Combining proactive and reactive approaches in smart services for the Web of Things

Nawel Sekkal[1], Sidi Mohamed Benslimane[1], Michael Mrissa[2], Boudjemaa Boudaa[1]

[1] LabRi Laboratory, Ecole Superieure en Informatique,
BP 73, El Wiam City, Sidi Bel Abbes, Algeria
n_iles@mail.univ-tlemcen.dz,s.benslimane@esi-sba.dz,
boudjemaa.boudaa@univ-tiaret.dz
[2] LIUPPA, University Pau & Pays Adour, BP 1155, 64013 Pau Cedex, France
michael.mrissa@univ-pau.fr

Abstract. The Web of Things (WoT), facilitates the interconnection of different types of real-world objects, integrating them into the virtual world and ensuring their interoperability through Web services. However, it remains a challenge to automate the tasks connected objects need to deal with. In this paper, we focus on the development of smart web services that automate service tasks, autonomously adapt to context changes in the object's environment, and to users' preferences. In this article, we propose a software framework for smart services that relies on a reactive and proactive approach to deal with context and its temporal aspects. Smart web services developed according to these principles can react to current situations and proactively anticipate an unforeseen situation in order to take the right decision.

1 Introduction

In recent years, the development of the Web has been characterized by the increased use of connected devices, such as sensors, actuators and mobiles devices. These devices enable a direct integration of an information source depending on the client application context. As such, the WoT (Web of Things) interconnects any sort of physical object deployed in the real world (surveillance devices, medical sensors, temperature sensors, intelligent sensors, etc.), to the Web in order to promote global interoperability.

In the perspective of contributing in the emergence and development of the WoT, existing Web standards are used such as: a) Uniform Resource Identifier (URI), b) Representational State Transfer (REST), c) Hypertext Transfer Protocol (HTTP) [1]. Furthermore, the usage of Web Semantics, such as Resource Description Framework (RDF) [2] and Linked Data (LD) [3] is necessary in order to: a) provide a semantic value to the data generated by heterogeneous sources, b) enable contextual interpretation, c) ease the management of the growing volume of data and d) promote interoperability at the semantic level.

In order to benefit from the vision of WoT, we adopt the notion of "Smart Services" developed in [4,5,6], that not only permit remote access to resources and their embedded functions, but offer intelligent services as well as adaptation to the application context. As a result, it provides the user with means to carry out its tasks automatically and in an autonomous manner. For instance: patient diagnostic, itinerary planning, adjusting home devices settings (thermostat, air-conditioning, radiator, etc.).

In this paper, we take interest in smart Web services and their diverse functionalities (acquiring, reasoning, adaptation rules depending on context changes). The latter enables the target system to make an autonomous decision without user intervention. Our approach to context adaptation relies on context and time Web. The temporal aspect is expressed with a hybrid proactive-reactive method [27]. The proactive part is interesting in the sense that we can predict the behavior of the system in the future and act before the situation happens. The reactive part is useful when the system needs to make a real-time decision. The rest of this paper is set out as follows: In Section 2, we describe the problem statement and introduce a motivating scenario in the smart home domain. Section 3 contains background information about WoT, context and smart Web services. The related works are presented in Section 4. In Section 5, we presented Framework for context aware proactive-reactive service. Finally, Section 6 outlines the conclusions and future work.

2 Problem statement and Motivating scenario

The problem we address in this paper is the following:

- How can smarts web services make decisions autonomously about the behavior of the system, or the temporal aspect is important in its actions?
- How can the smart web services adapt to the current situation of the user and to the change of the context of the different objects of the system of objects with a dynamic link?

Currently, there is a lack of work for the smart Web service to incorporate reactive and proactive behavior. The smart service must make decisions in advance, taking into account the events that will occur in advance and depending on the context. It must also adapt autonomously to changes in the context. Our research is motivated by these challenges to provide high-level information such as: detecting situations in the environment [10]; making predictions of situations in the future and reacting in advance, and determining the policy of actions to consume the appropriate services to adapt the environment in advance of the envisaged situation.

In rest of this section, we present a scenario in the field of smart home in order to illustrate the challenge and highlight our contribution. Our scenario involves a family that waters their plants on per-need basis and depending on water requirements. However, with the spring holydays (rain probability and invariable weather), they must leave their home vacant. The question is, how can they insure the daily watering tasks for their plants when they leave on holidays?

