

工學碩士 學位論文

멀티에이전트 기반의 자동화 컨테이너  
터미널 운영시스템 모델링에 관한 연구

A Study on Multi-Agent based Operation System  
Modeling for Automated Container Terminals

指導教授 林 宰 弘

2004年 2月

韓國海洋大學校 大學院

電子通信工學科

姜 警 遠

本 論文을 趙在晩의 工學碩士  
學位論文으로 認准함.

委員長 : 朴 東 國 (印)

委 員 : 孫 慶 洛 (印)

委 員 : 林 宰 弘 (印)

2004年 2月

韓國海洋大學校 大學院

電子通信工學科 姜 警 遠

# 목 차

Abstract .....	ii
제 1 장 서 론 .....	1
제 2 장 자동화 컨테이너 터미널 .....	3
2.1 자동화 컨테이너 터미널 개념 .....	3
2.2 국외 터미널 현황 .....	5
2.3 국내 터미널 현황 .....	9
2.4 자동화 시스템 구축 현황 .....	10
2.5 국내의 자동화 터미널 비교분석 .....	14
2.6 기존 시스템의 한계와 자동화 도입의 필요성 .....	17
제 3 장 에이전트 기술 동향 .....	20
3.1 에이전트 개요 .....	20
3.2 에이전트 응용 .....	20
3.3 멀티에이전트 .....	21
제 4 장 에이전트 통신 .....	29
4.1 에이전트 통신언어 .....	29
4.2 메시지 큐 .....	36
제 5 장 자동화 터미널 시스템의 모델 .....	44
5.1 자동화 터미널 시스템의 모델 .....	44
5.2 XML/JMS 기반의 통신모델의 장점 .....	55
제 6 장 결 론 .....	56
참 고 문 헌 .....	57

# Abstract

Trade between nations has been globalized since establishing the WTO(World Trade Organization). By lowering trading barriers under the WTO's system, trade in goods has been gradually increased. It requires global logistic system that transports goods in between nations. To save cost of product, cargo of product is containerized and container ships to carry container cargo is going to be bigger. However, container terminal that stevedore container cargo from ship is reached to the limitation of productivity to handle big ships. One of alternatives to increase productivity is to implement automated operations system for container terminal. In the market, there are many vendors to provide artificial intelligent modules to operate container terminal automatically. However, it requires hard efforts and long time to integrate modules because there are no standard interface between modules.

In order to integrate automated container terminal system easily and successfully, this paper proposes high-level XML/JMS(eXtensive Markup Language/Java Message Service) communication model and multi-agent based system architecture to share knowledges, solve problems, and achieve objectives by cooperating between autonomous and intelligent agents that are developed by 3rd party companies in the market.

This thesis analyzed current situation of advanced automated container terminal with case studies on implemented systems and difficulties to develop automated container terminal system, reviewed technologies of intelligent agent, communication and automation that unmaned automated container terminal is required.

# 제 1 장 서 론

독일의 통일에서 구소련의 붕괴, 동구권 국가의 체제변화와 대외 개방에 따라 탈이념적인 국제교류가 활발해지고, 우루과이라운드 (UR : Uruguay Round) 협상의 타결 및 세계무역기구(WTO : World Trade Organization)의 출범으로 각국은 무역, 투자, 금융 등 국제교역에서 장벽의 철폐 등 개방화를 추진하는 한편 유럽연합(EU : European Union), 북미자유무역협정(NAFTA : North American Free Trade Agreement), 아시아-태평양 경제협력체(APEC : Asia-Pacific Economic Cooperation) 등에 의한 경제블록화로 국제 분업체제를 구축하고 있어 기업경영활동의 범위도 국내에 국한되지 않고 국제적으로 많은 진출이 이루어지고 있다. 즉, 기업경영의 세계화가 활발히 이루어지고 결과적으로 국가간의 무역장벽의 철폐를 초래하고 국제화, 세계화를 지향하는 세계경제교류의 활성화, 촉진제 역할을 하고 있다[1].

정확하고 신속한 인도가 요구되는 21세기에는 물류경쟁력이 상품 경쟁력 못지 않게 중요해지며, 또한 세분화되는 소비자 요구에 보다 효과적으로 부응하면서도 리드타임을 단축하고 동시에 원가를 절감하기 위해 기업들은 대량 수송 수단으로 컨테이너선을 이용한다.

오늘날 대형 컨테이너 선사들은 범세계적인 해상 및 육상 수송망을 확충하고, 전용터미널, 창고 및 물류센타 등 물류거점의 확보와 기업의 세계화에 따른 욕구충족을 위하여 해운기업의 세계화를 적극 추진하고 있다. 특히 컨테이너선 분야에서 이루어지고 있는 세계화는 기업에게 화물추적 정보시스템의 제공이나 장비, 기기 관리를 위한 정보시스템 네트워크를 세계적으로 구축하는 것이다.

기업들의 수송시간 단축을 통한 비용절감의 필요성이 지속적으로

요구되면서 컨테이너 터미널은 기존의 운영 및 경영 개선을 통한 생산성 향상의 한계에 도달한 상황에서 보다 나은 생산성 증대를 위해 신기술에 의한 자동화 시스템을 도입하게 된다.

최근 컨테이너 터미널 자동화는 계획시스템(planning systems), 운영통제시스템(operating and monitoring systems), 하역장비(handling systems) 자동화 영역으로 구분해서 단일 에이전트를 구현하고, 이들 에이전트간의 통신으로 컨테이너 터미널의 운영을 부분적으로 자동화하고 있다. 현재 컨테이너 터미널 자동화를 위해 이용되고 있는 단일 에이전트는 필요한 정보를 기업의 데이터베이스로부터 수집하여 기능을 수행하고 있다. 그러나, 자연적, 인위적인 변경요인이 많은 항만수송 과정에서 각각의 에이전트는 변경된 정보를 데이터베이스로부터 다시 수집하여 이를 토대로 목적을 수행하는 과정을 거쳐야 하므로 많은 시간을 낭비할 뿐만 아니라, 정보교환이 빈번하게 이루어지는 에이전트간의 통신을 원활하게 할 수 없는 한계가 있다.

이를 위해 본 논문에서는 수시로 변경되는 정보를 인식하여 에이전트간의 정보교환을 위해 유동적으로 대처할 수 있는 XML(eXtensive Markup Language)과 JMS(Java Message Service)를 이용한 멀티에이전트간의 통신모델을 제안하고자 한다.

본 논문의 제 2 장에서는 자동화 컨테이너 터미널의 개념과 현재 국내외 터미널 현황을 소개하고, 제 3 장에서는 현재 에이전트 연구 동향을 살펴보고, 제 4 장에서는 에이전트간의 통신언어와 통신방법으로 본 논문에서 제안하는 XML과 JMS를 이용하는 새로운 통신모델을 제시하여 이를 기존의 에이전트 통신모델과 비교한다. 제 5 장에서는 본 논문에서 제안하고자 하는 자동화 터미널 운영시스템의 모델 및 장점에 대해서 기술하고, 끝으로 제 6 장에서 결론과 향후 연구과제에 대하여 기술한다.

## 제 2 장 자동화 컨테이너 터미널

### 2.1 자동화 컨테이너 터미널 개념

#### 2.1.1 자동화 시스템 도입배경

일반적으로 컨테이너 터미널은 기존 시스템의 생산성 증대가 한계에 도달한 상황에서 그 이상의 생산성 증대가 필요할 때 신기술에 의한 자동화 시스템의 도입을 추진한다. 이때 주 관심사는 장비의 이용률 극대화를 위한 컴퓨터 시스템 도입과 특수 장비 및 자동화 하역장비 개발이 된다.

컨테이너 터미널 자동화는 터미널의 운영을 기준으로 계획시스템 자동화, 운영통제시스템 자동화, 하역장비 자동화 등으로 구분하거나 자동화 대상을 기준으로 소프트웨어 측면과 하드웨어 측면으로 분류할 수 있다. 하역장비의 자동화는 운영통제시스템의 자동화가 선행되는 것을 조건으로 하기 때문에 하드웨어 자동화와 소프트웨어 자동화는 밀접하게 관련되어 있으며, 우선 순위를 따지자면 소프트웨어 측면의 자동화가 선결되어야 한다[1][2].

#### 2.1.2 자동화 시스템의 진전 단계

자동화 시스템을 도입하는 이유는 터미널 운영 측면에서 생산성 향상, 비용절감, 서비스 수준 향상, 인력 절감 등이고, 장비제조업체 혹은 정부 입장에서는 터미널 자동화 기술 축적이다.

자동화 시스템 도입은 다양한 요소기술을 필요로 하며 상호 밀접하게 관련되어 있기 때문에 기술 축적 정도에 따라 단계별로 이루

어지는 것이 타당할 것이다.

자동화 진전과정의 첫 단계는 계획시스템 자동화로써 터미널 운영이 성숙기로 접어들고 운영 기술 수준이 어느 정도 축적되는 단계에서 전문 인력에 의해 수행되던 선석 배정, 장치장, 본선하역 등의 계획업무를 인공지능 기법 등을 이용하여 지능화하는 것이다. 두 번째 단계는 계획시스템 자동화에서 한 단계 더 진전이 된 운영 및 통제 시스템의 자동화로써 계획 자동화를 바탕으로 하여 작업 지시 및 통제 부문을 계량 모형 및 인공지능 기법, 종합 관제시스템 등을 통하여 지능화하는 것이다. 세 번째 단계는 하역장비의 자동화 단계로서 인건비가 지나치게 높거나 장비 기사 등 전문 인력을 확보하기가 어려운 상황에서 적극 도입하게 된다. 향후 기술 진전이 이루어져서 수동 장비 운영을 능가하는 생산성이 보장되고 경제성이 보장될 때 일반화 될 것이다[3].

### 2.1.3 현황 및 전망

오늘날 터미널간의 경쟁이 심화되고 있는 상황에서 터미널 운영 목표는 생산성 향상 및 서비스 수준 제고로 설정될 수 있다. 일반적인 생산성 지표는 작업인원, 안벽길이, 크레인, 게이트 단위 수치당 컨테이너 처리량으로 나타나며, 선사에 대한 서비스 제고 관점에서의 서비스 지표는 선박의 정박시간을 단축할 수 있는 시간당 처리량으로 나타난다. 터미널이 서비스 경쟁체제로 전환되면서 정박시간 단축이 주 이슈가 되고 있으며, 이에 따라 선박 당 할당되는 장비 대수를 증가시키는 추세가 뚜렷하게 나타난다. 장비대수 증가는 장비간섭 문제로 한계가 있으며, 장비 대수가 증가할수록 최적장치계획 및 최적 장비할당을 위한 계획시스템 및 운영통제시스템의 중요성이 커진다고 할 수 있다.



컨테이너 터미널에서 사용되는 장비들은 생산성을 높이기 위해서 고도로 숙련된 장비 기사를 필요로 하나 국내외적으로 컨테이너 터미널 건설이 활발하게 이루어지면서 숙련된 장비 기사를 확보하는데 어려움이 있다. 또한 인건비의 비중이 높아지면서 인력 절감의 필요성이 높아지고 있다. 이러한 추세가 확산될 경우 장비 무인화를 통한 인력 수급 문제 해결 및 인건비 절감 방안이 모색될 것이다.

## 2.2 국외 터미널 현황

### 2.2.1 로테르담항

네덜란드 로테르담시 북해와 Rhine강 하구에 위치한 로테르담항은 17개의 컨테이너 전용선석과 1996년을 기준으로 컨테이너 물동량이 500만 TEU이다. 로테르담항에는 Delta 터미널과 세계에서 자동화가 가장 앞선 터미널로 알려진 Delta/Sea-Land 터미널이 있다.

Delta 터미널은 5만톤급 8선석과 암벽연장이 2.6km, 16기의 55~67톤급 젠트릭크레인이 있고, ECT(Europe Combined Terminal)에서 관리 운영을 담당하고 있다.

Delta/Sea-Land 터미널은 1993년 운영을 개시한 Delta 터미널 내의 Sea-Land사 전용 터미널로 3선석과 암벽연장이 960m, 연간 처리 물동량이 50만 TEU이고 ECT와 20년간 운영계약 체결하여 관리되고 있다. 또한 25대의 자동 장치크레인(Automated Stacking Crane : ASC)과 50대의 무인 운반대차(Automated Guided Vehicle : AGV)에 의해 수행되는 하역작업을 통제하는 자동 작업통제 시스템과 적부계획 시스템(Computer Aided Shipplanning System)으로

구성된 제 3세대형 터미널로 평면이동과 적재를 위한 무인자동화 설비를 설치 운영하여 컨테이너 운송은 완전 무인자동화 작업을 통해 인건비의 50% 정도를 절감하고 있다.

## 2.2.2 독일 함부르크항(HHLA)

독일 함부르크시 Elbe강 하구에 위치한 함부르크항은 27개의 컨테이너 전용선석과 1996년 기준으로 컨테이너 물동량이 307만 TEU이다. 함부르크 터미널은 하역작업을 완벽하게 통제하여 터미널의 이용률과 작업 속도를 높이며, 작업의 질을 향상시킴으로써 높은 경쟁력을 유지하고 있는 것으로 잘 알려져 있다. 70여대의 S/C(Straddle Carrier) 방식을 채택하고 있으며, 현재 연간 처리량이 150만에서 210만 TEU에 달하며, 터미널 용량을 최대한 이용하는데 중점을 두고 있다.

터미널 운영 시스템은 Satellite-controlled Container Handling으로 알려져 있는데 인공위성을 이용하여 터미널 계획과 작업을 완벽하게 통합하는 컨테이너 하역 시스템이다. 위성 시스템이 결합되어 수작업 없이 터미널 내의 모든 컨테이너 위치와 이동 상황을 실시간으로 파악하는 것이 본 터미널 운영 시스템의 가장 큰 특징이다. S/C가 야드상의 컨테이너를 들어올리는 순간 위성 시스템이 동시에 좌표점으로 그 컨테이너의 위치를 파악하며, 필요시 두 시스템에 의해 도출된 위치 차이를 보정한다. 실시간으로 파악된 개별 컨테이너의 위치 현황은 메인 컴퓨터에 전달되고 터미널 작업 계획에 사용되며, 사전에 수립된 작업 계획과 실시간으로 파악되는 야드 상황을 바탕으로 컴퓨터에 의해서 스트래들 캐리어의 하역 작업이 통제된다.

