sustainability

Article

Zone-Aware Service Platform: A New Concept of Context-Aware Networking and Communications for Smart-Home Sustainability

Jinsung Byun [1], Sanguk Park [2], Keonhee Cho [2] and Sehyun Park [2,*]

[1] Defense Agency for Technology and Quality, Jinju 52851, Korea; bjs561@gmail.com

[2] School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, Korea; pppssuu@cau.ac.kr (S.P.); thckwall@cau.ac.kr (K.C.)

* Correspondence: shpark@cau.ac.kr

Received: 5 December 2017; Accepted: 16 January 2018; Published: 20 January 2018

Abstract: Recent advances in networking and communications removed the restrictions of time and space in information services. Context-aware service systems can support the predefined services in accordance with user requests regardless of time and space. However, due to their architectural limitations, the recent systems are not so flexible to provide device-independent services by multiple service providers. Recently, researchers have focused on a new service paradigm characterized by high mobility, service continuity, and green characteristics. In line with these efforts, improved context-aware service platforms have been suggested to make the platform possible to manage the contexts to provide the adaptive services for multi-user and locations. However, this platform can only support limited continuity and mobility. In other words, the existing system cannot support seamless service provision among different service providers with respect to the changes of mobility, situation, device, and network. Furthermore, the existing context-aware service platform is significant reliance on always-on infrastructure, which leads to great amounts of energy consumption inevitably. Therefore, we subsequently propose a new concept of context-aware networking and communications, namely a zone-aware service platform. The proposed platform autonomously reconfigures the infrastructure and maintains a service session interacting with the middleware to support cost- and energy-efficient pervasive services for smart-home sustainability.

Keywords: context-aware networking and communications; ubiquitous computing; dynamic resource management; zone-aware service platform; smart-home sustainability

1. Introduction

Recent developments in the field of networking and telecommunications have eliminated the restrictions of time and space in the field of information services. These services are called ubiquitous services. The service paradigm is changing from passive service, which is provided at any time and space upon receiving a request from users, to active service, which is intelligently and individually provided upon considering the situation and the environment of the users.

With the increasing demand for these intelligent and personalized services, context-aware service (CAS) systems for connecting entities have been implemented to support them. CAS is a service that is provided to users using various sensors after collecting user contexts, such as gender, age, and preference; environmental contexts, such as time, location, and temperature/humidity; and network contexts, such as bandwidth, network interface, and the performance of a mobile device. The studies on the CAS for intelligent and personalized services have been actively conducted, and various CAS systems have been introduced. However, these systems have the following problems:

- *Increase in the cost according to the improvement in user satisfaction*: In the conventional CAS system, the service infrastructure must be enhanced to improve the level of user satisfaction with the service. For example, for high-quality CAS, sensors, actuators, and network infrastructure must be developed in the service domain.
- *Increase in the energy consumption according to the improvement in user satisfaction*: In the conventional CAS system, many sensors, actuators, and network components need to be always-on in order to provide high-quality CAS, which inevitably leads to a considerably large amount of energy consumption.
- *Lack of continuity and scalability*: The design goal of the current CAS systems is to provide situation-aware services in a single service domain. Thus, the systems do not guarantee users the finest continuity of services and are not flexible enough to utilize large-scale data in the multi-service domain environment.

Moreover, these problems are in conflict with other ubiquitous service paradigms, such as energy saving and green services. In conclusion, the future ubiquitous service paradigm will be shifted to intelligent and personalized services with green initiatives, which simultaneously meets the needs of users and energy conservation.

To address these challenges, we suggest a new concept of context-aware networking and communications, namely zone-aware service (ZAS). In the ZAS architecture, the term "zone" refers to a living space where the infrastructure is autonomously reconfigured based on a user's situation and a user's service session is continuously maintained. In the conventional CAS architecture, users are limited as service receivers (for example, user contexts are collected through fixed sensors to infer the situation, and a smart service is provided to users through fixed actuators). However, in the ZAS architecture, when users enter into the zone, the users become service receivers, and, at the same time, their other roles are created according to the dynamic purposes of the zone. This implies that in the ZAS architecture, users become organic members of the zone. Moreover, this system meets the requirements of the special issue on "The Advent of Smart Homes" for smart-home sustainability.

2. Related Works

Many works related to the CAS platform have focused on service reasoning and context management for intelligent and personalized services. There are various types of CAS systems based on their architecture.

First, the context model-based systems [1] focus on the context modeling and the service reasoning. Ghahfarokhi et al. [2] proposed a context-aware handover decision scheme of a media-independent handover framework for seamless content delivery based on context models. The context models represent the correlation between services and environments, such as space, device, person, and event. However, the systems with the fixed context modeling and the predefined event processing are difficult to adapt to the multi-service domains. Most of the existing systems only consider the hierarchical classification of the contexts and do not consider the dynamic context models, which is an important mechanism for multi-service domains [3,4]. Furthermore, the crucial issues, such as the interconnection of service domains, distribution/aggregation of contexts, and rule reconstruction, have not been considered in the existing system [5].

Second, the agent-based systems [6–10] focus on the reusable agent mechanisms that collect, classify, and reason the contexts for the service prediction. The agent can support the diverse high-quality services by taking into account the user mobility and the dynamic event generation in heterogeneous networks [11]. However, in multi-service domains, a centralized agent system will require the self-configuration with the interconnection with devices, systems, and networks. The platform also needs to set up the multi-device applications on the intelligent devices in a distributed manner [12] providing service convergence.

Third, the service-oriented systems [13,14] concentrate on the service-oriented context models in a single service domain. To accurately reason and manage services, the system processes the various events and situations. However, the excessive event processes may lead to an increase in the resource consumption and frequent unnecessary policy modifications.

In addition, various studies of the CAS architecture for supporting personalized services are being actively conducted. Park et al. [15] proposed the personalized service discovery framework, algorithm, and protocols considering user's preferences and contexts. Baladrón et al. [16] suggested the framework for intelligent service adaptation. This work proposed a solution for a converged context management and discussed an aspect of employment in a future. Bellavista et al. [17] proposed the context-aware semantic discovery platform that discussed how to exploit context-awareness based on the semantic-based method. Domingo et al. [18] discussed a context-aware service architecture based on the body sensor networks and social networks. Various application services to provide personalized services [19,20] have also been suggested based on the CAS. However, the existing studies focused on single or multiple service domains rather than a pervasive service domain.

