Design of Rule-based Inference Engine for the Monitoring of Harmful Environments in Workplace†

Yoon-Ae Ahn*

요 약 맨홀, 지하정화조, 저장탱크, 밀폐공간 등의 유해 작업장은 환기가 불충분한 상태에서 산소결핍, 유해가스로 인한 건강장해와 인화성 물질에 의한 화재, 폭발 등의 위험이 있다. 이와 같은 유해환경 정보를 작업장 내의 센서를 통해서 실시간으로 모니터링하고, 위험으로부터 작업자의 안전을 보장할 수 있는 시스템이 필요하다. 이 논문에서는 작업장의 유해환경을 모니터링하기 위한 추론엔진을 설계한다. 제안하는 추론엔진은 규칙기반 시스템의 구조를 가지며 JESS를 활용한다. 제안 시스템은 특정 컴퓨팅 플랫폼에 제약되지 않으며 OSGi 기반의 미들웨어와 연동이 쉬운 특징을 가진다.

핵심주제어 : 추론엔진, 지식베이스, 유해환경 모니터링

Abstract The risk of health impairment due to poor ventilation, fire and explosion by inflammable materials, and other unintended occurrences is always present in dangerous workplaces such as manholes, underground septic tanks, storage tanks and confined areas. Therefore, it a system which can monitor harmful working environment through sensors in workplace on a realtime basis and keep workers safe from the risk is needed. This paper has attempted to design an inference engine to monitor harmful environments in the workplace. The proposed inference engine has a rule-based system structure using JESS. This system is not confined to a particular computing platform and is easily interlocked with OSGi-based middleware.

Key Words : Inference Engine, Knowledge Base, Harmful Environment Monitoring

1. Introduction

In order for a context-awareness system to provide service or information in a ubiquitous environment, it is necessary to collect and analyze context data from its surroundings. A context-awareness system should provide the service needed to users by processing the contexts that have been acquired from various sensors as well as the creation of an upper level context through integration, inference and learning from a lower level context to adapt this new technology to a continuously changing environment. Therefore, many context-aware systems have been developed to support a ubiquitous computing environment [1].

To date, many studies have been conducted on USN but more are still needed on the implementation of inference functions in application services that are directly related to a context -awareness system [2, 3]. A variety of practical research is needed on the implementation of inference algorithms through which context-aware service can be flexibly provided. Studies also are needed on the design of inference engines in order to create a context model that can estimate a user's current status in a ubiquitous context-aware system and apply

† 이 논문은 충주대학교 대학구조개혁지원사업비(교육과학기술부 지원)의 지원을 받아 수행한 연구임

* 충주대학교 컴퓨터멀티미디어학과 부교수

this design to an OSGi-based framework. A context-aware application service model is particularly required in a USN environment along with a USN-based application service model and the establishment of a knowledge base [4].

This paper is a study of an initial stage in order to implement an application prototype that is applicable to a USN-based context-awareness system in real life. Especially, this paper focused on design a knowledge-base and an inference engine to monitor harmful environments in workplaces.

This paper investigates inference engine implementation techniques to provide context-awareness application services in a USN-based context-awareness system. For this, a knowledge base that can be used in an inference engine was established after designing the architecture for a context-awareness application service system, establishing an application service model and analyzing the application data. This paper suggests cases of implementation for inference engines which can be applicable to an OSGi-based framework using the knowledge base of an application service model.

To show the use of the designed application system architecture, a knowledge base has been established using harmful environment data from a manhole. The application service interface and inference algorithm were then designed, and a realtime harmful environment monitoring interface for the workplace was implemented. Lastly, JESS and JAVA-interlocking monitoring interface were implemented to make the system applicable to the OSGi-based framework.

This paper is structured as follows: Chapter 2 introduces the relation research papers of a context-awareness system. In Chapter 3, the harmful environment monitoring system for the workplace is designed. Chapter 4 describes the implementation of the JESS and JAVA-interlocking monitoring interface. In Chapter 5, lastly, the conclusion and future research directions are stated.

