Ubiquitous computing is necessary to define models for broad contextual information arising out of surrounding environment. It also helps comprehend how to model a mechanism of selectively collecting useful pieces of contextual information and of providing relevant intelligent services. Further, studies are also required on how to process contextual information, its maintenance, and reasoning. However, current context-aware researches are still in need of modeling techniques reflecting ontological characteristics. As a result, it is impossible to effectively provide relevant intelligent services. They are limited as well in terms of contextual reasoning and interoperability across different pieces of contextual information. Aware of the issues, this study proposes an ontology-based context-aware modeling technique, along with a relevant framework, in order to enable efficient specification of contextual information and, thereby, further to provide intelligent context-aware services for context management and reasoning. Moreover, we mobilize the maxim of “five Ws and one H” to process physical and logical contextual information and to support our proposed technique. The maxim-applied modeling technique sets forth an intuitive context-aware schema and demonstrates high applicability to sharing and integration of contextual information. Meanwhile, the ontology-based modeling supports reasoning on contextual information and facilitates more intelligent and reliable services.

1. Introduction

Ubiquitous computing is an environment unbounded by spatiotemporal constraints. It enables computing among humans, objects, and information through interaction among diverse devices immersed in the surroundings. To realize it, support should be secured for context-aware technology pertaining to environment, time, space, and user. One of the gravest problems to such realization lies in lack of the technology that manages and processes various situations arising out of the ever-changing ubiquitous environment. Context-aware technology roots its core in the ability to conduct self-adjustment of conditions, or the ability to detect changes to conditions and to deliver relevant information to users or their handsets [1]. The notion of “context” is dividable in two subcategories: explicit context and implicit context. The former refers to the information that clearly shows the properties of the context, while the latter indicates the collected information that does not reveal its characteristics. The components used for distinguishing the two are concerned not only with the properties of contexts themselves, but also with the devices detecting the contexts. Location-tracking data, for instance, become an explicit context only upon widespread use of global positioning system (GPS) devices.

Even in this case, the data locating a user at a hospital constitutes an explicit context, while the purpose of her visit remains implicit. The purpose, however, turns explicit upon availability of access to hospital reservation system and use thereof.

It is vital to realization of this context-aware technology to be able to provide such environment and technologies that invisibly and automatically process the contextual information received from sensors. In other words, it is necessary to design an intelligent context-aware model for computers’ self-detecting and -processing of contexts.

Under conventional ubiquitous computing circumstances, application of context-aware technology requires intervention of agents in order to provide services best tailored for each situation [2–6]. Aware of the barrier, studies have continued to find breakthroughs. For example, researchers are currently looking into the topic
of context-aware reasoning-supporting mechanism under which ontology-based devices carry out, for themselves, interoperability of different contexts and semantic analysis, as the context-aware technology advances based on intelligent sensors, devices, and systems [7]. Application of ontology to current ubiquitous computing environment is still faced with the issues shown below, despite the numerous studies on how to ontologically process context-aware information.

(i) First, how to form a context-aware model commands the utmost importance in developing an ontology-based framework. The strong coupling context-aware model excels due to its domain-specific quality services, while it requires additional mechanisms to process context-aware information in other domains. Likewise, a loosely coupled context-aware model is beneficial for its ability to use common contexts in processing various types of contextual information. It is impossible under loose coupling, however, to induce reasoning as to how to perform context application and combination to provide quality services.

(ii) Another problem is how to maintain interoperability over context-based data produced by diverse devices and sensors.

(iii) Finally, it is still a remote wish to provide perfect support for functions such as context reasoning for context combination. Especially, solutions must be found as to context application (e.g., services for the context B produced via the context A) and context combination (e.g., the context D as sum of the contexts B and C under the context A) in order to support intelligent, tailored services under the ubiquitous computing circumstances.

To address the problems, it is necessary to study how to define an efficient model that calibrates abstractness of the broad contextual information existing in ubiquitous computing environment. At the same time, it is also worthwhile to study how to model a mechanism of selectively collecting useful pieces of contextual information and providing relevant intelligent services. Further, studies are also required on how to process contextual information, its maintenance, and reasoning.

