



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

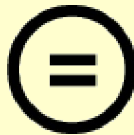
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

기술경영학석사 학위논문

스마트공장에서의 시스템 통합을 위한 객체홀론의 경량적 표현 및 운영방법



2018년 2월

부경대학교 기술경영전문대학원

기술경영학과

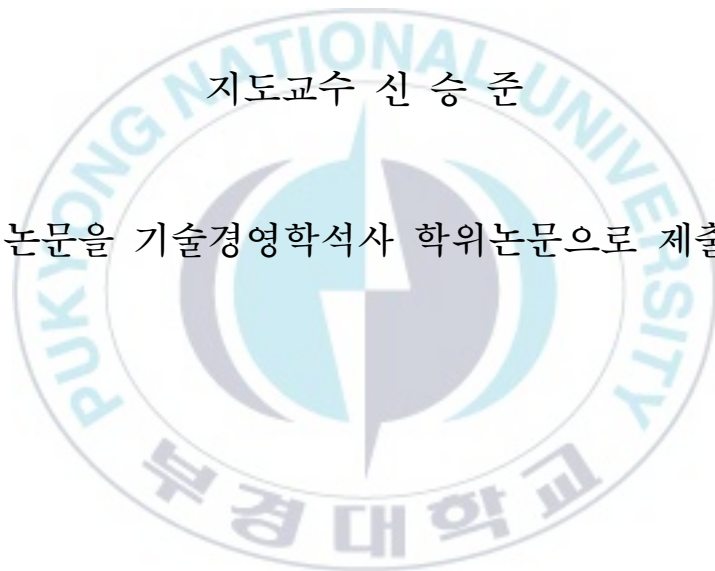
정 여 진

기술경영학석사 학위논문

스마트공장에서의 시스템 통합을 위한 객체홀론의 경량적 표현 및 운영방법

지도교수 신 승 준

이 논문을 기술경영학석사 학위논문으로 제출함.



2018년 2월

부경대학교 기술경영전문대학원

기술경영학과

정 여 진

정여진의 기술경영학석사 학위논문을
인준함.

2018년 2월 23일



위 원 장 공학박사 서 원 철



위 원 공학박사 옥 영 석



위 원 공학박사 신 승 준



Lightweight Representation and Operation of Holonized Objects for Systems
Integration on Smart Factory

Yeo-Jin Jeong

Graduate School of Management of Technology, Pukyong National University

Abstract

As the competitiveness of manufacturing increasingly becomes significant to cope with the era of the fourth industrial revolution, a lot of researches have been strived to develop and deploy Smart Factory. For the pursuit of Smart Factory which ensures flexibility and efficiency in manufacturing, it is vital to develop Cyber-Physical Production Systems (CPPS) that embody bi-directional mirroring between physical and cyber manufacturing objects. CPPS are recognized as the advanced model of Smart Factory in that they enable manufacturing objects to perform autonomous and collaborative decision-making. In CPPS, the information exchange in-between the objects is a fundamental requirement; however, it shows a limitation due to various and heterogeneous representations about the objects. This thesis presents the design of a structural and unique representation method that provides manufacturing objects with an open and seamless information exchange environment using Java Script Object Notation (JSON), which is a light-weight data interchange format. For such the purpose, this thesis includes the design of manufacturing objects' data, workflow and message models under the paradigm of Holonic Manufacturing Systems. It also includes a prototype implementation to demonstrate the feasibility and validity of the proposed models. The deliverable of the present work is expected to facilitate the systems integration in heterogeneous smart factory systems through gaining simplicity, commodity, openness and interoperability on manufacturing objects' representation.

목 차

제 1장 서론	1
1.1 연구 배경	1
1.2 연구 필요성	5
1.3 연구 목적	10
제 2장 선행연구	12
2.1 첨단제조시스템	12
2.1.1 스마트공장	12
2.1.2 사이버-물리 제조시스템	17
2.1.3 홀로닉 제조 시스템	20
2.2 에이전트 시스템	23
2.2.1 JADE	23
2.2.2 FIPA	25
2.3 데이터표현	27
2.3.1 XML	27
2.3.2 JSON	28
제 3장 시스템 설계	33
3.1 개념구조	33
3.2 행동 모델	35
3.3 데이터 모델	37
3.4 메시지 모델	40
제 4장 시스템 구현	43

4.1 구현시나리오	43
4.2 구현결과	44
제 5장 결론	48
5.1 연구 요약	48
5.2 연구 공헌	50
5.3 한계점 및 극복방안	51
참고 문헌	53
부 록	63
부록1. Factory 소스코드	63
부록2. Machine 소스코드	66
부록3. Factory Management System 소스코드	68



그림 목차

[그림 1-1] 주요국 산업별 부가가치 비중	1
[그림 1-2] 제조업 부가가치 비중 추이	1
[그림 1-3] GDP 대비 제조업 비중	2
[그림 1-4] 주요 산업의 전년동월대비 취업자 수 변동추이	3
[그림 1-5] 미국기업의 데이터 보유현황	9
[그림 2-1] CPS 구현을 위한 5C 아키텍처	18
[그림 2-2] CPS기반 분산서비스	19
[그림 2-3] JADE 아키텍처	24
[그림 2-4] FIPA 97 에이전트 참조모델	26
[그림 2-5] 프로그래밍 가능한 웹 디렉토리 XML과 JSON의 비율 ...	29
[그림 3-1] 홀로닉 제조 시스템의 기본 개념	33
[그림 3-2] JSON기반 시스템 통합 구조	35
[그림 3-3] 행동 모델	36
[그림 3-4] 데이터 모델	37
[그림 3-5] 메시지 모델	40
[그림 3-6] FIPA기반 메시지	41
[그림 3-7] JSON기반 메시지	41

표 목차

[표 1-1] 주요국 제조업 경쟁력 순위	5
[표 1-2] ‘중국제조 2025’ 주요 계획 지표	6
[표 2-1] 세계 주요국의 4차 산업혁명 도입 개념	13
[표 2-2] 주요국의 제조업 ICT융합 정책	14
[표 2-3] 스마트공장 구축기업 현황	16
[표 2-4] JSON, XML 데이터 전송속도	31
[표 2-5] JSON, XML의 CPU 사용량	31
[표 2-6] JSON vs XML	32
[표 3-1] FIPA, JSON기반 메시지 크기 비교	42

I. 서론

1.1 연구 배경

경제적 측면에서 제조업이 가장 중요한 분야임에도 불구하고(이상일 외, 2013) 한국은 제조업이 국내총생산(GDP)에서 차지하는 부가가치 비중이 30%대에 도달한 뒤 침체로 돌아서는 그림 1-3과 같은 한국의 ‘제조업 30% 법칙’¹⁾이 나타난다.¹⁾ 그에 반해 서비스업은 60% 정도로 서비스업이 2배가량 많다(강서진, 2014). 세계 주요 선진국들의 GDP는 한국에 비해 제조업의 부가가치 비중이 상대적으로 낮은 반면 서비스업 비중이 높게 형성되는 것이 특징이다. 제조업 부가가치 비중은 1970년 이후 지속적으로 감소하여 서비스업 비중이 제조업의 3,4배로 높게 형성 되었다(강서진, 2014).

그림 1-1 주요국 산업별 부가가치 비중

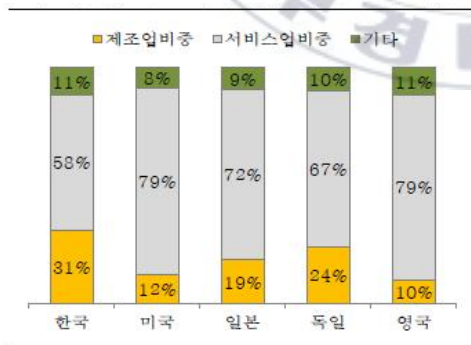
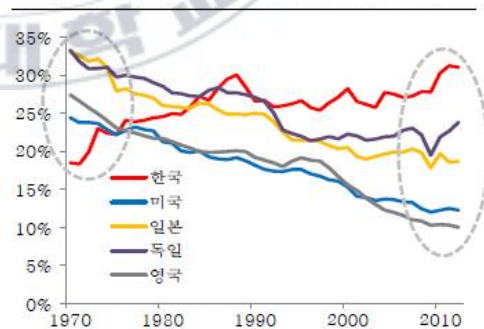


그림 1-2 제조업 부가가치 비중 추이

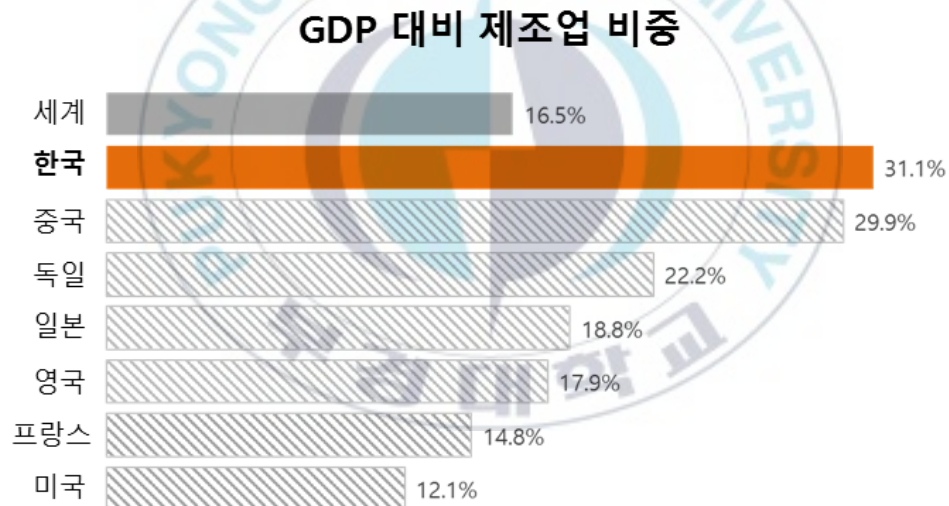


(강서진, 2014 참고문헌 자료; IMF)

1) 「‘제조업 30% 법칙’ 발목 잡힌 한국제조업 ‘퍼스트 무버 전략’으로 극복해야」, 『아주경제』, 2015년 4월 29일, <http://www.ajunews.com/view/20150429141527536>

한국에서 제조업을 중시하는 이유는 한국경제가 가지고 있는 고유한 특성에 있다. 첫 번째 특성은 한국은 자원빈국이기 때문에 원유와 같은 에너지 자원 수입은 전체 수입액의 4분의 1정도를 차지하고 있어 불가피하다. 이때문에 제조업의 활성화는 수출로 이어지고 외화획득의 가장 빠른 방법이다. 두 번째 특성은 한국의 협소한 내수시장에 있다. 협소한 내수시장의 한계 극복을 위해 제조업 위주의 수출을 통해 성장을 도모한다(강서진, 2014).

그림 1-3 GDP 대비 제조업 비중

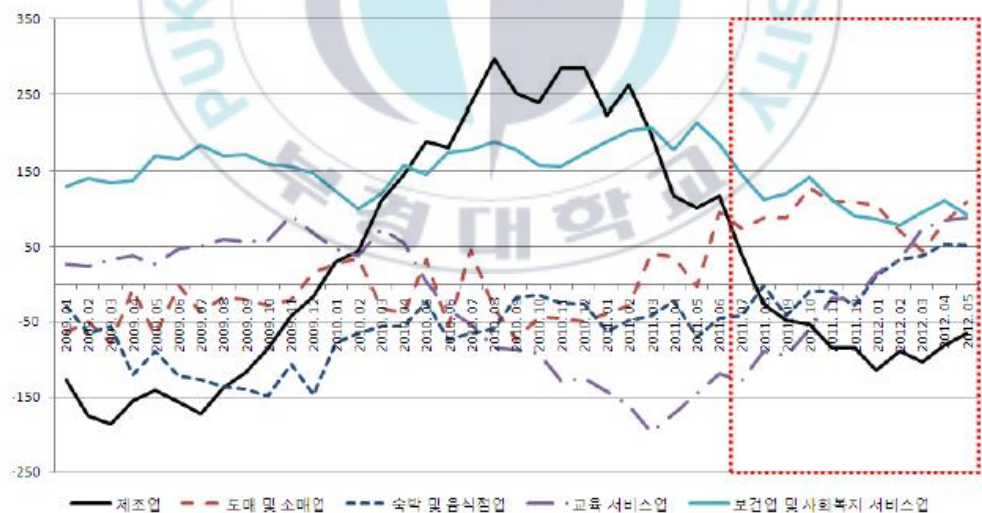


선진국의 경우는 한국과의 차이가 있다. 제조 선진국에 속하는 미국, 일본, 독일은 내수시장의 규모가 큰 대규모 경제라고 할 수 있다. 선진국의 제조업 부흥의 움직임이 최근 두드러진 첫 번째 이유로 2000년대 들어 아시아 신흥공업국들이 세계 제조업 시장을 빠르게 잠식해 오면서 크지 않은 제조업 비중에도 고용된 인력과 산업의 어려움으로 제조업의 몰락을 방지하

면 안 된다는 위기감 증대이다. 두 번째는 2008년 세계 금융위기 이후 제조업에 기반을 두지 않고 금융산업에 많이 의존 했던 미국, 일본, 영국 등의 국가가 상당한 어려움을 겪었다. 반면 제조업의 경쟁력이 높았고 중시했던 독일, 중국 등의 국가는 상대적으로 타격을 받지 않았던 면이 부각되면서 제조업에 다시금 관심을 가지게 되었다¹⁾.

제조업의 일자리 창출효과는 서비스업에 비해 큰 도움이 되지 못한다. 금융위기 이후 고용률 상승세를 이끈 것이 제조업에서의 취업자 수 증가였지만 2011년 하반기부터 제조업이 아닌 서비스산업에서의 취업자 수가 증가했다. 그림 1-4는 2011년 하반기 이후 전년동월대비 제조업 취업자 수 감소를 보여주고 있다.

그림 1-4 주요 산업의 전년동월대비 취업자 수 변동 추이 (단위:천명)



(김수현, 2017 참고문헌 자료; 통계청)

제조업의 공백을 메운 것은 도매 및 소매업과 숙박 및 음식점업과 같은 전통적 서비스산업으로 이는 고용의 질적 증가보다 양적 증가의 모습을 보

여주고 있다. 상대적으로 임금이 높고, 고용의 질적 수준이 높은 것으로 평가되는 제조업 일자리의 취업자는 감소하고, 임금이 낮고 비정규직의 비중이 큰 서비스산업의 취업자가 증가하고 있는 현실이다(김수현, 2012).

질적으로 우수한 고용을 늘려 청년실업의 현실을 줄일 필요가 있다. 2017년 현재 4차 산업혁명으로 제조업의 필요성과 중요성을 강조하면서 제조업 취업자가 첫 증가세를 보였지만 내수 부진으로 숙박, 음식업 고용감소 전환과 도소매 고용 둔화로 서비스업 취업의 감소로 여전히 청년 실업률은 최고치를 기록했다²⁾.

청년실업의 문제는 제조업만의 문제가 아닌 제조업과 서비스업의 균형 성장을 추구함으로써 안정적인 경제 성장의 토대가 마련된다. 현재 인더스트리 4.0으로 인해 전 세계적으로 제조업의 중요성과 제조업 부흥에 힘쓰고 있다. 선진국의 제조업 부가가치 비중은 1970년 이후 지속적으로 감소하여 서비스업 부가가치 비중이 제조업의 3, 4배 수준으로 높게 형성되어 있어 제조업은 20%대의 비중을 차지하고 있다¹⁾. 경제적 측면에서 제조업이 중요하나 고용 측면에서는 서비스업에 미치지 못하는 것이 사실이다(강서진, 2014).

