韓國컴퓨터情報學會 論文誌
第16卷 第6號, 2011. 6.

2011-16-6-4-1

선박 건조공정 개선을 위한 상황인지 컴퓨팅 기반의 강재적치처리시스템

강 동 훈*, 하 창 완**, 김 제 욱***, 오 훈****

Context-Aware Steel-Plate Piling Process System For Improving the Ship-Building Process

Dong-Hoon Kang*, Chang-Wan Ha**, Je-Wook Kim***, Hoon Oh****

요 약

선박 건조공정의 초기 단계인 강재적치처리공정에서는 입수한 강재를 어느 선박블록을 제작하는데 사용될 것인가에 따라 수십 개의 적치장을 이용하여 분류한다. 이 과정에서 강재를 잘못된 적치장에 적치하는 적치 오류, 무거운 강재를 다루는데 수반되는 작업자 안전 취약성, 각 적치장의 강재 정보 관리의 미비로 인한 작업 계획의 불확실성 등으로 인하여 생산성이 크게 저하된다. 이러한 문제점을 해결하기 위하여 상황인식 컴퓨팅 기술을 적용하여 지능형 강재적치처리시스템을 구축하였다. 연구의 효율성을 위하여 강재적치처리 작업과정을 보여줄 수 있는 시뮬레이터를 제작하였으며, 센서, RFID 태그, 실시간 위치추적시스템과 같은 기술을 활용하여 공간 내에 있는 객체들의 위치 및 작업 처리 상태 등, 상황정보를 실시간으로 획득할 수 있도록 유비쿼터스 컴퓨팅 공간을 구축 하였다. 본 연구는 유비쿼터스 컴퓨팅 기술을 적용한 선박 건조 생산성 제고는 물론 대규모 생산현장에 IT를 적용하는 제조IT 융합연구의 새로운 모델을 제시하였다는데 의미가 크다고 할 수 있다.

▸ Keyword : 선박 건조 과정, 상황인지컴퓨팅, 지능형 공간, 규칙 기반 정보.

Abstract

A gigantic ship is constructed by assembling various types of ship blocks, each block being made by cutting and piecing the steel-plates together. The steel-plate piling process as the initial stage

• 공동저자 : 강동훈, 하창완, 김제욱 교신저자 : 오 훈
• 투고일 : 2010. 10. 08, 심사일 : 2010. 11. 01, 게재확정일 : 2010. 12. 10
* 사우스다코다대학 석사과정 (Dept. of Electrical Engineering and Computer Science, South Dakota State University, U.S.A)
** 울산대학교 석사과정 (School of Electrical Engineering, University of Ulsan)
*** 울산대학교 박사과정 (School of Electrical Engineering, University of Ulsan)
**** 울산대학교 전기공학부 교수 (School of Electrical Engineering, University of Ulsan)

of ship construction sorts and manages the steel-plates according to the ship blocks that the steel-plates are used to make. The steel-plate piling process poses some problems such as process delay due to piling errors, safety vulnerability due to the handling of extra heavy-weight objects, and the uncertainty of work plan due to lack of information management in the pile spaces. We constructed a steel-plate piling process system based on the context-aware computing to resolve such problems. We built simulation system that can simulate the piling process and then established a smart space within the system by using tags, sensors and a real-time location system in order to collect context information. Workers receive an appropriate or intelligent service from the system.

▸ Keyword : Ship-building Process, Context-Aware Computing, Smart Space, Rule-cased Knowledge

I. 서 론

최근에 생산현장, 건설현장, 의료분야, 교육분야 등을 포함하는 다양한 분야에 유비쿼터스 컴퓨팅 기술의 활용에 대한 관심이 크게 증가하고 있다. 교량의 유지보수 미비로 인하여 발생하는 사고를 예방하기 위하여 교량의 주요 부분을 주기적으로 모니터링하고 이상적 현상이 감지되는 경우에 적절한 경고체계를 가동시키는 교량 모니터링 시스템 [1], 시간, 장소에 구애됨이 없이 사용자의 상태를 모니터링하고 의료 서비스를 제공하는 의료서비스 분야 [2], RFID를 이용하여 기록물 관리, 이력 추적, 보존 및 검색 등을 용이하게 하는 기록물 관리시스템 [3] 등은 유비쿼터스 컴퓨팅의 응용분야 중의 하나이다. 본 논문에서는 선박 건조공정의 초기 단계인 강재적치처리공정에 유비쿼터스 컴퓨팅 기술을 적용한 사례를 제시하고 적용과정에서 발견된 여러 가지 문제점을 논의한다.

1990년대 초에 마크 와이저 [4, 5]는 유비쿼터스 컴퓨팅 환경을 “우리의 일상 생활에 널리 퍼져있는 물리적인 사물들에 컴퓨팅 기술을 침투시켜 컴퓨터, 사물, 인간, 정보를 통합함으로써 지식 공유 및 작업 효율 개선은 물론 상황에 따라 인간에게 지능적인 서비스를 제공하는 컴퓨팅 및 정보 환경”으로 정의하였다. 기존의 에이전트 컴퓨팅과는 다른 방향의 컴퓨팅 기술의 진화를 예견하고 주창하였다 [6]. 유비쿼터스 컴퓨팅 환경을 구축하는데 있어서 여러 가지 하드웨어적인 요소기술과 소프트웨어적인 요소 기술이 필요하다. 유비쿼터스 공간을 구축하기 위해서는 각종 센싱 디바이스 기술, RFID 기술, 객체들의 위치추정 기술 등이 필요하고, 유비쿼터스 공간으로부터 변화하는 상황정보를 실시간으로 획득하여 서버로 전달하기 위한 유.무선 통신기술이 요구된다. 또한, 수집한 상황정보를 모델링하여 애플리케이션이 요구하는 상황정보를 생성하고 제공하는 상황정보 서비스 모델링 기술, 상황정보를 저장, 관리, 분석하고, 이러한 상황정보를 기반으로 지능형 서비스를 추론하는 기술, 지능형 사용자 서비스 인터페이스를 제공하기 위한 서비스 구조화 기술 등을 포함하는 상황정보 플랫폼 기술이 요구된다.

EasyLiving [7, 8], iRoom [9], AwareHome [10], HP CoolTown [11], MIT Oxygen [12] 등과 같은 대규모 프로젝트들을 통해서 유비쿼터스 컴퓨팅의 실현을 구체화하기 위하여 요구되는 여러 가지 기반기술의 연구가 진행되고 있다. 이와는 별도로 교량 안전 시설 모니터링, 교육, 건강 및 의료 서비스 분야에 유비쿼터스 컴퓨팅 환경을 구축하기 위한 다양한 응용 연구가 시도되었다 [13, 14]. 하지만, 유비쿼터스 컴퓨팅 환경을 생산 현장에 적용한 사례는 찾아보기 드물다. 본 논문에서는 선박 건조공정의 생산현장에 유비쿼터스 컴퓨팅 환경을 구축하는 과정에서 도출된 여러 가지 기술적 이슈와 문제점을 논의하고자 한다. 본 연구는 접근성이 비교적 제한된 선박 건조과정에 유비쿼터스 컴퓨팅 환경을 구축하는 선박IT융합의 가능성을 제시한다는 점에 있어서 기존의 IT 연구와는 차별성을 가지고 있다고 할 수 있다.

본 논문에서 다루고자 하는 강재적치처리공정에서는 정보교환을 위해서 사용자 인터페이스 구현이 제한적이고 사용성이 떨어지는 개인용 정보 디바이스를 이용하고 있으며 적치처리공정이 작업자의 경험적 판단에 의하여 운영되고 있다. 이로 인하여, 강재를 잘못된 적치장에 가져다 놓는 강재의 적치오류, 중량물의 이동으로 인한 작업자의 안전 취약성, 적치장 내에 쌓여있는 강재들의 정보 관리 미흡으로 적치계획의 불확실성 등이 심각한 문제점으로 지적되고 있다. 특히, 열악한 작업환경에서 장시간 작업을 진행해야 하는 관계로, 작업자가 실수로 강재를 잘못 적치하는 경우가 빈번히 발생하여 선박건조의 생산성이 크게 영향을 받고 있다.