Thus, in this paper, we define an ontology-based modeling technique upon application of the “five Ws and one H (5W1H)” maxim for clear definition and management of contextual information and further propose a framework that supports context management and reasoning. The framework proposed herein defines a maxim-applied ontology-based modeling technique (i.e., context-aware model based on ontology using five Ws and one H: CA5W1H Onto) as a method for interpretation and abstraction of semantic contexts. It is possible to define the CA5W1H Onto by applying formalism, which is used as a notion of ontology in connection with expressiveness of contextual information. That way, complexity is easily definable by means of ontological properties. In other words, the CA5W1H Onto provides some formalism as to contextual information via the ontology OWL-DL (web ontology language-description logic) [7]. In addition, with the use of the ontology OWL-DL, it is possible to model a specific domain by defining classes, instances, datatype properties (characteristics of instances), and object properties (relations between instances). Further, complexity descriptions of classes and properties are set forth by composing elementary descriptions through specific operations provided by the OWL.

The rest of this study is organized as follows: Section 2 describes the related work, and the concept and framework of the CA5W1H Onto is illustrated in Section 3, Section 4 describes its experiment and implementation, and Section 5 concludes this paper with recommendations for future work.

2. Related Work

Latest developments in the field of the semantic web have unfolded a new set of applications. The semantic web has enabled use of the tools necessary to handle computer-understandable semantics. Generally evolving from XML, these tools function as vehicle to enrich the description of web pages and, thereby, help fully understand the relations between the concepts. OWL [7] and resource description framework (RDF) [8] represent some of the most widely used instruments. These languages offer a significant advantage. They are machine readable and strongly related to description logics. A state of the art on this subject is found in [9]. The RDF language produces resource-related statements in the form of the “subject, predicate, object” triple. The “subject” denotes the resource, and the “predicate” refers to the relationship between the “subject” and the “object.” Built on top of the RDF, the OWL language offers a larger vocabulary and stronger syntax.

Numerous studies have been pursued to come up with modeling techniques for processing of contextual information. Especially, ontology-based context-aware models have the following merits in common: (i) they support ample relations through simple expressiveness; (ii) they enable sharing and integration of contexts belonging to different sources through formal semantics; (iii) they also enable consistent definition of relations to express contexts with ontological tools (e.g., protégé [10], TopBraid [11]); (iv) they further support automatic reasoning, in a consistent way, on expressiveness of various types of contextual information.

Hereunder, we will review some of the most relevant context modeling approaches. These are classified by the scheme of data structures, the structures that are used to exchange contextual information in individual systems.

Conventional key-value models [12, 13] take on the simplest data structure for context modeling purposes. A key-value model provides values of context information as environmental variables and conducts modeling by means of key-value pairing. It is easy to maintain the model. The model, however, is limited in its inapplicability to such structure as fit for efficient context-searching algorithms.

The markup scheme model is a hierarchical data structure consisting of markup tags retaining properties and contents [14]. Its markup tag is always reflexively defined by
other markup tags. The profile is a representative example of this technique. It operates off the serialization induced under standard generalized markup language (SGML), the “superclass” of all markup languages like XML.

The object oriented model is a technique that accommodates the major benefits of being object-oriented like encapsulation and recycle in order to address the problems arising in connection with dynamic attributes of a context in the ubiquitous environment [3]. The details concerning the context processing are encapsulated at the individual object level and hidden in other components, while access to context information is made only through a certain interface.

A logic-based model expresses, based on and through rules, the knowledge about various contexts and provides applications with the values of different contexts [15]. This logic-based approach is hard to maintain and consumes time and money to resolve conflicts, as the number of rules gets bigger.

The notion “ontology” as in the ontology-based models represents a vehicle used to describe concepts and correlations [16]. It provides vocabulary and terminology for sharing. Due to the diversity of context information, ontology is defined for each domain. This type of model supports sharing and recycle of context information in the ubiquitous computing environment. Meanwhile, the hierarchical ontology-based model creates lower-level ontology tailored for a certain domain, using the upper-level ontology. The ontology-based context-aware modeling technique, therefore, supports reasoning on context information based on correlations among context data. With the reasoning possible, a device conducts semantic analysis for itself.

In this and other related fields, several approaches have been proposed for designing architectures designed to mode users and minimize user interaction in the process of supporting situation-dependent personalization of services (e.g., [17]).

