

FMS modeling & analysis using Object Oriented Simulation : Basic concept & Literature Survey

서석주, 김경섭
연세대학교 산업시스템공학과

<Abstract>

본 논문에서는 유연생산시스템(FMS: Flexible Manufacturing System)의 객체지향 시뮬레이션(OOS: Object Oriented Simulation)에 대한 기존의 연구들을 소개한다. 먼저 FMS와 객체지향시뮬레이션의 일반적인 개념에 대해서 이야기하고 기존의 연구들을 대상 시스템의 추상화(abstraction)과정을 중심으로 분류하여 정리하였다. 추상화 방법론과 FMS를 위한 객체지향시뮬레이션의 현황 및 과제에 대해서도 간략하게 살펴보았다.

1. 서론

시뮬레이션은 컴퓨터 상에서 실제 시스템과 거의 유사한 시스템을 구현시켜 이를 평가하고 분석하는 작업을 말한다. 복잡하고 엄청난 비용이 드는 시스템을 실제로 설치하고 가동하기 이전에 이들의 퍼포먼스가 어떤 모습을 보일지를 미리 알아보는 것은 매우 중요한 일이다. 이를 위해 시뮬레이션은 지난 오랜 시간에 걸쳐 중요한 역할을 수행해 왔다.

특히 시뮬레이션은 생산시스템에 있어 강점을 보여왔으며 이와 관련한 많은 연구가 이루어져 왔다. 비록 시뮬레이션이 최적해를 제공해주지는 않지만 사용자도 하여금 시뮬레이션 결과를 통해서 적절한 의사결정을 하도록 도와 주어왔다. 특히 유연생산시스템(FMS: Flexible Manufacturing System)은 그 행동양식이 매우 복잡하고 시스템 설치후 확장되거나 변형되는 경우가 많기 때문에 컴퓨터를 이용해 시스템을 모델링하고 분석할 수 있는 시뮬레이션은 각광을 받아왔다.

FMS를 위해 특화된 시뮬레이터도 존재하는 것이 사실이지만 일반적으로 SIMAN과 같은 범용 시뮬레이션언어를 이용하거나 C와 같은 범용 언어들을 통해 FMS에 대한 시뮬레이션은 이루어져 왔다[2]. 이런 언어들 이용을 이용한 시뮬레이션은 나름대로 많은 성과를 이루었으나 시스템이 복잡하고 다이나믹해질수록 한계를 나타내기 시작했는데 그 대표적인 것이 확장성과 재사용성의 부족, 그리고 사용의 불편함을 들 수 있다[1].

즉 FMS와 같은 생산시스템의 변화에 따라 수정과 보안을 용이하게 할 수 있는 소프트웨어에 대한 필요가 생겨나기 시작했다. 종래의 언어들 대부분이 절차지향 언어로서

시스템에 따라 가변하는 제어구조에 의존했기 때문에 대상 시스템이 변경되거나 확장될 경우 시스템에 대한 재분석과 새로운 설계가 필요하게 된다.

이에 대한 대안으로서 객체지향기법이 대두되었으며 객체지향기법은 시스템의 구조나 제어흐름에 의존하지 않고 시스템을 구성하고 있는 객체를 중요시하기 때문에 위에서 말한 일반언어를 이용한 시뮬레이션의 한계를 극복할 수 있게 되었다. 물론 SIMAN과 같은 시뮬레이션언어들이 Block이라는 개념을 통해 객체에 대한 일면을 제시하였지만 이 또한 사용자가 객체를 새로이 정의하거나 기존의 객체를 수정할 수 없다는 점에서 한계점을 가지고 있는 것이 사실이다.

본 논문에서는 지금까지 이루어진 생산시스템을 위한 객체지향 시뮬레이션연구들을 추상화(abstraction)과정에 초점을 맞추어 정리하였다.