To solve this problem, the family installs an intelligent watering system that will monitor the garden remotely through a smart phone. This system enables surveillance in an autonomous way of the pertinent context in real-time (lighting, temperature, humidity, etc.) and decides when and how to act with an appropriate water dose

depending on the plant type. Furthermore, a wireless sensor network is placed to collect and transmit the aforementioned parameters. This application takes into account weather conditions and forecast. To perform this task, a GPS API is used to determine the location of the family home. We consider that the software system can control the existing watering system. Consequently, the watering task is triggered automatically by sending notification messages to the user smart phone and react in anomaly cases.
To provide a smart watering service, the following questions arise:
- How will the watering system take into consideration the different types of plants and their preferences in dosage and frequency?
- When and how will the watering system be triggered, taking into account the plant environment and the weather forecast?
- How to adapt the watering system in case of unforeseen situation (unexpected adversaries, other water source, etc.)?

An implementation of the scenario is necessary to validate our approach.

3 Background

Smart services are the result of the combination of multiple technologies: The Web of objects, semantic Web, the context -ware systems, Web services and linked data.

3.1 The Web of Things

An intelligent object is any object capable of processing information and act on its environment in an autonomous way. One of its principle characteristics is its capacity to communicate with other objects. It can be represented in a software / hardware form having autonomous capacity, to communicate and cooperate in addition to reasoning features. Moreover, each object [26]:

- Is identifiable in a unique manner, (barcode, RFID, IP address, etc.);
- A processing capacity in order to control and manage the object;
- Is able to store data and its priorities;
- Is equipped with a network interface card to interact with other objects. [26,8]

The objects use communication protocols such as 6LowPan, Bluetooth, Low Energy, ZigBee, etc. [14, 15].

In the course of the recent years, the Internet of Things has evolved at an accelerating speed, connecting numerous heterogeneous objects (sensors, actuators, smart phones, applications, home devices, etc.) over the Internet. Due to multiple communication protocols, it was difficult to connect objects to each other's. Therefore, on top of the Internet, the Web of things appeared to support the application that operates over this infrastructure, linked with the technologies and relying of the Web, enabling the users to interact with remote services (simple and complex). A physical object is seen as a set of services accessible over the Web and for which the environment defines its context. The principle advantage of the WoT is the usage of the Web norms / protocols like Hypertext Transfer Protocol (HTTP) and URIs to promote interoperability [7].

An architecture founded on resources, was proposed in [8] in the field of WoT (Fig. 1). It enables the integration of objects through services. This proposition is elaborated in the form of a layered architecture, structured around five layers:

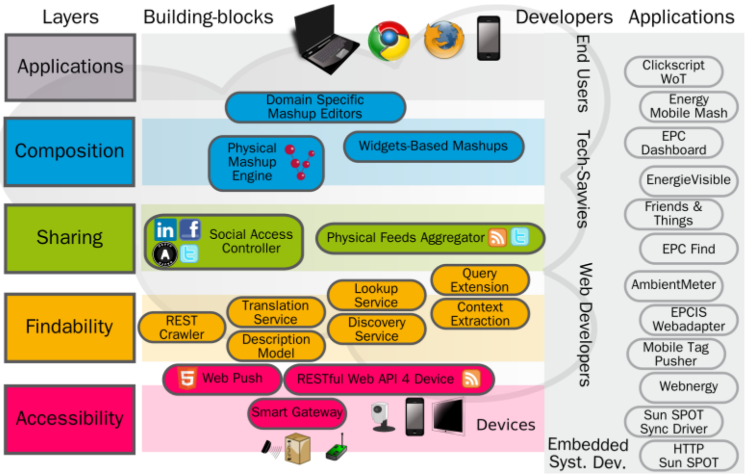

Fig1: Architecture of the WoT [8]

Accessibility layer: deals with the integration of the object in the Web;
Findability layer: enables searching and locating the pertinent services in the WoT;
Sharing layer: Manage the access to objects (social networks);
Composition layer: enables the composition of services and introduces the notion of physical mashup;
Application layer;

For the conception of the WoT, we apply the same tools, techniques, models and languages used in Web applications. On the other hand, the applications concern objects with limited resources, thus other techniques must be adopted such as: AJAX (Asynchronous JavaScript and XML), mashups techniques (physical-environment and virtual- environment), Event Driven approach [8]. The development of WoT requires the extension of the existing Web so that the real world objects can be integrated in it, either in a direct or indirect manner. Consequently, the usage of Web services is necessary in order to exploit the data and functionalities of the physical objects.