Traditional context-aware systems are to be introduced hereunder, to which ontology is applied. Some of the noticeable examples are context broker architecture (CoBrA) [16], service-oriented context-aware middleware (SOCAM) [18], middleware platform for active spaces (Gaia) [19], and context toolkit [20].

Supporting the ontological language “OWL [7],” CoBrA [16] is distinguishable from other systems in that it employs the intelligent space-based architecture. Space functions as agent for creation, management, and distribution of context information in the ubiquitous environment. CoBrA ontology was proposed for modeling context information through ontology. Designed in OWL, CoBaAA-Ontology aims at developing an independent ontology for expression in a system. CoBrA-Ontology supports reasoning and context determination for transmission to web services. In detail, it defines ontology each for action, agent, device, conference, digital document, space, and time. Then, it uses the definitions to provide services tailored for tag and sensor values, or time and space of a user. It lacks interoperability with other instances of context-aware ontology. The shortcoming comes from its use of nonintuitive ontological properties in the ontology-based modeling, and from its focus on development of broker agent for sharing and maintenance of all contexts.

Based on service-oriented architecture (SOA), SOCAM [18] is a middleware running on open services gateway initiative framework (OSGi). Using OWL-based ontology, it supports reasoning on expressions about contexts and on various forms of them. Within the middleware, OWL encoded context ontology (CONON) [21] is used to transform context information into ontology. CONON defines as ontology the upper classes (i.e., ontology) retained by most context-aware systems for its easy application to multiple domains. The feature enables other domain systems to use the upper classes and easy sharing. Based on SOA, the objective of SOCAM is to resolve problems concerning interoperability, sharing, and classification of knowledge. It, however, employs a very complex mechanism for sensing of context information. Further, as loose-coupled ontological modeling technique, CONON carries limited functionality in terms of reasoning on context information (e.g., context application and combination) and high-quality services.

Gaia [19] is a type of the conventional operating system expanded for expressing context information. Gaia collects information on objects in use of ontology servers and collects context information through sensor networks. Then, it extracts upper context (i.e., higher-level context) via reasoning based on the collected information on objects and contexts. Based on DAML+OIL [22], the context model of Gaia takes on the form of “a subject and an object” with a verb inserted in between. Lack of definition and expression of verbs renders semantically perfect context-awareness impossible. Moreover, the DAML+OIL applied in Gaia is more limited than OWL in the inferring functionality for provision of intelligent tailored services.

Context toolkit [20] employs the object-oriented approach and enables easy use of contexts by providing recyclable modules and easily developable frameworks. Strictly speaking, context toolkit lies outside ontology technology. However, it uses the attribute-value tuples for modeling, which assume ontological nature. This mechanism is not an appropriate tool for the purpose of our study, because the absence of a reasoning engine poses limitation in providing context-aware intelligent services, and programming for processing of context information is too complex.

Indeed, most of the aforementioned studies propose ontology-based context-aware modeling techniques. Still, they focus just on defining diverse types of contextual information in ontological language. Further, they employ different modeling techniques for different systems and do not efficiently carry ontological properties (e.g., reasoning) concerning modeled context-aware ontology. Consequently, the studies have failed to overcome the limits pertaining to undefined contextual information. In addition, adoption of different modeling techniques has led to problematic expansion of interoperability and integration among systems.
3. Our Proposed Approach

Reference [23] defines context as "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." In other words, context refers to a special form of knowledge. It constitutes an important modeling requirement. One of the examples is the tradeoff between expressiveness and complexity.

The tradeoff is an important component in determining expressiveness of knowledge and reasoning capability. To resolve the tradeoff issue concerning the context-aware modeling, ontology (i.e., OWL-DL) may be utilized. Ontology amply expresses concepts and their relationships and automatic reasoning in processing contexts based on the expressive capacity.

Proposed herein is a context-aware ontological model based on five Ws and one H (i.e., CA5W1HOnto) as a method of interpreting and abstracting semantic contexts. Our CA5W1HOnto model is designed to support intuitive integration of different context-aware schemas, to which the maxim is applied.