2. 배경

FMS(Flexible Manufacturing System)라 불리는 유연생산시스템은 다품종 소량생산을 요구하는 오늘날의 생산환경에 적합한 시스템이다. 일반적으로 FMS는 작업을 수행하는 워크스테이션단위로 이루어져 있다. 각 워크스테이션은 실제로 부품을 가지고 작업을 수행하는 기계와 입고버퍼, 출고버퍼 또는 이 두 가지 버퍼의 기능을 동시에 수행하는 단일버퍼로 이루어져 있으며 각 워크스테이션사이의 물류는 AGV나 컨베이어와 같은 MHS (Material Handling System)가 담당하고 있다.

기계나 물류이동을 담당하고 있는 MHS는

일반적으로 상당히 고가이기 때문에 이들의 활용도를 높이기 위해 여러 부품이 한 기계를 공유하는 경우가 많으며 이는 FMS의 성격을 이해하고 묘사하는데 어려움을 가져오게 된다. 또한 FMS에서는 거의 대부분이 부품의 작업순서가 해당 부품마다 다른 job shop형태이기 때문에 상당히 복잡한 행동양식을 가지게 된다.

이와 같은 이유로 FMS에서는 deadlock이나 blocking같은 시스템의 병목현상이 빈번하게 발생하며 이들을 효과적으로 방지, 또는 해결함으로써 FMS의 성능을 높일 수가 있다.

FMS를 모델링하고 분석하는 연구는 지금까지 상당히 많은 연구가 있어왔다. 특히 위에서 언급한 deadlock과 blocking과 같은 현상은 생산시스템에서는 상당히 치명적이기 때문에 사전에(설치이전)시뮬레이션을 통해 이런 현상이 발생하는 워크스테이션을 미리 알아보고 적절한 조치를 취하는 것은 매우 중요하다.

이와 관련해서는 일반적으로 크게 두 가지 연구방향이 있는데 하나는 수학적인 모델을 이용하는 것이고 또 다른 하나는 시뮬레이션을 이용하는 것이다. 수학적인 모델은 FMS를 주어진 여러 입력 자료들을 바탕으로 대기행렬이론, petrinet등과 같은 수학적 이론을 바탕으로 접근하는 것을 말하며 시뮬레이션은 컴퓨터상에서 FMS를 구현, 모의실험을 통해 분석 및 평가하는 것을 말한다[31]. 수학적인 모델을 이용한 접근방법은 FMS와 같이 복잡한 모델의 경우에는 많은 한계가 있기 마련이며 따라서 비록 최적해를 제시해 주지는 않지만 가상공간에서 시스템을 모델링해서 평가하는 시뮬레이션은 최적의 방법론으로 평가받고 있다[31].

객체지향시뮬레이션(OOS: Object-Oriented Simulation:OOS)는 세상의 모든 것을 객체(object)로 보는 객체지향접근방법을 시뮬레이션에 응용한 것이다. 객체지향접근법이 프로그램에 응용된 것은 아주 오래 전 부터이다. 전문가 시스템(expert system)이나 지식기반 시스템(knowledge-based System)을 위한 개발 도구로 객체 지향 접근법은 애용되어 왔다

OOS는 보통 C++과 같은 객체지향프로그램을 이용해 구현되어 왔다. OOS이전에도 물론 시뮬레이션을 지원하는 여러 언어들이 존재했었다. 최초의 객체지향 시뮬레이션 언어로 Simula가 개발된 이후에도 MODSIM, Sim++, Smalltalk같은 OOS언어들이 개발되었다. SIMAN, SLAM, GPSS와 같은 언어의 경우 QUEUE, RESOURCE, TRANSACTION과 같은 객체를 제공해 객체지향적 접근을 보여 주었지만 사용자가 자신만의 객체를 만들어 낼 수 없다는 점에서 진정한

OOS언어라고 할 수 없다.

OOS에서는 모델링하고자 하는 시스템의 모든 구성요소를 객체(object)로서 취급하며, 이들 객체들 간의 상호작용을 기술하는데 초점을 맞추고 있다. 객체의 기술과 생성을 지원한다는 점에서 OOS는 다른 시뮬레이션 언어와 차별화된다[1].

