

다중 에이전트 기반 가상 생산 운영 시스템에 관한 연구

김선*(서울대 대학원 기계항공공학부), 공상훈(서울대 자동제어특화연구센터), 김기범(서울산업대 자동화 공학과), 한영근(명지대학교 산업시스템공학부), 이교일(서울대 기계항공공학부)

A Study on the Multi-Agent based VM Operating System

Seon. Kim(Mechanical and Aerospace Engineering .School, SNU), Sang-Hoon Kong (Automatic Control Research Center., SNU) , Gibom Kim (Manufacturing Automation .Dept, SNUT), Young-Geun Han(Industrial and Systems Engineering .School, Myongji University), Kyo-Il Lee(Mechanical and Aerospace Engineering .School, SNU)

ABSTRACT

As manufacturing systems have become complex and globalized, rapid development and production of products are essentially requisites for competitiveness. The importance of agility in manufacturing is being emphasized and a new paradigm is necessary for reduction of the time and expenses related to planning, product development and production. To meet such requirements, virtual manufacturing (VM) environment was suggested.

In this paper, Multi-agent system is adopted into VM operating system. Because our system is flexible due to agent technology, agents can be added or deleted with ease. VM unit modules which were defined as DEVS models execute independent simulation of other modules in unit level and compose one VM system with other modules. They also execute simulation in system level. This research can contribute to usefulness of VM environment due to flexibility and extensibility of this system

Key Words : Multi-Agent(다중 에이전트), DEVS(이산 사건 시스템), VMS(가상 생산 시스템) , Monitoring

1. 서론

컴퓨터, 인터넷, 소프트/하드웨어등의 급격한 기술 발전에 따라 생산 환경 또한 글로벌 생산 체제로 변화하고 있다. 이러한 생산환경 하에서 제품 개발에 있어 시간과 비용의 절감은 매우 중요한 요소이며 이를 위한 다양한 방법들이 시도되고 있다.

가상 생산(Virtual Manufacturing)환경의 도입은 이러한 시도 중 하나로 여러 생산 활동의 제반 문제를 각 단계에서 미리 검토, 최적화 할 수 있도록 하여 생산 시간, 비용을 최소화하고 있다.

가상 생산 환경의 폭 넓은 적용을 위해서는 유연성과, 재사용 가능성, 구축의 용이성들이 바탕이 되어야 한다. 이를 위해서 본 논문에서는 다중 에이전트 기반의 가상 생산 운영 시스템을 제안한다.

각 에이전트들은 분산된 환경 하에서 독립적인 객체로서 나중에 시스템이 변화 되더라도 재사용이 가능하다. 전체 시스템은 이후 실제 시스템의 변동으로 인한 에이전트의 추가나 삭제와 같은 문제에 유연하게 대처할 수 있으며 전체 에이전트의 상태 정보를 원격지에서 실시간으로 감시함으로써 해당 가상 생산 환경을 보다 쉽게 검토할 수 있다. 이를 위해서 각 가상 생산 단위 모듈은 DEVS 형식론을 따라 구현되었으며 독립적으로 시뮬레이션을 수행하는 개별 가상 생산 단위 모듈이 모여 하나의 가상 생산 시스템을 구성함으로써 다중 에이전트 시스템에 이식할 수 있는 가능한 가상 생산 시스템을 구현하였다.

대표적인 연구로 Manufacturing 과 Shop-floor control 을 통합하는 디자인 시스템을 제안한

ABCDE¹⁾와 에이전트 기반 통합된 동시공학 시스템을 제안한 PACT²⁾과 에이전트 기반 협력 디자인 시스템을 제안한 CSCW³⁾등이 있다.

2. Agent Technology

2.1 에이전트(Agent)

에이전트란 주어진 환경에 대해 자율적으로 반응하며, 목적성을 가지고 서로 통신하는 개체로서 사용자가 원하는 작업을 자동적으로 해결하여 준다. 처음에는 인공지능 분야에서 시작되었으나 1980년대 이후 분산 협동처리와 에이전트간의 통신의 개념을 배경으로, 인공지능과 분리되어 하나의 연구 주제로 발전하였다.