### 2.2.3 싱가포르

싱가폴항은 세계 최대의 컨테이너항으로 전체 컨테이너 화물량의 75~80%가 환적화물이며 대량화물 우대, 환적화물에 대한 하역료 및 보관료 우대, 독자적인 컴퓨터시스템에 의한 항만관리 등으로 국제경쟁력을 갖춘 항만이다.

싱가폴항은 본선에서 양하되는 컨테이너를 야드내 장치장으로 이송하는데 세계 최초로 2단적재 트레일러를 도입하였고, 야드내 크레인의 회전율을 극대화하기 위해 수출입 컨테이너 1개당 트랜스태이너 타임(Transtainer Time : TT)을 설정하여 반입·반출시간을 컴퓨터 시스템(PORTNET)으로 지정하고 있으며, 게이트에 자동응답장치를 설치하여 완전 무서류의 게이트 통과를 가능케하여 컨테이너 1개당 처리시간을 대폭 단축시켰다.

또한 화물, 야드, 선박, 라디오 통신 등 터미널 내의 모든 사항을 실시간으로 파악하며, 게이트, 선박, 야드 상황은 컴퓨터 그래픽으로 파악하고 있다. 정교한 컨테이너 하역 기기와 기술 그리고 컴퓨터 네트워크가 결합하여 신속하고 효율적이며, 신뢰할 수 있는 터미널 운영이 이루어지고 있다.

컴퓨터 통합 터미널 운영 시스템은 전문가 시스템 기법을 이용하여 선석배당, 선적계획, 야드 관리 등의 계획을 자동적으로 수행하는 기술적으로 가장 앞선 시스템으로 알려져 있다.

### 2.2.4 홍콩항

홍콩 구룡반도 북서측 주강 삼각주 입구에 위치한 홍콩항은 19개의 컨테이너 전용선석과 1996년을 기준으로 컨테이너 물동량이 1,328만 TEU이다.

싱가폴항과 달리 홍콩항은 좁은 장치 공간으로 인하여 장치장을 효율적으로 이용하는 점에 중점을 두고 있다. 따라서 홍콩의 컨테이너 터미널들은 전산 정보 시스템화를 통하여 터미널 운영의 효율 증대에 지속적인 노력을 기울이고 있다.

홍콩항은 자유무역항으로서 세계 제1위의 컨테이너 취급량을 자랑하는 민영화된 항만이며, 민간의 항만개발에서부터 모든 항만관리 운영을 담당하는 싱가포르-상해간의 유일한 심해항만이다. 아시아의 중심부에 위치하고 있으며 중국화물의 환적항 역할을 하고 있어 동북아 중심항만을 목표로 하는 부산항과 경쟁관계에 있다.

한편 홍콩항은 현재의 세계적인 위상을 유지하기 위해서 향후 2011년까지 대단위 항만개발의 필요성을 인식하고 1990년 항만개발 위원회(Port Development Board : PDB)를 설립하여 항만개발에 관한 모든 사항에 대하여 자문하고 있으며, 정부는 공유수면의 매립과 부지조성, 주요도로망, 지원시설용지, 산업용지 등을 공급함에 있어 투자자인 민간기업에 대해 금융, 세제, 제도적 지원을 아끼지 않고 있다.

## 2.3 국내 터미널 현황

### 2.3.1 터미널 시설

자성대부두(HBCT)와 신선대부두(PECT)는 컨테이너 처리량 면에서는 세계 10위권에 속하는 대형 터미널이나 심각한 시설 용량부족 문제를 안고 있다. 특히 장치공간 부족 문제가 심각하여 부산시내에 산재한 30여개 이상의 사설장치장을 이용하고 있는 실정이며 양산 ICD가 2000년 4월초에 개장함에 따라 단계적으로 사설장치장이전이 추진되고 있다. 4단계 터미널(감만)이 1998년 개장하면서 안벽 중심의 시설 부족 문제는 해결되었으나, 향후 경기 회복에 따른 물동량 증가를 예상할 때 시설 부족 문제는 여전히 대두될 것으로 예상된다. 이에 가덕도의 북항과 남항에 부산신항만을 각각 2006년 2008년에 개장을 목표로 건설중에 있으며, 선진 항만의 자동화기술을 적용할 예정이며, 광양항에 무인 자동화 터미널을 2008년까지 구축할 계획이다.

현재 부산항의 모든 터미널은 토지이용 효율이 가장 뛰어난 트랜스퍼크레인 시스템을 채택하고 있으며, 장치장의 이용효율을 높이기 위한 여러 방안들을 모색 중에 있다. 컨테이너 크레인의 경우 선박이 대형화함에 따라 5-6단 16열까지 갑판적할 수 있는 Post-Panamax급 대형선박에 적합한 초대형 크레인을 설치 운영하고 있다.

## 2.4 자동화 시스템 구축 현황

일반적으로 컨테이너 터미널 자동화는 하역 장비의 무인화 측면에서 생각하게 되는 데 좀 더 구체적으로 고찰하면 계획시스템 자동화, 운영통제시스템 자동화, 하역장비 자동화 등으로 구분해서 고려해야 한다. 이러한 분류는 터미널의 운영을 기준으로 한 것이다. 하역 장비의 자동화는 운영통제시스템의 자동화가 선행되는 것을 조건으로 하기 때문에 하드웨어 자동화와 소프트웨어 자동화는 밀접하게 관련되어 있으며, 우선 순위를 따지자면 소프트웨어 측면의 자동화가 선결되어야 한다[16].

컨테이너 터미널 자동화 분야는 요소 기술 측면, 소프트웨어 부문 그리고 하드웨어 부문으로 나누어서 살펴볼 수 있다. 요소기술 부문은 자동화의 근간이 되는 기술로서 주요 내용은 무선 데이터 전송 시스템, 네트워크 및 데이터베이스, 자동장비 및 컨테이너 인식시스템, 관제(monitoring) 시스템 등으로 구성된다.

소프트웨어 부문은 계획자동화 시스템, 운영통제 자동화시스템, 터미널 계획 및 운영 평가 시뮬레이션 모형 등으로 구분할 수 있다. 하드웨어 부문은 게이트 자동화, 안벽크레인(Quayside Crane), 야드크레인(Yard Crane), 야드트랙터(Yard Tractor) 등 하역장비 자동화 등으로 구성된다. 이들 각 부문은 상호 밀접하게 관련되어 있으며, 자동화 부문의 각 기능과 관련 핵심 기술 그리고 목적 등은 <표 2-1>과 같이 정리할 수 있다.

<표 2-1> 컨테이너 터미널 자동화 분야 및 특성

	주요 모듈	요소기술	주목적	시스템간 연관성
계획시스템 자동화	-선석배정 -Ship Planning -Yard Planning	-최적화 모형 -인공지능기법	-시간단축 -장치장, 장비 이용률 극대화	-독립적
운영통제 시스템 자동화	-Gate 통제시스템 -장치장 통제시스템 -장비 통제시스템	-인식기술 -최적화 모형 -인공지능기법	-장치장, 장비 이용률 극대화 -생산성 향상	-독립적 -계획자동화와 연 계시 효율 극대화
장비자동화	-게이트자동화 -크레인, 트랙터 무인화	-인식기술 -컨트롤 기술	-시간단축 -인력 절감	-독립적 -대규모 터미널의 경우 계획, 운영시 스템과 연계 필요

컨테이너 터미널의 자동화 시스템은 터미널 운영을 중심으로 할 때 유인자동화 시스템과 무인자동화 시스템으로 대별 할 수 있다. <표 2-2>와 같이 유인자동화 시스템은 소프트웨어 부문의 자동화로서 운영의 효율을 극대화하는 것이 목적이며, 무인자동화는 하역 장비의 무인화에 초점이 모아지며 무인화가 궁극적인 목적이 된다.

<표 2-2> 자동화 시스템 형태

	유인 자동화 시스템	무인 자동화 시스템
주목적	-운영 효율성 추구	-무인화 추구
형 태	-계획자동화+재래운영시스템+유인장비 -계획자동화+운영자동화+유인장비	-계획자동화+운영자동화+장비무인화 -운영자동화+장비무인화

세계적으로 무인 자동화 터미널 시스템을 적용한 선진국 사례는 <표 2-3>과 같다.

<표 2-3> 무인자동화 선진 터미널 적용 사례

구 분	ECT/DSL	싱가폴 (PPT)	Thamesport	HHLA	
자동화 시스템 형태	AGV-RMG	AGV-OHBC	AGV-RMG	AGV-2RMG	
자동화 구간	안벽 크레인	대선박 :유인 대 AGV:자동	대선박 :유인 대 Y/T :유인	대선박 :유인 대 Y/T :자동 대 AGV:자동	
	YT(이송)	AGV:자동운영	Y/T:유인	Y/T:유인	
	야드크레인	ASC: 대 AGV:자동 야드내 :자동 대 SC :자동	무인 OHBC: 대 Y/T :반자동 야드내 :원격 RMG: 반출입 :유인 야드내 :유인	무인 RMG: 대 Y/T :반자동 시프트 :자동 야드내 :자동	무인 RMG: 대 AGV:자동 야드내 :자동
	외부차량과 접점	SC:유인	무인 RMG: 원격조정/유인	무인 RMG: 원격조정/반자동	무인 RMG: 원격조정/반자동
	철송	MTS:유인			
	비규격	ASC/M: 특수 스프레더, 유인			YT:유인
	공컨	RS/SL:유인			
야드 형태	단일형 수직배열	buffer 야드 수평배열	buffer 야드 수평배열	단일형 수직배열	
장치단적수	3 단 6 열	8 단 10 열	4 단 9 열	4 단 10 열	

## 2.4.1 자동화 시스템 요소 기술 및 동향

선진 터미널에서 이용하고 있는 무인 자동화 시스템의 적용기술은 <표 2-4>와 같다.

### (1) DHST(Dual Hoist System)

컨테이너를 선박에서 하역하는 설비로써, ECT의 방식은 rail-span내에 AGV의 주행로가 위치하며, CTA의 방식은 backreach 하부에 AGV의 주행로가 위치한다.

### (2) 자동화 Yard Crane (RMG, ASC, OHBC)

컨테이너를 야드에 장치할 수 있는 설비이다. ECT, Thamesport, PSA에서 사용하고 있는 자동화 야드크레인들



Nelcon사, NKK사, MITSUI사, ABB사에서 공급하고 있다.

<표 2-4> 무인자동화 선진 터미널 적용 요소기술

	회사 및 장비	터미널	ECT	CTA	Hessnatie (Antwerp)	Thameport
		적용기술				
Software	운영시스템		NAVIS	NAVIS	COSMOS	자체개발
	SAP	SAP	미적용	적 용	미적용	미적용
	Gottwald	Routing	적 용	적 용	미적용	미적용
장 비	AGV	Transponder	적 용	적 용	미적용	미적용
	ATC	Anti-Sway	일부적용	적 용	미적용	적 용
		ATC의 Spread에 CCTV적용	미적용	적 용	미적용	미적용
		Encoder	적 용	적 용	미적용	적 용
	Q/C	DHST방식	일부적용	적 용	미적용	미적용
기 타	냉동컨테이너 모니터링	Reefer Con. 장비를이용한 Monitoring	적 용	적 용	미적용	미적용

### (3) AGV

컨테이너를 안벽과 야드 사이를 이송하는 차량이다. ECT, PSA, CTA 의 AGV 운영하고 있으며, Huisman-Itrec사, Kamag사, MES사에서 공급되고 있다.

### (4) 자동화된 터미널 운영시스템

자동화터미널 운영시스템은 COSMOS사, HPC사, NAVIS사, TSB사에서 공급하고 있다.

### (5) OCR방식 자동화 게이트 시스템

일반 컨테이너 터미널에서도 도입하여 사용하고 있으며, PERCEPTICS사, ASIA VISION사, ESC(VASTACK)사에서 공급하고 있다.

## (6) 냉동컨테이너 모니터링 시스템

일반 컨테이너 터미널에서도 도입하여 사용하고 있으며, YORK(REFCON)에서 공급하고 있다.

## 2.5 국내외 터미널 운영시스템 비교 분석

### 2.5.1 외국 터미널 운영시스템의 특성

특징으로서는 먼저 완벽한 터미널 하역작업 통제를 들 수 있다. 외국의 주요 터미널들은 CCTV와 전산망 그리고 통신망을 이용하여 터미널 상황을 실시간으로 파악하고 있어서 효율적인 계획 수립이 가능할 뿐 아니라 하역작업을 완벽하게 통제할 수 있다. 특히, 독일 함부르크 HHLA의 경우 인공위성을 이용하여 터미널 상황을 실시간으로 모니터링하여 계획과 작업 통제에 사용하고 있으며, 로테르담의 경우 무인 반송차와 무인 야드크레인에 의하여 장치 작업이 이루어지고 있다.

둘째, 전문가 시스템을 이용한 터미널 운영계획시스템 구축을 들 수 있다. 싱가포르 터미널을 제외한 나머지 터미널들은 야드 장치 계획과 본선 적부 계획을 컴퓨터를 이용하여 수작업으로 행하는 Computer-Aided Planning System을 이용하고 있다. 홍콩 국제 터미널의 경우 기존의 시스템을 부분적으로 자동화한 계획 시스템을 구축하고 있으며, 야드 장치 계획 및 적부 계획 시스템의 자동화에 가장 앞선 싱가포르의 경우 계획가들의 지식을 바탕으로 한 전문가 시스템 기법을 구현하여 계획을 자동으로 수행하는 시스템을 사용하고 있다. 이로 인하여 작업 계획에 소요되는 시간을 단축하여 터

미널의 효율을 높이며, 작업기기의 자동화를 가능하게 한다.