2. Related works

In general, context-awareness services are provided based on a context-awareness system or middleware. A context-awareness system creates context data using information collected from sensors and delivers this to context-awareness services. The context-awareness services provide particular services to users depending on this context information. Context-awareness services must therefore be able to clearly understand what kind of context information is delivered from the context-awareness system [1, 5, 6].

"Context awareness" refers to an identity which distinguishes context, place, people and objects and environmental changes including both humans and objects. Within the environment in which the user is located, context can be defined as the user's emotional state, attention, position and direction, date and time, people and objects and others. Therefore, context is decided depending on whether or not the several attributes collected from sensors at a specific time are included in a certain area. In other words, the system should be able to decide context based on the primitive data received from sensors [2, 6].

A context-awareness middleware plays the role of a broker between application programs, which provide the required services to sensors and users. It also detects the user and neighboring environmental information and makes services and information available for users using the detected information. Furthermore, context-awareness middleware should be able to create a variety of context information through integration, inference and learning of the data collected from various sensors. Context data are used to provide customized services to users to provide better service to users, the context-awareness middleware should be able to create and process a variety of context information through integration, inference and learning [3, 7].

The OSGi service platform has a platform independence which has implemented the OSGi SPEC with different devices and an operating system based on the JAVA virtual machine. This service platform can provide basic service management functions and offer system security at different levels. It also supports various service hosting functions that are provided from different suppliers in a single gateway platform [8, 9].

The context-awareness recommendation services which are based on open context-awareness service architecture consist of an OSGi framework, context provider and context-awareness application. This OSGi framework-based open architecture consists of several service supply bundles, inference engine and a knowledge framework that stores and manages data. Useful information is provided to users through modeling, conversion and inference [4, 10, 11].

The risk of health impairment because of poor ventilation, fire and explosion by inflammable materials and other unintended occurrences is always present in harmful workplaces such as manholes, underground septic tanks, storage tanks and confined areas. Therefore, the development of a system which can monitor a harmful working environment through the sensors in the workplace on a realtime basis and keep workers safe from harm is needed [12, 13].

This paper has attempted to design an inference engine to monitor harmful environments in workplaces. The proposed inference engine has a rule-based system structure using JESS. This system is not confined to any particular computing platform and is easily interlocked with OSGi-based middleware.

3. Design of a harmful working environment monitoring system

Context-awareness services are provided based on a context-awareness system or middleware. A context-awareness system creates context data using the data collected from sensors and delivers them to context-awareness services. The context-awareness services provide particular services to users depending on the context information which has been delivered from a context-awareness system.

3.1 Harmful environment information in a confined workspace

The term 'confined space' refers to a workplace in which there is a risk of health impairment because of poor ventilation, fire and explosion by inflammable materials and other accidents. Entrance to a confined space is very limited, and ventilation is poor. In general, air is constituted by approximately 21 percent oxygen. All creatures on earth breathe oxygen in stay alive. At high altitudes, humans may become out of breath because of the low atmospheric pressure. Even though the amount of oxygen in

<Table 1> The effect of low oxygen on the human body

Oxygen Concentration (%)	Effect
21	Oxygen concentration level in normal atmosphere
19	Safety threshold concentration (the minimum level to prevent oxygen loss)
17	Difficulty in breathing, increased pulse, headache, nausea, candles will go out
15	More difficulty in breathing, pulse increases
12	Dizziness, vomiting and substantially weakening grip strength Risk of falling from a ladder
10	Pale, unconscious (cannot move), vomiting (may cause suffocation)
8	Loss of consciousness (death occurs in eight minutes)
6	No muscular response, unconscious, spasms, breathing stops
4	Loss of consciousness within 40 seconds

the air is the same, at 21 percent, the absolute amount of oxygen inhalation per breath decreases. In other words, high altitudes are far lower than lower altitudes in terms of the amount of oxygen that can be efficiently absorbed [14]. The effect of insufficient oxygen on human body is described in Table 1 below:

