



HAL
open science

A Survey of Middlewares for self-adaptation and context-aware in Cloud of Things environment

Olfa Souki, Raoudha Ben Djemaa, Ikram Amous, Florence Sèdes

► **To cite this version:**

Olfa Souki, Raoudha Ben Djemaa, Ikram Amous, Florence Sèdes. A Survey of Middlewares for self-adaptation and context-aware in Cloud of Things environment. *Procedia Computer Science*, 2022, 207: Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022) (Special issue), pp.2804-2813. <10.1016/j.procs.2022.09.338>. <hal-03835485>

HAL Id: hal-03835485

<https://hal.science/hal-03835485v1>

Submitted on 13 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC-ND 4.0 - Attribution - Non-commercial use - No Derivative Works - International License



26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

A Survey of Middlewares for self-adaptation and context-aware in Cloud of Things environment

Olfa Souki^a, Raoudha Ben Djemaa^b, Ikram Amous^c, Florence Sèdes^d

^aFaculty of Sciences Economics and Management of Sfax, University of Sfax, Miracl Laboratory, Tunisia

^bHigher Institute of Computer Science and Communication Technology, University of Sousse, Miracl Laboratory, Tunisia

^cNational School of Electronics and Telecommunications of Sfax, University of Sfax, Miracl Laboratory, Tunisia

^dToulouse Institute of Computer Science Research, University of Paul Sabatier Toulouse, IRIT Laboratory, France

Abstract

An important increase in the number of smart things, connected objects, smartphones, and sensors are founded every day, creating a huge amount of data called Big Data. Because of the dynamism of things and the limited capability of their resources and services to process and store this huge data, Cloud computing became an efficient, accessible, and reasonable solution. IoT in combination with Cloud functionalities provides a new phenomenon called the Cloud of Things (CoT) where many new possibilities are enabled. With the heterogeneity of different objects, the dynamism of their context, their distribution and the infinite cloud services, a middleware is a key solution. The main aim of this paper is to study middleware for self-adaptation of the Cloud of things environment into the context of objects. Toward this end, we first present some related features. Next, we compare some middlewares according to their characteristics. Then, we present some middlewares for static and dynamic adaptation and finally, we conclude this paper with a list of current challenges and issues in the design of a new middleware able to adapt dynamically and in execution time any cloud service into the context of the current object.

© 2022 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

Keywords: Middleware; Cloud of Things; Self-adaptation; Context-awareness; IoT.

1. Introduction

Internet of things (IoT) phenomenon appeared in 1999, as a system of a huge number of devices, objects or entities interconnected and provided with unique identifiers (UIDs). They are able to transfer data over a network without human-to-human or human-to-computer interaction. The devices create and share information everywhere and every time. This data gathered from various sources is recorded in databases and analyzed in the form of reports and then

E-mail address: soukiolfa92@gmail.com, raoudha.bendjemaa@isimsf.rnu.tn, ikram.amous@enetcom.usf.tn, Florence.Sedes@irit.fr

1877-0509 © 2022 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

10.1016/j.procs.2022.09.338

passed on to the higher ups in the management chain. Today, IoT is growing rapidly due to recent advancements in communications and sensor technologies. These sensors and objects continuously create a huge amount of data called Big Data. Due to massive increases in storage and the number of devices and computing power with a limited capabilities of things, IoT will be integrated with another system. With Cloud computing, even things with limited computational capabilities can perform intricate computations required for effective performance of the assigned task. Anything needs to have sensors and actuators and their decision making capabilities can be facilitated through the almost infinite computational capabilities of the Cloud [28]. However, Cloud-based IoT are deficient in handling service invocation that a sensor triggers with the occurrence of an event [4]. The interaction between events and services are absent in the actual IoT clouds.

There are several challenges that have been raised, such as management, aggregation, and storage for big produced data. So, to merge Cloud computing to the IoT as Cloud of Things (CoT), which provides virtually unlimited cloud services can enhance the large-scale IoT platforms. Despite this unlimited capability, there are still many other challenges to achieve. The most important, challenging problems are the heterogeneity of different objects, their dynamic context over the time and the heterogeneity of networks and protocols of communication. Also, the variety of cloud services which neglected the variation of context of the object (thing), is another challenge. Many researches are focused on the implementation of a middleware which sits between things and applications as a reliable platform for communication among things with different interfaces, operating systems, and architectures. Every study, focusing on context awareness in internet of things, neglects the context of object and treats only the context of the user. But in our work, we want to focus on implementing a middleware situated between Cloud environment and things. This middleware focuses on context-aware service on Cloud of things. It will rely suitable cloud service according to the context variation of objects in real time.

