Review Article

Hybrid Aspect of Context-Aware Middleware for Pervasive Smart Environment: A Review

J. Madhusudanan,¹ S. Geetha,² V. Prasanna Venkatesan,² U. Vignesh ,³ and P. Iyappan⁴

¹Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry, India

²Department of Banking Technology, Pondicherry University, Puducherry, India

³Department of Computer Science and Engineering, K L University, Guntur, Andhra Pradesh, India

⁴Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Thirunelveli, Tamil Nadu, India

Correspondence should be addressed to U. Vignesh; vigneshbun@gmail.com

Received 21 September 2017; Accepted 4 February 2018; Published 3 May 2018

Academic Editor: Habib M. Fardoun

Copyright © 2018 J. Madhusudanan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pervasive computing has made almost every device we see today to be communicated and function in collaboration with one another. Since the portable devices have become a part of our everyday life, people are more involved in a pervasive computing environment. They engage with many computational devices simultaneously without knowing the availability of their existence. The current world is being filled with more and more smart environments. These smart environments make them to be attracted towards the new technological emergence in the field of pervasive computing. Various researches are being carried out to improve the smart environment and their applications. Middleware plays a vital role in building the pervasive applications. The pervasive devices act based on the context of the situation, that is, they do their actions according to the environment of the application. They react to the situations smartly as they can take their own decisions based on the context developed for that specific application. Most of the pervasive applications were using its own middleware that is specific towards their need. As today, most of the applications are using their own middleware with their specific requirement, which leads us to unearth out their common features and their scope of using it. In this paper, a survey on the various hybrid aspects of the different context-aware middleware has been done. This middleware is classified based on service, context, and device aspects. Merits and demerits are identified from the existing smart environments, and future perspective of their development such as generic context-aware middleware need has been discussed.

1. Introduction

Pervasive computing is drastically shifting the day-to-day activities by bringing “anytime, anywhere, and anything” computing potential into the living environment. Computers spread across different environments and tend to merge and disappear into day-to-day objects [1]. They are capable of sensing the context of the environment, communicating with different objects and providing the right information to the user. Due to these rich computing powers, more pervasive computing applications are developed, and the need of the pervasive application is sensed today. As the demand of pervasive application grows enormously, more and more research are undertaken in the industry and in academia.

Most of the researches in pervasive computing focus on addressing the specific domain application development, for example, smart home-like aware home [2], UbiHome [3], and ThinkHome [4]. The objectives of these studies are to pattern the user behavior and to provide different services for matching the pattern that supports the “comfort of the user.” From the existing applications, researchers identified the major challenge like “how to handle different devices and different contexts” and also proposed middleware as a solution to it.

2. Evolutions of Pervasive Computing

According to Gabriel et al. [5], before three decades, “computer” is totally referred as mainframe systems which simply

FIGURE 1: Evolution of computing era.

processes input and produces output. This period is referred as the first wave of computing where the user and computer will have less interaction. The first era of computing is termed as “many users-one computer”; that is, one computer is shared by many users. By the rapid growth in the computing vicinity, personal computer came into existence. This period is referred as the second wave of computing, and it is termed as “one user-one computer”; that is, one user uses one computer. Personal computer is very powerful as a standalone system, and it takes part as a node when the network is evolved.

Enormous growth in smart devices/gadgets gave rise to the new computing dreams called ubiquitous computing; this was initiated by Mark Weiser who is called as the father of ubiquitous computing. This period is referred as the third wave of computing, and it is also termed as “one user-many devices”; that is, one user is supported by many devices such as computer, smart phone, and tab as shown in Figure 1.

Ubiquitous computing is a combination of pervasive computing and mobile computing [6]. Making the computing service available in any location context is the primary objective of mobile computing, whereas the primary objective of pervasive computing is to acquire the context information from the environment.

Ultimate focus of mobile computing and pervasive computing is on context-aware computing [7, 8]. So, the term “ubiquitous computing” is also called as pervasive computing, and it can be used either way [9, 10]. Ubiquitous computing is also referred as disappearing computing or everywhere computing or ambient intelligence or invisible computing or internet of things [11].

Pervasive computing has slowly and steadily taken the computing world into a new dimension with its features such as adaptive system, location sensitivity, energy-aware system, and context-aware system.

3. Pervasive Applications

There is a rapid growth in the device world where most of the devices support smartness and the environments are constructed with the intention of supporting smartness. By the integration of smart devices (e.g., microdevices and complex devices) and smart environments (e.g., embedded devices in the physical environment such that a wall can sense camera is recording), a new environment is created that supports any environment and any device, which will work together to produce the pervasive smart environment as shown in Figure 2.

Many applications are developed with the pervasive smart environment in different domains such as

- (i) buildings (smart home-like aware home [2], UbiHome [3], ThinkHome [4], SM4ALL [12], smart office-GreenerBuilding [13], and smart classroom and smart meeting hall [14]),
- (ii) industries [15],
- (iii) healthcare (smart monitoring system, patient tracking system, staff management system, and asset tracking system [16]),
- (iv) transportation [17],

FIGURE 2: Pervasive applications in different domains.

- (v) banking (smart ATM, smart counter, and smart wall [18]),
- (vi) supply chain (smart shelf, smart billing, and tagging of goods [19]),
- (vii) security and IT [20],
- (viii) networking [11].

4. Classifications of Pervasive Architecture

Architecture is the first step in the development of any system; pervasive architecture is classified into three types based on its function, data, and its working as shown in Figure 3 [8].

4.1. Direct Sensor Access Architecture. This architecture was widely used in the earlier stage of pervasive computing when devices were available for direct access. The client software gathers the required information from the available devices without any additional layer to acquire the information from the device. Separate drivers are hardcoded into the specific application. Hence, this can be used in different applications as there will be different drivers for different devices. Therefore, this architecture is well suited for standalone

FIGURE 3: Classification of pervasive architecture.

systems which will directly access the devices as it is capable of handling homogeneous devices.

4.2. Context Server Architecture. Context server architecture supports many client systems to access the remote data server concurrently. Data acquired are stored in the context server, and they are used according to the context of the application and contexts are monitored using different devices. Most of the devices used in context-aware systems are portable gadgets which are limited to power energy, computation power, memory space, and so on. These are vital features for a centralized context server. It will consider different issues like fault tolerance, quality of services, and scalability of dynamic devices.

4.3. Context-Aware Middleware Architecture. Most of the software designs use decomposition as a technique to split the layers, for example, business logic, data logic, and presentation logic. Hence, the middleware is one which can individually take a different logic to work. Context middleware layer is introduced with the intention of hiding the lower level complexities. It has an advantage of reusing the lower level coding (hardware-dependent device coding), and it is flexible to adapt to any change in the application logic.