## 2.5.2 우리나라 터미널 운영시스템과 비교

전세계적으로 컨테이너 터미널의 자동화는 자동 컨테이너 인식시스템이 실용화 단계에 있으며, 안벽크레인, 야드크레인 등 하역장비는 인력에 의존하는 유인 중심이 주류를 이루고 있다. 소프트웨어 측면에서는 장치장 및 장비 이용 극대화를 위한 computer-aided 수준의 계획시스템이 구축되어 있고 장비 이동시간을 최소화하는 운영통제시스템을 도입하는 단계에 있다. 자동화 수준으로서는 인공지능 기법을 도입한 계획시스템이 일부 주요 터미널을 중심으로 적극 도입되고 있으며, 운영 통제 부문은 아직 뚜렷한 진전을 보이지 않고 있는 실정이다. 통합운영 상용화 시스템은 (주)토탈소프트뱅크의 C3IT(Command Control Comunication and Intelligence for Terminal)가 유일한 시스템이다. 현재 부산항 4단계 일부 터미널에 시스템을 구축하였으며, 모니터링과 작업 프로세스 중심으로 구축된 통합 운영시스템이다. 그러나 대부분의 부산항 터미널의 운영시스템은 자체개발한 것으로써 개별 시스템 중심으로 개발되어 제반 하드웨어 기술과의 연계 및 통합화가 미흡하며, 자동화 측면에서 불 때 최적화 및 지능화 기술이 미흡한 한계를 내포하고 있다.

## 2.5.3 기존 자동화 터미널의 한계

ECT/HHLA 시스템을 제외하고 기존 터미널에 구축된 자동화 시스템은 부분적으로 자동화시스템을 채택하고 있다고 할 수 있다. 또한 터미널 제반 여건 특성에 맞는 방향으로 자동화가 진전되고 있

으며, 보편적 자동화 시스템은 없다고 할 수 있다.

기술 수준 측면에서 기존의 무인 자동화시스템 역시 요소기술의 한계를 안고 있다고 할 수 있다. 즉, ECT 시스템의 경우 처리 가능한 최대 물동량이 약 120만 TEU이며, 안벽길이 기준 약 1000m 규모인 것으로 밝혀졌다. 현재 장치 단적수는 평균 1.5단 정도이며, 향후 최고 3단적까지 시스템을 개선할 계획으로 알려져 있다.

#### 2.5.4 기존 자동화 컨테이너 터미널 시스템의 한계

상용 자동화 운영시스템 및 자동화 장비의 에이전트간 표준이 없어 개발 및 시스템 통합구축에 장시간과 많은 노력이 필요하며, 경우에 따라서는 시스템 구축을 실패할 수도 있다. 실제 자동화터미널 시스템 구축에 실패한 주요 원인 중에 하나는 각 장비의 개발 업체가 다르게 되어 장비간 인터페이스를 통한 전체 시스템 통합 작업시 문제점을 지적하고 있다. <표 2-5>은 자동화 터미널 시스템 구축의 실패 사례를 보여주고 있다.

기존 운영시스템의 경우에 DB를 매개로한 모듈간의 인터페이스로 자동화터미널의 다양한 요구사항을 수용하기 위한 에이전트의 변경이 어렵다. 즉, 개별 응용 에이전트의 인터페이스를 변경하기 위해서는 DB를 변경하여야 하며, 이는 에이전트와 관련없는 어플리케이션에도 변경이 필요한 경우가 발생한다. 다른 인터페이스 방법으로 JMS나 소켓을 통한 메시지를 이용하지만, 에이전트를 제공하는 업체별로 독특한 프로토콜을 사용하므로, 통합 운영시스템의 관점에서는 표준화 및 통일되지 않은 여러 인터페이스를 유지하여야 한다. 이들 두가지 방법은 운영시스템에 새로운 에이전트의 추가/교체 및 통합에 경직된 구조를 가지고 있다.

<표 2-5> 자동화 터미널 구축 실패 사례

터미널	항 목	실패 사유
PSA	시스템 통합	각 장비의 개발 업체가 다르게 되어 장비간 인터페이스를 통한 전체 시스템 통합 작업시 문제점 발생
가와사키	운영 시스템	장비 메이커(maker)에 의한 자동화 운영시스템 구축 실패 ① 프로세스 특성을 고려하지 못한 클라이언트/서버 설계 - 모든 프로세스가 서버에 있어 서버 프로세서의 과부하로 성능과 처리 속도 저하 - 선박 및 야드 계획과 같은 시뮬레이션 프로세스의 효율성 저하 - 운영 어플리케이션 설계와 사용자인터페이스의 유연성 저하 ② 데이터의 특성을 고려하지 않은 클라이언트/서버 설계 - 데이터베이스 서버에 모든 데이터가 위치해 네트워크상에 데이터 과부하로 성능 저하 - 디스플레이 속도의 저하 ③ 시스템 개발방법론의 부재 - 전통적인 시스템 개발 방법 적용 (업무단계별 연계가 선행이며, 개별 컴포넌트의 합을 전제)

## 2.6 기존 시스템의 한계와 자동화 도입의 필요성

컨테이너선의 대형화에 따른 항만생산성 향상 및 컨테이너 항만의 경쟁력 강화를 위해 인건비 절감 및 터미널 토지 활용성 제고가 필요하며, 이와 같은 필요성으로 인해 자동화 컨테이너터미널 건설이 하나의 대안으로 부각되고 있다. 우리나라도 자동화 컨테이너터미널을 광양항에 건설하기로 하고 기본계획을 수립 및 설계중에 있으며, 관련 핵심기술을 개발하고 있다.

### 2.6.1 생산성 수준 분석

1999년 기준 부산항의 연간 총 작업시간당 생산성은 약 20box이다. 8,000 TEU 컨테이너선 및 12,000 TEU선의 LPC(Lifts per Call)는 각각 3,600 lifts와 5,500 lifts로 추정된다. 8,000 TEU와 12,000 TEU 선박에 Q/C를 각각 5대와 6대를 할당할 경우 1999년 부산항 생산성 기준으로 laytime은 각각 43시간, 63시간이 소요된다. laytime을 24시간으로 제한할 경우 8,000 TEU와 12,000 TEU 컨테이너선의 요구생산성은 Q/C당 30box와 38box가 되어야 하며, 12,000 TEU의 경우는 laytime을 30시간으로 할 경우 생산성이 30box가 되어야 한다.

따라서 VLCS(Very Large Container Ship)에 대비한 터미널의 생산성은 현재 시간당 20box에서 최소 30box로 약 50%향상이 필요하다.

### 2.6.2 생산성 향상 대안으로서의 자동화 컨테이너터미널

자동화 Q/C, AGV, RMG 등의 장비, 야드 운영통제방식의 개선, 자동화된 터미널운영시스템을 적용하는 자동화 컨테이너터미널이 생산성 향상의 대안으로 제시되고 있다. Q/C 측면에서 기존 채래식 Single Trolley Q/C의 이론적 생산성이 40~45box/hr인데 비해 Dual Hoist Q/C나 수직순환식 Q/C의 경우 60~100box/hr까지 향상이 가능하다.

야드 운영통제방식에서는 장비풀링(equipment pooling)이나 듀얼 사이클(dual cycle)과 같은 복잡하고 정교한 야드운영이 자동이동장비(AGV)와 무인 야드크레인을 사용하는 자동화터미널에서 가능하다.

며, 이로 인한 생산성 향상은 20%정도이다. 또한 자동화터미널에서는 무인자동화장비를 사용하기 때문에 장비들의 시간 예측이 가능해 실시간 시스템의 자동화된 터미널 운영시스템 적용이 가능하다. 자동화 시스템을 통하여 다음과 같은 효과를 얻을 수도 있다.

- § 야드내 rehandling을 줄일 수 있음
- § 실시간으로 최적의 자원할당, 양적하 작업 수행
- § pooling system이나 dual cycle과 같은 복잡한 야드통제 가능 (fixed rule →adaptive rule)
- § 야드운영 여유 향상 가능 (50%→80%)
- § cargo-closing time없이 최적의 planning 가능 (서비스 향상 및 yard space 약 20%절감 가능 )

### 2.6.3 자동화 터미널 경제적 타당성 분석

5만톤급 컨테이너선 4선석을 채래식으로 건설할 경우 토목비 및 운영설비(장비) 등 상부시설비가 6,179억원이 소요되나, 자동화로 건설할 경우 총 6,546억원이 추가 소요되어 전체 공사비의 약 6.2%정도가 더 드는 것으로 나타난다. 특히 장비비는 채래식에 비해 자동화터미널의 경우 약 14%정도 더 투자되는 것으로 산정되나, 자동화터미널의 경우 연간 인건비가 채래식 터미널에 비해 46% 수준에 불과해 연간 약 214 억원 정도 인건비 절감효과가 발생한다. 연간 운영비 전체로는 자동화 방식이 약 564 억원으로 채래식의 758억원에 비해 약 25% 정도 절감된다.[3]

## 제 3 장 에이전트 기술 동향

에이전트란 다른 에이전트나 프로세스와 통신하면서 특정 목적에 대하여 사람의 지침이나 간섭 없이 자율적으로 작업을 수행하는 개체를 말한다.

초기 에이전트에 관한 연구는 분산인공지능, 로보틱스, 인공생명, 분산객체 컴퓨팅, 인간과 컴퓨터 상호작용, 지능형 인터페이스, 지능 탐색과 필터링, 정보검색, 지식획득, 최종 사용자 프로그래밍 등의 많은 분야에서 각각 연구되어 왔다. 이렇게 다양한 분야에서 에이전트 개념을 사용함에 따라 에이전트에 대한 정확한 개념없이 에이전트라는 용어가 광범위하게 사용되었다.

일반적으로 에이전트는 자율성, 반응성, 사회성, 사전 행동성, 추론능력, 적응성, 이동성, 학습성 등의 성질을 가지고 있다[4].

### 3.1 에이전트 응용

#### 3.1.1 에이전트 응용분야

에이전트는 이미 산업 전반에 걸쳐 많은 분야에 적용되어 이용되고 있으며, 산업부문, 공공부문 등 거의 모든 분야에 걸쳐 정보화의 진행 속도가 빨라짐에 따라, 더 많은 분야에서 인간의 힘들고, 귀찮은 일들을 대신해 줄 것으로 보인다.

응용의 예로서 첫째로, 산업응용 분야로서 프로세스 제어, 제조업 관련 응용, 항공 교통 제어, 물류관리 등이다. 둘째로, 상거래 응용으로 물품 검색 및 계약, 입찰 등의 전자상거래 응용과 의사결정을 보조하는 비즈니스 프로세스 관리 등이다. 셋째로 의학 응용으로 환자상태를 체크하는 환자감독, 건강관리, 넷째로 개인 보조 분야로 사용자의 디렉토리 관리를 위한 디렉토리 관리 서비스, 정보 여과와 검색을 위한 정보관리 서비스, 여행계획 서비스 등이다.



### 3.1.2 에이전트 응용의 문제점

아직까지는 에이전트 응용을 설계하고 구현하는데 있어 전문적인 지식이나 시스템에 대한 이해가 부족한 경우가 많이 있다. 또한, 에이전트 시스템을 구현하기 위해 지원되는 소프트웨어가 빈약하고, 그러한 시스템을 구현할 때 발생하는 문제점에 대한 이해가 부족하며, 대부분의 에이전트 시스템은 틀에 짜인 도구와 기술을 사용하여 구현되므로 다른 시스템에 쉽게 적용될 수 없다는 문제점을 안고 있다[5].

대부분의 경우 에이전트 시스템의 주요한 구성요소가 구현되기도 전에 많은 하부 구조가 요구되기 때문에 이러한 경우 대부분의 개발자는 새로운 에이전트 시스템 개발의 경우에 기존의 것을 사용하지 못하고 하부구조를 재구현해야만 한다.

## 3.2 멀티에이전트(Multi Agent)

### 3.2.1 멀티에이전트의 개념

멀티에이전트 시스템에서 에이전트란 “분산 환경에서 상호 협력을 통해 작업을 수행하는 컴퓨터 프로그램”을 말한다.

멀티에이전트 환경은 작은 단위의 응용 에이전트들이 모여서 이루어지는 에이전트 사회를 형성한다. 이 에이전트 사회의 구성원인 각 개별 응용 에이전트는 특정 도메인에 한정적인 작업의 수행만을 담당하는 단순 에이전트의 형태를 가지고 있으며, 따라서 복잡한 작업이 요청되는 경우, 하나가 아닌 다수의 응용 에이전트들의 상호작용과 협동에 의해 수행이 이루어진다.

결과적으로 멀티에이전트 시스템에서 대두되는 주요 쟁점중의 하나는 하나의 복잡한 작업을 달성하기 위해 필요한 에이전트의 조정이나 협동을 위해 다수의 응용 에이전트를 어떻게 효과적으로 관리하느냐 하는 것이다.

에이전트의 관리는 각 응용 에이전트의 생성에서부터 종료까지의 에이전트 생명주기를 제어하는 것을 말한다. 에이전트 표준기관인 FIPA(Foundation for Intelligent Physical Agents)에서 제시된 에이전트 도메인 생명주기 모델에서는 에이전트의 상태를 미등록, 활성, 일시정지, 종료 등으로 구분하고 각 상태의 변환을 위한 전이함수를 제안하였다.

이와 같은 에이전트의 상태를 관리해주는 모듈을 에이전트 관리 시스템(Agent Management System : AMS)이라 부르며, 이 AMS는 응용 에이전트와 대비하여 모든 에이전트 사회에 필수적으로 존재해야 하는 시스템 에이전트의 속성을 지닌다.

소수의 응용 에이전트만이 존재하는 단순 에이전트 사회의 경우 모든 에이전트를 미리 구동시킨다 하더라도 시스템 부하에 별 영향을 끼치지 않으므로 문제가 되지 않을 수도 있고, 그 결과로 이제까지는 복잡한 에이전트 관리에 관심을 가지지 않았다.

하지만 최근에 대두되는 실세계 환경에서는 하나의 에이전트 사회가 수십 또는 심지어 수백의 에이전트로 구성될 수 있으며, 따라서 매우 많은 수의 에이전트를 모두 미리 구동시키는 것은 시스템에 미치는 부하가 상당히 크게 되고, 자원의 한계로 볼 때 많은 문제점을 야기할 수 있다.

특히 이질적인 에이전트들로 구성된 에이전트 사회에서는 이러한 동적 에이전트 관리가 쉽지 않다. 에이전트가 이질적이라 함은 여러 측면에서 논의되지만 간단히 표현하면 자바와 C, Perl 등 서로 다른 프로그래밍 언어나 UNIX와 Windows와 같은 플랫폼을 이용하

여 개발된 에이전트를 가리킨다. 물론 이질적인 에이전트간의 상호 작용을 원활히 해주기 위한 연구는 많이 진행되고 있으며, 그 대표적인 예가 KQML(Knowledge Query and Manipulation Language) 이라고 할 수 있다. 제 4장에서 KQML과 XML(eXtensible Markup Language)을 비교하여 자세히 설명하겠다.