Recently, the development of Web services and their APIs turned away from traditional services that use SOAP and WSDL. Instead, the REST architectural style has been recognized as a good engineering practice. In the WoT, it is more interesting to use REST which is conceived in a resource oriented architectural model in addition to the HTTP protocol for a better representation of resources and coordinated Client / Server communication. RESTful services are based on representative state transfer [9], an architectural model that considers that each physical object is addressable /

identifiable through an URI. They are defined by the following four concepts: a) identification of a resource with a URI, b) the definition of a uniform interface using HTTP standard verbs (GET, POST, PUT and DELETE), c) the usage of hypertext links, d) communication through HTTP using formats such as JSON, YAML or XML. A comparative table between Ws-* and REST services is presented in [9].

3.2 Defining Context

The term context in "smart services" can refer to any kind of piece of information about an object, its placement, its modifications and also to the identity of users. Additionally, it can include computing environments, physical environment and even user's environments (social situation). The context is defined in [17] depending on three aspects: a) usage context, b) physical context, c) information context. A subsequent study extended this definition by adding the time context and social context [3]. Hence, a physical object is characterized by:

Physical context: describes the state of the object that contains the acquired information through sensors placed in the device or its running software components. They can be dynamic (such as placement, activity, relation with others, etc) or static (personal information, preferences, etc).

Usage context: represents the information that concerns the user, they can be dynamic or static. Dynamic information usually changes over time and requires periodic monitoring. On the other hand, static information is rarely modified such as the user habit, identity and personal preferences.

Environmental context: are all information related to the environment of the user such as the ambient lighting, temperature, humidity, etc.

Social context: is external information that concerns the preferences of the users, their public profiles and their behavior in friend group. The set of information enables the system to recognize the user in its social context.

There are context modelling techniques, such as the techniques of central context management based on the P2P model, as well as others using XML based ontology (i.g. in Context project). In the field of WoT, the Web services can become more intelligent through the use of context. Meaning that the service is aware of the events, the physical situations of the target objects or application. Thus, it can respond proactively and intelligently. Furthermore, they can share their contexts with other services in order to respond to users' requests.

3.3 Smart Web Services

The set of technologies associated to Web service enables remote software applications to communicate through the Web. This technology is based on the principles of service oriented computing and uses the languages of the Web for data description. Its development has progressed with the emergence of semantic Web services which represent a promising path in order to automate Web services tasks (search, invocation, composition). Another approach is to extend API semantics with the principles of "linked data" [11]. This technology is useful to provide access for the resource to its set of related data, to integrate and use them through RDF links which makes the Web an interconnected space. Currently, another interest is taken towards social networks by the use of AJAX technologies [13].

All these approaches aim to introduce new functionalities to Web services and make them smart and autonomous. With the technological advances in sensors and their growing availability in intelligent objects, context is becoming an essential source of information from which we can understand the different situations and adapt specific services. These services are equipped with intelligent capacities; they are called "smart services". They are available today on smart phones, highways, smart televisions, smart cities and health care devices, with the objective to provide customized functionalities without the need of direct users' intervention. In addition to the context, we recommend a proactive approach that allows a smart web service to meet the future needs of the user. We propose in this approach to proactively recommend a service that can respond to users or devices invoked.

A smart service is defined in [5] as a software service that helps the users in their daily lives activities, with an important productivity, an improved quality of services and an efficient communication between the users and objects. In [6], a smart service is defined as a general model that enables the integration of any functions of an ontology of a field where one or multiple Web services are exploited automatically in order to satisfy the specific need of an object. In [16] a smart Web service "SmartWS" is defined as a Web API that conform to standards (HTTP, URI), that uses / produces semantic data (RDF) and encapsulates a logical decision in an autonomous fashion. By comparison with the traditional Web service, a smart Web service is characterized by: a) autonomy, b) automatic behavior, c) intelligence, d) communication capacity, e) customization, f) surveillance and control capacities of objects.