객체는 자신만의 데이터 구조와 procedure를 가진 엔터티이며 encapsulation을 통해 모듈속에 데이터를 묶음으로써 객체들이 상호의존적으로 되는 것을 방지한다. 즉, 객체들은 메시지 전송을 통해서만 접근할 수가 있는 것이다. 메시지 전송(sending message)은 객체들 사이의 상호작용을 가능하게 한다. 메시지를 전달받은 객체는 그 메시지에 상응하는 메소드(method)를 생성시킨다. 이러한 현상은 생성된 procedure가 컴파일 단계에서가 아니라 실행단계에서 결정이 된다는 점에서 기존의 절차지향 언어가 함수를 호출하는 것과 차이가 있다. 이를 dynamic binding이라 부른다. 객체들은 클래스를 통해 계층적으로 조직되어 있으며 하위 클래스의 객체들은 상위 클래스의 속성을 그대로 물려 받는다. 이를 상속성(inheritance)라 부른다. 이 특성은 프로그램의 재사용성(reusability)을 보장해 주어 프로그램의 확장을 쉽게 만들어 준다[4,8,9,10].

3. Object Oriented Simulation for FMS

3.1 FMS를 위한 OOS의 장점

생산시스템은 일반적으로 machine, part, AGV와 같은 객체(object)로 이루어져 있다. 따라서 객체지향기법은 생산시스템을 표현하기에 적합한 방법론이 될 수 있다. 또한 위에서와 같은 물리적인 요소뿐만 아니라 생산시스템에서 필요한 정보들과 시스템의 구동을 위한 여러 논리적인 요소들과 의사결정을 지원하는 논리적인 표현 또한 객체의 개념으로 효과적으로 구현가능해 프로그램으로의 전이가 매우 용이하며 향후 생산시스템이 확장되거나 변경되어도 이에 잘 반응할 수가 있다. 상속성(inheritance)을 이용한 계층적인 클래스 구조는 프로그램이 지나치게 복잡하게 되는 것을 방지해 줄 뿐만 아니라 생산시스템이 확장되거나 변형되었을 때 이를 효과적으로 반영할 수 있도록 해준다. 다형성(polymorphism)은 메시지로써 통신을 하는 객체지향시뮬레이션에서 구현될 수 있는 특징으로 생산시스템에서 일어나는 여러가지 실제적인 사건을 표현하기에 적합하다.

3.2 Abstraction

객체지향시뮬레이션의 중요한 특징 중에 하나가 대상 시스템을 추상화(abstraction)할 수 있다는 것이다. 추상화는 구현하고자 하

는 시스템의 domain analysis를 통해 객체지향 시물레이션에서 구현되는 클래스들과 이에 필요한 변수 및 메소드를 정의하고 모델링하는 과정이다. 추상화를 통해 시물레이션 모델의 프레임워크가 결정된다. 객체지향 시물레이션에서 구현되는 다단계에 걸친 추상화는 대상 시스템을 "seize-hold-release"의 개념으로밖에 볼 수 없었던 전통적인 시물레이션 방법론을 극복하고 컴퓨터 프로그래밍으로의 좀더 자연스러운 전이(natural mapping)을 가능하게 하며 이를 통해 재사용성과 확장성등이 효과적으로 구현될 수가 있다. 또한 이 추상화를 기반으로 생산시스템에서 필요한 여러 구성요소(클래스)들간의 효과적인 통신이 이루어진다. 추상화는 구현언어에 따라서 조금씩 다른 모습을 보일 수 있지만 기본적으로 계층적 구조를 가지고 있으며 클래스의 재사용성 확보, 사용의 편의성이라는 목적을 가지고 진행된다. 따라서 객체지향시물레이션에서 추상화는 매우 중요한 단계이며 신중하게 설계될 필요가 있다.