### 3.2.2 관련 연구

멀티에이전트 기반구조에 관한 기존 연구로는 EMAF, JATLite (Java Agent Template Lite), JAFMAS(Java-Based Framework for Multi-Agent Systems), FIPA AP(Agent Platform) 등의 네 가지를 간단히 기술하고, 특별히 에이전트 관리 측면에서 이질적 에이전트와의 협동을 지원하는 지와 동적인 에이전트 관리가 이루어지는 지에 대해 비교 분석하고자 한다.

#### (1) EMAF

EMAF는 기존의 소프트웨어 시스템들이 서로 다른 환경 하에서 구축되는 이질성으로 인해 정보 공유나 협동을 효과적으로 조정하지 못하는 어려움을 해결하고자 하는 목적으로 출발하였다.

EMAF는 에이전트의 관리와 제어를 담당하는 블랙보드 기반의 조정 에이전트와 다수의 응용 에이전트로 구성되어 있으며, 에이전트 통신언어를 통해 다른 에이전트와 협동을 시도한다.

조정 에이전트는 각 에이전트의 능력과 기능 정보를 메타지식의 형태로 저장하고 있으며, 에이전트 상태 테이블 등을 유지하여 간단한 형태의 에이전트의 동적 구동을 실현하고 있지만 에이전트의 프로세스만을 구동하는 단계에 머물렀고 따라서 여러 에이전트가 협

동해야 하는 작업에 대한 동적 관리를 하지 못했다.

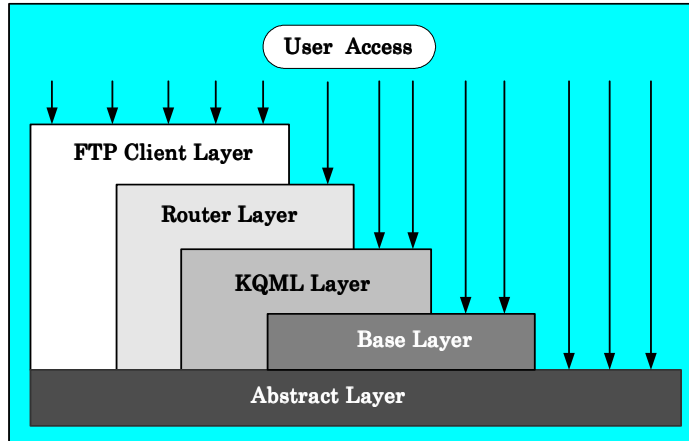
## (2) JATLite

JATLite는 에이전트 기반구조 개발을 위한 자바 패키지로서, 자바 언어를 사용하며, KQML을 이용한 메시지 교환방식의 통신 틀을 제공한다. 또한 기반구조상에서 에이전트들간의 협력작업을 조정하기 위한 에이전트 라우터를 제공한다. 라우터를 통하여 등록된 에이전트들간에 전송하고자 하는 메시지를 메시지 큐에 저장하여, 메시지의 전송이 즉시 이루어지지 못하여도 자동적인 재전송을 가능하게 한다.

JATLite의 시스템 구성은 <그림 3-1>과 같이 자바 클래스들을 각각의 특성에 따라 여러 개의 계층(Layers)으로 구성하여 이를 통해 유지보수를 용이하게 하는 구조를 가지고 있다.

이 시스템은 에이전트 통신과 초기 구동 등의 기본적인 기능이 지원되지만, 사용자와 시스템간의 인터페이스가 제공되고 있지 않으며, 협력작업을 할당할 때 메시지 수신자를 매번 사용자가 직접 설정해 주어야 하기 때문에 작업의 분할 및 규합 기능이 효율적이지 못하다.

또한 에이전트의 자동 구동과 종료가 지원되지 않으므로 많은 에이전트를 포함하는 복잡한 시스템의 경우 관리가 불가능하며, 라우터를 중심으로 하는 중앙집중형 구조이기 때문에 병목현상이 일어날 경우 시스템이 효율적으로 가동되기 어렵다.



<그림 3-1> JATLite 계층클래스 구조

JATLite 시스템은 자바 클래스들을 각각의 특성에 따라 4개의 계층으로 구성하여 이를 통해 유지 보수를 용이하게 하는 구조를 가지고 있다. 이 시스템의 문제점은 라우터를 중심으로 하는 중앙집중형 구조이기에 병목현상이 일어날 경우 시스템이 효율적으로 가동되기 어렵게 될 수 있으며, 동적 에이전트 관리를 고려하지 않으므로 모든 에이전트를 사용자가 직접 구동시켜야 하는 단점이 있다.

### (3) JAFMAS

JAFMAS는 자바 언어를 이용하여 일반적인 에이전트 구조와 이러한 에이전트를 구현하기 위한 클래스 집합을 제공한다. 모두 16개의 자바 클래스들을 통해 시스템에서 여러 가지 기능을 제공한다. 즉, 직접 통신과 방송 통신과 같은 통신 기능과 KQML을 기반으로 하는 메시지 구조를 제공하고, 협력작업에 대해서 일정한 규칙을 기반으로 에이전트간의 대화를 통해 공동작업을 조정한다. 이는 JAFMAS 내의 각 에이전트들이 가지고 있는 계획 즉, 시스템내의 다른 에이전트에게 바라는 동작에 대한 선택적인 과정 개념을 집합

의 형태로 전달함으로 이루어진다. 이러한 규칙의 수행을 위한 인터프리터가 지속적으로 작동하며, 다중 스레드를 지원하여 위에서 기술한 각각의 기능들을 동시에 수행할 수 있도록 각 작업에 대한 다중 스레드가 할당된다. JAFMAS는 시스템내의 각 에이전트가 서버와 클라이언트의 기능을 상황에 따라 모두 수행할 수 있도록 내부에 모든 구조를 가지는 특성을 지닌다. 따라서 비 중앙집중적인 구조를 가지게 되나, 그를 위해 각 에이전트별로 많은 정보와 복잡한 구조를 가지게 되고, 시스템의 가동을 위한 부하가 커지게 되며 사용자가 일일이 각각의 에이전트를 콘솔 윈도우에서 수행시켜야 하는 단점을 갖고 있다.

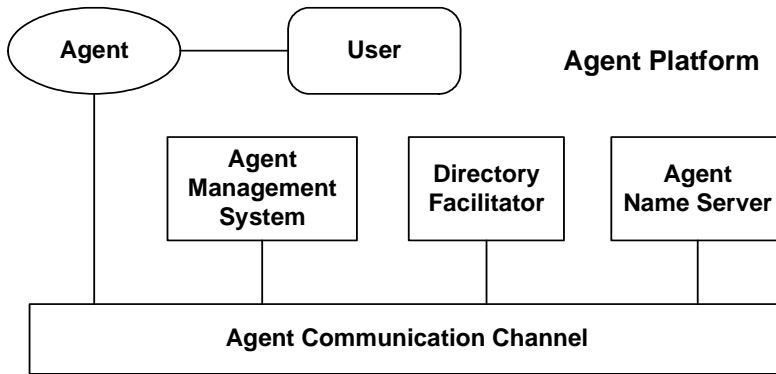
JATLite와 JAFMAS는 둘 다 자바 언어로 구현된 기반구조이며 KQML을 이용한 메시지 교환과 작업 수행을 진행한다. 하지만 두 시스템 모두 응용 에이전트도 자바로 구성해야 하며 동적 에이전트 구동에 대한 고려는 하지 않았다.

#### **(4) FIPA AP**

FIPA는 에이전트 표준화 단체이며, 새로 생성되는 에이전트 기반 응용 프로그램들과 서비스, 그리고 기술의 성공을 증진시킬 목적으로 만들어졌다. 이를 위해서 에이전트 기반 응용 프로그램들과 서비스와 기술들을 시기 적절한 방법으로 만들고, 전세계적으로 통용되는 기술이 될 수 있도록 이들간의 통용성을 강화하기 위한 AP를 제시하고 있다.

FIPA AP는 에이전트가 생성되고 동작되며 종료되는 일련의 과정을 지원하기 위한 핵심 기반구조를 제공한다. 에이전트가 같은 플랫폼 내 혹은 다른 플랫폼에 있는 다른 에이전트와의 협동작업을 위해서는 최소한 하나의 플랫폼에 등록이 되어야 하며 AP는 플랫폼

에 소속된 에이전트에 대해 에이전트 레벨의 서비스, 즉 에이전트 등록, 에이전트 이름과 주소 매핑, 에이전트 통신, 서비스 찾기, 에이전트 관리 등을 제공한다. FIPA AP의 구조는 <그림 3-2>와 같은 구성요소들이 내재되어 있으며 이러한 다수의 AP들이 더 큰 시스템의 구성요소로서 역할을 하게된다.



<그림 3-2> FIPA AP 구조

FIPA가 제안한 AP 구조에 따라 실제적으로 구현할 때, 그 특성상 보완을 요하는 문제점들이 발생한다. 첫째, 현재의 AP 구조에서는 AMS를 통해 에이전트 라이프 사이클을 관리하고 있지만 기본적인 가정이 에이전트를 처음 구동시키는 작업은 사용자에게 의해 이루어진다.

따라서, 하나의 커다란 작업을 위해 필요한 에이전트를 모두 초기에 구동시켜야만 한다. 이 개념은 적은 수의 에이전트로 구성된 간단한 멀티 에이전트 시스템에서는 효과가 있을 지 모르지만 워크플로우 제어를 위한 에이전트 시스템과 같이 매우 많은 수의 에이전트가 요구되는 경우에는 시스템 동작이 정상적으로 구현되기 어렵다. 그러므로 필요한 경우에만 구동되고 작업을 마치면 종료될 수 있는 에이전트의 자동 구동과 종료에 대한 기법이 요구된다.

둘째, 공동작업을 원하는 에이전트의 요청 메시지 전송과정이 상대

적으로 많기 때문에 작업수행의 효율이 저하될 수 있으며, 작업의 할당에 있어 요청 에이전트가 직접적인 선택을 하게 되므로 효율적인 기능을 제공하지 못한다.

셋째, AMS가 에이전트의 등록 및 접속확인, 에이전트의 구동, 작업 할당, 최종결과 규합 등의 역할을 수행함으로써 부분적으로 중앙집중적인 작업수행 구조를 가지게 되며, 이에 따라 병목현상이 일어날 수 있다. 또한, 많은 작업을 수행하게 되는 AMS의 구조상 구현의 복잡성이 높아진다.



## 제 4 장 에이전트 통신

### 4.1 에이전트 통신언어

멀티에이전트 시스템을 구성하는 주요한 이론적 근거는 다수의 에이전트가 공동체 사회를 이룸으로써 하나의 에이전트로부터 얻을 수 있는 이상의 부가가치를 제공해 준다는 것이다. 이러한 부가가치는 에이전트 서로가 협동하면서 얻을 수 있는 것이다. 만약 모든 에이전트가 전체 시스템에 대한 완벽한 정보를 가지고 있다면 에이전트들은 서로 통신할 필요 없이 완벽하게 대등한 입장에서 서로의 작업을 수행할 것이다.

그러나 아주 간단한 시스템의 경우를 제외하고 완벽한 지식을 가진 에이전트를 만든다는 것은 불가능하고 시스템에 대한 부분적인 전문지식만을 가지는 것은 가능하다.

대부분의 에이전트들은 서로의 정보를 교환하여 에이전트가 필요로 하는 모든 정보를 가지고 있지 않아도 에이전트간의 상호작용을 통해 보다 많은 서비스를 사용자에게 제공해야 한다. 따라서 각각의 독립된 에이전트는 자신의 목적을 위해 외부 에이전트 정보를 이용하거나 자신의 정보를 제공할 수 있는 메커니즘을 확보해야 한다. 이에 대한 가장 보편적인 해결책이 에이전트 사회에서 통용될 수 있는 공통된 정보교환 및 대화 수단을 제공하는 에이전트 통신 언어이다.

#### 4.1.1 KQML

에이전트간의 정보 교환 및 공유를 위해서는 공통된 언어를 지식 표현에 사용하거나 적어도 서로 번역 가능한 언어를 사용하여야 하며 서로간의 통신이 가능하여야 한다. 지식 표현을 위한 공통 언어

로 KIF(Knowledge Interchange Format)와 동일한 시스템간 혹은 이종 시스템간의 실시간 지식 교환을 위해 KQML을 사용하고 있다.

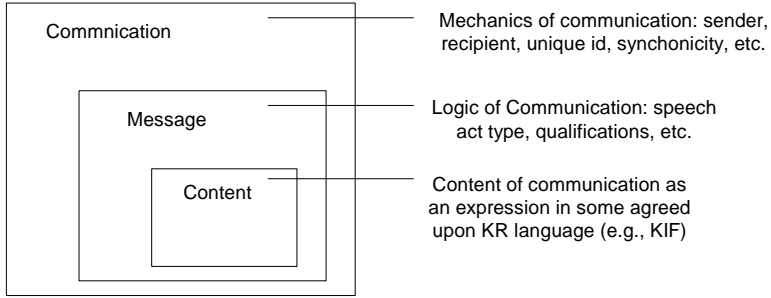
### **(1) 화행 이론(speech act theory)과 KQML**

Labrou와 Finin는 에이전트간의 의사소통 언어를 교환하고자 하는 내용에 대한 태도를 전달하는 매개물이고 의사소통의 내용은 단정, 요구, 질의 등이 되어야 한다. 이러한 경우는 발화 자체가 어떤 행위가 되는 경우로서, 언어학이나 언어철학의 화용론에서는 이를 화행(speech act)이라고 하고, 그때의 발화를 Austin의 이분법적 구분에 따르면 performatives, 즉 행하는 발화라고 한다.

KQML 언어는 멀티에이전트간의 의사소통 행위를 이러한 화행 이론에 기인하여 형식화되고 KQML의 기본 어휘를 performatives라고 한다. 이는 서로 교환하고자 하는 내용에 대한 태도를 표현하며 에이전트로 하여금 그러한 태도를 다른 에이전트에게 전달함으로써 자신의 요청을 처리하기에 적합한 다른 에이전트를 찾도록 한다.