3. Paradigm of Context-Aware Networking and Communications

The CAS platform has been advancing in the direction of providing pervasive services. These advances are divided into three stages (past, present, and future). In the case of the first stage, the CAS platform provides services in a single service domain. Sensors and networks are not combined together perfectly. In the case of the second stage, the CAS platform offers services in multiple service domains. Situation-aware services that guarantee the user's mobility and device-independent services are provided based on wireless sensor networks (WSNs) in the second stage. In the case of the third stage, the CAS platform provides services in pervasive service domains. It can provide autonomous services ensuring high mobility based on situation awareness. Figure 1 illustrates an example of the paradigm shift. A new paradigm of context-aware networking and communications characterized by service continuity, convergence of network and service, and increased reusability, mobility, and scalability is central to provide intelligent and personalized services in pervasive environments.

3.1. Advanced Service Continuity

The CAS platform will continue to advance in a way that service continuity will be strengthened even more. The service continuity will evolve into a form that it will be independent from time, space, and device, as well as independent from a service provider (SP). The newly-created services must be adapted for the underlying networks throughout various SPs in different locations. In this way, the existing services, such as voice calling, conference calls, messaging, and multimedia content delivery, as well as future network services, can be exposed to various SPs with a fast service delivery, independent of SPs, time, location, and devices. Then, new, value-added services can be launched easily on top of this concept to reach a large mass of users.

3.2. Convergence and Virtualization of Networks and Services

The middleware architecture and context model of the existing CAS platform are possible to manage the services in a single domain. However, the future CAS platform is required to connect to other service domains. For example, as shown in Figure 1 when Tom enters his car while downloading through his cell phone he continues the download using his car multimedia system by service and network convergence partially provided by the convergence network infrastructure (CNI). As he visits a coffeehouse downtown, he enjoys the seamless multimedia delivery using his laptop and finally ends the multimedia transfer on his friend John's smart TV where he can watch it by privilege delegation. Network virtualization is a key attribute for next-generation networking.

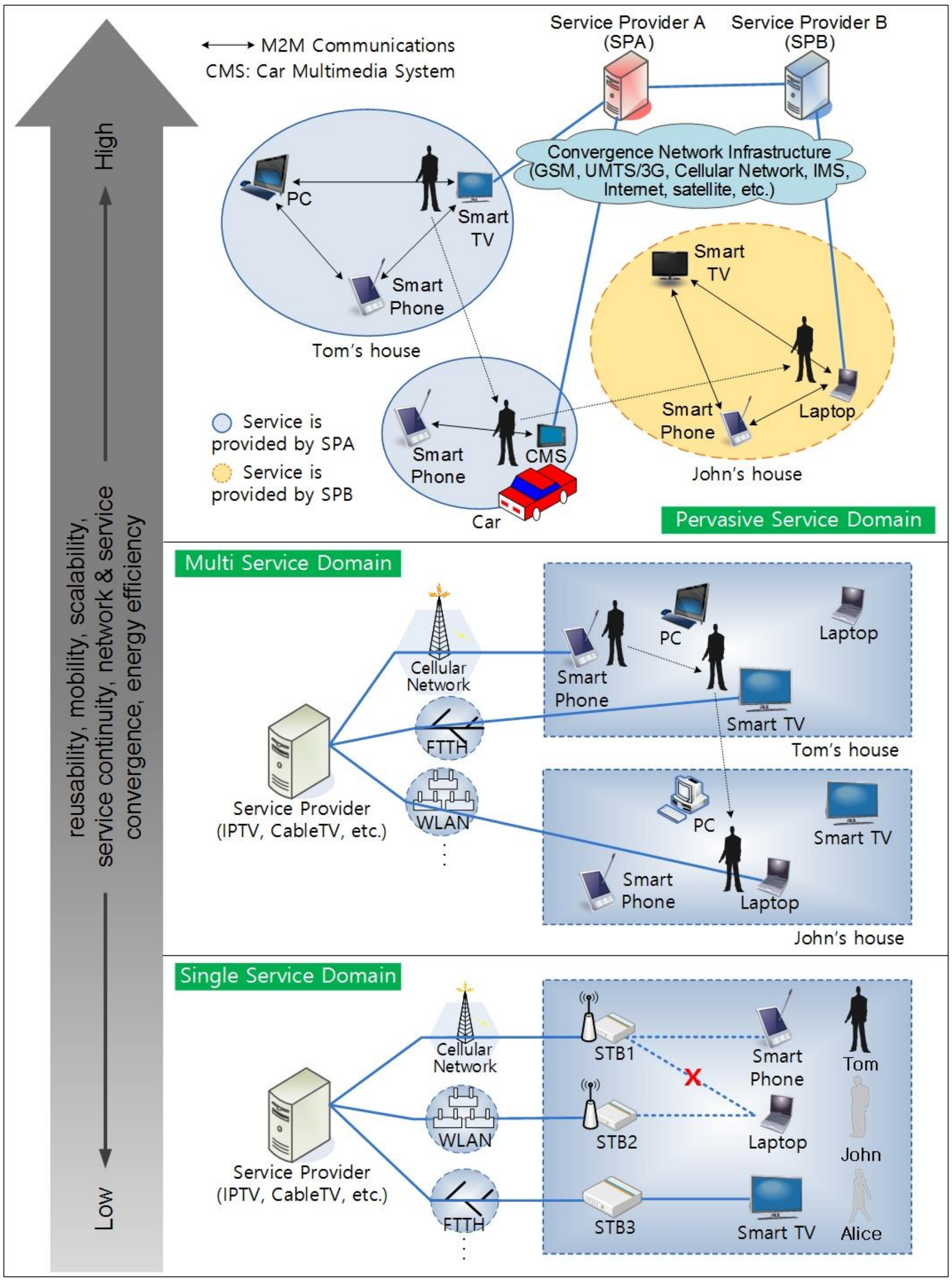

Figure 1. Paradigm shift of context-aware networking and communications.

3.3. Reusability

For the pattern generation and service prediction, the CAS platform gathers the user and environmental contexts in a single service domain. Since the platform has long-term pattern generation in various situations, the future CAS platform is required to gather the service history and environmental contexts in multiple service domains. In heterogeneous contexts and situations, the platform collects and fuses the similar instances for effective pattern generation.

3.4. Mobility

The CAS platform has been evolving into a form in which mobility is enhanced. Whereas most existing CAS platforms entail offering services in fixed space, such as the home and office, the future

CAS platform is expected to provide services seamlessly in the movable space, such as a vehicle or an aircraft. To ensure a high level of mobility, the platform seamlessly switches connection in the case of network failure, handover, and device changes.

3.5. Scalability

In the case of the existing CAS platform, it is not scalable since service is provided via fixed sensors and an actuator infrastructure. This platform has architectural problems. The most crucial problem is a strong correlation between the cost and service quality. To provide a high-quality service, a high-cost service infrastructure such as sensors, actuators, network equipment has to be built in each service domain. However, the future CAS platform should have self-scalability in order to enhance cost-efficiency. Service quality increases by utilizing users' knowledge and resources.