More studies on pervasive architecture are being conducted in different dimensions such as “interoperability of service,” “interoperability of the device,” “context modeling,” and “building smart environment”. The primary objective of pervasive computing is “context-aware computing” which deals with the different contexts of the application like location context, user context, and time context. These contexts are helpful to provide appropriate service to the user. Context can be defined as “any information used to predict the state of an object, where the object can be user, location, or any physical object.” Today, middleware for context-aware computing is one of the further steps for pervasive computing to move towards veracity.

5. Related Works on Context-Aware Middleware

According to Bandyopadhyay et al. [21], context-aware middleware is essential for three reasons:

- (i) To develop a common method to interact between heterogeneous devices belonging to a different domain.
- (ii) Middleware acts as a relationship to merge different heterogeneous components together.
- (iii) Hiding of complexity can be achieved in different applications with different domains.

Last two decades have been the peak periods in the development of context-aware middleware [22], where most of the middleware are classified into seven categories [23]: agent-based middleware, reflective middleware, metadata-based middleware, tuple space middleware, adaptive middleware, objective middleware, and OSGI-based middleware.

There are several surveys carried out in relation to the context-aware middleware field. The study and analysis of the survey are given in Table 1. Table 1 specifies the functional and nonfunctional characteristics of the existing context-aware middleware based on literature study [7, 24].

As per the result of the survey mentioned in Table 1, the working aspects of middleware can be classified into three levels of aspects, namely, service aspect, context aspect, and device aspect, from the context perspective as shown in Figure 4. Service aspects act as an interface between the application layer and the context aspects. It converts the requirement into service. Context aspects play a vital role in bridging the gap between the service aspects and the device aspects, where context information is derived from the device aspects. Device aspects act as a logical device to share information between context aspects and physical devices.

From the literature survey, it is obvious that the context-aware middleware is growing as an indispensable architecture component in building any pervasive application. From the study mentioned in Table 1, important parameters used for evaluating any context-aware middleware are identified as device management, interoperability, context awareness, security and privacy, autonomy, intelligence, and adaptability. These parameters are used to analyze the context-aware middleware.

A study on the various context-aware middleware is carried out, in order to understand the working of different aspects of the context-aware middleware and to make sure the number of domains that the context-aware middleware supports. Metrics for evaluating the existing context-aware middleware are achieved through the number of domains that the same middleware can support. The “multi” effects (multidomain, multiapplication, and multiculture) are to be addressed at the architecture level [32, 34]. In order to categorize the existing context-aware middleware, the following criteria are proposed:

- (i) If it supports three or more domains, then it is rated as high (H).
- (ii) If it supports two domains, then it is rated as medium (M).
- (iii) If it supports one domain, then it is called as low (L).
- (iv) If it supports no domain, then it is said as not applicable (N).

These metrics are verified on different aspects such as service aspects, context aspects, and device aspects.

5.1. Service Aspects. Service-oriented device middleware and service-oriented context middleware are two different middleware available to provide service aspects in the perspective of device aspects and context aspects, respectively. This leads the researcher to move towards a middleware to manage both device and context functionalities in a particular middleware. Existing middleware are developed to provide service for the specific domain application, and there are no provisions to alter the available service to suit or accommodate to a different domain application.

Considering an example to illustrate how a same service works differently in diverse domain applications, consider the tracking as a service and it is applied in different domain applications, such as in the following:

- (i) In Scenario 1, an application of the transport system, location is tracked using GPS or some other techniques. Here, location is the context, and GPS is a device used to trace the location [33].
- (ii) In Scenario 2, an application of hospital, location of the doctor, patient, and object can be traced using RFID tag. Here, location, users, and objects are context, and RFID acts as a device [35].
- (iii) In Scenario 3, an application of home, location of the user/object can be identified using RFID.

TABLE 1: Contribution towards context-aware middleware.

Authors	Parameters or attributes	Finding/report
Messer et al. [25]	Mobility, adaptability, power management, security, and QOS	How new ideas are converted to everyday project. Its major contribution is towards device aspect.
Kjær [24]	Environment, Storage, reflection, quality, adaption, migration, and composition	No single middleware is appropriate to handle different contexts. Its major contribution is towards service and context aspects.
Romero [26]	Adaptation, communication, context management, service discovery, and persistency	Most of the context-aware middleware are focusing on specific aspect of pervasive computing. Its major contribution is towards context and device aspects.
Lee et al. [27]	Decentralized, autonomy, context awareness, lightness, external sensory capture, intelligence, and web enable	Context is the key for smart environment application development. Its major contribution is towards context and device aspects.
Davidyuk et al. [28]	Specification, composer, interoperability, and end user involvement	Constant monitoring of user activity and representing in the standard format to support the development of middleware, which can act as a common interface. Its contribution is towards context aspect.
Saeed and Waheed [29]	Architecture style, location transparency, aspect oriented, decomposition, fault tolerance, interoperability, service discovery, and adaptability	Most of the research takes place in the fixed domain, and scope for the development of generic middleware is addressed. Its major contribution is towards service aspect.
Bandyopadhyay et al. [21]	Device management, interoperability, platform portability, context awareness, and security and privacy	Most of the middleware focus on specific aspect and need for generic middleware. Its major contribution is towards device and context aspects.
Vasanthi and Wahidabanu [30]	Interoperability, discoverability, location transparency, adaptability, context awareness, scalability, security, and autonomous management	Need to interface the heterogeneity protocol used in heterogeneous networks. Its major contribution is towards device aspect.
Nadia Gámez and Fuentes [31]	Services, implementation, heterogeneity, network technology, network prototype, and new services	Need to support heterogeneous devices to work together and to capture the context change and to adapt to it. Its major contribution is towards service aspect.
Bellavista et al. [7]	Context data model, processing, dissemination, routing overlay, and run-time adaption support	Context-aware application needs to work in different domains. Its major contribution is towards context aspect.

Here, location, users, and objects are considered as context, and RFID belongs to device aspects [36].

- (iv) In Scenario 4, an application of supply chain management, transporting of goods from one location to another can be monitored using RFID and GPS techniques. Here, goods, location, and so on act as a context, and RFID and GPS act as some devices [37].
- (v) In Scenario 5, an application of the security system, movement of users and abnormal activities will be monitored using camera and identification sensors. Here, movement of users and abnormal activities are considered as context, and camera and sensors act as devices [38].

From the above scenarios, it is evident that different tracking services are used for each pervasive application. Even though tracking is a common service, it varies only with the domain-specific features, and also, there is no architecture or framework or middleware to make the service common to all domain applications as a generic service.

In service aspects, there exist five middleware. They are OASIS [39], SAMI [40], SOCAM [41], NEXUS [42], and KASOM [43] as shown in Table 2. Each middleware works

FIGURE 4: Context-aware architecture with aspects.

TABLE 2: List of context-aware middleware on service aspects.

