The Framework to Support a Common Way for Context-aware Applications

Jong-kwun Baek[*], Hae-sun Jung[*], Chang-sung Jeong[*★]

Abstract

In this paper, we introduce the general way for producing context information to support context-aware applications. It can fetch raw data from the service environments, translate it to reasonable context information, and provide to multiple applications. It is designed originally for the ubiquitous computing middleware and based on the ontology processing model. Automated service applications can use this system as the form of libraries or of web services for deciding its semantic cause of action.

Keywords: context-aware, ubiquitous computing, ontology-based, wireless sensor networks, inference system, service-oriented design

I. Introduction

We already have heard about the ubiquitous computing environment is the expected future of the computer science. When we come into the society that the pervasive computing feature is realized, we will see almost everything related to the accessibility and scalability of the system are automated unaware of the users.

This kind of action related to the user interfaces occur without the user' s explicit action like pressing a button or handling of input devices. To realize this, the system must have the way to get the user' s implicit objectives to use. That is the mean for the context-awareness of the ubiquitous computing system.

If all IT-related companies are starting to produce their own ubiquitous application product massively, the interoperability issue of the product devices will be unavoidably appeared. The worst case of the issue is in the context-awareness means of the system.

The sensors for context-awareness are hardware resources. If two context-aware applications use different protocols for the sensors, the sensor devices must be installed simultaneously at the same places. That is the problem of cost waste.

Because of the issue like this, this paper proposes a common way to gather context-related information for the multiple applications. It is called as '**Context-aware Computing Framework**'. By using global means which the system can know the user' s situation, the cost problem would be answered.

It can be predicted that the systems would use the format of the context information arbitrarily at the ubiquitous computing, so our framework suggests common way to describe context information in order to all various applications will use the context information produced by context framework system. The mean will be introduced as the name of 'Context Domain Ontology'. It is based on the ontology web language (OWL), which is standardized by World Wide Web Committee (W3C).

Even though this system pursues common way to support context-aware applications, implemented system itself is not a stand-alone node. It is composed of multiple components which have their own roles in the overall system parted by the sub-tasks for providing context information to the applications. The components can reside in a local server node, but they can be at the other remote computing nodes. Context-aware framework system has such a distributed components capability.

II. Background

2.1 Context-awareness

Context-awareness feature is considered as the essential point for the pervasive computing background. The context in this term implies broad information about the system, surroundings, and human as the user. Because deliberating all of these for the system-dimension conclusion of service is fully heavy-computing problem, the optimization of processing semantic data is the key issue if the system has multiple general purposes.

2.2 Wireless Sensor Networks

Wireless sensor networks (WSN) are the promising infra for the ubiquitous computing. They can serve a general way of access or generating information about the service enabled area and users. Because of the constrained performance of WSN, the data that system can extract from it is always rack-

[*] Department of Electric Engineering, Korea University

★ Corresponding author

※ This work was partially supported by Smart (Ubiquitous) City Consortium by University of Seoul under Seoul R&BD Program.

of-meaning packet. The application or the system must retrieve the omitted data and restructure so the procedure becomes another burden.

2.3 Ontology-based Computing

We know the modern trend on the web technology is web-service and goes to the semantic web. Web service ensures the independent of communication and integration of heterogeneous service computing system units. Semantic web is the advanced method for more meaningful stubs of web elements. Ontology-based computing designs match the needs for the integration of components using semantic web services and the user-friendly providing of services based on the semantic process of the information space.

Therefore, the system has to include the ability to excerpt needed induced information from a sparse-sense data domain, and to infer probable decision.

Ⅲ. Concept

The context-aware framework mainly has two components for sensing raw data and providing semantics. One component connects to the WSN sink taking raw data and has the capability translating raw data to the pre-defined elements in the sense-data domain ontology. The connection interface gets quite abstract so the system can contact any raw-data streaming sources even if it is not the WSN.

The other component that serves the semantic information to its subscribers is implemented on the basis of the so-called inference engine. The translated information-level data are provided by the former component and this component merges them in its pre-known semantic data structure. The inference process builds up their semantic relations and evaluates their appropriateness. They′re almost measured by a series of rule-set, which is devised by the developer, would decide the fundamental direction of the semantics.