2) 「제조업 취업자 1년 만에 첫 증가했지만, 청년 실업률 최고치」, 『조선비즈』, 2017년 7월 12일,
http://biz.chosun.com/site/data/html_dir/2017/07/12/2017071201021.html

1.2 연구 필요성

한국의 제조업은 지난 40년간 고용·혁신 창출의 원천 역할을 수행해 왔으나, 급속한 대내·외 환경변화에 따라 국내 제조업의 국제경쟁력 강화를 위한 새로운 발전 전략을 수립할 시점이 되었다. 국내 제조업은 기술경쟁력보다 가격경쟁력에 기반을 두고 있어서 경쟁 심화 및 외부 경제 변화에 취약하여 이는 국내 제조업의 위기로 다가오고 있으며, 에너지·환경 문제, 고령화에 따른 일자리 문제, 작업환경 문제 등 대내·외 환경변화에도 직면해 있어, 이를 극복하기 위한 능동적인 대응 방안이 필요하다(김선재, 2017).

표 1-1 주요국 제조업 경쟁력 순위(김선재, 2017)

순위	2010	2013	2018(예상)
1	중국	중국	중국
2	인도	독일	인도
3	한국	미국	브라질
4	미국	인도	독일
5	브라질	한국	미국
6	일본	대만	한국

표 1-1은 주요국의 제조업 경쟁력 순위를 보여주는 것이다. 한국은 fast-follower 전략으로 제조 강국 반열에 올랐으나, 선진국의 제조업 부흥 정책 및 신흥국의 산업구조 고도화로 한계에 직면 하였고, 제조업 가동률 하락(73.5%), 출하 증가율 감소(-0.5%) 등 전반적 지표 하락과 더불어 글로벌 경쟁력은 '18년까지 6위로 하락될 것으로 예상된다(김선재, 2017).

현재 세계 각국은 침체된 제조업시장의 활성화를 위한 정책을 추진 중이

다(이용구, 2017). 그 중 하나가 스마트공장으로 4차 산업혁명 시대가 도래하면서 독일의 인더스트리 4.0으로 시작해 미국과 중국, 일본, EU 등 제조강국들은 제조업 부흥을 통한 제조업 경쟁력을 강화하고 있다. 미국은 첨단제조 능력 강화를 통한 제조업의 국가경쟁력 부활을 목표로 민·관·학·연 프로젝트를 추진 중이고, 중국은 향후 30년을 3단계로 나누어 산업구조를 고도화시키는 중국제조 2025 전략을 추진 중이다(강지원, 2015).

표 1-2 ‘중국제조 2025’ 주요 계획 지표(강지원, 2015)

분류	지표	2013년	2015년	2020년	2025년
혁신 역량	규모이상 제조업 매출액 대비 R&D 지출 비중(%)	0.88	0.95	1.26	1.63
	규모이상 제조업체 매출 1억 위안 당 발명 특허 수(건)	0.36	0.44	0.7	1.1
질적 성과	제조업 품질경쟁력 지수	83.1	83.5	84.5	85.5
	제조업 부가가치 증가율 제고	-	-	2015년 대비 2.0%p 증가	2015년 대비 4.0%p 증가
	제조업 노동생산성 증가율(%)	-	-	7.5	6.5
IT 제조업 융합	인터넷 보급률(%)	37	50	70	82
	디지털 R&D 설계 도구 보급률 (%)	52	58	72	84
	핵심공정 CNC 비중(%)	27	33	50	64

친환경 성장	규모이상 기업의 산업 생산량 단위당 에너지 소모 감축 비율(%)	-	-	2015년 대비 18% 감축	2015년 대비 34% 감축
	산업생산량 단위 당 이산화탄소 배출 감축 비율(%)	-	-	2015년 대비 22% 감축	2015년 대비 40% 감축
	산업생산량 단위 당 수자원 사용 감축 비율(%)	-	-	2015년 대비 23% 감축	2015년 대비 41% 감축
	공업용 고체폐기물 사용률(%)	62	65	73	79

EU는 스마트공장에 대한 통합적인 프로그램 및 개발 계획을 마련하고, 각국에서 주요 기술개발에 주력 중이며, 독일은 인터스트리 4.0이 포함된 미래기술 비전을 수립하고 2.5억 유로 규모의 국가 프로그램을 운영하고 있다(이현정 외, 2017).

이와 같은 추세에 따라, 스마트공장의 보급 확대 및 역량 강화를 위해 2017년 산업통상자원부에서는 ‘스마트 제조혁신 비전 2025’를 발표해 2025년까지 스마트공장 3만개 구축을 추진하는 정책을 발표하였다. 급변하는 소비자의 수요에 신속히 대응하기 위해서는 스마트공장 구축을 통해 개인 맞춤형 유연생산 체제로의 전환이 필수적이다(배경한 외, 2017).

스마트공장은 ICT(Information and Communication Technology)와 기존의 제조업 기술인 생산제조기술이 융합하여 사물인터넷, 빅데이터, 클라우드 컴퓨팅, CPS(Cyber-Physical System) 등의 기술을 통해서 공장 내의 장비, 장치 부품들이 서로 연결되고 상호 소통되는 생산 체계이다(이용구, 2017).

제조업의 스마트화는 ICT 융합을 기반으로 첫째, 제품, 설비, 공정이 지능화되어 네트워크로 연결되고, 둘째, 모든 데이터, 정보가 실시간으로 관리, 활용되어 셋째, 기획, 설계, 생산, 판매, 서비스 등의 전 과정이 최적으로 효율화되고, 넷째, 개인 맞춤형 제도가 가능한 다섯째, 협업적 생산체계가 구축, 운영되는 제조업 고부가가치화로 정의할 수 있다(이용구, 2017).

현재 제조 데이터와 의사결정 모델의 표준화를 많이 이루어져 있다. 제조 표준에 관한 연구로, MTConnect 기반의 이기종 기계 및 셀 간 확장형 인터페이스를 구현하여 양방향 통신을 통해 실시간 생산현장 정보를 연동, 관리할 수 있도록 하는 연구를 하였고(Yoon et al., 2013), PLM 비즈니스 프로세스 내의 정보와 데이터 표현이 가능한 OMG 표준인 PLM Services를 기반으로 중립 스키마를 정의해 상용 PDM 시스템을 포함한 이기종 시스템간의 제품정보 교환을 위한 기반 구현하였다(양태호 외, 2008). 이기종 기계, 셀 등이 혼재한 공장의 경우 제조사별 상이한 통신방식이 복잡하게 얽혀있기 때문에 표준에 의한 통합적인 데이터 수집이 매우 어렵다. 이러한 문제를 해결하기 위해 이기종 기계, 셀의 통합적인 데이터 수집 및 모니터링을 위해 생산설비와의 표준 통신규약 중 하나인 MTConnect를 채택하여 기계, 셀의 계층적 구성이 XML로 표현되도록 시스템을 개발하였다(Nam et al., 2013).

스마트공장이나 제조데이터의 표준화는 많이 이루어져 연구가 진행되고 있으나 제조 객체 자체에 대한 이기종 웹 환경에서의 데이터 교환 방식의 연구는 미약하여 연구가 필요한 부분이다.

스마트공장에서 불가피한 데이터 확보를 위해선 스마트 센서가 필요하다. 제작된 스마트 센서는 이기종 간 높은 상호 운용성을 제공하는 OPC UA 통신 미들웨어는 탑재하고 있어 스마트공장 관리 시스템에 쉽게 적용할 수 있다(이성준 외, 2017).

일반적으로 빅데이터의 특징을 규모(Volume), 다양성(Variety), 속도(Velocity)를 3V라 칭한다. 규모가 적게는 수 테라바이트, 많게는 수 페타바이트 정도의 규모이며, 다양성은 정리되고 구조화된 데이터뿐만 아니라 포털사이트의 뉴스나 자료, 소셜 미디어의 정해지지 않은 텍스트 정보, 음원이나 영상 같이 비정형 데이터도 상당한 규모에 이른다. 속도는 데이터를 일괄적으로 받아 분석하는 것이 아닌 실시간으로 유입되는 데이터를 분석한다는 의미이다(Manyika et al., 2011).

그림 1-5 미국기업의 데이터 보유현황(Manyika et al., 2011)

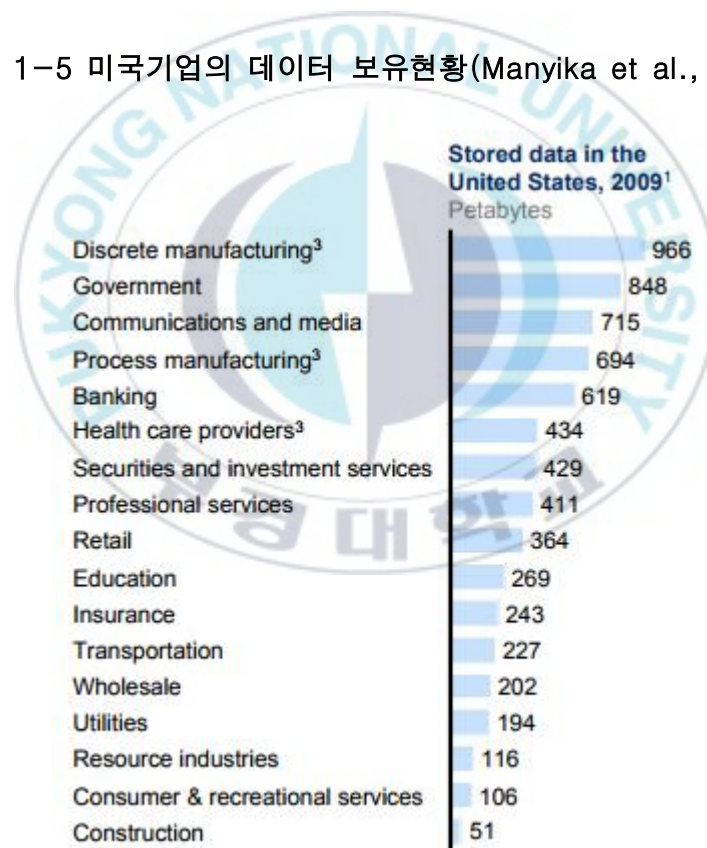


그림 1-5는 2009년 미국 내 기업들이 보유한 데이터의 양을 산업별로 나타낸 그래프이다. Discrete manufacturing의 경우 966페타바이트, Process manufacturing의 경우 694페타바이트로 17개 비교 산업군들 중 첫번째, 네번째로 많은 데이터를 보유하고 있음을 알 수 있다(Manyika et al., 2011). 이처럼 데이터 생성에서 제조업은 다른 산업과 비교할 수 없을 정도로 상당한 양의 데이터를 보유하고 있는 산업이다(장영재, 2012).

1.3 연구 목적

4차 산업혁명시대에 도래하면서 제조업기반의 산업 발전을 중요시하고 있는 추세이다. 독일은 2011년 4차 산업혁명에 대비해 인더스트리 4.0을 제시하였고, 이는 2013년 4차 산업혁명의 명칭과 개념이 최초로 대중에게 공개된 하노버 박람회를 이어져 일부에서는 4차 산업혁명을 인더스트리 4.0과 동일시하기도 한다(전은경, 2016).

인더스트리 4.0은 사물인터넷, 빅데이터, 3D 프린팅 등 그 동안 발전해온 IT와 물리분야 기술을 모두 활용해 생산방식을 전면적으로 재편하는 사이버 물리시스템을 구축하고자 하는 정책이다. 인더스트리 4.0의 최종목표는 스마트공장의 활성화로 공장 내부뿐만 아니라 전후방 관련기업들을 모두 포함하여 가치사슬에 참여한 모든 기업들과도 정보를 교환하여 최적의 생산을 지향한다(전은경, 2016). 하지만 4차 산업혁명의 스마트공장 구축을 위해선 이기종간의 다양한 기술 상호작용에 의한 협업이 중요하다. 생산효율을 높이고 생산실적 파악 등의 목적을 위해 관련 데이터의 통합에 대한 필요성이 증대되고 있다.

이호석 외 (2006)는 분류된 데이터를 기반으로 제조 프로세스 상에서 데이터를 표현하고 관리 할 수 있는 XML Schema를 제시하여 이질적인 데이터를 통합 활용함으로써 제조 프로세스의 Activity에 대해 효과적인 데이터 연계방법을 제시하였다. Balilai et al., (2013)은 제조 프로세스의 체계적 통합을 위해 XML 데이터 구조를 이용하여 STEP 표준의 부족한 부분을 극복하기 위해 XMLAYMOD라는 새로운 플랫폼을 연구했다.

본 논문에서 다루고자 하는 JSON을 제조에서 활용한 연구는 실시간 데이터를 통합하기 위해 Kepler 모델과 같은 체계적 워크플로우가 필요하며, 워크플로우의 입력과 출력을 JSON형식으로 표현하여 연구하였다 (Korambath et al., 2014). 제조 시뮬레이션은 고비용으로 소수 기업만 사용하여 있어 오픈 소스 시뮬레이션 툴의 필요성을 주장하고, 정량적 실험을 위해 JSON을 활용해 구현하였다(Dagkakis et al., 2013).

제조환경에서의 XML을 이용한 연구 사례는 많으나 본 논문에서 다루고자 하는 JSON의 경우는 매우 드물며, 특히 정보교환 관련 논문은 드물다. 스마트공장의 발전으로 데이터의 양은 나날이 증가할 것이며 의사소통을 위한 정보교환 데이터 또한 증가할 것이기 때문에 정확하고 민첩한 의사소통을 위해 경량화된 데이터 전달 방식이 JSON의 연구가 필요하다고 생각하여 연구해보고자 한다.

II. 선행연구

2.1 첨단제조시스템

2.1.1 스마트공장

스마트공장(Smart Factory)의 정의는 다양하다. 민관합동 스마트공장 추진단에서는 “설계·개발, 제조, 유통·물류 등 생산 전체 과정에서 디지털 자동화 솔루션이 결합된 선진적 정보통신기술을 적용하여 생산성, 품질, 고객만족도를 향상시키는 지능형 유연생산공장”, 산업통산자원부에서는 “제품의 기획·설계, 생산, 유통·판매 등 전 과정은 IT 기술로 통합, 최소비용·시간으로 고객맞춤형 제품을 생산하는 공장으로써 궁극적으로는 IoT, CPS를 기반으로 제조 단계가 자동화·정보화(디지털화)되고 가치사슬 전체가 하나의 공장처럼 실시간 연동되는 생산체계를 지향”, 미래창조과학부에서는 “외부 환경 변화(고객 주문, 설비고장 등)에 공장 내 기기들이 즉각 반응하여 자율적으로 최적솔루션을 제안하는 사이버물리시스템(CPS) 기반 지능형 생산공장”이라고 정의하고 있다. 또 다른 스마트 공장의 정의는 “ICT(Information and Communication Technology)와 기존의 제조업 기술인 생산제조 기술과 융합하여 사물인터넷, 빅데이터, 클라우드 컴퓨팅, CPS(Cyber Physical System)등의 기술 등을 통해서 공장 내의 장비, 장치 부품들이 서로 연결되고 상호 소통되는 생산 체계“이다(박형욱, 2015).

이러한 많은 스마트공장의 정의가 나온 데에는 독일의 인더스트리 4.0(Industrie 4.0) 정책이 그 첫걸음이다. 인더스트리 4.0은 2010년 독일에

서 탄생한 개념으로 독일 엔지니어링 협회에 의해 2011년 하노버 박람회에서 대중에게 소개된 이후 전 세계에서 사용되고 있고, 2012년부터 ICT와 제조 산업의 융합을 통해 ‘제조업의 완전한 자동 생산 체계를 구축하고 모든 생산 과정이 최적화’되는 인터스트리 4.0 정책을 적극 추진하고 있다(이용구, 2017). 4차 산업혁명이란 개념은 2016년 다보스포럼 이후 보다 더 자주 사용되고 있다³⁾. 표 2-1은 세계 주요국의 4차 산업혁명 도입 개념을 보여준다.