This paper focuses on middleware technologies for self-adaptation of cloud service to context of object and aims to study their roles and necessities in the CoT environment.

The objective of this paper is to primarily address the question: what are the related concepts of our domain and the existing middleware for self-adaptation of cloud of things services?

This paper is organized as follows: Section 2 provides a definition of Cloud of Things and some related concepts and terminologies. Section 3 discusses an overview of the related studies about middleware for self-adaptation in cloud of things. In Section 4, we discuss a few challenges and issues. Finally, we conclude this paper in Section 5.

2. CLOUD OF THINGS

Internet of things based on Cloud computing, with classical version of that requires more updating in response to the very quick variation of context of objects. An object needs to only have the sensors and the actuators, and their decision making capabilities can be facilitated through the almost infinite computational capabilities of the cloud. But authorized access to IoT cloud sensor data and services without relaxing user privacy is another challenging task [28]. Recently, the number of connected and embedded smart devices grows rapidly.

According to Cisco IBSG [16], the IoT world will include more than 80 billion objects in 2025. To resolve some issues related to this growth, the integration of the two systems: IoT and Cloud Computing called Cloud of things is strictly required. In a cloud-centric IoT architecture, the cloud platform provides storage resources for aggregated sensing data, as well as computing resources for data analysis and data mining for information retrieval and knowledge discovery on sensing data received via IoT objects [24]. To access to cloud service, things would be identified by Radio Frequency identification (RFID) to achieve intelligent identification and management. The general procedure for tag-based information acquisition services are as follows: First, an RFID tag is attached to a thing, then, a system of device accesses reads the information on the RFID tag (including the identity information on the thing), and makes a request to the name resolution server on the Internet of Things. In recent years, cloud and IoT have gained importance and a lot of work has been done in research and development as well as on the industrial side. Practical solutions are being proposed now, and it is the time where more scalable services could be created through the integration of these two interrelated and inter-dependended paradigms. CoT can help provide solutions for smart cities, smart healthcare, smart agriculture, remote healthcare, Big Data, data mining, social networking, vehicles connected to the clouds, fog computing/edge computing, energy efficiency, and many more. Internet of things [40] has a considerable application which can be classified based on technical features in four categories: information aggregation services, collaborative-

aware services, identity-related services and ubiquitous services. Objects in IoT environment have a dynamic context which we describe in detail in Section 2.1.

2.1. Context of objects

Context-awareness has become indispensable to the emergence of Pervasive and Ubiquitous Computing. Context plays a major part in leveraging information about situations and providing application adaptability. Due to the proliferation of sensors and smart devices along with the emergence of Internet of Things and the consequent generation of big data, the role of context is vital in making information meaningful by adding intelligence. Objects are physical objects/devices or virtual objects/devices/information having identities, physical attributes, and virtual personalities and use intelligent interfaces. Things are heterogeneous in nature. Also, an object which is an independent entity has some attributes which describe their context. This context information is for example location of object, status, responsibility, availability, type of service required, resource, event, the relationship between things, dependency etc. All these features will be acquired through our middleware as context of objects in CoT environment. Contextual information is dynamic in nature, and constant changes happen according to the varying situations. To manage the heterogeneity of objects, their dynamic and the many various types of cloud services, a middleware between things and cloud is our proposed solution to resolve these issues.

2.2. Middleware

Due to the huge population and the fact that all connected objects actively produce or request data and require various services, the issue of limited available resources is a key factor to be considered in the design of a middleware architecture. Middleware aims to provide solutions to frequently-encountered problems, such as heterogeneity, interoperability, security, and dependability [21]. Middleware is an important enabler that provides communication among heterogeneous things. It is a mid-layer between things and application services and provides an abstraction of the thing's functionality for application services.

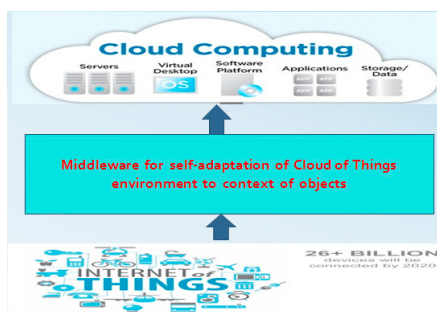


Fig. 1. Middleware for Cloud of things

In literature, middleware has many characteristics and features such flexibility, transparency, context management, interoperability, re-usability, platform portability, maintainability, trustworthiness management, adaptability, security and privacy, connectivity convergence and resource discovery.