이러한 강재적치처리 공정이 시행되는 공간을 유비쿼터스 공간으로 구축하고 작업자에게 지능적 서비스 시스템을 구축하는 것을 목표로 하고 있다. 실용화 연구의 어려움을 극복하고 연구의 효율성을 높이기 위하여 강재적치처리의 전 과정을 시뮬레이션할 수 있는 강재적치처리 시뮬레이터를 제작하였으며, 시뮬레이터를 이용하여 유비쿼터스 컴퓨팅 시스템을 구축하고, 작동의 정확성 및 시스템의 신뢰성을 검증하였다. 시뮬레이터 공간에 센서 기술, RFID 태그 기술, 실시간 위치추적 시스템 기술 등을 적용하여 유비쿼터스 공간을 구축하고, 공간 내에서 발생하는 물체의 이동위치 및 작업처리 상태에 대한 상황정보를 실시간으로 트래킹하고 관리할 수 있도록 하였다. 또한, 척박한 환경에서 근무하는 작업자들에게 업무 환경의 쾌적성을 제공하기 위하여 음성과 3D 그래픽을 사용하는 사용자 친화적인 인터페이스를 제공하였다. 작업자의 위치정보 및 강재의 이동정보를 실시간으로 트래킹하여 작업자의 안전성을 보장하도록 하였으며, 각 적치장의 강재 스택 정보를 상세하게 관리함으로써 보다 정확한 적치 및 출고 계획을 세울 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 배경을 설명하고, 3장에서는 본 논문에서 제시하는 강재적치처리시스템에 대하여 자세히 기술하며, 4장에서는 시스템에 대한 평가, 그리고 5장에서는 결론 및 향후 연구 방향을 제시한다.

II. 배 경

2. 배경

2.1 강제적치처리장

선박 건조공정은 여덟 개의 세부공정 즉, (1) 하물선에서 강재 하역, (2) 강재를 운반하여 임시 적치장 (본 논문에서는 가적치장으로 언급함)에 적치, (3) 가적치장에 쌓인 강재들을 적치장별로 분류, (4) 분류된 강재를 전처리 공장으로 이동, (5) 강재의 이물질이나 녹을 제거하는 전처리과정, (6) 강재 절단 및 붙이기를 통한 블록 제작, (7) 블록들을 도크로 운반, (8) 블록을 조립하여 선박을 건조 등의 세부과정으로 이루어져 있다. 현재 대형선박을 건조하는 방식으로서 불럭공법을 사용하고 있는데. 이 방식은 선박을 다수의 선박 블록으로 나누어 설계 및 제작하고 이러한 선박블록을 레고처럼 조립 및 용접함으로써 선박을 완성하게 된다.

선박 건조회사는 선박블록을 제작하는데 사용되는 강재들을 국내 혹은 해외 철강회사들에게 발주를 한다. 선박 건조회사는 선박을 수주하고 선박 블록들에 대한 설계가 완료되면 각 선박블록을 제작하는데 필요한 강재들을 발주하게 되는데, 이 때 주문하는 강재가 어느 블록 제작에 사용될 것인지가 정해진다. 발주한 강재들이 화물선을 통해서 생산현장에 도착했을 때 강재 위에 찍힌 스텐실 마크의 제품번호를 보고 각 강재가 사용될 블록에 따라서 강재들을 분류하는 작업을 수행한다. 분류가 완료되면, 개별 적치장에 쌓인 강재를 블록 제작 공장으로 이동시켜서 이들을 자르고 조립하여 해당 선박블록을 제작한다.

그림 1은 강재적치처리 작업 공간 사진이다. 위 사진에 있는 강재적치처리 베이의 크기는 대략 37 x 250 (m2)의 거대한 공간이며, 베이의 전방에 8개의 가적치장 (Temporary Pile Space 혹은 t-pile Space)있으며, 바로 이어서 60~70개의 적치장 (Pile Space)이 연속적으로 배치되어 있다. 가적치장을 포함하는 모든 적치장은 2열 (좌열, 우열)로 구성되어 있으며, 각 적치장은 고유 주소가 메겨져 있다. 가적치장은 분류되지 않은 강재를 무더기로 쌓아 놓는 장소이며, 적치장은 특정 블록을 제작하는데 필요한 강재를 분류해 적치해 놓는 장소이다. 적치되어 있는 강판 위쪽으로 보면, 3대의 오바헤드 크레인이 교각처럼 생긴 양쪽 지지대에 걸려있다. 지지대의 상단에는 레일이 설치되어 있으며, 크레인 운전기사 (Crane Operator)는 오바헤드 크레인을 레일을 따라 앞뒤로 움직이고, 크레인 헤드의 위치를 좌열, 우열로 이동시킬 수 있다. 크레인 운전기사 밑에 아치형으로 생긴 것이 크레인 발이며, 크레인 발에 자성을 가하여 강재를 들어올린다. 크레인이 처리하는 강재의 무게는 최대 3,500kg에 달하며, 크기는 3m(가로) x 5m(세로) x 15mm(두께)에 달하는 것으로 조사 되었다.

2.2 강제적치처리공정의 문제점

강재적치처리 공정은 입고 (pile-in), 적치(piling), 출고 (pile-out)의 세 단계로 나누어지며. 각 단계의 작업은 입고관리자 (pile-in manager), 크레인기사 (crane operator), 출고관리자 (pile-out manager)의 의해서 수행된다. 그림 2는 기존의 강재적치처리 과정을 설명한다. 입고관리자는 적치를 원하는 강판위에 올라가서 스텐실 마크에 있는 제품번호, 재질, 마크번호, 호선번호를 읽고, 이들 번호 혹은 경우에 따라서는 강재 제품번호를 PDA에 입력하여 서버로 전송한다. 서버는 수신한 제품번호를 키로 사용하여 DB에서 해당 강재를 검색하고 상세 강재 정보 (생호선번호, 마크번호, 재질, 제품번호, 무게, 크기 등)와 함께 해당 강재를 적치할 <적치장주소> 및 해당적치장에 적치할 경우 강재의 순서인 <연번>을 PDA로 보냄으로써 응답한다. 입고관리자는 PDA로

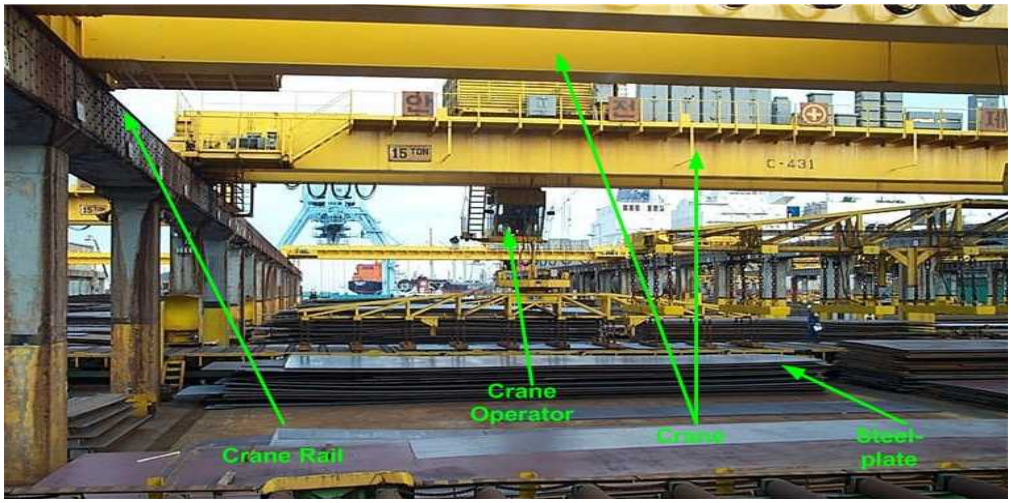


그림 1. 강재적치처리공정을 위한 강재적치처리장
Figure 1. The Space for the Steel-Plate Piling Process

수신한 <적치장 주소, 연번> 정보를 확인하고 강재 위에 적치장 주소 및 연번를 분필로 크게 기록한다. 이 과정에서 복잡한 제품번호 (영문자 및 숫자 최대 13자리)를 PDA에 입력할 때 오류가 발생할 수 있으며, 연번을 강판위에 기록하는 중에 기록 오류가 발생할 수도 있다.

크레인기사는 가적치장으로 크레인을 운전하여 이동한 후에, 7 ~ 8미터 상공에서 적치장 주소 및 연번를 읽고 기억한다. 크레인기사는 크레인 발의 자성을 이용하여 해당 크레인을 들어 올리고 기억하고 있는 적치장으로 운반하여 적치시킨다. 이 과정에서 크레인 기사가 먼 거리에 적힌 적치장번호를

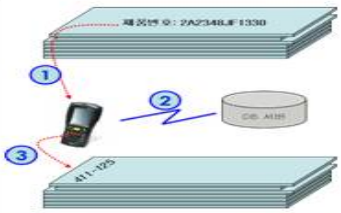
작업자		행위
입고관리자 (Pile-in Manager)		① 스텐실 위의 제품번호 (2A2348JF1330)를 읽고 PDA에 입력 ② 적치장 주소 (411) 및 연번 (125)을 DB로부터 수신 ③ 적치장 주소 및 연번을 강재위에 분필로기록
크레인기사 (Crane Operator)		① 크레인 좌석에서 적치장 주소 및 연번을 눈으로 확인 ② 강재를 리프트하여 해당 적치장으로이동 및 적치
출고관리자 (Pile-out Manager)		① 무전기를 사용하여 크레인 기사에게 다음 단계로 운반할 적치장 목록 전송 ② 강재를 컨베이어벨트로 하나씩 이동 ③ 적치장 주소 및 연번를 입력하여 출고 처리완료

그림 2. 기존 강재적치처리 프로세스
Figure 2. Current Steel-Plate Piling Process

잘못 읽는 오류를 범할 수 있으며, 실수로 잘못된 적치장에 적치시키는 오류를 범할 수 있다.

