
Proactive and reactive context reasoning architecture for smart web services

Nawel Sekkal* and
Sidi Mohamed Benslimane

LabRI-SBA Lab,
Ecole Supérieure en Informatique,
Sidi Bel Abbès, Algeria
Email: n.sekkal@esi-sba.dz
Email: s.benslimane@esi-sba.dz
*Corresponding author

Michael Mrissa

InnoRenew CoE,
Livade 6, 6310 Izola, Slovenia
and
Faculty of Mathematics,
Natural Sciences and Information Technology,
University of Primorska
Glagoljaška ulica 8, 6000 Koper, Slovenia
Email: michael.mrissa@innorennew.eu

Cheol Young Park

The C4I and Cyber Center,
George Mason University,
MS 4B5, Fairfax, VA 22030-4444, USA
Email: cparkf@masonlive.gmu.edu

Boudjemaa Boudaa

Département d’Informatique,
Université Ibn Khaldoun,
Tiaret, Algeria
Email: boudjemaa.boudaa@univ-tiaret.dz

Abstract: The web of things (WoT) uses web technologies to connect embedded objects to each other and to deliver services to stakeholders. The context of these interactions (situation) is a key source of information which can be sometimes uncertain. In this paper, we focus on the development of intelligent web services. The main requirements for intelligent service are to deal with context diversity, semantic context representation and the capacity to reason with uncertain information. From this perspective, we propose a framework for intelligent services to deal with various contexts, to reactively respond to real-time situations and proactively predict future situations. For the

semantic representation of context, we use PR-OWL, a probabilistic ontology based on multi-entity Bayesian networks. PR-OWL is flexible enough to represent complex and uncertain contexts. We validate our framework with an intelligent plant watering use case to show its reasoning capabilities.

Keywords: smart web service; the web of things; context reasoning; proactive; reactive; multi-entity Bayesian networks; MEBNs; PR-OWL.

Reference to this paper should be made as follows: Sekkal, N., Benslimane, S.M., Mrissa, M., Park, C.Y. and Boudaa, B. (xxxx) 'Proactive and reactive context reasoning architecture for smart web services', *Int. J. Data Mining, Modelling and Management*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Nawel Sekkal received her Magister in Computer Science from the University Abou Bekr Belkaid, in 2005. Since 2006, she is a Lecturer in Computer Science from the University Abou Bekr Belkaid. Currently, she is a member in the research team 'Service Oriented Computing' at the LabRI-SBA Laboratory (ESI-Sidi Bel-Abbès).

Sidi Mohamed Benslimane is Full Professor from the Higher School of Computer Science, Sidi Bel-Abbès, Algeria. He received his PhD in Computer Science from the Sidi Bel Abbes University, in 2007. He also received his MS and Technical Engineer in Computer Science from the Computer Science Department at the Sidi Bel Abbes University, Algeria, in 2001 and 1994, respectively. He is currently the Head of the Higher School of Computer Science, Sidi Bel-Abbès, Algeria. From 2001 to 2015, he was a member of the Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS. He currently heads the research team 'Service Oriented Computing' at the LabRI-SBA Laboratory. His research interests include semantic web, service-oriented computing, ontology engineering, information and knowledge management, distributed and heterogeneous information systems and context-aware computing.

Michael Mrissa is a researcher from the InnoRenew CoE and Full Professor from the University of Primorska, Slovenia. He received his PhD in 2007 and his accreditation to supervise research (HDR), in 2014 from the University Claude Bernard Lyon 1, France. His main research interests are related to service-oriented computing, semantic reasoning and web of things. He has published over 70 peer-reviewed papers in conferences and journals.

Cheol Young Park obtained his MS in Systems Engineering and PhD in Information Technology from the George Mason University (GMU). During his time at GMU, he pursued a research agenda focusing on machine learning for multi-entity Bayesian networks (MEBNs), inference algorithm for hybrid Bayesian networks, and development of predictive situation awareness system. He has researched and developed several machine learning algorithms for MEBN to support predictive situation awareness systems [e.g., a MSAW system (for smart manufacturing), a PROGNOS system (for maritime situation awareness), and a HERALD system (for critical infrastructure defence)]. He had worked as a Research Associate for the C4I and Cyber Center at GMU.

Boudjemaa Boudaa is an Assistant Professor from the Ibn Khaldoun University, Department of Computer Science, Tiaret, Algeria. He received his PhD in Computer Science from the Tlemcen University, in 2017. He also received his Magister and Engineer degrees in Computer Science from the Tiaret and Oran universities in 2009 and in 1995, respectively. He is currently a member in the research team 'Service Oriented Computing' at the LabRI-SBA Laboratory (ESI-Sidi Bel-Abbès). His interests include context-aware computing, recommender systems and software engineering.

This paper is a revised and expanded version of a paper entitled ‘Combining proactive and reactive approaches in smart services for the web of things’ presented at IFIP International Conference on Computational Intelligence and its Applications (IFIP CIIA 2018), Oran, Algeria, 8–10 May 2018.

1 Introduction

The web of things (WoT) interconnects various sort of physical objects deployed in the real-world (e.g., sensors, actuators, devices, and networks) to the web to promote global interoperability. These devices enable direct acquisition of context information, and consequently, it is much feasible to understand the current situations of the real-world. In the perspective of contributing in the emergence and development of the WoT, we adopt the notion of ‘smart services’ (SSF) developed in Lee et al. (2012), He et al. (2012), and Maleshkova et al. (2016), that not only enables remote access to resources and their embedded functions, but offers intelligent services as well as adaptation to the application context. As a result, it provides users with the means to carry out their tasks automatically and autonomously (e.g., patient diagnostic, itinerary planning, and adjusting home devices settings).

In this paper, we take interest in smart (or intelligent) web services and their diverse functionalities (e.g., acquiring, reasoning, and adaptation depending on context changes). Adaptation, one key requirement of smart web services, enables autonomous decision-making on the target system without user intervention. Our approach to context adaptation is based on contextual and temporal (or dynamic) aspects. The temporal aspect can be expressed with a hybrid proactive-reactive method (Krupitzer et al., 2015). The reactive part of the method can be useful when a smart system needs to make a real-time decision. On the other hand, the proactive part is useful in the sense that we can predict the behaviour of the system in the future and act before critical situations happen. With the technological advances of sensors and the growing availability of intelligent objects, context is becoming an essential source of information from which we can understand different situations and adapt specific services. Services that adapt to their context are called ‘SSF’. They are available today on smartphones, smart highways, smart televisions, smart cities, and smart healthcare devices, with the objective to provide customised functionalities without the need for direct user intervention.

In this paper, we introduce an approach that combines proactive and reactive reasoning into a single architecture to enable smart web services for the WoT. Our architecture, called smart WoT architecture (SWA) for proactive and reactive context reasoning, supports the operation of smart web services with five layers:

- 1 sensing/actuating layer
- 2 context acquisition layer
- 3 context modelling layer
- 4 reasoning/decision-making layer
- 5 adaptation layer to proactively and/or reactively respond to users requests and/or invoke control devices in smart web service systems.

Our work focuses especially on the reasoning/decision-making layer for semantically representation and reasoning about complex situations with uncertain information. To reach this aim, the system is based on multi-entity Bayesian network (MEBN) (Laskey, 2008) that combines first-order logic (FOL) with Bayesian networks (BNs) for representing and reasoning about uncertainty in complex, knowledge-rich domains. MEBN goes beyond common BNs by reasoning about an unknown number of entities interacting with each other in various forms of relationships (Park et al., 2016). Golestan et al. (2016) presented a comparative table on the different artificial intelligence methods in situation awareness (SAW) and recommended MEBN as having the most comprehensive coverage of features needed to represent complex situations. Furthermore, the reasoning methodology used in this research, relying on the PR-OWL ontology, provides a robust formalism to represent uncertainty while complying with the MEBN theory (Costa and Laskey, 2006).

The remainder of this paper is set out as follows: In Section 2, we describe the problem statement and introduce a motivating scenario using the smart home domain. Section 3 contains background information about WoT, Context-Aware System, MEBN/PR-OWL, and Smart Web Services. Related work on Smart Web Services is presented in Section 4. In Section 5, we present our architecture supporting proactive and reactive context-aware services. In Section 6, we illustrate our solution with a use case using MEBN. Finally, Section 7 concludes this paper with a reminder of our results and some insights for future work.

2 Problem statement and motivating scenario

In order to illustrate the research challenge and highlight our contribution, we present a smart home scenario that involves a family watering their plants on a per-need basis. However, on spring holidays (with rain probability), they must leave their home vacant. They need the plants to be taken care of without any human intervention.