4 Related Work

Numerous works have been conducted in the field of WoT. In [13] the works in this field are classified into five categories: pre-processing and storage of data, data analysis, services management, security and confidentiality, network and communication. A significant amount of work has been done for WoT modelling in order to represent real world objects in the Web such as: TinyRest, WebPlug, AutoWeb, SpitFire, SOCRADES, Avatar, etc [5]. Generally, these models are founded on LoWPAN and CoAP for the integration of objects in the Web as well as the REST / WS-* services in order to insure the interaction between intelligent objects. Nevertheless, very little work has been made on WoT smart services modelling. In [16,7] two intelligent applications are shown: "Energy Visualizer" for energy consumption surveillance and control of household devices, and "Ambient Meter" in order to obtain energy consumption of machines. A model of smart services (SSF) based on context adaptation is presented in [5] with a central system scenario. The authors base their work on [17] to define context. A centralized architecture is proposed and the processing is based on prerequisite knowledge that enables to adapt the offered services to better respond to user requirements. The work in question represents a first step towards intelligent services. Unfortunately, it is concentrated only on the notion of context adaptation and does not address heterogeneous data without explaining how sensors / actuators are registered in the platform or how they communicate.

In [6] an example of intelligent service for plant watering based on context is presented. In order to control the dosage and frequency of water spray, the service must detect the dryness of the object, the knowledge on the field of plants, plant watering "preferences", and use the weather prediction service. An intelligent interface was developed based on Thing-REST model. Subsequently, an extension for the aforementioned work by replacing Web Service RESTful with CoAP is presented in [9], introducing the semantic aspect in the event detection service. A new tendency to develop Web services based on micro-services also recently emerged [17].

While the aforementioned contributions were based on the notion of context, others were proposed that are focused on semantic Web. For instance, a personalized meteorological semantic service as well as an improvement of functionalities of a service for disaster management is presented in [18]. In [19], the authors describe an application in the field of social networks that is oriented towards integrating objects in the daily lives of people. Both applications use the WoT technology as well as the semantic, smart services. Nevertheless, there was no extension of the intelligence notion in these works. The model of smart Web service "SmartWS" in a form of a Web API introduced in [16] enables the automatic adjustment of settings and consequently, adapts to the context where the event is triggered. It is based on semantic technologies in order to describe input and output as well as linked data in order to provide access to all related resources. In [21] a testing model was proposed to evaluate and validate intelligent applications as well as QoS depending on user requirements, with the help of the Quality Model (QM) of the International Organization for Standardization (ISO). In [17, 20] a new avatar model in the Web of objects discovers the capacities of objects and exposes them as functionalities based on semantic techniques as part of the ASAWoO project [17]. In the same project, a meta-model of context [22,23] capable of responding to different adaptation questions for a given WoT application is proposed.

Table 1 is a summary of the works cited in the state of the art section. The different studies aim to introduce context of objects with a semantic description in Web services in order to make them "Smart". In the pursuit of smart service development, the discussed works are based on context adaptation with semantic description of data while taking decisions in an autonomous way and in real time (reactive approach).

Our proposition is founded on the generic model of smart Web service. It is not based only on context, but we have added temporal aspects as well in order to predict the future behavior of the system. This approach is original because it combines reactive and proactive decision-making, seeking to adapt automatically to the environment according to the time-evolving situations.

Table 1. Summary of the state of the art on smart Web services

Approach	*Context*	*Semantic*	*Architecture*	*Adapt*	*Protocol*	*Data format*	*Linked data*	*Protocol network*	*Application*
Guinard [20]	X	X	Web Architecture (SUN SPOT)	X	Rest	JSON	X	-	Ambient Meter , Energy Visualizer
JING HE [6]	✓	✓	Resource Oriented ROA	✓	Thing-rest	XML	X	HTTPs	Plant watering System
Ramaswamy [9]	✓	✓	Multi-Layer	✓	Rest	XML	X	6LowPAN/COAP	Plant watering System
Lee (SSF) [5]	✓	X	Multi-agent	✓	Reasoning Algorithm (HMM, KNN)	XML	X		Smart Home
Lee et al. WISE [18]	X	✓	M2M/Cloud	X	-	RDF	X	-	Meteorological Service
Beltran et al [19]	X	✓	Multi-Layer	X	REST	JSON/ LD	X	-	Social Field
Maleshkova (SmartWS [4]	✓	✓	Multi-Layer	✓	REST	XML/ JSON	✓	HTTPs	Medical Diagnostic
Zeiner [17]	X	X	Multi-service	✓	REST	JSON/ XML	X	HTTPs	/
Gyrad et al. [25]	X	✓	M3	X	-	RDF	✓		Mobile Application
Mrissa et al. [17,18]	X	✓	Multi-agent	X	REST	XML	X		Temperature regulator
Terdjimi [24]	✓	X	Multi agent	X	/	Contextual Data	X	-	Agriculture Field

5 Proposed architecture

Fig.2 gives an overview of a multi-layered architecture that represents the development of our smart web service that is context-based and proactive-reactive.

