

# 유연생산 시스템에서의 에이전트를 이용한 동적 작업배정규칙 할당에 관한 연구

김승호<sup>1</sup>, 최회련<sup>1</sup>, 이흥철<sup>1\*</sup>  
<sup>1</sup>고려대학교 정보경영공학전문대학원 정보경영공학과

## Dynamic manufacturing scheduling using multi-agent-system in FMS

SeungHo Kim<sup>1</sup>, HoeRyeon Choi<sup>1</sup> and HongChul Lee<sup>1\*</sup>

<sup>1</sup>Department of Information Management Engineering, Korea University

**요 약** 유연생산시스템은 고자동화 생산시스템일 뿐만 아니라, 사용자와 시장의 요구를 만족시키기 위해 유연한 생산시스템이 요구됨에 따라 작업배정규칙 할당에 있어서 시스템과 작업환경에 많은 제약을 받게 된다. 본 논문에서는 온톨로지 지식 기반의 작업환경을 고려한 유연생산시스템의 작업배정규칙 할당방법을 제시한다. 에이전트를 이용하여 작업장 환경과 작업내용을 모니터링하고, 이에 적합한 작업배정규칙을 온톨로지 지식으로부터 추론하여 시스템에 적용한다. 본 시스템 개발을 위해 멀티에이전트를 지원하는 JADE™ 플랫폼을 이용하였으며, 온톨로지는 Protege를 이용하여 개발되었다.

**Abstract** As flexible manufacturing systems (FMSs) are highly automated and required flexibility to satisfy market need, dispatching rules are constrained by specific environments of manufacturing systems such as mechanical failures, absence of employees, and lack of spare parts. In this paper, an ontology-based knowledge approach is proposed to improve efficiency of system through adapting suitable dispatching rules, considering context in a FMS shop floor, which consists multiple manufacturing cells. the multi-agents monitor manufacturing system status and job so that it figures out a dispatching rule considering context. To demonstrate the proposed approach, a proof-of-concept prototype system has been implemented in the JADE™ platform and Protégé to make OWL DL ontology.

**Key Words** : Flexible manufacturing system, Agent, Scheduling, Dispatching rule, Ontology

### 1. 서론

현대의 제조업은 짧은 제품 수명주기, 불확실한 수요, 혁신적인 제품생산방식 등의 내부적인 요인과 소비자의 요구 즉, 낮은 가격, 개인화된 주문 생산, 요구에 대한 빠른 응답 등의 외부적인 요인을 동시에 만족하게 해야 한다[1].

유연생산시스템(Flexible Manufacturing System: FMS)은 NC 머신과 물류운용시스템, 그 외 외부의 생산 장비들을 컴퓨터로 통합하여 컨트롤러하는 방식으로 제품설계,

생산 공정 및 생산수량의 변화에 실시간으로 대응할 수 있는 장점을 가지고 있다. 또한, FMS는 생산시스템의 유연성과 높은 기계 이용률, 낮은 단위당 생산비용, 작업시간의 단축 등의 특징을 가지는 시스템으로 현대 제조 시스템이 갖춰야 할 요구 사항을 가장 잘 반영하는 생산방식 중 하나이다[2]. 그러나, FMS는 조립라인이나 job shop 생산방식에 비해 작업장 환경이 복잡하여 혼류생산 방식에 따른 작업타입의 변화, 톨의 변경 등과 같이 예측 가능한 변화뿐만 아니라, 근로자의 장기결석, 기계고장/수리, 납기일의 변경 등과 같이 예측 불가능한 작업장의

본 논문은 지식경제부 국가플랫폼기술개발사업의 c-MES 공정지원 플랫폼 기술 연구과제로 수행되었음.

\*교신저자 : 이흥철(hclee@korea.ac.kr)

접수일 10년 06월 30일

수정일 (1차 10년 08월 04일, 2차 10년 08월 26일)

게재확정일 10년 09월 08일

환경 변화가 시스템에 미치는 영향이 다른 생산방식에 비해 크다고 할 수 있다. 따라서 작업장의 환경 변화에 따라 작업배정규칙(Dispatching rule)을 달리하는 동적 일정계획이 연구되어왔다.

