Ambrosetti [1] about the allocation of the basic items of public expenditure grouped into ten macro categories (including Environment, Health,...), one of the macro objectives identified by the Prevention Plan is the identification of strategies for reducing or prevent potentially harmful environmental exposures and improving health outcomes. The main focus is on administration's methods for agrochemical inputs in order to assure adequate plant nutrition and plant protection through organic nutrient sources and Integrated Pest Management (IPM), respectively.
IPM is a pest management strategy formally developed in the 1950s by entomologists and other researchers in response to a widespread development in agricultural settings of pesticide resistance in insects and mites, outbreaks of secondary and induced insect and mite pests resulting from pesticide use, and transfer and magnification of pesticides in the environment (Flint and Van den Bosch, 2012). The UN's Food and Agriculture Organisation (FAO) [2] defines IPM as *"the careful consideration of all available pest control techniques and subsequent integration of appropriate measures that discourage the development of pest populations and keep pesticides and other interventions to levels that are economically justified and reduce or minimize risks to human health and the environment"*.
IPM emphasizes the growth of a healthy crop with the least possible disruption to agro-ecosystems and encourages natural pest control mechanisms. Initially focusing on biological control of insects and mites in agricultural systems, over the last 70 years IPM has assumed an amplified role, encompassing management of diseases and weeds as well as insects, mites and other arthropods in agricultural, horticultural, and urban settings. IPM emphasizes selecting, integrating, and implementing complimentary pest management tactics to maintain pests economically acceptable while minimizing negative ecological and social impacts of pest management activities (Flint, 2012).

As depicted in Figure 1 (left hand side), the major challenge in the agricultural sector is identified as *no timely help* and *inadequate knowledge flow* in connection with weather data at the required time.

[1] The report is available at http://www.ambrosetti.eu/.

[2] http://www.fao.org/

Nocera, F., Noia, T., Mongiello, M. and Sciascio, E.
Semantic IoT Middleware-enabled Mobile Complex Event Processing for Integrated Pest Management.
DOI: 10.5220/0006369506380645
In *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*, pages 610-617
ISBN: 978-989-758-243-1

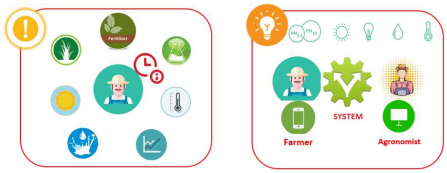

Figure 1: Challenges and proposed solution.

Agricultural domain presents challenges typically encountered in the realm of Cyber-Physical Systems (CPSs), such as: incomplete information, limited sources of information, knowledge and infrastructure deficiency, inadequate help and no timely support, external disturbances (weather), limited control authority (fertilizers cannot make a plant mature arbitrarily fast), etc.. Thanks to the spreading of sensors and to the diffusion of low-cost miniaturized computational resources, we are able to build objects that produce data and interact with each other thus producing a network of interconnected things, data, processes and services. On the other hand, CPSs are transforming the agriculture industry.

To solve these limitations and disadvantages, this paper proposes a smart infrastructure designed to process heterogeneous data sources input such as sensors data, weather data, and the agricultural knowledge collected into an ontology.

The proposed solution is based on a Complex Event Processing (CEP) (Luckham, 2002) module partially executed on mobile devices by introducing an Internet of Things (IoT) middleware (Razzaque et al., 2015; Fersi, 2015) that by using a programming language that supports reflection mechanism (Mongiello et al., 2016) serves as an interface between IoT components and the ontological knowledge. The proposed approach make possible: *(i)* a smoother and homogeneous communication among elements and a *(ii)* dynamic, configurable and extensible infrastructure. The identified solution and the relative instantiation in agricultural domain aims at becoming an instrument used for raising awareness about the use of treatments. In this way the farmer can *(i)* have access to all information related to the domain of interest at required time, *(ii)* create personalized defense plan, *(iii)* receive alerts of changing weather conditions and *(iv)* receive notification and recommendations about his treatment plans.