It provides the following functionalities:

- Data acquisition in order to define the context of objects and user;
- Identifying the current situation of the user in order to understand the user requested services;
- Selecting the appropriate service and adapt it in a proactive and reactive way in order to obtain the appropriate decision;
- Adapting the service in an autonomous way depending on the changes of the context;

The framework involves different types of objects that must collaborate. Each object is considered as a resource. Based on the REST architectural style, any resource represented with a sensor, is addressed by a unique identifier of standard format via Uniform Resource Locators (URL) using the Hypertext Transfer Protocol (HTTP) and its methods (e.g. GET, POST, PUT and DELETE) to access them.

Furthermore, we use the ontologies and domain knowledge that are needed to unify and understand the context of things and their preferences (e.g., dose, frequency, or properties) and user preferences. We think that our proactive-reactive approach can offer an added value to Web service mechanisms that adapt to context and usage preferences, in order to make it a smart Web service. For the development of our smart web service, we present a generic model in terms of architectural view made of multiple layer types.

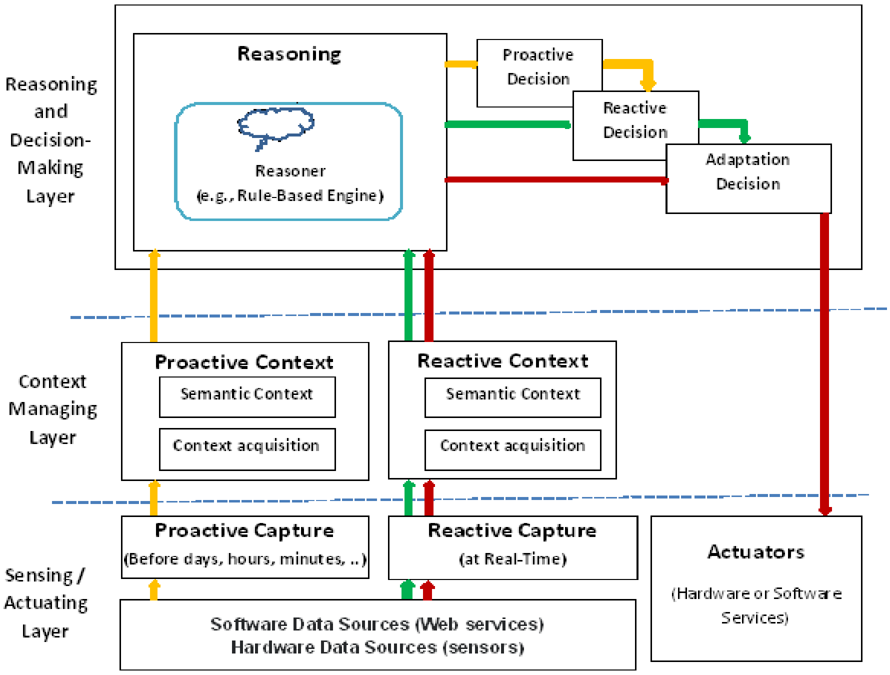

Fig. 2. Framework for context aware proactive-reactive service

"Sensing actuating layer": includes all data sources: hardware data sources, such devices, wearables, sensors, algorithms, and software components, such as web services. The capture of the data can be either reactive, i.e. real-time, or otherwise predict the data as it will be in the future (e.g., weather forecasts), this may help us make the right decision and avoid unwanted events.

"Context managing layer": enables to comprehend the situation and define the corresponding service. It is composed by the following sub components: a Proactive context and a Reactive context that are necessary for the smart web service in order to make the decision autonomously and adapts dynamically.

For the acquisition context, each resource is identifiable with a unique URI and HTTP, we use the REST model and its methods (GET, PUT, DELETE, POST) to gather the state of resources represented by a sensor. The semantic context enables to unify the context of the objects using the context ontology in order to facilitate interfacing between Web services.