KQML은 에이전트간의 지식 교환을 위한 메시지의 형식이자 상호간의 통신규약으로 지식을 표현한 것이 아니라 표현된 지식을 통신으로 전달하기 위한 형식이다. KQML은 실제 실어 나르는 지식 즉, 내용에 독립적이며 내용의 표현에는 여러 가지 언어가 쓰일 수 있다.

KQML은 <그림 4-1>에서 알 수 있듯이 내용 계층, 메시지 계층, 통신 계층의 3계층 구조로 이루어져 있다. 내용계층은 전달될 지식, 메시지 계층은 전달할 내용의 특성, 통신 계층은 통신을 위한 특성이 표현된다.



<그림 4-1> KQML의 3단계 계층

에이전트간 통신을 지원하기 위해 제안된 performative는 에이전트간의 실제 대화를 위한 것으로 <표 4-1>과 같다.

<표 4-1> Faciliator와 일반 에이전트를 위한 performative

통신 대상	이름	의미
일반 agent → Faciliator	advertise	agent가 자신의 정보를 알림
	unadvertise	agent의 정보를 삭제 요청
	request	서비스 요청
	accept	Faciliator의 서비스 요청
Faciliator → 일반 agent	wait	서비스 요청 agent에게 대기 명령
	kill	특정 agent를 kill
	dency	서비스 불가를 알림
	call	특정 agent에게 서비스 지시
	sleep	특정 agent의 작업중단 지시
일반 agent → 일반 agent	send-data	자료의 전송
	receive-data	자료의 수신
	receive-ok	정상적인 수신 확인

performative는 등록, 접속, 에이전트 열람, 주소요청, 에이전트 구동 등의 기능으로 분류된다.

## (2) KQML 메시지와 Performatives

KQML은 에이전트간에 교환되는 실제의 언어표현 자체가 아니라, 단지 그에 관한 태도를 표현하는 기본 어휘들의 집합으로 이루어져 있다. 따라서 KQML은 메시지의 유형을 정의하며 의사소통 행위의 논리적 측면을 나타내는, performative와 그 매개변수로 이루어진 메시지 층과 실제 전달하고자 하는 내용을 다양한 언어와 ontology를 기반으로 담고 있는 내용 층이 구별되어 있다.

Performative는 에이전트간에서 시도되는 허용 가능한 행위들을 정의하는 언어의 기본어휘이다. KQML의 명세를 위한 제안서에는 서로 다른 환경에서 개발되고 활용되는 에이전트가 서로간의 의사소통시 공통적으로 사용하기 위해 규정된 “reserved performatives”의 리스트를 아래 <그림 4-2>에서 보여주고 있다.

- **Basic query performatives**
  - evaluate, ask-if, ask-in, ask-one, ask-all
- **Multi-response query performatives**
  - stream-in, stream-all
- **Response performatives**
  - tell, achieve, cancel, untell, unachieve
- **Capability-definition performatives**
  - advertise, subscribe, monitor, import, export
- **Networking performatives**
  - register, unregister, forward, broadcast, route

<그림 4-2> reserved performatives

또한 어떠한 performative든지 일관적으로 사용해야 하는 매개변수라는 의미에서 :sender, :receiver, :from, :to, :in-reply-to, :reply-with, :language, :ontology, :content가 reserved performative parameters의 예로 사용되고 있다.

## 4.1.2 XML

### (1) XML의 개요

1996년 4월 팀 브레이(Tim Bray)는 구조화된 문서를 위한 XML의 설계목표를 발표했고, 1996년 11월 보스톤에서 열린 SGML(Standard Generalized Markup Language) '96 컨퍼런스에서 처음 XML 초안이 발표되었다. 1998년 2월 10일에는 W3C 권고에 따르는 XML 스펙 1.0이 발표되고 오늘날까지 꾸준히 연구되어 오고 있다[6][7].

웹에서 구조화된 문서를 전송 가능하도록 설계된 XML은 표준화된 텍스트 형식의 생성 언어로 SGML의 일부분이며 SGML보다 간결하고 인터넷에서 사용 가능한 문서를 표현하는 표준이다.

XML은 전체를 지원하는 소프트웨어의 개발이 용이하지 않는 SGML과 제한된 태그로만 분류되어 지정되지 않은 태그의 사용이 불가능하다는 HTML의 단점을 극복하였다. 그리고 XML은 구조화된 문서를 정의하고 자유롭게 태그를 정의할 수 있는 SGML과 인터넷에서 손쉽게 하이퍼미디어 문서를 제공할 수 있는 HTML의 장점을 그대로 취합하고 있다[8][9].

HTML과 SGML의 필수적인 기능만을 취합하고 복잡하고 어렵거나 비효율적인 부분은 제외함으로써, XML은 두 언어의 핵심적인 장점을 그대로 보유하고 있다.

### (2) XML의 특징

XML은 웹에서 정보를 전달하기 위하여 제안된 메타 언어이다. XML을 위한 소프트웨어는 URL(Uniform Resource Locator)을 HTML의 링크를 접근하듯이 접근할 수 있는 기능이 내재되어 있다.

일반적으로 XML 인스턴스 중에서 링크와 DTD(Document Type

Definition)는 URL의 사용이 가능해야 한다. 만약, 파서 및 기타 소프트웨어에서 URL을 지원하지 않는다면, XML 스펙에서 언급한 다양한 기능들이 무용지물이 된다. XML을 지원한다는 것은 인터넷 환경을 고려한다는 의미가 되므로 XML은 웹에 대한 활용이 높다.

XML에서는 기본적인 XML 문법에 맞게 인스턴스가 작성되어 있다면 DTD가 필요 없는 정형화된 문서가 있다. XML을 보다 효율적으로 사용하기 위해서 DTD를 작성하고, DTD에 따라 인스턴스를 생성해야 하는 과정은 초보자에게 문서 작성에 대한 부담감을 주게 된다. DTD가 웹 브라우저에 내장된 HTML과 달리, XML은 사용자에게 따라 DTD의 존재여부가 결정된다. 반드시 DTD 규칙을 따라야 하는 HTML보다 XML이 문서 작성하는데 용이하다.

XML의 스타일 언어인 XSL(eXtensible Stylesheet Language)은 사용자가 정의하는 문서에 대한 표현을 적용하는 메커니즘을 제공한다. 즉, XSL은 하나의 내용에 대하여 다양한 형식으로 표현할 수 있도록 해 준다.

예를 들어, 아래한글로 만든 제품 설명서를 웹이나 매뉴얼로 사용하기 위해서는 프로그래밍 작업을 다시 해야 한다. 즉, 동일한 내용일지라도 사용되는 환경에 따라 또는 표현형식에 따라 새로 작성을 해야 한다. 반면에, XSL을 이용하여 XML 문서의 내용을 표현하면, XML 문서 자체에 대한 수정작업 없이 XSL만 수정하면 된다.

XML은 데이터를 고유 태그로 표시하기 때문에 보다 정확하고 빠른 검색을 할 수 있다. 예를 들어, 책에 대한 검색의 경우 작가, 제목, 또는 다른 분류항목 등에 의한 표준방식으로 책을 손쉽게 분류할 수 있고 사용자가 필요한 부분만 세분화하여 검색할 수 있다.

#### 4.1.3 에이전트 통신을 위한 KQML과 XML

에이전트간의 지식 교환을 위한 메시지의 형식이자 상호간의 통신 규약인 KQML과 에이전트들이 서로 공유하여 활용할 수 있는 에이전트 온톨로지에 관련된 연구에서 제시되고 있는 XML을 비교해보면 다음과 같다.

### **(1) KQML을 사용할 경우**

에이전트가 받은 KQML performative를 처리하는 독립적인 모듈로서 확장성과 외부 에이전트와의 연결 및 효과적인 사용을 위한 인터페이스를 지원하는 KQML 처리기와 에이전트에 전달하고자 하는 지식을 KIF로 표현하고 KQML을 통해 메시지 통신을 하기 위한 KIF 번역기가 필요하다.

KQML 처리기의 하부구조에 속한 KIF 번역기는 KQML 처리기를 통해 전달받은 메시지를 번역하여 에이전트에 보내거나 외부로 보내고자 하는 정보를 KIF 형태의 메시지로 변환하여 외부 에이전트의 라우팅 정보를 찾아준다.

메시지 통신 기능을 담당하는 KQML 처리기는 에이전트에게 정보를 등록하는 메시지, 정보를 알려주는 메시지, 정보에 대해 질문하는 메시지를 처리하는 역할을 한다.

### **(2) XML을 사용할 경우**

진정한 에이전트 기반에서 에이전트들이 상호 의사소통을 하면서 거래, 협상하기 위해서 기본적으로 필요한 것이 에이전트들이 서로 공유하여 활용할 수 있는 에이전트 온톨로지이다.

특정 분야에서 사용되는 표준 어휘들의 모음이라고 정의되는 온톨리지를 구성하는 기본 언어로는 KQML, MCF(Meta Content Format), XML 등이다.

XML은 에이전트의 외부언어인 KQML과 달리 KIF와 같은 내부

언어를 별도로 가지고 있지 않아도 될 뿐만 아니라 분리되어 있는 에이전트에서 관련 정보를 찾고, 그것을 처리하여 포괄적인 보고서로 만드는 등, 관련된 정보를 찾고 처리하는데 특화된 능력을 가진 질의 언어이다. 또한, 에이전트가 필요로 하는 정보가 있는 데이터베이스의 종류와 무관하게 정보를 질의하여 얻을 뿐만 아니라 각각의 데이터베이스에서 얻은 정보를 취합하여 하나의 새로운 정보로 재생할 수 있는 질의 언어이다.

KQML에 비해 XML은 객체 및 객체들 간의 관계를 묘사하기 쉬울 뿐만 아니라 의미론을 표현하고, 에이전트 사이의 보다 높은 차원의 상호 운용성을 구축할 수 있다.

## 4.2 메시지 큐(Message Queue : MQ)

### 4.2.1 MQ의 도입 배경 및 목적

오늘날 대부분의 네트워크를 사용하는 에이전트에서 반드시 고려되어야 하는 조건은 각 협력 시스템들간의 연동성, 가용성, 시스템의 트랜잭션 성능이다.

대부분의 많은 멀티 에이전트를 이용해 통합적인 운용을 위해서는 TCP/IP(Transmission Control Protocol/Internet Protocol) 방식의 기존의 프로토콜을 사용해왔었다. 분명 이 방식들은 부인할 수 없는 효율적인 운영방식에 대한 인증을 마친 상태이다. 그러나 기존의 검증된 프로토콜을 사용하는 방식들로부터 제기되고 있는 문제점은 이러한 효율성 이외의 연동성과 가용성에 대한 것이다.

TCP/IP 방식은 동기식의 연결지향형의 통신방식을 사용하고 있



지만 네트워크가 정상적으로 동작하지 않을 가능성은 언제나 충분히 내포하고 있다. 물론 이런 경우를 대비하여 충분한 예비회선을 구축하고는 있지만 거기에 따른 비용이 문제되고 있다. 또한 각각의 에이전트들이 분산되어진 경우 발생하는 동작과 복잡한 데이터베이스간의 변경설정들은 순차적으로 또는 동시에 발생할 수 있는 여건을 가지고 있다. 사건들이 언제나 발생하여 동작들에 대한 요구가 지속적으로 발생한다면 동기적인 통신방식은 적당하다고 할 수 있으나 안전성이 충분히 보장되는 상태에서는 적당히 속도를 조절하여 대역폭의 사용여유와 항상 실시간으로 동작을 동기화 시키는데 대한 비용을 감소시켜 네트워크에 대한 부하를 줄여나갈 수 있을 것이다.

또한 기존의 프로토콜을 응용한 통신 시스템은 확장에 대한 유연성에 있어서 오늘날의 서비스들이 제공하는 환경보다는 상당히 유연하지 못할뿐더러 새로이 등장하는 서비스들과 비교해 볼 때 확장에 드는 추가적인 비용 또한 상당하다고 할 수 있다.

위에서 언급한 것과 같이 오늘날 멀티 에이전트 운용에 있어서 변수가 되는 요인들은 나날이 증가하고 있는 환경적인 변수의 증가와 기존의 복잡한 프로토콜에 의존하는 방식에 따른 복잡한 기술소요를 충족시키기 위해 드는 많은 구축비용에 대해서 한계성이 지적되고 있다.

어떤 시스템을 유지하기 위한 비용에서 중요한 사항들은 개발비용, 시스템 관리비용, 작업 단위별 비용, 확장성 비용 등이다. 그러나 기존의 시스템들은 이중 어떤 면에서도 유리하다고 할 수 없는 실정이다.

따라서 이에 대한 대안으로 제시되고 있는 것이 엔터프라이즈 메시징 시스템(Enterprise Messaging system) 또는 MOM(Message Oriented Middleware)이라 불리는 어플리케이션간 메시징 시스템이

다. 이러한 메시징 시스템들은 미들웨어란 이름으로 어플리케이션간의 통신을 책임지게 된다.

#### 4.2.2 MQ의 개요 및 특징

미들웨어란 다른 소프트웨어 응용 프로그램이나 데이터베이스로부터 어느 정도 추상화를 제공함으로써 프로세스 간 통신이 프로세스 자체 개발과는 별도로 처리되도록 하는 것을 목적으로 하는 광범위한 소프트웨어를 말한다.

메시지 기반 미들웨어는 어플리케이션간의 통신논리를 처리하게 되는 것에서 나아가 메시징에 더욱 초점을 맞추고 보안층의 제공과 버퍼 크기 최적화 등의 다른 관리층의 서비스를 포함하도록 되어 있다. 또 메시지 기반 미들웨어는 상대방 어플리케이션이 동작하지 않는 순간에도 메시지들의 송수신이 가능하며 비동기 통신 방식을 사용하기 때문에, 송신 어플리케이션이 수신 어플리케이션이 응답을 보낼 때까지 기다리지 않아도 된다는 것을 의미한다.

어플리케이션간의 통신에 이용되는 메시지 기반 미들웨어의 구성도는 <그림 4-3>과 같다.