The proposed model consists of <Concept, Instance, Context> triples, and the first two elements of the triple set utilize the properties defined through the existing ontology. The Context element carries all six attributes of the maxim, or why, who, what, where, when, and how. The elements are, in turn, mapped onto corresponding ingredients of the context-aware processing (i.e., goal, role, action, status, location, and time). Figure 1 shows the key elements constituting our proposed model and their relations in class diagram.

3.1. Context-Aware Framework. Figure 2 illustrates the proposed context-aware framework consisting of two parts: "Context Modeling based on ontological concepts" and "context managing and reasoning."

The "context modeling based on ontological concepts" (upper part in Figure 2) is named CA5W1HOnto. The CA5W1HOnto model consists of Concept, Instance, and Context. The element of Context contains various contextual activities. In turn, the Context component helps define the basic characteristics of the maxim and the schemas mapped among context-aware elements. The CA5W1HOnto model proffers services tailored for a specific time, space, and set of user preferences across different domains. The model performs modeling of essential elements by defining, in accordance with the maxim, the contexts required for integration and interoperability of the defined contextual information.

The CA5W1HOnto model defines, in the unit of <Concept, Instance, Context>, the ontological elements (e.g., concept, instance, datatype, data property, object property, etc.) and the elements (i.e., goal, role, action, status, location, and time) for context-aware definition. By defining each in an independent component module, adaptability and independence are guaranteed when developing a context-aware model applicable to diverse domains. In other words,
the maxim-applied context-aware modeling technique is an intuitive model in nature and thereby enables interoperability between systems or models throughout integration and sharing of schemas belonging to diverse domains. Detailed explanation ensues hereunder as to how to map between the properties of the maxim and the context-aware elements defined by the CA5W1H Onto model.

One of the merits from ontologically modeling contextual information lies in its capabilities of automatically extracting new knowledge about current contexts, and providing ample formalism with streamlined expressiveness about the knowledge. Incorporating the merits in the CA5W1H Onto model for support of better intelligent tailored services requires defining algorithms and rules for context reasoning and managing procedures.

“Procedure manager” and the “context reasoner” of the proposed context-aware framework are the functions taking charge of “context management and reasoning” (lower part in Figure 2), respectively. The former manages specifications, processes, and services concerning the contexts defined through the CA5W1H Onto model, consisting of “context specification,” “context operation/interpretation,” and “context service recognition.” In the meanwhile, the latter consists of “context-aware engine,” “reasoning algorithms,” and “reasoning rules,” the components that support context application and reasoning for provision of better intelligent context-aware services.

In this light, the proposed context-aware framework enables combination of high-level context-aware services via efficient management and application of contextual information. Due to the scope of this study, we focus on context-aware modeling and process, leaving for the future such topics as reasoning algorithms and rules for context management and reasoning.

3.2. Modeling of Context Using 5W1H. To process contextual information used across distinct domains requires a model designed to process very complex and multiple contextual situations. Developing this type of model, utilization of the maxim of five Ws and one H helps share and integrate contextual information and, eventually, helps provide quality context-aware services. This section explores, in detail, the proposed CA5W1H Onto model as a context-aware, maxim-applied modeling technique.

Table 1 demonstrates how the CA5W1H Onto constituting maxim attributes are mapped onto the ingredients required for context-aware definition. Specially, the CA5W1H Onto model retains, to define contexts, the six elements of role, goal, Action, Status, location, and time. In addition, the mapping is coupled as why::Goal, who::Role, how::Action, what::Status, where::Location, and when::Time. Under the proposed context-aware modeling, why::Goal represents a desired state that one may wish
to pursue, while its who::Role and how::Action indicate actual steps to be taken to achieve the state.

(i) why::Goal: as the ultimate goal of our context-aware processing, the coupling of why::Goal corresponds to the “why” property of the maxim, and is divided into Personal.Goal, Functional.Goal, NonFunctional.Goal, and Role.Goal.

(ii) who::Role: it defines the role necessary for processing of who::Role, stipulating the roles of Role.Goal, Organization, Actor, and System. It represents the “who” property. It further retains Profile and SW.Agen as subclasses required for defining the role of actor.

(iii) how::Action: corresponding to the “how” attribute of the maxim, how::Action shows how context-aware services are processed. In other words, it represents the context-aware processing stage leading up to realization of why::Goal. Its ingredients are Atomic_Action, Composite_Action, Expectation, Precondition, Effect, Input, Output, and the like.