The remaining of the paper is organized as follows: Section 2 illustrates background and related work while Section 3 introduces the proposed approach. In the following section we instantiate and validate the approach in a real scenario. Conclusion and future work close the paper.

2 BACKGROUND AND RELATED WORK

There is no unified definition of a CPS. Generally, CPSs are defined as physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. Internet transformed how humans interact and communicate with one another, revolutionized how and where information is accessed.. Similarly, CPS will transform how humans interact with and control the physical world around us. CPSs perceive the physical world, process the various data by computers, and affect and change the physical world (Hu et al., 2012). He JiFeng presented the concepts of "3C": Computation, Communication, and Control. Examples of CPS include medical devices and systems, transportation vehicles and intelligent highways, defense systems, aerospace systems, robotic systems, process control, building and environmental control and smart spaces. CPS interact with the physical world, and must operate dependably, safely, securely, and efficiently and in real-time. CPS is an emerging area, which cannot work efficiently without proper software that handles both the data and the business logic. The software development of CPS is a critical issue because of its complexity in a large scale pragmatic system. Furthermore, an object-oriented approach (OOA) is often used to develop CPS software, which needs some improvements according to the features of CPS.

IoT Middleware and *domain knowledge* are the soul of a CPS. One of the most important challenge related to CPSs is knowledge collection and representation in a usable and efficient way as it is a cumbersome, time consuming and expensive process. Sharing of the same knowledge in different applications and its reuse without or with little modifications in solving separate problems is a critical factor in knowledge aggregation and dissemination. Among knowledge representation techniques, ontology formalisation can be used to model real world in a consistent, formal, manageable and reusable way (Jasper et al., 1999; Stevens et al., 2004). Ontologies can be used as a tool to handle several aspects of knowledge management such as knowledge representation, knowledge sharing and reuse, knowledge classification, knowledge evolution and knowledge search and retrieval (Alavi et al., 2001).

One of the most cited definitions of ontology as "a shared and common understanding of a domain that can be communicated between people and across application systems"(Gruber et al., 1993) leaves the room for several interpretations of the term (Guarino,

1998).
Various ontology-driven applications are developed in agricultural domain including knowledge systems (Ahsan et al., 2014; Pereira et al., 2012; Xie et al., 2007; Malik et al., 2015; Di Noia et al., 2016), expert systems (Kang and Gao, 2013; Barshe and Chitre, 2012) and advisory systems (Chaudhary et al., 2015). The most influential works are the following.
(Pereira et al., 2012) presented a Recommender System constructed as a Semantic Model that gives information on planning better intercropping.
(Ahsan et al., 2014) presented a knowledge Model based on Ontologies to study the process of knowledge acquisition, best practices in the agricultural domain. The knowledge Base contained Wheat Crop information.
(Malik et al., 2015) presented an ontology designed and developed for the agricultural domain in which various entities and relationships are represented. A knowledge intensive system is developed based on a small ontology.
(Barshe and Chitre, 2012) proposed an Agrosearch system; the prototypic interface uses a relational architecture for keyword maintenance while ontology can be saved in the shape of URI.
(Kang and Gao, 2013) demonstrated the ontology application in an agriculture information retrieval system. The process of knowledge retrieval analyses the characters of agricultural domain and the search mechanism was designed with knowledge retrieval ontology .
(Walisadeera et al., 2013) designed farmer centered ontology. Many motivation scenarios are made into a model with set of questions and ontology was developed for all stages of farming lifecycle.
(Chaudhary et al., 2015) presented an advisory system for the cotton farmers in Gujarat using cotton ontology, web services and Mobile Application Development Advisory system.
In recent developments, monitoring and agriproduct systems are developed using various others technologies. (Bo and Wang, 2011) presented and discussed in their work: *(i)* advantages of IOT and Cloud Computing in agriculture field; *(ii)* various applications of cloud computing and IOT in Agriculture and Forestry on safety of Agriproduct, agricultural information transformation and intelligent detection, precision irrigation and forest identification.
The IOT and Cloud Computing applications contribute huge development towards agricultural sector but it does not satisfy to meet all current challenges. IOT is closely related to Cloud Computing in a way that IOT obtains massive computing tools through cloud computing and cloud computing finds the best practicing channel based on IOT. Limited sources of Information, the agricultural data is scattered in different places, domains making it difficult to find the right information at the right time.
In the agricultural field, ontologies may provide farmers timely support to acquire information. Indeed, there are many well-established and authoritative controlled vocabularies, such as AGROVOC [3], CAB Thesaurus [4] and NAL Thesaurus [5].
AGROVOC is the most used vocabulary in several systems, covering all areas of interest of the FAO, including food, nutrition, agriculture, fisheries, forestry, environment etc. It is published by FAO and edited by a community of experts. It is possible to extract ontological concepts from AGROVOC thesaurus and use them to build domain specific ontologies, retrieving and organizing data in agricultural systems. These practices are important in order to make this information accessible to any user, especially regardless of age.