Hydrogen sulfide (H2S) has an addled egg smell. It can be detected in organic sludge sediment. Even though 10ppm or less H2S may be detected after cleaning a tank, a high level of H2S (100-200ppm) is sometimes measured from the mud sediment. Proper ventilation is then extremely important as well as protective equipment such as oxygen masks [15]. The effect of hydrogen sulfide on human body is stated in Table 2 below:

<Table 2> The effect of hydrogen sulfide on the human body

H2S Concentration (ppm)	Effect
1 - 2	Slight bad smell
2.4	Mild bad smell
3	Severe bad smell
5 - 8	Bad smell even from an analyzer
80 - 120	Mild symptoms; endurable for about 6 hours
200 - 300	Smell is as mild as in a low concentration, but strong pain in eyes, nose and throat occurs in 5-8 minutes Endurable for 30-60 minutes
500 - 700	Inhalation for about 30 minutes can cause semi-acute intoxication
1,000 - 1,500	Acute intoxication, unconsciousness and death

Carbon dioxide is common in air. However, a high concentration of carbon dioxide in a confined space is very dangerous. Despite low toxicity, this odorless gas can cause headache and even a breathing problems with long-term exposure [16]. Table 3 below demonstrates the effect of carbon dioxide on the human body:

<Table 3> The effect of carbon dioxide on the human body

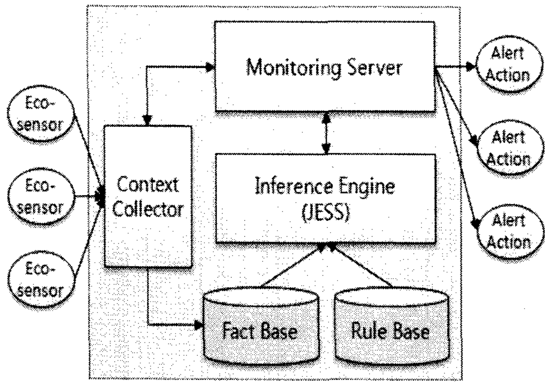
Concentration (ppm)	Effect
60,000(6%)	Inhalation for 30 minutes can cause breathing problems, drowsiness and headaches.
100,000(10%)	Narcolepsy, dizziness and unconsciousness
250,000(25%)	Death

3.2 Structure of rule-based inference engine

This study has designed a rule-based inference engine. The rule-based system defines action rules based on several contexts and makes a decision with the occurrence of a context that meets a rule. Therefore, this engine is suited to the inference engine structure that decides the risk of a work environment depending on the value of realtime harmful environment information collected from sensors in a workplace. The proposed system is not confined to a particular computing platform but inter-locks easily with OSGi-based middleware. Figure 1 shows a system structure that monitors a harmful workplace environment, consisting of monitoring the server, inference engine, context collector, knowledge base and eco-sensor.

A monitoring server is a server module that controls the operation of an entire system. It collects data on harmful gas in the workplace, stores them in a knowledge base, monitors the harmful work environment through an interference engine and triggers an action alert through a mobile device in the work area if danger is detected.

An inference engine monitors worker safety using information on the work environment. It infers a dangerous situation with rules related to breath analysis of oxygen, carbon dioxide, hydrogen sulfide and other gases and delivers the results to the monitoring server.



<Figure 1> Work environment monitoring system

A context collector covers the section in which sensor information is stored. The data collected from environmental sensors in the workplace are classified into the following three attributes: name, time and value. Name refers to the name of the context from sensors while time and value represent the time that the context value is entered and the context value from the sensors respectively.

The knowledge base is divided into a rule base and a fact base. The former refers to the rules to be executed in a JESS inference engine while the latter is context information that is stored in the working memory of the inference engine. An eco sensor senses environmental information in harmful workplaces such as manholes, underground septic tanks, storage tanks and confined space and delivers it to the server. A harmful environmental gas sensor detects the concentration of oxygen, carbon dioxide, hydrogen sulfide, carbon monoxide, organic compound, ammonia and sulfur dioxides.