표 1. 현 강재적치처리공정의 문제점
Table 1. The problem of Current Steel-Plate Piling Process

과정	문제점
입고	- 제품번호를 PDA에 입력하는 과정에서 입력오류에 의한 적치 오류 - 강재 위에 <적치장 주소> 기록 오류에 인한 적치 오류
적치	- 크레인 기사가 강재 위에 기록한 <적치장 주소>을 잘못 읽어서 발생하는 강재 적치 오류 - 크레인 기사의 작업실수로 인하여 잘못된 적치장에 강재를 적치하는 오류
출고	- 단순 출고 관리를 수행하는 출고관리자의 인력 낭비

출고관리자가 크레인기사에게 출고할 적치장 주소 목록을 알려 주면 크레인 기사는 요청한 적치장의 목록 중에서 하나의 적치장을 선택한다. 선택한 적치장의 강재를 모두 강재 전처리장으로 이동할 때까지 하나씩 컨베이어벨트로 옮긴다. 각 강재가 컨베이어벨트에 옮겨질 때마다, 출고관리자는 적치장 주소 및 연번을 PDA에 입력함으로써 해당 적치장의 연번에 해당하는 강재가 출고 완료되었다는 것을 DB에 기록한다. 하나의 적치장 출고를 다 끝내고 나면 다음 적치장을 선택하여 위와 같은 작업을 반복한다. 이 과정에서는 단순 출고작업을 수행하는 출고관리자의 인력 낭비가 문제점으로 지적되었다.

입고, 적치, 출고 업무는 서로 독립적으로 수행된다. 기존의 적치처리과정에서 일어나는 문제점들을 표 1에 요약하였다. 이러한 문제점으로 인하여 선박건조 공정의 지연이 발생함으로써 선박 생산성이 크게 떨어지는 것으로 조사되었다.

2.3 작업 시뮬레이터

생산현장에서 개발 및 검증을 수행하는 것은 여러 가지 어려움이 따른다. 첫 째, 빡빡한 현장 작업 일정으로 인하여 작업을 중단 요청하는 것과 같이 작업을 방해할 수 없다. 두 번째, 시스템 신뢰성 테스트를 위하여 여러 가지 비정상적인 작업 시나리오를 재현하는 것이 어렵다. 세 번째, 센서, 태그 같은 여러 가지 장치 및 통신 장치들을 자유롭게 설치 및 교체하는 것이 어렵다. 끝으로, 소프트웨어 모듈과 관련 장치 혹은 기기들의 작동을 연동하기 위해 현장 기술자에게 상황 재연을 요청하기 어렵다.

이러한 현장 개발의 문제점을 극복하고 개발의 효율성을 높이기 위하여 다양한 강재적치처리 작업 시나리오를 재현해 볼 수 있는 강재적치처리 시뮬레이터를 그림 3과 같이 제작하였다. 본 시뮬레이터는 강재적치처리 베이의 축소판으로 약 2.2(가로) x 3(세로) x 2(높이) m3의 크기로 제작되었으며, 2개의 가적치장, 6개의 적치장, 컨베이어벨트, 오버헤드크레인, 자성을 가할 수 있는 크레인 발 등으로 구성되어 있다. 위에 있는 걸쳐있는 오버헤드크레인은 프레임 위에 설치된 레일을 따라 전후로 움직일 수 있으며, 빨간 램프를 가지고 있는 크레인헤드는 좌우로 움직일 수 있다. 원격 무선 제어기를 사용하여 크레인의 움직임 및 크레인 발의 자성을 ON/OFF 제어할 수 있으며, 강재를 들고 가적치장에서 적치장, 적치장에서 컨베이어벨트로 이동 시킬 수 있다. 컨베이어벨트는 강재가 위에 놓여 지면 자동으로 강재를 탐지하고 외부로 이동시킨다.

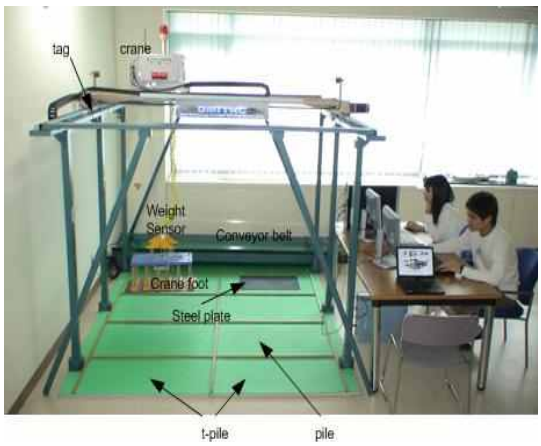


그림 3. 강재적치처리 시뮬레이터
Figure 3. Steel-Plate Piling Process Simulator

시뮬레이터 상에 RFID태그, 센서, 위치추적시스템, 카메라, 로드셀 센서 등을 설치하여 지능형 공간을 구축하였다. 시뮬레이터 내의 어떤 동적 객체에 대한 상황 정보가 변경되면 즉시 무선네트워크를 통하여 서버에 전송된다. 그림 4는 시뮬레이터의 물리적 공간의 작업상태를 3D 그래픽으로 나타는 것으로서, 시뮬레이터 공간에서 움직이는 각 실물들은 3D 공간의 대응하는 가상 객체들의 움직임과 정확히 동기화 된다.

각 작업자는 안전모에 실시간 위치추적을 가능케 하는 RFID 태그를 부착하고 있으며, 각 강재는 메탈태그 (Metal Tag)를 부착하고 있다. 크레인 기사가 강재를 들면 크레인 발에 부착된 RFID 판독기는 메탈태그로부터 강재의 정보를 읽어내고 강재 정보를 이용하여 해당 강재가 적치될 강재의 적치장 번호와 강재 일련번호를 DB로부터 자동으로 가져와서 화면에 표시한다. 크레인 기사는 화면을 보고 해당 적치장으로 이동하여 강재를 들고 해당 적치장으로 이동하게 된다. 이동하는 과정에서 크레인의 위치를 포함하는 모든 상황정보가 트래킹되고, 이를 기반으로 다음 수행할 작업을 3D 그래픽 혹은 음성인터페이스를 이용하여 가이드 한다.

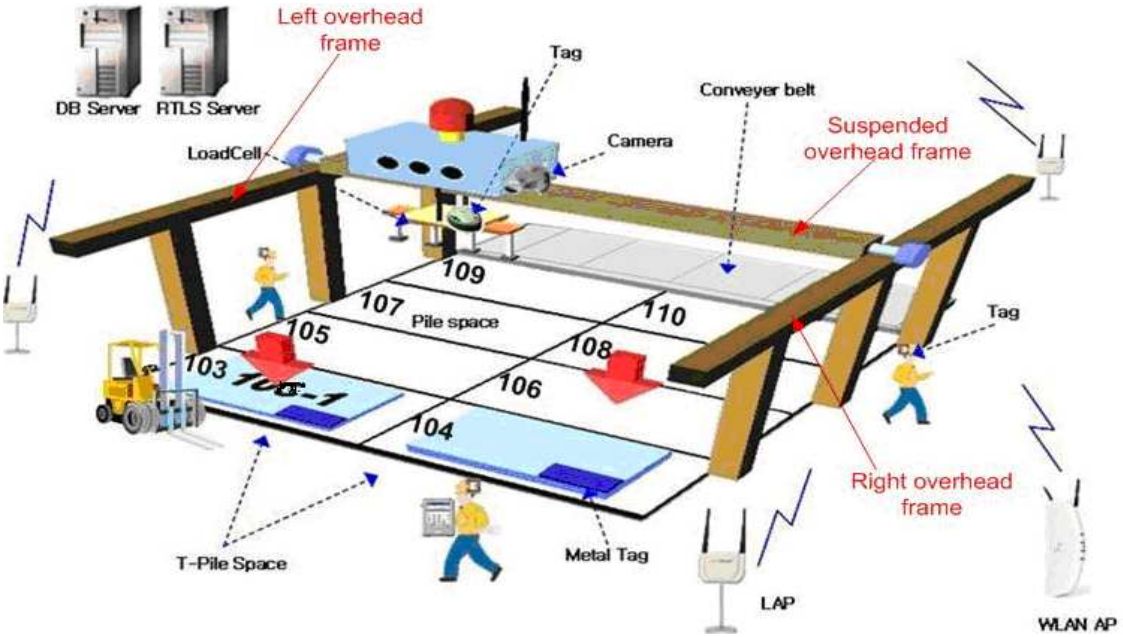


그림 4. 3D 사용자 인터페이스 화면
Figure 4. 3D User Interface Screen

표 2. u-Space 구축에 요구되는 장치 사양 및 용도
Table 2. Devices required for building u-Space and their usage