FMS 환경에서의 동적 일정계획의 기존연구를 살펴보면 Solot[3]은 일정계획문제를 입력순서, 공정순서, 기계선택, 부품 선택, 운반 선택 문제로 구분하여 전문가 시스템과 OR기법을 사용하여 온라인 실시간 방식으로 문제 해결을 위한 구조를 제시하였으나 구체적인 해결 방법은 제시하지 못하였다.

Ishii 등 [4]은 특정 시점의 작업장의 프로세스 타임, 작업 대기시간, 납기일 등으로 이루어진 Index 값을 계산하여 실시간으로 작업배정규칙을 할당하는 알고리즘을 제시하였으나 작업장의 상태변화를 측정하는 수리식이 전체 작업장의 평균값을 사용하고 있어 각 작업장의 상태 변화에 둔감한 단점이 있다.

Lee 등 [2]은 job 특성을 RPJ(Rate of Prior Job), RLJ(Rate of Large Job), RSJ(Rate of Shortest due date Job), JIT(Job Interval Time)와 같이 4가지로 분류하고 작업장의 환경변화가 일어날 때 job특성을 계산하여 시물레이션을 통해 사전에 도출된 작업배정규칙 테이블과 비교, 최적의 작업배정규칙을 할당하는 방법을 제시하였으나 job특성 즉, 작업장 환경변화 감지에 배치(Batch)평 균을 이용하므로 변화에 민감하게 대응할 수 없는 문제점이 있다.

이러한 기존 연구들은 공통으로 작업장의 환경변화 감지에 둔감한 문제점을 가지고 있다. 따라서 최근 위의 문제점을 해결하기 위해 작업장의 환경을 실시간으로 감지할 수 있는 에이전트(Agent)를 활용한 동적 일정계획에 대한 연구가 진행되고 있다. 관련연구를 살펴보면 Shen[5]은 에이전트를 활용한 동적 일정계획 방법을 제시하였으나 문제해결방법이 기존에 소개된 일반적인 동적 일정계획 방법론에서 크게 벗어나지 못하였다.

Cheeseman 등 [2]은 CFP(Call For Propose)를 이용하여 에이전트간의 경쟁을 통해 최적의 조건을 제시하는 에이전트에게 작업을 배정하는 일정계획 방법을 제시하였으나, 조건에 따른 에이전트 선택 알고리즘 없이 가장 짧은 작업 시간을 제시하는 에이전트를 선택하는 방법으로 작업조건 및 기타 작업장 상황을 고려한 작업 할당이 불가능하다.

최근 에이전트를 이용한 동적 일정계획관련 연구는 에이전트에 최적해 탐색 알고리즘을 내장하여 작업조건에 따라 최적의 에이전트를 선택하는 동적 일정계획 방법이 다양하게 연구되고 있다. Wang 등[7]은 FBS(Filtered-Beam-Search), Safiye[8]는 Petri nets 를 이용해 최적의 에

이전트를 찾아 작업을 할당하는 방법을 제시하였다.

위의 에이전트를 이용한 기존의 동적 일정계획 연구들은 작업장의 변화하는 시점을 민감하게 감지할 수 있으나, 최적의 일정계획을 찾기 위해 특정한 발견적 방법(Heuristic Algorithm)을 이용하므로 탐색환경, 탐색조건 등 시스템의 환경이 변하게 되면 해당 발견적 방법이 변화된 시스템에서도 일정계획을 위한 최적의 발견적 방법이라 말할 수 없어, 방법론 범용성이 떨어진다.