Serial number	Middleware	Device management	Interoperability	Platform Portability	Context awareness	Security and privacy	Autonomy	Intelligence	Adaptability	Domain specific
1	NEXUS	N	M	L	N	N	N	M	N	SOA service discovery
2	SOCAM	M	N	M	M	L	L	L	M	Event-based application
3	OASIS	M	N	M	N	N	M	N	N	SOA middleware for sensor networks
4	SAMI	M	M	N	N	N	M	N	N	Share information among devices
5	KASOM	M	M	L	M	M	M	N	M	SOA-based application

for a specific application like sharing of information among devices in smart home, interface for network service in smart office, and so on. Thus, all the service aspects of context-aware middleware support for a specific domain application.

5.1.1. NEXUS. It finds a new way to merge service-oriented architecture and pervasive computing to create a smart, healthy, and flexible network to access different active service resources [42].

5.1.2. SOCAM. Pervasive application can be developed using service-oriented context-aware middleware platform. It mainly works with two layers. The first layer is used to control and to manage context information, whereas the second layer helps to frame different policies to support various client applications [41].

5.1.3. OASIS. It is object-centric service-oriented middleware to support different sensor devices, and it works according to the need of the service. It also supports automatic discovery of the service and the registry of service [39].

5.1.4. SAMI. It is a self-adaptive information sharing middleware to locate different devices, and it is used for sharing the device information despite its dynamic location. Device information files are grouped as data [40].

5.1.5. KASOM. The knowledge-aware and service-oriented middleware (KASOM) was illustrated as a new idea to combine pervasive computing and service-oriented cloud computing to create a novel era in the computing generation. The major purpose of knowledge-aware and service-oriented middleware is to provide sophisticated and enhanced pervasive services to everyone connected to the smart environment.

From the above discussion, to support the diverse domain applications, the different domain services are to be maintained, and they should be provisioned to discover the suitable service and also to maintain the hierarchy of service [43].

5.2. Context Aspects. Context-aware applications acclimatize according to the context of the user, context of the location, nearby people in the environment, objects, and the reachable devices in the environment, and changes take place to those objects over a period of time. A context-aware application with these potentials will analyze the environment and adapt according to the context of the environment [44].

Context of the application is monitored and processed with the support of smart devices, which acquires context information of the environment or its surrounding. In order to make context acquisition in reality is a demanding task. In order to achieve this, context aspect of the middleware is used, and it undergoes different challenges. They are as follows [45]:

- (i) What are the different contexts that are available in general?

There are different types of context available like the location, user, device, time, and activity [24]. Context cannot be fixed in general, but it can be fixed for a specific application.

- (ii) From where context is discovered?

Context information is taken from the available environment of the application like the location, user, and time and is monitored with the help of smart devices.

- (iii) How to process the context information?

Context is processed generally in four different ways [7]. They are

- (a) context data history,
- (b) context data aggregation through logical reasoning and probabilistic reasoning,
- (c) context data filtering using both time-based filtering and change-based filtering,
- (d) context data security.

- (iv) How context is represented?

Context information is represented in any one of the five different models [7]. They are

- (a) key-value model,
- (b) markup schema model,
- (c) object-oriented model,
- (d) logic-based model,
- (e) ontology-based model.

- (v) How the conflict in the context is handled?

Addressing two contexts of the application at the same time is a crucial decision to be made. It can be addressed in different ways [46]:

- (a) Conflict resolution Policy
- (b) QOC parameter
- (c) Priority to the context

- (vi) How same contexts differ from one domain to another domain?

Same contexts will change from one domain application to another. For example, consider a location context which changes according to the domain. In smart hospital application, location contexts will be the doctor's room, ICU, general ward, and so on. In smart bank application, location contexts will refer to the manager's room, locker room, counter, and so on. In smart home application, location contexts will include bedroom, kitchen, hall, and so on.

In the context aspects, there exist twenty-four middleware, to mention few such as AURA [47], CARMEN [48], and CARISMA [49] as shown in Table 3. Here, different context-aware middleware work in different domains like smart home, smart hospital, and smart office. Each context-aware middleware works for a specific application.

TABLE 3: List of context-aware middleware on context aspects.

Serial number	Middleware	Device management	Interoperability	Platform Portability	Context awareness	Security and privacy	Autonomy	Intelligence	Adaptability	Domain specific
1	AURA	M	N	N	M	M	M	N	M	Support context-aware application
2	GATA	M	M	M	M	M	L	L	M	Metaoperating system
3	CARISMA	N	H	N	H	N	H	N	H	Mobile application
4	COOLTOWN	M	H	N	H	N	H	N	N	Interfacing devices with web-enabled environment
5	COBRA	N	M	M	M	N	L	M	N	General middleware
6	TOTA	N	L	L	H	N	N	N	N	Representation of context information
7	IADABS	M	M	M	M	L	N	N	M	Mobile application
8	COMPACT	M	N	N	M	N	N	N	N	Managing context-aware data
9	MARKS	M	N	N	M	L	N	N	L	Mobile application
10	CARMEN	M	N	N	M	N	M	M	M	Mobile agent for context application
11	CORTEX	M	M	N	M	M	M	L	M	Context-aware home application
12	MIDDLEWHERE	N	N	N	H	H	N	N	N	Home-based application
13	MOBILPADS	L	L	L	L	N	L	N	L	Mobile application
14	MIPEG	M	M	N	M	N	L	N	L	Grid application
15	CAMPS	M	M	M	M	N	N	L	N	Smart classroom application
16	AMiCA	M	N	N	M	L	M	N	N	Hospital application
17	IBICOOP	L	L	L	L	L	N	N	N	Home server application
18	CDTOM	M	M	N	M	N	L	N	N	Elder home automation
19	UBIROAD	M	M	M	M	M	L	N	L	Context-aware application
20	UMOVE	M	M	N	M	N	M	N	M	Mobile application
21	SMEPP	M	N	M	M	M	N	N	M	Context management for context-aware application
22	CAMEO	N	N	N	H	N	H	H	H	Mobile-based application
23	AWARE	N	N	N	H	N	H	H	H	Mobile-based application

5.2.1. AURA. AURA is a context-aware middleware to design and develop a system to work as “personal information atmosphere” regardless of their location, and it acquires support of wearable computing, handheld device, sensor, desktop computers, users, and infrastructure computers [47].

5.2.2. GAIA. GAIA is a distributed middleware system and acts as a metaoperating system to coordinate the different types of devices in the pervasive environment. As an operating system, it allocates resources, manages different file systems, and establishes communication with different resources [50].

5.2.3. CARISMA. CARISMA uses policies to work under dynamic context of the environment, and it adapts to the need of the application. Conflicts in the policies are solved during run time [49].