3 PROPOSED APPROACH

We now illustrate the proposed approach and the related CPS architecture identified. The whole infrastructure consists of two parts: the backend server consisting an IoT middleware that automatically perform actions according to the values received by the sensors of devices by matching the set of formal rules and the mobile device part principally based on a CEP module. Figure 2 shows a graphical schema of the proposed approach represented by means of an abstract architecture.

For the ***backend server part***(components contained in the light blue box in Figure 2) we propose the adoption of a reflective paradigm (Buschmann et al., 1996) for modeling an IoT middleware, by implementing the software design pattern *Reflection*.
The main concept in *Reflection* pattern is the distinction between *base-level*, *meta-level* and the *Meta-Object Protocol (MOP)* (Buschmann et al., 2007). The *base-level* contains the application logic basic software while the meta-level takes care of all the aspects that can change independently from the base level.
A *meta-level* contains those parts of the application that may vary at run-time, with the goal of creating new applications starting from existing base-level

[3]http://www.fao.org/agrovoc
[4]http://www.cabi.org/
[5]http://agclass.canr.msu.edu/

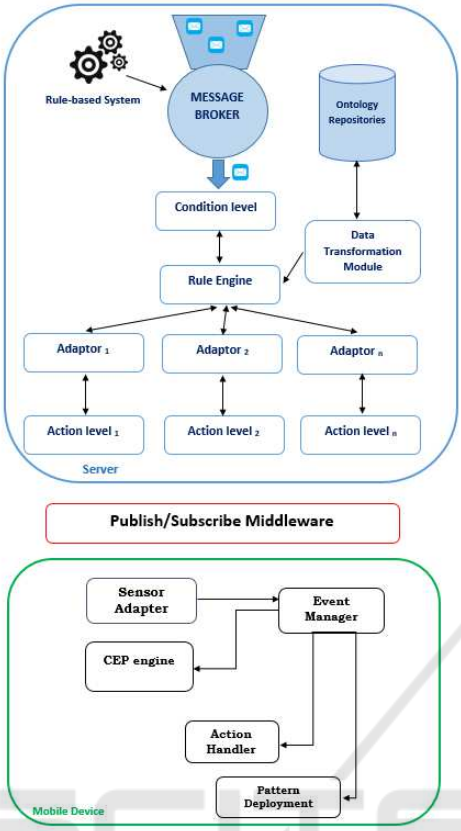

Figure 2: Graphical schema of proposed model.

classes. The adaptation of the meta level is performed indirectly thanks to the MOP, a specific Interface that also makes possible the change of connections between the base-level and meta-objects. A meta-object is defined as an object that manipulates, creates, describes, or implements other objects (including itself)(Buschmann et al., 2007). In this way, by using a programming language that supports reflection we can design a completely *configurable*, *extensible* system which *adapts* to different operating environments where it is also possible to change the structure of the objects themselves at run-time and then make the software system more *flexible*.