3.3 Knowledge base design

JESS is a JAVA- and rule-based engine which uses a Rete algorithm. A rule is defined and inferred using internally supported rule-based language or XML. The three basic factors provided by JESS include a 'fact list and instance list' which will be stored in the memory, a 'knowledge base' and an 'inference engine' that controls rule execution. The template information is similar to JAVA class declaration.

A template provides related technology for a simpler and clearer management of slot-based complex but single fact information. Figure 2 below defines the template to create knowledge information using the 'deftemplate' keyword. In 'status' template, four attributes are defined.

Each fact is stored in working memory while an inference engine manages all inference-related context information based on the fact information. Figure 2 allocates an initial value to a 'status' template. 'Deffacts' is a keyword which allocates each fact to each template attribute. The 'slot parent' is the stored fact created with a duplicate function. It is used to decide the sequence of execution of the rule and in debugging. The category 'multislot' can store several lists such as oxygen, carbon dioxide, hydrogen sulfide and carbon monoxide-related information. The 'last-move' category stores the selected rule.

```
(deftemplate MAIN::status
    (slot search-depth)
    (slot parent)
    (multislot ochc2)
    (slot last-move)
)
(deffacts MAIN::initial-positions
    (status
        (search-depth 1)
        (parent root)
        (ochc2 (get_o2) (get_co2) (get_h2s) (get_co))
        (last-move no-move))
)
(defmodule SOLUTION)
(deftemplate SOLUTION::moves
    (slot id)
    (multislot moves-list)
)
```

<Figure 2> Definition of fact list using template

```
(defrule MAIN::SAFE-ONE
    ?node <- (status (search-depth ?num) (ochc2 ?o2&:(and (<= ?o2 100) (>= ?o2 90))
              ?co2&:(<= ?co2 10) ?h2s&:(< ?h2s 20) ?co&:(< ?co 20)))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 ?*pp*) (+ ?co2 ?*pp*) ?h2s ?co) (last-move SAFE-ONE))
)
(defrule MAIN::SAFE-TWO
    ?node <- (status (search-depth ?num) (ochc2 ?o2&:(and (< ?o2 90) (> ?o2 75))
              ?co2&:(< ?co2 25) ?h2s&:(< ?h2s 30) ?co&:(< ?co 30)))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 ?*pp*) (+ ?co2 ?*pp*) ?h2s ?co) (last-move SAFE-TWO))
)
(defrule MAIN::SAFE-THREE
    ?node <- (status (search-depth ?num) (ochc2 ?o2&:(and (<= ?o2 75) (>= ?o2 65))
              ?co2&:(<= ?co2 35) ?h2s&:(< ?h2s 40) ?co&:(< ?co 40)))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 ?*pp*) (+ ?co2 ?*pp*) ?h2s ?co) (last-move SAFE-THIRD))
)
(defrule MAIN::SAFE-FOUR
    ?node <- (status (search-depth ?num) (ochc2 ?o2&:(and (< ?o2 65) (> ?o2 45))
              ?co2&:(< ?co2 55) ?h2s&:(< ?h2s 50) ?co&:(< ?co 50)))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 ?*pp*) (+ ?co2 ?*pp*) ?h2s ?co) (last-move SAFE-FOUR))
)
(defrule MAIN::SAFE-FIVE
    ?node <- (status (search-depth ?num) (ochc2 ?o2&:(and (<= ?o2 45) (>= ?o2 25))
              ?co2&:(<= ?co2 80) ?h2s&:(< ?h2s 50) ?co&:(< ?co 50)))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 ?*pp*) (+ ?co2 ?*pp*) ?h2s ?co) (last-move SAFE-FIVE))
)
(defrule MAIN::SAFE-SIX
    ?node <- (status (search-depth ?num) (ochc2 ?o2&:(and (< ?o2 25) (>= ?o2 0))
              ?co2&:(<= ?co2 100) ?h2s&:(< ?h2s 60) ?co&:(< ?co 60)))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 ?*pp*) (+ ?co2 ?*pp*) ?h2s ?co) (last-move SAFE-SIX))
)
(defrule MAIN::SAFE-VENTILATION
    (declare(salience 100))
    ?node <- (status (search-depth ?num&:(= (mod ?num 10) 0)) (ochc2 ?o2 ?co2 ?h2s ?co))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (+ ?o2 5) (- ?co2 5) (- ?h2s 1) (- ?co 1)) (last-move SAFE-VENTILATION))
)
(defrule MAIN::UP-H2S-CO
    (declare(salience 100))
    ?node <- (status (search-depth ?num&:(= (mod ?num 6) 0)) (ochc2 ?o2 ?co2 ?h2s ?co))
    => (duplicate ?node (search-depth (+ 1 ?num)) (parent ?node)
        (ochc2 (- ?o2 (get_pwh2s)) (- ?co2 (get_co)) (+ ?h2s (get_pwh2s)) (+ ?co (get_co)))
        (last-move UP-H2S-CO))
)
```