따라서 본 연구에서는 작업의 형태와 작업장의 상황 변화에 유연하게 대처할 수 있는 동적 작업 할당 방법론을 적용한 범용적이고, 확장 가능한 시스템을 제안한다. 제안하는 시스템은 에이전트를 이용하여 작업장의 환경 변화를 실시간으로 감지하고 작업시간, 작업일정, 작업우선순위 등의 작업장의 작업 데이터를 수집하여 실시간 내에 유연하게 대처할 수 있는 시스템의 기반을 마련한다. 수집된 데이터를 이용하여 온톨로지 기반(Ontology-based)의 작업배정규칙(Dispatching Rules) 지식을 생성하여 변화된 작업 환경에 적합한 작업배정규칙을 실시간으로 추론, 적용함으로써 시스템의 범용성과 확장성을 높인다. 본 연구에서 제안하는 시스템은 작업장의 변화를 실시간으로 감지하고 작업 데이터의 수집을 위해 MAS(Multi-Agent-System)를 지원하는 JADE™ [9]플랫폼 기반의 에이전트 시스템을 이용하여 개발하였다. 또한, 동적 작업할당을 위한 작업배정규칙 지식은 Protégé[10]를 이용하여 온톨로지 기반으로 작성되어 지식의 공유, 재사용, 수정 등의 특징을 갖는다.

본 연구의 이후 진행은 2장에서 멀티에이전트 시스템을 설명하고, 3장에서는 온톨로지와 온톨로지를 이용하여 FMS 동적 작업할당에 대해 기술한다. 4장에서는 제안된 시스템의 구현환경과 실험 결과를 설명하고, 마지막 장에서는 연구의 결론 및 향후 연구되어야 할 부분에 대해 기술한다.

## 2. 멀티 에이전트 시스템

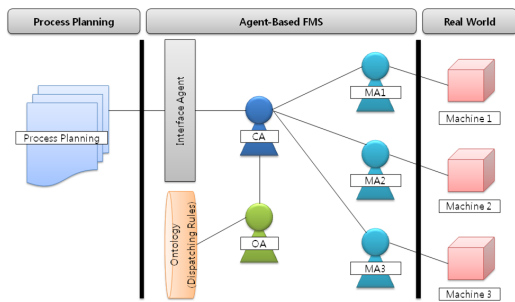
기존 연구들을 통하여 작업 환경이 복잡하고 실시간으로 변화하는 FMS 하의 동적 일정계획 문제에서 에이전트시스템이 효과적으로 작업장의 변화를 감지할 수 있음을 알 수 있다. 이는 에이전트가 다음의 4가지 특성을 갖기 때문이다.[11]

- Autonomous: 사용자, 관리자의 개입 없이 자율적으로 동작하는 특성.
- Proactive: 에이전트는 부여된 작업 목적, 목표를 위

해 독립적으로 동작 가능한 특성.

- **Reactive:** 환경의 변화, 사용자의 기계조작, 데이터의 입력 등의 변화에 반응하는 특성.
- **Social:** 다른 에이전트와의 통신을 이용한 정보교환이 가능한 특성.

본 연구에서는 위와 같은 에이전트의 특성을 이용하여 그림 1과 같은 에이전트 기반 FMS를 제안하고자 한다. 제안된 시스템은 4가지 종류의 에이전트들로 구성되어 있으며, 에이전트의 종류 및 기능은 다음과 같다.



[그림 1] 멀티에이전트기반 동적작업량할당 시스템 구성도

■ IA(Interface Agent)

인터페이스 에이전트는 이종(異類) 시스템, 또는 다른 에이전트로부터 전달받은 정보(또는 메시지)를 시스템에 적합하도록 변환하여 관리 에이전트(Control Agent)에게 작업량, 작업종류, 공정순서 등의 작업정보를 전달한다. 또한, 관리에이전트로부터 각 기계의 부하, 작업자의 부재 등의 작업장 상황을 전달받아 공정계획 파트에 전달하여 작업장 상황을 고려하여 공정계획을 수립할 수 있도록 한다.

■ CA(Control Agent)

관리 에이전트는 인터페이스 에이전트로부터 작업정보를 전달 받아 작업배정규칙 추론을 위해 온톨로지 에이전트(Ontology Agent)에 전달한다. 추론된 작업배정규칙과 작업내용 메시지를 해당 기계 에이전트(Machine Agent)에게 보내 작업요청을 한다.