The *message broker* component models the Publish/Subscribe mechanism for defining the relationships between messages containing physical sensors information, and actions. The proposed approach maps an IoT middleware on the three levels of a reflection pattern according to the following matching: the *Condition* to the *Meta level*, the *Action* to the *Base level* and the *Rule* to the *Meta-Object Protocol*. RULES are *(i)* modeled through a data transformation module from the ontology and *(ii)* stored in a RULE-BASED SYSTEM. A RULE ENGINE is at the basis of the reasoning algorithm. The *Adaptor* component works as a driver and translates the received command in a real action. It is possible to use place different Adaptor for each action.

The ***mobile device part***(components enclosed in the green box in Figure 2) includes five components. The core component is the *Complex Event Processing (CEP) engine*. In recent years, CEP over event streams has become an increasingly important tool to extract relevant situational knowledge from distributed systems in real-time or "near real-time".

The concept of CEP was introduced by David Luckham in his seminal work (Luckham, 2002) as a "*defined set of tools and techniques for analyzing and controlling the complex series of interrelated events that drive modern distributed Information Systems(IS)". Event pattern rules* are used in CEP to create complex events representing the combined activities of sets of atomic events (Luckham, 2002). The key characteristic of a CEP system is its capability to handle complex event situations, detecting patterns, creating correlations, aggregating events and making use of time windows. Specification languages for event patterns are frequently inspired by regular languages and therefore have automata based semantics (Hopcroft et al., 2001). Several CEP systems have been developed in the last few years, each one proposing a different processing model. Currently, the most popular are: *StreamBase CEP*[6], *Esper*[7], *Apache Flink*[8], *StreamDrill*[9]. *Sensor Adapters* connect to sensors and translate information obtained from sensors into events format. All formatted events are sent to *Event Stream Management* component, which dispatches event streams to other components such as the CEP engine or action handlers or sends the events to the server through publish methods. *Pattern Deployment* deploys/un-deploys the patterns, which are dispatched by server to the CEP engine on the mobile device. In order to interact with a user, *Action Handler* execute actions like playing alarm audio, displaying alerts and/or recommendations on smartphone in case the pre-defined trigger events are detected. The two part communicate through a publish/subscribe middleware composed by a Distributed Service Bus (DSB), a Google Cloud Messaging service and a Subscription Web Service.

[6]https://www.streambase.com/
[7]http://www.espertech.com/esper/
[8]http://flink.apache.org
[9]https://streamdrill.com/

4 INSTANTIATION AND VALIDATION OF THE MODEL

In this Section, to explain the abstract architecture defined in Section 3 let us consider its instantiation in a real use case scenario. We instantiate the model for an Italian company.

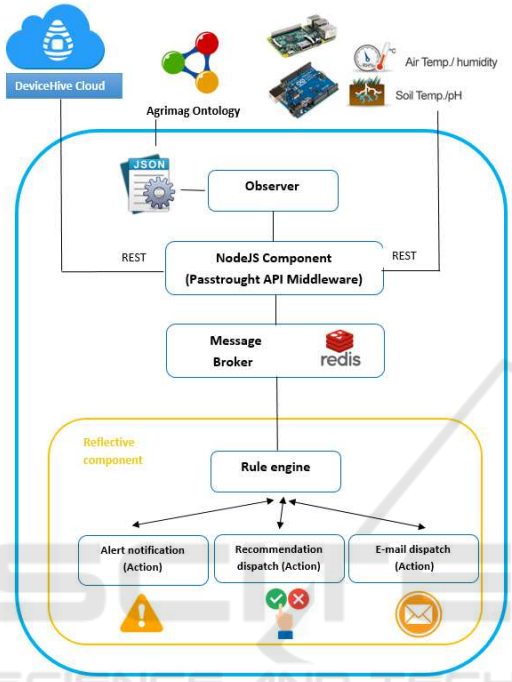

Figure 3: Architectural schema of server part.