Flexibility

This is one of the most important capabilities that a middleware can offer to the CoT systems. As a result, flexibility moves from the application level to the middleware to handle different forms of flexibility as response time or delay.

Transparency

Concerning transparency is in the form of platform where middleware can run on a variety of platforms, according to their requirements and network transparency when we do not need to know whether the resources are located locally or on remote devices.

Context management

CoT is a set of ubiquitous devices. Therefore, it is necessary to know the diversity of the objects technically, logically, and physically, which will lead to proper context management. The last one is a coordination, management concept. It allows users to choose and configure a service or subject in one application, and then all other applications containing information about that specific subject will adjust themselves to the same user defined setting. In our middleware, we are interested in context of devices and things. The nature of software and hardware heterogeneity of CoT requires interoperable components.

Interoperability

It will contribute to standardization and has different degrees depending on the heterogeneity level of the environment or system components. It allows two sets of applications on interconnected networks to be able to exchange data and services meaningfully with a different assumption on protocols, data models, and configuration, without any problems and additional programming efforts by the developers [17].

Re-usability

Concerning re-usability is to make designing and developments easier by modifying system components and assets for specific requirements. Cloud-based networks are crucial for obtaining this feature. These types of systems offer everything as a service that can be used multiple times.

Platform portability

In terms of platform portability, CoT platforms should be able to communicate from everywhere, anytime, with any device and can provide independence from network protocol, programming language, OS, and others. The maintainability is the ability of systems, applications or devices to respond to failures properly, and rapidly return to normal functionality without any problem [17].

Resource discovery

The resource discovery which is important feature in CoT systems. The last one includes multiple heterogeneous devices. Resource discovery is actually a process used by a node to search and probe for intended resources, such as services or data types, in the entire nodes of a network. After sending a search query, the resource discovery protocol will automatically choose the best resource that offers the most effective services and information [6].

Trustworthiness management

The next characteristic of middleware in CoT is trustworthiness management, which concerns authentication, hashing, encryption and others.

Adaptability

This feature can be divided into two big categories: static and dynamic or self-adaptation. It will be explained in detail in the next section.

Security and privacy

There are many data transmissions and operation occurs through CoT environment. So we need some security by considering confidentiality, integrity, and availability.

Connectivity convergence

Objects may use communication technologies, such as Zigbee and RFID. It is even possible for sensors to send directly analog signals or send digital signals after processing them. Therefore, connectivity APIs and management are essentially needed, which are convergent with interoperability. In the next section, we will present the adaptability features with their different type.

Adaptability

Middleware can react both statically and dynamically. Deploying optimized dynamic methods resulting better and higher adaptability which is critical in hard real-time systems. In contrast to flexibility, being adaptable, it denotes durability against long term changes in the systems. In CoT systems, environment and network infrastructure often change due to different reasons, including node mobility, power drain, topology shifts and others. Many usually have fixed capabilities and cannot be customized to dynamic and unpredictable situations. The adaptation can also be autonomic, supervised, structurally or behaviorally. We will be interested in static and self-adaptation[12] processes.

Static adaptation. Adaptation is related to a change in the structure or operation of an organization that makes it better suited to its environment. Static adaptation is the basic and easy process to adapt any service, application or system. It is necessary to have some patterns prepared in advance to react to context change. It is not an efficient solution in case of continuous and unpredictable dynamites of object context.

Self-adaptation. The CoT system is an active and very dynamic environment. Objects change context many times in run time. It is necessary to have a technique integrated in middleware to resolve this issue and adapt cloud services to new contexts of object in run time and with transparency. Self-adaptation in our middleware is the mechanism of reaction to events related to context of the object and deals with the unpredictable changes that can occur in the CoT environment. To increase the degree of autonomy, robustness and flexibility of such a system we need a middleware for self-adaptation (Section 3.3). The autonomy reflects the ability of the system to adapt to change on its own without any human intervention, while the robustness reflects the ability of the system to ensure continuity of service in the event of failure and to constantly adapt to the operating environment [35]. Flexibility, on the other hand, reflects the flexibility of the system. This feature should be made in the way of adaptation. It must remain completely transparent to the user and the object.

3. RELATED WORK TO MIDDLEWARE IN CLOUD OF THINGS

With the recent integration of the cloud computing and the IoT environment to create Cloud of things environment, we can find many approaches and systems to resolve relative issues and problems. In this section we present the recent researches that focused on context awareness and middleware for self-adaptation in cloud of things.