3.6. Energy Efficiency

The existing CAS platform is significant reliance on always-on infrastructure. To accurately reason and manage services, the system processes the various events and situations, which lead to great amounts of energy consumption. To adequately support the energy efficient services, the service platform will need to monitor the power consumption to meet energy requirements.

4. Context-Aware Service Infrastructure

In this chapter, we start with a look at the architecture of conventional CAS system. From the perspective of service-oriented architecture, conventional CAS systems typically consist of a sensor network and device layer, a context and control layer, and a service and application layer.

4.1. Sensor Network and Device Layer

The collected contexts from sensors are the raw contexts related to users. The context can be divided into two types, physical context and logical context, according to the context acquisition method. The physical context is the environmental information that can be acquired by hardware devices called physical sensors. On the other hand, the logical context (e.g., user profile) can be collected by software applications called logical sensors. Some examples of logical sensors are smartphone applications, and server databases providing services.

These sensors are distributed for context acquisition by composing heterogeneous networks via wireless connection with several network resources. In particular, according to the deployment of the sensors, the route and topology of the sensor network can be changed for the reliable and optimal transmission of acquired data. This flexibility of the network prevents the wastage of the sensors' resources, and improves the reliability of data transmission. The wireless technologies mainly used in the sensor network are as follows:

- The WiFi is based on the IEEE standard 802.11, and designed for high data rates and reliability covering small area networks like office and campus area. The data rate is different for each standard; and it can be from several hundred Mbps to several Gbps.
- Bluetooth Low Energy (BLE) is an extension of the conventional Bluetooth, which provides low power, cost, and complexity with guaranteed quality of service and security. BLE operates in the 2.4 GHz over a relatively short range (50 m).
- Zigbee is suitable for applications that need to provide low power consumption, medium data rates, and reliable communications. The transmission distances range from 10 m to 100 m.
- Near-Field Communication (NFC) is a short range radio technology operating at 13.56 MHz with data rates varying from 105 to 424 Kbit/s. the NFC transmits data at short distance (10 cm or less).

4.2. Context and Control Layer

This layer can be generally implemented in context-aware middleware. The middleware is separated from both sensors in the lower layer and from applications in the upper layer. It works in two stages: context management and service management.

4.2.1. Context Management

Low-level contexts acquired by a sensor are simple data in different data formats. Thus, the acquired contexts need to be converted into a high-level context in order to provide practical CAS. This conversion process consists of the following four processes:

- *Context Management and Aggregation:* In this process, the middleware manages and aggregates the contexts acquired from heterogeneous sources. Since the acquired contexts are in different data formats, the data formats of the contexts are unified through the aggregation process. For the speed and accuracy improvement of the context reasoning process, categorizing the contexts will be essential in this process.
- *Context Reasoning:* As the low-level context is just a measured value, such as power consumption and temperature, the measured value needs to be inferred for the effective use of user applications. To achieve this, reasoning mechanisms, such as rule-based, ontological, and probabilistic methods, deduce new types of context (e.g., a user is watching TV or that it is time to turn off the air conditioner). Further, uncertain contexts, such as imperfect, ambiguous, wrong, and unknown contexts from the sensing stage can be detected in this process [15]; this improves the quality of the context, thereby securing reliability. Since users require highly advanced contexts, the reasoning mechanisms should be complex, logical, and computationally efficient.
- *Context Storing:* Low-level contexts from the sensors and new types of context from the context reasoning process are stored in a database through this process. In the database, not only is there real-time context from the sensors, but there are also past contexts that are helpful for a context analysis, and inferred contexts from the reasoning process. However, due to redundancy of the contexts, reducing the amount of contexts repeatedly acquired from the sensors is a necessity. Therefore, an organization process that decides which context needs to be stored or removed is required at this stage.
- *Context Retrieval:* Context retrieval is a process of searching suitable context for service provision. The context can be divided into a static context and dynamic context. The static context remains unchanged, whereas the dynamic context changes with time, requiring periodic updates in order to provide appropriate service. Thus, the middleware should utilize effective retrieval mechanisms that can search suitable contexts and keep the contexts up-to-date for the user applications that require the latest contexts.

4.2.2. Service Management

When a user requires services of the middleware, the middleware searches the required services and service-related contexts. Service management in the middleware provides the user applications with these application-level services in the form of an API. The user can request these services through the Web, and the Web-based services can be operated by using a simple object access protocol (SOAP).

4.2.3. Service and Application Layer

In this layer, a user can be provided services in a wide range of environments, such as the home and office. The user can invoke the CASs through service APIs from the service management by using the applications of a smartphone or smart TV. However, these services should be autonomously provided to users without any direct requirements. Due to the autonomous service provision, the applications should be utilized as logical sensors that have the user's profile, including the user's preference and location information. On the basis of this user's profile, the context-aware application can autonomously invoke various CASs, when the user accesses the service domain.

5. Zone-Aware Service System Architecture

5.1. Architectural Issues and Requirement

When an existing CAS infrastructure is used for providing an intelligent and personalized service to a user, a diverse set of problems may occur because a fixed infrastructure and an IP-based network are used for providing the service. The most important example as stated in the previous section is that high-cost infrastructure is needed to increase service satisfaction. Moreover, significant energy consumption is inevitable to operate this infrastructure. Another well-known example is the very poor performance obtained when coupling with the reliability and the retransmission-based authorization control implemented in the session layer. For designing intelligent and personalized services, the architecture of the future CAS system would need to provide the following characteristics:

- *Network availability:* Seamlessly switches connection-accessible networks in the case of network failure, handover, and device changes. This indicates that the service guarantees device-to-device and end-to-end continuity in all places.
- *Service continuity:* Autonomously resumes a service interrupted by the user or frozen by the absence of service coverage. The connectivity of service needs to be maintained independently from space and time and needs to be managed by a service session.
- *Infrastructure mobility:* Autonomously reconfigures the service infrastructure by using mobile resources if some user-defined policies are met. If a user enters a zone, the resources (i.e., mobile devices and knowledge) that the user has are registered after some authentication and authorization processes. These resources are utilized for service creation.
- *Autonomous device configuration:* Autonomously discovers devices and updates the necessary device configurations. If the user enters the zone, the device is autonomously configured to provide a service that suits a specific situation considering the user's mobile devices and already installed devices in the zone.
- *Performance management of networks and services:* Periodically monitors performance and reports necessary measurements and events. A performance management server (PMS) periodically collects usage statistics and Quality of Service (QoS) or Quality of Experience (QoE) parameters. The QoS/QoE reporting capabilities let service providers conduct measurements at the service level.

5.2. Overview of ZAS Architecture

The proposed ZAS architecture is shown in Figure 2. In this architecture, the contexts are monitored in a sensor network and user's mobile device, and situation-aware services are accessed by the user from smart appliances, such as a smart TV and smart air-conditioner via the CNI.