"Reasoning and decision-making layer": depending on the acquired context, this component deduces the corresponding situation based on the reasoning algorithms that are of three types: Sequential reasoning algorithms, relational algorithms, decision tree. The reactive decision takes the decision to execute a service depending on the situation in real-time, according to the context reactive and the proactive decision is based on the proactive context to recommend a course of action and the appropriate service in the near future. This mechanism can be implemented based on Bayesian Networks to estimate the probability of a situation to expire in the future. Particularly, the final decision taken by our approach is founded on the decisions of the reactive and proactive mechanisms, since the temporal aspect is crucial to respond to the intelligent system requirement. The decision adaptation adapts and customizes the associated service depending on the changes of context. For example, following unexpected changes or services errors. As a result, our system must adapt and reconsider the decision. We can use a loop such as MAPE-K that contains the state indicators: monitoring, analysis, planning, and execution. It is necessary to implement our scenario to validate this approach. Fig. 3 shows a sequence diagram for the proposed proactive-reactive framework:

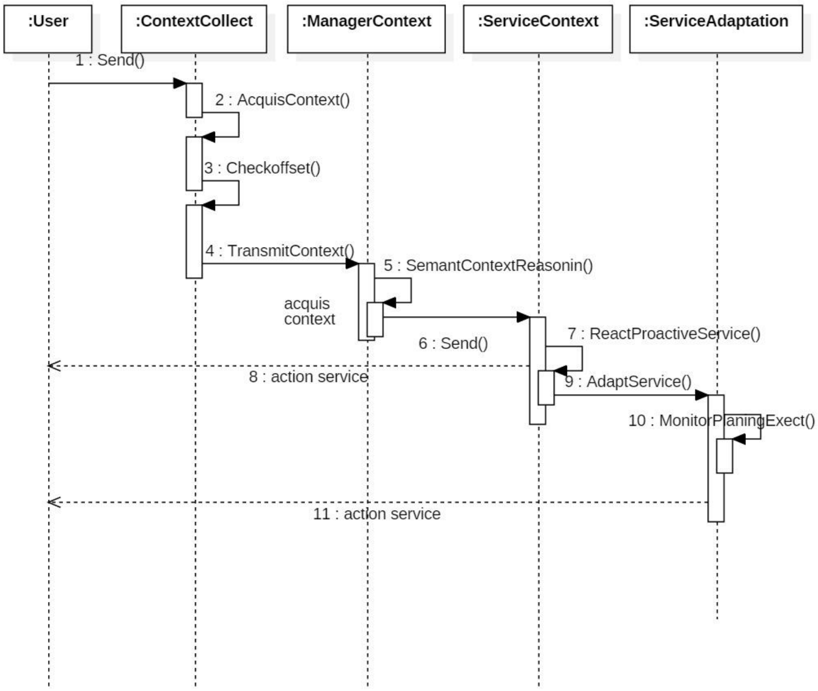

Fig. 3. Sequence diagram for the proactive-reactive framework.

6 Conclusion

A major challenge of the WoT is to allow applications to adapt to their environment. In this article, we discuss the notion of smart web services that plays an important role in our society. We have established a state of the art in this field, and we have proposed a framework of smart web service solution for domain of web of things, based a proactive and reactive approach. The proactive part is interesting in the sense that we can predict the behavior of the system in the future and act before the situation happens. The reactive part is useful when the system needs to make a real-time decision. This approach is supported by semantic technology and the concept of the context, to enable the smart web service to make the most effective decision in a way. autonomous and adapt it according to the change of context.

Possible directions for future research include:

i) to propose a context meta-model in Wot that allows transforming raw context data, into semantically annotated contextual information, and then solving easily adaptation requests. The model can be represented based on UML diagram of formal declarations,

ii) an approach for handling reactive and proactive behaviors using different techniques: dynamic Bayesian network, Markov chain, fuzzy logic... ,

iii) Improve the proposed architecture in relation to the proposed model,

iv) Extend the use case of watering by introducing a scenario that supports proactive action (drought, abundance of rain, failure of irrigation equipment, etc.), and testing with the architecture proposed over a real world automated environment.