<Figure 3> Definition of inference rules for monitoring

Sensing Data :

```
(deffunction file_read()

    (bind ?file (new java.io.File input.txt))
    (bind ?fin (new java.io.FileInputStream ?file))
    (bind ?din (new java.io.DataInputStream ?fin))
    (return ?din))
```

Oxygen :

```
(deffunction get_o2()

    (bind ?din (file_read))
    (bind ?o2 (call ?din readInt))
    (return ?o2))
```

Carbon Dioxide :

```
(deffunction get_co2()

    (bind ?din (file_read))
    (bind ?co2 0)
    (for (bind ?i 0) (< ?i 2) (++ ?i) (bind ?co2 (call ?din readInt)))
    (return ?co2))
```

Hydrogen Sulfide :

```
(deffunction get_h2s()

    (bind ?din (file_read))
    (bind ?h2s 0)
    (for (bind ?i 0) (< ?i 3) (++ ?i) (bind ?h2s (call ?din readInt)))
    (return ?h2s))
```

Carbon Monoxide :

```
(deffunction get_co()

    (bind ?din (file_read))
    (bind ?co 0)
    (for (bind ?i 0) (< ?i 4) (++ ?i) (bind ?co (call ?din readInt)))
    (bind ?p (call ?din readInt))
    (bind ?*pp* ?p)
    (return ?co))
```

<Figure 4> Functional definition for execution of inference engine

The rule information is set to infer a conclusion after an inference process based on fact information. Using the 'defrule' keyword, rules are defined as follows. The rules defined in Figure 3 determine the safety conditions in the work environment.

<Table 4> Functions of inference rules

Rule	Function
SAFE-ONE	Judges the context of normal oxygen concentration
SAFE-TWO	Judges the context of minimum oxygen concentration to prevent lack of oxygen
SAFE-THREE	Judges the context that causes pulse increases and headaches
SAFE-FOUR	Judges the context of lack of judgment, emotional instability and mental instability
SAFE-FIVE	Judges the context that causes unconsciousness and disorder of the central nervous system
SAFE-SIX	Judges the context of a coma and lack of breathing
SAFE-VENTILATION	Judges the context of ventilation for supply of oxygen
UP-H2S-CO	Judges the context of increase in hydrogen sulfide and carbon monoxide

Table 4 describes the inference functions of the rules defined in Figure 3, which include SAFE-ONE, SAFE-TWO, SAFE-THREE, SAFE-FOUR, SAFE-FIVE, SAFE-SIX, SAFE-VENTILA TION and UP-H2S-CO.

The proposed inference rule algorithm, in this paper, is based on the Table 1, 2, and 3 that are described in section 3.1. This inference rule algorithm is composed of the real data that is applicable to the application system proposed in this paper.