To solve this problem, the family installs an intelligent watering system that will monitor the garden and can be accessed remotely through a smartphone. This system enables surveillance in an autonomous way using pertinent context in real-time (e.g., lighting, temperature, and humidity) and decides when and how to act with an appropriate water dose depending on the type of the plants. Furthermore, a wireless sensor network is installed to collect and transmit the aforementioned context parameters. This application considers weather conditions and forecast. To perform this task, a global positioning system (GPS) application programming interface (API) is used to determine the location of the family home. We consider that a software system can control the existing watering system. Consequently, the watering tasks are triggered automatically. The system sends notification messages to the user smartphone and reacts in case of an anomaly. To provide a smart watering service, the following questions arise:

- How can the watering system predict the moisture or drought of the soil of the plant in a region?
- Why will the watering system take into account the different types of plants and their state of health in his decision?

- When and how will the watering system be triggered, taking into account the plant environment and the weather forecast?
- How can to achieve the adaptation of smart web services according to the current situation of the user and to the change of the context of different system's objects?

As this scenario illustrates, the research problems we address in this paper be summarised as follows:

- How can smart web services take autonomous decisions about the behaviour of the system, and how to handle the temporal and uncertain aspects as parts of decisions?
- How can smart web services adapt to the current user situation and to the context changes of the different objects of the system?
- How to offer a semantic value to context data, generated by heterogeneous sources while managing their uncertainty?

Currently, there is a lack of work in how smart web services incorporate proactive and reactive behaviour to drive an autonomous system. A smart service must make decisions in advance, taking into account the events that could occur in the future using the current and past application context. It must adapt autonomously to frequent changes. Our research is motivated by these challenges to provide high-level information such as detecting situations in the environment, making predictions of situations in the future and acting accordingly, determining the course of actions to consume appropriate services, and adapting to the environment in advance for predicted situations. Our approach is presented in Sections 5 and 6, in dealing with these questions.

3 Background

Our SWA for proactive and reactive context reasoning is at the crossroad of multiple domains: WoT, context-aware systems, MEBNs, and smart web services. In this section, we introduce an overview of the different technologies, languages, and tools that are necessary for a good understanding of our contribution.

3.1 Web of things

An intelligent object is an object that is able to process information and acts on its environment in an autonomous way. One of its main characteristics is the capacity to communicate with other objects. It can be represented in a software/hardware form, having the autonomous capacity, to communicate and cooperate in addition to reasoning features. Moreover, each object (Mattern and Floerkemeier, 2010):

- is identifiable in a unique manner (barcode, RFID, IP address, etc.)
- has 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 (Mattern and Floerkemeier, 2010; Guinard et al., 2010).

Objects can use communication protocols such as 6LowPan, Bluetooth, Low Energy, ZigBee, etc. (Zeng et al., 2011; Mathew et al., 2013). In the WoT, intelligent objects and their services are fully integrated into the web by reusing and adapting technologies commonly used for traditional web content. Protocols like Hypertext Transfer Protocol (HTTP), URIs, and the REST architectural style (Guinard et al., 2011) can be used to promote interoperability. A general architecture of WoT presented by (Guinard et al., 2011) enables the integration of things with services on the web and facilitates the creation of web-based applications that operate on real-world things. This proposition is elaborated in the form of a layered architecture, structured around five layers:

- 1 The *accessibility layer* deals with the integration of the object in the web. This layer exposes things as RESTful things using resource-oriented architecture (Elias et al., 2012).
- 2 The *findability layer* enables searching and locating the pertinent services in the WoT.
- 3 The *sharing layer* manages the access to objects with the use of existing social networking platforms like Facebook, Twitter, and LinkedIn, as sharing hubs for things.
- 4 The *composition layer* enables the composition of services and introduces the notion of physical mash-up.
- 5 The last layer is the *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 Asynchronous JavaScript and XML (AJAX), mash-ups techniques (physical-environment and virtual-environment), and event driven approach (Guinard et al., 2011). The development of WoT requires extending the web so that real-world objects can be integrated into it, either in a direct or indirect manner. Consequently, the usage of web services has been a typical solution to exploit the data and functionalities of physical objects.

Recently, the development of web services and their APIs branched out from traditional services that use SOAP and WSDL. Instead, the REST architectural style has been recognised as a good engineering practice (Bouguettaya et al., 2013). In the WoT, it is more interesting to use REST which is conceived in a resource-oriented architectural model implemented with the HTTP protocol for a better representation of resources and coordinated client/server communication. RESTful services are based on representative state transfer (Elias et al., 2012), an architectural model that considers that each physical object is addressable/identifiable through a URI. They are defined by the following four concepts:

- 1 identification of a resource with a URI
- 2 the definition of a uniform interface using HTTP standard verbs (GET, POST, PUT and DELETE)
- 3 the use of hypertext links

- 4 communication through HTTP using formats such as JSON, YAML or XML (Gyrard et al., 2015).

A comparative table between Ws-* and REST services is presented in Mashal et al. (2015).

3.2 Context-aware system

In the following, we present several definitions to describe the notion of context-aware system.

- *Context*: Different definitions are presented by many researchers, but the most complete is the one given by Dey (2001) as follows: “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application.”
- *Context-aware*: According to Dey (2001) again, a context-aware system “provides relevant information and/or services to the user, where the relevance depends on the user’s tasks.”
- *Context information life cycle*: It explains where the data is generated and where the data is consumed. Context awareness can be applied using four main phases (Perera et al., 2014):
 - a *Context acquisition*: Context needs to be acquired from various sources. The sources could be physical sensors or virtual sensors. The techniques used, can be varied based on the role, frequency, context source, sensor type, and acquisition process.
 - b *Context modelling*: The collected data needs to be modelled and represented in a meaningful way. The most popular context modelling techniques are: key-value, mark-up schemes, graphical, object-based, logic-based, and ontology-based modelling.
 - c *Context reasoning*: Context reasoning can be defined as a method of deducing new knowledge, and better understanding, based on the available context. We classify context-reasoning techniques broadly into six categories: supervised learning, unsupervised learning, rules, fuzzy logic, ontological reasoning, and probabilistic reasoning.
 - d *Context dissemination*: Finally, both high-level and low-level context need to be distributed to the consumers who are interested in the given context.

In this paper, we focus on contextual data modelling and context reasoning, which will add intelligence to web services. Our context reasoning uses PR-OWL language that is based on MEBNs.

3.3 MEBN, PR-OWL and UnBBayes

MEBN, proposed by Laskey (2008), combines the expressivity of FOL with the inferential power of BNs and can address reasoning challenges for complex and uncertain situations. MEBN provides a means for defining probability distributions over an

unbounded and varying number of interrelated hypotheses with the aid of a formal syntax, a set of model construction, and inference processes and semantics. MEBN can interpret the world as a set of entities that have attributes and have causal relationships with other entities (Park et al., 2016).

MEBN consists of a collection of MEBN fragments (MFrags) organised into an MTheory. An MTheory represents a particular domain of discourse. An MTheory is a collection of MFrags that satisfies conditions given in Laskey (2008) ensuring the existence of a unique joint distribution over its random variables (RVs). Each Mfrag, as a modular component, represents knowledge about specific subjects within the domain of discourse and it models probability information about a group of related RVs. Three types of nodes (resident nodes, context nodes, and input nodes) are defined in the MTheory. An important advantage of MEBN is that there is no fixed limit on the number of RV instances, and the RV instances are dynamically instantiated as needed. MEBN it has been applied to a wide variety of domains (Patnaikuni et al., 2017).

Park et al. (2013) introduced an MEBN machine learning from a relational database. Park and Laskey (2018) introduced a mapping between a relational schema and a partial MTheory. This mapping is called MEBN-RM mapping (or MEBN-RM). So, using MEBN-RM, the modeller (human or machine) can design the MEBN model seamlessly from a relational database.

- *Probabilistic web ontology language:* PR-OWL (Costa and Laskey, 2006) is used to enable MEBN reasoning compatible with ontology. It provides a set of ontology constructs to express the probability distribution information associated with ontology elements. PR-OWL is based on MEBN logic and can provide support to reasoning over uncertainty using FOL and probability. Furthermore, the reasoning process in PR-OWL ontology is an automatic generation of situation-specific Bayesian network (SSBN) determining the probabilities for a query.
- *The UnBBayes tool (Matsumoto et al., 2011):* It provides both graphical user interface (GUI) and Java APIs to build MEBN models, generate probability-annotated ontology to represent the MEBN models, and make uncertainty reasoning. It supports various probabilistic graphical models (e.g., BNs, influence diagrams, MSBN, OOBN, HBN, MEBN/PR-OWL, and PRM). The human-aided MEBNs learning ‘HML’ is an UnBBayes plug-in that enables users to create an MEBN model from a relational database (Park et al., 2016).

