

논문 2009-46CI-4-13

규칙기반 상황인식 모듈관리에 의한 가변위치 부품창고 시스템

(Warehouse System of Parts in Variable Location by Rule-Based Module Management for Context Awareness)

민 드 레*, 전 병 환**

(Deul-Le Min and Byung-Hwan Jun)

요 약

본 논문에서는 자동차 부품과 같이 다량·다종의 부품을 효율적으로 관리할 수 있는 ubiWarehouse 시스템을 제안한다. 먼저, 5WIH 상황표현으로 사용자 작업상황을 체계적으로 정의하고, 상황인식기를 기능별로 구분하여 관리규칙에 의해 각 모듈을 독립적으로 운영한다. 입출고 부품의 선반 적합도를 계산하여 효율적인 이동경로를 선정하고 자주 동시에 취급되는 관련 부품들이 인접할 수 있도록 한다. 이때, 다른 사용자가 현재 방문하고 있거나 바로 방문할 예정인 경우에는 충돌해결을 위해 방문순서를 변경하여 양보한다. 결과적으로, 제안한 시스템은 경쟁을 피하면서도 효율적으로 부품을 입출고할 수 있는 경로를 사용자에게 제공하고, 관련 부품들이 가까운 위치에 모이도록 선반을 배정하여 창고 공간을 효과적으로 활용한다. 또한 입출고 이력을 이용한 사용자별 성향을 파악하여 개인화된 서비스를 제공하고, 실시간적인 입출고 갱신과 경보상황 처리로 재고정보의 정확도를 향상시키는데 기여할 수 있다.

Abstract

In this paper, we propose ubiWarehouse system which can efficiently manages mass and many kinds of parts such as automotive parts. First, user's work situation is defined systematically as context type of 5WIH, context-aware system is separated on function, and each module operates independently. An efficient route is selected and relevant parts frequently treated at the same time are gathered by calculating the suitability of parts and racks. If other users currently are on a visit or are supposed to visit soon, the turn of visit is changed making a concession for preventing collision. As a result, the proposed system with avoiding competition can provide users with routes for inbound or outbound parts, and can effectively use spaces of a warehouse by arranging racks to gather relevant parts to near location. Also, individual service can be offered by evaluating user propensity using history of the warehouse job, and the accuracy of stock information can be improved by processing of unexpected context and real-time renewing of warehousing and delivering.

Keywords : 5WIH, Rule-based module management, Context awareness, ubiWarehouse system

I. 서 론

제조공장이나 물류유통에 사용되는 기존의 창고관리 시스템은 주로 부품별로 보관할 선반을 미리 지정한다. 이 경우 선반의 공간이 부족하거나 잔여공간이 활용되

지 않는 경우가 많고 동시에 취급되는 연관부품들이 멀리 떨어진 선반에 보관되는 경우가 많다. 또한 다수 사용자들이 수시로 입출고 하는 현장에서 정확한 재고정보를 파악하기 어려우며, 특정 선반에 여러 사용자들이 경쟁적으로 접근하려는 경우가 빈번히 발생한다^[1]. 이러한 문제점은 부품의 종류가 많을수록 더욱 심해진다. 예로써, 자동차의 경우에는 2만개 이상의 부품으로 구성되고 차종 및 연식에 따라 다양한 모델이 존재한다^[2]. 최근 정부에 의해 RFID 부착 의무화가 추진되고 있으며^[3], 이에 따라 유비쿼터스 창고(ubiWarehouse)에 관한 연구가 필요하다.

* 학생회원, 공주대학교 컴퓨터공학과
(Dept. of Computer Engineering, Graduate School,
Kongju National University)
** 평생회원, 공주대학교 컴퓨터공학부
(Division of Computer Science and Engineering,
Kongju National University).
접수일자: 2009년2월26일, 수정완료일: 2009년7월3일

1994년 Schilt와 Theimer^[4]는 위치(location), 인접한 사람과 물체의 신원(identity), 그리고 그 물체에 대한 변화(change)로 상황(context)을 최초로 정의하였다. 이후에도 상황을 정의하고자 하는 시도가 지속적으로 있었으나, 주로 특정적인 예나 상황의 유사어로 정의됨으로써 실제 응용에 적용하기 어려웠다. Dey와 Abowd^[5]는 상황을 체계적으로 분류하고 일반화하기 위해, 사용자와 응용간의 상호작용에 연관되는 실체(entity) 즉, 사람, 장소, 사물에 관한 특징 정보로 상황을 정의하였다. 구체적으로는 신원(identity), 시간(time), 위치(location), 행위(activity)를 Who, What, When, Where, How(Act)를 정의하는 주요 상황유형으로 선정하고, 이러한 정보로 왜(Why) 현재 상황이 발생하는지를 파악하도록 하였다.

다수 사용자들이 한정된 자원을 경쟁하는 스마트 공간에서 충돌을 해결하면서 각 사용자의 의도에 맞는 서비스를 실시간으로 제공하기 위해서는 상황인식기가 체계적으로 모듈화되어야 한다.

Context-Toolkit^[6]은 단순히 각 센서로부터 단위 상황을 생성하는 위젯(Widget), 상황을 통합하는 통합기(Aggregator), 그리고 상위 레벨 상황정보로 바꾸어주는 해석기(Interpreter)로 구성된다. SOCAM^[7]은 내부 혹은 외부의 상황제공기(Context Provider), 그리고 상황추론기와 상황 지식베이스로 구성되는 상황해석기(Context Interpreter)로 구성된다. CAMidO^[8]는 센서에이전트들로부터 상황을 수집하는 수집관리기(Collection Manager), 추론기와 해석규칙을 이용하는 상황해석기(Context Interpreter), 관련된 상황 변화를 감지하는 상황해석기(Context Analyser) 등으로 구성된다. ubi-UCAM^[9~10]은 ubiSensor와 ubiService로 구분되고, ubiService는 초벌상황을 수집하는 자기조직 관리기(Self Configuration Manager), 5W1H 상황을 구성하는 상황통합기(Context Integrator), 사용자의 서비스 실행 조건을 해석하는 해석기(Interpreter), 사용자 의도를 파악하고 서비스 충돌을 해결하는 상황관리기(Context Manager), 그리고 서비스제공기(Service Provider)로 구성된다. Chaari 등^[11]은 하위레벨 상황을 공급하는 상황제공기(Context Provider), 상위레벨 상황으로 통합하는 상황해석기(Context Interpreter), 온톨로지에 상황을 맵핑하는 상황관리기(Context Manager), 도메인 특정 규칙을 선택하는 모듈, 추론과 판단을 수행하는 상황인식서비스로 구성된다.