주요 요소		사양	용도
Metal RFID		주파수: 900MHz 읽기 거리: Max. 4 m	- 강재 정보 획득
로드셀 + 센서 모트		인터페이스: RS-232, OS: Tiny OS 모트 주파수: 2.4GHz, 최대 전송거리: 50 m	- 강재 무게 획득 - 강재 리프트 오류 체크
RTLS	LAP	주파수: 2.4GHz DSSS 802.11b 외부 전송거리: >= 200m, 최대 오차: 3m	- RTLS 태그 정보 수신 및 서버로 송신
	익사이터 (Exciter)	전송거리 (변경가능): 0.5~2m	- 크레인 위치정보, 즉 주소 파악
	T2 태그	외부 전송거리: >= 200m 읽기 거리 (125KHz): 0.5~7 m 신호 전송주기: 6.5 ms ~ 3.5h	- <태그ID, 익사이터 ID>를 LAP으로 전송
RTLS 서버		시스템: Desktop PC 소프트웨어: RTLS 엔진 software	- RTLS 정보 관리
DB 서버		시스템: Desktop PC 데이터베이스: MS SQL	- 강재 주문정보, 적치공간의 강재 스택관리 - u-SPPS 상황 정보 관리
ULS	1 수신기 (40 KHz) 4 송신기, 1 싱크 발생기	주파수: 40 KHz, RF 싱크: 315MHz 커버 범위: 7 m, 최대 오차: 5cm	- 지상 작업자 및 적치할 강재의 위치추적

그림 4에 있는 3D 사용자 인터페이스로부터 현재 상황을 해독하면 다음과 같다. 크레인 (헤드)의 현재 위치는 107번 적치장 주소에 위치하고 있으며, 현재 크레인 기사에게 크레인을 빨간색 화살표가 지시하는 가적치장 103번지로 움직이도록 요청하고 있다. 103번 가적치장의 강재 윗면에 적힌 106-1은 해당 강재를 빨간색 화살표가 가리키는 적치장 106번지로 이동하라는 것을 의미하며, 연번을 나타내는 1은 그 강재가 해당 적치장에 적치되는 첫 번째가 될 것이라는 것을 의미한다. 현 상황이 실시간으로 트래킹되기 때문에 강재가 잘못된 적치장에 적치하려는 경우에는 경고 메시지를 출력한다.

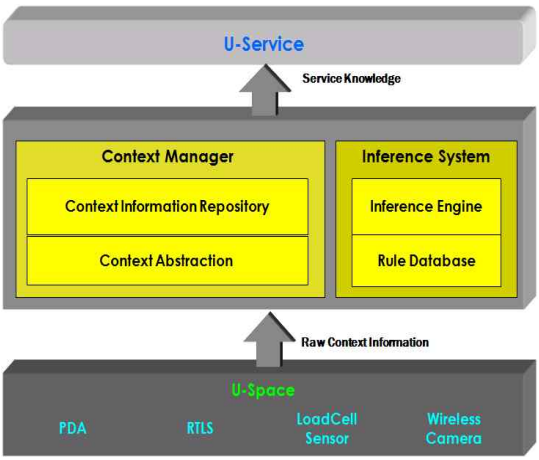


그림 5. u-SPPS 구조
Figure 5. u-SPPS Architecture

III. u-SPPS

이 장에서는 u-SPPS (Ubiquitous Steal-Plate Piling Process System)의 아키텍처 및 구성요소들에 대하여 설명하고, 핵심이 되는 상황 인지 추론 모델을 자세하게 기술한다.

3.1 u-SPPS 아키텍처

u-SPPS는 그림 5의 u-Space, 상황정보 관리 (Context Manager), 추론 시스템 (Inference System), u-Service 계층으로 이루어져 있다. u-Space는 RFID, 센서 등이 설치되어 여러 가지 상황정보를 서버로 전송할 수 있도록 설계된 공간이다. u-Service는 추론엔진에 의해 결정된 서비스를 받아서 다양한 사용자 인터페이스를 통하여 서비스를 실행한다. 컨텍스트 관리자 (Context Manager)는 u-Space로부터 받은 가공되지 않은 상황 정보를 추상화하여 상황정보 저장소에 저장하고 관리한다. 추론 시스템은 추론 엔진과 추론 룰 데이터베이스로 구성된다. 추론 엔진은 상황정보 저장소에 주기적으로 갱신되는 정보에 따라 룰을 실행시킴으로써 상황에 적합한 서비스 혹은 행위를 추론한다. 추론 결과는 u-Service에 전달되고 해당 함수를 실행함으로써 사용자에게 서비스를 제공한다.

3.2 u-Space 구축

u-Space를 구축하는데 사용한 여러 가지 기술들을 표 2에 정리하였다. 강재 정보를 획득하기 위하여 메탈 RFID기술을 사용하였다. 일반 RFID의 경우에는 강재가 전파를 흡수하는 성질이나 혹은 충격에 약하기 때문에 적합하지 않다. RFID 판독기가 부착된 PDA는 각 강재 위에 부착된 메탈 RFID 태그로부터 강재에 대한 정보를 읽고 DB 서버로 보낸다. DB 서버는 수신된 정보를 사용하여 해당 강재의 적치 정보 <적치장 번호, 연번> 을 검색하여 크레인 기사가 사용하는 클라이언트로 전송한다.

크레인이 강재를 들었는지 (UP) 혹은 내려놓았는지 (DOWN)를 감지하기 위하여 로드셀 무게센서를 사용하였다. 무게 센서는 현재 들어 올린 강재의 무게를 측정하여 DB로 전송하고 시스템은 측정된 무게와 DB에 저장되어 있는 해당 강재의 무게를 비교하여 크레인이 정확한 강재를 들었는지 혹은 복수 개의 강재를 들었는지를 검사할 수 있다. 실제 시스템에 자성을 이용하여 강재를 들기 때문에 두 개의 강재를 들어 올리는 오류를 범할 수 있다.

공간 내에서 크레인의 움직임을 파악하고 현재 위치를 알아내기 위하여 RTLS (실시간 위치추적 시스템)을 사용하였다. RTLS 시스템은 RTLS 서버, 3개의 LAP (Location Access Point), 2개의 익사이터 (Exciter), 6개의 RTLS 태그를 사용하였다. 익사이터는 자신의 ID를 주기적으로 전송하고, 수신 거리에 있는 RTLS 태그는 자신의 ID와 수신한 익사이터의 ID를 합쳐서 RTLS 서버에게 전송한다. 따라서, 서버는 익사이터가 어떤 RTLS 태그 가까이에 있는지를 인지할 수 있다. 익사이터는 크레인 헤드를 지탱하는 오브헤드프레임의 양쪽 끝에 각각 하나씩 부착하였으며, 시뮬레이터 상에서 최대 전송거리 50cm로 설정하여 한 번에 하나의 RTLS 태그만 익사이터 ID를 수신할 수 있도록 하였다.

크레인헤드에 하나의 RTLS 태그를 부착하였으며, 영역을 포함하는 5개 열의 각 영역을 구분하기 위하여 대응하는 오버헤드 프레임의 각 부분에 RTLS 태그를 하나씩 부착하였다 (그림 4의 각 열은 두 개의 적치장 혹은 하나의 컨베이어 벨트에 해당한다). RTLS 태그의 수를 줄이고 협소한 시뮬레이

터 공간에서 동일한 익사이터의 ID를 두 개의 서로 다른 RTLS 태그가 수신하지 못하도록 하기 위하여 5개의 RTLS 태그를 지그재그 (ZigZag) 형태 즉, 103, 106, 107, 110, 컨베이어벨트에 해당하는 좌측 혹은 우측 오버헤드 프레임에 부착하였다. 이러한 RTLS 태그 및 익사이터의 배치를 통하여, 크레인헤드가 좌측으로 이동하는 경우에 프레임의 좌측에 부착된 익사이터가 발신하는 익사이트 ID를 크레인헤드에 부착된 RTLS 태그가 읽고 <익사이터 ID, RTLS 태그 ID> 조합을 RTLS 서버로 보내게 되고, 서버는 현재 크레인헤드가 좌측에 있다는 것을 알 수 있다. 이 때 좌측 혹은 우측 프레임에 부착된 RTLS태그도 익사이터 ID를 읽게 되고 걸쳐있는 프레임 (suspended overhead frame)이 적치장 베이의 어느 열에 위치하고 있는지를 알 수 있다. 열과 크레인헤드의 위치를 조합하여 크레인이 어느 적치장에 있는지를 알 수 있다.