3.4 Smart web services

A smart service is defined in Lee et al. (2012) 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 He et al. (2012), a smart service is defined as a general model that enables the integration of any function 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 Maleshkova et al. (2016), a smart web service ‘SmartWS’ is defined as a web API that conforms to standards (e.g., HTTP and 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 characterised by:

- 1 autonomy
- 2 automatic behaviour
- 3 intelligence
- 4 communication capacity
- 5 customisation
- 6 surveillance and control capacities of objects.

4 Related work

WoT is a promising area in technology that is growing day by day, and numerous works have been conducted in this field. In Liu et al. (2011), 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, (Lee et al., 2012; Terdjimi et al., 2016). Nevertheless, very little work has been made on WoT SSF modelling. In Guinard et al. (2010, 2011), two intelligent applications are shown: ‘energy visualiser’ for energy consumption surveillance and control of household devices, and ‘ambient metre’ in order to obtain energy consumption of machines.

A model of SSF based on context adaptation is presented in Lee et al. (2012) with a central system scenario. The authors founded their work on Schilit et al. (1994) to define the context. The work in question represents the 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 He et al. (2012), an example of intelligent service for plant watering depending on context is presented based on Thing-REST model. Subsequently, an extension of the aforementioned work by replacing HTTP with CoAP is presented in Elias et al. (2012), introducing the semantic aspect in the event detection service. A new tendency to develop web services built on micro-services also recently emerged (Zeiner et al., 2016). While the aforementioned contributions were based on the notion of context, others were proposed that are focused on semantic web. For instance, a personalised meteorological semantic service, as well as an improvement of functionalities of a service for disaster management, is presented in Lee et al. (2014). In Beltran et al. (2014), 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 semantics and SSF. Nevertheless, there was no extension of the intelligence notion in these works. Alirezaie et al. (2017) presented a framework for smart homes called e-care@home, which able to perform context recognition based on the activities and the events occurring in the home. The authors proposed also a semantic model for the smart homes.

Table 1 Summary of the state-of-the-art on smart web services

Approach	Domain	Proactive	Context	Semantic	Modelling	Reasoning	Uncertainty	Architecture	Protocol	Data format	Application
Guinard et al. (2010)	WoT	-	-	-	-	-	-	Web architecture (Sun Spot)	REST	JSON	Ambient metre energy visual
He et al. (2012)	WoT	-	✓	✓	Pair/triplet	Thing rest	-	Resource oriented ROA	Thing+rest	XML	Plant watering system
Lee et al. (2012)	WoT	-	✓	-	UML	Markov, Bayesian network	-	Multi-agent	-	XML	Smart home
Wei and Jin (2012)	IoT	-	✓	✓	Ontology	Bayesian network	✓	SOA	-	RDF	-
Biswal et al. (2015)	IoT	-	-	-	-	Edge and cloud	-	Multi-layer	MQTT, CoAP	Context. data	Agriculture
Khalifi and Benslimane (2015)	Pervasive comput.	✓	✓	✓	UML	Rule/inference	-	SOA	Universal Plug and Play	XML	Touristic
Mrissa et al. (2015)	WoT	-	✓	✓	Ontology	-	-	Avatar	REST	XML	Temperature regulator
Maleshkova et al. (2016)	WoT	-	✓	✓	-	Rules/inference	-	Multi-layer	REST	XML/JSON	Medical diagnostic
Alirezai et al. (2017)	IoT	-	✓	✓	Ontology	Rules/inference	-	Multi-agent	Time-slotted channel hopping (TSCH)	Contextual data	Smart home
Kissoon et al. (2017)	IoT	-	-	Decision tree	-	-	-	Microsoft Azure Cloud	-	Raw data	Agriculture
Li et al. (2017)	PSA	-	✓	✓	Ontology	SWRL	✓	-	-	RDF	Underwater robots
Machado et al. (2017)	AAL	✓	✓	✓	Ontology	MEBN	-	SOA	-	RDF	Smart houses
Izuddin et al. (2018)	MATLAB and Simulink	-	-	-	-	PR-OWL	-	-	-	Raw data	Agriculture
Thamaraimandalan et al. (2018)	IoT	-	-	-	-	Fuzzy logic	-	-	-	IOS	Nodecnu
<i>Our approach</i>	<i>WoT</i>	✓	✓	✓	<i>PR-OWL</i>	<i>Multi-entity Bayesian network/MEBN-HML</i>	✓	<i>Multi-layer</i>	<i>REST</i>	<i>RDF/OWL</i>	<i>Agriculture</i>

The model of smart web service ‘SmartWS’ in a form of a web API introduced in Maleshkova et al. (2016) enables the automatic adjustment of settings and consequently, adapts to the context of the triggered event. Alahmadi and Qureshi (2015) proposed a testing model to evaluate and validate intelligent applications as well as QoS depending on user requirements. In Mrissa et al. (2015), a new avatar model in the web of objects discovers the capacities of objects and exposes them as functionalities based on semantic techniques. A meta-model of context (Terdjimi et al., 2017) capable of responding to different adaptation questions for a given WoT application is proposed. Štrumbelj and Šikonja (2015) is interested in prediction based on managerial game data in making their decision. Wei and Jin (2012) presented the research based on the context model, the ontology, and the BN, and suggested a context-aware service discovery architecture for IoT to support smart service provision.

Machado et al. (2017) presented an approach based on the idea that an undesired situation may often occur as a result of previous situations or actions mistaken by the patient. This approach makes use of BNs and aims at an early detection of these undesired situations in order to avoid them. This work presented the importance of mechanisms that act not only reactively, but also proactively in AAL. Tiwari and Kothari (2016) presented a smart system in the medical field using rough set theory, from vague, incomplete, and uncertain data. With the internet of things and the computation power of the cloud, several studies proposed smart web services in the domain of agriculture, such as Biswal et al. (2015), Kissoon et al. (2017) and Thamaraimanalan et al. (2018). In Izzuddin et al. (2018), a smart irrigation system was simulated and based on fuzzy logic. However, these works do not take in consideration the semantic aspect and do not represent the uncertainty of the data.

Table 1 is a summary of the work cited in the state-of-the-art section. The different studies aim to introduce the context of objects with a semantic description in web services in order to make them ‘smart’. Our proposition is founded on the generic model of smart web service. It is not based only on context, as we have added temporal and uncertain aspects as well in order to predict the future behaviour of the system. This approach is based on proactive decision-making, seeking to adapt automatically to the environment according to the time-evolving situations. This combines with the characteristics of WoT, the semantic representation of the context, and MEBN-RM (Park and Laskey, 2018) is used to reason on uncertain information to derive high-level contexts from low-level time-series data streams to sense current situation. HML (Park et al., 2016) can be used to develop MEBN models.

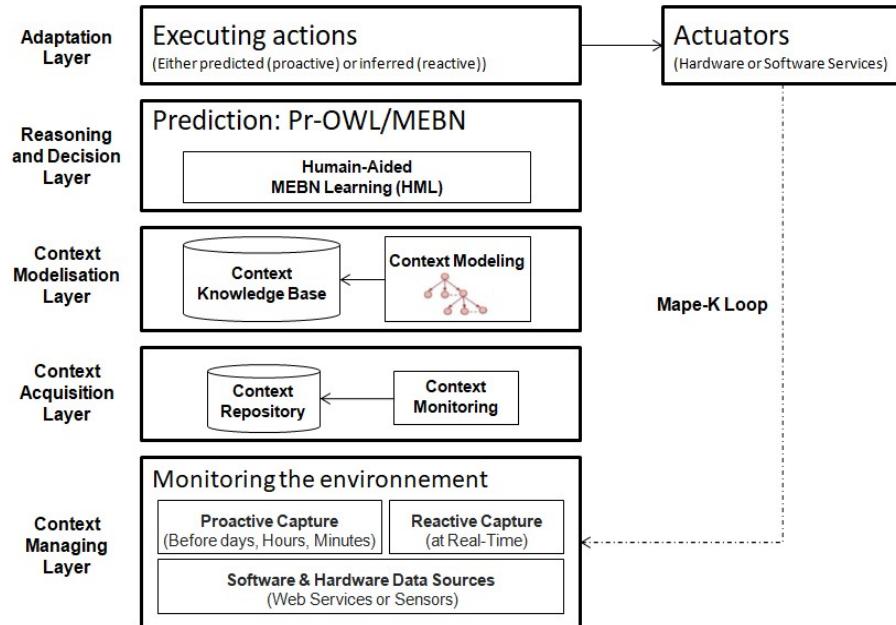
5 Proposed architecture

Figure 1 shows an overview of a multi-layered architecture that represents our framework of smart web services. The services provided by the context-aware framework generally comply with the lifecycle of context awareness (Perera et al., 2014). It provides the following functionalities:

- 1 data acquisition in order to define contextual data of objects and user
- 2 identifying the current situation of the user in order to understand the requested services

- 3 selecting appropriate services and adapt them in a proactive and reactive way in order to obtain appropriate system decision
- 4 adapting services in an autonomous way depending on context changes.