The ZAS platform is the main component of our proposed architecture. It is installed in a smart appliance and smartphone, and autonomously aggregates the service-based events and dynamically controls the device application according to the service patterns and the interconnection with other systems in a service domain. The ZAS platform also allows users to establish or negotiate a situation-aware service considering a complex set of access selection criteria, thereby accessing the services in the best possible way.

The Mobile resource (MR) is defined as an available mobile device and knowledge that a user has. In existing CAS architecture, to provide a high-quality service, many sensors and actuators, a high-performance processor, and storage have to be installed in a service domain. However, in the ZAS architecture, when a user enters a service domain (zone), the MEs are registered to the ZAS platform after some processes of authentication and authorization and thereby the MEs can be used for service creation. In this way, the proposed architecture can provide cost-effective and relatively high-quality services through only a few resources, such as sensors, installed in the service domain.

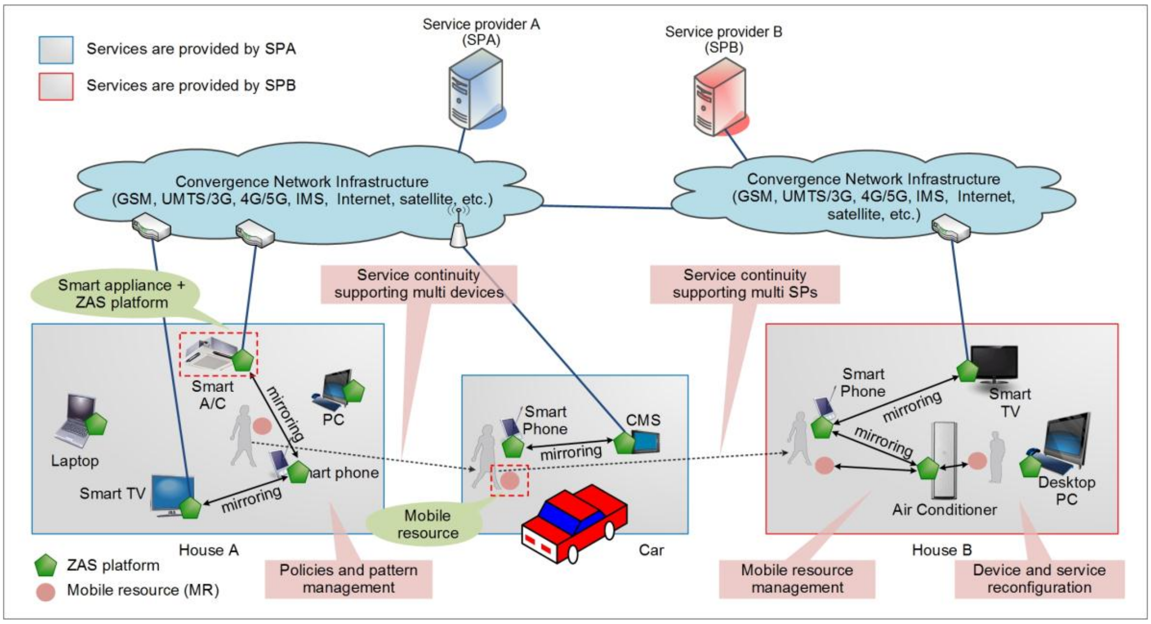
Services are provided by SPA
Services are provided by SPB
Service provider A (SPA)
Service provider B (SPB)
Convergence Network Infrastructure (GSM, UMTS/3G, 4G/5G, IMS, Internet, satellite, etc.)
Convergence Network Infrastructure (GSM, UMTS/3G, 4G/5G, IMS, Internet, satellite, etc.)
Smart appliance + ZAS platform
Service continuity supporting multi devices
Service continuity supporting multi SPs
Smart A/C
mirroring
PC
Laptop
Smart phone
Smart TV
mirroring
Smart Phone
CMS
mirroring
Mobile resource
Smart Phone
mirroring
Smart TV
mirroring
Air Conditioner
Desktop PC
House A
Car
House B
ZAS platform
Mobile resource (MR)
Policies and pattern management
Mobile resource management
Device and service reconfiguration

Figure 2. Overview of the ZAS architecture.

For implementing the ZAS architecture, considering the requirements as mentioned in the previous section, the ZAS platform considers several following features:

- Enhancement of service continuity
- Dynamic management of policies and pattern generation
- Adaptive device and service reconfiguration
- Network and information convergence
- Dynamic management of MEs

The zone-aware service system architecture proposed in this paper is suitable for a smart-home service domain. In domains where many users reside, there are limitations to applying the proposed architecture due to the problem of service conflicts. Collision of services frequently occurs when many users request services. In such a case, although we should decide the priority of service based on predefined policies, in this study we did not deal with the decision of priorities of service. Therefore, in this paper we proposed a context-aware networking and communication concept for smart-home sustainability.

5.3. Architecture of the ZAS Platform

In line with the new paradigm of context-aware networking and communications, we have designed and implemented the ZAS platform for supporting the intelligent and personalized service considering novel features as mentioned above. Figure 3 shows the platform architecture of the proposed system. We mainly focus on the different components compared to the existing system.

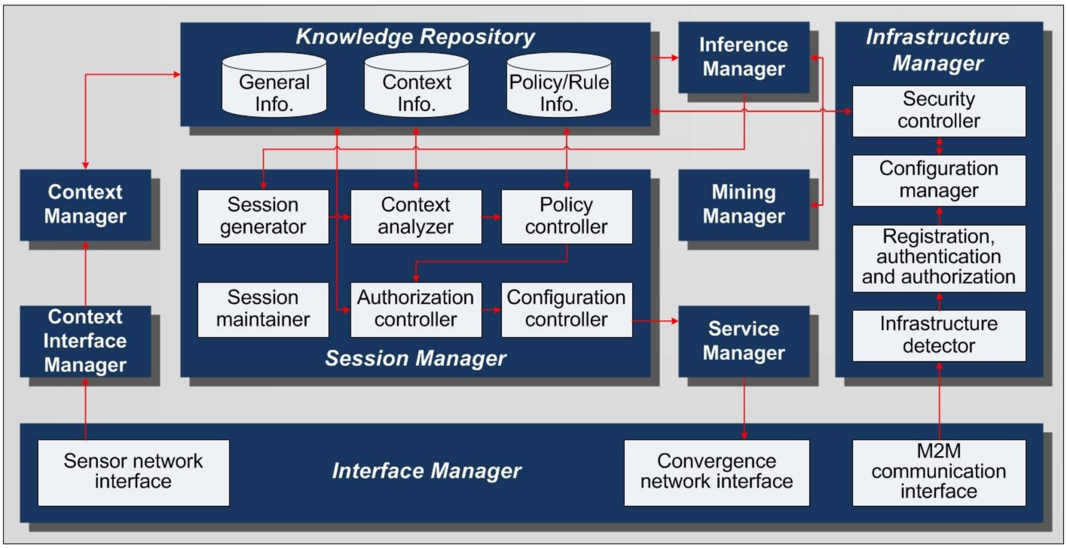

Figure 3. Architecture of the ZAS platform.