Ⅳ. Structure

According to the above design scheme, the system is designed to the several unit components. Context interpreter and Context aggregator are in the category of context receiving functionality, and Context analyzer and Context provider are in that of context providing functionality.

4.1 Context Interpreter

Context interpreter stands on the very back side of the overall system. It has responsibility of receiving raw-data from the external sensor devices. Because the contacting device cannot be definite, the interface also must not be concrete. The translation process occur same problem above by which we cannot assure what kind of translation scheme is applied in. We decided it to use the sense-data domain ontology model, but any key-value pairs table can be used as the transition reference like the RDBMS stored table, The translator-role component also get abstracted.

We emphasize Context interpreter itself has no ability to retreat semantic operation to infer new information like Context analyzer does. It merely switches the numbered data with words using pre-defined basic elements similarly to dissemblers.

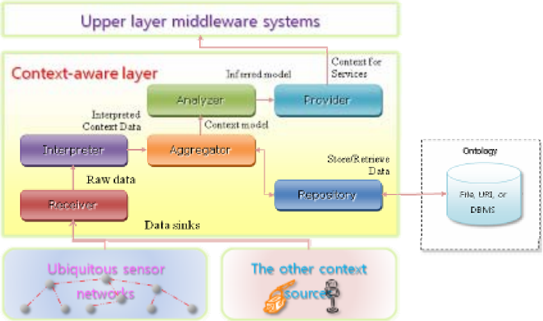

Fig. 1 Overall architecture of Context-aware framework

4.2 Context Aggregator

Time-specific context data are continuously flooded into the system, so a procedure-keeping collection operator is needed. It is Context aggregator in our system. For the reason of requirement of semantic process, one context data element is equivalent to one individual of the context domain-ontology model. Its class name is ′Affair′ in our system. Context aggregator accumulates these affairs in the ontology model it manages, sends to Context analyzer, and stores it using Context repository if needed.

4.3 Context Repository

As the name indicates, Context repository has a responsibility of storing and retrieving context-related data. Stored data can have any form of types like files in the local system or tables in the RDBMS, but usually they′re usually described by the web domain-ontology model.

4.4 Context Analyzer

Context analyzer is the most important component in our system because of its prime role which infers newly probable facts. It constructs full relationships among the received context information and produce service-area context according to the pre-known facts and rules.

4.5 Context Provider

Context provider is the output interface for the front-side applications. It outputs minimal set of context information for invoking new service unit. Principally the assignation of purified context information for the parameters of a procedure is depends on the application using this system, but the application can demand context provider what is needed for its regular working.

Ⅴ. Ontology

Despite of above context operation components, the system yet needs more things to operate properly. They are the ontology and rule set about the specified context. Our system has two major different ontology domains. One is the permanent elements ontology domain and the other is space-time specific context domain.

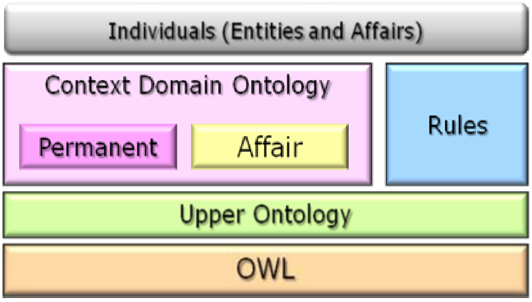

Fig. 2 Layered structure of Context Domain Ontology

5.1 Permanent Ontology Domain

We can always say there are classes that represent all of the individuals about the things around our life-space and the classes can exists regardless of the instance. It is the thinking basis which we see and consider our surroundings.

As the same way, we can construct the representation structure and relationships among classes that compose the structure in the ontology computing system. In our system, it is the permanent ontology model. It is composed of classes that represent all of the problem space like Person, Location, and Object etc.

Not only the classes are available in the domain model, but also the individuals of the classes can exist according to the time-dependent situation. This domain is the main workplace for the Context analyzer to infer new fact about its problem space.

5.2 Space-time Specific Context Domain

In the consideration of the conventional information science ontology computing, there is only the present context situation. The world presents not only on the present time, so we must keep in mind that there are passed time facts that dependent both space and time too.