<그림 4-3> 메시지 기반 미들웨어

메시징 시스템은 결합되지 않고 분리된 어플리케이션이 비동기적으로 신뢰성있게 통신할 수 있도록 해 준다. 메시징 시스템 아키텍

처는 일반적으로 각 어플리케이션 사이의 관계를 클라이언트/서버 모델에서 점대점 관계로 변경한다. 즉, 각각의 에이전트는 다른 에이전트에 메시지를 전송할 수 있으며, 또한 다른 에이전트로부터 메시지를 전달받을 수 있다.

메시징 시스템의 특징은 동적이고, 신뢰성 있고 유연한 시스템을 구현할 수 있도록 해 주며, 그에 따라 시스템의 나머지 부분에 영향을 주지 않고 하위 어플리케이션의 전체적인 구성을 변경할 수 있다.

메시징 시스템은 어플리케이션간의 통신에서 메시지를 이용하여 객체간의 결합성을 약화시키고 안전성, 확장성, 동기화, 보안 등의 문제에 대한 해결책을 제시하고 있다.

오늘날 메시징 시스템을 이용하고 있는 대표적인 소프트웨어는 마이크로소프트사의 MSMQ(Microsoft Message Queue)와 선마이크로시스템사의 JMQ(Java Message Queue)이다.

앞에서 언급한 대표적인 두 가지 메시지 기반 미들웨어인 마이크로소프트사의 MSMQ와 선마이크로시스템사의 JMS를 간단히 비교해 보면 다음과 같다.

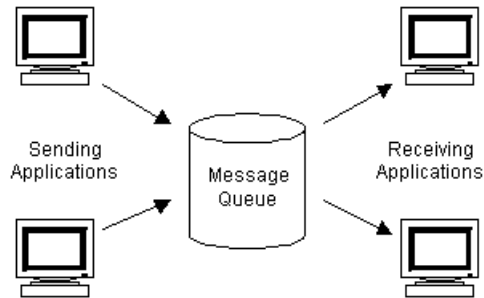
### 4.2.3 MSMQ

MSMQ는 각기 다른 시간에 실행되는 응용 프로그램들이 일시적으로 오프라인이 될 수도 있는 혼합된 네트워크와 시스템이 통신하는 것을 가능케 한다. 응용 프로그램은 메시지 큐로 메시지들을 보내고 메시지 큐로부터 메시지를 읽는다. <그림 4-4>는 응용프로그램에 의해 사용된 메시지를 메시지 큐가 처리하는 과정을 나타낸다.

메시지 큐는 보장된 메시지 배달, 효율적인 라우팅, 보안 및 우선순위에 기반을 둔 메시지 서비스를 제공하여 비동기와 동기 시나리

오에 대한 솔루션을 구현하기 위해 사용되고 있다.

MSMQ는 서버와 클라이언트의 계층적인 구조에 의존하고, 최상위 루트는 큐의 위치와 큐의 메시지를 사용하는 다른 서버와 클라이언트들에 대한 정보가 담겨진 기본 엔터프라이즈 컨트롤러라고 불리는 객체이다. 이는 그 하부 개념인 사이트와 단일 네트워크를 관리하게 된다.



<그림 4-4> 메시지 큐의 역할

MSMQ에서 메시지는 전송중이거나 처리중인 메시지들이 모여있는 메시지 큐에 저장 가능하며 큐는 디스크 기반의 큐나 메모리 기반의 큐 등으로 나뉘어 안정성이나 속도 등에 따라 선택이 가능하다.

MSMQ에서 사용되는 메시지 전달방식은 기본 엔터프라이즈 컨트롤러를 중심으로 구성된 트리간의 통신으로 이루어진다. 트리는 때로는 복잡한 경로를 따라야 하고 이때 MSMQ가 적용하는 것이 각 연결간의 가중치를 통한 비용이라는 것을 설정하도록 하여 효율적인 경로를 찾아내도록 한다[11].

#### 4.2.4 JMS

자바를 기반으로 하는 메시징 서비스는 자바 메시지 서비스(Java

Message Service : JMS)라는 이름으로 J2EE(Java 2 Enterprise Edition)의 일부분으로 자바 개발자들이 엔터프라이즈 메시징 시스템의 공통적인 특징에 접근하기 위해 표준 응용프로그램 인터페이스(Application Program Interface : API)를 제공한다.

JMS는 머신 아키텍처와 운영체제에 상관없이 클라이언트 어플리케이션의 이식성뿐만 아니라 메시징 제품에 상관없는 이식성을 갖도록 해 준다. 즉, JMS로 작성된 클라이언트 어플리케이션은 JMS 호환 메시징 시스템에서 변경할 필요 없이 사용할 수 있다. 또한 많은 메시징 시스템 벤더들은 그들의 제품에 맞게 JMS를 구현하고 자바를 사용하여 시스템의 기능에 접근할 수 있도록 하고 있다[12].

JMS는 발행/구독(publish/subscribe) 모델과 점대점 모델을 지원하고 임의의 자바 객체로 구성된 메시지 타입을 생성할 수 있도록 해 준다.

발행/구독 메시징 시스템은 메시지의 전송에 참여하고 있는 생산자와 소비자가 이벤트를 사용하여 통신한다. 생산자는 이벤트를 “발행”하고 소비자는 “구독”한다. 즉, 생산자는 이벤트를 생성하고 소비자는 자신이 관심을 갖고 있는 이벤트를 수신하여 그 이벤트를 소비한다. 생산자는 특정한 토픽에 연관된 메시지를 토픽에 등록한 소비자에게 전달한다.

점대점 메시징 시스템에서 메시지는 개개의 소비자에게 전달되며, 각각의 소비자는 들어오는 메시지를 저장하는 큐를 갖고 있다. 메시징 어플리케이션은 지정된 큐에 메시지를 보내고, 클라이언트는 큐로부터 메시지를 읽어들인다[13].

위에서 언급한 것과 같이 발행/구독과 점대점을 모델로 하여 일대다와 일대일 통신형태를 구현한다. JMS는 이 두 가지 모델을 기반으로 해서 중앙 집중적 또는 분산적으로 구성되는 통신 형태에 적합한 환경을 제공하도록 하고 있다.

JMS는 독립적인 응용프로그램이 가진 복잡성과 호환성 문제를 고려하지 않고 메시지 지향 미들웨어의 모든 이점을 실현할 수 있도록 한다. JMS는 개발자들을 위해 표준화된 API 및 공통 메시징 전략을 제공하는 산업 표준인 JMS 1.0.1에 의해 구현되므로 JMS가 하는 일은 메시지 큐가 실현해 보이는 것과 유사하다.

#### 4.2.5 MSMQ와 JMS의 비교

MSMQ와 JMS는 이질적인 환경에 대한 이식성은 뛰어나지만 현재 MSMQ가 마이크로소프트 단독의 밴더에 의해서 개발되고 있어 제품의 완성도 측면에서 JMS를 능가하고 있는 것처럼 보이지만 JMS 개발은 현재 활발히 진행 중이다.

MSMQ와 JMS는 마이크로소프트의 MTS에서 발전한 COM+와 EJB(Enterprise Java Bean)를 하부구조로 한다.

MSMQ는 대부분의 저차원 개발언어를 지원하는 Common Language Runtime 구동환경에서 내부적으로 MSIL(Microsoft Intermediate Language)라는 저차원 언어로 구성되어 있다.

JMS는 자바 바이트 코드를 가지고 자바 가상 머신(Java Virtual Machine : JVM)에서 동작한다는 의미에서 JMS의 개발 밴더들이 다른 언어에서도 자바 바이트 코드를 생성할 수 있다.

JMS의 개발 밴더들은 MSMQ보다 늦은 출발에 비해 현저히 빠른 속도로 진화를 거듭하고 있다. 이는 JMS를 설명할 수 있는 중추적 하부구조인 EJB 2.0의 발표 시기가 마이크로소프트의 COM+에 비해 3년정도 늦었다는 점과 JMS가 Sun을 중심으로 하여 기존의 MOM에 관련한 많은 밴더들이 공동으로 참여하고 있는 실정임을 감안한다면 이것은 크게 문제되지 않는다. MOM 제품의 개발이 활발하게 이루어져 하루가 다르게 변하고 있다는 점을 감안한다면 이

런 격차는 중요하게 생각할만한 문제가 아니다[14][15].

현재 MSMQ의 문제점은 마이크로소프트사의 제품들에 의존해야 하므로 MSMQ가 독자적으로 독립적 기능을 수행할 수 있는 솔루션들을 모두 갖추고 있지 않는 한 확장성에 관한 문제에 직면하게 될 것이다.

이에 반해 JMS의 구성은 개발 밴더들의 집합체이므로 충분한 확장성에 대한 보장을 하고 있다. 또한, MOM 시장은 비용이 많이 드는 사업으로 치부되어져 왔으나 JMS를 비롯한 여러 개발 밴더들의 꾸준히 노력한 온 결과 MOM 제품은 보다 저렴한 가격으로 시장에 공급되고 있어 가격경쟁력에서 MSMQ를 능가하고 있다. 즉, MOM이 추구하는 확장성과 이식성과 경제적인 측면에서 MSMQ보다 JMS는 훨씬 유리한 입장이다.

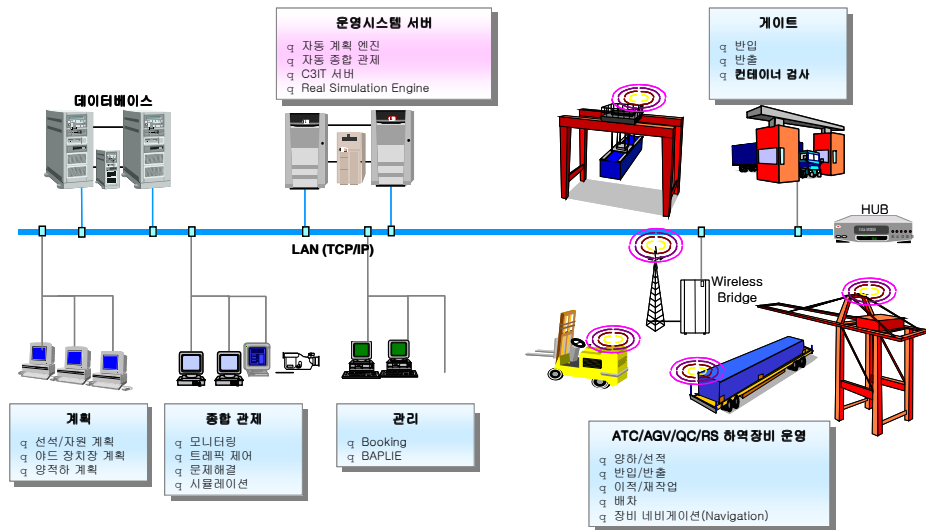
# 제 5 장 자동화 터미널 시스템의 모델

## 5.1 자동화 터미널 시스템의 모델

### 5.2.1 자동화 터미널 운영시스템의 구성도

자동화 터미널의 운영시스템은 다양하고 복잡한 개별 응용 에이전트들이 분산환경에서 개별적으로 동작할 수 있는 사용성(usability), 유연성(flexibility), 확장성(scalability), 상호운용성(interoperability)이 확보된 환경이어야 한다. 또한 다양한 형태의 응용 에이전트들이 실행될 수 있는 기반과 에이전트간 적시에 적절한 통신을 할 수 있어야 한다.

<그림 5-1>은 자동화 터미널 운영시스템의 개념적인 구성도이다.



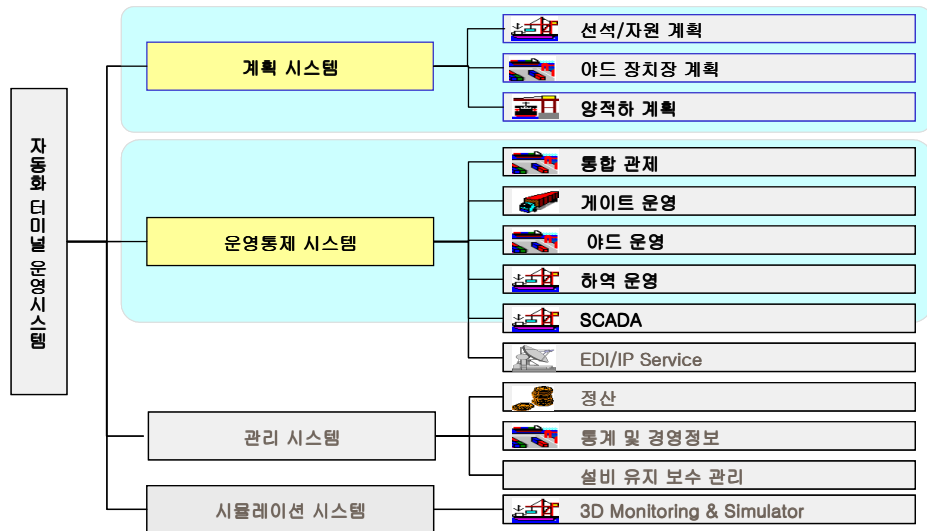
<그림 5-1> 자동화 터미널 운영시스템 구성도

응용 시스템의 구성 3요소를 데이터, 프로세스, 프리젠테이션으로



할 때, 이들 구성이 시스템 아키텍처의 설계에 따라 자유자재로 배치되어 분산 처리될수 있는 멀티티어(multi-tier)아키텍처가 가능하여야 한다. 즉, 데이터는 데이터베이스 서버에 의해서 관리되며, 비즈니스 프로세스 및 공유되는 개별 응용 에이전트는 운영시스템 서버에 의해서 운영되고, 이동 장비 및 특정 자동화 장비(게이트)의 운영을 위한 개별 응용 에이전트는 무선통신 환경 및 특정 분산 네트워크에서 운영시스템 서버내의 에이전트들과 상호 협력하여 동작한다. 또한 사용자와 운영시스템간 상호 인터페이스를 위한 프리젠테이션을 위한 클라이언트 응용에이전트가 운영된다.

<그림 5-2>는 자동화 터미널의 운영시스템에 대한 소프트웨어 기능을 보여 준다.



<그림 5-2> 터미널 운영시스템 소프트웨어 기능도

일반적인 터미널 운영시스템은 크게 기능에 따라 계획 시스템, 운영통제시스템, 관리 및 시물레이션 시스템으로 나눌 수 있다.