5.3.1. Session Manager

The session manager is implemented to accommodate the seamless service delivery at multiple devices and multiple SPs. The session manager seamlessly switches connections in the case of network failure, handover, and device changes, providing the services with end-to-end and device-to-device continuity in all places. The session manager can offer its services to more than one application. The session manager utilizes the session ID with situation ID and service ID to offer service continuity. When the service is interrupted because of a network handover or an explicit user request, the session ID is stored by the session manager, together with all the intermediate contexts (e.g., location, file information, device information, and user information). The client application restarts the service by asking to resume a specific session, unequivocally identified by the session ID. To offer support to a pervasive service, it performs a device switch (from one device to another device) while in the session of service provision by querying the session ID and restoring the context.

- The policy controller sets up the service policies and the conflict solutions with regard to the policy modification and the analysis of the situation patterns. When multiple users move to another domain, in the situations which create the needs of the services or the conflicts, the Policy Manager transmits the related policy and rule to the system of another domain.
- The authentication and authorization controller performs the processes of authentication and authorization. As an example of a simple authorization scheme, it can send a short message service (SMS) with a uniform resource locator (URL) and a passphrase to validate the mobile user. Then, authentication may be performed using the information contained within the SMS, and the service provider cannot access information about the user until the user accepts the service.
- The configuration controller organizes a new service environment adapting to the changes for the optimized service. When the service is delivered seamlessly from one device to another device, there could be a change of the service environment such as a change in device type and performance (e.g., multimedia is delivered from a smart TV to a smartphone). Thus, the configuration controller changes the delivered service into a proper type of service that can be provided in the changed service environment.

5.3.2. Infrastructure Manager

The infrastructure manager handles all of the resources used in the ZAS. The resources used in the ZAS include not only sensing devices, but also a variety of information (e.g., user profile, user experience, and knowledge). Thus, the main function of the infrastructure manager is utilizing the most suitable resource according to the dynamic purposes of the zone to acquire meaningful contexts for the service provision. Moreover, in order to secure the mobility of the ZAS, monitoring the user's resources and network resources is performed periodically in case of the network failure, handover, and device changes.

The security controller plays a role of managing the data security function. In this paper, basically, the data security function is performed by utilizing standardized technologies. That is, the security controller plays a role of adaptively managing standardized data security technologies to ensure reusability and scalability. The security management policies of the security controller are mainly determined by the surrounding network and device environments, security performance, processing speed, security level, service type, etc. Especially in the service environments where network convergence occurs, applicable data security standards differ based on the security policy. Therefore, the security controller plays a role of analyzing the environments and requirements for security performance, and combining the appropriate standard.

5.3.3. Knowledge Repository

The knowledge repository manages the ontology about contexts, policies, and rules. The ontology and the rule using the F-logic update and modify the situations, the service status, and the conflict.

5.3.4. Inference Manager

The inference manager has the main role of deciding when and which services should be provided to a user in a given situation determined by the knowledge repository. It resolves the conflicts and modifies the service status based on the patterns of the user activities and the service history. It builds the situation IDs modeling every active policy. It performs two main tasks: (1) it combines the policies among them considering the system, user and device constraints; and (2) it resolves conflicts among those rules. During this task, all possible static and dynamic conflicts are foreseen.

5.3.5. Context Manager

The context manager categorizes the input data and transforms the adaptive format for the storage to the knowledge repository. It analyzes the situation status, as well as the environment status for updating the situation model. Mining Manager collects and sorts the crawling contexts from Web in respect of the user requests. These engines also autonomously update and offer the mining contexts according to the user profiles and patterns. Service manager consists of a decision manager and a service analyzer. The service analyzer analyzes and interprets the service requirement of a consumer's request. When the service is requested, the service analyzer determines which and how many system resources are needed.

Our context model defines a set of context interconnections for describing events, status, environments, and services. The context represents environmental information and situational information for the dynamic service reconstruction. The environmental information includes the descriptions of the environmental elements, such as devices, places, users, and types of network infrastructure. The situational information includes the descriptions of the elements related to dynamic services and events, such as events, service status, device status, and dynamic variations of situation status. The contexts are categorized into three layers, which are the service layer, he management layer, and the data layer. The service layer contains the set of service-related contexts (service status, service profiles, and requirements of the service) and highly-conceptual contexts (entity, space, and user). The contexts of the service layer assist the query processing and interrelation representation. The management layer has concrete physical contexts and management-related contexts which contain task profiles, location, device profiles, sensing information, activity patterns, solution patterns, policies, and time. The management layer manages the dynamic situations and environment information; and also correlates with various policies for the implementation of service availability and interoperability. Finally, the data layer contains the static contexts.

5.4. Implementation of the ZAS Platform for Seamless Service Provision

We have designed and implemented the ZAS platform for supporting seamless user-centric services. The key functions of the ZAS platform are mobility and situation management, infrastructure reconfiguration, and session management. The ZAS platform reasons and predicts the seamless user-centric services according to various events, as follows:

- *The event of a user's request*: (1) The interface manager gathers the events and analyzes the meaning of contexts, such as environmental and situational data. (2) The context manager categorizes the contexts and transforms the format for the knowledge repository. (3) The session manager creates the user's session and registers the information about a user, a device, network infrastructure, domain, etc. (4) The inference manager reasons user's situations, and it determines the policy of service management. (5) The mining manager gathers and sorts the contexts from sensors and internet according to user's requests. (6) The service manager correlates between the application and the policy regarding services according to the context models. (7) The service manager configures the networks infrastructure and devices according to the user's situations and domain status.
- *The event of user movement to another domain*: (1) The interface manager gathers the events and analyzes the meaning of contexts, such as environmental and situational data. (2) The context manager categorizes the contexts and transforms the format for the knowledge repository. (3) The session manager creates the user's session and registers the information about a user, a device, network infrastructure, domain, etc. (4) The session manager collects the information, such as service/device/network infrastructure status in the previous domain. (5) The inference manager determines which services are provided to a user in a given service session. It modifies the service status based on the user's patterns and service history. It makes the situation ID combines the policies in consideration of user's requirements, and device/network constraints; and it resolves service conflicts among differing policies. (6) The service manager performs a task of authentication. As an example of a simple authentication scheme, it sends a short message service (SMS) message with the one-time password in order to validate the nomadic user. (7) The service manager transmits the policies and rules of the previous domain to the system of the other domain.
- *Service prediction for seamless user-centric services*: (1) The interface manager gathers the events and analyzes the meaning of contexts, such as environmental and situational data. (2) The context manager categorizes the contexts and transforms the format for the knowledge repository. (3) The mining manager automatically and periodically updates the mining contexts. (4) The service manager builds the policies for services and the conflict solutions according to policy modification and situation patterns. (5) The service manager makes the correlations between the service patterns/policies and the collected information (user, device, network infrastructure, etc.) for effective provision of the seamless user-centric service.