또한, 시뮬레이터 공간에서 각 이동 물체의 정밀한 위치를 추적하기 위하여 ULS (Ultrasonic Location System)을 사용하였다. ULS 수신기는 작업자의 안전모에 부착되어 작업자의 정확한 위치를 실시간으로 파악할 수 있으며, 작업자와 크레인이 안전거리보다 가까워지는 경우에는 시스템은 경보장치를 가동하여 경보 메시지를 출력한다 (단, 넓은 현장에서는 3미터 이내의 정확도로 물체를 추적할 수 있는 RTLS 시스템만으로 객체의 위치를 추적하는데 문제가 없다).

마지막으로, 크레인헤드의 하측에 무선 카메라를 설치하여 현재 리프트한 강재의 영상을 보여주고 강재 영상과 오버랩하여 해당 강재의 상세 정보 (적치장 주소, 연번, 마크번호, 호선번호, 무게, 크기 등)를 표시함으로써 화면을 통하여 작업하고 있는 강재를 현실감 있게 확인할 수 있다.

3.3 u-Service

u-Service는 선택된 룰의 결과를 실행하여 서비스를 제공할 수 있는 서비스 라이버러리를 포함하고 있으며, 이들 서비스 함수는 작업 공정 순서에 대한 3D화면 혹은 음성 가이드, 에러 발생 시에 경고 메시지 출력, 위험신호 발신 함수 등과 같은 것을 사용자에게 알려 주기 위하여 다양한 사용자 인터페이스를 제공한다.

3.4 상황정보 관리자(Context Manager)

u-Space로부터 획득한 상황정보는 해석, 변환, 조합 등의 추상화 과정을 거쳐서 상황정보 저장소에 저장된다. 상황정보 저장소에 저장된 상황정보는 u-Space내의 객체들이 변화와 실시간으로 동기화된다. 즉, 상황정보는 물리공간이 정보의 형태로 표현되는 디지털 전자공간이라고 할 수 있다.

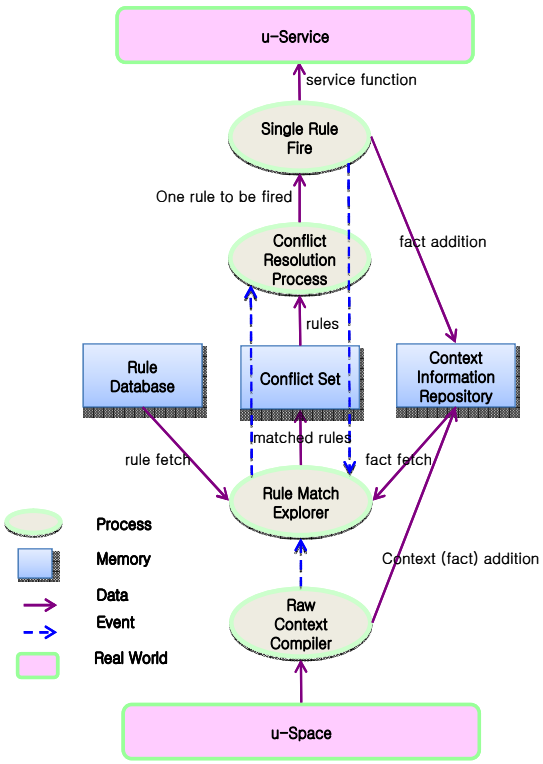


그림 6. 상황인지 및 추론 모델
Figure 6. Context-Aware and Reasoning Model

상황정보 저장소에 실시간으로 갱신되는 상황정보들은 유비쿼터스 컴퓨팅 물리공간에서 작업자들에게 현재 상황에 적합한 서비스를 도출하는 데 사용된다. 상황정보는 추론엔진에 속한 룰 들을 실행하기 위한 입력 조건으로 사용되는 팩트 (fact)가 된다.

3.5 추론시스템(Inference System)

본 논문에서 사용하는 추론 시스템은 추론에 필요한 정의된 룰을 저장해 놓은 룰 DB와 추론 엔진으로 구성되어 있으며, 추론 엔진은 상황정보 저장소에 있는 상황정보 (팩트)들을 사용하여 조건이 만족되는 룰을 선택 및 실행한다.

추론 엔진은 그림 6과 같은 상황 인지 및 추론 모델을 사용하였으며, 이 모델은 상황정보를 관리하고 상황정보 저장소에 있는 팩트에 기반한 서비스를 추론해 내는 과정을 도식화하고 있다. 이 모델은 타원으로 나타낸 실행 프로세스, 기 정의된 룰을 저장하는 룰 DB, 현재 상황정보를 가지고 여러 가지 룰을 임시로 저장하기 위한 Conflict Set 메모리, 상황정보 저장소 등으로 구성되어 있다.

3.5.1 Logic

• 상황정보 수집 (Raw Context Compiler)

상황정보수집 프로세스는 유비쿼터스 공간 속에 내재되어 있는 각종 디바이스들로부터 상황정보를 수집하고, 그 정보를 상황 정보 저장소(Context Information Repository)에 저장시키는 역할을 담당한다. 그리고, 이벤트를 사용하여 상황정보의 변화가 발생하였다는 것을 알려 줌으로서 "Rule Match Explorer" 프로세스를 작동시킨다.

• 룰 매치 탐색 (Rule Match Explorer)

상황정보수집 프로세스로부터 이벤트를 받은 룰 매치 탐색기는 룰 DB에 있는 각 룰의 실행 조건들에 상황정보 저장소에 있는 팩트들을 대입하여 실행조건을 만족하는 룰을 선택하여 Conflict Set에 저장한다. 이러한 과정을 통하여 현재의 상황에 부합하는 모든 룰들을 찾아서 저장한 후에 경쟁해결 (Conflict Resolution) 프로세스에 이벤트를 전송하여 해당 프로세스를 작동시킨다.

• 경쟁해결 (Conflict Resolution)

경쟁해결 프로세스는 Conflict Set에 저장된 룰들 중에서 가정 적합한 룰을 하나 선택하여 "Single Rule Fire" 프로세스가 실행하도록 한다.

• 단위 룰 실행 (Single Rule Fire)

경쟁해결 프로세스가 선택한 룰을 실행하고 이를 통하여 새로운 정보를 저장하거나 기 저정된 정보를 삭제한다. 그리고, 룰 매치 탐색기에게 이벤트를 보내서 룰 매치 탐색기로부터 과정을 반복하도록 한다. 추론이 끝나면, 그 결과에 해당하는 함수정보는 u-Service에 보내고 u-Service는 해당 함수를 실행함으로써 사용자에게 서비스를 제공 한다.

```
{defrule PileinStart
(PileinMode ?)
(WaitToPileinStart ?)
(CorrectPileSpace ?)
(Lifted ?)
=>
(Remove WaitToPileinStart ?)
(Add PileinStart ?)
(Run ChangeDest ?)
}
```

그림 7. 룰(rule) 표현방식
Figure 7. Rule Representation

3.5.2 공유메모리 (Shared Memory)

• 상황정보 저장소 (Context Information Repository)

상황정보 저장소는 상황정보수집 프로세스가 유비쿼터스 공간으로부터 취득한 상황정보를 저장하거나 룰 실행 프로세스의 실행 결과로 얻어지는 새로운 상황정보 혹은 팩트를 저장한다. 저장된 정보는 룰 실행의 결과로 삭제될 수도 있다.

• 룰 데이터베이스 (Rule Database)

룰 데이터베이스는 현재 상황에서 수행할 작업이나 위험상황 판단 등을 하기 위해서 기 정의한 룰 들을 저장한다. 각 룰은 현재의 팩트 (상황)들을 입력 값으로 하여 조건이 맞으면 해당 룰에 연계된 함수를 수행하는 형태로 구성되어 있다. 룰은 그림 7과 같이 표현된다. 각 룰은 괄호 "{ }"에 의해 분리되고, defrule을 따라 나오는 룰의 이름에 의해서 정의된다. 룰의 이름 뒤에 한 개 이상의 전제조건 (antecedent)이 따라 나오고, 전제조건과 실행함수를 분리하는 "=>" 심볼, 그리고 전제조건이 참인 경우에 수행하는 하나 이상의 실행함수 (consequent)가 정의된다.

3.6 u-SPPS에서 상황 인지 및 추론 과정

u-SPPS에서의 상황 인지 및 추론 과정을 입고모드 (PileinMode)와 출고모드 (PileoutMode)로 나누어 설명한다.