For the mobile device part (see Figure 2), considering the limited computing resources on mobile devices we use the light weight **Esper engine** as CEP engine. Esper enables rapid development of applications that process large volumes of incoming messages or events, regardless of whether messages are historical or real-time in nature. Esper filters and analyzes events in various ways, and responds to conditions of interest. Esper language specification is called *Event Processing Language (EPL)*. It is a SQL-like language with *SELECT, FROM, WHERE, GROUP BY, HAVING* and *ORDER BY* clauses.
As depicted in Figure 3, we choose the following technologies for the ***backend server part*** described in Section 3 (Figure 2):

- *DeviceHive*[10] *as IoT middleware.* The reasons for this choice are: ease of installation, the rich documentation, the high integration with a wide range of programming languages and IoT protocols. In particular, we used the shell instance *Cloud Playground*;

- *Redis as a Message Broker.* Redis[11] is a NoSQL DBMS and allows a system to translate a message from the messaging protocol of the sender to the recipient's messaging protocol. Through a decoupling between publishers and subscribers, Redis guarantees a greater scalability and a dynamic network topology;

- *Node JS as Event-driven component.* It receives data about the sensor and the variable of the sensor through a REST interface;

- *Observer.* This component observes rules extracted from the Rule based-systems. In our implementation, the rule based systems is a configuration file.

The system will automatically perform actions according to the values received by the sensors of devices by matching the set of formal rules. Variables to be monitored are air, temperature, humidity, fine dust emissions.
The hardware used to provide a source of data is based on the *Arduino* platform, a small electronics board equipped with a microcontroller, which is used to quickly achieve hardware prototypes. In addition to the *Arduino* board we have used the *RaspberryPi B+* board to send and receive data from/to the local server which then makes a transition of data to *Cloud DeviceHive* servers.
Within the component, a configuration JSON file contains the definition of the rules on ontology data extracted and the actions to take if the rule is fired. The OBSERVER (of the data flow) notifies the NODE JS that forwards to the MESSAGE BROKER about the extracted active rule. The active rule together with the information about sensors and variables, (the predicate of our Condition) is published on the *Redis channel*. Information about the arrival of new data is published on the message channel . The MESSAGE BROKER works as a through for data flow and the active rule that are forwarded to the Reflective part. Receivers subscribe to the MESSAGE BROKER and are notified of the message. The MESSAGE BROKER forwards the whole message made up of sensor, variable, and the rule, (extracted from the configuration file) to the Rule Engine that executes it.
Reflection is applied thanks to the signing of the receiver component of the Redis channel.
The *Condition Level* and the RULE ENGINE enable

[10] http://devicehive.com/

[11] http://redis.io/

the action level (Figure 3 depicts the actions implemented) of the reflective component.
In this instantiation we constructed an OWL 2 ontology called *Agrimag* based on the AGROVOC thesaurus. The ontology contains the main knowledge related to the agricultural field, in particular the aspects concerning pest management and control, with a view to respecting the environment and operator health. Figure 4 shows a screenshot of the *Protégé*[12] editor GUI illustrating the Class and Object Property hierarchies of our ontology.

The ontology describes the main concepts related to the management of cultivation and chemical treatments necessary to ensure agricultural production, with particular reference to the guidelines issued by the Italian Ministry of Agriculture.

Table 1: Ontology metrics.