3.1. Context awareness for object

The term of context awareness has appeared since 1994. Many works of literature have been introduced and defined this concept. [11] defines the term of "context awareness" as "A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context of use". To manage any context, we should apply four essential steps: context acquisition, context modeling, context reasoning and context distribution [29]. Acquiring the context is based on responsibility, frequency, context source, sensor type, and acquisition process. Modeling the context is based on modeling techniques. Reasoning the context is deducing new information from the available context using reasoning techniques. Finally, distribution of context is delivering context to the users using distribution methods like query and subscription. In Table 1, we present some recent context-aware middleware.

We present, in Table 1, some recent Context-aware Middleware with their methods of context management. For example, Context-aware knowledge based Middleware realized by [37], is a context-aware information management middleware. It addresses selective information delivery with respect to the user's role in the system. An ontology-based technique for context modeling and organization was used that defines users, roles, devices, alarms, and places in order to describe the environment, and a semantic reasoner was used for context reasoning. A building automation prototype was used to evaluate the feasibility of the middleware. Concerning Industry Middleware [15] is a distributed framework for monitoring and control of large scale systems by integrating heterogeneous smart objects. Key-value and ontology-based techniques were used to model the context and rule-based technique was used to reason the context. The middleware provides services such as dynamic spectrum management, distributed control logic, object virtualization, Wireless Sensor and Actuator Network (WSAN) gateways, an SCADA gateway service, and data fusion transport capability.

Table 1. Evaluation of context-aware middleware

Context-aware Middleware	Year	Context modeling	Context Reasoning	Information Organisation
Context-aware knowledge based Middleware	2015	Ontology	Semantic reasoner	Ontology
Industry Middleware	2015	Key-value, Ontology	Rule-based	Ontology
IoP Middleware	2015	×	×	×
SeCoMan	2016	Ontology	Semantic reasoner	Ontology
CONASYS	2017	Key-value, Markup	Rule-based	Logic
Cooperative Middleware	2017	Ontology	Rule-based, Ontology	Ontology
POSTMAN	2017	Logic	NS	Database
RAON	2017	×	×	×
Senz	2017	×	×	×
MidSHM	2018	×	×	×

3.2. Revisiting the literature

Many researches are focused both on Cloud computing and the internet of things. Some of them treat the notion of context-awareness and the adaptability of service in the internet of things. In addition to the heterogeneity of devices in CoT, the environment cloud and things are very open, highly dynamic and uncertain. The dynamic nature is due to the frequent changes taking place in smart environments while the uncertain nature is due to the appearance of these changes in a random and unpredictable manner. These ongoing changes are related to the volatility of present objects (additions and withdrawals of objects, mobility in space) and mobility of users and their needs. In addition, the various finding in a smart environment can connect and disconnect in an unpredictable way as they move or because of breakdowns or energy saving. As a result, new features are being offered by devices that appear and other features are deleted because of the devices that are disappearing. This latter case may cause a consequent degradation of the quality provided by an object. This degradation can range from a simple momentary unavailability of certain functionalities to their final disappearance in case of major failures. Some answers to these challenges are now offered by the new approaches middleware to hide the heterogeneity and distribution of the different devices and cloud environment. In addition, the high dynamism of these environments imposes application flexibility and a loose aspect of inter entity coupling involved. Oriented service paradigm is one of the answers promising at this request. In the researches of [39], [10] and [32], authors focused their studies on the combination of Cloud computing and IoT. Concerning the work of [14], author proved that CoT is necessary to build a smart environment. According to [1], there are situations where exchanging data between IoT and Cloud is not reasonable, so they propose a smart gateway which can analyze requests and decides whether to answer them locally or sends them to the Cloud. Concerning middleware, we find some researches which focused in CoT or even in IoT. For example, [30] proposes a middleware for its called CASSARAM (Context-Aware Sensor probe, Selection And Ranking Model). This one is addressing the research challenges of choosing sensors when there are a large number that are overlapping and sometimes redundant. In addition, research of [31] is about a middleware which is a new application layer resolution for interoperability. The main concept of this middleware is to use device semantics provided by available specifications as semantic services. Other studies concern middleware for mobile devices that tries to collect and process sensor data for resource constrained proposed by [29]. Some of them have treated context-aware computing since IoT based on Cloud. But all of them concentrate in context of user and neglect context of the object (thing). In order to model a context and adapt it to any domain, it is necessary to create an ontology that captures generic concepts to a higher level. [3] proposes a new context model which provides mechanisms to extend the specific information of a context called CAMEOnto. It is an ontology based on the principles of 5Ws: who, when, what, where and why. There are some other work about context-aware modeling based on Ontology is presented in [34] which presents the emergent concept "Semantic Web of Things" (SWoT) [22] and [36]. SWoT is the combination of five elements: human (such as users, service providers), machines (such as computer), physical things (such as sensors or devices), abstract things (such as data, information or services), and the working environment (such as the Ad hoc network, the Sensor Network, or the Web). The authors of [2] present a reflective middleware for context-aware applications in the cloud called CARMiCLOC. This one offers various services for context-aware systems, among which is found the modeling. The context modeling is proposed