표 2-1 세계 주요국의 4차 산업혁명의 도입 개념³⁾

2011년 미국	‘첨단 제조 파트너십 2.0’. 고급 제조업 일자리 창출과 세계 경쟁력 강화
2011년 영국	‘전송 센터’. 제조업의 GDP 기여를 두 배 확장.
2012년 이탈리아	‘지능형 공장 클러스터’. 4개의 프로젝트 연구 개발 및 활용 제조 커뮤니티 구성.
2014년 벨기에	‘지능형 공장 클러스터’. 미래형 공장 개발 지원.
2014년 중국	‘중국제조 2025’. 10개 부문 우선 디지털화와 현대화로 제조 강국으로 탈바꿈.
2015년 일본	‘재활성화 및 로봇 전략’. 서비스 산업 생산성 향상, 2020년까지 로봇 배치 증가.
2015년 한국	‘제조업 혁신 3.0’. 신기술 기반 제조업 생태계 구축 및 스마트공장 개발 촉진.
2015년 프랑스	‘미래의 산업’. 자율주행차, 전기 비행기 등 특화 제품 개발 지원.

독일은 제조업의 미래 경쟁력 제고를 위해, 2012년부터 ICT와 제조 산업의 융합을 통해 ‘제조업의 완전한 자동 생산 체계를 구축하고 모든 생산

3) 롤랜드버거 (2017). 4차 산업혁명 이미 와 있는 미래, 다산 3.0

과정이 최적화'되는 4차 산업혁명을 의미하는 Industry 4.0 정책을 적극 추진하고 있다(이용구, 2017).

미국의 제조혁신네트워크(NNMI), 독일의 인더스트리 4.0, 일본의 개혁신 2020, EU의 Horizon 2020: Factories of Future, 중국의 중국제조 2025 등 제조 선진국들은 후발국들과의 제조 기술 격차를 늘리기 위해 후발국은 선진국과의 기술 격차를 줄이기 위해 국가적 차원에서 ICT융합을 통한 제조업 스마트화 정책을 활발히 추진하고 있다(이용구, 2017).

표 2-2 주요국의 제조업 ICT융합 정책(이용구, 2017)

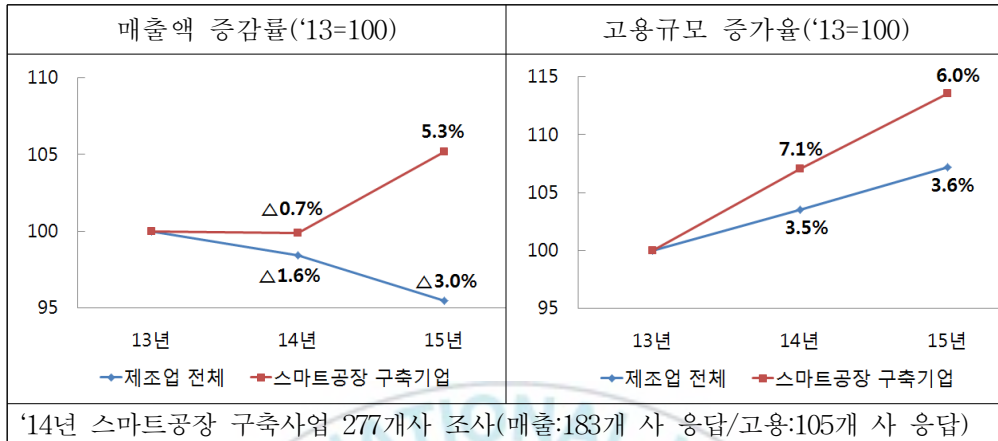
구분	정책·전략	주요 프로그램
미국 (거점형)	첨단 제조 파트너십(AMP) - 기술혁신 플랫폼 제공, 첨단제조 기술 로드맵 작성 - 중소기업이 사용가능한 시설 정비 - 중점 4영역 : 안전보장, 첨단재료, 차세대 로봇틱스, 제조공정·에너지 효율성 향상	제조혁신네트워크(NNMI) 45개 제조혁신연구소(IMI) 설치(2012~) - America Makes (적층가공기술) 시행 - 파워일렉트로닉스, 경량·신소재 금속, 디지털 제조·설계, 첨단복합재료의 거점 구축
독일 (프로젝트형)	하이테크 전략 - CPS로 네트워크화된 스마트 제조 - 국내 제조기반 강화와 제조 시스템 수출을 겨냥한 듀얼 전략	Industrie 4.0(2011~) - 차세대 제조업 연구 - Autonomikfür Industrie 4.0, Smartfactory KL (인공지능 연구소 등)
일본 (민생형)	일본재흥전략 - 미래투자를 통한 생산성 혁명: 혁신·벤처 창출 - IoT, 빅데이터, 인공지능을 통한 산업·취업 구조 변혁 - 중견·중소기업의 '수익력'강화	개혁신 2020 성장 가속화 민·관 합동 프로젝트 : 자동주행, 수소사회, 첨단로봇 등의 분야에서 2020년까지 가시적인 성과 창출

EU (문제 해결형)	Manufacture 전략 연구 아젠다 • 고부가가치의 새로운 제품·서비스와 새로운 비즈니스 모델의 창출 • 새로운 제조 공학의 창출 • 연구·교육 인프라의 정비	Horizon2020(2014~2020) ·Factories of Future • 선진적 제조공정 • 응용성 있는 스마트한 제조 시스템 • 가상화되고 자원 고효율인 공장 • 연계·이동가능성이 큰 기업 활동 • 인간 중심의 제조 • 소비자의 요구에 따른 제조
중국 (복합형)	• 공업화와 정보화의 고도융합 • 네트워크화, 디지털화, 지능화 기술의 개발·이용, 인터넷과의 융합중시 • 자원 활용의 효율화	중국제조 2025 • 하이테크산업 진흥프로그램 및 국가자연과학기금 등 • 중국과학원 연구개발센터 설치

독일의 인더스트리 4.0 전략의 핵심은 IoT(사물인터넷) CPS(사이버물리시스템), 인공지능, 센서 등을 통해 제조전주기를 고도화하는 스마트공장의 구현이라 할 수 있다(김선재, 2017).

한편 한국에서도 민관합동 스마트공장추진단을 중심으로 스마트공장 보급 사업을 적극 추진 중이며 기초, 중간1, 중간2, 고도화 참조모델을 제공하고 있다. 산업통상자원부가 4차 산업혁명에 대비한 제조업 혁신 가속화를 위해 스마트공장 보급 확대 및 고도화, 국내 스마트공장 공급 산업 경쟁력 강화를 위해 2017년 5000개 보급을 목표로 추진 중이다. 민관합동으로 총 1,108억 원을 투입해 2,200개 이상(누적 5,000개) 중소·중견기업의 스마트공장 구축을 지원한다(산업통상자원부, 2017).

표 2-3 스마트공장 구축기업 현황(산업통상자원부, 2017)



정부는 ‘14년부터 스마트공장의 확산을 통해 제조업 생태계를 혁신하고 주력산업 경쟁력 제고를 꾀하기 위해 ‘제조업 혁신 3.0 전략’을 수립 추진 하였다(김선재, 2017). 2014년도 스마트공장 보급·확산 사업의 성과로는 참여기업들의 매출·고용 등 경영실적을 분석한 결과, 국내 제조업 매출이 부진하던 2014년, 2015년에 사업 참여기업의 매출은 성장세로 전환되었고, 제조업 평균 대비 높은 고용 증가세를 유지 하였다. 최근엔 2016년까지 스마트공장 보급·확산사업의 지원을 받은 2,800개 중소·중견기업들에게서 스마트공장 구축에 따른 성과가 점차 가시화되고 있는 것으로 나타났다(산업통상자원부, 2017).

스마트공장 역량이 글로벌 제조 혁신으로 이어지기 위해 구성모델, 기능요소, 상호 정보 교환모델 등 표준화에 대한 전략적 투자가 이루어져야했고 정부는 ‘14년부터 스마트공장의 확산을 통해 제조업 생태계를 혁신하고 주력산업 경쟁력 제고를 위해 ‘제조업 혁신 3.0’을 추진하였다(김선재, 2017). 신승준 외 (2016)는 스마트공장의 필수요소로 최적의 생산계획 및 제어를 위해 시기적절하고 정확한 의사결정을 가능하게 하는 빅데이터 애널리틱스

(Big Data Analytics)를 꼽았다.

2.1.2 사이버물리제조시스템

CPS(Cyber Physical System)란 “현실세계의 다양한 현상과 디지털 사이버세계가 긴밀하게 결합”된 시스템이다. ICT 기술의 급속한 혁신으로 IoT가 성작하면서 CPS도 동시에 부상하고 있다. IoT와 CPS는 유사한 개념이지만 IoT는 현실세계, CPS는 사이버세계에 중점을 둔 차이가 있고, 현실세계와 사이버세계의 연결과 연결 후 현실·가상 정보데이터의 융합·분석과 분석 데이터를 현실에 피드백 하는 것에 중점을 두고 있다 (이정아, 2015).

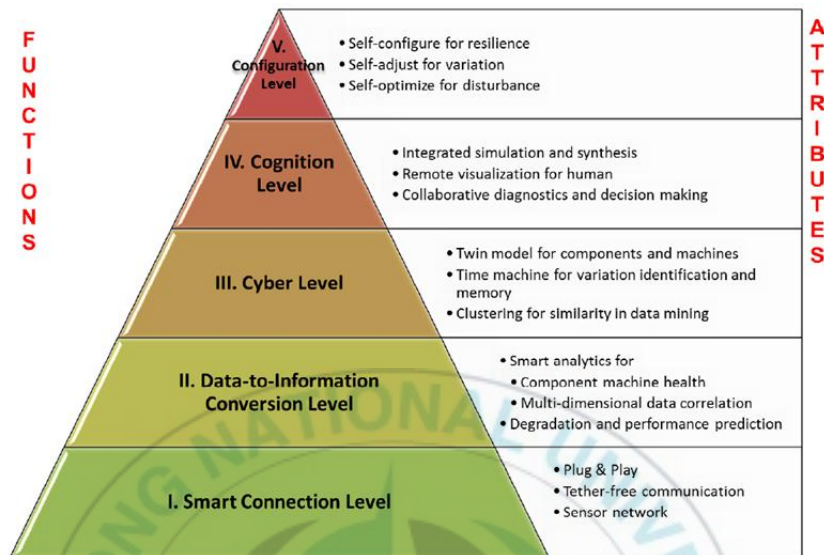
CPS의 가장 큰 3가지 기술적 요소는 communication, computation, control이다. communication은 밑에서 발생하는 데이터를 수집하는 기술, computation은 수집된 데이터를 바탕으로 어떤 계산을 통해서 공장을 제어하거나 사용자에게 의사결정을 지원하기 위한 정보를 제공하는 기술이며, control은 그 정보로 다시 공장을 제어하기 위한 기술이다⁴⁾.

일반적으로 CPS의 두 가지 주요기능을 가지고 있다. 첫째, 실시간으로 데이터를 수집하고 사이버 공간의 정보와 데이터를 주고받는 연결성. 둘째, 사이버 공간을 구성하는 지능형 데이터 관리와 분석기능이다.

Lee et al. (2015)는 5C 아키텍처는 제조업의 CPS의 중요성과 명확한 정의를 위해 제시하여 그림 2-1로 나타냈다.

4) 「CPS는 지능형 시스템 구축 기술...가상-리얼 연동한 공장 구현」, 『헬로티』, 2016년 7월 11일,
http://www.hellot.net/new_hellot/magazine/magazine_read.html?code=201&sub=002&idx=29783

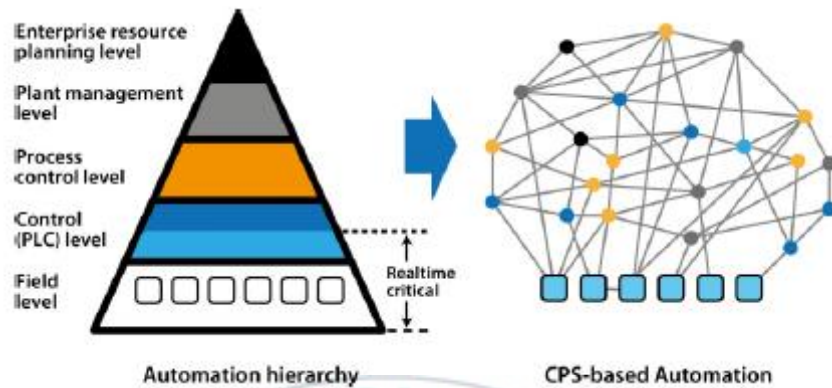
그림 2-1 CPS 구현을 위한 5C 아키텍처(Lee et al., 2015)



제조 환경에서는 무선, 센스네트워킹과 데이터분석을 통해 쓸모 있는 데이터를 유추해야 한다. 사이버수준은 정보의 허브 역할로 연결된 모든 컴퓨터의 정보를 통해 시스템 네트워크가 형성된다. 엄청난 양의 정보가 수집되어 기계의 성능을 비교하고 평가할 수 있는 자체 비교평가 시스템에 제공해 자율적인 업무가 가능하다. 인식(Cognition) 수준에서는 획득한 지식을 사용자에게 전달하기 위해 적절한 그래픽 정보를 주게 된다. 구성(Configuration)단계에서는 모든 기계를 자체 구성 및 적응을 할 수 있도록 제어가 가능하다(Lee et al., 2015).

전통적인 자동화 피라미드에서 가장 아랫부분인 물리적 층에는 센서, 작동장치, 임베디드 컴퓨터 네트워크 구성요소(생산 측정 기준 제공, 활동 수행, 연산과 정보 릴레이)가 있다. CPS 기반 자동화는 분산된 서비스를 통해 서로 다른 계층에서 활용 가능한 정보들이 제한되지 않고 인간-기계 인터페이스에 자율적이고 협업적인 시스템을 확보한다(Monostori, 2014).

그림 2-2 CPS기반 분산서비스(Monostori, 2014)



최근의 발전으로 인해, 센서, 데이터 수집 시스템 및 컴퓨터 네트워크의 가용성과 경제성이 높아지면서 오늘날 산업의 특성으로 인해 더 많은 공장에서 첨단기술을 구현하는 방향으로 나아갔다. 그 결과로 선서와 네트워크화된 기계의 사용이 계속 증가하면서 빅데이터라 불리는 대용량의 데이터가 지속적으로 생성된다(Shi et al., 2011).

제조분야에서의 CPS는 스마트 제조를 실현하기 위한 핵심기술이며 클라우드, IoT, 빅데이터와 같은 기술과 밀접한 관련을 갖고 연구되고 있다. 제조 분야의 CPS연구는 아직 초기 단계에 있으며 대부분의 연구는 실현보다는 모델링, 개념화, 활용 계획에 중점을 둔다(Lee et al., 2015).

CPPS(Cyber Physical Product System)는 기존의 자동화 계층구조(automation hierarchy)와는 부분적으로 다른 구조인 분산된 제어구조를 가지고 있다(Kang et al., 2016).

국가적으로 제조업에 ICT 기술을 융합하여 스마트공장 구현이 진행 중으로, IoT 개념은 CPS기술과 접목하여 제조·생산 분야에 적용하는 CPPS 기술 개발과 확산을 추진 중이다(이용구, 2017).

스마트공장 환경에서는 이기종 설비 간 상호작용을 위한 통신 및 연동이

중요하다. 현재 산업현장에서 사용되고 있는 산업용 통신방식이나 구조가 점차 진화하는 스마트공장의 대용량 처리를 위한 구조로 적합하지 않다. 전형국 외 (2017)에서는 스마트공장을 위한 CPPS 연동 미들웨어 프레임워크를 제안해 스마트공장에서의 CPPS의 중요성을 연구하였다.