(iv) what::Status: it represents unique attributes of a context object, which have been detected through sensing. This coupling corresponds to the “what” property and is expressed with Atomic_Status and Composite_Status ingredients.

(v) where::Location: The where::Location corresponds to the “where” property and provides location-related information.

(vi) when::Time: the when::Time represents the “when” attribute and uses Start_Time, End_Time, and Repetition_Time for context-aware processing.

The overall structure of the CA5W1HOnto modeling is presented in Figure 3. In addition, Figure 3 illustrates how contextual information is processed, upon application of our proposed CA5W1HOnto modeling technique.

Processing of contextual information starts with determination of which properties unique to an object are likely to be sensed (i.e., selection of sensed data). This phase corresponds to the “what” property of the maxim and is processed at the Status. Once selected, the Location and the Time, which represent the “where” and the “when” attributes, respectively, provide the time and location information on the data sensed at the Status. In charge of how to process the sensed contextual information, the Action remits to the Goal such property-processing results as precondition, expectation, effect, input, and output. In this case, the Precondition class preliminarily processes the constraint(s). Finally, the Goal sets forth the services necessary to process and define contexts, based on the processing results obtained at the Action and the Role.

Set forth hereunder for our proposed CA5W1HOnto model are the definitions of the ingredients, the maxim attributes and their mapping relations, and the context-aware modeling techniques.

**Definition. CA5W1HOnto Model**

\[
CA_{5W1H} = \{\text{Concepts, Instances, Contexts}\},
\]

**Context Model = \{5W1H(Who, Why, How, What, Where, When,)\},**

Where = \{Role(Actor(Profile), Organization, System)\},

Why = \{Goal(Personal.Goal, Role.Goal, Nonfunctional.Goal, Functional.Goal)\},

How = \{Action(Atomic_Action, Composite_Action, Expectation(Failure), Input, Output, Effect, Precondition, contextual, trustworthy)\},

What = \{Status(Atomic_Status, Composite_Status)\},

Where = \{Location(Atomic_Location, Composite.Location)\},

When = \{Time(Start_Time, End_Time, Repetition_Time)\}.

The contextual information that has undergone modeling through the six steps goes on through another three processes for context management at the “context managing and reasoning,” as shown as Figure 2: “context specification,” “context operation,” and “context service recognition.” Going
through the three processes of processing and selection, the contextual information leads to provision of significant context-aware services. Such contextual information-processing technique, upon application of the five Ws and one H maxim, provides a more intuitive model as to the information and serves as criterion for integrating context-aware data used in diverse domains. Thereby, it enables consistent integration and application of the information in defining and managing it.

4. Experiment

The CA5W1HOnto has been defined in OWL, the web ontology language. As ontology editor, we have used the TopBraid composer [24]. In order to ensure our approach intuitive and understanding of the ontology simple, we have assumed the simple scenario as described in Section 4.2.

The TopBraid Composer is an enterprise-class modeling environment for developing semantic web ontologies and building semantic applications. Fully complying with W3C standards [24], composer offers comprehensive support for developing, managing, and testing configurations of knowledge models and their instance knowledge bases. As part of TopBraid suite, composer incorporates a flexible and extendible framework with a published API to develop semantic client/server or browser-based solutions that can integrate disparate applications and data sources.

Furthermore, in order to enable broad tool support, and to ensure computational completeness and decidability, the ontology has been developed with the aim to conform to the OWL-DL subset of the OWL ontology language.

4.1. Classes and Properties. Figure 4 illustrates the class hierarchy of the ontology, whereas Figure 5 shows the list of created properties, object properties (blue), datatype properties (green), annotation properties (yellow) for the CA5W1HOnto. The class hierarchy splits the classes into two groups, or “context” classes and “ontology” classes.

The category of the “context” contains definitions of 39 classes like goal, action, status, and location. Thus defined, the classes are fixed, since they are vital to context-aware processing.

The ontology class sets forth definitions of Car_Server, Conference, and Emergency_Center. The proposed CA5W1HOnto model defines ontologies and contexts after separating them in the form of module. Therefore, the predefined ontologies become adaptable to the model without modifications. In other words, additional ontologies...
are flexibly addible to each domain for context-aware processing.