이상과 같이 상황인식기는 기능별로 더욱 세분화되는 추세에 있다. 그러나 모듈제어 방식이 체계적이지 못하고 각 모듈의 독립성이 부족하여, 하나의 모듈이 주어진 상황을 처리하는 동안 다른 대부분의 모듈들이 대기하는 경우가 많다.

본 논문에서는 다수 사용자가 다량·다종종의 부품을 동시에 입출고하는 창고의 특성에 적합하도록, 먼저 육하원칙(5W1H)에 따라 각 사용자의 상황을 표현하고, 규칙기반 모듈관리 방식으로 기능별로 독립된 모듈들을 효율적으로 운영할 수 있는 ubiWarehouse 시스템을 제안한다.

II. ubiWarehouse 시스템

제안하는 ubiWarehouse 시스템은 RFID 방식을 기본으로 채택한다. 따라서 사용자, 부품, 선반에 RFID 태그(tag)를 부착하고, 사용자의 PDA 단말기에는 휴대용 RFID 인식기를 그리고 출입구 단말기에는 고정형 RFID 인식기를 설치한다. 전체 시스템은 그림 1과 같이 PDA 단말기, 출입구 단말기, 관리용 단말기, 그리고 웹기반으로 연결되는 서버로 구성된다.

먼저 PDA 단말기는 RFID 인식기와 사용자 인터페이스(UI)를 센서로 사용하여 각각 초벌 상황데이터(Context Data) CD를 만든 후 통합하여 서버에 전송한다. 유사하게 출입구 단말기는 사용자 혹은 부품을 감지하는 RFID 인식기와 사용자의 이동을 감지하는 CCD 카메라를 센서로 사용하여 상황데이터를 전송한다. 서

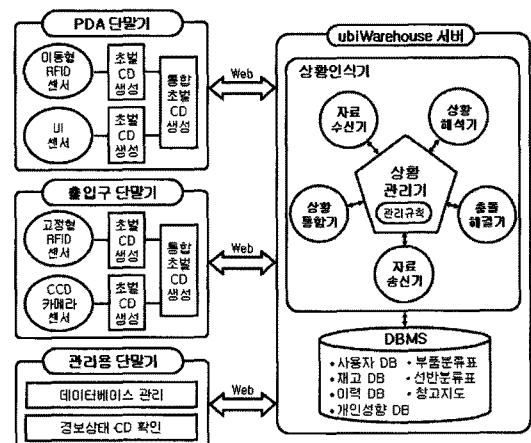


그림 1. ubiWarehouse 시스템 구조

Fig. 1. ubiWarehouse system structure.

표 1. 5W1H 상황 정의
Table 1. 5W1H context definition.

5W1H		정의
Who	사용자 정보	• 사용자 고유번호(U-ID)와 월영사진
When	시간 정보	• 현재시간과 입실 및 퇴실시간
Where	사용자의 위치 정보	• 출입구 또는 선반번호(R-ID)와 방문경로
What	사용자 의도의 대상이 되는 사물 정보	• 부품 모델과 입고/출고, 요구수량, 선택선반
How	사용자의 행위 정보	• 이동 행위: 입실, 퇴실 • 스캔 행위: 사용자스캔, 부품스캔, 선반스캔 • UI 행위: 작업시작, 부품선택, 선반선택, 입출고확인, 작업완료
Why	사용자의 의도 정보	• 작업시작, 부품선택, 선반방문, 입출고, 작업완료

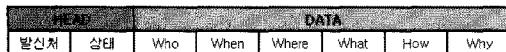


그림 2. 상황데이터(CD) 구조
Fig. 2. Context data structure.

버는 수신된 CD와 데이터베이스를 토대로 상황인식으로 사용자의 의도를 파악하여 서비스한다. 한편 관리용 단말기에서는 각종 DB를 관리하고 서버로부터 전송되는 실수 혹은 절도 등의 경보상황을 확인할 수 있다.

상황(Context)은 표 1과 같이 육하원칙(5WIH)에 따라 정의한다. Who, When, Where, What은 각각 사용자 고유번호, 현재시간, 출입구 또는 선반번호, 부품 모델과 같은 기본정보 이외에 관련된 부가정보가 사용된다. 이때 위치정보는 출입구에서 사용자 태그가 인식되거나 PDA 단말기로 선반 태그를 스캔할 때 취득된다.

그림 2는 실제 시스템에서 취급하기 위한 상황데이터 CD의 형식이며, 단말기와 서버의 모든 과정에서 동일하게 사용된다.

사용자는 PDA 단말기와 서버 간에 지속적으로 생성되는 CD를 주고받으면서 창고작업을 수행하는데, 원활한 정보교환을 위해 5W1H 데이터와 더불어 HEAD에 발신처와 상태정보를 표시한다. 사용자의 명시적 혹은 암시적 요구를 서버의 상황인식기가 파악하여 서비스하는 일반적인 경우에는 일반상태(G)로 CD를 전송한다. 만일 방문할 선반이 모두 사용중일 경우에는 대기상태(W)로 상황데이터 집합에 있게 된다. 또한 출입구에서 감지된 출고부품이 실수나 절도에 의한 것인지를 파악하기 위해 서버에서 사용자에게 문의상태(Q)로 전송할 수도 있다. 만일 이미 완료된 작업이 사용자의 실수로

관명되거나 익명 사용자의 고의적인 절도로 판단된 경우에는 CD를 수정하여 관리자에게 경보상태(A)로 전송한다.

III. 상황인식기

상황인식기는 그림 3과 같이, 관리규칙 집합 S_{Rule} 과 각 모듈의 상태를 나타내는 작업상태표 T_j 를 이용하여 모듈들을 제어하는 상황관리기(Context Manager, CM), PDA 단말기나 출입구 단말기로부터 수신되는 상황데이터 CD를 수신 큐 Q_R 에 저장하는 자료수신기(Data Receiver, DR), 현재 취급중인 임시 상황데이터 CD_T 와 상황데이터 집합에 있는 기존의 CD_k 를 사용자 단위로 통합하는 상황통합기(Context Integrator, CI), 작업시작, 부품선택, 선반방문, 입출고, 작업완료, 경보상황 등의 사용자의 의도를 파악하는 상황해석기(Context Interpreter, CP), 여러 사용자가 동시에 같은 자원 즉, 관심부품이 있는 선반에 밀집하지 않도록 조정하기 위한 충돌해결기(Conflict Resolver, CR), 그리고 사용자의 의도에 맞는 서비스를 제공하기 위해 송신 큐 Q_T 의 CD를 전송하는 자료송신기(Data Transmitter, DT)로 구성된다.