5.2.4. COOLTOWN. COOLTOWN middleware is used to connect the web resources by means of physical objects, and also, it starts interaction with different gadgets to share information according to the context of the environment [51].

5.2.5. COBRA. COBRA is an agent-based context-aware middleware to support context-aware application in the pervasive environment. It is the responsibility of the agent to collect the context information from the devices, environments, and other agents [52].

5.2.6. TOTA. TOTA (“Tuples on the Air”) is a new middleware for supporting the programming model of a application, which adapts to the context-aware information received from the pervasive computing environment. It works only for the specific application domain [53].

5.2.7. JADABS. JADABS middleware works in the dynamic mobile environment and to adapt the user behaviors. It is built to cope up with changing devices in different environments. It also stores the context information of the environment [54].

5.2.8. COMPACT. COMPACT middleware is mainly for context representation which is an important aspect of pervasive computing systems. In this middleware, data are retrieved regularly from the environment and processed. The processed data are stored as dynamic context information for future use (Maria Strimpakou et al., 2006).

5.2.9. MARKS. MARKS middleware provides information on how to use the knowledge inferred from the users in the given system, how to discover the resources available in an environment, and how to handle a resource failure and provide solution for it (Moushumi Sharmin et al., 2006).

5.2.10. CARMEN. CARMEN is a middleware for handling context-aware resources in a wireless network, and it supports

to create resource metadata. Metadata help to find the dynamic context of the application and work according to it [48].

5.2.11. CORTEX. CORTEX middleware is based on sentient objects, which sense and observe the behavior of its surrounding environment and objects. From the observation, it generates the context information and works according to it [26].

5.2.12. MIDDLEWHERE. MIDDLEWHERE is a context-aware middleware which gives the location information well in advance to applications, and it integrates a different range of location-sensing methods. A separate reasoning engine determines the context of the environment from the location information [26].

5.2.13. MOBILPADS. MOBILPADS is a context-aware middleware for mobile environment. It is a service entity, which can migrate between different mobile entities. It works in the principle of the client-server system, and it observes the changes in the entities to keep the context information updated [26].

5.2.14. MIPEG. MIPEG is a middleware for pervasive grid applications, and it supports for integration of various mobile devices at anytime. It extracts the context information from the devices and executes the task according to the user task [55].

5.2.15. CAMPS. CAMPS is an agent-based middleware to support context-aware service in the smart environment. Its primary objective is creating a standard to represent the context information and the method to process it [56].

5.2.16. AMiCA. Ambient middleware for context awareness is used to continuously and implicitly adapt to the environment to meet evolving user expectations. Up-to-date valid context information is the key requirement for successful transparent interaction between the smart environment and the middleware [27].

5.2.17. IBICOOP. IBICOOP middleware is developed to support many challenges like abstraction of device configuration, connectivity of devices, and heterogeneity of devices [57].

5.2.18. CDTOM. Context-driven task-oriented middleware focuses on the user’s work or task in different situations, rather than focusing on the various devices and services in the environment. It also makes the system to work intelligently as per the user request [58].

5.2.19. UBIROAD. UBIROAD is a middleware developed to handle the smart road environment. Here, context information is acquired from the vehicle and road in which the

vehicle is travelling (road condition, traffic, another vehicle detail, etc.) to create a smart environment for the user [21].

5.2.20. UMOVE. UMOVE middleware will consider every device as an entity, and also, it works based on the entity-layered architecture. It collects the information from the entity as entity data and processes the context of the entity data according to the application [59].

5.2.21. SMEPP. SMEPP is a middleware to support security in the pervasive environment. It creates various security policies to access the context information which is acquired from the smart environment. According to the policy, middleware ensures the system is secured [21].

5.2.22. Cameo. CAMEO middleware works for mobile devices and its context. It collects the context from the environment and analyze the context information. It provides a common programming interface to connect the smart environment. CAMEO middleware architecture also supports nonfunctional qualities like integrability, reusability, and ease of creation [60].

5.2.23. AWARE. AWARE middleware is developed to collect the context from the environment. Accumulated context is managed and shared. It supports to present the context information to the stakeholders of the application. AWARE middleware architecture also supports nonfunctional qualities like scalability, integrability, reusability, and ease of creation.

Most of the context-aware middleware work for the context representation such as the key-value model, logic-based model and ontology-based model, context acquisition, and context processing such as logical reasoning and probabilistic reasoning. There is no middleware to support different domain applications in terms of all the aspects.

The main objectives of the existing context-aware middleware are to investigate how contexts are acquired, processed, and modeled for a specific domain application. From the existing context aspect middleware, it is evident that there is no generic context-aware middleware to support context aspects. Therefore, the need for the generic context-aware middleware is reported [61].

5.3. Device Aspects. Currently, middleware is available to support specifically the device aspects of the domain which help in device management of a specific application, and in few middleware, generic device aspects are also supported. Hydra [62] and SenseWrap [63] are the middleware to support diverse domain applications, and both these middleware concentrate on

- (i) how to interface the heterogeneous physical device with the application software;
- (ii) how to handle the heterogeneous network protocol;
- (iii) device management.

It provides a device platform for context-aware middleware; despite this, it does not support context aspects. It uses virtual entity to map the physical devices. The virtual entity interfaces the physical devices to maintain the device properties. Any changes observed in the physical devices will be reflected in the virtual entity. These mappings of the virtual entity help to achieve abstraction in the device management aspect.

Scope for interfacing the context aspect with the device aspect is shown in the study of Lowe. R et al. (2013). Both device aspects and context aspects combine together to provide a domain service to the user.

In device aspects, there exist twenty middleware such as HYDRA [21], MADAM [64], and PERLA [65], as shown in Table 4. Each middleware works for a specific application like interfacing the different devices in the specific environment of a domain.

Different middleware work in different domains like smart home, smart hospital, and smart office. Device aspect acts as a bridge between the physical device and context aspect of the system. It manages the devices in the smart environment to coordinate and act as a group to solve the different tasks of the application.

5.3.1. PICO. PICO is developed to hide the complexity of integrating the heterogeneous devices of the smart environment. Its objective is to provide device-integrated service to any place at anytime. It is developed to support hospital-based application [66].

5.3.2. HOMEROS. HOMEROS middleware coordinates different devices with different properties in the pervasive smart environment. It provides the right service at right time to facilitate the context-aware application [67].

5.3.3. M-ECHO. M-ECHO is used to support performance of the system even when available devices are connected through a weak network. It also adapts the morphing step to tune the peer-to-peer performance of the system and focuses on the data that are acquired [68].

5.3.4. MADAM. MADAM mainly works to address the two issues such as to support mobile clients and heterogeneous devices. It also provides support to heterogeneity for both the client and the server. In the client, different context structures are framed to support different users, and in the server, specific services are created to support different requests [64].