2.1.3 홀로닉제조시스템

홀로닉 제조 시스템은 홀론들이 수평적이면서도 계층적인 홀론조직을 구성하며, 자율적이고 협업적인 역량을 통해 홀론성을 상승시켜 특정 목적을 달성하고자 하는 유기적 제어 구조를 의미한다(Brussel et al., 1998). 계층적 구조는 상위 요소가 하위 요소를 엄격히 제어하기 때문에 운영에 있어 안전성이 높아 가장 알맞은 제시를 할 수 있지만 자율성이 거의 없으므로 환경의 변화에 민첩하게 대처해 나가는 면이 부족하다. 수평적 구조는 각 요소가 자율적인 의사결정 권리와 수행권리를 가지고 있어서 시스템 제어의 복잡성을 감소시켜주고 환경 변화에 빠른 적응이 가능하고 정보처리도 아주 신속하게 이루어지지만 제어 요소가 많이 분산되고 자율화되어 시스템의 운영에 있어 통일적이지 못하고, 안전성 보증을 못하며 개개 요소의 최적화는 가능하게 실현될 수 있으나 전체 시스템 목표의 최적화는 실현되기 어렵다. 이영해 외 (1997) 홀로닉 제조 시스템의 용어들은 다음과 같이 정의 할 수 있다(Brussel et al., 1998).

- 홀론(holon) : 정보 및 물리적 물체를 변형, 운송, 저장 하는 요소로 자율적이며 협업의 가능한 제조요소. 홀론의 종류로는 설비홀론(resource holon), 제품홀론(product holon), 주문홀론(order holon)이 있다.
- 자율성(autonomy) : 홀론 자체의 계획, 전략을 생성하고 통제할 수 있는 능력

- 협업성(cooperation) : 복수개의 홀론들 간 상호작용하여 계획을 실행하는 프로세스
- 홀론조직(holarchy) : 목표달성을 위해 협력하는 홀론 시스템. 홀론조직에 의해 홀론의 자율성이 제한될 수 있음.

기본홀론의 개념에서 설비홀론(resource holon)은 물리적 부분, 즉 제조 시스템의 생산 설비, 설비제어 정보처리 부분을 담당한다. 공장, 기계, 컨베이어, 팔레트, 원료, 공구, 자재보관, 인적자원, 에너지 등을 포함하고 있다. 제품홀론(product holon)은 완벽한 품질의 제품을 제조할 수 있도록 프로세스와 제품정보를 가지고 있다. 제품수명주기, 사용자요구사항, 설계, 공정 계획 등의 정보가 포함되어 있다. 주문홀론(order holon)은 설비와 제품을 연결하는 주문을 나타낸다. 고객주문, 시제품 제작, 수리 주문 등을 포함하는 개념이다(Brussel et al., 1998).

계층적 구조(hierarchical organization structure)의 단점 해결을 위해 유연 생산 시스템의 개체를 홀론으로 정의된 에이전트로 구성하여 에이전트 간 메시지 전달을 통한 상호작용으로서 홀로닉 제조 셀 제어를 객체지향 접근법에 의해 개발하였다(김병주 외, 1999). 홀로닉 제조 시스템의 추상적 개념을 제조현장(shop floor)에 적용하기 위한 시스템 설계 및 개발에 대한 연구가 시도되었으며, 주로 제어자동화, 작업할당, 장애극복, 실시간 제어, 생산 계획 및 스케줄링 등의 분야에 활용되었다. 제어자동화(automation)는 공장자동화를 위하여 저수준(low level)에서의 분산제어용 기능단위 구조 및 고수준(high level)에서의 홀론 간 상호작용 및 통합구조를 설계하는 것이다(Babiceanu et al., 2006). Colombo et al(2006)은 실시간 분산 제어를 위한 메카트로닉스 컴포넌트 행동기반 제어 플랫폼을 구현하였고, Barbosa et al.. (2015)은 왜란(disturbance) 영향의 최소화를 위한 행동 그리고 구조

모델에 기인한 진화론적 자가 구성 제어구조를 구현하였다. 작업할당(task allocation)은 가용 설비로의 최적 작업 분배를 위하여 정보를 공유하거나 기능을 수행하는 것으로, 비딩(bidding) 프로세스를 통해 작업을 할당하는 Contract Net Protocol (CNP) 기술을 보편적으로 활용하였다(Babiceanu et al., 2006). Babiceanu et al(2004)은 왜란이 발생했을 때 광역적 최적화 및 실시간 대응을 위한 작업할당이 이루어지는 홀로닉 제어 구조를 제안하였다. Owliya et al(2013)은 작업할당에 적용되는 에이전트 토폴로지 모델을 개발하고 시뮬레이션을 통하여 리드타임, 비용 및 활용성 측면에서 기존의 토폴로지 모델들과 비교를 실시하였다. 장애극복(fault-tolerance)은 장애상황의 자동감지, 장애원인 진단 및 장애복구 활동을 의미한다(Babiceanu et al., 2006). Leitao(2011)는 지연 및 주문체증(rush orders)에 의한 제조현장 장애상황을 예측적이며 예방적 접근방식으로 해결하는 분산 구조를 설계하였다. Marinho da Silva et al(2014)은 페트리 네트를 이용하여 장애발생 시 분산 제조시스템의 재구성력 향상을 위한 서비스기반 홀로닉 제어구조를 제안하였다. 실시간 제어는 주어진 시간제약내에 이벤트에 반응을 하는 제어로서, 시간제약을 놓쳤을 때 데미지가 큰 경우를 경성 실시간(hard realtime), 놓치더라도 데미지가 크지 않은 경우를 연성 실시간(soft realtime)으로 구분할 수 있다(Babiceanu et al., 2006). 여기서는 실시간 제어를 위한 저레벨의 분산 제어 구조 설계가 중요하며, Brennan et al(2002)은 IEC 61499 기능 블록 모델을 활용한 실시간 분산제어 시스템을 설계했으며, Indriago et al(2016)는 실시간 피드백 제어가 가능하도록 홀로닉 기능단위 설계 및 구현을 실시하였다. 생산 계획 및 스케줄링은 홀론조직의 최적화 운영을 위하여 홀론들이 생산계획 및 스케줄링을 결정하는 활동이다(Babiceanu et al., 2006). Gou et al.(1998)는 복수 개의 셀로 구성된 공장을 대상으로 홀론간 분해와 조정기반 스케줄링 방법을 제안하였으며,

Shen et al(2000)은 설계, 계획, 스케줄링, 시뮬레이션 및 실행의 통합을 위하여 중재자 에이전트를 둔 아키텍처를 개발하였다. 또한, Gamboa Quintanilla et al(2016)은 생산 계획 및 스케줄링의 유연성 확보를 위하여 페트리 넷를 이용한 교착 자유(deadlock-free) 다중 공정계획 경로 및 최적 경로 결정을 위한 홀로닉 제어 구조를 개발하였다.^{가)}

2.2 에이전트 시스템

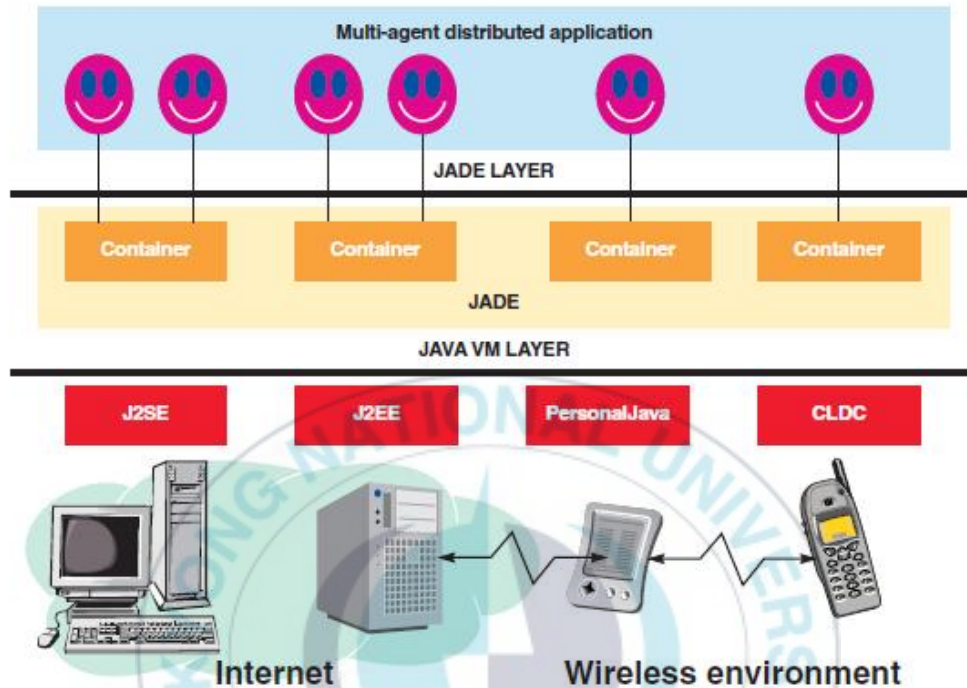
2.2.1 JADE

JADE(Ja Agent DEvelopment Framework)는 여러 FIPA 에이전트 간의 조직성을 서포트하고, 에이전트간의 통신을 용이하게 하는 통신언어인 FIPA-ACL의 표준을 제공하며⁵⁾ 분산된 멀티 에이전트 개발을 용이하게 돕는 미들웨어이다. JADE의 목적은 시스템 서비스와 에이전트의 포괄적인 집합화를 위해 표준에 준하는 환경을 제공함으로써 개발을 단순화 함에 있다. FIPA-OS(Foundation for Intelligent Physical Agents-Open Source)를 기반으로 하고 있으며, AMS(Agent Management System), DF(Directory Facilitator), MTS(Message Transport System)으로 구성되어 있다(F.Bellifemine et al., 2013).

5) <http://jade.tilab.com>

가) Shin, S. J., Woo, J. Y. & Seo, W. C.(2017), Development of a Data and Model-Interconnected Holonic Architecture for Intelligent Decision-Making on Cyber-Physical Production Systems, Journal of the Korean Institute of Industrial Engineers, 43(6), 451-463

그림 2-3 JADE 아키텍처(F.Bellifemine et al., 2013)



AMS는 에이전트의 생성, 등록, 제거, 복구 등의 전반적인 에이전트의 생애주기에 관한 내용을 관리하며 에이전트 플랫폼에 대한 접근과 사용에 대한 관리해 하나의 플랫폼에 하나의 AMS만이 존재한다. DF는 에이전트에 대한 옐로우 페이지 서비스(Yellow page service) 기능을 제공해 에이전트 플랫폼내의 에이전트들이 제공하는 능력이나 서비스에 대한 정보를 각 에이전트에게 제공하는 역할을 한다. MTS는 같은 플랫폼 뿐 아니라 다른 플랫폼 내에서 교환되는 메시지를 관리한다(노선택 외, 2007; F.Bellifemine et al., 2013).

노선택 외 (2007)는 JADE의 자율적이고 협업적인 통신체계를 이용하여 JADE 플랫폼에서 행동하는 자율적이고 자치적인 멀티 에이전트 시스템에 긴박한 응급 의료 시스템 모델(MAEMS : Multi-Agent based Emergency

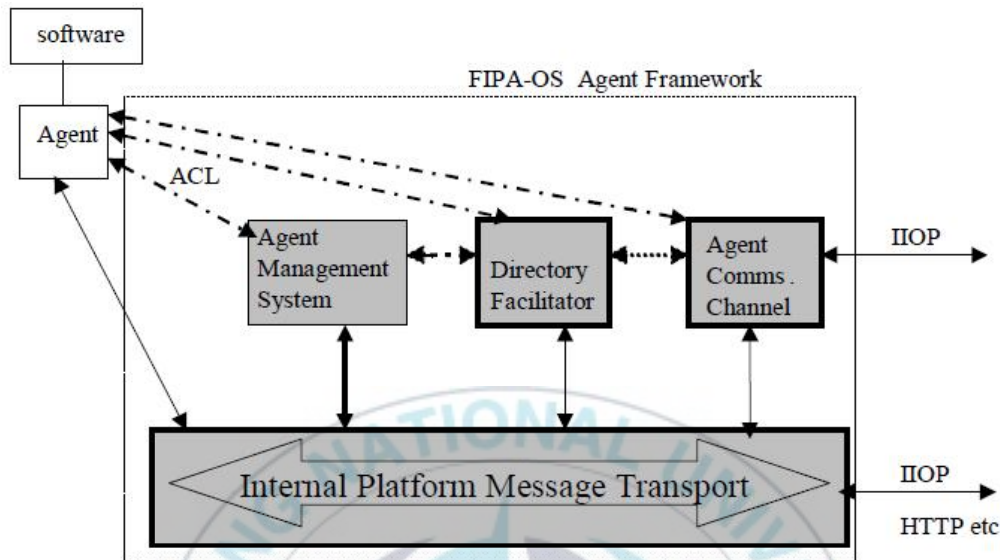
Medical System)을 제안해 연구했다. 박홍석 외 (2011)는 진화형 제조시스템(EMS : Evolvable Manufacturing System)의 개발을 위해 인지에이전트는 PC에 설치된 JADE 플랫폼 기반으로 구동되며, 에이전트 간의 메시지 교환 및 데이터 전송을 위한 포맷으로 XML, 에이전트 간의 통신언어로는 FIPA-ACL가 사용되어 구현되었다. 임선종 외 (2007)에서 JADE 패키지를 분석하고 JDK 1.4를 이용한 플랫폼 구현을 통해 독립된 E-Beam system의 운영 시스템 뿐 아니라 가공 라인 구축에서 협력 시스템의 운영 시스템으로 활용할 수 있음을 보여주는 연구를 했다.

2.2.2 FIPA

FIPA(Foundation for Intelligent Physical Agents)는 에이전트 기반 기술 및 표준과 다른 기술과의 상호 운용성을 추구하는 IEEE 컴퓨터 협회 표준 조직이다. 1996년 스위스의 비영리단체로 설립되어 이기종 및 에이전트의 상호작용, 에이전트 기반 시스템에 대한 소프트웨어 표준 사양을 만들었다⁶⁾. 스위스의 비영리단체는 2005년 해체되어 IEEE 표준위원회가 설립되었다. 그림 2-4는 FIPA 97 에이전트 참조모델이다.

6) <http://www.fipa.org>

그림 2-4 FIPA 97 에이전트 참조모델 (Poslad et al., 2000)



에이전트의 운영과 생성 및 ACL 통신을 제공하기 위해 다음의 기본 에이전트 및 요소를 포함하고 있다. DF(Directory Facilitator), 에이전트 관리 시스템(Agent Management System), 에이전트 통신 채널(Agent Communication System), IPMT(Internal Platform Message Transport) 등이다.

DF는 옐로우 페이지 서비스로 에이전트의 검색을 위한 서비스를 제공한다. AMS는 에이전트 등록, 해지를 담당한다. ACC는 에이전트 간 통신을 지원해 서로 다른 플랫폼에서 상호 운용성을 촉진한다. IPMT는 메시지 라우팅 서비스를 제공한다(Poslad et al., 2000).

FIPA 표준 중 가장 많이 사용되는 것은 에이전트 관리 및 에이전트 통신 언어(FIPA-ACL)이다⁶⁾. 실제로 에이전트는 사용자, 시스템 등 협업, 협상을 진행하기 위해선 서로 통신 할 수 있어야 한다. 에이전트 통신 언어(Agent Communication Languages : ACL) 이라는 특수한 통신 언어를 사용해 상호작용하며 통신한다. 현재 가장 많이 연구되고 사용되는 에이전트

통신 언어는 FIPA ACL이다. FIPA ACL의 주요 특징은 정해진 상호작용 프로토콜을 통해 다양한 언어와 대화를 관리 할 수 있다는 것이다 (Dagkakis et al., 2007).

2.3 데이터 표현

2.3.1 XML

XML은 eXtensible Markup Language로 확장성 마크업 언어이다. 문서처리를 지원하기 위해 문서에 추가되는 정보인 마크업 정보를 표현하는 언어이다. 가장 먼저 나온 마크업 언어인 SGML(Standard Generalized Markup Language)은 일반 마크업 언어를 정의하고 기술하기 위한 언어로(주용하, 2001) 특정 문서를 정의하는데 필요한 태그를 임의로 생성해 문서 구조를 정의할 수 있으나 너무 복잡하여 활용하기 어렵고 이를 지원하는 소프트웨어를 개발하기가 어렵다는 문제가 제기되면서⁷⁾ 제한된 태그만을 사용한 HTML(Hyper-Text Markup Language)이 쉬운 사용법으로 인터넷 대중화에 성공했다(주용하, 2001). HTML은 표현중심의 언어로 사용된 태그로 문서를 어떻게 표현하느냐는 알 수 있지만 문서의 구조와 태그에 포함된 내용의 의미를 전달하기에는 부적합한 언어로(주용하, 2001) 이러한 단점을 보완하기 위해 XML이 등장했다.