에이전트를 이용한 여러 연구 방향이 있으나 본 논문에서는 유연생산시스템과 같은 복잡한 이산사건시스템에 적용하기 위하여 다중 에이전트 시스템에 초점을 맞추었다. 다중 에이전트 시스템은 시스템 내에서 다수의 에이전트와 이들 사이의 행동을 조정하기 위한 원칙을 포함하는 방법론이다. 따라서 여기서 에이전트는 '분산환경에서 상호 협력을 통해 작업을 수행하는 컴퓨터 프로그램'이다.⁴⁾

서로 간의 소통과 협력을 전제로 하고 있으므로 다중 에이전트 시스템은 각 에이전트 간의 보다 구체화된 소통을 가능하게 하며, 적용에의 유연성과 용이성을 제공한다.

2.2 KQML(Knowledge Query Manipulation Language)

에이전트는 서로 간의 소통이 가능해야 하는데 이때 메시지 교환은 동일한 프로그램 뿐 아니라 다른 프로그램에서도 가능해야 한다. 이를 위해서는 표준이 필요하다. 또한 그 메시지는 복잡한 작업을 효과적으로 수행할 수 있는 표현력이 뛰어난 규격이 있어야 한다. 이 두 가지를 만족시키는 표준언어로 고안된 것이 KQML⁵⁾이다.

KQML은 DARPA Knowledge external interface group이 제안하였으며, 통신(communication), 메시지(message), 내용(content) 계층의 3단계 계층으로 이루어져 있다. 통신 계층은 에이전트 사이에 교환되는 전체 메시지를 의미하며, 메시지 계층은 내용 계층을 포함한 수행할 행위를 기술하는 수행문(performative)으로 구성된다. 내용 계층은 수행할 내용을 나타낸다. 본 논문에서는 KQML을 따르되 KQML의 기본 수행문(performative) 외에 필요한 수행문을 시스템에 맞게 정의하여 사용하였다.

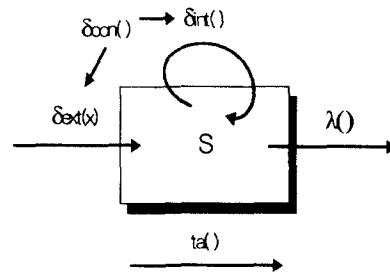


Fig. 5 대상 가상 유연 생산 시스템

(S: 상태

$\lambda()$: 출력에 따라 외부 사건 발생 출력 함수

$\delta_{int}()$: 내부 상태 이행 함수

$\delta_{ext}(x)$: 외부 상태 이행 함수

$\delta_{con}()$: $\delta_{int}()$ 와 $\delta_{ext}(x)$ 의 순서를 기술하는 함수

$ta()$: 시간 함수)

3. DEVS(Discrete Event System Specification)

Zeigler⁶⁾에 의해 개발된 DEVS 형식론은 이산사건 시스템을 계층적이고 모듈화 된 모델링이 가능하게 한다.

DEVS 형식론은 다른 형식론과는 달리 명백히 입력과 출력이 구분되어 시간이 지원되며, 내부, 외부의 상태 이행 함수가 구분되어 자발적인 이행을 표현하기 쉽다. (Fig. 1 참조)

DEVS 형식론에서는 가장 기본적인 모델로서 원자(Atomic) 모델을 정의하고 이 기본 모델을 계층화, 모듈화 한 결합(Coupled) 모델을 정의한다.

원자 모델은 더 이상 나눌 수 없는 모델로 입력사건, 출력사건, 상태의 집합, 다음 상태를 기술하는 이행함수, 출력 함수, 시간 함수로 이루어져 시스템의 동적 특성을 표현한다.

결합 모델은 여러 모델간의 연결 관계를 기술하는 모델로, 이를 이용해 좀더 복잡한 구조의 모델링이 가능하다. 원자 모델간의 연결 관계는 사건을 기본으로 이루어지며 한 원자 모델에서 나온 출력이 또 다른 원자 모델의 입력으로 들어가는데 어떠한 변수도 공유하지 않으므로 결합 모델을 이루는 원자 모델은 그 하나가 모듈화 된 형태이다. 또한 결합모델은 원자 모델들을 계층화된 구조로 연결하여 DEVS 형식론은 계층화되고 모듈화 된 모델링을 하는데 유용하다.