자동화 터미널 운영시스템은 일반 터미널 운영시스템과 비슷한 시스템의 구성을 가지고 있으나, 각 서브 시스템별 모듈에 대한 기능은 훨씬 다양하다.

<표 5-1> 자동화 터미널 운영시스템의 소프트웨어 모듈1

모듈	기능 및 주요 고려사항
선석계획	안벽의 활용도와 터미널 물류를 최적화하는 방향으로 선석 배정 장치장까지의 AGV 운행 거리 고려 입항예정시간 등의 변경에 대비할 수 있는 의사결정 지원 시스템
양적하게획	양적하 순서 결정 알고리즘에 기반한 의사결정 지원 시스템 자동화 장치장의 상황에 적합한 계획논리 개발 (부하균등화, 재취급 최소화) 본선작업 중 신속한 계획 변경이 가능한 계산속도 필요 CC 활용도 및 양적하 작업의 효율을 극대화 할 수 있도록 계획 CC 작업 영역 제약 및 상호간의 간섭 고려
철송계획	화차에 대한 적재 및 하역 효율 최대화 철송장 운영과 관련된 장비 활용 극대화 관련 장치장의 ATC 작업 부하를 고려한 철송 작업 시간대 결정
시뮬레이션	계획을 포함하여 운영에 관한 다양한 대안들에 대한 성능평가 다양한 운영 시나리오 비교 평가 여러 가지 운영 성능에 관한 지표 및 분석결과 제공
운영관계	터미널의 상태, 작업 진행 상황, 문제 발생 등을 파악할 수 있어야 함 운영자의 수동 개입이 손쉽게 이루어질 수 있어야 함 운영 관련 각종 파라미터와 우선순위를 수동 조정 가능해야 함 고 품질의 GUI 필요
이벤트 관리	각 입력 이벤트 별로 운송모델 수립(예외상황 구분) 실시간 작업 스케줄링(우선순위 고려), 작업 동기화
장치 위치 결정	최적 장치 위치의 실시간 결정 작업부하 균등화 및 재취급 최소화
작업 배정	자동화 장비, 수동 장비, 작업자 등에 대한 작업을 배정 간섭을 피하고 장비 활용도를 높일 수 있도록 자원의 상태 고려
작업관리	작업을 해당 장비 수퍼바이저에게 전달하고 결과를 보고 받음 Event Manger의 요구 시간을 준수할 수 있도록 작업 진행을 관리
장비 수퍼바이저	AGV의 경우 라우팅과 교통 제어를 수행 신뢰성 확보가 가장 중요한 과제
예외 처리	예상하지 못한 상황 발생에 유연하게 대처할 수 있도록 함
게이트 운영	반출입 예정, 반출입 작업, 특수 컨테이너 반입작업 관리 및 정보 제공 블록 배정정보, 반출입 결과정보 등의 제공
야드 운영	작업지시/결과정보, 장비이력/상태정보, 보세, 특수 컨테이너 정보 제공 발생 가능한 모든 상황에 대응 가능 해야 함
하역 운영	모선 입출항, 양적하 작업, CC운영 정보 등의 제공 장치장 현황 정보, 컨테이너 특성 등을 고려
철송 운영	철송하역작업, 철송정보, 작업 배정현황 등 고려 발도착 예정정보를 근거로 화차 양적하 계획 수립 열차 도착시점에 하역장비에 작업지시

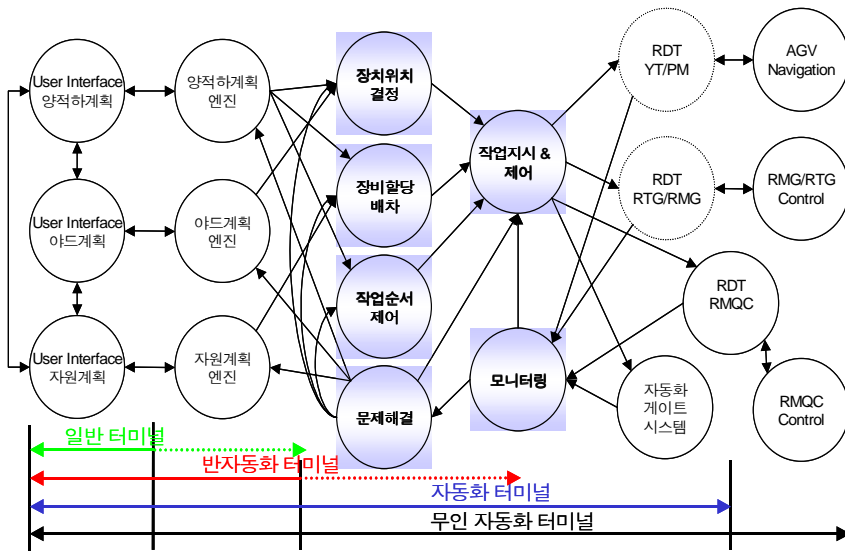
<표 5-2> 자동화 터미널 운영시스템의 소프트웨어 모듈2

모듈	기능 및 주요 고려사항
CFS 관리	수출입작업, CFS현금수납, CFS반출입 정보, 재고관리 작업의 특성에 따라 규정된 내용의 정보를 전달
일반장비 작업관리	유인장비작업지시, 무인장비 작업지시, 장비상태 정보 관리
On-Dock관리	수출입운영, ECS관리, On-Dock 정보 관리 선입선출 알고리즘을 적용하여 장기체류 및 이적발생 최소화
EDI 관리	대고객 EDI(선사, 운송사, 화주), 대정부 EDI(세관, 항만관리국) 웹 기반 표준 XML 데이터 교환양식이용
일반관리	재무/회계, 인사/노무, 자재/구매, 경영/기획 업무 관리
정산관리	정산자료, 청구, 매출 관리 등 관세/청구유형 정보 등
고객관리	고객관계관리(CRM), 고객불만처리, 정보서비스 제공 등
경영 의사결정지원	각종 경영 정보 생성, 장비가동/하역/반출입/장치 실적 정보 등 하역실적, 집안실적, 물량통계, 생산성실적/분석 통계 등
종합물류정보 관리	장치장, 본선 및 On-Dock정보, 보세 관련 정보, CFS, 정산 정보 장치현화, 자원배정, 모선 입출항, 컨테이너 정보 등
장비 관리	장비 이력관리, 정비 관리, 부품관리, 도면관리, 제원/코드관리 예방정비계획/주기, 부품별 재고, 긴급정비, 가용장비, 운행정보 등
시설물 관리	시설물 목록/도면 관리, 유지보수, 하자보수
환경/보안관리	오염물규제/감시, 위험물관리/감시, 재난예방/접수/상황 시설물감시, 주차, 차량등록, 방문자정보
게이트 자동화	무정지 통과 또는 정지시간 최소화 반출입 사전정보 미등록 차량에 대한 처리 특수 규격의 컨테이너 반출입을 위한 레인 운영
가변안내 시스템	운영 시스템과 연계하여 장치장 진입 트럭에 대한 통제 내용 안내 행선지(블록 등) 표시, 대기/주정차/진행 지시, 터미널 내 긴급상황 안내 등
DGPS 시스템	DGPS 기준국 설치 ATC의 스프레더, Y/T, R/S, 냉동컨테이너 관리차량 등의 위치 파악
CCTV 시스템	넓은 지역에 대한 세밀한 감시, 특정지역에 대한 집중 감시 기능 운영관리실에서 원격제어 운영 장애 시 자동 녹화 기능 및 관련부서 통보 기능
출입보안 시스템	시설 중요도에 따라 인가자의 출입 제한 가능 모든 출입자의 출입기록을 저장하여 출입현황 파악

터미널 자동화는 운영뿐만 아니라, 관리, 지원, 보완 시스템등 전반적으로 정보시스템을 구축 적용하는 추세이다. <표 5-1>과 <표 5-2>는 국내 광양항에 건설될 자동화 터미널의 정보시스템의 각 모듈들이다. 또한 기존 터미널에서 사용되거나 혹은 검증된 상용소프

트웨어를 구매하여 통합하는 방식으로 프로젝트를 진행할 예정이므로 모듈간의 통합이 큰 과제이다.

자동화 터미널 운영시스템 구축시 필요한 개별 응용 에이전트간의 통신 필요한 핵심 에이전트들은 작업 계획과 작업을 관제하는 응용에이전트들과 실제 자동화 장비를 제어 운영하는 무인자동화 기능을 수행하는 에이전트들이다. 이들 에이전트간의 상호 관계 및 구성도는 <그림 5-3>과 같다.



<그림 5-3> 터미널 운영시스템 에이전트 구성도

본 그림에서 보는것과 같이 무인 자동화 터미널을 구축하기 위해서는 다수의 응용에이전트들이 필요하고, 에이전트간 상호 작용이 일반 터미널에 비해 훨씬 복잡하다.

### (1) 자원계획 엔진

각 작업에 대해 시설, 취급장비 그리고 인력을 어느 시점에 얼마만큼 할당할지와 어떻게 계획·운영할 것인가에 대해 의사결정하여 자원능력을 효율적으로 분배함으로써 어떠한 자원도 시스템의 한계

요소가 되어 전체 생산성을 저하시키는 경우가 없도록 하는 것이 목적이다.

터미널의 처리능력, 효율성 등은 여러 작업부하와 장비의 결합이 어떻게 이루어지는가에 의존한다. 취급장비할당, 선석할당, 게이트 할당의 세부분으로 나누고 계획 측면에서의 장비배분과 운영 측면에서의 장비 배차로 단계화시켜 접근하여야 한다.

기존 터미널과 비교하여 정보시스템의 역할이 보다 강화되고 작업자의 개입 없이 이루어지는 의사결정이 증가된다. 이러한 특성을 반영할 수 있는 의사결정 구조로서 분산형 구조를 제안한다. 자동화 컨테이너 터미널에서 선석할당과 게이트할당은 기존 터미널과 비교하였을 때, 차이점이 크지 않다. 선석할당은 터미널에 입항하는 선박들의 접안시간과 접안위치를 결정하고 본선하역 작업을 수행하는 Q/C의 수를 결정하는 문제이다.

각 선박들이 최대한 안벽상의 선호위치에 접안할 수 있게 하며 각 선박들이 원하는 입항시간과 출항시간을 최대한으로 만족시키는 선박의 접안시점을 결정한다. Q/C 할당과 함께 결정하는 것이 바람직하다. 선석할당의 전략은 재래식터미널의 경우와 유사하고, 이는 하역작업의 전면적인 자동화가 아직 힘들고 터미널의 통제범위를 벗어나는 다양한 상황과 요구조건이 발생하기 때문이다.

게이트 할당은 운송 트럭이 터미널의 반·출입 게이트에 도착했을 때 진입 레인을 결정하는 문제이다. 작업 종류, EDI 사용 유무, 그리고 전용레인 운영 여부 등의 정보를 이용하여 각 트럭이 진입할 레인 위치를 할당한다.

## **(2) 야드계획 엔진**

장치장의 상황을 고려하여 컨테이너의 저장위치를 정하는 의사결정으로 장치장 적재 상태뿐만 아니라 취급장비의 이동 및 작업영역

에 대한 작업부하의 배분과도 밀접한 관계가 있다.

장치능력하에서 장비의 작업능력이 최대화될 수 있게 작업부하를 배분하는 장치장 운영은 터미널의 생산성을 최대화하는 과정이다.

장치장 운영의 의사결정을 계획단계의 장치공간 계획과 운영단계의 장치위치 할당의 계층적 구조로 구성되어야 하며, 저장공간 계획은 개별 컨테이너가 도착하기 이전에 선박별 작업물량과 예정작업 시각을 고려하여 사전에 장치공간을 배정하는 단계이고 장치위치 할당은 컨테이너가 도착하는 시점에 특성과 장치장의 현재 상황을 고려하여 계획된 저장공간내에서 장치위치를 실시간으로 결정하는 단계이다.

하역장비의 설계능력, 이송장비의 주행 여건, 컨테이너의 흐름 등과 이에 따른 생산성의 차이를 고려하여 장치장이 수평배치인 경우와 수직배치인 경우에 대해서 각각 장치장 운영전략을 구분하여 적용되어야 한다.

장치장 운영의 목적은 작업시간의 단축, 장치장 활용도의 증대, 작업부하의 평준화이다.

컨테이너 각각에 대해서 구체적인 장치위치를 결정하지는 않고 필요한 공간만큼 장치장의 저장능력을 배분하는 의사결정 장치위치를 결정할 때 초기 탐색범위가 될 뿐이지 각 위치에 대한 사용순서나 정확한 장치물량을 결정하지는 않는다.

선박별 장치공간할당은 선석할당의 내용과 관련된다. 선박의 접안위치는 장치공간의 영역에 관계되고 할당 Q/C 수는 본선 하역작업이 ATC의 작업과 연동되어 수행되므로 장치공간이 어느정도 확보되고 분산화되어야 하는지를 결정할 때의 고려요소이다.

미래의 불확실한 상황과 예측자료의 변동으로 인해 의사결정의 자동화가 힘들며 수시로 재계획·수정·보완하는 과정이 필요하다. 따라서 수리적 모델들의 단점을 고려하여 현장경험을 중심으로 알고

리즘을 적용하여야 한다.

### **(3) 양적하계획 엔진**

인공지능 기법을 이용하여 자동적으로 크레인 작업순서 결정, 양하 작업순서 결정, 적하 작업 순서 결정, 이선적 계획 등을 수행하는 기능과 선박의 안정성을 검증하는 안정성 시뮬레이션(stability simulation) 기능을 포함한다. 현재 GA(Genetic Algorithm) 및 전문가 시스템을 이용한 자동 양적하계획 응용프로그램들이 개발되어 있다. 현재는 타 응용에이전트와 상호협동할 수 없는 형태이다. 이를 보완하여 하역중 돌발적으로 발생할 수 있는 문제해결을 할 수 있도록 양적하 계획 응용에이전트가 타 에이전트와 상호 협력하여 자동적으로 계획을 변경하여 작업처리할 수 있도록 하여야 한다.