XML의 특징은 단순성, 일반성, 인터넷을 통한 사용가능성으로 볼 수 있다(김상현 외, 2014). 장점으로는 첫째, XML은 문서의 문법과 구조를 문서

7) 한국기록학회. (2008). “기록학 용어 사전”, 역사비평사

의 표현과 분리가 가능하다. 둘째, 문서의 내용에 관련된 독자적인 태그를 만들 수 있다. 셋째, XML은 사람이 읽을 수 있는 형식이다(주용하, 2001). 나머지로써는 수많은 종류의 데이터를 표현하는데 효과적이며, 주로 다른 시스템과 데이터를 쉽게 주고받을 수 있어 HTML보다 풍부하고 유연하게 사용할 수 있는 장점이 있다. XML은 HTML과의 비교가 많았다. 2000년 초부터 HTML의 역할을 XML이 대체하면서, 보다 능동적이고 고도화된 작업이 웹을 통해 가능해졌다(김상현 외, 2014). 이러한 XML의 활용분야로는 인터넷결제, 검색, 건설 분야에서의 활동 등을 들 수 있다(주용하, 2001). 장점이 있는 만큼 단점으로 장황하고 복잡하다는 점을 들 수 있다. 프로그래밍 언어나 데이터베이스의 시스템을 입력하기 위해 XML을 맵핑(Mapping)하기가 어렵고, 데이터가 특정 애플리케이션에 맞춰 구조화된 경우는 더 까다로웠다. 그리고 많은 양의 태그로 용량이 늘어나 응답이 느린 것도(김현수 외, 2016) 꼽을 수 있어 최근 JSON이 개발자 사이에서 더 유연하고 빠른 형식으로 대체되고 있다(Patrizio, 2016).

2.3.2 JSON

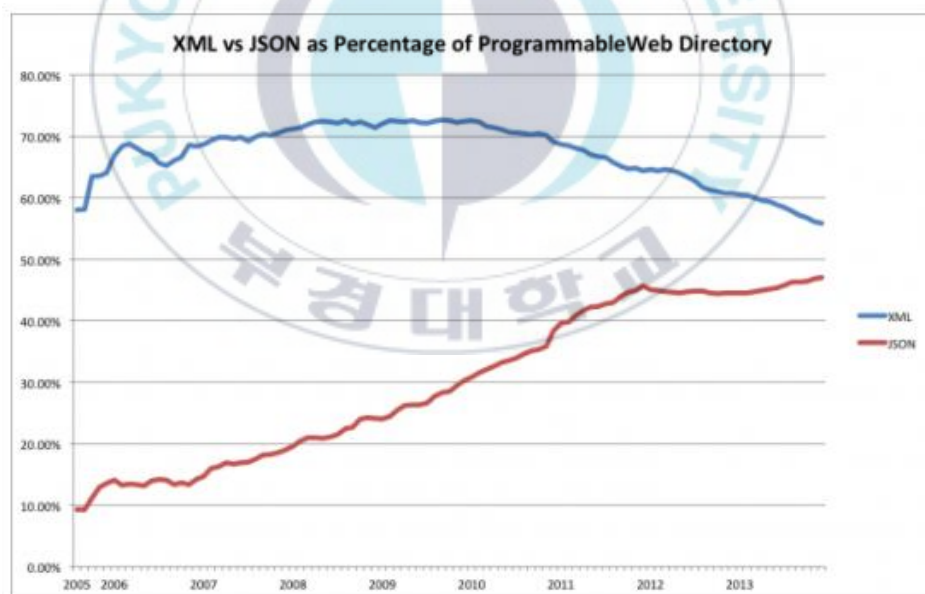
JSON은 Java Script Object Notation의 약어이다. 2001년 더글라스 크락포드는 텍스트 형식으로 데이터를 저장하는 자바스크립트 문법을 ‘JSON’이라 이름 붙이고 문서화 했다(Rausschmayer, 2012). 자바스크립트 언어 기반의 객체 표기, 즉 자바스크립트 언어에 그 근본을 두고 있어 다른 언어는 JSON의 형식을 이해 할 수 없기 때문에 라이브러리를 설치 후 사용 가능하다. 매우 경량화된 데이터 구조의 정의, 저장, 교환을 위한 일종의 메시지 포맷으로 모든 프로그래밍 언어 간의 구조화된 데이터 교환을 용이

하게 사용할 수 있는 텍스트 형식이다. XML은 마크업 언어처럼 JSON도 일종의 마크업 언어로 마크업문법을 가진다(김희정, 2016).

JSON은 XML과 자주 비교되는 마크업 문법이다. JSON은 XML보다 매우 간결하여 인코딩과 디코딩이 직관적으로 빠르게 진행된다. 이러한 간결성을 기반으로 모바일, 클라우드 컴퓨팅 시대에 글로벌 웹서비스들의 API 호출 규약에서 기존 XML로 표준화되었던 많은 부분들이 점차 JSON을 표준 입출력 포맷으로 규정하고 있으며(김희정, 2016) JSON은 사실상의 표준으로 알려져 있다.

그림 2-5 프로그래밍 가능한 웹 디렉토리 XML과 JSON의 비율

(Vander, 2013)



JSON이나 XML이냐는 논쟁은 10년간의 프로그래밍 개발의 큰 주제 중 하나이다. XML은 1996년부터 정의된 표준과 같은 장점이 몇 가지 있지만 JSON의 경량화된 표현방법이 보급되면서 2009년 중반이 후 XML의 데이터 형식이 감소하고 있고 JSON은 8년간 계속 증가하고 있다. 그림 2-5는 2005년 말부터 JSON과 XML을 지원하는 ProgrammableWeb 디렉토리에 추가된 API의 비율을 보여준다(Vander, 2013). XML은 줄어들고 있는 반면 JSON은 증가하는 것을 볼 수 있다.

XML과 JSON을 비교하는 연구는 여러번 진행되어 왔다. 김현수 외 (2016)에서 헬스케어 디바이스와 헬스케어 플랫폼과의 데이터 정보 표현, 교환, 메시지 규격 등 국제 표준준수가 필요하다고 기술하여 헬스케어 디바이스와 OM2M 플랫폼 간 효율적인 전송 구문에 대한 평가를 위해 XML과 JSON간 단리 처리과정의 패킷 사이즈와 전송 패킷 수 등을 분석하였다. 그 결과 XML 보다 JSON 사용 시 리소스 소모 및 응답시간이 더 낮은 것을 확인, JSON 방식이 XML방식보다 적합함을 확인하였다.

Nurseitov et al(2009)는 현재 업계에서 사용되고 있는 두 가지 데이터 교환 방식인 XML과 JSON을 비교하여 데이터 전송 속도와 성능을 비교하여 어느 것이 더 효율적인 영향을 미칠지 연구하여, 표 2-4, 2-5는 연구 결과 데이터로 전송되는데 걸리는 시간과 CPU 사용량에서 JSON이 더 유리한 연구결과를 도출하였다.

표 2-4 JSON, XML 데이터 전송속도(Nurseitov et al., 2009)

	JSON	XML
Number Of Objects	1000000	1000000
Total Time (ms)	782527.9	4546694.78
Average Time (ms)	0.08	4.55

표 2-5 JSON, XML의 CPU 사용량(Nurseitov et al., 2009)

	Average% User CPU Utilization	Average% System CPU Utilization	Average% Memory Utilization
JSON	86.13	13.08	27.37
XML	54.59	45.41	29.69

김현철(2015)는 RFID를 이용한 제조공정 자동화 시스템에 이력관리 서버를 추가하여 인건비를 절약해 낭비되는 인력을 줄이는 연구가 진행되었는데 안드로이드(Android)나 태블릿 PC를 이용해 이동 중에도 모니터링 할 수 있는 프로그램을 만들었다. Java서버에서는 JSON파싱을 이용해 안드로이드에도 전송이 가능하기 때문에 JSON 관련연구도 참고해 연구하였다.

본 연구를 위해 JSON과 XML을 비교하여 다음의 표 2-6로 정리하였다.

표 2-6 JSON vs XML

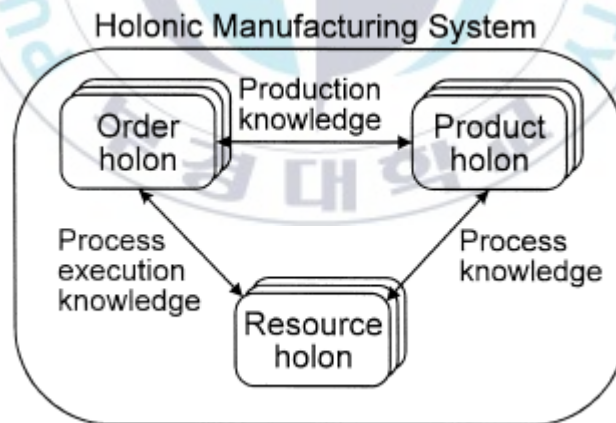
JSON	XML
• 텍스트 기반으로 사람과 기계 모두 해석 가능 • 프로그래밍 언어 및 OS에 독립적으로 상이한 시스템간의 객체 교환이 가능 • 자바스크립트 문법을 따르므로 웹 프로그래밍에 적합 • 각 주요 브라우저가 기본 파서를 탑재하고 있어 범용성 높음 • Syntax와 Datatype을 제공하지 않음 • 용량이 커질 경우 파싱(parsing) 자체의 시간이 오래 걸림 • 데이터 분석시 각 센서별로 들어오는 작고 빠른 데이터의 경우 유리함 • 문서의 정합성은 담보하지 못함 • 사실상 표준	• 텍스트 기반으로 사람과 기계 모두 해석가능 • 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어의 한가지인 마크업 언어 • HTML보다 풍부하고 유연한 사용 • 태그 사용으로 인해 JSON 대비 문서의 용량이 큼 • JSON 대비 복잡함 • W3C (World Wide Web Consortium)에서 제정한 표준

Ⅲ. 시스템 설계

3.1 개념구조

홀로닉 제조시스템의 기본홀론은 제품홀론(product holon), 오더홀론(order holon), 리소스홀론(resource holon)으로 구성되어 있다(Brussel et al., 1998). 홀론들은 자신의 정보를 관리하고 시스템 전체의 목표 수행을 위해 협력한다. 각 홀론들은 다른 홀론과의 정보교환을 위해 공통의 데이터 형식을 갖는다(이영해 외, 1997).

그림 3-1 홀로닉 제조 시스템의 기본 개념(Brussel et al., 1998)



홀로닉 생산시스템의 구성을 위해 갖추어야 할 요소는 다음과 같다. 첫째, 시스템의 목표에 영향을 줄 수 있는 시스템 요소들에 대한 홀론을 구성할 필요가 있다. 둘째, 각 홀론들의 역할이 명확하게 정의되어야 한다. 각 홀

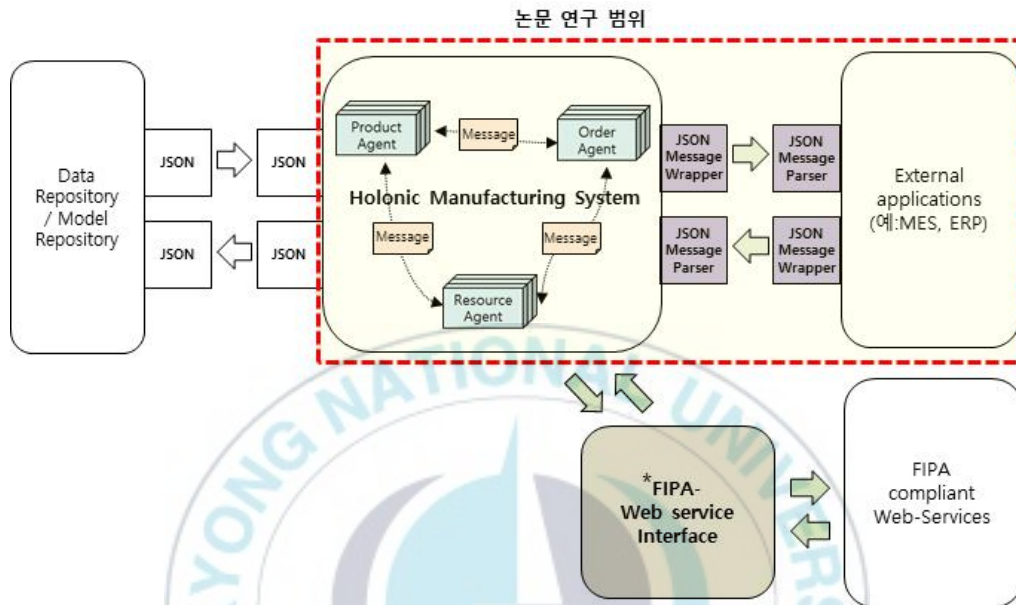
론은 어떤 정보를 유지해야 하고 또 누구에게 자신이 가지는 정보를 어떤 형태로 제공해야 하는지, 다른 홀론들과 어떤 관계를 갖게 되는지 등이 명확하게 정의되어야 한다. 셋째, 홀론간의 협력 및 협상을 위한 룰이 명확하게 정의되어야한다(이영해 외, 1997).

그림 3-1은 본 논문의 개념구조를 나타낸다. Holonic Manufacturing System의 개념에 따라 product agent, order agent, resource agent 간 공장 내부의 세 개의 에이전트로 운영된다. 내부 세 개의 에이전트 간 자율적인 메시지 주고받으며 협업하는 구조이다.

데이터 저장소에는 제조 에이전트 저장소 (Manufacturing Agent Repository), 의사결정 모델 저장소 (Decision-making Model), JSON에 기반을 둔 제조데이터 저장소 (Manufacturing Data Repository)가 있다. 내부(Internal)과 외부 어플리케이션(external applications)사이에 FIPA 호환이 불가능한 경우 외부 어플리케이션과의 연결을 위해 단순한 표현방법인 JSON을 통한 정보교환을 목적으로 한다.

본 논문의 연구범위는 그림 3-2의 빨간색 점선으로 FIPA 구축이 되어 있지 않은 경우 JSON을 이용한 정보교환 운영방식을 연구했다.