4. 시스템 구현

시스템은 근간을 이루는 다중 에이전트 시스템과 각 에이전트안에 이식되어 있는 가상생산 단위

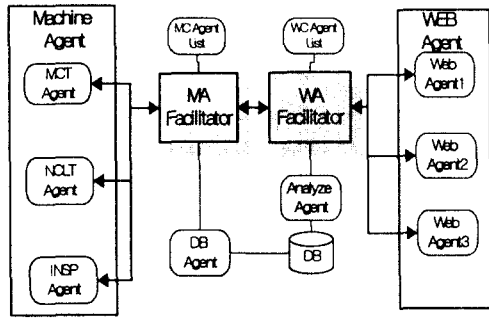


Fig. 2 다중 에이전트 시스템

모듈들의 두 부분으로 이루어져 있다.

본 시스템은 독립적으로 수행하며 서로 소통하는 다중 에이전트 시스템을 이용하여 각 기계 에이전트(Machine Agent)들의 상태정보를 웹 에이전트를 통해 원격지에서 감시할 수 있으며 더불어 가상 생산 단위 모듈 사이의 운영정보를 Machine Facilitator (MA Facilitator)를 통해 전달하여 시스템은 다음 단계를 수행할 수 있다.

4.1 전체 시스템 구조(다중 에이전트 시스템)

전체 시스템은 다중 에이전트 시스템으로 이루어져 있다. 대상이 될 기계 에이전트 그룹이 한 축이 되고, 그 기계 에이전트들의 운영 정보를 원격지에서 검토할 수 있는 웹 에이전트 그룹이 또 다른 한 축이 된다.

이 두 그룹간의 정보 전달을 매개하고 정보의 성격에 따라 정보 전달의 대상을 결정하는 역할을 MA Facilitator 및 WA Facilitator가 수행한다. 그 밖에 운영 정보를 DB에 저장하는 DB 에이전트와 정보를 분석하는 분석 에이전트(Analyze Agent)들이 있다. 각 에이전트들 간의 정보전달은 에이전트 표준 언어인 KQML을 따르고 있으며, 에이전트를 구현하기 위해 자바 언어로 된 에이전트 개발 패키지인 JATLite⁷⁾를 사용하였다. 시스템은 KQML을 기반 자바로 작성되었으므로, KQML을 따르는 다른 언어로 된 에이전트라면 시스템에 연결할 수 있다.

4.2 가상생산 모듈

Fig.3 과 같이 가상 생산 모듈이 내부에 이식되어 있는 기계 에이전트들과 관리 모듈이 이식되어 있는 MA Facilitator가 전체 가상 생산 운영시스템을 이룬다.

가상 생산 단위 모듈은 스스로 입력과 발생 사건에 따라 독립적으로 해당하는 기계처럼 행동하며 서로에게 MA Facilitator를 매개로 출력을 전달하여 다음 단계의 행동을 수행한다.

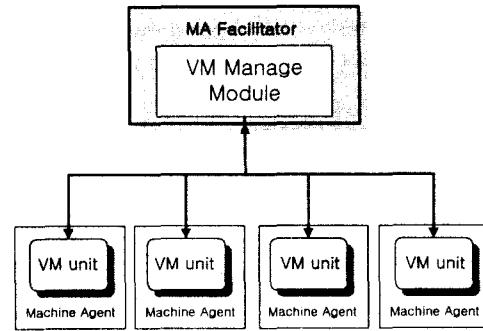


Fig. 3 가상 생산 모듈의 연결관계

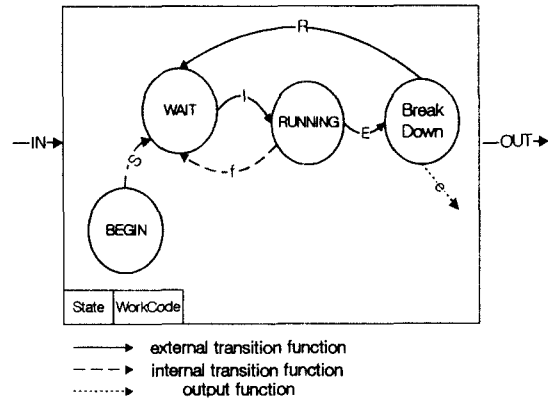


Fig. 4 가상 생산 단위 모듈의 상태 이행 선도

가상 생산 단위 모듈은 스스로 입력과 발생 사건에 따라 독립적으로 해당하는 기계처럼 행동하며 서로에게 MA Facilitator를 매개로 출력을 전달하여 다음 단계의 행동을 수행한다.