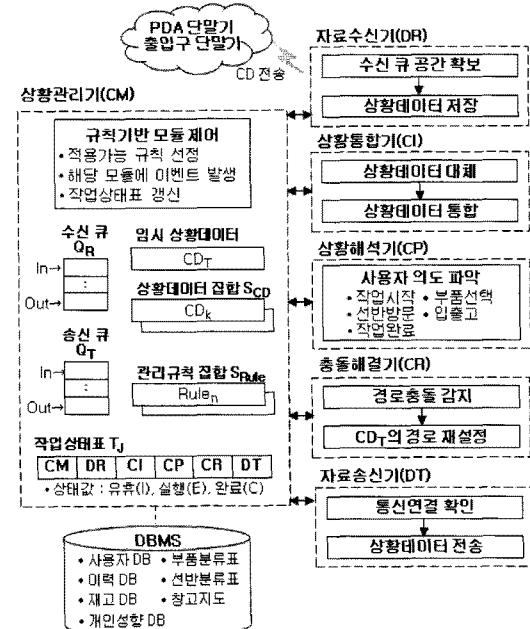


그림 3. 상황인식기 구조
Fig. 3. Context-aware system structure.

1. 규칙기반 모듈관리

여러 사용자를 동시에 효율적으로 서비스하기 위해 서는 상황인식기를 기능별로 모듈화하여 가능한 한 동시에 실행할 수 있도록 해야 한다.

본 논문에서는 관리규칙 집합 S_{Rule} 에서 적용가능한 규칙을 선정하여 적용함으로써 해당 모듈을 실행하고 T_j 를 생성하는 방식을 제안한다. 일반적으로 주어진 CD_T 를 처리하기 위해 자료수신기, 상황통합기, 상황해석기, 충돌해결기, 자료송신기의 순으로 실행되고, 각 모듈의 상태는 실행가능한 상태를 의미하는 유휴상태(I), 실행중임을 의미하는 실행상태(E), 모듈의 작업이 완료되고 통상적인 후속 모듈의 실행이 필요한 상태임을 의미하는 완료상태(C), 그리고 임의의 상태(-)로 나타낸다. 표 2는 모듈관리를 위한 규칙 집합이다.

표 2. 모듈관리를 위한 규칙 집합

Table 2. Rule set for module management.

규칙	조건	실행
1. 자료수신기 실행	• $T_j = (E, I, I, I, I, I)$	• DR 실행
2. 대기 CD 실행	• $T_j = (-, -, I, I, I, -)$ • 대기상태의 CD_k 존재	• CD_k 상태 \leftarrow 일반 • CR 실행
3. 상황통합기 실행	• $T_j = (-, -, I, I, I, -)$ • 대기상태의 CD_k 없음	• $CD_T \leftarrow pop(Q_R)$ • CI 실행
4. 상황해석기 실행	• $T_j = (-, -, C, I, I, -)$	• CI 유휴 • CP 실행
5. 문의 CD 처리	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 문의	• CP 유휴 • DT 실행
6. 경보 CD 처리	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 경보	• insert(이력DB, CD_T) • CP 유휴 • DT 실행
7. 작업시작 의도	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 일반 • CD_T .Why = 작업시작	• insert(S_{CD} , CD_T) • CP 유휴 • DT 실행
8. 부품선택 의도	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 일반 • CD_T .Why = 부품선택	• CP 유휴 • CR 실행
9. 선반방문 의도	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 일반 • CD_T .Why = 선반방문	• insert(S_{CD} , CD_T) • CP 유휴 • DT 실행
10. 입출고 의도	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 일반 • CD_T .Why = 입출고	• update(재고DB, CD_T) • CP 유휴 • CR 실행
11. 작업완료 의도	• $T_j = (-, -, I, C, I, -)$ • CD_T .상태 = 일반 • CD_T .Why = 작업완료	• insert(이력DB, CD_T) • CP 유휴 • DT 실행
12. 자료송신기 실행	• $T_j = (-, -, I, I, C, -)$	• insert(S_{CD} , CD_T) • CR 유휴 • DT 실행
13. 시스템 종료	• $T_j = (I, E, -, -, -, -)$	• DR 유휴

2. 상황통합 과정에서의 보안처리

PDA 단말기나 출입구 단말기로부터 서버에 수신되는 상황데이터 CD_T 와 사용자 고유번호(U-ID)가 명시된 기존의 상황데이터 CD_k 의 통합과정은 표 3과 같다.

먼저, 출입구에서 전송되는 CD_T 를 고려해보자. 사용자 고유번호가 인식된 경우에는 현재 작업중인 사용자의 CD 집합인 S_{CD} 나 작업을 완료한 사용자의 이력 DB에서 동일한 U-ID의 CD_k 를 찾아 통합하면 된다. 그러나 사용자 고유번호가 인식되지 않은 경우에는 U-ID를 알 수 없으므로, 창고작업을 시작하기 위해 입실하거나 끝마치기 위해 퇴실하는 사용자를 찾아 통합해야 한다. 이때 CCD 카메라의 비전센서에 의해 파악된 이동방향으로 입실과 퇴실을 구분한다.

특히, 사용자의 실수나 고의적인 절도로 부품이 출고되는 것을 감지하기 위해서는 부품을 갖고 퇴실하는 사용자에 의해 출입구에서 감지된 CD_T 에 대한 처리가 중요하다. 이때 창고 안에서 이미 UI로 작업완료를 하고 퇴실하는 사용자의 상황데이터 CD_k 는 이력 DB에 존재하고, 입출고 작업을 마쳐서 방문예정 선반이 남아있지 않는 상태에서 완료명령 없이 퇴실하는 사용자의 CD_k 는 S_{CD} 에 존재한다.

한편, PDA로부터 발신되는 CD_T 는 로그인 사용자의 고유번호가 항상 명시되어 있으므로 통합하기 쉽다. 일

표 3. 상황데이터의 통합방법

Table 3. Integration method of context data.