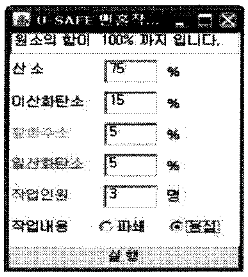
4. Implementation of JAVA- and JESS-interlocked monitoring interface

JESS is a JAVA- and rule-based engine which uses a Rete algorithm. A rule is defined and inferred using internally supported rule-based language or XML. The use of JESS allows the creation of a JAVA program which can implement inference based on rules. JESS offers a method to share values between the system control module and JESS engine via functions. In a JESS-based inference engine, functions are defined as follows: 'deffunction' is a keyword that defines functions, and file_read() and get_o2() are the names of the defined functions.

Figure 4 indicates the functions that call the foreign sensing data, oxygen, carbon dioxide, hydrogen sulfide and carbon monoxide values that are necessary for the JESS inference engine. It plays the role of receiving data in the input interface in the initial work environment.

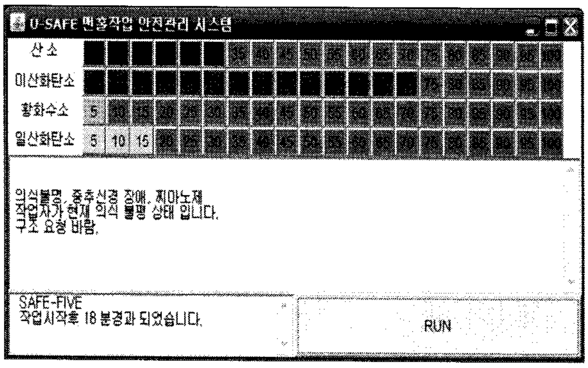
To read the data that have been received from JESS, all data are stored in a file format. If data are entered in a user interface, they are stored in a file format after creating a file object. Here, the total sum of oxygen, carbon dioxide, carbon monoxide and hydrogen sulfur should not exceed 100.

The workplace state monitoring interface creates the needed objects and performs continuous checks during the inference of the knowledge base.



<Figure 5> Interface to set up initial data

The monitoring interface shows the current environment of the workplace using information collected from sensors on a realtime basis. If potential danger is detected in the work area, a warning message appears on the monitor screen and an alarm is sounded.



<Figure 6> User state monitoring screen

A monitoring interface shows a graph using values from the input screen. The graph is expressed in buttons by five units. Oxygen, carbon dioxide, hydrogen sulfur and carbon monoxide are shown in blue, red, orange and yellow respectively.

5. Conclusions

Many studies to date have been conducted on USN sectors but finding papers on the implementation of inference function in application services that are directly related to a context-awareness system is still difficult. Practical, diverse studies on the implementation of an inference algorithm which can flexibly provide context-awareness services in USN environment is particularly needed.

This study has designed the architecture of a context-awareness application system and established an application interface and knowledge base in order to monitor harmful environments in workplaces. This paper has suggested research directions for the implementation of inference functions in application services that are directly related to a USN-based context-awareness system. In addition, with regard to the implementation technique of inference algorithm which can flexibly provide context-awareness services under a USN environment, practical case studies have been performed.

This paper has proposed a model applicable to the development of an open context-awareness framework through the implementation of JESS- and JAVA-interlocked inference engine. It appears that the proposed system structure could be very useful in managing contexts which are used in various systems such as a home network system, context-awareness recommendation system and LBS system.

Further study will be systematically conducted for the completion and comparative evaluation of a user state model aimed at monitoring harmful environments in the workplace, the establishment of a knowledge base and the implementation and analysis of an inference engine. The results of this paper could be applied to underground facility-related industries in advance. For this, a prototype applicable to diverse industries that manage underground facilities needs to be completed.

References

[1] A. Harter, A. Hoper, P. Steggles, A. Ward, and Paul Webster, “The Anatomy of a Context-aware Application”, Journal of Wireless Networks, vol.8, pp.187-197, 2001.