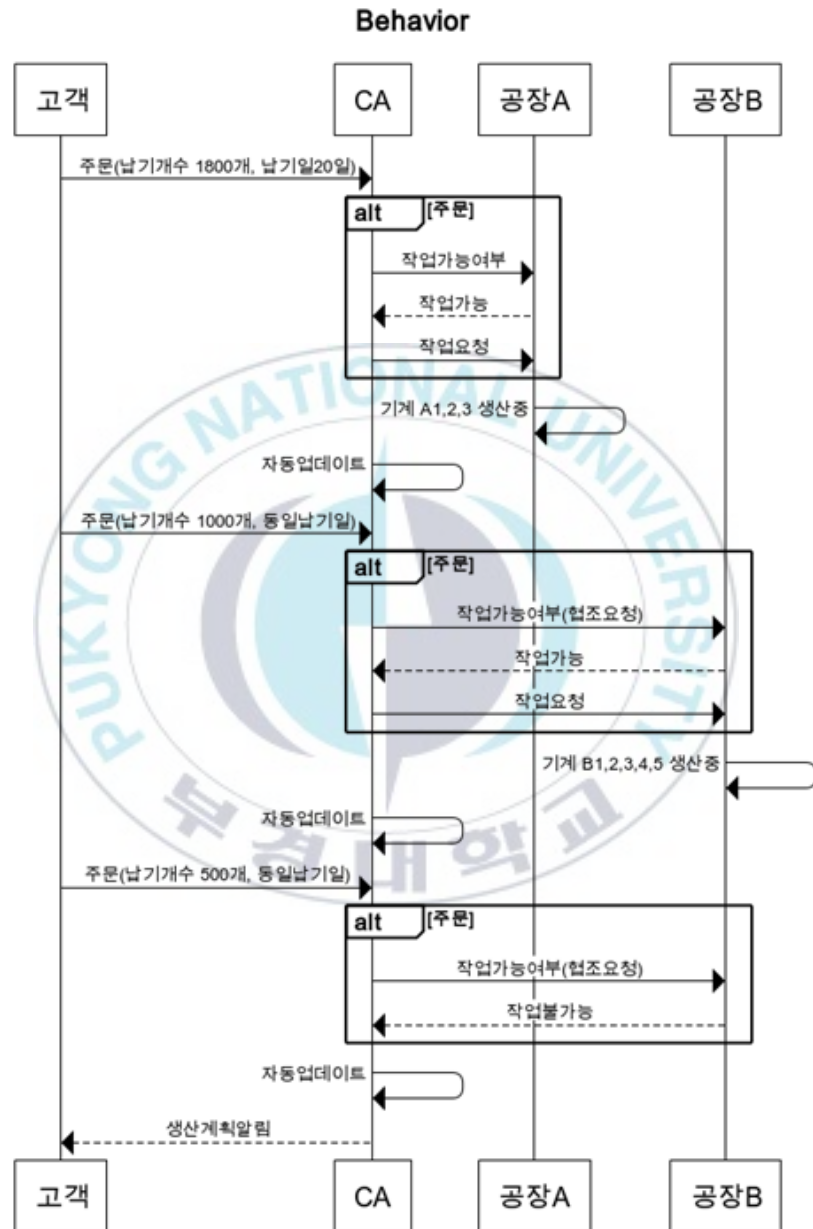
그림 3-2 JSON기반 시스템 통합 구조



3.2 행동 모델

시스템의 전반적 행위에 대해 표현하는 것이 행동 모델로 시퀀스 다이어그램으로 표현하였다. 시퀀스 다이어그램은 클래스, 구성 요소, 하위 시스템들 간 어떤 메시지를 주고받는지, 어떤 순서로 상호작용하여 발생하는지 보여준다. 다이어그램은 정보를 수직적 측면과 수평적 측면으로 전달한다. 수직 측면에서는 탑다운(top down) 방식으로 메시지/호출이 발생한 시간 순서를 나타내고, 수평측면에서는 좌측에서 우측으로 메시지가 전송된 객체 인스턴스(instance)를 보여준다. 그림 3-3 행동 모델은 Web Sequence Diagrams를 활용하여 작성하였다.

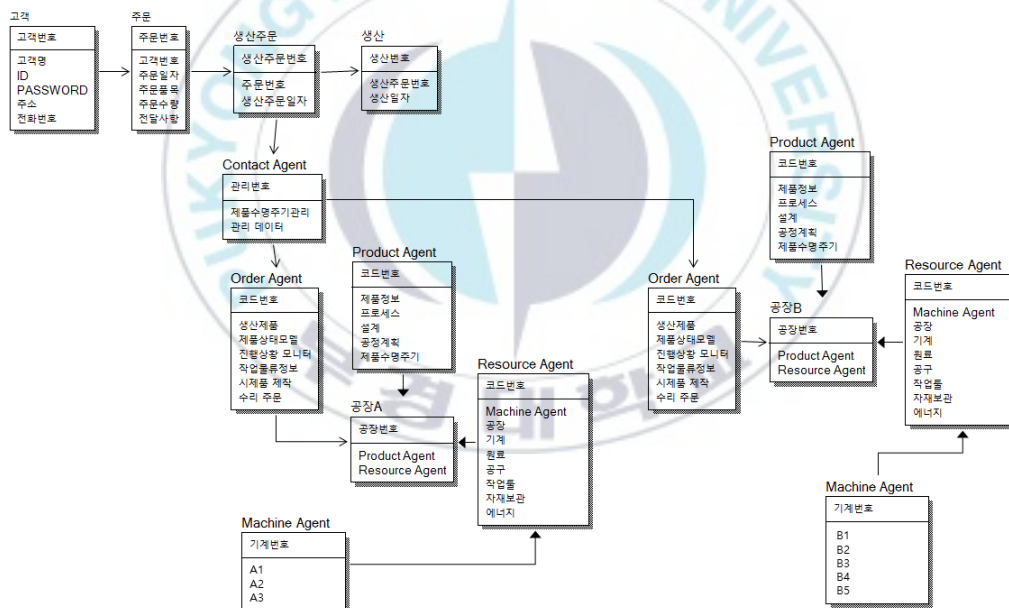
그림 3-3 행동 모델



3.3 데이터 모델

데이터 모델링이 중요한 이유는 파급효과(Leverage), 복잡한 정보 요구사항의 간결한 표현(Conciseness), 데이터 품질(Data Quality)로 정리할 수 있다. 데이터 모델은 구축할 시스템의 정보 요구사항과 한계를 가장 명확하고 간결하게 표현할 수 있는 도구이다⁸⁾.

그림 3-4 데이터 모델



검정색으로 채워진 화살표는 종속관계를 의미하고, 채워지지 않은 화살표는 데이터의 흐름을 나타낸다.

8) <http://www.dbguide.net>

각 객체의 데이터는 다음과 같다. 고객의 식별자는 고객번호로 속성은 고객명, ID, PASSWORD, 주소, 전화번호를 가지고 있다. 주문의 식별자는 주문번호, 속성은 고객번호, 주문일자, 주문품목, 주문수량, 전달사항. 생산 주문의 식별자는 생산주문번호, 속성은 주문번호, 생산주문일자. 생산의 식별자는 생산번호, 속성은 생산주문번호, 생산일자. Contact Agent(CA)의 식별자는 관리번호, 속성은 제품수명주기관리, 관리데이터. Order Agent의 식별자는 코드번호, 속성은 생산제품, 제품상태모델, 진행상황 모니터, 작업 물류정보, 시제품 제작, 수리 주문. Product Agent의 식별자는 코드번호, 속성은 제품정보, 프로세스, 설계, 공정계획, 제품수명주기. Resource Agent의 식별자는 코드번호, 속성은 Machine Agent, 공장, 기계, 원료, 공구, 작업툴, 자재보관, 에너지. 공장 A,B의 식별자는 공장번호, 속성은 Product Agent, Resource Agent. 공장A의 Machine Agent는 기계번호를 식별자로 하며 속성은 A1, A2, A3, 공장B의 Machine Agent는 기계번호를 식별자로 하며 속성은 B1, B2, B3, B4, B5를 포함하고 있다.

여기서 가장 중요한 역할을 하는 에이전트는 Contact Agent로 관리자 역할을 한다. 고객이 요청한 주문을 담당자에게 전달하며 제품수명주기관리를 한다. 제품수명주기관리는 PLM시스템의 기능요소는 7가지 주요기능으로 분류할 수 있다. 첫째, 제품구조관리(Product Structure Management), 둘째, 전자금고 및 문서관리(Vault & Document Management), 셋째, 프로젝트 관리(Product Management), 넷째, 검색 및 분류(Retrieval & Classification), 다섯째, 응용시스템 관리(Application Management), 여섯째, 응용모듈(Application Module), 일곱째, 프로세스는 제품수명주지 관점에서 제품개발에 투입되는 모든 인적자원과 개개인의 작업 프로세스를 포괄적으로 지원한다(Bae et al., 2016).

Order Agent는 모든 제품에 대한 지식을 보유하고 있다. 고객 주문서 수

정 또는 취소 요청 처리이다. 필수정보는 제품 유형, 생산기한, 수량 및 고객의 우선순위이다. 이 에이전트는 생산 중 현재 제품 주문 상태 또는 시스템 성능에 관한 정보도 수집한다.

Machine Agent는 특정 프로세스와 서비스를 제공하는 제조자원(일반적으로 기계)을 나타낸다. 'hard workers(열심히 일하는 근로자)'이기 때문에 시스템에서 중요한 역할을 한다. 이 에이전트는 자신의 특정 응용 프로그램 도메인에 대한 지식이 있다. 리소스가 제공하는 서비스는 DF(Directory Facilitator)에 등록되어 SA(Supply Agent)가 적절한 리소스를 보다 쉽게 찾을 수(parsing) 있도록 한다. 필수정보는 작업툴, 공차, 표면 품질 요구사항에 대한 지식을 수집한다(Zeichen, 2009).

Product Agent는 완벽한 품질의 제품을 정확하게 제조할 수 있도록 프로세스 및 제품지식을 가지고 있다. 제품수명주기, 사용자 요구사항, 설계, 공정계획, BOM, 품질보증절차 등 최신정보다 들어있다. 제품설계, 공정계획 및 품질 보증에서 전통적으로 다루는 기능성을 포함하고 있다. 다른 에이전트에 대한 정보 서버 역할을 한다(Brussel et al., 1998).

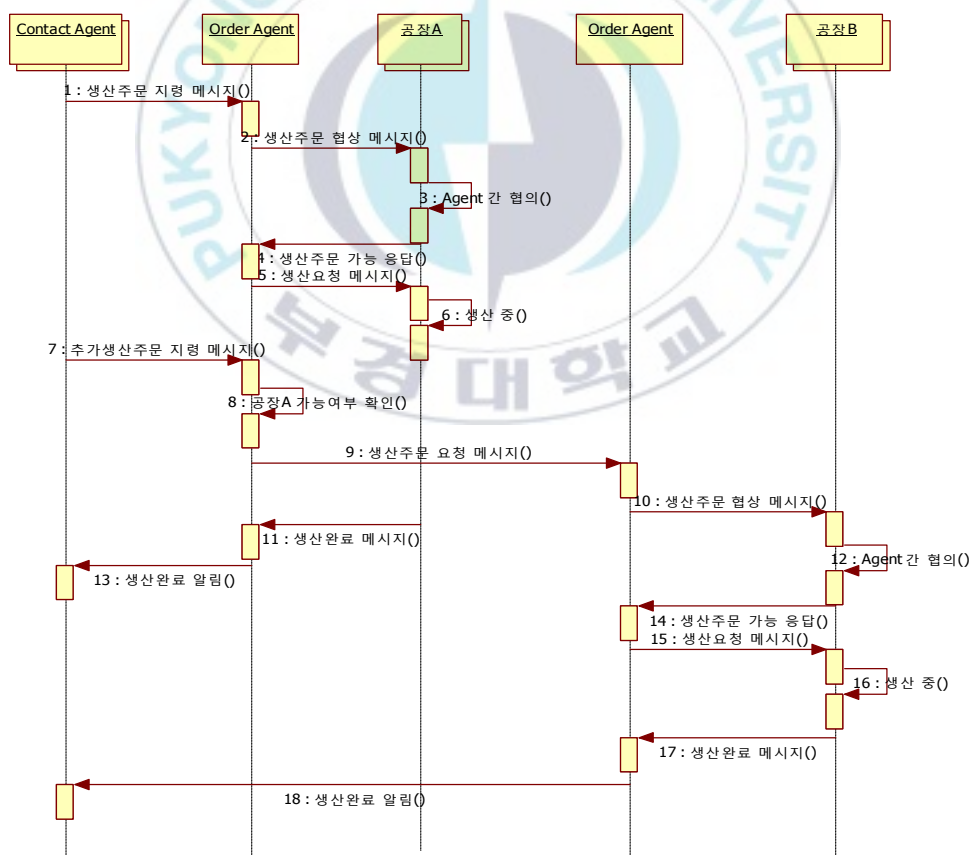
결과적으로 에이전트는 인간의 인지 행위와 유사한 기능을 수행한다. 인지 활동은 인식, 추론 및 실행과 같은 세 단계의 루프를 수행한다. 또한 인지 에이전트는 자율성, 사회적 능력, 반응 및 선행위 등의 전통적 에이전트의 모든 특성을 상속한다. 자율성은 다른 에이전트로부터 지원받지 않고 목표를 달성할 수 있도록 하는 능력이다. 반면 에이전트 간의 상호 작용은 시스템의 총괄적 목표를 달성하기 위해 사용되는 에이전트의 사회적 능력이다. 지각과 행동 사이의 관계를 기반으로 한 에이전트의 반응은 제조 환경의 변화에 대응하기 위해 요구되는 능력이며 에이전트의 선행위 능력은 목표 지향적 행위들을 나타낸다. 선행위 특성의 개선에 의해 표현되는 전통적 에이전트와 비교되는 인지 에이전트의 다른 특성은 인지 에이전트의

지능성이다. 지능은 환경 변화에 대한 적절한 결정을 위해 지식과 추론 메커니즘을 사용하는 에이전트의 능력이다(박홍석 외, 2011).

3.4 메시지 모델

행동 모델과 마찬가지로 시퀀스 다이어그램을 사용하여 나타냈다.

그림 3-5 메시지 모델



각 에이전트가 주고받는 메시지 포맷이다.

그림 3-6 FIPA기반 메시지

```
(request
  : sender(Contact Agent)
  : receiver(Order Agent)
  : language FIPA-SL
  : protocol fipa-request
  : content
    ““(( action
      (agent-identifier :Order Agent)
      (생산주문: 주문일 2017/11/04
        완료일 2017/11/24
        납기개수 1800개)
      ))””
)
```

그림 3-7 JSON기반 메시지

```
{“task”:1800,“due date”:“20일”,“from”:“CA”,“to”:“공장A”,
“message”:“작업가능여부 요청”}
```

그림 3-6은 FIPA기반 메시지 교환 시 포맷이고, 그림 3-7은 본 논문에서 제시하는 JSON기반 메시지 교환 포맷이다. 육안으로 확인했을 때 데이터 사이즈가 JSON기반의 메시지 포맷이 확실히 작음을 알 수 있다. 한 가지 메시지만 보았을 때는 큰 차이를 모를 수 있으나 제조 데이터의 경우 상당한 양이 생성되며 교환된다고 생각하였을 때 JSON기반 교환 방식이 훨씬 더 데이터 용량의 차이가 큰 것을 알 수 있다.

표 3-1은 간단히 메모장을 이용해 크기 비교를 한 표이다. 그림 3-6, 그림 3-7의 메시지 포맷을 같은 개수 메시지로 작성하였을 때 FIPA기반 메

시지의 경우 7.22KB, JSON기반 메시지의 경우 3.53KB로 2배 이상 차이가 나는 것을 볼 수 있다.

표 3-1 FIPA, JOSN기반 메시지 크기 비교

	FIPA	JSON
크기	7.22KB(7,400바이트)	3.53KB(3,620바이트)
디스크 할당 크기	8.00KB(8,192바이트)	4.00KB(4,096바이트)



IV. 시스템 구현

4.1 구현시나리오

본 장은 공장 간의 주문 내용이 변경 되었을 때의 경우에 대한 시스템 동작 구성을 설명한다. 공장A와 공장B 간의 정보교환에서의 경량화 표현을 목표로 한다.

고객은 납기개수 1800개, 납기일 20일로 주문했다. 공장A의 기계이름은 A1, A2, A3로 표기한다. 공장A에서 운영되고 있는 3대의 기계는 주문받은 동일한 파트를 가공 생산한다. 5일 뒤 갑작스레 고객은 동일 납기일내 1200개의 추가주문을 한다. 공장A내 3대의 milling machine은 이미 작업 중이므로 추가제작이 불가능하여 Contact Agent는 다른 공장B의 시스템에 자동적으로 가능성 여부를 확인 후 5대의 기계를 확인했다. 공장B의 기계 이름은 B1, B2, B3, B4, B5로 공장A의 기계들과 같은 종류의 기계이다. 단, 공장A는 기계 1대 당 하루 최대 30개 생산하며, 공장B는 기계 1대 당 하루 최대 20개까지 생산가능하다. 작업 15일째 500개의 두 번째 추가주문이 있지만 공장A, B 모두 작업 중이며 처리할 수 있는 기계가 없어 “주문을 처리할 수 없습니다.”라는 메시지를 보내온다.