Metrics	
Axiom	7694
Logical axiom count	6297
Class count	32
Object property count	35
Data property count	07
Individual count	2012
Class axioms	
SubClassOf axioms count	24
DisjointClasses axioms count	02
Object property axioms	
SubObjectPropertyOf	18
InverseObjectProperties	03
ObjectPropertyDomain axioms count	21
ObjectPropertyRange axioms count	22
Data property axioms	
SubDataPropertyOf	01
FunctionalDataProperty	01
DataPropertyDomain	05
DataPropertyRange	05
Individual axioms	
ClassAssertion axioms count	1893
ObjectPropertyAssertion axioms count	3161
DataPropertyAssertion axioms count	1139
Annotation axioms axioms	
AnnotationAssertion	1323

The most important classes are the following:
$< Crop >$: identifies the planted crop in the field. Each culture is characterized by a specific set of harmful agents that affect, as well as by a set of active ingredients and commercial authorized products;
$< Pest >$: modeling agents harmful to plants (eg. Mushrooms, Insects, Weeds,etc.), which the farmer has to worry about finding a remedy through the use of Active Substance;
$< ActiveSubstance >$: is a chemical compound used in agriculture as an antagonist of one or more kinds of pathogens/pests;
$< CommercialProduct >$: indicates a formulation commercially available. Its spectrum of action is determined by its composition (made of ActiveSubstance), as well as legislative constraints, regulations and authorizations imposed by the Ministry of Health;
$< InstitutionalActiveSubstanceUsage >$: models a ternary relationship that relates a *Pest* with an authorized *ActiveSubstance* to a specific *crop*.

Table 1 summarizes the metrics associated to our ontology which is publicly available.[13] Possible use cases identified for a farmer are the following:

- query the ontology via a mobile app or company Web page, to learn about the solutions to be implemented (interventions and curative active substance, notes) to cope with all possible diseases for each crop;
- the mobile device through the GPS module locates the exact position of the field on the map and can receive information related to defense plans for their treated cultures;
- based on treated cultures (stored into mobile device), it is possible to receive recommendations/alerts about their treatment (period, healing product, doses, etc.)

Figure 5 shows a screenshot of the Web application concerning the creation of the defense plan. Once selected the crop (*Cherry tree*) and one related disease (*Archips rosanus*), the farmer mobile device displays the information solution for each selected disease.

5 CONCLUSION AND FUTURE WORK

Agriculture in urban areas have become a new trend. In Cyber-Physical Systems(CPSs) perspective, this domain presents typical challenges such as incomplete information, limited sources of information, knowledge and infrastructure deficiency, inadequate help and no timely support, limited control authority and so on. One of the challenges in building CPS is the way software will be developed and composed on the top of flexible infrastructures and integration architectures.

[12] http://protege.stanford.edu/

[13] A full description of Ontology metrics is available at http://protegewiki.stanford.edu/