using an ontology [27], because it is a technique that gains force as a standard, due to its semantic expressiveness and interoperability.

Various studies related to context-aware application has been realised in [30], [38] and [26]. For example, work of [30] shows an ideal conceptual architecture of context-aware application and a taxonomy of functions and components that were present in 50 studies between 2001 and 2011 in the context-aware application in the internet of things. There are some research concentrates on the evaluation of the context management middleware. Hence, a context middleware is assumed that is able to support the users in their everyday life with their device(s) moving through their usual environment seamlessly [26]. Concerning [9], They present SAM (the Semantic-aware Middleware) and the SAM Store. The latter resembles a registry. It allows component providers to publish semantically enriched interfaces and to offer corresponding local and remote implementations. SAM is a middleware that enables various IPC (interprocess communication messages) patterns for Mobile Cloud Computing (MCC).

In addition, the paper of [33] presents a general service-architecture illustrated with DropLock platform. It highlighted the convergence between the Internet of Things and Mobile Cloud Computing. In literature, we find some cloud-based middleware, which supported context-awareness and are designed for specific applications: Mobile computing, Smart city, Smart Home, E-Health, Mobile multimedia application, etc. But there are middleware, such as LinkSmart (an ambient intelligent middleware), that offer a wide range of services for different needs. It acts and reacts based-on object presence [8]. From these middleware, we have been selected some middleware which are cloud-based and covering all architectural styles, working on different application domains, being commercialized and using different infrastructure. The [23] proposes ABC and S -based car parking middleware (the Always Best Connected and best Served). The authors proposed a three layer architecture that consists of the sensor layer, communication layer, and application layer. The last one is responsible for providing some cloud-based services for actions. CHOReOS middleware is a set of software components that implement and execute large-scale web service compositions. This one has been realized by [20]. CHOReOS is a mixture of four main modules: eXecutable Service Composition (XSC), which is responsible for coordinating the proper and required services for things. The second eXtensible Service Access (XSA) that is used for accessing and selecting services and things. Third eXtensible Service Discovery (XSD) , which is a management framework for protocols and processes to find requested or suitable services and things. The Forth is Cloud and Grid Middleware that is responsible for computer components and control implementation of choreographs. XSC has two main components: Composition and Estimation (C and E) and Reconfiguration Management for Service Substitution. [17] identified the features and characteristics of middleware such as flexibility, transparency, context management, interoperability, re-usability, platform portability, maintainability, resource discovery, trustworthiness management, adaptability, security and privacy, and connectivity convergence. In Table 2, we choose the most important features to compare some middleware for IoT based on cloud computing. From these features, what concerns us are context awareness and the adaptability.

Table 2. Comparison of some middleware IoT cloud-based

Middleware	Event detection	Service Discovery	Adaptability	Platform portability	Interoperability	Context awareness	Real time
ABC&S	√	√	√	√	×	√	×
CHOReOS	√	√	√	√	√	√	√
C-MOSDEN	√	√	√	√	√	√	NS
OpenIoT	NS	√	√	√	√	√	NS
ThingWorx	√	√	√	√	√	√	√
Carriots	NS	√	√	√	√	NS	√
DropLock	NS	NS	√	√	NS	NS	×
Rimware	√	×	√	√	√	×	NS
Xively	√	√	√	√	√	√	√