We had to consider about passed time context for inducing new fact from it. For example, the system can conclude some usage habits of one user from the old usage history. In our system it is represented as ˊAffairˊ class that includes fragmented facts about the received context property.

Affair has three efficient properties. They are the target object of change, the space information (Location), and the timestamp. The target object can be a Person or any subclass of Object that has a property can be changed on the context scheme.

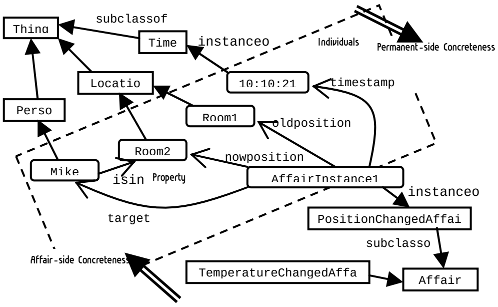

Fig. 3 Example context domain ontology model

Ⅵ. Implementation

We implemented the system assuming that it would be used by the ubiquitous computing middleware, but it is usable in any general application that needs a context-detection feature. The implementation is coded in Java-platform language and used Jena semantic web framework for inference engine.

We assume that each component of the system can be stand-alone in any local or remote computing machines, so the application and components can interact with direct library call or remote procedure call with any protocol like RMI, SOAP etc. However, because the information interchange bandwidth weights heavily on the performance of overall system, component separation onto the several computing machine units might to be considered carefully.

Ⅶ. Conclusion

Our suggested context-aware framework system shows that all the context-aware featured application can use common way to get context information by using the unified interface gateway. From it, the application developer gets advantages of some develop convenience and time reduction. The developers donˊt have to concern about the transmitting of context information occurred by the low-level platform, and the integration problem and WSN reusability.

In this paper, we donˊt describe all of the detailed elements that compose the system like implying functions list of components and ontology classes. They are yet under the revision for more generalized feature but stable because the basic flow concept is absolute. Just by adding more classes and derivation, anyone can easily adopt the merits of this generalized context processing mechanism.

References

[1] Albrecht Schmidt, Michael Beigl, Hans-W. Gellersen "There is more to Context than Location" , Proc. of the Intl. Workshop on Interactive Applications of Mobile Computing (IMC98), 1999. num.6 vol.23: 893-901.

[2] Glassey, R. and Stevenson, G. and Richmond, M. and Nixon, P. and Terzis, S. and Wang, F. and Ferguson, R. I. (2003) "Towards a middleware for generalised context management" . First International Workshop on Middleware for Pervasive and Ad Hoc Computing, Middleware 2003.

[3] Avancha, S. Patel, C. Joshi, A. "Ontology-driven adaptive sensor networks" , Mobile and Ubiquitous Systems: Networking and Services, 2004. :194 □ 202

[4] D.J. Russomanno, C. Kothari and O. Thomas "Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models" , The 2005 International Conference on Artificial Intelligence, 637-643.

[5] Guofei Jiang, Wayne Chung, George Cybenko "Semantic Agent Technologies for Tactical Sensor Networks" , Proceedings of the SPIE Conference on Unattended Ground Sensor Technologies and Applications V, 2003. 11-320.

[6] Jena Semantic Web Framework, http://jena.sourceforge.net/

저자소개

Jong-kwun Baek (Member)

2005 BS degree in Material Science and Engineering, Korea Univ.

2007 MS degree in Electronics & Computer Engineering, Korea Univ.

Concern area: Ubiquitous Computing, Context-awareness computing, Embedded Software

Hae-sun Jung (Member)

1992 BS degree in Computer Science, Myongji University.

2001 MS degree in Electronics & Computer Engineering, Korea Univ.

2003-2008 PhD degree in Electronics & Computer Engineering, Korea Univ.

Concern area: Grid Computing, Ubiquitous Computing

Chang-sung Jeong (Member)

1981 BS degree in Electric Engineering, Seoul Univ.

1984 MS degree in Computer Science, Northwestern Univ.

1987 PhD degree in Computer Science, Northwestern Univ.

1987-1992 Professor at POSTECH

1992-2008 Professor at Korea Univ.

Concern area: Network Virtual Environment, Distributed & Parallel Supercomputing, Image Processing, Grid Computing, Ubiquitous Computing