기존 연구들의 추상화과정을 살펴보면 일단 다음의 두가지 흐름으로 구분할 수가 있다. 첫 번째는 시스템의 논리적인 면과 물리적인 면, 여기에 첨가해서 시스템에 필요한 정보적인 면을 구분해야 된다는 관점으로 추상화하는 것이다. Mize et. al.가 제안한 OSU-SIM[14,15,16]에서는 생산시스템을 '정보처리'와 '의사결정구조'라는 두 측면에서 봐야 하며 이를 위한 시스템의 구성요소로서 physical, information, control/decision object를 제안했다. physical object는 생산시스템의 물리적인 구성요소, information object는 정보적인 구성요소, control/decision object는 시스템의 구동로직과 의사결정을 지원하기 위한 것이다. 또한 Guo et. al.[29]은 생산시스템의 프로세스를 'physical material flow', 'information flow'의 상호연관된, 그러나 분리할 필요가 있는 엔터티의 흐름으로 이루어졌다는 인식에서 추상화를 진행하고 있다. Karacal과 Mize[12]의 연구에서는 물리적인 구성요소를 위한 System-Structure, BOM과 같은 정보적인 요소를 위한 ProductDataBase, 그리고 생산시스템에서 일어나는 동적인 정보와 SystemStructure, ProductDataBase를 유기적으로 연결시키기 위한 OrderDataBase의 3가지 구조를 제안했다. Narayaman et. al.이 제안한 OOSIM [22,30]의 연구 또한 생산시스템을 material flow control과 supervisory control의 관점으로부터 추상화 해야된다는 점에서 같은 맥락으로 볼 수가 있겠다. 이런 시각을 통해 material, locations, controllers, process plans라는 추상화를 유도했으며 이를 토대로 클래스를 개발했다. Adiga et. al.은 BLOCKS/M[20,21,30]을 소개하고 있다. BLOCKS/M

은 생산시스템, 특히 반도체 시스템을 위한 객체지향시물레이션의 선구자적인 역할을 해왔다. BLOCKS/M은 클래스의 재사용성에 역점을 두고 클래스라이브러리를 개발해왔다. 이들의 기본적인 추상화는 lots, resources, tasks, routers의 4가지 구성요소로부터 출발하며 'one function - one objects'의 원리를 채택, 클래스를 더욱 간단하게 디자인했다. 또한 sub-classing개념을 통해 기존의 클래스로부터 필요한 클래스를 확장할 수 있도록 하고 있다.

추상화의 또 다른 방법은 시물레이션 시스템을 'base-simulation-manufacturing'의 단계로 계층화 시키는 것이다. 즉 일반적인 여러 오퍼레이션을 위한 base 클래스와 RNG(Random Number Generator), 시물레이션 시간진행 메카니즘, 각종 통계값들의 추출 등 기본적인 시물레이션의 프레임워크를 위해 simulation클래스를 두고 특별히 생산시스템을 위한 manufacturing 클래스를 개발하는 것이다. Shewchuk과 Chang[7]의 연구는 생산시스템의 시물레이션을 위한 클래스를 base 클래스, simulation support 클래스, manufacturing systems 클래스로 계층화 시켰으며 Joines와 Robots[28]는 foundation 클래스, simulation 클래스, simulation modeling 클래스로 계층화시켰다. Ball et. al.[5]또한 abstract 클래스, support 클래스, application 클래스의 3가지 카테고리로 나누어 추상화를 전개시키고 있다.

이 두 가지 흐름이외에도 독특한 추상화를 보이고 있는 연구들이 있다. Macro et. al.[26]의 연구는 기존의 연구들이 생산시스템안의 엔터티흐름을 통제하는 제어장치의 상호작용보다는 엔터티 흐름과 연관된 사건에 중점을 둔다는 점과 생산공정중의 서로 다른 엔터티 타입의 흐름간에 존재하는 공동작용을 명확하게 하는 세부적인 프로세스 플랜을 포함하고 있지 않다는 점을 지적하며 ROOCH(Recursive Object - Oriented Coordination Hierarchy)를 확장한 HOOPLS(Hierarchical, Object Oriented Programmable Logic Simulator)를 소개하고 있다. 이 연구에서 추상화의 핵심은 "fractal unit"이라는 ports, queues, inhibit flags로 정의되는 input/output component의 set로 이루어진 모듈이다.