3.3. Middleware for self-adaptation

In literature, we find some researches which focused on self-adaptation of cloud services. For instance, [19] proposes three Fuzzy algorithms for self-adaptive SaaS (Software as a Service) to prevent SLA (Service Level Agree-

ment) violations for composite SaaS. The proposed algorithms have the merit of confronting three important issues, namely: the dependence of QoS parameters (e.g. availability) on low level metrics (e.g. uptime, downtime); the selection of service adaptation strategies that can cover both the business and IT infrastructure layers and have different time and computational complexity; and minimization of the cost and impact of changes. Autonomous computing is a paradigm for the design of self-adaptive software systems [5]. Such systems have the ability to self-adapt, during execution, to maintain a quality of service, topology or, in general, optimizing an objective function. The architecture of such systems is based in general on a model of closed control loop [18]. This architecture, presented in [25] known as MAPE-K, is based on a monitoring module (Monitor) of the system to be managed, an analysis module (Analyze) of the collected information, a planning module (Plan) Actions to be executed by the analysis phase and a module to execute (Execute) the decisions taken. All these modules shares a body of knowledge (Knowledge) that can be linked to the software system and also to the resources on which the system is running. In the literature, There are three reasons for adapting an application. Each corresponds to a type of adaptation. These are reactive adaptation, evolutionary adaptation and adaptation through integration. Reactive adaptation changes the behavior of the application in line with changes in execution context or user preferences. The scalable adaptation extends the functionality of an application to correct errors or to increase its performance. A change in the quality of user wishes can trigger an evolutionary adaptation. Integration adaptation deals with problems with incompatible services or components (different hardware/software interfaces, different protocols, etc.). This type of adaptation is often used when a decision to adapt comes to be taken. For example, when an adaptation was triggered by a change in context, the implementation of that adaptation may lead to one component not being able to communicate with another because they use different protocols. In this case, the system may use integration, adaptation. In addition, there are three ways to decide when to make an adaptation: context-driven decision, QoS-driven decision, and user-driven decision. They often work together, but with different priorities depending on the situation (depending on the intentions of the application and the reasons for the adaptation). Context-driven methods make a decision when they receive a context change event and after analysing all contextual information. The general loop of an adaptive system includes the observation of the environment, the choice of adaptations and their implementation. We will call it CADA loop which means "Collection, Analysis, Decision and Action" [13]. Adaptive middleware must provide basic mechanisms and services for adaptation. These core services must be highly configurable and dynamically capable of being added/removed. The adaptation decision is generally divided into static, dynamic and artificial intelligence approaches. According to the literature, there are six different approaches which are: logic adapts algorithm-based, logic adapts based on adaptation policies, logic adapts based on planning, logic adapts based on graphs, logic adapts based on artificial intelligence. In Table 3, we present some middleware of adaptation and their logic for adaptation. In literature, we find four cate-

Table 3. Middleware for dynamic and static adaptation

Middleware	Static	Algorithm-based	Policy-based	Planning-based	Graph-based	A.I-based
AxSel					√	
CADeComp						√
CAPUCCINO			√			
Dynaco		√				
GAIA			√			
Kalimucho	√		√			
MUSIC				√		
UbiStar			√			
WComp	√		√			

gories of adaptation: 1) Autonomic adaptation when the application manages the adaptation and adapts itself without external support, 2) Overseen adaptation, which the platform that manages the adaptations and the adaptation is transparent to the applications, 3) Structural adaptation when the structure of the program changes. For component-based systems, this change may relate to composition by adding or deleting components, and 4) Behavioral adaptation is also called functional adaptation. It is about changing the behavior of the program. This means adding or deleting certain features or changing the quality of service. In addition, equivalent services or components may be used to replace certain services or components used [13]. Table 3 shows the different method of adaptation. The middleware

AxSel [20] for example, have a service-based architecture. It is a middleware for graph-based adaptation. An application of AxSel is represented by a two-dimensional graph with flexible dependency where a node represents services and components. On the other hand, the middleware CADeComp [7] Artificial Intelligence based use learning technology. It can be useful in an adaptation decision-making process because of its potential for rich planning and its possibilities for reasoning through learning. CADeComp uses an A*-algorithm, which is used in adaptation system, to reduce the complexity of the component placement problem. In general each middleware uses one of the approaches of adaptation.

4. DISCUSSION

Adaptation systems are a solution for adapting the software to the environment and thus satisfying the constraints of the environment Cloud computing. Adaptation, and in particular real-time and self-adaptation, raises complex scientific problems as well as new challenges for software execution and development. For example, how to collect context information in a highly dynamic, distributed environment? How to adapt the Cloud application and services to achieve good performance in a changing context of things? How can objects exploit heterogeneous infrastructures while taking advantage of the opportunities offered by Cloud of things (CoT) environments? A self-adaptive middleware for Cloud of things must be able to provide appropriate solutions to these questions. There are three essential challenges: Context Managing of objects, Cloud service Recommending and Self-adaptation of Cloud service to context of things.