The session manager is designed and implemented to accommodate the seamless user-centric service delivery in the pervasive service domain. The session manager seamlessly switches connections in the case of network handover; thus, it provides the user-centric services with entity-to-entity continuity in the pervasive service domain. The ZAS platform utilizes the situation ID, the session ID, and the service ID, in order to guarantee service continuity and service personalization. Each time network handover occurs, the session ID is saved by the session manager, and the related contexts (i.e., space information, device information, user preference, etc.) are also saved to resume the session.

5.5. Network Architecture and Service Flow Based on Service Scenarios

5.5.1. Service Scenario 1: Improvement in Service Continuity through Dynamic Session Management

In multimedia delivery of dedicated devices as shown in Figure 4a, when a user moves to another location, the system verifies the meaningful events to the given situation and predicts the service based on the service patterns of the user. The new device passively processes the whole procedures of a push service from the beginning (device and user registration, content request, transcoding, downloading). In seamless multimedia delivery from the Multiple SPs as shown in Figure 4b, Tom must consent to be served through John's devices by the John's SP. Thus, before the multimedia of Tom can be requested from the other device which the user does not own, Tom's SP must request Tom to handle the authorization. Tom and a foreign device (John's device) are registered and authenticated through the Tom's SP registration interface. The SP can use the SMS for authorization. After the prerequisites are fulfilled, Tom can request the new download of authorized devices. Tom sends a download request for a particular user, John, identified by mobile subscriber integrated services digital network (ISDN) number identifiers (MSISDNs) or an alternative pseudonym(s), depending on the selected privacy solution. If the positioning of John's particular MSISDN and device are authorized with Tom's consent, the request is sent toward the SP. Then the SP allows resuming downloading through John's device.

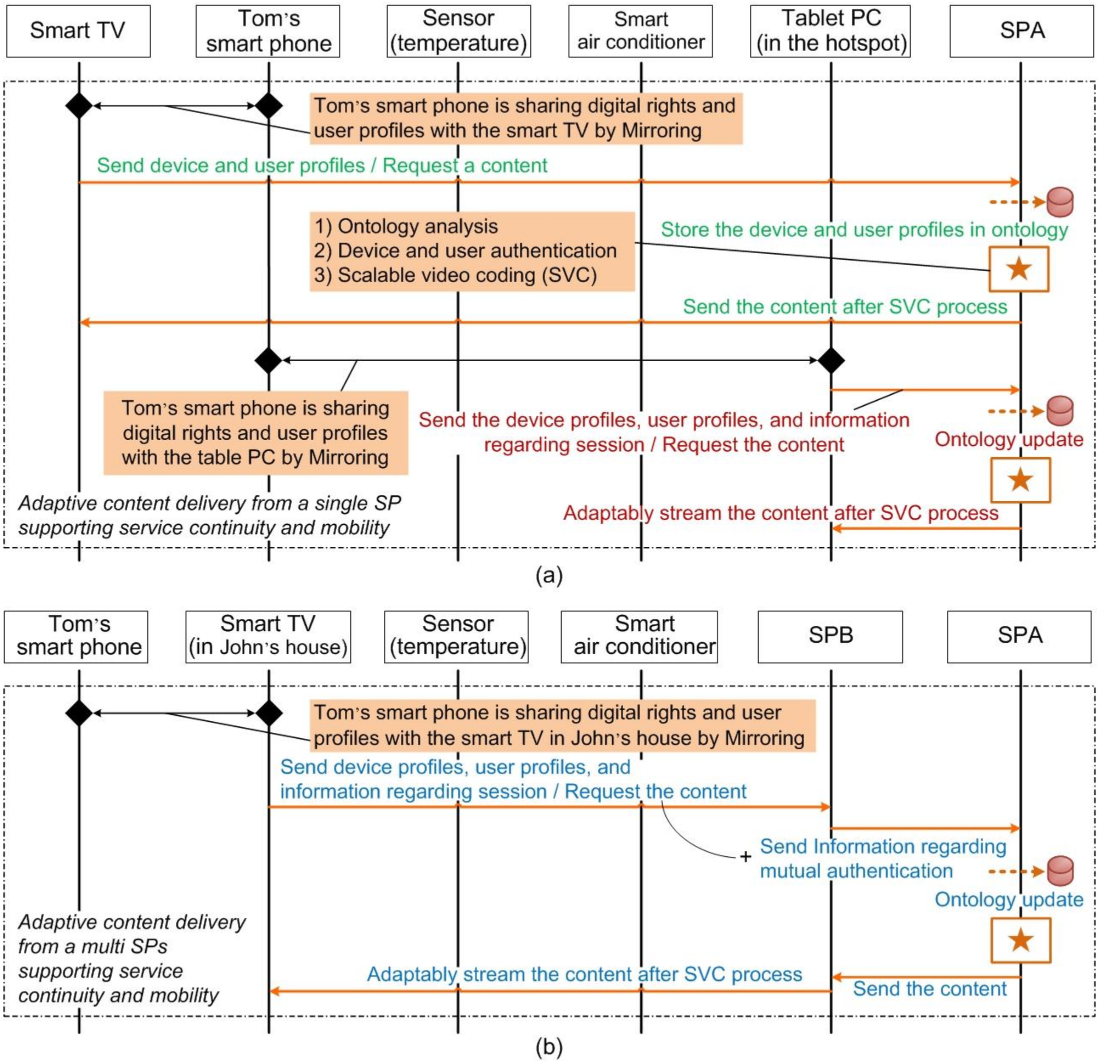
Smart TV
Tom's smart phone
Sensor (temperature)
Smart air conditioner
Tablet PC (in the hotspot)
SPA
Tom's smart phone is sharing digital rights and user profiles with the smart TV by Mirroring
Send device and user profiles / Request a content
Store the device and user profiles in ontology
1) Ontology analysis
2) Device and user authentication
3) Scalable video coding (SVC)
Send the content after SVC process
Tom's smart phone is sharing digital rights and user profiles with the table PC by Mirroring
Send the device profiles, user profiles, and information regarding session / Request the content
Ontology update
Adaptably stream the content after SVC process
Adaptive content delivery from a single SP supporting service continuity and mobility
(a)
Tom's smart phone
Smart TV (in John's house)
Sensor (temperature)
Smart air conditioner
SPB
SPA
Tom's smart phone is sharing digital rights and user profiles with the smart TV in John's house by Mirroring
Send device profiles, user profiles, and information regarding session / Request the content
+ Send Information regarding mutual authentication
Ontology update
Adaptably stream the content after SVC process
Send the content
Adaptive content delivery from a multi SPs supporting service continuity and mobility
(b)

Figure 4. Service flow based on service scenarios: (**a**) device-independent; and (**b**) SP-independent multimedia delivery.