3.6.1 Rule DB

강재적치처리 시스템을 구현하기 위하여 사용된 룰은 Ri, i는 숫자로 표기하며 입고모드 및 출고모드에서 사용되는 룰들은 표 3에 정의되어 있다.

3.6.2 입고 모드

입고모드에서의 작업공정은 가적치장에 쌓여있는 강재 파일에서 각 강재를 해당 적치장으로 옮겨 놓기까지 일어나는 작업 과정이다. 따라서, 입고모드는 적치모드를 포함하여 기술한다. 작업공정은 상태천이도로 표현할 수 있으며, 네 개의 정상처리 상태를 가진다 (WaitToPileinStart, WaitToPileinEnd,, PileinStart, PileinEnd). 상태 천이는 실시간으로 트래킹 되는 유비쿼터스 컴퓨팅 공간의 상황정보에 기반하여 이루어지며 상태천이도의 천이를 표현하는 화살표에 대응하는 룰이 실행된다. 그림 8은 입고모드 상태천이도이며, 각 상태에 대한 설명은 다음과 같다.

- 상태 InitialState는 크레인이 작업을 시작하기 위하여 Power를 켠 상태이다.
- 상태 WaitPileinStart는 크레인이 현재 적치를 원하는 강재가 있는 가적치장으로 이동하여 강재를 들어 올릴 준비가 된 상태이다.

표 3. 입고 모드 및 출고 모드 작업 처리 룰
Table 3. Rules used in Pile-in and Pile-out mode

입고모드에서 사용되는 룰		출고모드에서 사용되는 룰	
<R1> {defrule PileinStart (PileinMode ?) (WaitToPileinStart ?) (CorrectPile ?) (UP ?) => (Remove WaitToPileinStart ?) (Add PileinStart ?) (Run ChangeDest ?) }	<R2> {defrule WaitToPileinEnd (PileinMode ?) (PileinStart ?) (CorrectPile ?) (UP ?) => (Remove PileinStart ?) (Add WaitToPileinEnd ?) }	<R11> {defrule PileoutStart (PileoutMode ?) (WaitToPileoutStart ?) (CorrectPile ?) (UP ?) => (Remove WaitToPileoutStart ?) (Add PileoutStart ?) (Run ChangeDest ?) }	<R12> {defrule WaitToPileoutEnd (PileoutMode ?) (PileoutStart ?) (CorrectPile ?) (UP ?) => (Remove PileoutStart ?) (Add WaitToPileoutEnd ?) }
<R3> {defrule PileinEnd (PileinMode ?) (WaitToPileinEnd ?) (CorrectPile ?) (DOWN ?) => (Remove WaitToPileinEnd ?) (Add PileinEnd ?) (Run PileinEnd ?) (Run ChangeDest ?) }	<R4> {defrule WaitToPileinStart (PileinMode ?) (PileinEnd ?) (CorrectPile ?) (DOWN ?) => (Remove PileinEnd ?) (Add WaitToPileinStart ?) }	<R13> {defrule PileoutEnd (PileoutMode ?) (WaitToPileoutEnd ?) (CorrectPile ?) (DOWN ?) => (Remove WaitToPileoutEnd ?) (Add PileoutEnd ?) (Run PileoutEnd ?) (Run ChangeDest ?) }	<R14> {defrule WaitToPileoutStart (PileoutMode ?) (PileoutEnd ?) (CorrectPile ?) (DOWN ?) => (Remove PileoutEnd ?) (Add WaitToPileoutStart ?) }
<R5> {defrule PileinEndError (PileinMode ?) (PileinStart ?) (IncorrectPile ?) (DOWN ?) => (Remove PileinStart ?) (Add PileinEndError ?) (Run InsertCurDest ?) (Run WarningWrongPilePutdown ?) }	<R6> {defrule PileinStart (PileinMode ?) (PileinEndError ?) (CorrectPile ?) (UP ?) => (Remove PileinEndError ?) (Add PileinStart ?) (Run ChangeDest ?) (Run HideWarning ?) }	<R15> {defrule PileoutStartError (PileoutMode ?) (PileoutEnd ?) (IncorrectPile ?) (UP ?) => (Remove PileoutEnd ?) (Add PileoutStartError ?) (Run InsertCurDest ?) (Run WarningWrongPileLifted ?) }	<R16> {defrule PileoutEnd (PileoutMode ?) (PileoutStartError ?) (CorrectPile ?) (DOWN ?) => (Remove PileoutStartError ?) (Add PileoutEnd ?) (Run ChangeDest ?) (Run HideWarning ?) }
<R7> {defrule PileinEndError (PileinMode ?) (WaitToPileinEnd ?) (IncorrectPile ?) (DOWN ?) => (Remove WaitToFileinEnd ?) (Add PileinEndError ?) (Run InsertCurDest ?) (Run WarningWrongPilePutdown ?) }	<R8> {defrule PileinStartError (PileinMode ?) (PileinEnd ?) (IncorrectPile ?) (UP ?) => (Remove PileinEnd ?) (Add PileinStartError ?) (Run InsertCurDest ?) (Run WarningWrongPileLifted ?) }	<R17> {defrule PileoutStartError (PileoutMode ?) (WaitToPileoutStart ?) (IncorrectPile ?) (UP ?) => (Remove WaitToPileoutStart ?) (Add PileoutStartError ?) (Run InsertCurDest ?) (Run WarningWrongPileLifted ?) }	
<R9> {defrule PileinEnd (PileinMode ?) (PileinStartError ?) (CorrectPile ?) (DOWN ?) => (Remove PileinStartError ?) (Add PileinEnd ?) (Run ChangeDest ?) (Run HideWarning ?) }	<R10> {defrule PileinStartError (PileinMode ?) (WaitToPileinStart ?) (IncorrectPile ?) (UP ?) => (Remove WaitToPileinStart ?) (Add PileinStartError ?) (Run InsertCurDest ?) (Run WarningWrongPileLifted ?) }	<R100> {defrule WaitToPileinStart (PileinMode ?) (InitialState ?) (CorrectPile ?) (DOWN ?) => (Remove InitialState ?) (Add WaitToPileinStart ?) }	<R200> {defrule WaitToPileoutStart (PileoutMode ?) (InitialState ?) (CorrectPile ?) (DOWN ?) => (Remove InitialState ?) (Add WaitToPileoutStart ?) }

- 상태 PileinStart는 크레인이 가적치장에서 강재를 들고 강재를 적치하기 위하여 적치장으로 이동 중인 상태이다.
- 상태 WaitToPileinEnd는 크레인이 목적지 적치장에 도착한 상태이다.
- 상태 PileinEnd는 크레인이 들고 있는 강재를 목적지 적치장에 적치하고, 다음 작업을 위해서 적치를 요청한 가적치장으로 이동 중인 상태이다.

- 상태 PileinEndError는 강재적치 오류가 발생한 상태이다.
- 상태 PileinStartError는 잘못된 가적치장에서 강재를 들어올린 상태이다.

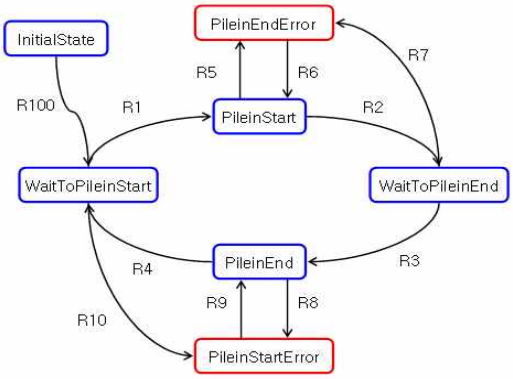


그림 8. 입고모드 상태천이도
Figure 8. State transition diagram in the Pile-in mode

유비쿼터스 컴퓨팅 공간 내에서의 변화로 인하여 상황정보의 변화가 발생하면 상황정보 저장소의 내용이 변경되고 이때 "룰 메치 탐색" 프로세스가 실행된다. 특정 룰이 선택되어 실행되는 경우에 상태천이가 발생한다. 다음은 정의된 여섯 개의 상태에 대한 상태천이를 설명한다.