5. CONCLUSION

Objects of IoT environment integrated in cloud computing to create a new ecosystem called Cloud of things (CoT) raises many issues and challenges. To fix on the heterogeneity of things and their dynamic context, a middleware for self-adaptation of cloud of things environment to variation of object context is the key solution. This new middleware which we want to realized is a hybridation of the two type of middleware which are middleware of context-aware and a middleware for self adaptation. In this paper, we introduce some important terms related to our domain. Then we present some related work. Next, we describe a few of them for dynamic and static adaptation, and we conclude with a discussion and future issue and challenge.

ACKNOWLEDGEMENTS

This work was funded by the “PHC Utique” program of the French Ministry of Foreign Affairs and Ministry of higher education and research and the Tunisian Ministry of higher education and scientific research in the CMCU project number 18G1431.

References

- [1] Aazam, M., Hung, P.P., Huh, E.N., 2014. Smart gateway based communication for cloud of things. In 2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP) , 1–6.
- [2] Aguilar, J., Jerez, M., Exposito, E., Villemur, T., 2015. Carmicloc: context awareness middleware in cloud computing. In 2015 Latin American Computing Conference (CLEI) IEEE , 1–10.
- [3] Aguilar, J., Jerez, M., Rodríguez, T., 2018. Cameonto: Context awareness meta ontology modeling. Applied computing and informatics 14, 202–213.
- [4] Alam, S., Chowdhury, M., Noll, J., 2010. Senaas: An event-driven sensor virtualization approach for internet of things cloud. In 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications IEEE , 1–6.
- [5] Andersson, J., Lemos, R.D., Malek, S., Weyns, D., 2009. Modeling dimensions of self-adaptive software systems. In Software engineering for self-adaptive systems. Springer, Berlin, Heidelberg. , 27–47.
- [6] Atzori, L., Iera, A., Morabito, G., 2010. The internet of things: A survey. Computer networks 54, 2787–2805.
- [7] Ayed, D., Taconet, C., Bernard, G., Berbers, Y., 2008. Cadecomp : Context-aware deployment of component-based applications. Journal of Network and Computer Applications 31, 224–257.
- [8] Badii, A., Crouch, M., Lallah, C., 2010. A context-awareness framework for intelligent networked embedded systems. In 2010 Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services IEEE , 105–110.