5.3.5. GAS-OS. GAS-OS is a middleware to separate the application logic from the complex pervasive environment by applying abstraction on the working of the heterogeneous device and environment [69].

5.3.6. GSN. GSN is a middleware to provide virtual devices as a simple and high-level abstraction. It uses an XML

TABLE 4: List of context-aware middleware on device aspects.

Serial number	Middleware	Device management	Interoperability	Platform Portability	Context awareness	Security and privacy	Autonomy	Intelligence	Adaptability	Domain specific
1	PICO	M	M	M	L	L	L	L	M	Hospital-based service application
2	HOMEROS	H	H	H	L	L	N	N	H	Mobile application
3	M-ECHO	M	M	N	M	N	M	M	N	ROBOT application
4	GAS-OS	M	M	M	N	N	M	N	L	Interface the device
5	MADAM	M	M	N	N	N	N	N	L	Mobile application
6	GSN	H	N	H	N	M	L	N	L	Device management
7	SATIN	M	M	M	N	N	N	N	N	Mobile application
8	MUNDOCORE	L	N	L	L	N	N	N	N	Communication middleware
9	SAMI	H	H	N	N	N	H	N	N	Share information among devices
10	ATLAS	M	M	M	N	N	L	N	L	Device management
11	SANDMAN	H	N	N	N	N	N	N	N	Device management
12	PANOPLY	H	H	N	N	M	N	N	N	Communication interface
13	MISSA	M	N	N	N	N	M	N	L	Stream of data from devices
14	UBISOAP	M	M	M	N	N	N	N	N	Communication middleware
15	ISMB	M	N	M	N	N	M	N	M	RFID-based application
16	HYDRA	H	H	H	N	N	N	N	H	Hospital application in the device level
17	SIRENA	M	M	M	N	L	L	N	L	Device management
18	ASPIRE	L	N	L	N	N	L	N	L	RFID-based application
19	PERLA	H	H	H	N	N	M	N	M	Communication interface
20	PACES	M	M	M	L	L	L	L	L	Smart home application

representation to represent the various devices and is also used to connect the devices in the smart environment [70].

5.3.7. SATIN. SATIN is a component-based middleware developed to support various mobile computing devices to handle the dynamic scenario in the smart environment [71].

5.3.8. MUNDOCORE. MUNDOCORE is designed to support the communication between the networks and devices. It provides the platform-independent environment to support heterogeneous devices [72].

5.3.9. SAMI. SAMI is a self-adaptive information sharing middleware which works in a dynamic pervasive computing environment. It is used to share the information among the devices regardless of their network and environment [40].

5.3.10. ATLAS. ATLAS is a sensor and actuator middleware platform to support integration of devices in the pervasive smart environment. Its main challenge is to handle heterogeneous devices and make them to work together to support different pervasive applications [73].

5.3.11. SANDMAN. SANDMAN middleware is developed to reduce the energy used in the smart environment by means of using the efficient protocol stack for communication. It also deactivates the unused devices to sleep mode which helps to reduce the power consumption [74].

5.3.12. PANOPLY. The main objective of PANOPLY is to group the devices in the smart environment to solve the particular task according to the context of the pervasive smart environments [75].

5.3.13. MISSA. The primary objective of MISSA is to separate the service logic from the data stream. Data stream is given more importance for the collection of context information from various devices and the proper standard for processing it [76].

5.3.14. UBISOAP. UBISOAP middleware supports ubiquitous (anytime services) device services in the pervasive smart environment. It is achieved only by making the devices as a service provider and a service receiver [77].

5.3.15. ISMB. ISMB is a middleware to support communication between devices and networks that can maintain one-to-one, one-to-many, and many-to-many relationships. It automatically discovers the device in the environment and also information maintenance of the mobile device [21].

5.3.16. HYDRA. HYDRA middleware supports its application by allowing heterogeneous physical devices to flexibly adapt

into their environments. It manages the devices in the environment and considers each device as a separate entity [21].

5.3.17. SIRENA. SIRENA is mainly developed for the integration of smart devices in different domain applications. Devices are seamlessly connected in the smart environment. It is demonstrated with two domains, namely, industries and smart homes [21].

5.3.18. ASPIRE. ASPIRE is a middleware to support tracking application along with many RFID devices and tools. It helps to track the number of goods in the environment [21].

5.3.19. PERLA. PERLA is developed to provide support for managing and controlling the heterogeneous smart devices in the pervasive environment. It has its own language to describe the specification of the smart devices used in the application [65].

5.3.20. PECES. Pervasive Computing in Embedded Systems (PECES) project is developed to create a framework to combine different devices of the smart environment to cooperate and work together to solve a particular task. It is demonstrated with the help of smart home application.

Most of the context-aware middleware work for the device management, and establishes communication between the various devices and middleware, and its focus is on device aspects. There is no middleware to support different domain applications in terms of all the aspects [78].

Device aspects provide support to the physical layer and start with how to connect different devices without any problem and to maintain an active observer/listener to monitor the device and its changing information/data. Device data/information will be represented in a common way and stored in a common format for generic sharing of information. Devices are categorized as a group in order to interact with each group and solve a task effectively and smartly.

In order to support device aspects, a separate list of live devices is to be maintained, to retrieve and control the device information. The live list will give the exact number of devices working in the environment along with its properties. Different devices are grouped to achieve a common goal, and also, it should be ready to handle many requirements at the same time.

6. Existing Tool-Based Development Efforts

In order to test the working of the application model/architecture, a tool-based method is needed for the developer. The development tool acts as a test bed for most of the application developers. In recent times, pervasive application developers have started using various pervasive developer tools for their applications. The Context Toolkit [79] model is made up of different context widgets, and it is placed in a distributed environment. These context widgets are used to access the context information and also hide the context-

sensing procedure. Context widgets are software components that are used for developing pervasive application. This model deals only with the context-sensing aspect.

The tool Olympus [80] is to specify the devices, locations, and user in a high-level description form which in turn is converted into actual active space components that are used to build programs. Since this model specifies the context in a high-level description, it is difficult for the developer to understand. It is domain specific, so there are more difficulties in defining the specification of the application.

Archface is a tool [81] that acts as an interface between the architecture and methods and to implement the application. It has its own architecture description language to describe the specification of application, and also, it has programming level interface to convert the specifications into the implementation. The major drawback of the tool is that it deals only with architecture perspective and not with the various components that are used to develop the application.

PervMl [82] is a domain-specific language tool used to define the requirement specification in the metamodel design. This metamodel is in turn mapped to convert it into program that hides the high-level complexities involved in the development of the pervasive application. This model defines the specifications of the tool in a descriptive form, which makes the developer difficult for defining the specifications without predefined knowledge about the domain.

Diasuite [83] is a tool that allows the designer to define the taxonomy of specific application environment by using the existing application model. It has its own simulation tool to simulate the working of the application with the pre-written specification of the application. These tools are taken as reference for the tool-based development.