Figure 1 Framework for context-aware proactive-reactive services (see online version for colours)



5.1 Context managing layer

This includes all data sources, sensors, algorithms, and software components (e.g., 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 to make the right decision and avoid unwanted events.

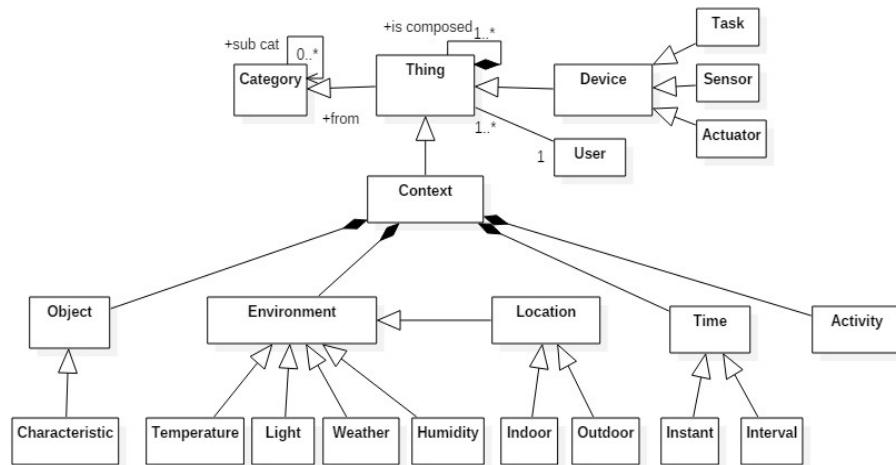
5.2 Context acquisition layer

The framework involves different types of objects that must collaborate. To acquire the context, each object is considered as a resource. Based on the REST architectural style, any resource equipped with a sensor is addressed by a unique identifier of standard format via uniform resource locators (URL) using the HTTP and its methods (e.g., GET, POST, PUT, and DELETE) to access them. Some context data are static while others are dynamic and sometimes, uncertain. We think that our proactive-reactive approach can offer an added value to web service mechanisms that adapt to context and user preferences, in order to make it a smart web service. These raw data are recorded in the ‘context repository’.

5.3 Context modelling layer

We use UML to model context in order to unify and understand it. Context plays an important role in intelligent service customisation depending on the situation. Each device is attached to a real-world object to identify it (e.g., RFID) or monitor and collect information with sensors, or react to change the state of an object with actuators. The work of Dey (2001) categorises context based on the 5W (why, where, who, when, and what). In our case, we represent context depending on the identity of an object, its environment, time, activity, and location (see Figure 2). Semantic context enables to unify context of objects using context ontology in order to facilitate interfacing with web services.

Figure 2 Context model



5.4 Reasoning and decision-making layer

It is the most important layer for the representation of the proactive system. It deducts situations based on acquired context. Figure 3 represents a model for contextual situations and their relationships with other entities:

- *Thing* is an entity (person, plant, building, etc.) that can connect to the digital world through device.
- *Device* is a component, software or hardware, which can provide information about an entity through sensors, actuators or RFID.
- *Context* represents a contextual data in the form $C_i = \text{Predicate}(\text{Entity}, \text{Value})$.

Entity E^* : a set of entity names.

Value $\in V^*$: a set of values for entities $\in E^*$.

Predicate $\in P^*$: a set of predicates, i.e., has-temperature, is-raining, is-located-in, has-humidity.

Context: $C = \sum_{i=1}^n C_i$, connected with Boolean operators (e.g., union, intersection, complement).

- *Event* is defined according to Etzion and Niblett (2011) as: “an occurrence in a system or a particular domain.” They can change the state of the environment and therefore produce new situations.

$$\text{Event} = (\text{name}, \text{type}, \text{predict}(R_c, \text{value}), t)$$

where name: name of the event and type: internal/external.

$$R_c : (\text{Relevant context}) = a_1, a_2, \dots, a_n$$

where $a_i \in E^*$ is a set of context attributes to modify.

Predicate (R_c , value): changes of values of R_c .

t : the time [$t = t_i$ if the event is discreet (instant), $t = [t_i, t_f]$ with $i < f$ if the event is continuous].

- *Situation ‘Sit’*: a set of contextual data instances, connecting between each to provide variable information in a specific timeframe.

$$\text{Sit} = (\text{Id}, t, \sum C_i) : (\text{current situation}), i = 1, \dots, n$$

with Id: object identity and t : time of context gathering.

$$\sum C_i : \text{set of context instances.}$$

- *Action ‘Act’*: represents the consequence of the applied rules on the system situation ‘Sit’. $\text{Act} = (\text{Sit}, \text{R})$ while, Sit: the current situation of context with R: the rules applied to the context.

The system can provide reactive and proactive action behaviour to manipulate the situation. In this work, we have studied the proactive action because we need the uncertain information in our scenario for better management quality:

- Reactive action is defined as the execution of an action that influences events that are happening and characterises the current situation. This situation may be detected through inference rules supported by the semantic web rule language (SWRL). We utilised Jess (<http://www.jessrules.com>), which is an inference engine, to execute these rules. For example:

```
GardenPlant(?x, true) ^ hasHumidity(?y, High) ^ isRain(?e, false)
  ^ isWind(?e, false) ^ hasTemperature(?e, ?tem) ^ swrlb : greaterthan(?tem, 30)
  → Dry(?x, true)
```

- Proactive action perceives the events that are happening and determinate an unwanted future situation.

In our framework, the reasoning is based on MEBN to represent and reason on complex, dynamic and uncertain situations. The work of Golestan et al. (2016) concluded that

MEBN covers the necessary characteristics to represent a complex situation in the most complete possible way. The proactive action of the system is deduced from MEBN implemented in PR-OWL to describe a form of correlation between the events that constitute the situation in the future. Manually modelling by an expert of the field is laborious and is founded on the process of uncertainty modelling for semantic technology (UMP-ST) (Carvalho et al., 2017). We considered the work of Park et al. (2016) as a base of this work that introduced a process model for MEBN learning assisted by a human, which is called HML. It contains the four main disciplines, the presenting process contains the four steps (Figure 4):

- 1 analyse requirements
- 2 design world model and rules
- 3 construct reasoning model
- 4 test reasoning model.

Figure 3 Situation model

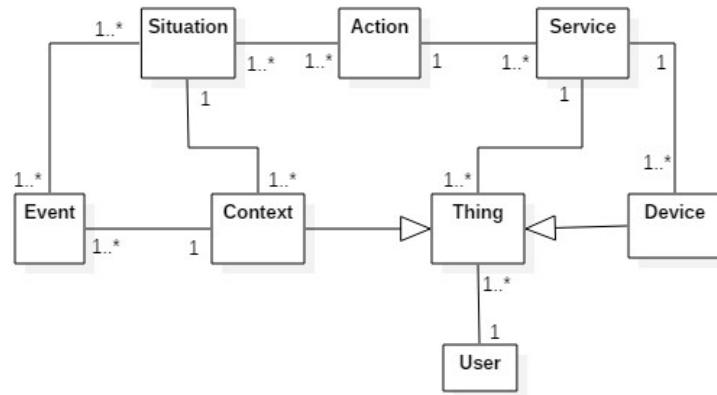
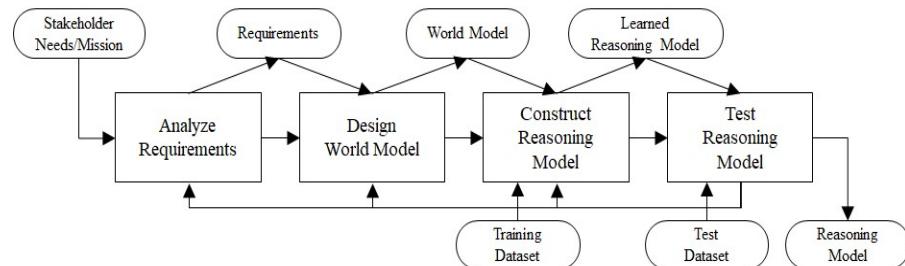


Figure 4 The process of HML



Source: This figure provided by permission of Park et al. (2016)

In the analyses requirements step, specific requirements for a reasoning model are identified. These are used to develop an MTheory. In the design world model and rules step, a target world model and rules for attributes in the world model are defined. In the

construct reasoning model step, a training dataset can be an input for MEBN learning to learn a reasoning model. In the test reasoning model step, a test dataset can be an input for the evaluation of the learned reasoning model.