CD_T 구분		내용
PDA 발신		<ul style="list-style-type: none"> 일반상태의 경우: CD_k를 CD_T로 대체 문의상태의 경우 <ul style="list-style-type: none"> 사용자 승인: CD_k를 CD_T로 대체 사용자 거부: 문의의 CD를 '경보상태'로 설정
출입구	Who 명시 퇴실	<ul style="list-style-type: none"> CD_T에 CD_k를 통합 S_{CD}에 CD_k가 존재하는 경우 <ul style="list-style-type: none"> CD_T의 부품이 CD_k에 존재하면 통합 그렇지 않으면 부품확인을 위해 '문의상태' 설정 최근 이력DB에 CD_k가 존재하는 경우 <ul style="list-style-type: none"> CD_T의 부품이 CD_k에 존재하면 확인완료 그렇지 않으면 '경보상태'로 설정 CD_k가 존재하지 않는 경우 <ul style="list-style-type: none"> CD_T에 부품이 존재하면 '경보상태'로 설정
		<ul style="list-style-type: none"> S_{CD}에 최근에 시작한 CD_k가 존재하면 통합 그렇지 않으면 CD_T 삭제
발신	Who 불명 퇴실	<ul style="list-style-type: none"> S_{CD}에 최근에 CD_T의 부품이 있는 CD_k가 존재하면 확인완료 S_{CD}에 잔여 입출고작업이 없고 CD_T의 부품이 있는 CD_k가 존재하면 작업완료 확인을 위해 '문의상태'로 설정 CD_k가 존재하지 않으면 '경보상태'로 설정

반적으로 PDA와 서버간에 전송되는 상황데이터는 해당 사용자의 창고작업에 대한 최신정보를 유지하므로 CD_k 를 CD_T 로 대체하면 된다. 반면 사용자가 누락한 출고부품이나 자동적인 작업완료를 확인하기 위해 서버가 PDA에 문의상태로 상황데이터를 전송할 수 있다. 이때 사용자가 승인한 경우에는 CD_T 로 대체하지만, 거부한 경우에는 원래의 문제 CD 를 경보상태로 설정하여 추후 관리자에게 통지될 수 있도록 한다.

3. 가변적인 부품위치 기반의 상황해석

상황해석기에서 파악되는 사용자의 의도(Why)는 주로 행위(How) 정보를 토대로 파악되며, 이에 관련된 서비스가 제공된다.

그림 4와 같이, 사용자의 행위는 크게 이동행위, 스캔행위, UI행위로 구분되며, 각 행위에 관련된 사용자의 의도를 화살표로 연결하고 있다. 여기서 점선은 부차적인 의도를 나타낸다. 예로써, 사용자가 입실하면 작업이 시작될 뿐만 아니라 최근의 취급빈도에 따른 사용자의 부품 선호도를 토대로 입출고용 부품목록을 제시할 수 있다. 부품 선호도는 창고작업이 완료되어 CD 가 이력 DB에 저장될 때마다 갱신된다. 또한, 사용자의 서로 다른 행위가 동일한 의도에서 비롯될 수 있다. 예로써, PDA 단말기에서 작업시작을 명령하지 않아도 입실하는 과정에서 사용자 태그가 감지되거나 PDA 단말기로 사용자 자신의 태그를 스캔하면 창고작업이 시작된다.

부품을 스캔하거나 선택하면 부차적으로 선반방문

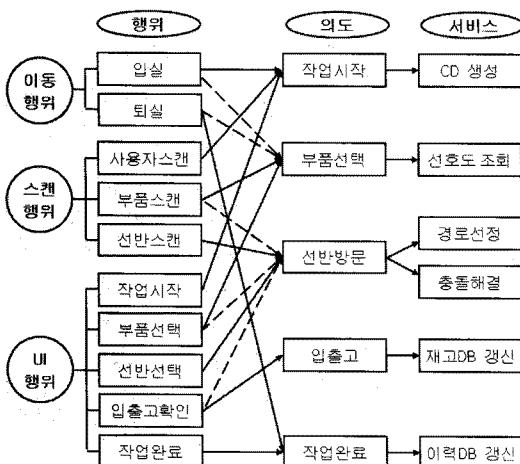


그림 4. 사용자의 행위, 의도, 서비스간의 관계
Fig. 4. Relation among action, intention, and service of users.

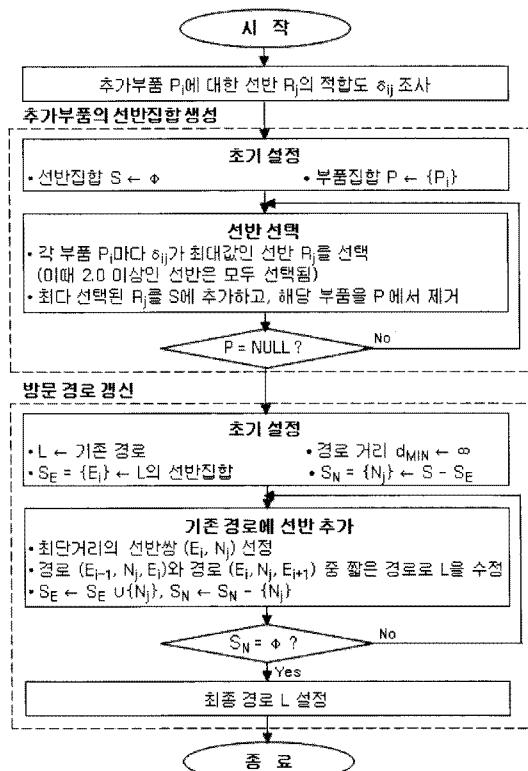


그림 5. 부품의 선반 적합도에 의한 경로선정
Fig. 5. Routing by rack fitness for parts.

의도가 파악되어 해당 부품을 입출고하기에 적합한 선반을 찾아 경로를 선정한다. 그러나 현재 선반에서 입출고를 마치고 확인버튼을 눌렀을 때, 방문할 다음 선반에서 다른 사용자와의 충돌이 예상된다면 이를 회피하기 위해 다음 절에서 소개하는 충돌해결을 수행한다.

한 종류의 부품이 가급적 동일한 선반에 밀집하고, 같은 선반에서 여러 부품을 입출고하며, 사용자의 이동거리를 최소화하기 위해, 그림 5와 같은 경로생성 알고리즘을 제안한다.