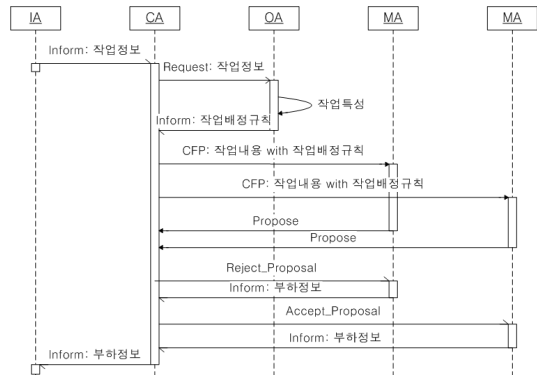
■ OA(Ontology Agent)

온톨로지 에이전트는 전달받은 작업정보를 이용하여 작업특징을 추출하여 작업특징에 따른 최적의 작업배정규칙을 사전에 지식화 해 놓은 ‘작업배정규칙 온톨로지’로부터 최적의 작업배정규칙을 추론하여 관리 에이전트

에게 전달한다.

■ MA(Machine Agent)

기계 에이전트는 시스템 내부의 각각의 기계를 나타낸다. 각 기계의 작업스케줄, 가용여부, 기계부하 등 해당 기계의 정보를 가지고 있으며, 관리 에이전트에게 받은 작업 요청을 기계정보와 비교하여 작업가능 여부를 관리 에이전트에게 회신하고, 관리에이전트로부터 전달받은 작업일정에 따라 기계에 작업을 지시한다.



[그림 2] 에이전트 메시지 전달

그림 2는 에이전트들의 실제 상호 작용과 에이전트들 사이의 메시지전달과정 사이클을 순서 다이어그램으로 표현한 것이다.

IA가 작업정보를 CA에 전달하면 CA는 OA에게 작업 정보 메시지를 보내고 작업배정규칙을 요청하게 된다. OA는 작업정보를 이용하여 job특성을 추출, 해당 job특성에서 최적의 작업배정규칙을 추론해 CA에 전달한다. CA는 해당 작업이 가능한 모든 MA들에 작업 내용과 추론된 작업배정규칙을 CFP(Call For Propose) 메시지 형태로 보낸다. MA는 요청 받은 작업 내용을 자신이 대표하는 기계의 작업일정과 기계부하 등 작업장상황을 고려하여 작업 가능 여부 및 작업 종료시간을 CA에게 전달하게 되고, CA는 작업을 수락한 MA에게 최종 작업지시를 내리게 된다. 또한, 작업가능여부와 함께 각각의 MA로부터 전달받은 기계부하정보를 취합하여 IA를 통해 공정계획에 전달한다. 그림 2에서 사용된 메시지의 형식은 FIPA (Foundations of Intelligent Physical Agents)[12]에서 정의한 ACL (Agent Communication Language) 메시지를 사용하였다. 표 1은 ACL 메시지 형태와 기능을 기술한 것이다.

[표 1] ACL 메시지 형태

메시지 종류	설명
Request	다른 에이전트에게 특정 행위를 이행하도록 요구를 표현
Agree	앞으로 어떠한 행위를 이행하는데 있어서 동의를 표현
Inform	에이전트가 다른 에이전트에게 현재 상태 또는 조건이 사실임을 표현
Failure	에이전트가 시도는 하였으나 실패한 행위에 대한 표현
Reject	Propose에 대한 거부 표현
Propose	일정 조건에 맞는 행위가 실행 가능함을 표현

### 3. 온톨로지 기반의 FMS 동적 작업할당

#### 3.1 작업할당 규칙 온톨로지

Gruber[13]는 ‘컴퓨터 온톨로지(Computational Ontology)는 특정 영역(Domain)의 합의된 지식(consensual knowledge)을 공유하는 개념화의 형식적이고 명확한 명세이다.’라고 정의하였다. 정의의 의미를 풀어 설명하면 컴퓨터 온톨로지는 특정 영역의 지식을 표현하기 위해 모든 구성원이 수용 가능한 보편적인 개념과 개념들 간의 관계를 추상적으로 모델링하고, 이를 컴퓨터가 읽고, 처리할 수 있도록 제약조건을 명시적으로 정의한 지식표현방법이라 할 수 있다. 본 연구에서 제안된 시스템에 작업배정규칙을 온톨로지 지식으로 적용함에 따라 다음과 같은 장점이 있다.