References

1. Klyne, G., Carroll, J. J. Resource Description Framework (RDF): Concepts and Abstract Syntax." W3C Recommendation (2004)
2. Heath, T., Bizer, C. Linked data: Evolving the web into a global data space. Synthesis lectures on the antic web: theory and technology 1, no. 1 (2011)
3. Garriga, M., Mateos, C., Flores, A., Cechich, A., Zunino, A. RESTful service composition at a glance: A survey. Journal of Network and Computer Applications, elseiver (2015)
4. Maleshkova, M., Philipp, P., Sure-Vetter, Y., Studer, R. Smart Web Services (SmartWS) – The Future of Services on the Web. IPSI BgD Transactions on Advanced Research (2016) 15-26
5. Lee, J. Y., Kim, M. K., La, H. J., Kim, S. D., A software framework for enabling smart Services. SOCA (2012)
6. He, J., Zhang, Y., Huang, G. A Smart Web Service Based on the Context of Things, Journal ACM Transactions on Internet Technology (TOIT) Vol. 11(2012)
7. Guinard, D., Trifa, V., Mattern, F., Wilde, E. From the Internet of Things to the Web of Things: Resource oriented Architecture and Best Practices. In Architecting the Internet of Things," Springer, Berlin Heidelberg (2011)
8. Guinard, D. A Web of Things Application Architecture - Integrating the Real-World into the Web. these doctorat (2011)
9. Ramaswamy, P., Vinob, C., Elias, S.Manoj, S.S. A REST Based Design for Web of Things In Smart Environments. 2nd IEEE International Conference on Parallel, Distributed and Grid Computing (2012)

10. Khalfi, M.F., Benslimane, S.M.: Proactive Approach for Service Discovery Using Web Service for Devices On Pervasive Computing. ICCASA '14 Proceedings. Dubai, United Arab Emirates. (2014)
11. Jamont, J.P. Démarche, modèles et outils multi-agents pour l'ingénierie des collectifs cyber-physiques. (2016)
12. Terdjimi, M., Médini, L., Mrissa, M., Le Sommer, N. An Avatar-based Adaptation Workflow for the Web of Things. 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (2016)
13. Liu, W., Li, X., Huang, D. A Survey on Context Awareness. Proceeding of the International Conference on Computer Science and Service System (CSSS 2011), IEEE (2011) 144-147
14. Mathew, S.S., Atif, Y., Sheng, Q., Maamar, Z., The Web of Things—challenges and enabling technologies. Internet of things and inter-cooperative computational technologies for collective intelligence, vol 460. Springer, Berlin (2013) 1–23
15. Zeng, D., Guo, S., Cheng, Z. The Web of Things: A Survey. journal of communications col 6. N° 6 (2011)
16. Guinard, D., Trifa, V., Wilde, E. A Resource Oriented Architecture for the Web of Things. Proceedings of the Internet of Things Conference (2010)
17. Schilit, B.N., Theimer, M. Disseminating active map information to mobile hosts. IEEE Network, Vol. 8 (2002) 22 -32
18. Lee, J.W., Ki, Y. W., Kwon, S. WISE: "An Applying of Semantic IoT Platform for Weather information service Engine. the International Semantic Web Conference (2014)
19. Beltran, V., Ortiz, A.M., Hussein, D. A semantic service creation platform for Social IoT. the meeting of the WF,IoT (2014)
20. Mashal, I., Alsaryrah, O., Chung, T., Yang, C., Kuo, W., Agrawal, D. Choices for interaction with things on Internet and underlying issues. journal elseveir (2014)
21. Zeiner, H., Goller, M., Juan, V., Salmhofer, F., Haas, W. Secos:Web of things platform based on a microservices architecture and support of time-awareness. e&I (2016) 158–162
22. Mrissa, M., Médini, L., Jamont, J. Semantic discovery and invocation of functionalities for the web of things. IEEE WETICE (2014) 281–286
23. Mrissa, M., Médini, L., Jamont, J., Sommer, N.L., Laplace, J. An avatar architecture for the web of things. IEEE Internet Computing, (2015) 30–38
24. Terdjimi, M., Médini, L., Mrissa, M., Le Sommer, N. An Avatar-based Adaptation Workflow for the Web of Things. 25th IEEE International Conference on Enabling Technologies (2016)
25. Gyrard, A. et Al. Assisting IoT Projects and Developers in Designing Interoperable Semantic Web of Things Applications. Data Science and Data Intensive Systems (DSDIS), IEEE International Conference. (2016)
26. Mattern, F., Floerkemeier, C. From the Internet of Computers to the Internet of Things. From Active Data Management to Event-Based Systems and More. Springer (2010) 242-259
27. Krupitzera, C., Rotha, F.M., VanSyckela, S., Schieleb, G., Beckera, C. A Survey on Engineering Approaches for Self-Adaptive Systems. Pervasive and Mobile Computing. Elseveir. (2015)