먼저, 추가부품 P_i 와 임의의 선반 R_j 의 적합도 δ_{ij} 는 식 (1)과 같다.

$$\delta_{ij} = \delta_{ij}^A + \delta_{ij}^B \quad (1)$$

$$\delta_{ij}^A = \begin{cases} \frac{Q_R}{Q_H} + 1, & Q_H \geq Q_R \\ \frac{Q_H}{Q_R}, & Q_H < Q_R \end{cases} \quad (2)$$

$$\delta_{ij}^B = \begin{cases} 1, & Q_H > 0 \\ 0, & Q_H = 0 \end{cases} \quad (3)$$

여기서 보유량 Q_H 와 요구량 Q_R 은 입고할 부품의 보관공간이나 출고부품의 수량을 의미한다. 식 (2)의 δ_{ij}^A 는 보유량 Q_H 와 요구량 Q_R 이 일치하는 정도이고, 식 (3)의 δ_{ij}^B 는 같은 종류의 부품 P_i 가 선반 R_j 에 이미 존재하는지의 여부이다. 적합도 δ_{ij} 는 0~3의 값을 가진다. 예로써, 적합도가 2~3이면 관심부품이 이미 존재하고 보유량이 사용자의 요구량을 만족시키는 선반임을 나타낸다. 다음으로 각 부품 P_i 마다 적합도가 최대인 선반을 선택하되 δ_{ij} 가 2.0 이상인 선반은 모두 선택한다. 그리고 추가 부품들이 모두 해결될 때까지 최대인 선반을 으로 선반을 선정해나간다. 끝으로 이와 같이 선정된 선반들을 기준 경로에 추가하는데, 기존 선반과의 거리가 가장 가까운 신규 선반부터 추가한다.

4. 양보방식의 경로변경에 의한 충돌해결

사용자가 현재 선반에서 부품 입출고를 완료하면 작업할 다음 선반으로 이동해야 한다. 그러나 예정된 다음 선반이 다른 사용자에 의해 사용중이거나 곧 사용될 예정일 수 있다. 따라서 본 논문에서는 그림 6과 같이

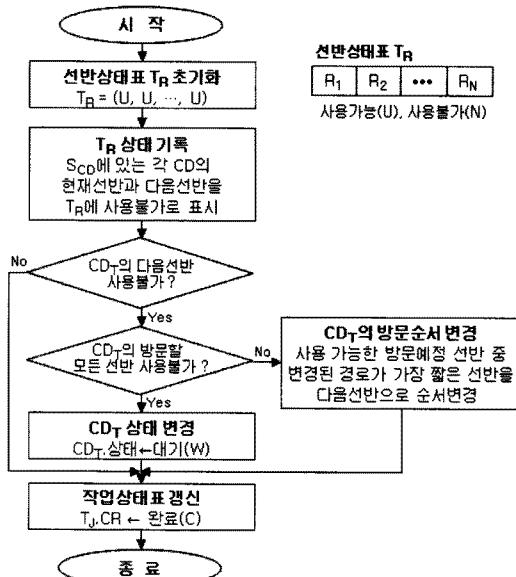


그림 6. 충돌해결을 위한 경로변경
Fig. 6. Route change for conflict resolution.

부품이라는 자원을 취급할 수 있는 업무공간의 중첩을 피하기 위해 경로를 변경함으로써 충돌을 해결한다. 이 때, 실시간으로 충돌을 해결하기 위해 이전 경로에 선정된 선반을 그대로 유지하되 방문순서만을 변경한다.

먼저, 선반상태표 T_R 에 작업중인 다른 사용자들의 현재 선반과 바로 다음에 방문할 선반을 사용불가(N)로 표시한다. 만일, 사용자의 다음 선반이 사용불가(N)이지만 방문예정 선반 중에 사용가능한 선반이 존재한다면, 앞서 소개한 경로선정 알고리즘에서 신규 선반을 추가하는 방식으로 경로를 변경한다.

IV. ubiWarehouse 시스템의 구현 및 분석

1. 제안하는 시스템의 구성 및 개발환경

본 논문에서 제안하는 시스템은 그림 7과 같이, 이동형 RFID 인식기가 탑재된 PDA 단말기, CCD 카메라와 고정형 RFID 인식기가 연결된 출입구 단말기, 서버로부터의 경보상황을 접수하고 DB를 관리하기 위한 관리용 단말기, 그리고 상황인식기와 DBMS를 운영하는 웹 기반 서버로 구성된다.

제안하는 ubiWarehouse에서는 사용자, 부품, 선반에 RFID 태그가 부착된다. 사용자가 입고할 부품을 소지하고 창고에 입실하면, 출입구 단말기의 CCD 카메라에서 포착된 영상과 진행방향 정보, 그리고 고정형 RFID 인식기에서 감지된 사용자와 부품의 RFID 태그 정보가 서버에 전송된다. 서버가 이를 토대로 생성된 CD를 사

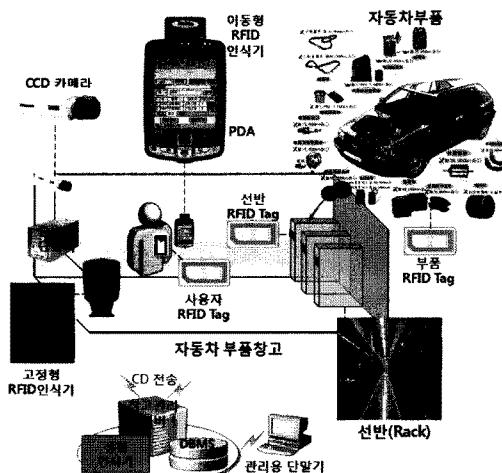


그림 7. 제안하는 시스템의 구성
Fig. 7. Configuration of the proposed system.

표 4. 시스템 개발환경

Table 4. System development environment.

구분	H/W 및 OS	S/W
PDA	<ul style="list-style-type: none"> • HP iPAQ hx2790 624MHz • Windows Mobile 5.0 • RFID Reader Hand IT-2G 	<ul style="list-style-type: none"> • .NET Compact Framework 2.0 • RFID Driver Mx Pocket PC
PDA Emulator	<ul style="list-style-type: none"> • Pentium IV 3.0GHz • Windows XP 	<ul style="list-style-type: none"> • .NET Framework 2.0 • .NET Based C#
출입구 단말기	<ul style="list-style-type: none"> • CCD Camera SDC-410 • RFID Reader D800H • Pentium IV 3.0GHz • Windows XP 	<ul style="list-style-type: none"> • RFID Driver Inside 3Mx • .NET Based C#
관리자 단말기	<ul style="list-style-type: none"> • Pentium IV 3.0GHz • Windows XP 	<ul style="list-style-type: none"> • ASP.NET
시스템 서버	<ul style="list-style-type: none"> • Pentium IV 3.0GHz • Windows Server 2003 	<ul style="list-style-type: none"> • Microsoft SQL Server 2005 • .NET Based C#

용자의 PDA 단말기에 전송하면, 사용자는 누락된 입고부품을 스캔하여 추가하고 출고할 부품을 검색하여 등록한다. 그러면 서버는 사용자의 입출고 부품에 대한 효율적인 경로를 설정하여 사용자에게 안내한다. 사용자가 해당 장소에 도착하여 선반 RFID 태그를 스캔하면, 서버는 재고 DB를 갱신하고 부품의 입출고 완료를 사용자에게 확인을 받는다. 또한 다음에 방문할 선반에 다른 사용자에 의해 사용중이거나 사용예정이면 경로를 변경하여 안내한다.