- 합의된 지식의 사용으로 같은 시스템을 사용하는 다른 생산시설뿐만 아니라 유사업종 시스템에서 지식의 재사용이 가능하다.
- 명확한 명세화로 인해 FMS의 시스템과 제약조건을 온톨로지 안에 정의할 수 있다.
- 컴퓨터가 읽고, 처리할 수 있는 방법으로 지식을 표현하기 때문에 FMS의 NC머신, AGV의 기능, 사양 및 제약조건을 지식으로 표현, 컴퓨터를 이용해 지식을 사용하고 해당 기능을 제어할 수 있다.
- 지식을 개념들의 관계로 표현하므로 지식의 추가, 삭제가 자유롭다. 지식을 추가하거나 삭제하여도 해당 부분의 개념 및 관계만 재설정하면 전체 온톨로지 지식에는 영향을 미치지 않으므로 지식을 유연하게 관리할 수 있다.

본 연구에서는 OWL (Web Ontology Language)을 이용하여 온톨로지를 구현하였다. 온톨로지는 XML, RDF, OWL등 다양한 마크업 언어를 이용해 표현할 수 있으나, OWL은 W3C(The World Wide Web Consortium)[14]에 의해 개발되었으며, 현재 다른 마크업 언어보다 표현력이나 추론 능력에 있어서 가장 뛰어난 언어로 평가되고 있다.[15] OWL은 OWL Lite, DL(Description Logics), Full이 있으며, 후자로 갈수록 표현력이 향상되는 대신에 지식의 추론이 어려워진다. OWL Full의 경우 가장 풍부하게 지식을 표현할 수 있지만, 현재까지 완벽하게 지원하는 추론 시스템이 개발되어있지 않다. 따라서 본 연구에서는 OWL DL을 이용하여 ‘작업배정규칙 온톨로지’를 생성하였다. OWL DL은 추론 가능하며, FMS의 복잡한 작업배정규칙 및 제약조건을 표현할 수 있는 표현력을 가지고 있다.[16]

OWL DL은 클래스(Class)와 속성(Property)을 이용해 지식을 표현한다. 본 연구에서는 Lee[2]의 작업배정규칙 시뮬레이션 결과를 참고하여 클래스를 크게 시스템에 적용할 작업배정규칙들로 이루어진 작업배정규칙 클래스와 작업배정규칙 할당을 위해 시스템의 작업내용 및 작업특성을 정의한 job특성 클래스로 정의하였으며, 이들의 관계를 온톨로지 속성으로 표현하였다. 표 2는 제안된 작업배정규칙 클래스 중 SPT(Shortest Processing Time)규칙을 마크업 언어로 표현한 예이다. 정의된 SPT는 작업배정규칙 클래스에 속하며, 모든 작업배정규칙 클래스는 Job특성 클래스(Job\_Features)를 속성 값으로 갖는다. 제안된 ‘작업배정규칙 온톨로지’의 모든 클래스의 관계는 아래의 그림 3과 같으며 온톨로지 제작 도구 Protégé를 이용하여 작성되었다.

[표 2] 작업배당규칙 온톨로지의 문법의 예

```

<owl:Class rdf:ID="DPR_SPT">
  <rdf:subClassOf rdf:resource="#Dispatching_Rules"/>
</owl:Class>
<owl:Class rdf:ID="Dispatching_Rules">
  <rdf:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasJobFeatures"/>
      <owl:someValuesFrom rdf:resource="#Job_Features"/>
    </owl:subClassOf>
  </owl:Class>
  
```



[그림 3] 작업배정규칙 온톨로지

### 3.2 FMS 동적 작업할당

제안된 FMS에서의 동적 작업배정규칙 할당은 작업장의 상태 및 작업내용의 특징을 이용해 최적의 작업배정규칙을 추론하여 시스템에 적용하는 것이다. 이를 위해 기존의 관련 연구 및 시뮬레이션을 이용해 작업장 상황에 따른 최적의 작업배정규칙을 온톨로지로 작성하고, 작업장의 상태 및 작업내용이 변경될 때 빠르게 변경된 환경에 맞는 작업배정규칙을 추론하여 시스템에 적용한다.