- WaitToPileinStart 상태는 InitialState상태에서 크레인이 입고 계획된 가적치장으로 이동한 상태에서 R100이 실행되는 경우에 혹은 크레인이 강재를 적치완료 한 상태인 PileinEnd에서 다음 강재를 적치하기 위하여 목적지 가적치장으로 이동 완료한 후에 R4가 실행된 상태이다.
- WaitToPileinStart에서 강재를 들어 올린 경우에 R1이 실행되면 PileinStart 상태로 천이된다.
- PileinStart상태에서 목적지 적치장에 도착하고 난 후에 R2가 실행되면 WaitToPileinEnd 상태가 되고, 이 상태에서 목적지 적치장에 적치하지 않고 실수로 지나가서 다른 적치장에 적치하는 경우에 R7이 실행되면 PileinEndError 상태로 천이된다. 또한, PileinStart상태에서 크레인이 잘못된 적치장으로 바로 이동하여 적치하는 경우에 R5가 실행되면 PileinEndError 상태가 된다. 적치 오류가 발생하면 경고 메시지가 출력되고, 오류를 교정하기 위하여 잘못 적치한 강재를 다시 들고 난 후에 R6가 실행되면 PileinStart 상태로 복귀한다.
- PileinEnd 상태는 WaitToPileinEnd 상태에서 목적지 적치장에 적치를 완료하고 R3가 실행되면 천이되는 상태이다.
- PileinEnd 상태에서 목적지 가적치장에 도착하고 난 후에 R4가 실행되면 WaitToPileinStart 상태가 되고, 이 상태에서 목적지 가적치장에서 강재를 들지 않고 실수로 지나가서 다른 가적치장에서 강재를 드는 경우에 R10이 실행되면 PileinStartError 상태로 천이된다. 또한, PileinEnd 상태에서 잘못된 가적치장으로 바로 이동하여 강재를 드는 경우에 R8이 실행되면 PileinStartError 상태가 된다. 강재 들기 오류가 발생하면 경고 메시지가 출력되고, 오류를 교정하기 위하여 강재를 들어 올린 원래 위치에 다시 내려놓고 난 후에 R9가 실행되면 PileinEnd 상태로 복귀한다.

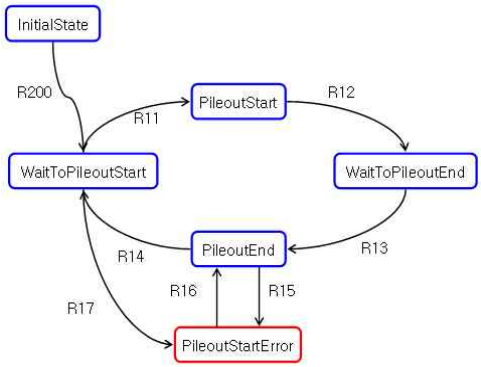


그림 9. 출고모드 상태천이도
Figure 9. State transition diagram in the pile-out mode

3.6. 3 출고 모드

출고모드에서의 작업공정은 강재의 분류 작업이 끝나고, 해당 적치장에 적치된 강재를 컨베이어벨트에 옮겨 놓는 작업을 포함한다. 크레인 기사는 출고 작업지시가 내려진 적치장에 쌓여있는 강재들을 모두 전처리 작업장으로 이동할 때까지 반복해서 강재를 컨베이어벨트로 옮긴다. 작업공정은 상태천이도로 표현할 수 있으며, WaitToPileoutStart, PileoutStart, WaitToPileoutEnd, PileoutEnd 등, 정상 동작상태를 가지며, 상태 천이도의 상태 천이는 입고 모드에서처럼 상태천이도의 천이를 나타내는 화살표에 대응하는 룰을 선택 및 실행 (fire)함으로써 이루어진다. 그림 9은 출고모드 상태전이도이며 각 상태에 대한 설명은 다음과 같다.

- 상태 InitialState는 크레인이 작업을 시작하기 위하여 Power를 켠 상태이다.
- 상태 WaitPileoutStart는 크레인이 현재 출고하도록 지정된 적치장으로 이동하여 강재를 들어 올릴 준비가 된 상태이다.

- 상태 PileoutStart는 크레인이 적치장에서 강재를 들고 출고하기 위하여 컨베이어벨트로 이동 중인 상태이다.
- 상태 WaitToPileoutEnd는 크레인이 강재를 내려놓을 컨베이어벨트에 도착한 상태이다.
- 상태 PileoutEnd는 크레인이 들고 있는 강재를 컨베이어벨트 상에 내려놓고, 다음 작업을 위해서 출고진행 중인 적치장으로 이동 중인 상태이다.
- 상태 PileoutStartError는 잘못된 적치장에서 강재를 들어 올린 상태이다.

다음은 출고모드에서 정의된 여섯 개의 상태에 대한 상태 천이를 설명한다.

- WaitToPileoutStart 상태는 크레인이 InitialState 상태에서 출고 계획된 적치장에 이동 완료한 상태에서 R200이 실행되는 경우에 혹은 강재를 컨베이어벨트에 내려놓은 후 즉, PileinEnd 상태에서 다음 강재를 출고하기 위하여 지정된 적치장으로 이동한 상태에서 R14가 실행되면 천이되는 상태이다.
- WaitToPileoutStart에서 강재를 들어 올린 경우에 R11이 실행되면 PileoutStart 상태로 천이된다.
- PileoutStart상태에서 컨베이어벨트로 이동한 후에 R12가 실행되면 WaitToPileoutEnd 상태가 되고, 이 상태에서 강재를 내려놓으면 PileoutEnd 상태로 천이된다.
- PileoutEnd 상태에서 출고 계획된 적치장이 아닌 다른 적치장으로 가서 강재를 드는 경우에 R15가 실행되면 PileoutStartError 상태가 된다. 혹은, 계획된 적치장에 도착하면 즉시 WaitToPileoutStart 상태가 된다. 하지만, 계획된 적치장을 지나쳐서 다른 적치장에서 강재를 들어 올리는 경우에 PileoutStartError 상태로 천이되고 R17이 실행되면 경고 메시지가 출력된다.
- 적치오류가 발생한 경우에 잘못 들어 올린 강재를 원래 위치에 다시 내려놓을 때 R16이 실행되면 PileoutEnd 상태로 천이되고, 원래 출고 계획된 강재 적치장으로 이동하였을 때 R14가 실행되면 WaitToPileoutStart 상태가 되어 정상적인 처리 상태로 복귀하게 된다.

3.7 U-SPPS 강재적치처리공정

크레인 기사는 크레인을 운전하여 가적치장으로 이동한 후에 강재를 들어올린다. 크레인의 발에 부착된 RFID 리더기는 강재의 RFID 태그를 읽어서 제품정보를 DB에 전송하고 서버는 해당강재를 적치할 <적치장주소, 연번>을 크레인 기사 운전실에 있는 시스템으로 보낸다. 크레인 기사는 해당 제품 번호에 대한 목적지 즉, <적치장 주소, 연번>을 읽거나 혹은 음성 메시지를 듣고 목적지 적치장으로 이동하여 적치시킨다.

강재 출고의 경우에, 강재분류 작업이 완료되면, 크레인 기사는 출고할 적치장목록을 수신한다. 출고할 적치장번호를 선택하고 해당적치장에 쌓여있는 강재를 하나씩 컨베이어벨트로 옮긴다. 강재를 옮겨서 컨베이어벨트에 내려놓으면 작업자의 개입이 없이 출고 처리가 자동으로 완료된다.

위 과정을 통해서 상황정보를 실시간으로 트래킹 함으로써 강재정보 읽기 오류, 복수의 강재를 들어 올리는 오류, 잘못된 적치장에 적치하는 오류, 입고 및 출고 관리자의 인력 낭비 등의 문제들은 해결할 수 있으며, 다양한 사용자 인터페이스를 통하여 크레인 기사의 정신적 오버헤드를 줄여준다..

표 4 기존 시스템과 u-SPPS의 비교
Table 4 Comparison of current system and u-SPPS

주요 내용	기존시스템	u-SPPS
제품번호 입력	문자판 입력	RFID/문자인식 자동입력
강재적치오류 검사	사용 안함	자동검사
강재 무게 인식	못함	무게 센서 사용
적치장 주소 기록	분필을 사용한 쓰기	필요하지 않음
적치 오류 체크	못함	가능
출고 관리	출고관리자가 처리	자동
적치장의 스택관리	수행하지 않음	수행 함
작업인력	입고, 적치, 출고 관리자 포함 3명	적치관리자 1명 (크레인기사)
작업 순서 가이드	없음	음성 및 3차원 그래픽

u-SPPS는 지능형 작업 공정 서비스 제공하는 것 외에 작업공정 오류 방지 및 작업자 안전성을 높인다.

IV. u-SPPS 평가

4.1 기존의 시스템과 비교

그림 2에서 기존의 강재적치처리 프로세스를 기술하였으며 표 2에서 문제점을 정리하였다. 기존의 강재적치처리시스템은 시스템을 전산화하여 운영의 효율성 추구하고자 하였으며, 데이터 입력 과 전송에만 적용되었다. 하지만, u-SPPS는 여러 가지 IT를 적용하여 운영의 효율성은 물론 작업자에게 지능형 서비스를 제공하는데 중점을 두었다는 점에 있어서 근본적인차이가 있다. 두 시스템의 비교 내용을 표 4에 정리하였다.