사용자가 작업을 마치고 퇴실할 때, 출입구 단말기에서는 사용자 또는 출고부품 태그를 감지할 수 있다. 만일 사용자 태그가 감지된 경우에는 작업완료나 실수로 누락된 출고부품의 처리를 쉽게 서비스할 수 있다. 반면, 출고부품만 감지된 경우에는 최근에 작업완료 버튼을 누르고 퇴실하는 사용자(해당 CD가 iLCD에 존재)나 입출고할 부품이 남아있지 않은 사용자(해당 CD가 SCD에 존재)를 대상으로 그 부품이 출고부품 목록에 존재하는지를 검사하고 필요한 경우에 유력한 사용자에게 문의한다. 만일 확인되지 않은 출고부품이 존재하면, 서버는 절도로 판단하여 관리자에게 경보한다.

ubiWarehouse 시스템의 구현을 위한 개발환경은 표 4와 같다.

2. 제안하는 시스템의 분석 및 평가

가. 규칙기반 모듈관리의 효율성

제안하는 시스템에서는 상황인식기를 기능별로 모듈화하고 상황관리기에 의해 각 모듈의 수행을 관리한다. 이때 단순한 구조의 규칙을 사용함으로써 각 모듈을 체계적으로 관리할 수 있고, 규칙의 수정 및 추가로 관리

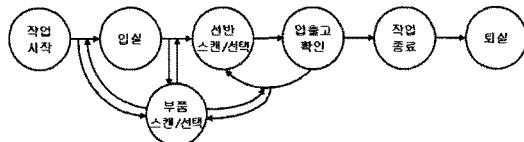


그림 8. 통상적인 창고작업에서 CD 전송을 일으키는 행위

Fig. 8. Actions for CD transmission in general warehouse job.

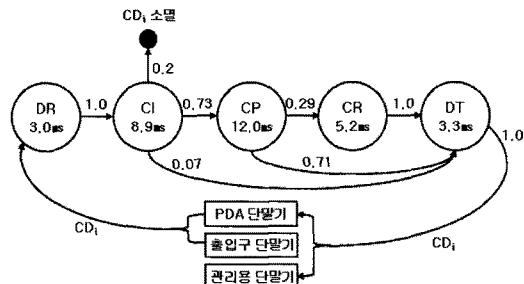


그림 9. 상황인식기 내부의 CD 진행과정

Fig. 9. CD courses in context-aware system.

방식을 간편하게 수정 및 확장할 수 있다. 또한, 여러 모듈이 동시에 독립적으로 수행됨으로써, 각 모듈의 유휴시간이 감소되고 여러 사용자의 창고작업을 보다 신속하게 서비스할 수 있다.

이를 단말기와 서버간의 CD 교신에 따른 각 진행과정을 분석하여 비 모듈화 방식과 제안하는 규칙기반 모듈관리 방식의 성능을 비교해보자. 먼저, 통상적인 작업과정에서 서버로의 CD 전송을 일으키는 사용자의 행위는 그림 8과 같다. 여기서 알 수 있듯이, 창고작업은 통상적으로 7회 이상의 단말기와 서버간의 CD 교신을 수행하며, 취급 부품에 따른 방문선반 수에 비례하여 CD 교신 횟수도 증가한다.

그림 9는 단말기와 서버간의 CD 교신주기에서 상황인식기 내부의 가능한 진행과정을 나타낸다. 여기에 각

표 5. 상황인식기의 운영방식에 따른 처리성능

Table 5. Performance according to operating type of context-aware system.

CD의 각 진행과정	진행과정 발생률	소요시간	
		비 모듈화 방식	제안하는 모듈관리 방식
• DR-CI	0.20	2.38	1.78
• DR-CL-DT	0.07	1.06	0.62
• DR-CL-CP-DT	0.52	14.10	10.83
• DR-CL-CP-CR-DT	0.21	6.86	5.53
평균	-	6.10	4.69

모듈의 평균 처리시간과 각 모듈에서 다음으로 진행 가능한 스텝(아크)의 평균 발생률이 표시되어 있다.

표 5는 그림 9를 토대로, 가능한 진행과정경로에 대해 비 모듈화 방식과 제안하는 모듈관리 방식의 처리시간을 추정한 것이다. 이때 상황관리기(CM)는 비 모듈화 방식의 경우에도 나머지 모듈에 분할되어 거의 그대로 존재하기 때문에 이의 처리시간은 고려하지 않는다.

여기서, 각 CD 진행과정의 발생률은 그 과정에 존재하는 모든 진행 스텝의 발생률을 곱한 것이다. 비 모듈화 방식의 경우에는 진행과정의 각 모듈이 독립적이지 않기 때문에 동시에 수행될 수 없다. 따라서 각 진행과정의 소요시간은 그 과정에 있는 모듈의 평균소요시간의 합에 그 과정의 발생률을 곱하여 계산한다. 한편, 제안하는 모듈관리 방식의 경우에는 CI와 CR이 직접 취급하는 CD_T 이외에도 나머지 다른 CD와 통합 혹은 참조하기 때문에, 부분 과정 CI-CP나 CI-CP-CR은 더 이상 분할되어 독립적으로 수행되기 어렵다. 따라서 제안하는 모듈관리 방식에서는 독립적으로 중첩하여 수행가능한 DR과 DT의 처리시간을 제외하고 평균 소요시간을 추정할 수 있다.

한 가지 주목할 점은, CI는 현재 취급하는 CD_T 와 다른 CD를 통합할 가능성이 있으므로 CP나 CR과 독립적으로 수행할 수 없다. 그러나 CR의 경우에는 다른 CD를 참조만 하기 때문에, CD_T 자체만으로 수행되는 CP와 독립적으로 수행될 수 있다는 사실이다. 이를 토대로 상황인식기의 모듈을 보다 독립화하여 효율적으로 운영하기 위해서는 관리규칙의 추가와 다중 CD_T 의 취급에 관한 연구가 요구된다.

나. 부품 입출고의 정확도 향상과 보안

창고에 입실할 때 감지되거나 선호도에 따라 추천된 부품은 반드시 사용자의 확인을 거침으로써 정확성을 기한다. 또한, 퇴실할 때 감지된 부품이 사용자의 실수나 고의적인 절도로 판단되면, 사용자에게 문의하여 확인하거나 관리자에게 통지함으로써 입출고 업무의 정확성을 향상시키고 보안서비스를 제공한다.