4.2. CA5W1H Onto Definition. In this chapter, a usage example is offered, based on a simplified scenario, to attest to the validity of our proposed CA5W1H Onto model. Specifically, it is shown how our modeling technique works and how contextual information is processed.

A circle in Figure 6 represents each class, and a rectangular indicates each property. Further, the rhombus indicates each instance. Hereunder, we will describe the created classes, object, and datatype properties and visualize their relations in TopBraid-OWL editor’s graph.

Scenario. Working for SYS. Lab, Jane hits the road to attend a conference scheduled at 3:00 pm in city B. On the highway, however, she runs into an autoaccident. Sensing the impact through the autosensors, the Car Server reads GPS coordinates with the navigation mounted on her vehicle and sends the information to the nearest emergency center and police station. At the same time, Jane’s scheduler mounted on her handset makes inference on the situation and sends out her absence notice to the conference organizer.

The Car Server (i.e., ontology) is related to the Emergency Center, or another ontology. In the former, various situations possible to occur to her vehicle are stored as ontology instances. When the relation between the two is processed as an explicit context “Auto accident,” its impact affects the context that Jane previously pursues in connection with attending a conference. As a result, a new relation forms between the Car Server and the Conference (i.e., ontology), and the new one is processed as an implicit context “Inability to attend.”

The context arising between the Car Server and the Emergency Center is processed, based on the five Ws and one H maxim. why::Goal has Personal Goal and Functional Goal as its subclasses. Under the scenario, the explicit goal is to send out a notice on the autoaccident to the Emergency Center by the Car Server. At the same time, it becomes an implicit goal to notify the conference organizer of her failure to attend due to the accident, an act that she has been pursuing prior to the accident. Therefore, Functional Goal is to notify the accident, while Personal Goal is to inform the organizer of her absence. why::Goal forms a relation with who::Role as its object.

who::Role is as its subclasses actor and organization, with Jane as actor instance under the scenario. Meanwhile, why::Goal retains how::Action to understand what operations are in need among devices and what::Status that shows the device information. As subclasses of how::Status, Atomic Action and Composite Action exist with the former carrying out the process single handedly and the latter operating it in a composite way. Likewise, what::Status retains Atomic Status, which single handedly processes subclasses, and Composite Status, which carries it out in a composite way. The subclasses of Transmission_GPS, Detecting_Impact, and Send_Message are to be processed together through Composite_Action to locate
The accident point for processing explicit goal as to the accident, to determine whether or not an accident indeed has occurred, and to determine whether to send an absence message (i.e., implicit goal), respectively. Composite_Status has as its instances the navigation to operate what::Status and to process explicit goal, the mobile handset (i.e., mobile_scheduler) to process Car_Sensor with implicit goal. Finally, to process the onset point of the context, the process retains relations with where::Location for understanding the location and when::Time for understanding the time. where::Location has as its subclasses Atomic_Location for processing the information on a single location, and Composite_Location for processing the information on multiple locations. The location information retains longitudinal and latitudinal coordinates as instances of Atomic_Location. when::Time processes the information on the time of the accident. It has Start_Time and End_Time as its subclasses. In the scenario, 2:30 pm makes the instance of the former, while 3:00 pm makes that of the latter.

Definition of Scenario-Based Contexts. Model-based CA5W1H_Onto

CA5W1H_Onto = {Concepts → Car_Server, Emergency_Center, Conference;
Instances → Detecting_Impact, Transmission_GPS, Cannot_attend_Conf, Car_accident, Jane, Send_message, Car_sensor, Navigation, Mobile_scheduler, Latitude, Longitude, 2:30 PM, 3:00 PM;
Contexts → Inability_to_attend, Auto_accident};
Context Model = {5W1H(what, where, when, why, who, how)};
Who = {Role(Actor → Jane)};
5. Conclusion and Future Work

Under ubiquitous computing circumstances, users and systems should be able to detect and process contextual changes on a real-time basis to provide services appropriate for the changes. Therefore, we employed an ontology-based context-aware modeling technique, based on the maxim of five Ws and one H (i.e., CA5W1HOnto). Then, we proposed a framework enabling efficient specification of contextual information as a method of characterizing and standardizing entities pertaining to the contextual information existing in the ubiquitous computing environment. For that purpose, we also defined and implemented the ontological context-aware schema. The proposed CA5W1HOnto model independently separates ontologies and contexts in the form of modules, prior to defining them. In short, our proposed model is advantageous in terms of adaptability and interoperability with the ontologies already developed in diverse domains. Due to these features, the proposed model shows high levels of expandability and recyclability.