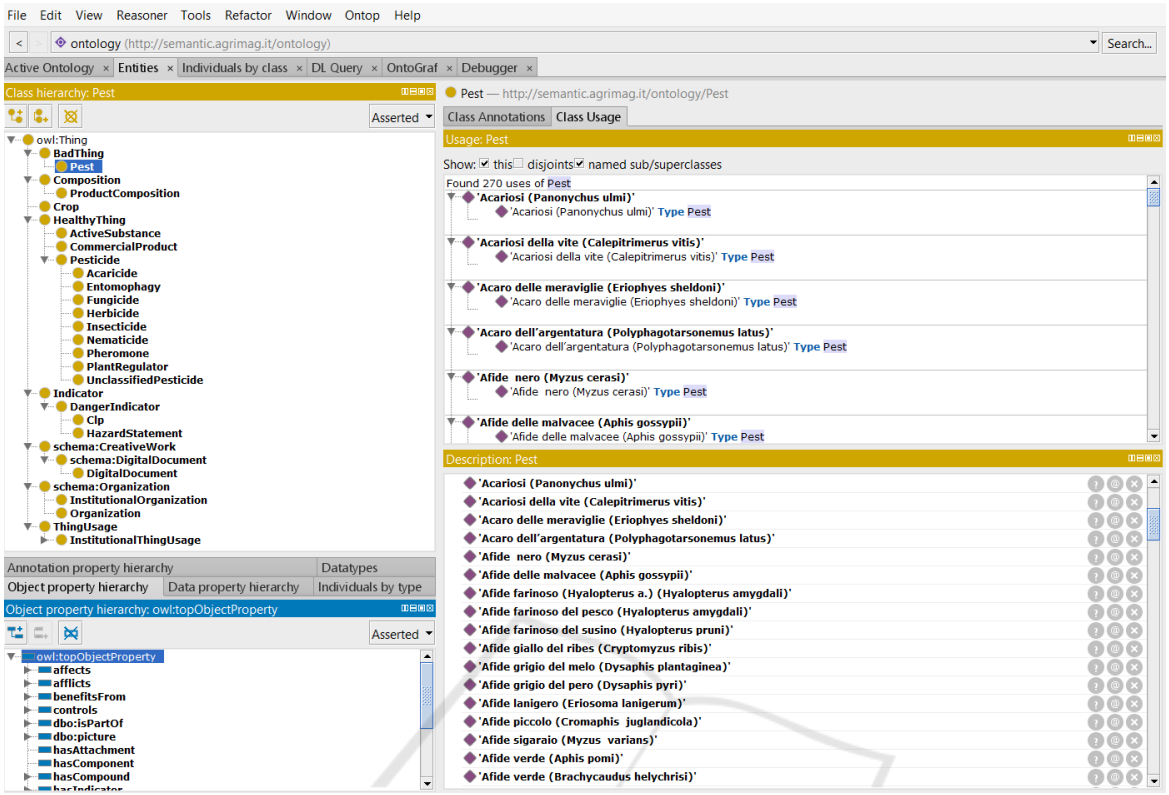

Figure 4: Entities Tab *Protégé*.

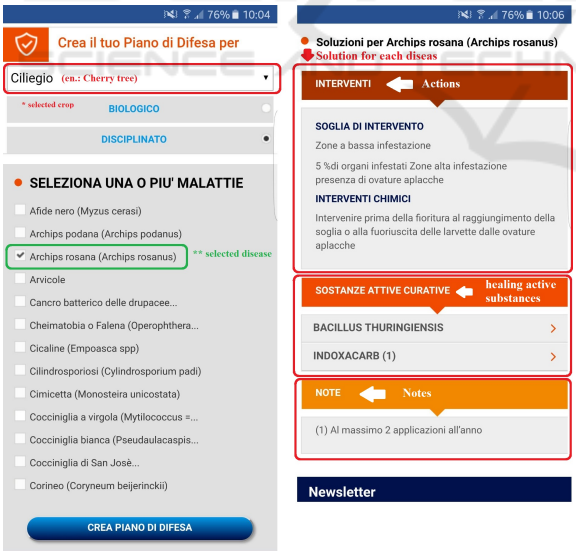

Figure 5: Mobile app screenshots.

To solve the limitation, disadvantages and challenges described, in this paper we present an infrastructures for distributed CEP that will be partially executed on mobile devices by introducing server side a Reflective IoT Middleware. The main advantages of the system are: *(i)* using CEP engine at system run-time enables event-driven monitoring and update notifications, and *(ii)* modeling system with human-machine understandable ontologies ensures easier reconfiguration of the system. To validate the model we propose an instantiation in a real scenario also designing an OWL 2 Ontology that encodes knowledge about aspects related to Integrated Pest Management practice. The solution identified aims at becoming a useful instrument by ecosystem actors for raising awareness about the use of chemical treatments.

In this stage of the work, we performed a first set of experiments to validate the approach providing the tool to a Company and testing single components. We are current working to extend the system for the creation of an Advanced Cyber-Physical System.

ACKNOWLEDGEMENTS

The authors acknowledge support of Simone Salerno for ontology building. Francesco Nocera acknowledges support of Exprivia S.p.A Ph.D grant 2016.