출입구에서 사용자에 의해 출고되는 부품이 정당한지는 III장 2절에서 소개한 상황데이터 통합과정에 의해 파악된다.

그림 10에서 퇴실하는 사용자 U_0 은 입출고 작업을 마치고 바로 퇴실하는 U_1 일수도 있고(이때 U_1 의 CD_1 은 SCD 에 존재), 혹은 퇴실 전에 작업완료 명령을 하고 나

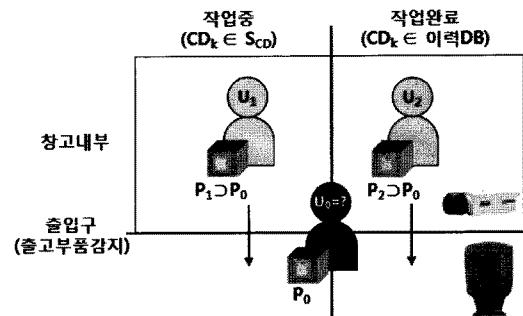


그림 10. 퇴실 사용자의 작업 상태
Fig. 10. Job state of leaving user.

오는 U_2 일 수도 있다(이때 U_2 의 CD_2 는 이력DB에 존재).

먼저, 사용자 U_0 의 ID가 함께 감지되어 누구인지 아는 경우에는, 상황데이터 집합 SCD 나 이력 DB에서 동일한 사용자의 CD를 찾고, 감지된 출고부품이 모두 존재하는지를 검사한다. 만일 일치하는 사용자가 작업완료 처리되지 않은 U_1 인 경우에는 CD_1 을 수정 및 전송하여 사용자의 실수를 처리할 수 있도록 지원한다. 이와 달리 이미 작업완료 처리된 U_2 인 경우에는 이력 DB에 저장되어 있는 CD_2 를 경보상태로 변경하고 관리자에게 이 사실을 통보한다.

한편, 사용자 U_0 이 불명인 경우에는 감지된 출고부품을 포함하는 상황데이터를 이력 DB에서 찾아 그 사용자(U_2 에 해당)의 작업완료 사실을 확인하면 된다. 만일 일정기간 이내에 해당 CD를 찾을 수 없다면, SCD 에서 입출고 작업이 남아있지 않고 해당 출고부품을 포함하는 CD를 찾아 그 사용자(U_1 에 해당)의 확인을 거쳐 작업완료 처리를 서비스한다. 그러나 해당 CD를 이력 DB나 SCD 에서 찾을 수 없는 경우에는 절도로 판단하여 현재 CD를 경보상태로 저장하고 관리자에게 통지한다.

다. 가변위치 부품의 경로선정

부품에 대한 선반의 적합도에 따라 입고 혹은 출고할 선반을 선택한 후 최단의 경로를 선정함으로써, 같은 부품뿐만 아니라 자주 동시에 취급되는 부품들이 동일한 혹은 인접한 선반에 배치되고 사용자의 이동거리도 단축된다.

표 6은 예로 든 입출고 부품에 대한 후보선반의 적합도를 구한 것이다. 부품에 대한 선반 적합도와 경로선정 과정은 III장 3절에 소개되어 있다.

각 부품에 대한 선반 적합도(0.0~3.0)는 해당 부품이 존재하는 선반인지, 요구량을 충족시키는지에 따라 1.0

표 6. 입출고 부품에 대한 선반의 적합도 예
Table 6. Example of rack suitability for parts.

부품		선반				
부품명	입출고	요구량	선반	보유수량	잔여공간	적합도
엔진오일	출고	17개	R ₁	32개	-	2.5
			R ₄	181개	-	2.1
오일필터	입고	(25×13×30)×10개 = 97500cm ³	R ₄	32개	487500cm ³	2.2
			R ₅	0개	191400cm ³	1.5
타이밍벨트	출고	37개	R ₆	16개	-	1.4
			R ₁₀	26개	-	1.7
에어클리너	출고	17개	R ₁₃	57개	-	2.3
선정된 선반집합: (R ₄ , R ₆ , R ₁₀ , R ₁₃)						

표 7. 방문경로 탐색의 예
Table 7. Example of visiting route searching.

단계	현재 방문경로	추가 선반집합	최단 선반쌍 (기준, 추가)
1	<G>	(R ₄ , R ₆ , R ₁₀ , R ₁₃)	(G, R ₄)
2	<G, R ₄ >	(R ₆ , R ₁₀ , R ₁₃)	(G, R ₆)
3	<G, R ₄ , R ₆ >	(R ₁₀ , R ₁₃)	(R ₄ , R ₁₀) <G, R ₁₀ , R ₄ > > <R ₄ , R ₁₀ , R ₆ >
4	<G, R ₄ , R ₁₀ , R ₆ >	(R ₁₃)	(R ₆ , R ₁₃)
5	<G, R ₄ , R ₁₀ , R ₆ , R ₁₃ >	Ø	-
최종	<G, R ₄ , R ₁₀ , R ₆ , R ₁₃ , G>	-	-

이상 혹은 2.0 이상의 값을 갖는다. 엔진오일은 잔여 수량이 보다 적어서 적합도가 더 높은 선반 R₁뿐만 아니라 2.0 이상인 R₄도 함께 후보로 추천한다. 오일필터는 해당 부품을 보유하고 있고 잔여공간이 충분한 R₄를 추천한다. 이때, 해당 부품이 존재하지 않는 R₅도 고려되는 것을 알 수 있다. 타이밍벨트는 요구수량을 충족하는 선반이 없지만 가장 많이 보유하는 R₁₀을 추천하고, 에어클리너는 유일한 보유선반 R₁₃을 추천한다. 이러한 과정에서 가장 많이 추천된 R₄가 방문선반 중의 하나로 결정되고 엔진오일과 오일필터의 입출고가 해결된다. 이상의 과정을 남은 관심부품들에 대해 반복하여 최종적으로 R₄, R₆, R₁₀, R₁₃이 방문선반으로 결정된다.

표 7은 최종 선반집합 {R₄, R₆, R₁₀, R₁₃}에 대해 방문 경로를 구하는 과정을 보여준다. 일반적으로 창고작업 도중에도 입출고할 부품을 추가할 수 있으며, 여기서는 현재 출입구(G)에서 부품을 추가한다고 가정한다.