표 5. 주요기술 및 기술적 요구사항
Table 5. Key technologies and technical requirements

기술	문제점
메탈태그 전송거리	900Mhz 대역의 RFID 태그 인식거리가 최대 7m정도로 알려져 있으나 강재에 부착하는 경우에 전자파 흡수로 인하여 전송거리가 줄어들고 신뢰성이 낮아짐. 테그 고정형의 경우에 인식거리는 안테나 방식에 따라 1 ~ 3m정도 다양함
메탈태그 내구성	메탈태그의 두께는 현재 8mm정도 가능하나 훨씬 더 슬림화되고 낙하 충격에 견딜 수 있어야함
메탈태그 부착방식	메탈태그의 강재부착 방식으로 자석방식이 사용되고 있으나 스팟용접 방식, 양면테이프 방식, 스크류 방식 등의 활용이 필요함. 하지만, 자석이나 테이프부착 방식은 충격, 고온, 저온, 방수 등을 고려하기 어려움
태그정보 복수읽기	적재된 강재 더미로부터 복수 개의 태그를 읽는 오류가 발생함. 인식거리 조정 및 응용 시스템과의 연동 개발을 통하여 문제를 회피할 수 있음
태그크기	태그 사이즈가 작으면 인식거리가 줄어들지만, 뚜께 및 길이의 소형화가 요구됨

4.2 기술적인 한계 및 해결방안

강재적치처리시스템을 구축하는데 있어서 RFID 기술의 현황 및 한계점을 표 5에 기술하였다.

RFID 기술을 강재적치처리시스템에 적용하기 위해서는 표에서 열거한 기초 기술의 실용화 연구가 선행되어야 한다. 하지만, 문자인식 시스템을 사용하면 문제를 쉽게 해결할 수 있을 것이다. 강재 상판에 새겨진 스텐실마크를 영상 카메라를 사용하여 읽어 들이고 문자 인식 기술을 통하여 제품번호, 호선번호 등을 읽을 수 있다. 이 경우에 강재 생산기업들이 좀 더 뚜렷한 스텐실마크를 사용하도록 유도할 필요가 있다.

4.3 논의

(조선IT융합) 조선 분야는 전통적인 노동집약적 산업으로 건조 공정의 대부분이 작업자의 경험적 능력에 의존하고 있는 실정이다. 생산 현장의 규모가 방대하고, 현장 접근성이 낮으며, 대부분의 선박건조 작업 과정이 정형화되어 있지 않아서 IT응용의 어려움이 있다. 이러한 특징을 갖는 조선 건조공정에 유비쿼터스 컴퓨팅 공간을 구축하는 연구를 수행하였다.

(유비쿼터스 컴퓨팅 응용 모델 제시) 선박 건조 작업장에 유비쿼터스 컴퓨팅 공간을 구축하여 지능형 서비스를 제공함으로써 작업 공정의 생산성 및 안전성 제고와 같은 유비쿼터스 컴퓨팅 기술의 새로운 응용 모델을 제시하였다.

(오류방지) A중공업 경우에 하루 평균 7건의 강재 적치오류가 발생하는 것으로 보고되고 있는데, 적치오류가 한 건 발생할 때마다 오류 수정을 위한 많은 시간과 자원이 낭비된다. u-SPPS는 오류 발생을 실시간으로 감지하고 경고한다.

(안전성) 작업 공간 내의 객체들의 위치 관리를 통하여 작업자의 위험성을 실시간으로 점검하고 경고해 준다.

(쾌적성) 척박한 작업환경에서 작업자 친화적인 멀티미디어 서비스 인터페이스를 제공함으로써 생산성 제고와 더불어 작업환경의 쾌적성을 제공한다.

(적치 계획의 효율성) 각 적치장에 대한 적치 스택을 관리함으로써 적치 및 출고 계획의 효율성과 투명성을 제공한다.

(확장성) u-SPPS를 구현하는데 있어서 룰 기반 추론 모델을 사용하였다. 교육, 의료 등, 타 분야에 적용하기 위해서는 상황정보 모델링과 룰 셋을 개발하면 가능하다.

V. 결론 및 연구 방향

IT분야에 신기술 개발과 기술의 실용화는 별개의 문제이다. 개발된 기술이 특정 환경에 적용될 때 여러 가지 기술적인 문제들이 노출된다. 이러한 문제를 해결하려면 원천기술에 대한 이해를 기반으로 한 근본적인 기술 수정이 요구되는 경우가 대부분이다. 기술의 실용화 한계를 사전분석하고 이를 극복하는 연구를 원활하게 진행하기 위하여 강재적치처리장의 작업을 시뮬레이션할 수 있는 시뮬레이터를 제작하였다. 이를 기반으로 본 논문에서 제시하는 시스템을 개발하고 시스템의 신뢰성을 검증하였다. 많은 선박 제조공정의 작업처리과정이 정형화 되어 있지 않고 작업자의 경험에 의존 하여 수행되어 왔기 때문에 IT를 적용하기 위하여 작업자들의 업무 패턴을 모델화하는데 큰 어려움이 있었다. 따라서, 여러 현장 작업자들의 자문을 통하여 지속적인 기술 변경이 이루어졌다.

강재적치처리공정에서 사용되는 오버헤드 크레인을 로봇화하고 원격 제어 화면을 통해서 강재적치처리 작업을 수행하는 무인 자동화 강재적치처리 시스템을 구축하는 연구가 추후에 진행되면 흥미가 있을 것으로 생각한다. 이를 위하여 컴퓨터공학과 기계공학의 융합연구가 이루어질 필요가 있다.

참고문헌

[1] Yeongjong BMS, http://211.195.163.22/english/budae/10-budae.asp

[2] j. W. Kim, "Ubiquitous healthcare model based on context recognition," Journal of Korea Society of Computer and Information, vol. 15, no. 9, pp. 129-136, 2010
[3] Y. S. Jang, S. Y. Lee, "RFID Ubiquitous Public Information Documented Administration System Construction and Security Research," Journal of Korea Society of Computer and Information, vol. 15, no. 10, pp. 111-121, 2009
[4] M. Weiser, "Hot topic : Ubiquitous Computing," IEEE Computer, pp. 71 - 72, 1993
[5] M. Weiser, "The Computer for the 21st Century," Scientific American, pp. 94 - 104, 1991
[6] O. Yoosoo, J. Seiie, W. Woontack, "User Centered Context-aware Smart Home Applications," Korea Information Science Society, Lecture Notes in Software and Application, vol. 31, no. 1, pp. 111 - 125, 2004
[7] S. Shafer, B. Brumitt, B. Meyers, "The EasyLiving Intelligent Environment System," CHI Workshop on Research Directions in Situated Computing, pp 97-119, 2000
[8] S. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, D. Robbins, "The New Easy-Living Project at Microsoft Research," DARPA/NIST Workshop on Smart Space, pp. 127-130, 1998
[9] A. Fox et, al., "Integrating Information Appliances into an Interactive Workspace," IEEE Computer Graphics and Applications, pp. 54-65, 2000
[10] K. D. Anind, S. Daniel, D. A. Gregory, "A Context-based Infrastructure for Smart Environments," Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments, pp.114-128, 1999
[11] HP CoolTown, "http://www.cooltownstudios.com/"
[12] MIT Oxygen, "http://oxygen.lcs.mit.edu/overview.html"
[13] T. Suganuma, K. Yamanaka, Y. Tokairin, H. Takahashi, and N. Shiratori, "A Ubiquitous Supervisory System based on Social context Awareness," 22nd International Conference on Advanced Information Networking and Applications, pp. 370 - 377, 2008
[14] J. E. Bardram, "Applications of context-aware computing in hospital work: examples and design principles," Proceedings of the 2004 ACM Symposium on Applied Computing Ages, pp. 1574 - 1579, 2004

저 자 소 개

강 동 훈
2009 : 울산대학교 정보통신공학과 공학사
현재 : 사우스다코다 주립대학 석사과정
관심분야 : 컴퓨터공학
Email : eyes1216@nate.com

하 창 완
2009 : 울산대학교 컴퓨터공학과 공학사
현재 : 울산대학교 컴퓨터공학과 석사과정
관심분야 : 상황인지컴퓨팅,임베디드시스템
Email : imagineer81@naver.com

김 제 욱
2004 : 울산대학교 정보통신공학 공학사
2007 : 울산대학교 정보통신공학과 공학석사
현재 : 울산대학교 박사과정
관심분야 : 상황인지컴퓨팅, 무선통신
Email : kjw1623@gmail.com

오 훈
1981 : 성균관대학교 공학사
1993 : 텍사스A&M대학교 전산학과 공학석사
1995 : 텍사스A&M대학교 전산학과 공학박사
현재 : 울산대학교 컴퓨터정보통신공학과 부교수
관심분야 : 상황인지컴퓨팅, 임베디드시스템, 실시간시스템, 무선통신
Email : hoonoh@ulsan.ac.kr