5.5.2. Service Scenario 2: Improvement in Service Satisfaction by Using a Mobile Sensor

Coolness does not guarantee human comfort. The user's comfort can be evaluated as the thermal comfort level (TCL), which represents the psychometric degree of human comfort [16]. The TCL is dependent on indoor temperature and relative humidity. Therefore, in this system, smart devices, such as a smartphone and smart air conditioner using the ZAS platform, share contexts by utilizing a user's humidity sensor and preinstalled temperature sensor for a service of air-conditioner control. As shown in Figure 5, when Tom, who has a smartphone and humidity sensor, enters John's house, Tom's smartphone requests authentication and authorization from a smart air conditioner in order to receive the customized service for Tom. Then, the smart air-conditioner directly sends the request message to the John's smartphone since it is John's house. With John's approval, the smart air-conditioner receives contexts periodically from the Tom's smartphone. The temperature information from the preinstalled sensor and the humidity information from Tom's sensor are utilized for evaluating the TCL. The smart air-conditioner estimates optimal indoor temperature by comparing the evaluated TCL with an optimal TCL that can provide comfortable feeling for Tom, and adjusts the indoor temperature of the smart air-conditioner. In this case, the ZAS achieved not only the reduction of energy consumption but also the guarantee of the user's comfort.

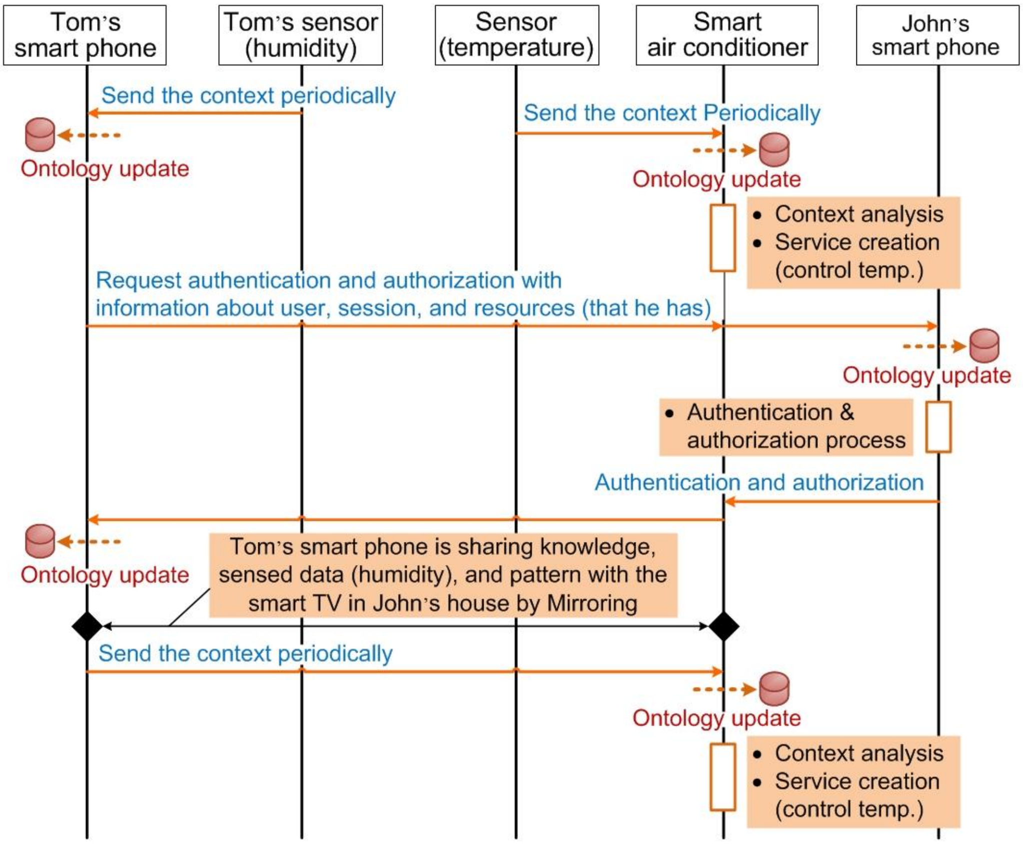
Tom's smart phone
Tom's sensor (humidity)
Sensor (temperature)
Smart air conditioner
John's smart phone
Send the context periodically
Ontology update
Send the context Periodically
Ontology update
• Context analysis
• Service creation (control temp.)
Request authentication and authorization with information about user, session, and resources (that he has)
Ontology update
• Authentication & authorization process
Authentication and authorization
Ontology update
Tom's smart phone is sharing knowledge, sensed data (humidity), and pattern with the smart TV in John's house by Mirroring
Send the context periodically
Ontology update
• Context analysis
• Service creation (control temp.)

Figure 5. Service flow based on a service scenario of appliance control.

5.5.3. Service Scenario 3: Improvement of an Energy Efficiency in Smart Home Environments

One of the advantages that can be obtained by using the proposed ZAS platform is that it can increase energy efficiency. Energy saving is also possible by using the existing CAS through collection of environmental information and intelligent control of home appliance (air conditioner, TV, electric lamp, etc.). However, the services based on the existing CAS require an additional preparatory period (t) to collect sensor data, reason user's situation, and predict a service. Additionally, since context-aware services are proposed based on always-on devices, there are limitations on saving energy. Therefore, in this paper, the ZAS platform can reduce the additional preparatory period and enhance energy saving effects by reusing the contexts and sharing the resources in the service domain. The following shows the advantages in terms of energy efficiency.

- *Reduction of preparatory period:* The ZAS platform have the effect of reducing preparatory period through the service continuity based on the session migration. This allows for additional energy savings.
- *Resource sharing and context reuse:* The ZAS platform can save additional energy as it can reduce the processes of sensor data collection, user's situation inference, and service prediction in other service domains through resource sharing and context reuse.

We compared the proposed model using the ZAS architecture with a simple CAS model to verify the energy saving effect of the proposed architecture. Energy saving using the CAS architecture is x kWh through the intelligent appliance control by utilizing context. On the other hand, the energy saving using the ZAS architecture is $(x + y + z)$ kWh including the energy reduction during the preparatory period through context reuse and resource sharing. The difference in the energy reduction amount depends on preparatory period, t. This implies that energy saving effects using the ZAS architecture is larger than energy saving effects using the CAS architecture in the pervasive service domain where the user's movement between service domains frequently occurs.