REFERENCES

Ahsan, M., Motla, Y. H., and Asim, M. (2014). Knowledge modeling fore-agriculture using ontology. In *Open Source Systems and Technologies (ICOSST), 2014 International Conference on*, pages 112–122. IEEE.

Alavi, M., Leidner, D. E., et al. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, 25(1):107–136.

Barshe, P. and Chitre, P. (2012). Agricuture system based on ontologyagrosearch. *Int. J. Emerg. Technol. Adv. Eng*, 2(8).

Bo, Y. and Wang, H. (2011). The application of cloud computing and the internet of things in agriculture and forestry. In *Service Sciences (IJCSS), 2011 International Joint Conference on*, pages 168–172. IEEE.

Buschmann, F., Henney, K., and Schmidt, D. C. (2007). *Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing*. Wiley.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1996). *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc., New York, NY, USA.

Chaudhary, S., Bhise, M., Banerjee, A., Goyal, A., and Moradiya, C. (2015). Agro advisory system for cotton crop. In *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–6. IEEE.

Di Noia, T., Marina, M., Francesco, N., and Eugenio, D. S. (2016). Ontology-based reflective iot middleware-enabled agriculture decision support system.

Fersi, G. (2015). Middleware for internet of things: A study. In *Distributed Computing in Sensor Systems (DCOSS), 2015 International Conference on*, pages 230–235. IEEE.

Flint, M. L. (2012). *IPM in practice: principles and methods of integrated pest management*, volume 3418. UCANR Publications.

Flint, M. L. and Van den Bosch, R. (2012). *Introduction to integrated pest management*. Springer Science & Business Media.

Gruber, T. R. et al. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

Guarino, N. (1998). Formal ontology and information systems. 98(1998):81–97.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). Introduction to automata theory, languages, and computation. *ACM SIGACT News*, 32(1):60–65.

Horne, R., Khatiwada, S., and Kuhn, S. (2016). World employment and social outlook: trends 2016.

Hu, L., Xie, N., Kuang, Z., and Zhao, K. (2012). Review of cyber-physical system architecture. In *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on*, pages 25–30.

Jasper, R., Uschold, M., et al. (1999). A framework for understanding and classifying ontology applications. In *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW*, volume 99, pages 16–21.

Kang, J. C. and Gao, J. L. (2013). Application of ontology technology in agricultural information retrieval. In *Advanced Materials Research*, volume 756, pages 1249–1253. Trans Tech Publ.

Luckham, D. (2002). *The power of events*, volume 204. Addison-Wesley Reading.

Malik, N., Sharan, A., and Hijam, D. (2015). Ontology development for agriculture domain. In *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, pages 738–742. IEEE.

Mongiello, M., di Noia, T., Nocera, F., di Sciascio, E., and Parchitelli, A. (2016). Context-aware design of reflective middleware in the internet of everything. In *Federation of International Conferences on Software Technologies: Applications and Foundations*, pages 423–435. Springer.

Pereira, D. H. G., Dantas, C. F. F., and Ribeiro, C. M. (2012). A pragmatic approach for sustainable development based on semantic web services. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, pages 82–90. ACM.

Razzaque, M., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2015). Middleware for internet of things: a survey. *Internet of Things Journal, IEEE*, PP(99).

Stevens, R., Wroe, C., Lord, P., and Goble, C. (2004). Ontologies in bioinformatics. In *Handbook on ontologies*, pages 635–657. Springer.

Walisadeera, A. I., Wikramanayake, G. N., and Ginige, A. (2013). Designing a farmer centred ontology for social life network. In *DATA*, pages 238–247.

Xie, N., Wang, W., and Yang, Y. (2007). Ontology-based agricultural knowledge acquisition and application. In *International Conference on Computer and Computing Technologies in Agriculture*, pages 349–357. Springer.