An output of the process is the evaluated reasoning model. The work of Park et al. (2016) describes in detail these four steps. An intelligent plant-watering example, based on this process, will be presented in the next section, to explain the different steps of the process.

5.5 Adaptation layer

The adaptation decision adjusts and customises 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 (Bouguettaya et al., 2013) that contains the state indicators: monitoring, analysis, planning, and execution.

6 Watering system use case

In this section, we introduce an illustrative example using the proposed framework. The purpose of this use case is to show how the architecture is applied to an intelligent watering system based on WoT and introduce the proactive decision-making for the intelligent watering system. The intelligent watering system, which is a proof-of-concept system, aims to support automatic decision-making on whether to provide water or not to a region depending on environmental and plant conditions (e.g., weather and plant health status).

Our approach is generic and useful for several other scenarios. As examples, but not limited to, it can also be applied to disaster prediction, where the sensors and data coming from different sources can contribute to anticipating and helping proactive management of natural disasters such as storms or floods.

For the proactive decision-making, this system uses MEBNs, a foundation of PR-OWL, which can be used to the predictive analysis for future situations (Park et al., 2017). Basically, an MEBN model or MTheory, representing situations, is required to develop the system. In this research, a dynamic watering system MTheory, representing dynamic situations over time regarding plant watering, is developed.

The MTheory is used at the level of the reasoning phase in the system. Given environmental and plant factors, the system responds proactively, as the context changes. Then, the future situation is predicted in order to respond accordingly. Especially, we focus on the prediction of the degree of dryness or soil moisture.

An MTheory can be constructed by experts as well as a machine learning approach. In this paper, we use a machine learning approach, called HML (Park et al., 2016). HML contains the four main steps:

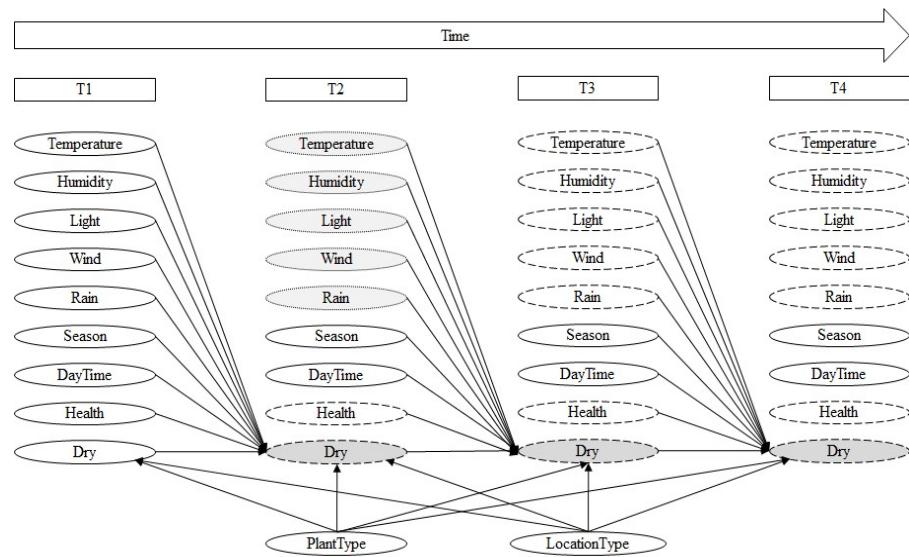
- 1 analyse requirements
- 2 design world model and rules
- 3 construct reasoning model
- 4 test reasoning model.

We use these steps to develop the dynamic watering system MTheory. The following sub-sections describe them.

6.1 Analysis requirements

In this step, we identify requirements for the target MTheory. To do this, we conduct an operational scenario for a simple watering situation. The goal of the intelligent watering system is to proactively find out whether the land is dry or not. We identify several dryness factors regarding as shown in Figure 5.

Figure 5 Environmental and plant factors regarding the dryness in a certain region and their causal relationships



In Figure 5, the ellipse with the solid outline stands for the observation. For example, T1 means the current time and the current temperatures observed (the ellipse under the time T1 in Figure 5), while the light grey-coloured ellipse with the dotted outline stands for the forecast. For example, T2 means the first future time and we get forecasts from the news, so we can set these as the evidence. The ellipse with the dashed outline stands for the uncertain factors. For example, we do not know these now and there are no forecasts as well, but these factors are used to estimate and/or predict the degree of dryness. The dark grey-coloured ellipse with the title dry stands for the target, the degree of dryness, or soil moisture. For example, the ellipses dry in T2, T3, and T4 are uncertain and our target RVs we want to predict. The model contains two types of attributes: static (e.g., plant type) and dynamic (e.g., temperature). The decision of the action to be taken depends on the attribute dry. The context depends on the location (e.g., region), the environment (e.g., wind and rain), characteristics of the plant (e.g., indoor or outdoor), its state (e.g., health), and the time (e.g., day or night). These are essential elements to trigger proactive actions.

The above model is based on several assumptions. For example:

- 1 the previous temperature does not influence the current temperature
- 2 there are no other factors influencing the dry
- 3 the current rain does not influence the current temperature.

The assumptions in the above model should be more realistic and this will be our future research agenda.

By analysing the context of the watering situation, we identify the system goal, query, and supporting evidence as shown by the following:

- Goal: Predict the degree of dryness in a certain region.
- Query: The degree of dryness (dry).
- Evidence: Environmental and plant factors (temperature, humidity, light, wind, rain, season, daytime, health, dry, plant type, and location type).

To achieve this goal, the system uses the dynamic watering system MTheory to reason about the dry using environmental and plant evidence (e.g., temperature and humidity). In the next sub-section, a target world model and rules are derived.

6.2 Design world model and rules

In this step, we develop a world model and rules (or causal relationships). The world model represents the context where the system deals with by specifying entities, attributes of the entities, and relationships between entities. We represented a real use case by a relational model as shown in Figure 6. The rules indicate causal relationships between the attributes of entities.

Figure 6 World model and rules of the intelligence watering system

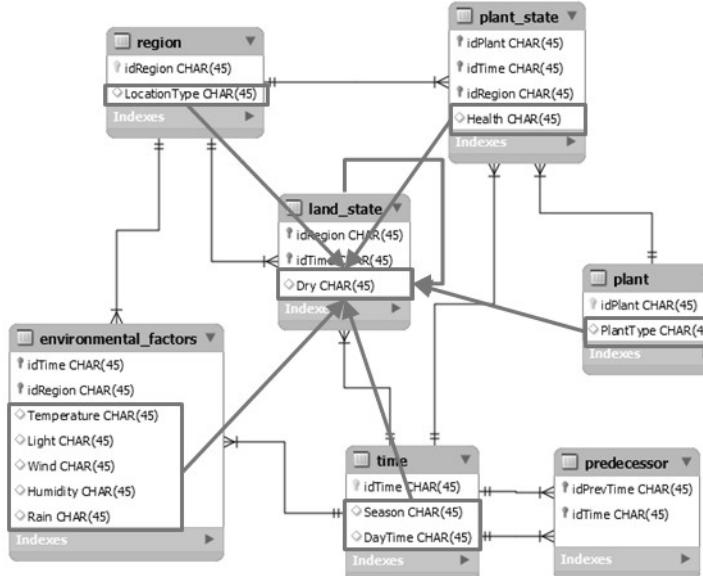


Figure 6 shows the world model and rules of the intelligent watering system. The world model contains seven semantic relations (*region*, *plant_state*, *land_state*, *environmental_factors*, *time*, *plant*, and *predecessor*) and three entities (*region*, *time*, and *plant*). Some semantic relations contain attributes. For example, the semantic relation *environmental_factors* includes the attributes temperature, light, wind, humidity, and rain. As we discussed in the previous sub-section, these attributes influence the degree of dryness (i.e., the dry). Arrow lines in the figure show these causal relationships and are used to define the conditional probability P .

$$P(D_{t+1} | D_t, H_t, T_t, G_t, W_t, M_t, R_t, S_t, DT_t, PT, LT)$$

where t denotes a time stamp and the letters D , H , T , G , W , M , R , S , DT , PT , and LT denote the attributes dry, health, temperature, light, wind, humidity, rain, season, day time, plant type, and location type, respectively. In our model, the attributes PT and LT are time-invariant, while other attributes are time-variant. The distribution for the above conditional probability is learned in the next sub-section.