온톨로지 구조는 TBOX(Terminological BOX)와 ABOX(Assertional BOX)로 구성되어 있다. TBOX는 ‘엄마는 여자이며, 자녀를 갖는 사람이다’ 등의 온톨로지의 스키마를 포함하고 있으며, ABOX는 ‘영희는 여자이다’ 같은 인스턴스로 구성되어 있고, TBOX형식에 따른 각 인스턴스에 대한 지식이 표현된다.

온톨로지의 추론은 TBOX와 ABOX추론으로 크게 나누어진다. TBOX추론은 온톨로지를 설계하는 과정에서 스키마에 정의된 클래스와 속성에 대한 검증에 활용되며, 온톨로지 안에 정의된 암묵적인 포함 관계를 명시적으로 추론하는 용도로 이용된다. ABOX추론은 ABOX안에 구축되어 있는 수많은 인스턴스 관계를 추론하는 기능으로, 하나의 인스턴스가 정의된 클래스에 포함되는지를 검사한다.[17] 따라서, 본 연구에서의 TBOX추론은 ‘작업배정규칙 온톨로지’에 정의된 각각의 작업배정규칙 간의 암묵적인 관계를 명시적으로 표현하고 전체 온톨로지를 검증하는데 사용되었으며, ABOX추론은 새롭게 추가된 작업특성 인스턴스가 정의된 작업할당 클래스 중 어디에 해당하는지 추론한다. 본 연구에서는 TBOX추론과

ABOX추론 모두 RACER PRO™[18]의 추론엔진을 이용하였다.

## 4. 시스템 구현 및 실험

본 연구에서 제안하는 시스템은 온톨로지와 에이전트를 이용하여 작업배정규칙 지식으로부터 작업의 특성을 고려한 최적의 작업배정규칙 선택을 도와준다. 작업내용과 작업특성 및 작업장의 상황을 전달해주는 에이전트를 구현하기 위해 JADE™ 플랫폼을 이용하여 시스템을 구축하였다. JADE™ 플랫폼은 MAS를 제공하는 자바(JAVA) 기반의 에이전트 개발 프레임 워크로 FIPA기준에 의해 에이전트의 생성, 대기, 소멸 등 에이전트의 생명주기(life \*작업시간, 도착간격 단위: 분 cycle)를 관리해준다. 또한, 그래픽유저인터페이스(Graphic User Interface: GUI)와 다양한 에이전트 개발관련 도구를 제공하여 에이전트들의 상태 및 에이전트들 간의 메시지 전달 과정을 실시간으로 확인할 수 있다.

제안하고자 하는 시스템을 실험하기 위해 4대의 기계에 4종류의 작업을 가정하여 실험하였다. 또한, 시스템의 주요 장애는 없다고 가정하였다. 표 3은 4종류의 작업과 각 작업 종류의 공정, 그리고 각 공정별 할당된 기계등을 정리한 표이다. Job특성을 계산하기 위해 각 작업의 납기일과 작업우선순위(상,중,하)는 각각의 작업 별로 임의로 생성하여 실험하였다.

[표 3] 시뮬레이션 작업 목록

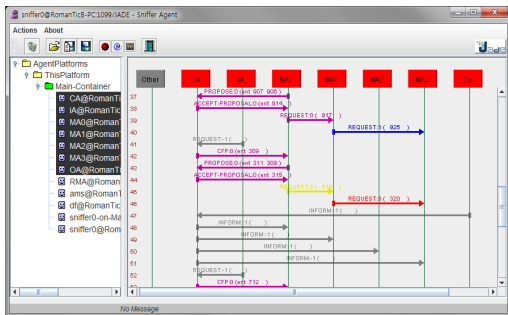
Job Type	Operation	Machine				Inter arrival Time
		M1	M2	M3	M4	
1	11	27				~N(13,1)
	12		40			~N(10,5)
	13				34	~N(16,4)
2	21		41			~N(16,1)
	22			21		~N(16,5)
3	31	29				~N(10,5)
	32			19		~N(18,4)
	33				36	~N(17,5)
4	41			25		~N(12,1)
	42			20		~N(19,1)