7. Limitations of the Existing Work

From the observation of the literature survey, context-aware middleware is working on three different aspects as shown in Figure 5. The first aspect of the context-aware middleware concerns on domain on which it is operating, such as smart home, smart office, smart class, and smart hospital. The second aspect of the context-aware middleware focuses on the various devices operating on the pervasive smart environment and devices involved are dynamic or fixed to what extend is the great challenge.

The third aspect of the context-aware middleware deals with the context of the diverse domain environment. Context predicts the situation of the pervasive smart environment and adapts according to it. The different contexts are the location, time, device, operation, and so on, and the main objective is to assess how many contexts can the application support.

From the study, it may be observed that there are mainly three reasons to state middleware is specific to the application:

- (1) Context-aware middleware architecture is specific to the pervasive applications. There is no generic context-aware middleware architecture to support diverse domain applications.

FIGURE 5: Different aspects of context-aware middleware.

- (2) Existing context-aware middleware implicitly states the limited nonfunctional qualities. From the literature survey, existing context-aware middleware architecture exhibits the limited number of nonfunctional qualities, and also, they are implicitly stated.
- (3) Context-aware middleware is biased to the specific aspects. Most of the existing context-aware middleware are biased of specific aspects like service aspect alone or only to context aspect or particularly to the device aspect.

8. Conclusions

The evolution of pervasive computing is presented. Various pervasive applications are obtained from the literature and classified according to their domain. The different architectures for pervasive computing are analyzed, and it is found that context-aware middleware architecture provides a better solution to build pervasive application. In order to better understand the demands of stakeholders of context-aware middleware, a set of different aspects of context-aware middleware are derived from the literature and presented.

Context-aware middleware is classified based on three aspects such as context aspect, device aspect, and service aspect. Classification of the context-aware middleware has been given to obtain a clear understanding about the requirements of the different aspects. On analyzing the requirements of the context-aware middleware, the limitations are obtained, and the scope for the development of generic context-aware middleware is reported.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

- [2] C. D. Kidd, R. J. Orr, G. D. Abowd et al., "The aware home: a living laboratory for ubiquitous computing research," in *CoBuild'99, Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, LNCS, vol. 1670, pp. 191–198, Springer-Verlag, ISBN: 978-3-540-665960, October 1999.
- [3] Y. Oh, S. Jang, and W. Woo, "User centered context-aware smart home applications," *Journal on Korea Information Science Society (KISS): Software and Applications*, vol. 31, no. 2, pp. 111–125, 2004.
- [4] C. Reinisch, M. J. Kofler, and W. Kastner, "ThinkHome: a smart home as digital ecosystem," in *Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pp. 256–261, Dubai, UAE, April 2010.
- [5] P. Gabriel, M. Bovenschulte, E. Hartmann et al., *Pervasive Computing: Trends and Impacts*, BSI Godesberger Allee, Bonn, Germany, 2006, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Percenta/Percenta_elay_pdf.pdf?__blob=publicationFile&v=4.
- [6] K. Lyytinen and Y. Yoo, "The next wave of nomadic computing: a research agenda for information systems research," *Information Systems Research*, vol. 13, no. 4, pp. 377–388, 2002.
- [7] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Computing Surveys*, vol. 44, no. 4, pp. 1–45, 2013.
- [8] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.
- [9] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10–17, 2001.
- [10] R. Zhao and J. Wang, "Visualizing the research on pervasive and ubiquitous computing," *Scientometrics*, vol. 86, no. 3, pp. 593–612, 2011.
- [11] D. Cook, J. Augusto, and V. Jakkula, "Ambient intelligence: technologies, applications and opportunities," *Journal of Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.
- [12] I. Georgievski, V. Degeler, G. Andrea Pagani, T. Anh Nguyen, A. Lazovik, and M. Aiello, *Optimizing Offices for the Smart Grid*, Distributed Systems Group Institute for Mathematics and Computer Science, University of Groningen, Groningen, Netherlands, 2011.
- [13] V. Degeler and A. Lazovik, "Architecture pattern for context-aware smart environments: creating personal, social, and urban awareness through pervasive computing," *Advances in Human and Social Aspects of Technology*, pp. 108–130, ISBN: 978-1-4666-4695-7, 2014.
- [14] S. S. Yau, S. K. S. Gupta, F. Karim, S. I. Ahamed, Y. Wang, and B. Wang, "Smart classroom: enhancing collaborative learning using pervasive computing technology," in *Proceedings of the ASEE 2003 Annual Conference and Exposition*, Nashville, TN, USA, June 2003.
- [15] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Journal of Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [16] Y.-W. Wang, H.-L. Yu, and Y. Li, "Internet of things technology applied in medical information," in *Proceedings of the International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 430–433, Xianning, China, April 2011.
- [17] Y. Chen, J. Guo, and X. Hu, "The research of internet of things' supporting technologies which face the logistics industry," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS)*, pp. 659–663, ISBN: 978-1-4244-9114-8, Shanghai, China, December 2010.
- [18] G. Chong, L. Zhihaoand, and Y. Yifeng, "The research and implement of smart home system based on internet of things," in *Proceedings of the International Conference on Electronics, Communications and Control (ICECC)*, pp. 2944–2947, ISBN: 978-1-4577-0320-1, Ningbo, China, September 2011.
- [19] L. W. F. Chaves and C. Decker, "A survey on organic smart labels for the internet-of-things," in *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS)*, pp. 161–164, E-ISBN: 978-1-4244-7910-8, Kassel, Germany, June 2010.
- [20] K. Markantonakis and K. Mayes, *Secure Smart Embedded Devices, Platforms and Applications*, Springer Science and Business Media, Berlin, Germany, 2014.
- [21] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "A survey of middleware for internet of things," in *Communications in Computer and Information Science*, vol. 162, pp. 288–296, Springer, Berlin, Germany, 2011.
- [22] L. Chen, J. Hoey, C. Nugent, D. Cook, and Z. Yu, "Sensor-based activity recognition: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 6, pp. 790–808, 2012.
- [23] H.-L. Truong and S. Dustdar, "A survey on context-aware web service systems," *International Journal of Web Information Systems*, vol. 5, no. 1, pp. 5–31, 2009.
- [24] K. E. Kjær, "A survey of context-aware middleware," in *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*, pp. 148–155, Innsbruck, Austria, February 2007.
- [25] A. Messer, H. Song, P. Kumar, P. Nguyen, A. Kunjithapatham, and M. Sheshagiri, "InterPlay: A Middleware for Integration of Devices, Services and Contents in the Home Networking Environment," in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC 06)*, Las Vegas, NV, USA, 7–10 January 2006, pp. 1083–1087.
- [26] D. Romero, "Context-aware middleware: an overview," *Revista Electrónica Paradigma en Construcción de Software*, vol. 3, pp. 1–11, 2008.
- [27] K. Lee, T. Lunney, K. Curran, and J. Santos, "Ambient middleware for context-awareness (AMiCA)," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 1, no. 3, pp. 66–78, 2008.
- [28] O. Davidyuk, N. Georgantas, V. Issarny, and J. Riecki, "MEDUSA: middleware for end-user composition of ubiquitous applications," in *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*, pp. 197–219, IGI Global, Hershey, PA, USA, 2009.
- [29] A. Saeed and T. Waheed, "An Extensive Survey of Context-Aware Middleware Architectures," *American Journal of Computer Architecture*, vol. 1, no. 3, pp. 51–56, 2012.
- [30] R. Vasanthi and R. S. D. Wahidabanu, "A middleware platform for pervasive environment," *International Journal of Computer Science and Information Security*, vol. 9, no. 4, 2011.
- [31] N. Gámez and L. Fuentes, "FamiWare: a family of event-based middleware for ambient intelligence," *Personal and Ubiquitous Computing*, vol. 15, no. 4, pp. 329–339, 2011.
- [32] R. Cloutier, D. Verma, M. Bone, and K. Sommer, "System architecture entropy," in *Proceedings of the INCOSE International Symposium*, Singapore, July 2009.
- [33] L. Kovács, P. Mátételki, B. Pataki, and M. Sztaki, "Service oriented context-aware framework," *Young Researchers Workshop on Service-Oriented Computing*, pp. 15–26, 2009.