6.3 Construct reasoning model

In this step, an MTheory is developed by using MEBN learning (Park et al., 2017; Park and Laskey, 2018). MEBN learning requires a training dataset that is stored in a relational database management system (e.g., MySQL). For the relational database, the relational schema in Figure 6 is used and a synthetic training dataset is used. Note that an actual dataset will be used in our future research. In order to make the training dataset, a BN model was developed according to the model in Figure 5. In addition, the parameters of the BN were inputted according to patterns from open IoT data (<https://thingspeak.com/channels/130691>, <https://thingspeak.com/channels/429987>, <https://thingspeak.com/channels/14664>). The BN was used to generate the training dataset with 864 cases. Figure 7 shows the trends of dryness (Y-axis) over time (X-axis) in the training dataset. These trends differ according to plant types (*garden plant* and *others*), land types (*indoor* and *outdoor*), and seasons (*spring*, *summer*, *fall*, and *winter*). HML-UnBBayes (<https://hml-unbbayes.github>) is a software tool for MEBN learning. We modified the tool so that it can learn a dynamic MTheory. The source codes for our research (used for the BN, the synthetic training/test dataset, the learned MEBN model, and the SSBN constructed from the MEBN model) can be found in the GitHub repository (https://github.com/HML-UnBBayes/hml/tree/master/src/test/java/hml/text_mode_test/watering_system_dynamic).

Figure 8 shows the learned watering system MTheory and Figure 9 shows the learned conditional probability distribution for dryness. The MTheory consists of seven MFrag, including *plant*, *region*, *time*, *environmental_factors*, *plant_state*, *predecessor*, and *land_state_Dry*. Each MFrag represents probabilistic knowledge for entities, RVs, and local probability distributions. For instance, the MFrag *land_state_Dry* represents a situation where the soil state of the plant depends on different inputs or conditions (e.g., type plant, temperature, and region type).

In addition, some entities (e.g., *time*, *region*, and *plant*) are defined by using the IsA contexts with ordinary variables. For example, the entity Region for the land the plant is located is defined by using the IsA context is A (*land_state_idRegion*, *region*), where the first argument *land_state_idRegion* is the ordinary variable and the second argument

region is the entity region. In this MTheory, the remarkable part is the MFrag predecessor, which is used to define sequential time stamps. Because there is no sequential order between time entities, it is necessary to explicitly define this by using a predicate (i.e., the node predecessor in the MFrag predecessor). The node predecessor ($predecessor_idPrevTime$, $predecessor_idTime$) means that the time $predecessor_idPrevTime$ occurs immediately before the time $predecessor_idTime$. The learned MTheory contains discrete and continuous resident nodes. Table 2 shows the list of these resident nodes.

Figure 7 Trends of the dryness in training dataset according to plant types, land types, and seasons

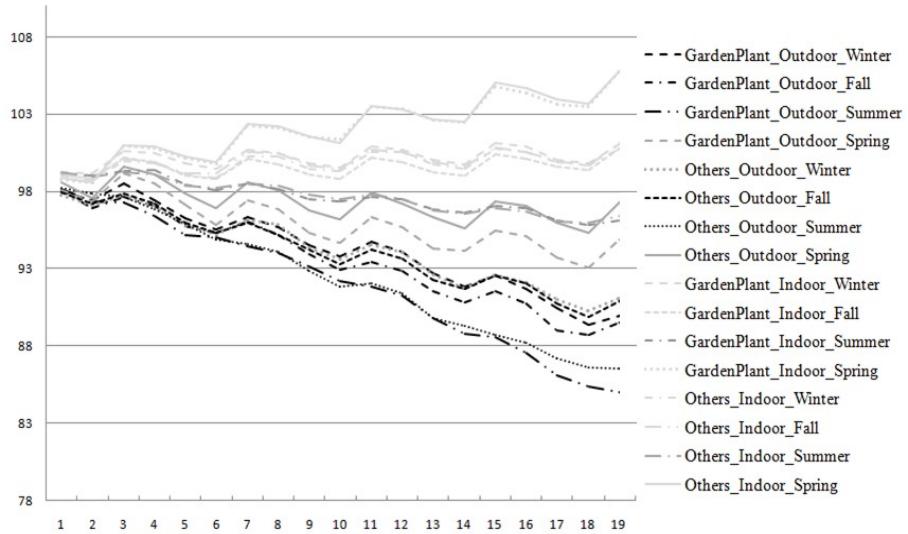


Table 2 Resident nodes in the learned MTheory

Name	Type	Values
plant_PlantType (PT)	Discrete distribution	GardenPlant/others
region_LocationType (LT)	Discrete distribution	Outdoor/indoor
time_Season (S)	Discrete distribution	Winter/Fall/Summer/Spring
time_DayTime (DT)	Discrete distribution	H24_6/H6_12/H12_18/H18_24
plant_state_Health (H)	Gaussian distribution	Real number
environmental_factors_Temperature (T)	Gaussian distribution	Real number
environmental_factors_Light (G)	Gaussian distribution	Real number
environmental_factors_Wind (W)	Gaussian distribution	Real number
environmental_factors_Humidity (M)	Gaussian distribution	Real number
environmental_factors_Rain (R)	Gaussian distribution	Real number
land_state_Dry (D)	Conditional linear Gaussian distribution	Real number
predecessor (PD)	Discrete distribution	False/true/absurd

Figure 8 The learned watering system MTheory

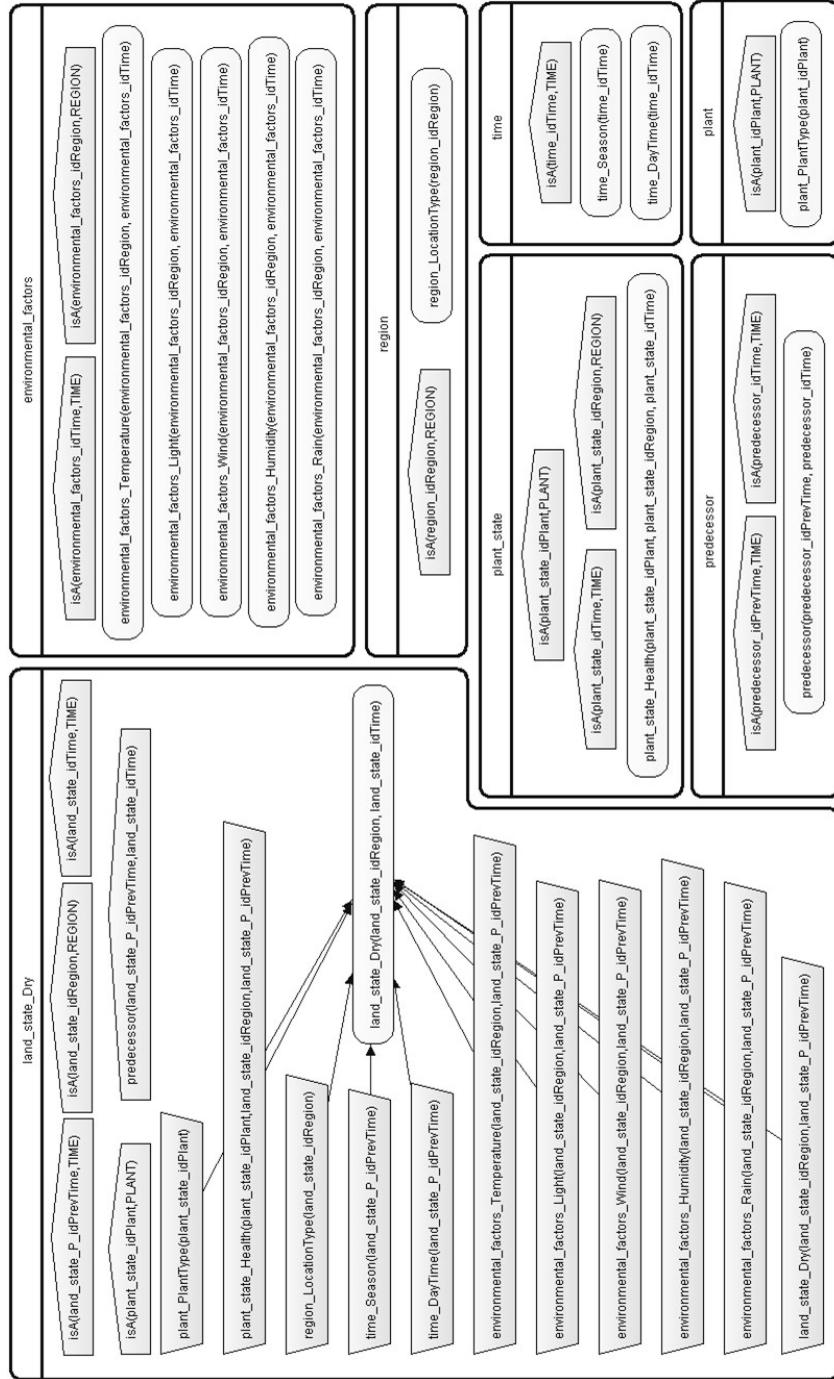
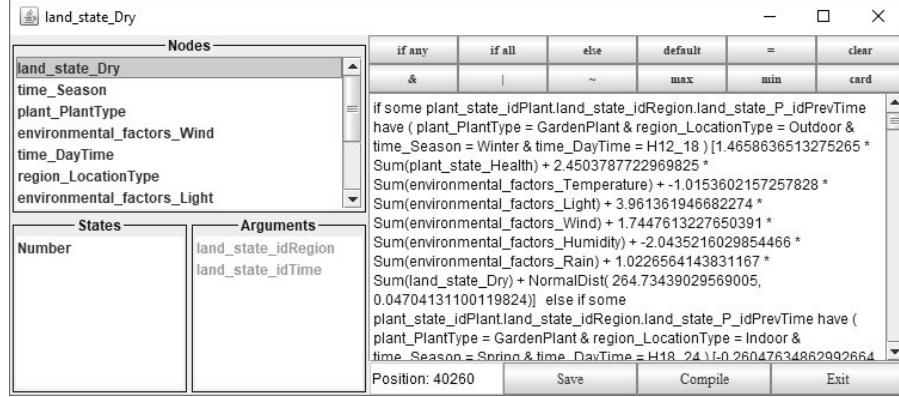