4.2 구현결과

구현환경은 JAVA Eclips Neon이다. JAVA에서 JSON 사용을 위해 JSON Object Library인 'json-simple-1.1.jar' build path 설정 후 진행하였으며, Factory, Machine, Factory Management System 클래스를 생성하였다. 자바 이클립스 소스코드 입력 후 결과창인 콘솔(Console) 내용을 다음과 같이 간단히 정리하였다.

(시작)

Day:1 - {"task":1800,"due date":"20일","from":"CA","to":"공장A",
 "message":"작업가능여부 요청"}
 - {"task":1800,"due date":"20일","from":"Contact Agent",
 "to":"공장A","message":"작업 요청"}

공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --

Day:2 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --

Day:3 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --

Day:4 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --

- Day:5**
- {"task":1200,"due date":"20일","from":"CA","to":"공장A",
"message":"작업가능여부 요청"}
 - {"task":1200,"due date":"20일","from":"공장A","to":"공장B",
"message":"작업가능여부 요청"}
 - {"task":1200,"due date":"20일","from":"Contact Agent",
"to":"공장B","message":"작업 요청"}

공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

- Day:6**
- 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
- 공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

- Day:7**
- 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
- 공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

- Day:8**
- 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
- 공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

- Day:9**
- 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
- 공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

- Day:10**
- 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
- 공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

- Day:11**
- 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

Day:12 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

Day:13 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

Day:14 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

Day:15 - {"task":500,"due date":"20일","from":"CA","to":"공장A",
"message":"작업가능여부 요청"}

- {"task":500,"due date":"20일","from":"공장A","to":"공장B",
"message":"작업가능여부 요청"}

- {"task":500,"due date":"20일","from":"Contact Agent",
"to":"공장A","message":"주문을 처리할 수 없습니다."}

공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

Day:16 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --기계 생산량 B1 20/ B2 20/ B3 20/ B4 20/ B5 20

Day:17 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30

공장B 생산 --

Day:18 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
공장B 생산 --

Day:19 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
공장B 생산 --

Day:20 공장A 생산 --기계 생산량 A1 30/ A2 30/ A3 30
공장B 생산 --

본 논문에서의 구현은 부록1, 2, 3의 Factory, Machine, Factory Management System으로 나눠 소스코드를 구성한 결과를 위에서 보여주고 있다. JSON메시지를 이용한 작업지시와 응답으로 경량화된 표현방법을 보여주고 있으며, 기계간 통신에 문제가 나타나지 않아 사람이 인식하기에도 어려움이 없는 것을 알 수 있다. 공장 A, B 사이에 간단한 메시지로 이기종간 통신이 가능하며, 본 구현의 결과가 가장 합리적인 방법이라는 것을 얻었다.

V. 결론

5.1 연구요약

4차 산업혁명시대에 도래하면서 제조기반의 산업 발전이 중요시되고 있는 추세이다. 2008년 글로벌 금융위기 이후 금융산업에 많은 비중을 두었던 미국, 일본, 영국 등의 국가가 많은 어려움을 겪었지만 제조업에 비중이 높았던 독일, 중국 등의 국가는 상대적으로 적은 타격을 받으면서 재조명 받게 되었다.

한국은 제조업의 경쟁력 확보를 통해 수출주도의 성장을 지속해 왔으나 최근 국내 제조업의 설비투자 부진 및 해외이전 증가, 고유가, 원화절상 등의 영향으로 악화되고 있는 상황이다. 한국에서 제조업을 중시하는 이유는 첫째, 자원이 많이 부족한 자원빈국으로 원유와 같은 에너지 수입은 불가피하다. 둘째, 협소한 내수시장을 가지고 있는 한국은 내수시장의 한계를 극복하기 위해 제조업 위주의 수출로 성장을 도모해야한다.

현재 한국이 해결해야 할 여러 가지 문제 중 하나는 청년 실업률이다. 청년들이 원하는 ‘양질의 일자리’가 줄고 제조업의 불황으로 수출부진이 지속되어 구조조정 영향으로 제조업체의 감원이 계속되면서 2017년 10월을 기준으로 15세~29세 청년층 실업률은 8.6%로 1999년 이후 가장 높은 수치를 기록했다. 한국 제조업 경쟁력 순위는 매년 감소추세를 보이고 있어 2010년 3위, 2013년 5위, 2018년도는 6위로 예상되고 있다.

현재 세계 각국은 침체된 제조업시장의 활성화를 위해 다양한 정책을 추진 중이다. 독일의 인더스트리 4.0을 시작으로, 미국, 중국, 일본, EU 등 제

조강국들은 제조업 부흥을 통한 자국의 경쟁력 강화를 계획하고 있다. 그 중 가장 중요하다고 생각되는 것은 스마트공장의 개발, 보급, 활용이다. 4차 산업혁명의 가장 중심이 되는 스마트공장은 제조업의 기본 전략이며 제조강국으로의 발전을 도모할 수 있는 가장 핵심이 되는 부분이다.

한국도 스마트공장의 보급 확대 및 역량 강화를 위해 2017년 산업 통상자원부에서 ‘스마트 제조혁신 비전 2025’를 발표해 2025년까지 스마트공장 3만개 구축을 목표로 정책을 내놓았다. 스마트공장의 핵심기술은 사물인터넷, 빅데이터, 클라우드 컴퓨팅, CPS 등의 기술을 통해 공장 내의 장비, 장치, 부품들이 서로 연결되고 상호 소통되는 생산 체계이다. 더 나아가 내부만의 의사소통이 아닌 외부 시스템들 간의 연결도 중요한 부분이다. 이기종간의 다양한 기술 상호작용에 의해 생산효율을 높이고 생산실적 파악 등의 목적을 위해 관련 데이터 통합의 필요성이 증대하고 있다.

제조업은 센서가 부착된 기계장비에서 생산·처리되는 방대한 양의 데이터 때문에 다른 산업보다 더 많은 데이터를 생성하고 저장하여 수많은 데이터를 추출하고, 축적된 데이터를 분석해 기계와 사람 또는 기계와 기계의 대화가 가능하도록 한다. 자율적, 협업적 시스템 개발을 위해 계층적 구조는 상위요소가 하위요소를 엄격히 통제해 운영상 안전하지만 자율성의 거의 없어 민첩한 대응이 부족하며 수평적 구조는 시스템 제어의 복잡성이 적어 환경 변화에 빠른 대처가 가능하고 정보처리도 신속히 이루어지지만 제어요소가 많아 분산되고 자율화되어 통일된 시스템운영이 어렵고 안전성을 보증하지 못해 전체 시스템 목표의 최적화는 실현되기 어렵다. 그렇기 때문에 장점은 흡수하고 단점은 보완한 홀로닉제조시스템이 필요하다.

스마트공장의 시대가 오면서 대부분의 업무가 이기종간의 통합운용과 데이터 교환에 중점을 두어 연구할 것이다. 본 논문은 이기종 간 에이전트간의 자율적이고 협업적인 의사결정을 목표로 경량화된 데이터 전달 방식

이 JSON을 이용하였다. 이기종 및 에이전트의 상호작용을 위해선 FIPA 프로토콜을 이용하지만 FIPA 프로토콜의 호환이 되지 않는 경우는 사람과 기계 모두 읽고 쓰기 용이하며 프로그래밍 언어 및 OS에 독립적으로 상이한 시스템간의 객체 교환이 가능한 경량화된 표현방식인 JSON의 방식을 채택해 데이터교환 방식의 범용성을 확보하고자 하였다.

5.2 연구공헌

현대사회에 있어 제조업이 필수적인 근간을 제공하는 만큼, 제조업의 성장과 중요성을 간과할 수 없다. 4차 산업혁명의 시대가 오면서 기술발전으로 인해 내 일자리가 사라지는 것은 아닌지, 기술에 의지하는 일이 많아지면서 기계에게 주객전도되는 것은 아닌지 등의 염려를 하게 된다. 하지만 대다수 전문가들은 사라지는 직업만큼 새로운 직업도 등장할 것이라고 예측하고 있다. 인공지능은 간단한 노동을 하게 하는 데도 상당한 시간과 비용을 들이는 연구개발이 요구된다. 맞춤형 생산과 소비의 문화와 더욱 보급될 것으로 많은 이들이 전망하고 있어 인공지능을 원활히 쓰기 위해서라도 인간의 창의력 같은 고도의 능력은 반드시 필요하다. 아직까진 일자리 위협을 염려하기엔 이르지만 4차 산업혁명은 곧 일자리 감소라는 공식은 계속 되어 왔을 만큼 4차 산업혁명의 특징이 되었다.

제조업의 발전이 경제적, 사회적 문제에 다양한 파급효과를 가져올 수 있다. 제조업은 센서가 부착된 기계장비에서 생산·처리되는 방대한 양의 데이터 때문에 다른 산업보다 더 많은 데이터를 생성하고 저장하여 수많은 데이터를 추출하고, 축적된 데이터를 분석해 기계와 사람 또는 기계와 기

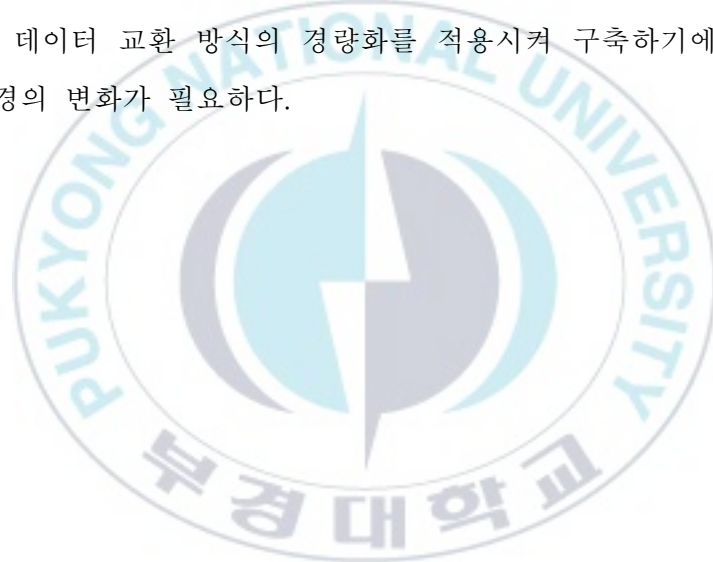
계의 대화가 가능하도록 한다. 시간이 지남에 따라 데이터의 양은 더 늘어날 것이고 데이터의 활용은 방대하고 다양하게 이루어질 것이다. 데이터는 스마트 제조에 있어 항상 존재하는 불가결한 실재임에는 틀림이 없다. 본 연구가 제조 시스템간의 데이터 표현의 간결함과 개방성, 호환성에 기본을 둔 스마트공장의 제조혁신에 기여한다.

5.3 한계점 및 극복방안

본 연구는 제조 객체 간의 자율적이고 협업적인 의사결정을 위해 JSON을 활용한 정보교환을 목적으로 하고 있어 데이터의 경량화와 교환 방식에 대한 측면이 기여한다. 하지만 현재의 구현이 미흡하여 실제 공장에서 사용되기에는 많은 어려움이 있으며 데이터 모델의 구성과 시나리오와 맞지 않는 부분은 추후 연구를 통해 극복해 나갈 부분이다. 자율적인 의사소통을 위해 이기종간의 연구가 불가피하여 이기종 프로그래밍 환경에서의 데이터 교환 구현 및 검증이 필요하다.

국내 스마트공장 시범사업 결과로 16년 기준 2,800여개 사의 스마트공장 구축을 지원했으며 그 중 2,300여개 사의 구축이 완료되었지만 대부분은 '기초' 이하 단계이다. 스마트공장의 수준별 구축형태는 기초, 중간1, 중간2, 고도화로 나뉜다. 현재 한국의 경우 기초수준 82.3%, 중간1 12.4%, 중간2 3.1%로 기초수준에 머물러 있는 곳이 많으며, 고도화 기업은 거의 없다. 기초는 바코드, RFID를 기초적 물류정보 수집 수준으로 사용하여 공정물류 중심의 실적관리의 수준이 가능하다. 중간1은 설비로부터 실시간 데이터를 수집하여 집계된 실적 중심의 공장 운영 분석 후 실시간 정보교류를

추진한다. 중간2는 설비 제어 자동화가 가능하여 실시간 의사결정 및 설비 직접제어로 시장과 고객 요구에 능동적으로 대응한 실시간 의사결정 및 통제 가능하다. 고도화는 제어자동화 및 데이터 식별이 결합된 IoT형 자동화로 CPS, IoT, 빅데이터를 이용한 자가진단과 제어능력을 갖춘 지능형 생산이 가능해 가치사슬 연계를 통한 실시간 고객 맞춤 서비스가 구축되는 단계이다⁹⁾. 본 연구는 고도화가 수반되었을 때 비로소 그 가치를 발휘할 수 있을 것으로 현재의 구축 형태로는 한계가 있다. 한국의 스마트공장이 아직 기초 단계에 머물러있기 때문에 효과 면에서 스마트공장의 시스템 통합을 위한 데이터 교환 방식의 경량화를 적용시켜 구축하기에는 더 많은 연구와 환경의 변화가 필요하다.



9) <http://www.smart-factory.kr>

참 고 문 헌

강서진(2014), 한국 GDP 구성의 특징과 과제, KB금융지주경영연구소

강지원(2015), 중국제조 2025, 산업연구원, 산업정책해설, 29-39

김병주, 허성민, 최경현, 이석희(1999), 홀로닉 개념을 이용한 통합 자동화 시스템의 구축, 한국정밀공학회 학술발표대회 논문집, 422-425

김상현, 이정혁, 오창세, 서민석, 김영돈, 이창석, 박현주, (2014), IoT 서비스 지능화를 위한 JSON 방식의 센싱 데이터 처리, 한국정보과학회 학술발표논문집, 316-318

김선재(2017), 4차 산업혁명 대응을 위한 스마트공장 R&D 현황 및 시사점, 한국과학기술평가원

김수현(2012), 양적 지표 개선에도 고용의 질적 수준 우려 [2012 하반기 전망 ②] 노동시장 전망, 새로운사회를여는연구원

김현수, 천승만, 정윤석, 박종태(2016), 사물인터넷 헬스케어 서비스를 위한 oneM2M 기반 ISO/IEEE 11073 DIM 전송 구조 설계 및 구현, 전자공학회논문지 53(4), 3-11

김현철(2015), RFID를 이용한 제조공정 이력관리 시스템, 석사학위논문, 호서대학교, 46 pages

김희정(2016), “JSON 기초 프로그래밍”, 휴먼코딩

노선택, 이근상, 문석재, 엄영현, 정계동, 최영근(2007), MAEMS : 멀티 에이전트 기반 응급 의료 시스템 모델링, 한국정보과학회 학술발표논문집, 34(1)

롤랜드버거(2017), 4차 산업혁명 이미 와 있는 미래, 다산 3.0

박형욱(2015), 스마트 팩토리 와 연관된 생산제조기술 동향, 한국통신학회지, 33(1), 24-29

박홍석, 박진우, Tran Ngoc-Hien, 김동훈, 송준엽(2011), 진화형 제조시스템 실현을 위한 생체 모방 기술, 전자공학회지 38(3), 43-47

배경한, 이진성, 송병준, 이광기, 김용국(2017), 스마트공장 보급 · 확산사업 현황, 한국산업기술평가관리원

산업통상자원부(2017), 4차 산업혁명 선도하는 스마트공장 올해까지 5,000개 보급

신승준, 우정엽, 서원철(2016), 스마트공장을 위한 빅데이터 애널리틱스 플랫폼 아키텍처 개발, Journal of Korea Multimedia Society, 19(8), 1516-1529

아주경제(2015), ‘제조업 30% 법칙’ 발목 잡힌 한국 제조업 ‘퍼스트무버 전략’ 극복해야, 아주경제 홈페이지,

<http://www.ajunews.com/view/20150429141527536>

양태호, 윤태혁, 최상수, 노상도(2008), 상용 PDM을 포함한 이기종 시스템 간의 제품정보 교환에 관한 연구, 한국CAD/CAM학회 논문집, 13(3), 175-186