- [9] Baraki, H., Schwarzbach, C., Jakob, S., Jahl, A., Geihs, K., 2018. Sam: A semantic-aware middleware for mobile cloud computing. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) IEEE , 375–382.
- [10] Botta, A., Donato, D.W., Persico, V., Pescapé, A., 2016. Integration of cloud computing and internet of things: a survey. *Future generation computer systems* 56, 684–700.
- [11] Byun, H.E., Cheverst, K., 2004. Utilizing context history to provide dynamic adaptations. *Applied Artificial Intelligence* 18, 533–548.
- [12] Cherif, S., Ben Djemaa, R., Amous, I., 2019. Reflective approach to improve self-adaptation of web service compositions with autonomic middleware. *International Journal of Pervasive Computing and Communications* .
- [13] Da, K., 2014. Plateforme d'adaptation autonome contextuelle à base de connaissances. Doctoral dissertation. UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR.
- [14] Diega, G.N.L., 2016. Clouds of things: Data protection and consumer law at the intersection of cloud computing and the internet of things in the united kingdom. *Journal of Law Economic Regulation* 9, 69–93.
- [15] Djamel, F.H.S., Gomes, L.L., Eisenhauer, M., Kelner, J., 2015. A middleware for industry. *Computers in Industry* 71, 58–76.
- [16] Evans, D., 2011. The internet of things: How the next evolution of the internet is changing everything. CISCO white paper , 1–11.
- [17] Farahzadi, A., Shams, P., Rezazadeh, J., Farahbakhsh, R., 2018. Middleware technologies for cloud of things: a survey. *Digital Communications and Networks* 4, 176–188.
- [18] Faye, M.D., 2015. Déploiement auto-adaptatif d'intergiciel sur plate-forme élastique. Doctoral dissertation. Ecole normale supérieure de lyon-ENS. LYON.
- [19] Grati, R., Boukadi, K., Abdallah, H.B., 2017. Une approche pour la prise de décision d'adaptation à base de logique floue pour les saas composites. *Ingenierie des Systemes d'Information* 22, 77.
- [20] Hamida, A.B., Kon, F., Lago, N., Zarras, A., Athanasopoulos, D., Pilios, D., Bouloukakis, G., 2013. Integrated choreos middleware-enabling large-scale, qos-aware adaptive choreographies. *ARLES - Software architectures and distributed systems* .
- [21] Issarny, V., Caporuscio, M., Georgantas, N., 2007. A perspective on the future of middleware-based software engineering. In 2007 Future of Software Engineering, IEEE Computer Society , 244–258.
- [22] Jara, A.J., Olivieri, A.C., Bocchi, Y., Jung, M., Kastner, W., Skarmeta, A.F., 2014. Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. *International Journal of Web and Grid Services* 10, 244–272.
- [23] Ji, Z., Ganchev, I., O'Droma, M., Zhao, L., Zhang, X., 2014. A cloud-based car parking middleware for iot-based smart cities: Design and implementation. *Sensors* 14, 22372–22393.
- [24] Kantarci, B., Mouftah, H.T., 2015. Sensing services in cloud-centric internet of things: A survey, taxonomy and challenges. In 2015 IEEE International Conference on Communication Workshop (ICCW) IEEE , 1865–1870.
- [25] Kephart, J.O., Chess, D.M., 2003. The vision of autonomic computing. *Computer* 1, 41–50.
- [26] Knappmeyer, M., Kiani, S.L., Reetz, S., Baker, N., Tonjes, R., 2013. Survey of context provisioning middleware. *IEEE Communications Surveys Tutorials* 15, 1492–1519.
- [27] Nabli, H., Ben Djemaa, R., Amous, I., 2022. Cloud services description ontology used for service selection. *Service Oriented Computing and Applications* 16, 17–30.
- [28] Parwekar, P., 2011. From internet of things towards cloud of things. In 2011 2nd International Conference on Computer and Communication Technology (ICCCCT-2011) IEEE , 329–333.
- [29] Perera, C., Jayaraman, P.P., Zaslavsky, A., Christen, P., Georgakopoulos, D., 2014. Mosden: An internet of things middleware for resource constrained mobile devices. In 2014 47th Hawaii International Conference on System Sciences IEEE , 1053–1062.
- [30] Perera, C., Zaslavsky, A., Christen, P., Compton, M., Georgakopoulos, D., 2013. Context-aware sensor search, selection and ranking model for internet of things middleware. In 2013 IEEE 14th international conference on mobile data management 1, 314–322.
- [31] Song, Z., Cárdenas, A.A., Masuoka, R., 2010. Semantic middleware for the internet of things. In 2010 Internet of Things (IOT) IEEE , 1–8.
- [32] Tao, F., Cheng, Y., Xu, L.D., Zhang, L., Li, B.H., 2014. Cciot-cmfg: cloud computing and internet of things-based cloud manufacturing service system. *IEEE Transactions on Industrial Informatics* 10, 1435–1442.
- [33] Vinh, T.L., Bouzeffrane, S., Farinone, J.M., Attar, A., Kennedy, B.P., 2015. Middleware to integrate mobile devices, sensors and cloud computing. *Procedia Computer Science* 52, 234–243.
- [34] Wang, X., Zhang, D., Gu, T., Pung, H.K., 2004. Ontology based context modeling and reasoning using owl. In *Percom workshops* 18, 22.
- [35] Yachir, A., 2014. Composition dynamique de services sensibles au contexte dans les systèmes intelligents ambiants. Ph.D. thesis. Université Paris-Est. Paris-Est.
- [36] Ye, J., Dasiopoulou, S., Stevenson, G., Meditskos, G., Kontopoulos, E., Kompatsiaris, I., Dobson, S., 2015. Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing* 23, 1–25.
- [37] Yulia, E., Puttonen, J., Dvoryanchikova, A., Lastra, J.L.M., 2015. Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems. *Engineering Applications of Artificial Intelligence* 43, 111–126.
- [38] Zhang, D., Huang, H., Lai, C.F., Liang, X., Zou, Q., Guo, M., 2013. Survey on context-awareness in ubiquitous media. multimedia tools and applications. *Multimedia tools and applications* 67, 179–211.
- [39] Zhou, J., Leppanen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., Yang, L.T., 2013. Cloudthings: A common architecture for integrating the internet of things with cloud computing. In *Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD)* , 651–657.
- [40] Zorgati, H., Ben Djemaa, R., Amous, I., 2022. Finding internet of things resources: A state-of-the-art study. *Data & Knowledge Engineering* , 102025.