Figure 9 The conditional probability distribution for the dry

The learned conditional probability distribution for the dry (or `land_state_Dry`) contains various configurations for states of discrete parent nodes (e.g., $PT = GardenPlant$, $LT = Indoor$, $S = Fall$ and $DT = H12_18$). The total number of configurations for the dry is $64 = |PT| * |LT| * |S| * |DT|$.

Each configuration is described by the if-then statement and leads a conditional linear Gaussian distribution as shown in Figure 9. The equation below shows an example of the conditional linear Gaussian distribution, which was learned by the conditional linear Gaussian parameter learning (Park et al., 2013).

$$\begin{aligned}
 & P(D_{t+1} | PT_{t,GardenPlant}, LT_{t,Outdoor}, S_{t,Winter}, DT_{t,H12_18}, H_t, T_t, G_t, W_t, M_t, R_t, D_t) \\
 & = 1.46 * \text{Sum}(H_t) + 2.45 * \text{Sum}(T_t) - 1.02 * \text{Sum}(G_t) + 3.96 * \text{Sum}(W_t) + 1.74 \\
 & * \text{Sum}(M_t) - 2.04 * \text{Sum}(R_t) + 1.02 * \text{Sum}(D_t) + \text{NormalDist}(264.73, 0.05)
 \end{aligned}$$

where $\text{Sum}(X)$ denotes a summation function for instance RVs constructed from the resident node X and $\text{NormalDist}(a, b)$ denotes a normal distribution with a mean of a and a variance of b .

6.4 Test reasoning model

In this step, the learned MTheory (Figure 8) is evaluated using a test dataset in terms of usefulness for proactive decision-making. To evaluate the usefulness, the prediction accuracy of the MTheory is measured. Although in this research, synthetic datasets were used, the test in this step can support the feasibility of the SWA by confirming the predictive accuracy of the MTheory. To do this, the test dataset was generated by using the BN model in Sub-section 3. The same number of test cases was generated as in the training cases. Recall the operational scenario for the watering situation in Sub-section 1. There are the observed factors (e.g., temperature, wind, and humidity) at the current time $T1$. Given this evidence, the degree of dryness over time $T2$ is predicted. The posterior probability distribution for the degree of dryness is populated by a SSBN constructed from the learned MTheory (the SSBN image can be found in the smart watering example Github, https://github.com/HMLUnBBayes/html/blob/master/example_data/watering_system_dynamic/ssbn_from_the_learned_MEBN.png).

For this operation, we constructed the SSBN using some entity instances associated with the entities (region, time, and plant) in Sub-section 2. We define two times entity instances $T1$ and $T2$. In addition, we define 1 region entity instance $R1$ and one plant entity instance $P1$. Because there are no orders between the time entity instances, we define such orders by using the resident node predecessor. For example, predecessor $(T1, T2) = true$. Then, we constructed the SSBN containing RVs from the resident nodes $D, H, T, G, W, M, R, S, DT, PT$, and LT .

To reason about the degree of dryness, we set some evidence according to our operational scenario (i.e., the ellipse with the solid outline in Figure 5). The examples of the evidence include the RVs $plant_PlantType_P1 = GardenPlant$ and $time_DayTime_T1 = H12_18$.

Then the posterior distribution for the RV $and_state_Dry_R1_T2$ and the actual value corresponding to the attribute $land_state_Dry$ in the test dataset were compared using mean absolute error (MAE) and mean squared error (MSE). Specifically, the predicted means of the RV $and_state_Dry_R1_T2$ were used to compare to the actual values in the test dataset. Table 3 shows the comparison results.

Table 3 Summary of the MAE and the MSE between predicted mean values and actual values

Measure of difference	Values
Mean absolute error (MAE)	0.2106
Mean squared error (MSE)	0.0895

The prediction accuracy shows that the learned MTheory fits well the test data. This means that the learned MTheory can be used to predict the degree of dryness in order to support proactive decision-making.

7 Conclusions

One of major challenges of the WoT is intelligent decision-making and adaptable execution of WoT applications according to internal and external information. WoT applications should be able to understand a situation, to develop plans, to make decisions and to perform appropriate actions according to their environment. In this paper, we introduce a SWA for proactive and reactive context reasoning, which supports the operation of smart web services. The architecture contains four layers:

- 1 a sensing/actuating layer
- 2 a context acquisition layer
- 3 a context modelling layer
- 4 a reasoning/decision-making layer.

The architecture is based on PR-OWL to represent uncertain and dynamic contexts that are frequent in the WoT and proactively and reactively reason about such contexts. Proactive decision-making is interesting in the sense that it allows predicting dynamic situations and anticipate appropriate actions before such situations happen. Reactive decision-making is important for real-time response.

Ongoing research effort includes testing our architecture with a set of physical environments that will bring different constraints such as changes of geographical locations or user privacy constraints, and different requirements at the application level, in terms of quality of service (e.g., speed and accuracy). We also intend to extend our context model together with our reasoning mechanisms to support complex use cases. To do so, we will implement and test an extended set of reasoning rules with real life scenarios that require complex proactive actions (e.g., drought, an abundance of rain, and failure of irrigation equipment).

Acknowledgements

Michael Mrissa acknowledges the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program.

Cheol Young Park acknowledges BAIES, LLC for funding the ‘Development of a planning system using multi-entity Bayesian networks’ project (Grant Agreement #07152017).

We also acknowledge the Algerian Ministry of Higher Education and Scientific Research for funding the ‘Smart service and interface for ubiquitous computing’ project (Grant Agreement #C00L07ES220120120180002) under the University Training Research Program (PRFU).

References

Alahmadi, M. and Qureshi, M.R.J. (2015) ‘Improved model to test applications using smart services’, *Science International*, Vol. 27, No. 3, pp.2275–2280.

Alirezaie, M., Renoux, J., Köckemann, U., Kristoffersson, A., Karlsson, L., Blomqvist, E. and Loutfi, A. (2017) ‘An ontology-based context-aware system for smart homes: e-care@ home’, *Sensors*, Vol. 17, No. 7, p.1586.

Beltran, V., Ortiz, A.M., Hussein, D. and Crespi, N. (2014) ‘A semantic service creation platform for social IoT’, in *IEEE World Forum on Internet of Things (WF-IoT)*, pp.283–286.

Biswal, V., Singh, H., Jeberson, W. and Dhar, A. (2015) ‘Greeves: a smart houseplant watering and monitoring system’, *International Journal of Science, Engineering and Technology Research (IJSETR)*, Vol. 4, No. 7, pp.2499–2507.

Bouguettaya, A., Sheng, Q.Z. and Daniel, F. (2013) *Web Services Foundations*, 1st ed., Springer, New York.

Carvalho, R.N., Laskey, K.B. and Costa, P.C.G. (2017) ‘PR-OWL – a language for defining probabilistic ontologies’, *International Journal of Approximate Reasoning*, Vol. 91, No. C, pp.56–79.

Costa, P.C.G. and Laskey, K.B. (2006) ‘PR-OWL: a framework for probabilistic ontologies’, in Bennet, B. and Christiane, F. (Eds.): *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS2006)*, ser. *Frontiers in Artificial Intelligence and Applications*, IOS Press, Baltimore, MD, USA, November, Vol. 150, pp.237–249 [online]. Available:<http://www.booksonline.iospress.nl/Content/View.aspx?piid=2210>.

Dey, A.K. (2001) ‘Understanding and using context’, *Personal and Ubiquitous Computing*, Vol. 5, No. 1, pp.4–7.