### **(4) 문제해결, 모니터링 및 작업 지시 제어**

관제부문은 실시간으로 터미널내에서 발생하는 모든 이벤트를 모니터링하는 것으로 게이트, 장치장, 선석 등 터미널내에서 발생하는 모든 컨테이너의 이동을 추적하고, 모든 하역 장비의 상태 및 작업 상황을 추적하는 것이다. 작업 지시 및 통제 부문은 장치장 통제, 안벽 통제, 게이트 통제 등으로 구성되며, 최적화 및 자동화 알고리즘을 통하여 가양(rehandling) 작업과 장비 이동 거리를 최소화할 수 있도록 하역 장비를 할당하는 기능을 담당한다. 문제해결 기능은 사전에 모형화된 문제 형태에 근거하여 실시간으로 문제를 파악하거나 시뮬레이션 등을 통하여 문제를 파악하고, 상황에 적절한 계획을 수립하고 이에 따라 작업지시를 내리게 된다. 통합경보기능은 실시간으로 문제 및 사전에 설정된 계층을 따라 사용자에게 여러가지 형태로 경보를 발한다.

### **(5) 장비할당 배차**

ATC 배분을 위해 작업부하를 산출하는 방법은 작업특성을 고려하여 본선하역작업과 반출입작업으로 구분한다. 장치장 배치형태별로 ATC의 운영방법이나 고려요소가 다르므로 각각에 대한 운영전략을 제안한다.

수평배치의 경우는 ATC의 작업영역에 대한 결정이 같이 이루어져야 한다. ATC의 움직임은 최소화 하는 것을 목적으로 하기 때문에 동일한 ATC가 동일한 Q/C를 집중적으로 서비스할 가능성이 많다. 수직배치에서는 ATC의 작업영역이 고정적이므로 작업부하의 균형화 측면이 보다 중요하다. 따라서 장비배분의 문제보다는 장비배차의 문제가 효율성에 영향을 많이 미친다.

AGV는 고정배차로 사용하느냐 아니면 공유배차로 사용하느냐에 따라 다른 의사결정 규칙이 필요하다. 공유배차를 사용하는 경우에는 몇 대의 Q/C를 범위로 설정할 것인지가 중요하다. 고정배차의 경우에는 작업조건에 따라 차별화할 수 있다.

ATC 배차(dispatching)란 특정 장비에 구체적인 작업을 할당하는 것을 말한다. 장비의 활용도를 극대화하고 작업의 지연이나 대기시간을 최소화하는 것이 배차의 목적이다. AGV 배차는 중앙집중형 AGV 배차전략은 효율적인 의사결정을 할 수 있는 반면 과정이 복잡한 단점이 있다. 따라서 분산형 AGV 배차에 대한 접근방법이 효율적이다.

### **(6) 장치위치 결정**

실제 장치계획을 적용할 경우에는 불확실한 변동 요인들이 많아 수시로 재계획을 통해 조정한다.

ATC, AGV 그리고 외부트럭의 상호 연계작업을 전제로 장치 위치가 할당되므로 장비의 작업할당과 관련이 크고 의사결정 시점에



는 이들의 상황을 충분히 고려하는 것이 필요하다.

장치계획에서처럼 장치장의 배치형태나 컨테이너의 종류에 따라 고려요소가 달라지므로 먼저 배치형태에 따라 수평인 경우와 수직인 경우를 구분하고 이를 다시 수출컨테이너와 수입컨테이너로 나누어 각각에 대한 장치위치 전략을 적용하여야 한다.

의사결정 문제는 두 가지가 있는데, 하나는 장치베이를 결정하는 것이고, 다른 하나는 베이내에서의 적재위치를 결정하는 것이다.

### 5.2.2 XML/JMS 기반의 멀티에이전트간 통신 모델 제안

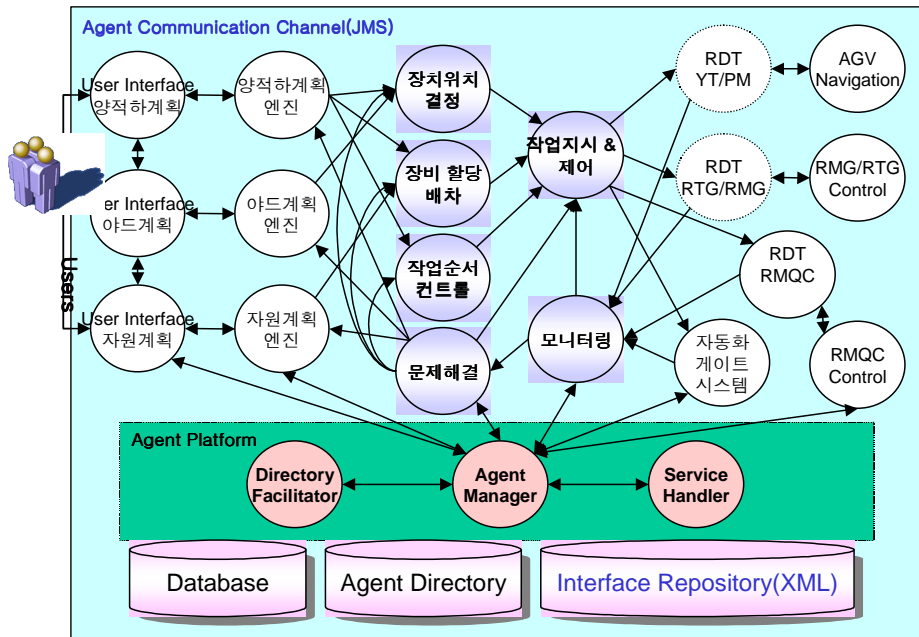
자동화 운영시스템 핵심 에이전트는 터미널 운영에 있어서 상시 서비스(non-stop service)를 제공하여야 하며, 에이전트간 실시간 협력력이 필요한 것들을 의미한다. 무인 자동화 터미널로 발전될수록 에이전트의 기능이 다양하며, 에이전트간 통신이 복잡해진다. 또한, 단일 업체에서 모든 에이전트를 공급할 수 없으므로, 운영시스템 개발을 위한 업체가 다양하게 된다. 그러므로 에이전트간 통신 및 인터페이스의 표준이 꼭 필요하다.

이에 자동화 터미널 운영시스템 구축을 위한 멀티에이전트간 통신 모델을 <그림 5-4>와 같이 XML/JMS 기반의 모델로 제시한다.

멀티에이전트의 플랫폼으로는 FIPA의 AP모형을 기반으로 구성하였다. 즉, 개별 응용에이전트의 관리는 중앙집중적인 AP에 의해서 이루어진다. AP는 플랫폼에 소속된 에이전트에 대해 에이전트 레벨의 서비스, 즉 에이전트 등록, 에이전트 이름과 주소 매핑(agent name resolution), 에이전트 통신, 서비스 찾기(service location), 에이전트 관리 등을 제공한다.

Agent Manager는 시스템 내의 에이전트들의 작동 및 정지와 같은 에이전트 라이프 사이클을 관리한다. Directory Facilitator는 일

종의 "Yellow Page" 제공자의 역할을 하여, 시스템 내의 모든 구성 에이전트들이 가지고 있는 서비스의 리스트에 대한 정보를 Agent Directory에 저장하고, 이를 제공한다. Service Handler는 생성된 계획규칙에 따라 작업흐름을 제어하며 에이전트들 간의 인터페이스 역할을 한다. Interface Repository는 에이전트 시스템에 필요한 규칙과 역할 정보를 XML 형태로 관리한다. 이 시스템의 조정 작업을 위해 작업 규칙을 계층적으로 구성하는데, 먼저 계획규칙을 통해 시스템에서 제공할 서비스의 기본 흐름을 정의하고 Service Handler로 하여금 이 계획규칙을 접근하게 한다. 보다 세부적인 작업 순서는 그룹규칙으로 정의하여 그 그룹 내의 에이전트들의 흐름을 제어한다. 그룹 내의 하위 작업 에이전트들의 능력은 작업규칙으로 정의한다. 에이전트간 통신을 위한 Agent Communication Channel은 OS 환경에 독립적인 자바 기반의 메시징시스템인 JMS로 한다.



<그림 5-4> XML/JMS 기반 Multi-Agent 통신 모델

자동화 컨테이너 터미널의 설계 사상은 다음과 같다.

에이전트 혹은 모듈간 인터페이스를 위한 표준 에이전트 프레임워크(에이전트 인터페이스 프로세스 스펙 및 교환 정보 스펙)가 있어야 하며, 통합공유 및 간편하게 개선할 수 있어야 한다.

에이전트는 ATC, AGV와 같은 이동장비에 탑재할 수 있는 간편한/간단한 통신방법이어야 한다.

특정 플랫폼에 독립적인 언어/기술이어야 한다.

적용될 기술은 국제/산업계 표준이어야 한다.

## 5.2 XML/JMS 기반의 통신모델의 장점

자동화터미널의 운영시스템은 멀티에이전트간의 상호 협동하여 목표를 달성하는 분산구조로 구성되어야 한다고 제안한다. 또한 단일 업체에서 모든 에이전트를 구현 공급할 수 없으므로, 다양한 업체의 상용 에이전트간 인터페이스와 통합 구축을 위해서는 표준화된 JMS 기반의 XML 통신모델을 제안한다. 표준화된 통신모델을 기반으로 급변하는 자동화터미널 운영시스템의 요구사항을 충족하는 다양한 상용에이전트를 추가/확장/교체/통합할 수 있을 것으로 기대한다.

## 제 6 장 결론

본 논문에서는 현재 국내외적으로 활발하게 논의되고 있는 컨테이너 터미널 자동화에 대하여 살펴보고, 자동화 터미널 시스템 모델에 대하여 제시하였다. 새로운 자동화 터미널 시스템 모델은 급변하는 항만 환경 변화에 부응하며, 기존 시장의 자동화 기술을 통합할 수 있는 표준 기반을 제시하였다.

본 논문에서 제시하는 자동화 컨테이너 터미널 시스템의 모델에 대하여 살펴보면 다음과 같다.

첫째, 컨테이너 터미널의 운영을 위해 필요한 모듈을 상호 협동하여 목적을 달성할 수 있는 에이전트화 하는 것이다. 현재 터미널 자동화를 위하여 개발된 자동화 모듈의 기능과 개발 동향 및 타당성에 대해서 알아보았다.

둘째, 에이전트간 협동 및 지식 공유를 위한 멀티에이전트에 대한 연구들을 살펴 보았고, 자동화 터미널의 멀티에이전트 기반 시스템으로 FIPA의 AP 모델을 적용하였다. 또한 에이전트간의 통신을 위하여 통신언어에 대하여 살펴 보았으며, KQML과 XML중 개발 환경에 유연한 XML을 적용하였다. 통신 프로토콜로서는 안정적인 메시징 시스템이 가장 적합하며, 산업계 여러 표준기술 중 플랫폼에 독립적인 JMS를 적용하였다.

마지막으로 자동화 터미널 운영시스템을 위한 기능과 요소기술을 통합구축을 위한 전체적인 통합시스템의 모델을 제시하였다.

향후 XML/JMS 기반의 멀티에이전트 시스템 구축을 위한 인터페이스 모델을 기반으로 자동화 터미널의 에이전트간 표준 통신을 위한 가이드라인과 시스템 개발을 위한 프레임워크의 프로토타입을 구현하는 것이 필요하다.

## 참 고 문 헌

- [1] 최장림, “자동화 컨테이너터미널 통합운영시스템의 개념적 설계”, 한국해양대학교 석사학위논문, 2000년 8월
- [2] 김우선, “컨테이너 터미널 운영 개선을 위한 시뮬레이션 모형 설계 및 개발”, 한국해양대학교 석사학위논문, 1999년 2월
- [3] 양창호, 김영훈, 최상희, 배종욱, 이종은, “컨테이너 터미널 자동화의 작업과 시스템 디자인에 대한 연구”, 한국해양수산개발원, 2000년 12월
- [4] 박영택(숭실대), 최중민(한양대), 이근배(포항공대), “에이전트 속성 및 설계에 관한 연구”, 과학기술부, 1999년10월
- [5] 박승수, “대규모 지식기반 에이전트를 위한 소프트웨어론 시스템의 연구”, 과학기술부, 1999년 10월
- [6] W3C Extensible Markup Language(XML) 1.0  
[http://www.w3.org/TR/2000/REC-xml-20001006\(second edition\)](http://www.w3.org/TR/2000/REC-xml-20001006(second%20edition))
- [7] W3C, “XML 1.1” <http://www.w3.org/TR/xml11/>
- [8] W3C HTML, <http://www.w3.org/MarkUp/#historical>
- [9] A Beginner’s Guide to HTML <http://archive.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html>
- [10] 이강찬, 손홍, 박기식, “XML 표준화 동향”, 정보과학회지 제 19권, 제 1 호, pp. 6-14, 2001
- [11] Alex Homer, David Sussman, “MTS MSMQ”, WROX, 1998
- [12] Richard Monson Haepel, David Scharpel, 김기현(역) “Java Message Service”, 한빛미디어, 1999
- [13] Subrahmanyam Allamaraju , 신정환 , 이대회, “Professional Java Server Programming J2EE 1.3 Edition”, 정보문화사,

2002

- [14] SUN "웹서비스 구축을 위한 기본으로서의 J2EE와 .NET", Sun newsletter, pp.13-23, January 2002
- [15] Brian Goetz, "IBM developerworks Java theory" <http://www-106.ibm.com/developerworks/java/library/j-jtp0410/index.html?dwzone=java>
- [16] 김종렬, "국내 자동화 컨테이너 터미널의 개발 방향에 관한 연구", 한국해양대학교 석사학위논문, 2001년 2월
- [17] 유선영, "XML 기반의 이기종 DBMS간 데이터 복제 웹 에이전트 설계 및 구현", 한국해양대학교 석사학위논문, 2002년 2월
- [18] (주)토탈소프트뱅크, "유전알고리즘을 이용한 최적화 자동 컨테이너 위치결정 및 적하 계획 시스템 개발 최종보고서", 한국해양대학교 산업기술연구소, 2002년 9월
- [19] 첨단항만핵심기술개발사업단, "첨단항만핵심기술개발 최종보고서", 2003년 9월
- [20] (주)토탈소프트뱅크, "동적 AGV 컨테이너 작업할당과 주행 알고리즘 개발 최종보고서", 한국해양대학교 산업기술연구소, 2003년 10월
- [21] 한국컨테이너부두공단, "광양항 3단계 2차 자동화 컨테이너터미널 정보전략계획수립 최종보고서 발표자료", 2003년 3월