가상 생산 단위 모듈은 자바를 이용하여 앞서 설명한 DEVS 형식론을 따라 구현되었다. Fig.4 은 가상 생산 단위 모듈의 상태 이행 선도로 모듈은 외부로부터 입력이 오면 입력에 따라 그림과 같이 변화하며 시간이 지나 내부 상태 변화를 일으키는 사건이 발생하면 내부 상태를 변화시킴과 동시에 출력을 발생시켜 다음 단계를 수행하게 한다.

MA Facilitator는 그 출력을 받아 다음 프로세스를 결정하고 그 출력을 넘겨 사건을 발생시킬 에이전트를 선택하는 방식으로 관리, 조정 역할을 수행한다

5. 적용사례

위에서 개발된 다중 에이전트 가상 생산 운영시스템을 가상 유연 생산 시스템에 적용하였다.

5.1 대상 가상 유연 생산 시스템

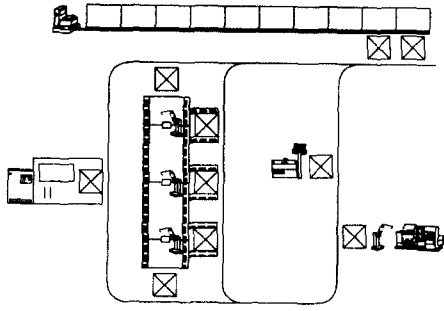


Fig. 5 대상 가상 유연 생산 시스템

대상 가상 유연 생산 시스템은 다음과 같이 구성하였다. 대 중 소 3 종의 전기모터를 가공, 조립 생산하는 가상 공장으로 아래에 정의 되지 않은 구체적인 사양은 서울대학교 FMS/CIM 센터의 사양을 따른다.

생산품은 대 중 소 3 종의 전기모터를 위한 부품인 case, rotor, shaft cap, end cap, 전체 조립공정의 총 15 품목이다. 주문 및 생산 계획단위는 주문을 취합하여 월 단위 계획 생산을 한다. 일단 한 달에 각 소형 모터 200 개, 중형모터 100 개, 대형모터 150 개를 생산하는 것으로 가정한다.(추후 생산량은 조정될 수 있으며 주단위 계획 생산, 긴급주문 처리, 생산량 조정, 검사 후 불량 처리, 가공 중 에러 처리 등도 추후 고려될 것이다.)

필요한 자재는 하나의 전기모터를 생산하기 위한 소재 4 가지(case, rotor, shaft cap, end cap), 2 종류의 베어링, 1 종류의 볼트 등 총 8 개가 필요하다

설비	개수	기타
Load/Unload Station	2 대	
ASRS	1 대	총 Pallet 수: 60 개, MCT 용 30 개, 선반용 30 개
검사기(INSP)	1 대	
머시닝 센터(MCT)	1 대	
터닝 센터(NCLT)	1 대	

Table. 1 대상 FMS 의 설비

5.2 적용 내용

Fig.6 은 기계 에이전트의 실행 화면이다. 현재 진행 정도를 알 수 있으며 주기적으로 웹 에이전트에 상태 정보를 보낸다. 그리고 작업이 끝나면 MA Facilitator 를 통해 다음 단계를 수행하기 위한 출력

을 보내고 웹 에이전트에도 알린다. Fig.6 은 여러 기계 에이전트 중 MCT 에이전트의 내용이다.