- [34] J. M. Neighbors, *Software Construction Using Components*, Ph.D. thesis, ICS-TR-160, University of California at Irvine, Irvine, CA, USA, 1980.
- [35] L. Catarinucci, R. Colella, A. Esposito, L. Tarricone, and M. Z. appatore, "RFID sensor-tags feeding a context-aware rule-based healthcare monitoring system," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3435–3449, 2011.
- [36] E. U. Warriach, "State of the art: embedded middleware platform for a smart home," *International Journal of Smart Home*, vol. 7, no. 6, pp. 275–294, 2013.
- [37] A. Kalmar and R. Vida, "Extracting high level context information using hierarchical temporal memory," in *Proceedings of Conference of Advances in Wireless Sensor Networks*, pp. 27–34, ISBN: 978-963-318-356-4, Qingdao, China, October 2013.
- [38] A. Ema and Y. Fujigaki, "How far can child surveillance go? Assessing the parental perceptions of an RFID child monitoring system in Japan," *Surveillance & Society*, ISSN: 1477-7487, pp. 132–148, 2011.
- [39] I. Amundson, M. Kushwaha, X. Koutsoukos, S. Neema, and J. Sztipanovits, "OASIS: a service-oriented middleware for pervasive ambient-aware sensor networks," *Pervasive and Mobile Computing Journal on Middleware for Pervasive Computing*, 2006.
- [40] A. Negash, V.-M. Scuturici, and L. Brunie, "SAMi: a self-adaptive information sharing middleware for a dynamic pervasive computing environment," in *Proceedings of the IADIS International Conference Wireless Applications and Computing*, Lisbon, Portugal, July 2007.
- [41] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1–18, 2005.
- [42] N. Kaveh and R. G. Hercock, "NEXUS-resilient intelligent middleware," *BT Technology Journal*, vol. 22, no. 3, pp. 209–215, 2004.
- [43] I. Corredor, J. F. Martínez, M. S. Familiar, and L. López, "Knowledge-aware and service-oriented middleware for deploying pervasive services," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 562–576, 2011.
- [44] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA, December 1994.
- [45] S. Presecan, *Pervasive Computing: Collaborative Architecture Applied in Business Environment*, Ph.D. thesis, Babeş-Bolyai University, Cluj-Napoca, Romania, 2011.
- [46] A. Manzoor, H. L. Truong, and S. Dustdar, "Using quality of context to resolve conflicts in context-aware systems," in *Proceedings of the 1st International Conference on Quality of Context (QuaCon'09)*, vol. 5786, pp. 144–155, Stuttgart, Germany, June 2009.
- [47] D. Garlan and J. P. Sousa, "Aura: an architectural frame work for user mobility in ubiquitous computing environments," in *Proceedings of the IFIP 17th world computer congress—TC2Stream/3rd IEEE/IFIP conference on software architecture: system Design, Development and Maintenance*, pp. 29–43, Montréal, QC, Canada, August 2002.
- [48] A. Toninelli, *Semantic-Based Middleware Solutions to Support Context-Aware Service Provisioning in Pervasive Environments*, Dissertation thesis, 2008.
- [49] L. Capra, W. Emmerich, and C. Mascolo, "CARISMA: context-aware reflective middleware system for mobile applications," *IEEE Transactions on Software Engineering*, vol. 29, no. 10, pp. 929–945, ISSN: 0098-5589, 2003.
- [50] M. Román, C. Hess, R. Cerqueira, A. Ranganat, R. H. Campbell, and K. Nahrstedt, *Gaia: A Middleware Infrastructure to Enable Active Spaces*, National Science Foundation, grant NSF 98-70736, NSF 9970139, and NSF Infrastructure grant NSF EIA 99-72884, 2002.
- [51] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *Computer*, vol. 36, no. 3, pp. 25–31, 2003.
- [52] H. Chen, T. Finin, and A. Joshi, "An intelligent broker for context-aware system," in *Proceedings of the Adjunct Proceeding of Ubicomp'03*, pp. 183–184, Seattle, WA, USA, 2003.
- [53] M. Mamei and F. Zambonelli, "Programming pervasive and mobile computing applications with the TOTA middleware," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications (PERCOM'04)*, pp. 263–273, Orlando, FL, USA, March 2004.
- [54] A. R. Frei, *Jadabs—An Adaptive Pervasive Middleware Architecture*, Doctoral dissertation, Swiss Federal Institute of Technology, Zurich, Switzerland, 2005.
- [55] A. Coronato and G. De Pietro, "MiPeG: a middleware infrastructure for pervasive grids," *Journal of Future Generation Computer Systems*, vol. 24, no. 1, pp. 17–29, 2008.
- [56] N. Sahli, "Survey: agent-based middlewares for context awareness," *Electronic Communications of the EASST*, vol. 11, 2008.
- [57] A. Bennaceur, P. Singh, P.-G. Raverdy, and V. Issarny, "The iBICOOP middleware: enablers and services for emerging pervasive computing environments," in *Proceedings of the IEEE Middleware Support for Pervasive Computing Workshop, PerWare*, pp. 1–6, , October 2009.
- [58] H. Ni, B. Abdulrazak, D. Zhang, and S. Wu, "CDTOM: a context-driven task-oriented middleware for pervasive homecare environment," *International Journal of UbiComp*, vol. 2, no. 1, 2011.
- [59] P. Bruegger, *uMove: A Wholistic Framework to Design and Implement Ubiquitous Computing Systems Supporting User's Activity and Situation*, Ph.D. thesis, University of Fribourg, Fribourg, Switzerland, 2011.
- [60] V. Arnaboldi, M. Conti, and F. Delmastro, "CAMEO: a novel context-aware middleware for opportunistic mobile social networks," *Journal of Pervasive and Mobile Computing*, vol. 11, pp. 148–167, 2013.
- [61] D. Ferreira, *AWARE: A Mobile Context Instrumentation Middleware to Collaboratively Understand Human Behavior*, Doctoral dissertation, University of Oulu Graduate School, University of Oulu, Faculty of Technology, Department of Computer Science and Engineering, ISBN: 978-952-62-0189-4, 2013.
- [62] M. Eisenhauer, P. Rosengren, and P. Antolin, "A development platform for integrating wireless devices and sensors into ambient intelligence systems," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, pp. 1–3, Rome, Italy, June 2009.
- [63] P. Evensen, *Event Processing Applied to Streams of TV Channel Zaps and Sensor Middleware with Virtualization*, Ph.D. dissertation, University of Stavanger, Stavanger, Norway, 2013.
- [64] X. Zhi, L. Rong, and W. Tong, "Adaptive middleware for uniform access to legacy systems for mobile clients," in *Proceedings of the International Symposium on Pervasive Computing and Applications*, pp. 46–50, Xinjiang, China, August 2006.
- [65] F. A. Schreiber, R. Camplani, M. Fortunato, M. Marelli, and G. Rota, "PerLa: a language and middleware architecture for