5.5.4. Service Scenario 4: Improvement in Cost Efficiency by Using a Mobile Gateway

In smart grids (SG), home area networks (HANs) are formed by using various wired and wireless network technologies. The various networked devices, such as a smart meter, networked home appliance, smart distribution panel, energy storage device, and renewable energy system are connected by the gateway (GW) in HANs. However, in existing HANs, many fixed GWs should be pre-installed, so that it is difficult to extend the system. In the proposed system architecture using the ZAS platform, the user's smartphone can be utilized as the nomadic GW between the various networked devices. Since the user's smartphone does not continuously maintain the connection to the networked devices, a Bluetooth pairing process for networking is executed when the smartphone detects the advertisement signal from the networked devices.

6. Conclusions

New trends of ubiquitous and green computing will require cost- and energy- efficient pervasive services to converge service domains where new innovative services are available by network and service convergence. First, we discussed the paradigm of context-aware networking and communications and suggested six important characteristics and their requirements. In line with these requirements, we proposed a new concept of context-aware networking and communications, namely, the ZAS platform for improving smart-home sustainability, such as service continuity, mobility, scalability, energy efficiency, and interoperability. We designed and implemented the architecture of the ZAS platform for supporting seamless and personalized services. The proposed platform autonomously reconfigures the infrastructure and maintains a service session to support cost- and energy-efficient pervasive services for smart-home sustainability. We also introduced the relevant service scenarios and discussed various aspects of the advantages that can be obtained by using the proposed architecture. We expect this study to contribute to the basic work for developing a new type of system architecture for context-awareness, considering sustainability.

Acknowledgments: This work was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program (10067795)) funded By the Ministry of Trade, Industry and Energy (MOTIE, Korea) and the Chung-Ang University Research Scholarship Grants in 2017.

Author Contributions: Jinsung Byun, Sanguk Park, and Sehyun Park planned the design of the study. Jinsung Byun wrote the manuscript and Keonhee Cho supported the implementation. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Ruiz, J.L.; Dueñas, J.C.; Cuadrado, F. Model-based context-aware deployment of distributed systems. *IEEE Commun. Mag.* **2009**, *47*, 164–171. [CrossRef]
2. Ghahfarokhi, B.; Movahhedinia, N. Context-aware handover decision in an enhanced media independent handover framework. *Wirel. Pers. Commun.* **2013**, *68*, 1633–1671. [CrossRef]
3. Roy, A.; Das, S.K.; Basu, K. A predictive framework for location-aware resource management in smart homes. *IEEE Trans. Mob. Comput.* **2007**, *6*, 1270–1283. [CrossRef]
4. Lee, H.; Choi, J.S.; Elmasri, R. A static evidential network for context reasoning in home-based care. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2010**, *40*, 1232–1243. [CrossRef]
5. Brdiczka, O.; Crowley, J.L.; Reignier, P. Learning situation models in a smart home. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2009**, *39*, 56–63. [CrossRef] [PubMed]
6. Chen, C.; Li, S.S.; Chen, B.; Wen, D. Agent recommendation for agent-based urban-transportation systems. *IEEE Intell. Syst.* **2011**, *26*, 77–81. [CrossRef]
7. Santos, R.; Marreiros, G.; Ramos, C.; Neves, J.; Bulas-Cruz, J. Personality, emotion, and mood in agent-based group decision making. *IEEE Intell. Syst.* **2011**, *26*, 58–66. [CrossRef]
8. Tentori, M.; Rodriguez, M.; Favela, J. An agent-based middleware for the design of activity-aware applications. *IEEE Intell. Syst.* **2011**, *26*, 15–23. [CrossRef]
9. Horng, G.-J.; Lin, C.-K.; Tseng, C.-H.; Wang, C.-H.; Yang, J.-F. An agent-based smart home system and its service-scheduling mechanism: Design and implementation. *Wirel. Pers. Commun.* **2014**, *78*, 521–542. [CrossRef]
10. Li, J.-P.; Hsu, C.-W.; Huang, C.-L.; Horng, G.-J.; Cheng, S.-T. Using ontology and RFID technology to develop an agent-based system for campus-safety management. *Wirel. Pers. Commun.* **2014**, *79*, 1483–1510. [CrossRef]
11. Bellavista, P.; Corradi, A.; Montanari, R.; Toninelli, A. Context-aware semantic discovery for next generation mobile systems. *IEEE Commun. Mag.* **2006**, *44*, 62–71. [CrossRef]
12. Van Kranenburg, H.; Bargh, M.S.; Iacob, S.; Peddemors, A. A context management framework for supporting context-aware distributed applications. *IEEE Commun. Mag.* **2006**, *44*, 67–74. [CrossRef]
13. Bannazadeh, H.; Leon-Garcia, A. A distributed probabilistic commitment control algorithm for service-oriented systems. *IEEE Trans. Netw. Serv. Manag.* **2010**, *7*, 204–217. [CrossRef]
14. Cheng, S.-T.; Chou, C.-L.; Horng, G.-J. Priority-oriented architecture service management on OSGI home-service platform. *Wirel. Pers. Commun.* **2013**, *71*, 611–628. [CrossRef]
15. Park, K.-L.; Yoon, U.H.; Kim, S.-D. Personalized service discovery in ubiquitous computing environments. *IEEE Pervasive Comput.* **2009**, *8*. [CrossRef]
16. Baladron, C.; Aguiar, J.M.; Carro, B.; Calavia, L.; Cadenas, A.; Sanchez-Esguevillas, A. Framework for intelligent service adaptation to user's context in next generation networks. *IEEE Commun. Mag.* **2012**, *50*. [CrossRef]
17. Bellavista, P.; Corradi, A.; Montanari, R.; Toninelli, A. Context-aware semantic discovery for next generation mobile systems. *IEEE Commun. Mag.* **2006**, *44*, 62–71. [CrossRef]
18. Domingo, M.C. A context-aware service architecture for the integration of body sensor networks and social networks through the IP multimedia subsystem. *IEEE Commun. Mag.* **2011**, *49*. [CrossRef]
19. Lee, H.; Kim, J.-G.; Yang, S.-J.; Hong, J. Personalized TV services based on TV-anytime for personal digital recorder. *IEEE Trans. Consum. Electron.* **2005**, *51*, 885–892. [CrossRef]
20. Uhm, Y.; Lee, M.; Byun, J.; Kim, Y.; Park, S. Development of portable intelligent gateway system for ubiquitous entertainment and location-aware push services. *IEEE Trans. Consum. Electron.* **2010**, *56*. [CrossRef]