이규택, 이진재(2015), 스마트공장 R&D로드맵 소개, 한국산업기술평가관리원

이상일, 류광열, 신문수, 김보현, (2013), 국내 금형산업을 위한 제조서비스 모델 개발, 한국CACAM학회 2013학술발표회 논문집, 216-222

이성준, 김춘경, 이재덕(2017), OPC UA를 이용한 스마트센서 시스템 개발, 2017년 대한전기학회 학술대회 논문집, 12-14

이영해, 김정(1997), 홀로닉 생산시스템의 개요 및 연구방향, 제어로봇시스템학회지, 3(1), 33-40

이용구(2017), 스마트 팩토리, 2017년 대한기계학회 기계저널, 57(8), 26-53

이정아(2015), 사이버물리시스템(CPS) 기반의 사회시스템 최적화 전략, NIA 한국정보화진흥원

이현정, 김용진, 임정일, 김용운, 이수형(2017), 스마트공장 구축을 위한 현장실태 및 요구사항 분석, 한국정밀공학회지 34(1), 29-34

이호석, 신상훈, 박철순(2006), 제조 프로세서에서 XML기반의 데이터 통합 및 활용방안, 한국산업경영시스템학회 2006 춘계학술대회논문집, 11-12

임선중, 송준엽(2007), JADE를 이용한 에이전트 구축과 응용, 한국정밀공학회 학술발표대회 논문집, 411-412

장영재(2012), 제조 분야에서의 빅데이터 기술 활용, 한국통신학회지, 30-35

전은경(2016), 4차 산업혁명의 도래와 대응방안, 국회입법조사처

전형국, 이수형, 전인결(2017), 사이버-물리 생산 시스템 연동 미들웨어 프레임워크, 2017년 한국통신학회 하계종합학술발표회, 618-619

조선비즈(2017), '제조업 취업자 1년 만에 첫 증가했지만, 청년 실업률 최고치, 조선비즈 홈페이지,

http://biz.chosun.com/site/data/html_dir/2017/07/12/2017071201021.html

주용하(2001), XML의 탄생 배경과 활용, CAD&Graphics, 116-119

한국기록학회(2008), 기록학 용어 사전, 역사비평사

헬로티(2016), [제조혁신 이끄는 스마트CPS] CPS는 지능형 시스템 구축 기술 ... 가상-리얼 연동한 공장 구현, 헬로티 홈페이지,

http://www.hellot.net/new_hellot/magazine/magazine_read.html?code=201&sub=002&idx=29783

Babiceanu, R. F. & Chen, F. F.(2006), Development and applications of Holonic manufacturing systems: A survey, Journal of Intelligent

Manufacturing, 17, 111-131

Babiceanu, R. F., Chen, F. F. & Sturges, R. H.(2004), Framework for the control of automated material-handling systems using the holonic manufacturing approach, International Journal of Production Research, 42(17), 3551-3564.

Bae, J. K. & Kim, S. C.(2016), A Study on Efficient Design and Construction Plan of Product Life-cycle Management Systems: Focused on PLM System of Korean D Company, The e-Business Studies, 17, 217-239

Balilai, O. F. & Houshmand, M.(2013), A collaborative and intergrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm, Robotics and Computer-Integrated Manufacturing, 110-127

Barbosa, J., Leitao, P., Adam, E. & Trentesaux, D.(2015), Dynamic self-organization in Holonic multi-agent manufacturing systems: The ADACOR evolution, Computers in Industry, 66, 99-111.

Brennan, R. W., Fletcher, M. & Norrie, D. H.(2002), An agent-based approach to reconfiguration of real-time distributed control systems, IEEE Transactions on Robotics and Automation, 18(4), 444-451.

Brussel, H. V., Wyns, J., Valckenaers, P., Bongaerts, L. & Peeters, P.(1998), Reference architecture for holonic manufacturing systems: PROSA, Computers in Industry, 37, 255-274

Colombo, A.W., Schoop, R. & Neubert, R., (2006), An agent-based intelligent control platform for industrial Holonic manufacturing systems, IEEE Transactions on Industrial Electronics, 53(1), 322-337.

Dagkakis, F., Carie, G., Greenwood, D.(2007), Developing Multi-Agent Systems with JADE, John Wiley & Sons

Dagkakis, G., Heavey, C., Robin S. & Perrin, J.(2013), ManPy: An Open-Source Layer of DES Manufacturing Objects Implemented in SimPy, EUROSIM Congress on Modelling and Simulation, 357-363

F.Bellifemine, G., Caire, A. Poggi. & G. Rimassa.(2013), JADE - A White Paper, exp, 3(3), 6-19

Gamboa Quintanilla, F., Cardin, O., L'Anton, A. & Castagna, P.(2016), A petri net-based methodology to increase flexibility in service-oriented holonic manufacturing systems, Computers in Industry, 76, 53-68

Gou, L., Luh, P. B. & Kyoya, Y.(1998), Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation, Computers in Industry, 37, 213-231

<http://jade.tilab.com> JADE

<http://www.dbguide.net> 데이터 전문가 지식포털

<http://www.fipa.org> FIPA

<http://www.smart-factory.kr> 스마트공장 추진단 홈페이지

Indriago, C., Cardin, O., Rakoto, N., Castagna, P. & Chacon, E.(2016), H²CM: A Holonic architecture for flexible hybrid control systems, Computers in Industry, 77, 15-28.

Kang, H. S., Lee, J. Y., Choi, S. S., Kim, H., Park, J. H., Son, J. Y., Kim, B. H. & Noh, S. D.(2016), Smart Manufacturing: Past Research, Present Findings, and Future Directions, International Journal of Precision Engineering and Manufacturing0Green Technology, 3(1), 111-128

Korambath, P., Wang, J., Kumer, A., Hochstein, L., Schott, B., Graybill, R., Baldea, M. & Davis, J.(2014), Deploying Kepler Workflows as Services on a Cloud Infrastructure for Smart Manufacturing, Procedia Computer Science, 29, 2254-2259

Lee, J., Ardakani, H. D., Yang, S. & Bagheri, B.(2015), Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. Procedia CIRP, 38, 3-7

Lee, J., Bagheri, B. & Kao, H. A.(2015), A Cyber-Physical Systems Architecture for Industry 4.0-based manufacturing systems, *Manufacturing Letters*, 3, 18-23

Leitao, P.(2011), A holonic disturbance management architecture for flexible manufacturing systems, *International Journal of Production Research*, 49(5), pp.1269-1284

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. & Byers, A. H.(2011), Big data: The next frontier for innovation, competition, and productivity, McKinsey Global Institute, 1-143

Marinho da Silva, R., Blos, M.F., Junqueira, F., Santos Filho, D. J. & Miyagi, P. E.(2014), A service-oriented and holonic control architecture to the reconfiguration of dispersed manufacturing systems. In *Proceedings of 2014 IFIP Advances in Information and Communication Technology*, 111-118

Monostori, L.(2014), Cyber-physical production systems: Root, expectations and R&D challenge, *Procedia CIRP* 17, 9-13

Nam, S. H., Baek, J. Y., Yoon, J. S., Lee, D. H., Kwon, K. E. & Lee, S. W.(2013), Realtime Manufacturing Data Management Framework for Dynamic Shop Operation in Heterogeneous Machine and Cell Shop Configuration, *Korean Society Of Precision Engineering*, 1469-1470

Nurseitov, N., Paulson, M., Reynolds, R. & Izurieta, C.(2009), Comparison of JSON and XML Data Interchange Formats: A Case Study, Proceedings of the ISCA 22nd International Conference on Computer Applications in Industry and Engineering, 157-162

Owliya, M., Saadat, M., Jules, G.G., Goharian, M. & Anane, R.(2013), Agent-based interaction protocols and topologies for manufacturing task allocation, IEEE Transactions on Systems, Man, and Cybernetics: System, 43(1), 38-52

Patrizio, A.(2016), XML을 위한 견배, JSON이여 영원하라, CIO홈페이지
<http://www.ciokorea.com/news/30044#csidx7a5c9bf0092b694898334497c4e2cdf>

Poslad, S., Buckle, P. & Hadingham, R.(2000), The FIPA-OS agent platform: Open Source for Open Standards. Proceedings of 5th Int, Conf on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM), 355 - 368

Rauschmayer, A.(2012), The Past Present and Future of JavaScript, O'Reilly

Shen, W., Maturana, F. & Norrie, D. H.(2000), MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing, Journal of Intelligent Manufacturing, 11, 237-251

Shi, J., Wan, J., Yan, H. & Suo, H.(2011), A Survey of Cyber-Physical

Systems, Wireless Communications and Signal Processing, Wireless Communications and Signal Processing, 9-11

Vander, A. D.(2013), JSON's Eight Year Convergence With XML, ProgrammableWeb

<https://www.programmableweb.com/news/jsons-eight-year-convergence-xml/2013/12/26>

Yoon, J. S., Nam, S. H., Baek, J. Y., Kwon, K. E., Lee, D. H. & Lee, S. W.(2013), Process Management Systems for Integrated Real-Time Shop Operations in Heterogeneous Multi-Cell Based Flexible Manufacturing Environment, Journal of the Korean Society of Manufacturing Technology Engineers, 22(2), 281-286

Zeichen, G.(2009), Knowledge-based Multi-Agent Architecture Applied in the Assembly Domain, Master's Thesis, Fakultät für Elektrotechnik und Informationstechnik, Vienna, Austria, 118 pages

부록

1. Factory 소스코드

```
package factory;
import java.util.LinkedHashMap;
import java.util.Map;

import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;

import Config.SystemConfig;

public class Factory {

    //공장의 이름
    private String factoryName;

    //공장의 머신목록
    private Map<Object,Machine> machineList = new LinkedHashMap();

    //다른공장과 통신을 위한 다른공장의 목록
    private Map<Object,Factory> factoryList;

    //이름, 머신이름, 생산시간
    public Factory(String name, String[] machineNames, int[] speed,Map
factory){
        this.factoryName = name;
        for(int i=0; i<machineNames.length;i++){

            machineList.put(machineNames[i],
                            new Machine(
                                machineNames[i],
                                speed[i])
                            );
        }
        this.factoryList = factory;
    }

    public void addTask(String message) throws Exception {
        System.out.println(" - " + message);
    }
}
```

```

JSONParser parser = new JSONParser();
JSONObject jsonObj = (JSONObject) parser.parse(message);

int task = (int)(long) jsonObj.get("task");
int amount = task / machineList.size();

Object[] keys = machineList.keySet().toArray();

for(int i=0;i<keys.length;i++){
    machineList.get(keys[i]).addTask(amount);
}
}

public boolean canWork(String message) throws Exception{

    System.out.println(" - " + message);
    JSONParser parser = new JSONParser();
    JSONObject jsonObj = (JSONObject) parser.parse(message);

    int task = (int)(long) jsonObj.get("task");

    Object[] keys = machineList.keySet().toArray();
    int amount = 0;
    for(int i=0;i<keys.length;i++){
        amount += machineList.get(keys[i]).getAmount();
    }

    return amount < SystemConfig.maximunTaskTime ? true :
false;
}

public String getIdleFactory(String message) throws Exception {
    JSONParser parser = new JSONParser();
    JSONObject msg = (JSONObject) parser.parse(message);

    //System.out.println(message);
    String idleFactoryName = "";

    Object[] keys = factoryList.keySet().toArray();

    for(int i=0;i<keys.length;i++){

```



```

        if(factoryName.equals(keys[i])) {

            continue;
        }

        JSONObject jsonObj = new JSONObject();
        jsonObj.put("from", this.factoryName);
        jsonObj.put("to", (String)keys[i]);
        jsonObj.put("task", msg.get("task"));
        jsonObj.put("message", "작업가능여부 요청");
        //jsonObj.put("day", msg.get("day"));
        jsonObj.put("due date", "20일");

        boolean canWork =
factoryList.get(keys[i]).canWork(jsonObj.toJSONString());
        if(canWork) {
            idleFactoryName = (String)keys[i];
            break;
        }
    }
    return idleFactoryName;
}

public void doOperate() {

    System.out.println("\t공장 " + factoryName + " 생산 ---");
    Object[] keys = machineList.keySet().toArray();

    for(int i=0;i<keys.length;i++){
        machineList.get(keys[i]).work();
    }

}

@Override
public String toString() {
    return "Factory [factoryName=" + factoryName + ",
machineList=" + machineList + "]";
}

public int getNumOfMachine() {
    return machineList.size();
}
}

```

2. Machine 소스코드

```
package factory;

import java.util.LinkedList;
import java.util.Queue;

public class Machine {

    //설비 이름
    private String machineName;
    private int speed;
    private int numOfTask;

    public void work() {
        if(numOfTask == 0)
            return;

        if(numOfTask - speed > 0) {
            numOfTask -= speed;
            System.out.println("\t\t기계 " + machineName + "
생산량" + speed);
        }else {
            System.out.println("\t\t기계 " + machineName + "
생산량" + numOfTask);
            numOfTask = 0;
        }
    }

    public Machine(String name,int speed){
        this.machineName = name;
        this.speed = speed;
        this.numOfTask = 0;
    }

    public int getAmount() {
        return numOfTask * speed;
    }

    public String getMachineName() {
        return machineName;
    }
}
```

```
public void setMachineName(String machineName) {  
    this.machineName = machineName;  
}  
  
public void addTask(int amount) {  
    numOfTask += amount;  
}  
  
public boolean hasTask() {  
    return numOfTask != 0;  
}  
}
```



3. Factory Management System 소스코드

```
package system;
import java.util.LinkedHashMap;
import java.util.Map;

import org.json.simple.JSONObject;

import factory.Factory;

public class FactoryManagementSystem{

    Map<Object,Factory> factoryList = new LinkedHashMap();
    private String firstFactory;

    //공장들, 공장설비들 초기화
    public FactoryManagementSystem(){
        String[] factoryList = {"FACTORY-A","FACTORY-B"};

        String[][] machineList = {
            {"A1","A2","A3"},
            {"B1","B2","B3","B4","B5"}
        };

        int[][] amountList = {
            {30,30,30},
            {20,20,20,20,20}
        };

        firstFactory = factoryList[0];
        for(int i=0;i<factoryList.length;i++)
            this.factoryList.put(factoryList[i],
                                new Factory(factoryList[i],
                                                machineList[i],
                                                amountList[i],
                                                this.factoryList
                                ));
    }
}
```

```

public void start() throws Exception{

//System.out.println(factoryList.get("FACTORY-A").getNumOfMachine());
    Integer[][] order = {
        {1,1800},
        {5,1200},
        {15,500}
    };

    System.out.println("시작");
    for(int t=1;t<100;t++) {
        System.out.println("day : " + t);
        int task = 0;
        for(int i=0;i<order.length;i++) {
            if(t == order[i][0]) {
                task = order[i][1];
            }
        }
        if(task != 0) {
            JSONObject jsonObj = new JSONObject();
            //jsonObj.put("day", t);
            jsonObj.put("from", "Contact Agent");
            jsonObj.put("to", firstFactory);
            jsonObj.put("task", task);
            jsonObj.put("message","작업가능여부 요청");
            jsonObj.put("due date","20일");

            String possibleFactory = firstFactory;
            boolean canWork =
factoryList.get(firstFactory).canWork(jsonObj.toJSONString());

            if(!canWork) {
                possibleFactory =
factoryList.get(firstFactory).getIdleFactory(jsonObj.toString());
            }

            if("".equals(possibleFactory)) {
                jsonObj.put("message", "주문을
처리할 수 없습니다.");
            }
            System.out.println(" - " + jsonObj);
        }else {

```

```

        jsonObj.put("message" ,"작업 요청");
        jsonObj.put("to",possibleFactory );

factoryList.get(possibleFactory).addTask(jsonObj.toJSONString());
    }
}

Object[] keys = factoryList.keySet().toArray();

for(int i=0;i<keys.length;i++){
    factoryList.get(keys[i]).doOperate();
}

}

}

```