- data management and integration in pervasive information systems,” *IEEE Transactions on Software Engineering*, vol. 38, no. 2, pp. 478–496, ISSN:0098-5589, 2012.
- [66] M. Kumar, B. A. Shirazi, S. K. Das, B. Y. Sung, and D. Levine, “PICO: a middleware framework for pervasive computing,” *IEEE Transactions on Pervasive Computing*, vol. 2, no. 3, pp. 72–79, 2003.
- [67] S. W. Han, Y. B. Yoon, H. Y. Youn, and W.-D. Cho, “A new middleware architecture for ubiquitous computing environment,” in *Proceedings of the Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS’04)*, pp. 117–121, Vienna, Austria, May 2004.
- [68] H. Raj, K. Schwan, and R. Nathuji, “M-Echo: a middleware for morphable data-streaming in pervasive systems,” in *EESR ’05: Workshop on End-to-End, Sense-and-Respond Systems, Applications, and Services*, pp. 13–18, ISBN: 1-931971-32-3, Berkeley, CA, USA, 2005.
- [69] N. I. Drossos, C. A. Goumopoulos, and A. D. Kameas, “A conceptual model and the supporting middleware for composing ubiquitous computing applications,” *Journal of Ubiquitous Computing and Intelligence*, vol. 1, no. 2, pp. 1–13, 2006.
- [70] K. Aberer, M. Hauswirth, and A. Salehi, *Middleware Support for the Internet of Things*, Universitt Stuttgart, Stuttgart, Germany, GI/ITG KuVS Fachgesprch “Drahtlose Sensornetze”, 2006.
- [71] S. Zachariadis, C. Mascolo, and W. Emmerich, “The SATIN component system—a metamodel for engineering adaptable mobile systems,” *IEEE Transactions on Software Engineering*, vol. 32, no. 10, pp. 910–927, 2006.
- [72] E. Aitenbichler, J. Kangasharju, and M. Muhlhauser, “MundoCore: a light-weight infrastructure for pervasive computing,” *Journal of Pervasive and Mobile Computing*, vol. 3, no. 4, 2007.
- [73] A. Helal, H. Yang, J. King, and R. Bose, *Atlas—Architecture for Sensor Network Based Intelligent Environments*, ACM, New York City, USA, 1073-0516/01/03 00-0034, 2007.
- [74] G. Schiele, M. Handte, and C. Becker, “SANDMAN: an energy-efficient middleware for pervasive computing,” in *PERCOM ’08-Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 504–508, Washington, DC, USA, 2008.
- [75] K. Eustice, V. Ramakrishna, N. Nguyen, and P. Reiher, “The smart party: a personalized location-aware multimedia experience,” in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 873–877, Las Vegas, Nevada, USA, January 2008, E-ISBN: 978-1-4244-1457-4, 2008.
- [76] S. Kang, Y. Lee, S. Ihm, S. Park, S.-M. Kim, and J. Song, “Design and implementation of a middleware for development and provision of stream-based services,” in *Proceedings of the IEEE 34th Annual Conference on Computer Software and Applications*, pp. 92–100, Seoul, Republic of Korea, July 2010.
- [77] M. Caporuscio, P.-G. Raverdy, and V. Issarny, “ubiSOAP: a service oriented middleware for ubiquitous networking,” *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 86–98, 2010.
- [78] R. Zhao and N. Speirs, “Development tools for context aware and secure Pervasive Computing in Embedded Systems (PECES) middleware,” *Journal of Networks*, vol. 8, no. 1, 2013.
- [79] A. K. Dey, G. D. Abowd, and D. Salber, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications,” *Journal of Human-Computer Interaction*, vol. 16, no. 2, pp. 97–166, 2001.
- [80] A. Ranganathan, S. Chetan, J. Al-Muhtadi, R. H. Campbell, and M. D. Mickunas, “Olympus: a high-level programming model for pervasive computing environments,” in *Proceedings of the IEEE Third International Conference on Pervasive Computing and Communications*, pp. 7–16, ISBN: 0-7695-2299-8, Kauai Island, HI, USA, March 2005.
- [81] N. Ubayashi, J. Nomura, and T. Tamai, “Archface: a contract place where architectural design and code meet together,” in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, vol. 1, pp. 75–84, Cape Town, South Africa, May 2010.
- [82] J. Muñoz, V. Pelechano, and C. Cetina, “Software engineering for pervasive systems: applying models, frameworks and transformations,” in *Proceedings of the IEEE International Conference on Pervasive Services (ICPS)*, pp. 290–294, E-ISBN: 1-4244-1326-5, Istanbul, Turkey, July 2007.
- [83] D. Cassou, J. Bruneau, C. Consel, and E. Baland, “Toward a tool-based development methodology for pervasive computing applications,” *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1445–1463, 2012.
- [84] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Pearson Education, Third Indian Reprint, London, UK, 2002.
- [85] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine, “Kronos: a model-checking tool for real-time systems,” in *Proceedings of the 10th International Conference on Computer Aided Verification (CAV ’98)*, vol. 1427, pp. 546–550, Springer Verlag LNCS, Vancouver, BC, Canada, July 1998.