In the near future, we plan to quantitatively evaluate our proposed framework. Furthermore, we intend to study algorithms and rules for context reasoning.

Acknowledgments

This work was supported by the Second Brain Korea 21 Project, research funding from Seoul R&BD Program (WR080951), and the basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (2011-0025588). D.-K. Baik is the corresponding author.

References

[1] G. D. Abowd, M. Ebling, G. Hunt, L. Hui, and H. W. Gellersen, "Context-aware computing," IEEE Pervasive Computing, vol. 1, no. 3, pp. 22–23, 2002.
[2] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, pp. 85–90, Santa Cruz, Calif, USA, December 1994.
[3] A. Schmidt, M. Beigl, and H. W. Gellersen, “There is more to context than location,” Computers and Graphics, vol. 23, no. 6, pp. 893–901, 1999.
[4] A. Schmidt and K. Van Laerhoven, “How to build smart appliances?” IEEE Personal Communications, vol. 8, no. 4, pp. 66–71, 2001.
[5] K. Cheverst, K. Mitchell, and N. Davies, “Design of an object model for a context sensitive tourist GUIDE,” Computers and Graphics, vol. 23, no. 6, pp. 883–891, 1999.
[6] C. Ghidini and F. Giunchiglia, “Local Models Semantics, or contextual reasoning = locality + compatibility,” Artificial Intelligence, vol. 127, no. 2, pp. 221–259, 2001.
[7] Web Ontology Language (OWL), http://www.w3.org/2004/OWL/.
[8] Resource description framework (RDF), http://www.w3.org/RDF/.
[9] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, The Description Logic Handbook, Cambridge University Press, 2003.
[10] Protégé, RDF/OWL editor, http://protege.stanford.edu/overview/protege-owl.html.
[11] TopBraid Composer, TopQuadrant, http://topquadrant.com/products/TB_Composer.html/.
[12] M. Samulowitz, F. Michalhelies, and C. linnhoff-popien, “Capeus: an architecture for context-aware selection and execution of services,” in New Developments in Distributed Applications and Interoperable Systems, pp. 23–39, Kluwer Academic, 2001.
[13] B. Schilit, N. Adams, and R. Want, “Context-aware computing applications,” in Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 85–90, December 1994.
[14] WAPFORUM. User Agent Profile (UAProf), http://www.wapforum.org.
[15] V. Akman and M. Surav, “The use of situation theory in context modeling,” Computational Intelligence, vol. 13, no. 3, pp. 427–438, 1997.
[16] GoBrA( Context Broker Architecture), http://cobra.umbc.edu/.
[17] A. Salden, R. Poortinga, M. Bouzid et al., “Contextual personalization of a mobile multimodal application,” in Proceedings of the International Conference on Internet Computing (ICOMP ’05), pp. 294–300, Las Vegas, Nev, USA, June 2005.
[18] T. Gu, H. K. Pung, and D. Q. Zhang, “A middleware for building context-aware mobile services,” in Proceedings of the 9th IEEE Vehicular Technology Conference (VTC ’04), pp. 2656–2660, May 2004.
[19] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahorst, “A middleware infrastructure for active spaces,” IEEE Pervasive Computing, vol. 1, no. 4, pp. 74–83, 2002.
[20] D. Salber, A. K. Dey, and G. D. Abowd, “Context toolkit: aiding the development of context-enabled applications,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’99), pp. 434–441, Pittsburgh, Pa, USA, May 1999.
[21] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, “Ontology based context modeling and reasoning using OWL,” in Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PerCom ’04), pp. 18–22, March 2004.
[22] DAML+OIL, http://www.daml.org/.
[23] G. D. Abowd, D. Anind, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context.
and context-awareness,” in Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC ’99), pp. 304–307, 1999.

[24] The World Wide Web Consortium (W3C), http://www.w3.org/standards/.