[2] J. Farringdon, A. J. Moore, N. Tilbury, J. Chru -ch, and P. D. Biemond, “Wearable Sensor Badge & Sensor Jacket for Context Awareness”, Proceedings of the 3rd International Symposium on Wearable Computers, pp.107-113, 1999.

[3] N. Priyantha, A. Chakraborty, and H. Balakr-ishnan, “The Cricket Location-Support System”, Proceedings of the 6th ACM International Conference on Mobile Computing and Networking, pp.32-43, 2000.

[4] B. N. Schilit, N. Adams, and R. Want, “Context -aware Computing Applications”, Proceedings of the Workshop on Mobile Computing Systems and Applications, pp.85-90, 1994.

[5] D. I. Ahn, C. S. Shin, and S. C. Joo, "Context

Information Supporting System Based on Indoor Location for Healthcare Home Service", Proceedings of Korean Information Science Society Conference, Vol.33, No.1(D), pp.64-66, 2006.
[6] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System", Journal of ACM Transactions on Information Systems, Vol.10, No.1, pp.91-102, 1992.
[7] J. A. Lee, S. H. Lim, J. H. Kim, and W. D. Cho, "Design and Implementation of Monitoring System on Context-aware Middleware In Ubiquitous Environment", Proceedings of Korea Computer Congress, Vol.34, No.1(B), pp.249-254, 2007.
[8] Y. A. Ahn and J. S. Park, "Design of Location Based Service Manager in Open Context Awareness Architecture", Proceedings of Korea Multimedia Society, Vol.11, pp.138-141, 2008.
[9] OSGi Alliance, "Open Service Gateway Initiative", http://www.osgi.org
[10] D. J. Kim and J. H. Kwon, "Design of Recommendation Service in Open Context-aware Service Architecture", Proceedings of Korea Computer Congress, Vol.33, No.1(C), pp.19-21, 2006.
[11] M. H. Ahn, D. H. Kim, and J. H. Kwon, "Design and Implementation of OSGi-based Context-aware Service Architecture in Ubiquitous Environment", Proceedings of Korea Computer Congress, Vol.33, No.1(C), pp.16-18, 2006.
[12] J. M. Yun and P. Park, "User-Centered Design of Workplace Environment Monitoring Service using IP-USN", Proceedings of Korea Society of IT Service, pp.266-271, 2008.
[13] Y. A. Ahn, "Context Awareness Inference Engine for Location Based Applications", Proceedings of International Conference on Convergence and Hybrid Information Technology, pp.213-216, 2009.
[14] D. H. Han and J. H. Lee, "New Occupational Hygiene Management", DongHwa Technology Publishing, pp.161-163, 2007.
[15] G. M. Ryu, H. H. Park, and G. J. Jeong, "Study of Cause and Prevention for Asphyxia Disaster in the Enclosed Geometry", National Institute for Occupational Safety and Health, Research Report 2008-140-1477, pp.72-76, 2008.
[16] B. S. Son, J. A. Park, B. G. Jang, T. W. Jeong, W. H. Yang, G. J. Kim, B. H. Lee, N. Y. Cho, S. B. Choi, and C. N. Gho, "New Environment Science", DongHwa Technology Publishing, pp.161-162, 2006.

안 윤 애 (Yoon-Ae Ahn)

• 정회원
• 1992년 2월 : 한남대학교 전자계산공학과 (공학사)
• 1996년 2월 : 충북대학교 전자계산학과 (이학석사)
• 2003년 2월 : 충북대학교 전자계산학과 (이학박사)
• 2003년 3월 ~ 2006년 2월 : 청주과학대학 조교수
• 2006년 3월 ~ 현재 : 충주대학교 컴퓨터멀티미디어학과 부교수
• 관심분야 : 상황인식, RTLS, 모바일 시스템

논문접수일 : 2009년 11월 2일
논문수정일 : 2009년 11월 20일
게재확정일 : 2009년 12월 3일