표 3과 같은 작업을 1000개씩 총 10배치(Batch)를 생성하여 데이터베이스에 입력하고 인터페이스에이전트를

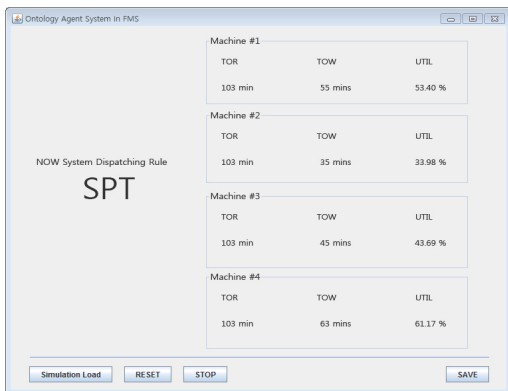
통해 시스템 내부로 전달하여 실험하였다. 작업은 특정한 사이즈 및 규칙을 가지고 도착하지 않는다. 작업특성 분석 및 작업할당 규칙은 Lee[2]가 제안한 방법을 이용하여 분석하였으며, 실험 결과는 10회의 실험결과를 평균하였다.

JADE™에서 제공되는 툴을 이용해 에이전트들의 메시지 전달 과정을 그림 4에서 볼 수 있다. 각각의 엘리먼트는 시스템에서 구동중인 에이전트를 나타낸다. 화살표는 에이전트간의 메시지 전달 경로를 표현하며, 화살표 상단에는 ACL 메시지 타입과 메시지의 고유 ID를 표시하여 메시지 전달 과정을 실시간으로 확인 및 추적할 수 있다.

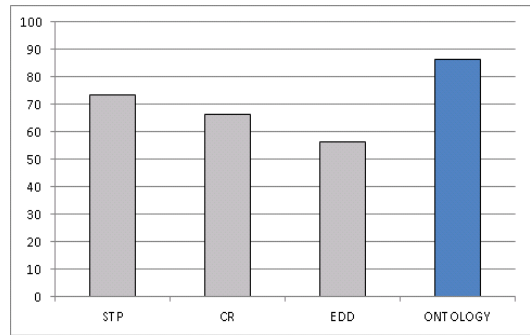
그림 4와 그림 5는 각각 제안된 시스템의 구현 화면과 실험결과이다. 각 기계의 총 가동시간, 작업시간, 가동률 및 실험이 진행되는 동안 실시간으로 각 머신에 할당된 작업배정규칙 추론 결과를 그림 5를 통해 알 수 있다. 그림 6은 본 논문에서 제시하는 동적 작업배정규칙 할당 방법과 일반적으로 사용 가능한 3가지(STP, CP, EDD) 작업배정규칙과의 실험결과를 비교한 그림이다. 실험 결과를 통해 한가지의 규칙을 적용한 것보다, 작업 내용의 변화에 따라 적합한 작업 할당규칙을 적용하도록 한 본 논문에서 제시한 방법이 나음을 알 수 있다.



[그림 4] 에이전트들의 메시지 전달



[그림 5] 동적작업할당 시스템의 예



[그림 6] 평균 작업 처리량 비교

## 6. 결론

본 연구에서는 에이전트와 온톨로지를 이용하여 작업 할당규칙을 작업의 형태와 작업장의 상황 변화에 유연하게 적용 가능한 범용적이고, 확장 가능한 시스템을 제안하였다.

본 연구에서 제시한 온톨로지를 이용한 동적 작업할당 규칙 할당 방법론은 기존의 방법론과는 다음의 차이점이 있다.

첫째로 지식의 재사용이다. 온톨로지를 이용하여 작업 배정규칙들을 지식화하여 생산시스템과 지식을 분리하였기 때문에 유사한 다른 생산시스템에서 지식의 재사용이 용이하다. 둘째로 시스템의 유연성이다. 온톨로지는 지식을 독립된 클래스들과 클래스들 간의 관계로 지식을 표현한다. 따라서 온톨로지를 이용해 지식화된 작업배정규칙들은 각각의 독립된 클래스로 구성되어 있어 작업배정규칙 지식을 추가/삭제하여도 다른 작업배정규칙 클래스에 영향을 주지 않는다.

따라서, 제시된 방법론은 유연생산 시스템 환경 하에서 시스템의 변경 및 유사한 시스템으로의 이전시 에이전트 및 온톨로지 지식의 재개발 없이 변경된 시스템에 적용 가능할 것으로 기대된다.