Elias, S., Shivashankar, S. and Manoj, P. (2012) 'A REST based design for web of things in smart environments', in *2nd IEEE International Conference on Parallel, Distributed and Grid Computing*, Solan, India, pp.337–342.

Etzion, O. and Niblett, P. (2011) *Event Processing in Action*, Manning, Londres, Greenwich.

Golestan, K., Soua, R., Karray, F. and Kamel, M.S. (2016) 'Situation awareness within the context of connected cars: a comprehensive review and recent trends', *Information Fusion*, Vol. 29, No. C, pp.68–83, Elsevier.

Guinard, D. (2011) *A Web of Things Application Architecture – Integrating the Real-word into the Web*, PhD thesis, ETH Zurich, Switzerland.

Guinard, D., Trifa, V. and Wilde, E. (2010) 'A resource oriented architecture for the web of things', in *Proc. Internet of Things (IOT)*, Tokyo, Japan, November/December, pp.1–8.

Guinard, D., Trifa, V., Mattern, F. and Wilde, E. (2011) 'From the internet of things to the web of things: resource-oriented architecture and best practices', in *Architecting the Internet of Things*, pp.97–129, Springer, Berlin, Heidelberg.

Gyrard, A., Bonnet, C., Boudaoud, K. and Serrano, M. (2015) 'Assisting IoT projects and developers in designing interoperable semantic web of things applications', *IEEE International Conference in Data Science and Data Intensive Systems (DSDIS)*, pp.659–666, Sydney, Australia.

He, J., Zhang, Y., Huang, G. and Cao, J. (2012) 'A smart web service based on the context of things', *ACM Transactions on Internet Technology*, Vol. 11, No. 3, pp.1–23.

Izzuddin, T.A., Johari, M.A., Rashid, M.Z.A. and Jali, M.H. (2018) 'Smart irrigation using fuzzy logic method', *ARPN Journal of Engineering and Applied Sciences*, Vol. 13, No. 2, pp.517–522.

Khalfi, M.F. and Benslimane, S.M. (2015) 'Meta model context based space for ubiquitous computing', *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)*, Vol. 7, No. 2, pp.51–66.

Kissoon, D., Deerpaul, H. and Mungur, A. (2017) 'A smart irrigation and monitoring system', *International Journal of Computer Applications*, Vol. 163, No. 8, pp.975–8887.

Krupitzer, C., Roth, F.M., VanSyckel, S., Schiele, G. and Becker, C. (2015) 'A survey on engineering approaches for self-adaptive systems', *Pervasive and Mobile Computing*, Vol. 17, No. PB, pp.184–206.

Laskey, K.B. (2008) 'MEBN: a language for first-order Bayesian knowledge bases', *Artificial Intelligence*, Vol. 172, Nos. 2–3, pp.140–178.

Lee, J., Kim, Y. and Kwon, S. (2014) 'Semantic WISE: an applying of semantic IoT platform for weather information service engine', in *International Semantic Web Conference*, 19–23 October, Riva del Garda, Italy.

Lee, J.Y., Kim, M.K., La, H.J. and Kim, S.D. (2012) 'A software framework for enabling smart services', in *Fifth IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, Taipei, Taiwan, pp.1–8.

Li, X., Martínez, J.F. and Rubio, G. (2017) 'Towards a hybrid approach to context reasoning for underwater robots', *Applied Sciences*, Vol. 7, No. 2, p.183.

Liu, W., Li, X. and Huang, D. (2011) 'A survey on context awareness', in *Computer Science and Service System (CSSS)*, pp.144–147.

Machado, A., Maran, V., Augustin, I., Wives, L.K. and de Oliveira, J.P.M. (2017) 'Reactive, proactive, and extensible situation-awareness in ambient assisted living', *Expert Systems with Applications*, Vol. 76, No. C, pp.21–35.

Maleshkova, M., Philipp, P., Sure-Vetter, Y. and Studer, R. (2016) 'Smart web services (SmartWS) – the future of services on the web', *Transactions on Advanced Research*, Vol. 12, No. 1, pp.15–26.

Mashal, I., Alsaryrah, O., Chung, T.Y., Yang, C.Z., Kuo, W.H. and Agrawal, D.P. (2015) 'Choices for interaction with things on internet and underlying issues', *Ad Hoc Networks*, Vol. 28, No. C, pp.68–90.

Mathew, S.S., Atif, Y., Sheng, Q.Z. and Maamar, Z. (2013) 'The web of things – challenges and enabling technologies', in *Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence*, pp.1–23, Springer, Berlin, Heidelberg.

Matsumoto, S., Carvalho, R.N., Ladeira, M., da Costa, P.C.G., Santos, L.L., Silva, D. and Cai, K. (2011) 'UnBBayes: a Java framework for probabilistic models in AI', in *Java in Academia and Research*, pp.34–70, iConcept Press, Anerley, Australia.

Mattern, F. and Floerkemeier, C. (2010) 'From the internet of computers to the internet of things', in *From Active Data Management to Event-based Systems and More*, pp.242–259, Springer, Berlin, Heidelberg.

Mrissa, M., Médini, L., Jamont, J.P., Le Sommer, N. and Laplace, J. (2015) 'An avatar architecture for the web of things', *IEEE Internet Computing*, Vol. 9, No. 2, pp.30–38.

Park, C.Y. and Laskey, K.B. (2018) *MEBN-RM: A Mapping between Multi-entity Bayesian Network and Relational Model*, pp.1–25, arXiv preprint arXiv: 1806.02455.

Park, C.Y., Laskey, K.B., Costa, P. and Matsumoto, S. (2013) 'Multi-entity Bayesian networks learning in predictive situation awareness', *Proceedings of the 18th International Command and Control Technology and Research Symposium (ICCRTS 2013)*, Alexandria, Virginia, Vol. 4444, No. 703, pp.1–20.

Park, C.Y., Laskey, K.B., Costa, P.C. and Matsumoto, S. (2016) 'A process for human-aided multi-entity Bayesian networks learning in predictive situation awareness', in *19th International Conference on Information Fusion (FUSION)*, Heidelberg, Germany, pp.2116–2124.

Park, C.Y., Laskey, K.B., Salim, S. and Lee, J.Y. (2017) 'Predictive situation awareness model for smart manufacturing', in *20th International Conference on Information Fusion (Fusion)*, Xi'an, China, pp.1–8.

Patnaikuni, P., Srinivasan, R. and Gengaje, S.R. (2017) 'Survey of multi entity Bayesian networks (MEBN) and its applications in probabilistic reasoning', *International Journal of Advanced Research in Computer Science*, Vol. 8, No. 5, pp.2425–2429.

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014) 'Context aware computing for the internet of things: a survey', *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, pp.414–454.

Schilit, B., Adams, N. and Want, R. (1994) 'Context-aware computing applications', in *First Workshop on Mobile Computing Systems and Applications*, Santa Cruz, USA, pp.85–90.

Štrumbelj, E. and Šikonja, M.R. (2015) 'Predictive power of fantasy sports data for soccer forecasting', *International Journal of Data Mining, Modelling and Management*, Vol. 7, No. 2, p.154.

Terdjimi, M., Médini, L., Mrissa, M. and Le Sommer, N. (2016) 'An avatar-based adaptation workflow for the web of things', in *IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Montreal, Canada, pp.62–67.

Terdjimi, M., Médini, L., Mrissa, M. and Maleshkova, M. (2017) 'Multi-purpose adaptation in the web of things', in *International and Interdisciplinary Conference on Modeling and Using Context*, Springer, Cham., pp.213–226.

Thamaraimanalan, T., Vivekk, S.P., Satheeshkumar, G. and Saravanan, P. (2018) 'Smart garden monitoring system using IoT', *Asian Journal of Applied Science and Technology (AJAST)*, Vol. 2, No. 2, pp.18–192.

Tiwari, K.S. and Kothari, A.G. (2016) 'Design of intelligent system for medical applications using rough set theory', *International Journal of Data Mining, Modelling and Management*, Vol. 8, No. 3, pp.279–301.

Wei, Q. and Jin, Z. (2012) 'Service discovery for internet of things: a context awareness perspective', in *Proceedings of the 4th Asia-Pacific Symposium on Internetworkware*, ACM, New York, NY, USA, Vol. 25, No. 6, p.25.

Zeiner, H., Goller, M., Jiménez, V.J.E., Salmhofer, F. and Haas, W. (2016) 'Secos: web of things platform based on a microservices architecture and support of time-awareness', *e & i Elektrotechnik und Informations technik*, Vol. 133, No. 3, pp.158–162.

Zeng, D., Guo, S. and Cheng, Z. (2011) 'The web of things: a survey', *Journal of Communications*, Vol. 6, No. 6, pp.424–438.