현재 방문경로의 선반들과 추가할 선반집합의 선반 간의 최단거리 선반쌍 (R_i, R_j)를 구하고 두 선반을 연결함으로써 선반을 추가한다. 이때, 단계 3에서와 같이 기존 경로의 중간에 있는 선반 R_i에 추가선반 R_j를 연결할 때에는 경로 <R_{i-1}, R_j, R_i>와 <R_i, R_j, R_{i+1}> 중에서 짧은 것을 취한다. 그림 11은 단계 3에서 경로 <G, R₄, R₆>에 R₁₀이 추가되어 경로 <G, R₄, R₁₀, R₆>으로

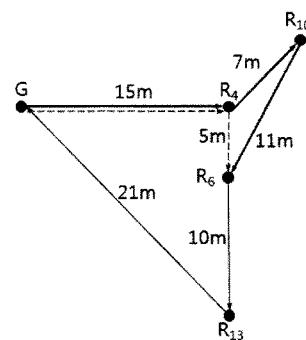


그림 11. 방문경로 탐색의 한 장면
Fig. 11. A scene of visiting route searching.

변경되는 장면을 보여준다.

그러나 동일한 부품이 한곳에만 적재되면 사용자들이 자주 대기할 수 있으므로, 다수 사용자가 선호하는 부품은 분산하여 배치할 수 있도록 보완되어야 한다.

라. 양보방식의 충돌해결

충돌해결기는 바로 다음에 방문할 선반에서 다른 사용자와 작업공간의 중첩이 발생할 수 있는지를 실시간으로 점검한다. 이때, 양보정책을 사용하지만 사용자 자신은 이를 인지할 필요 없이 자원에 대한 경쟁이 없는 경로로 안내된다.

그림 12는 앞 절에서 예로 든 방문경로 <G, R₄, R₁₀, R₆, R₁₃, G>로 진행하다가, 선반 R₄에서 입출고를 마쳤을 때 다음에 방문할 선반 R₁₀에서 충돌이 예측되어 가능한 최단 경로 <G, R₄, R₆, R₁₀, R₁₃, G>로 변경되는 장면이다.

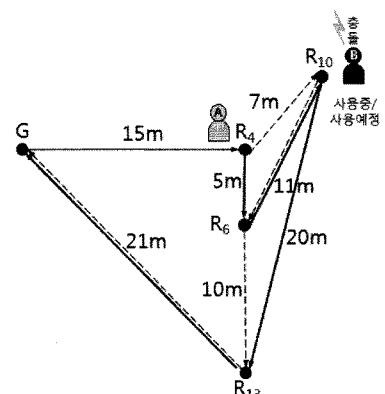


그림 12. 충돌해결을 위한 경로변경의 한 장면
Fig. 12. A scene of route changing for conflict resolution.

V. 결 론

본 논문에서는 자동차 부품과 같이 다양·다종종의 부품을 체계적으로 관리하고 사용자의 창고작업을 효율적으로 지원하기 위한 ubiWarehouse 시스템을 제안하였다. 즉, 5W1H 상황유형에 따라 작업상황을 정의하고, 기능별로 모듈화된 상황인식기를 독립적으로 운영하기 위해 관리규칙을 사용한다.

결론적으로, 사용자에게 충돌이 없는 효율적인 이동 경로를 제공하고, 연관성이 높은 부품들이 인접함으로써 가변위치 기반의 창고공간을 효과적으로 활용할 수 있다. 또한 개방된 환경에서 실수나 절도를 실시간으로 감지함으로써 재고정보의 정확도를 향상시킬 수 있다.

향후에는 각 단말기 자체에도 상황인식기가 탑재되어 에이전트화 하는 연구가 필요하고, 창고와 더불어 공장이나 사무실과 같은 여러 공간 사이에 연계될 수 있도록 시스템이 확장되어야 한다.

참 고 문 헌

- [1] 민들레, 전병환, “다종종 자동차 부품관리를 위한 ubiWarehouse 시스템”, 대한전자공학회 학술대회논문집, 제31권 제1호, 725-726쪽, 2008년 6월
- [2] 최진열, 자동차부품산업, 하나경제연구소, 2005년
- [3] *Design u-World RFID/USN*, 한국정보사회진흥원, 2008년
- [4] B. Schilit and M. Theimer, “Disseminating Active Map Information to Mobile Hosts,” *IEEE Network*, Vol. 8, no. 5, pp. 22-32, 1994.
- [5] A. K. Dey and G. D. Abouwd, “Towards a Better Understanding of Context and Context Awareness,” Proc. of CHI 2000: Workshop on the What, Who, Where, When, and How of Context-Awareness, Apr. 2000.
- [6] D. Salber, A. K. Dey and G. D. Abouwd, “The Context Toolkit: Aiding the Development of Context-Aware Applications,” In the Workshop on Software Engineering for Wearable and Pervasive Computing, Jun. 2000.
- [7] T. Gu, H. K. Pung and D. Q. Zhang, “Toward an OSGi-Based Infrastructure for Context-aware Applications,” *IEEE Pervasive Computing*, Vol. 3, no. 4, pp. 66-74, Oct.-Dec. 2004.
- [8] N. B. Behloui, C. Taconet and G. Bernard, “An Architecture for Supporting Development and Execution of Context-Aware Component Applications,” In IEEE International Conference on Pervasive Services, ICPS 2006, Lyon, France, pp. 57-66, Jun. 2006.
- [9] Y. Oh, C. Shin, S. Jang and W. Woo, “ubi-UCAM 2.0: A Unified Context-Aware Application Model for Ubiquitous Computing Environments,” The First Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems, 2005.
- [10] 서영정, 박영민, 윤효석, 우운택, “유비쿼터스 환경에서 개인화된 스마트 오브젝트 제어 및 미디어 콘텐츠 제공을 위한 맥락 인식 모바일 중장 협실 시스템,” 전자공학회논문지, 제44권 CI편, 제3호, 57-67쪽, 2007년 5월
- [11] T. Chaari, D. Ejigu, F. Laforest and V. M. Scuturici, “A Comprehensive Approach to Model and Use Context for Adapting Applications in Pervasive Environments,” *The Journal of System and Software*, Vol. 80, pp. 1973-1992, Mar. 2007.

저 자 소 개



민들레(학생회원)

2007년 대전대학교 컴퓨터공학과 학사 졸업.

2009년 공주대학교 컴퓨터공학과 석사.

<주관심분야 : 유비쿼터스 컴퓨팅, 컴퓨터비전>



전병환(평생회원)-교신저자

1989년 연세대학교 전자공학과 학사 졸업.

1991년 연세대학교 전자공학과 석사 졸업.

1996년 연세대학교 전자공학과 박사 졸업.

1997년 ~ 현재 공주대학교 컴퓨터공학부 교수.

2000년 ~ 2001년 쭈모리아테크놀로지 연구소장.

<주관심분야 : 컴퓨터비전, 가상현실, 유비쿼터스 컴퓨팅>