Fig.7 과 Fig.8 은 웹 에이전트가 유연 생산 시스템의 각 기계 에이전트들에서 정보를 전달 받아 상태를 감시하는 화면이다. Fig.7 에서 왼쪽의 유저 리스트에서는 웹 에이전트로 들어와 있는 다른 유저들을 알 수 있으며 왼쪽에서 감시하여 의견을 나눌 수 있다. 가운데의 네 격자로 나누어진 부분은 기계 에이전트들의 상태를 감시하는 부분으로 상태 바로 현재 진행 정도를 알 수 있으며 작업에 대한 정보는 그 밑에 리스트에 저장된다.(Fig.8 의 a) 참조) 오른쪽의 텍스트 박스에서는 이 정보를 더 자세히 보여준다.(Fig.8 의 b) 참조)

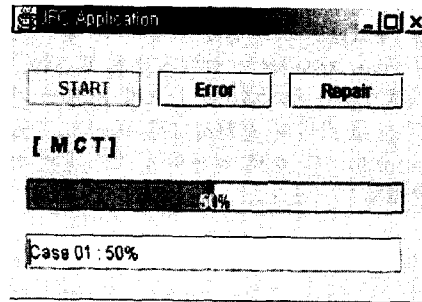


Fig. 6 MCT 에이전트 실행화면

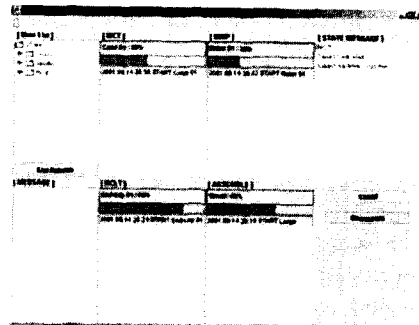


Fig. 7 웹 에이전트 화면

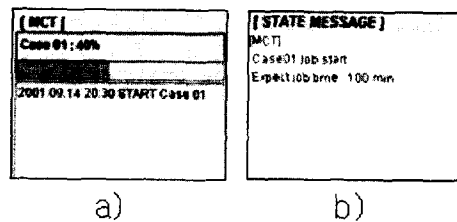


Fig. 8 a) 기계 에이전트 상태 바와 정보 리스트
b) 정보를 자세히 보여주는 부분

6. 결론

본 논문에서는 다중 에이전트 기술과 DEVS 형식론을 이용하여 가상 생산 운영 시스템을 개발하였으며, 이를 가상 유연 생산 시스템에 적용하였다. 다중 에이전트 시스템을 기반으로 하여 시스템에 확장성과 유연성을 획득할 수 있었으며 DEVS 형식론을 따른 가상 생산 모듈을 이용하여 이미 획득된 유연성과 확장성에 알맞은 가상 생산 시스템을 구현할 수 있었다.

이러한 가상 생산 시스템은 여기서 적용된 유연 생산 시스템과 다른 환경이라도 시스템이 기반이 되는 다중 에이전트 시스템을 적용하고 해당하는 에이전트의 가상 생산 모듈로 에이전트를 구성하여 그 모듈들의 관리 모듈만 수정하면 쉽게 다시 적용할 수 있다. 또한 에이전트의 소통 기능으로 구현한 감시 기능이 웹 기반으로 가능하기 때문에 가상 생산 운영 시스템을 원격에서도 검토할 수 있다.

참고문헌

1. S. Balasubramanian and D.H. Norrie, "A multi-agent intelligent design system integrating manufacturing and shop-floor control", Proc. First Int. Conf. On multi-agent Syst., San Francisco, pp.3-9, 1995
2. M. R. Cutkosky, R. S. Engelmores, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum and J. C. Weber "PACT: An experiment in integrating concurrent engineering systems", IEEE Computer, Vol.26, No. 3, pp.28-37, 1993
3. L. Monolaisir, "An integrated CSCW architecture for integrated product/process design and development", Robotics and computer-integrated Manufacturing 15, pp.145-153, 1999
4. 박진수, 에이전트 기반의 업무 프로세스 모델링에 관한 연구, 1999
5. The DARPA Knowledge Sharing Initiative External Interfaces Working Group, "Specification of the KQML agent-Communication Language - plus example 에이전트 policies and architectures"
6. B. P. Zeigler and S. Vahie, "DEVS formalism and methodology: unity if conception/diversity of application", Proc. Winter Simulation Conference, 1993
7. Stanford's Agent Based Engineering Group, Java Agent Template Lite, <http://java.Stanford.edu>

8. J .Sun, Y. F. Zhang, "agent-based product design and planning for distributed concurrent engineering", Proc.2000 IEEE, International Conference on Robotics and Automation