본 연구에서는 시스템의 변화에 따른 ‘작업배정규칙 온톨로지’의 학습에 대해서는 고려하지 않았다. 향후 연구로는 작업장 및 작업내용의 변화에 따라 동적으로 온톨로지 지식의 업데이트를 고려한 적응형 동적 작업배정규칙 할당에 대한 연구가 필요하다.

## 참고문헌

[1] Firdaus, M. Mustapha, Safaai Deris and Razib, M.

Othman, "A Survey on Ontology and Problem-Solving Method for Scheduling in Manufacturing", Postgraduate Annual Research Seminar, 3-4 July, 2007.

[2] Lee Sang-Wan, Kim Hae-Sic and Cho Sung-Youl, "A Study on Dynamic Scheduling in Flexible Manufacturing System Environment", Journal of the Society of Korea Industrial and Systems Engineering, Vol. 27, No. 2, pp.17-23, 2004.

[3] Solot, P, "A concept for planning and scheduling in an FMS", European Journal of Operational Research, Vol. 45, No. 1, pp.85-95, 1990.

[4] Ishii, N., Talavage, J. J., "A transient-based real-time scheduling algorithm in FMS", International Journal of Production Research, Vol. 29, No. 12, pp.2501-2520, 1991.

[5] Shen Weiming, "Distributed Manufacturing Scheduling Using Intelligent Agents", IEEE Intelligent Systems, Vol. 17, No. 1, pp.88-94, 2002.

[6] Cheeseman, M. J, Swann, P., Hesketh, G. B. and Barnes, S., "Adaptive manufacturing scheduling: a flexible and configurable agent-based prototype", Production Planning and Control, Vol. 16, No. 5, pp.479-497, 2005

[7] Wang Shi-jin, Xia Li-feng and Bing-hai, "FBS-enhanced agent-based dynamic scheduling in FMS", Engineering Applications of Artificial Intelligence, Vol. 21, No. 4, pp.644-657, 2008

[8] Safiye Turgay, "Agent-based FMS control", Robotics and Computer-Integrated Manufacturing, Vol. 25, No. 2, pp.470-480, 2009

[9] JADE(Java Agent Development Framework), <http://jade.tilab.com>

[10] Protégé, <http://protege.stanford.edu>

[11] Wooldridge, M. and Jennings, N.R., "Intelligent agents: Theory and practice", Knowledge Eng, Vol. 10, No. 2, pp.115-152, 1995

[12] Foundations of Intelligent Physical Agents(FIPA), <http://www.fifa.org>

[13] Gruber Thomas R., "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, Vol. 5, No. 2, pp.199-220, 1993

[14] W3C(World Wide Web Consortium), <http://www.w3.org>

[15] 노상규, 박진수, "인터넷 진화의 열쇠 온톨로지", gods Toy business, pp.138, 2007

[16] W3C(World Wide Web Consortium), <http://www.w3.org/TR/2004/REC-owl-features-20040210>

[17] Park Young-Tack, Choi Joongmin, "Overview and

Research Trends in Ontology Reasoning", KIISE, Vol. 24, No. 4, pp.17-23, 2006

[18] Racer, <http://www.racer-systems.com>

---

### 김 승 호(Seungho Kim)

[준회원]



- 2008년 9월 ~ 현재 : 고려대학교 정보경영공학과 석사과정

<관심분야>

생산 및 정보 관리 시스템, AI, 온톨로지

---

### 최 희 련(Hoeryeon Choi)

[정회원]



- 2003년 3월 ~ 현재 : 고려대학교 정보경영공학과 박사과정

<관심분야>

생산 및 정보 관리 시스템, AI, 온톨로지

---

### 이 흥 철(Hong-Chul Lee)

[정회원]



- 1983년 2월 : 고려대학교 산업공학 학사
- 1988년 2월 : Univ. of Texas 산업공학과 석사
- 1993년 2월 : Texas A&M Univ. 산업공학박사
- 1996년 3월 ~ 현재 : 고려대학교 정보경영공학과 교수

<관심분야>

SCM, 생산 및 물류 정보시스템, PLM