Lin과 Yeh[13]는 독특한 재사용성 매커니즘을 가지고 있는 simulation application framework을 제안했다. 이는 simulation context라는 특별한 개념의 모듈로 이루어져 있으며 시스템의 model frame과 experimental frame을 분리시켜 확장성과 재사용성을 제공하고 프로그램에러도 줄여 준다고 한다. 그들은 이 개념의 구현을 위해 service

object, scheduler object, model object를 개발했다.

Zeigler et. al.이 제안한 DEVS(Discrete Event System Specification)[23,24,25]에서는 다른 연구에서처럼 생산시스템의 물리적인 요소로의 직접적인 추상화를 시도하지 않았다는 점에서 주목할 만 하다. 즉, 시스템의 구조와 행동양식을 시뮬레이션 수행으로부터 추상화시키기 위해 모델을 독립적이고 검증 가능한 sub-system의 집합으로 인식하였다. 이 sub-system은 서로 연결이 가능한 모델에 의해 설계되며 전체적으로는 계층적구조를 가진 시스템으로 개발될 수가 있다. DEVS는 시뮬레이션 방법론과 시뮬레이션 구조론의 양쪽 모두의 성격을 띠고 있다.

Thomsma et. al.이 제안한 SmartSim/SmarterSim[17,18,19]은 DEVS의 정형론(formalism)의 원리로부터 아이콘을 이용한 그래픽적인 시뮬레이션 환경을 구현했다. 즉 DEVS의 추상화를 기반으로 생산시스템의 part, workstation, conveyors, router의 부가적인 추상화를 제공하기 위해 DEVS의 sub-system개념을 이용했다. 이 연구는 객체지향 시뮬레이션이 사용 용이한(user friendly) GUI환경을 구현할 수 있다는 것을 보여주고 있다.

3.3 Abstraction 방법론

언급했듯이 추상화과정은 시스템개발에 있어 핵심적인 요소이며 매우 신중하게 계획되어야만 한다. 생산시스템의 물리적인 부분을 효과적으로 묘사해야 하며 생산시스템이 가지고 있는 복잡한 행동양식 또한 추상화과정에 반영되어야 한다. Venkatesh와 Zhou[11]는 이에 대한 효과적인 방법론을 제시하고 있다. 이들은 특히 기존의 많은 연구들이 생산시스템의 동적인 행동양식을 표현하는데 미흡했다고 지적하며 객체지향기법, OMT(Object Modeling Technique), PN(PetriNet)을 통합한 방법론을 제시했다. 즉 시스템의 정적인 부분을 OMT의 object model을 이용해 표현하고 필요한 클래스를 추출하며 petrinet을 이용해 생산시스템의 동적인 모습을 표현하여 시스템을 분석하고 시뮬레이션을 통해 평가했다. 이 과정에서 필요에 따라 OMT에서 추출된 클래스와 petrinet모델을 적절히 수정해서 최종적으로 필요한 객체와 그들의 정적, 동적인 관계, 데이터구조등을 추출하는 것이다. 그리고 객체지향언어를 이용해 이를 구현할 것을 제안하고 있다.

3.4 현황과 과제

지금까지 FMS를 위한 많은 연구가 이루어져 왔으며 위에서 우리는 기존 연구들의 추상화과정에 대해 간략하게 살펴보았다. 살펴본 연구들 중에서 대학을 중심으로 대규모로

진행되고 있는 주목할 만한 연구로는 BLOCS/M(University of California, Berkeley), OOSIM(Georgia Institute of Technology), OSU-SIM(Oklahoma State University), DEVS(University of Arizona), SmartSim/SmarterSim(University of Michigan)정도를 언급할 수 있으며 생산시스템의 시뮬레이션과 모델링을 위한 통합된 환경(unified framework)을 개발하기 위한 연구가 필요한 시점에 있다[30]. 이에 Li, Hong Bing Yuan[27]은 다음과 같은 5개 항목의 FMS 시뮬레이션시스템이 갖추어야 될 조건을 제시하고 있다.

- 1) 사용자 편의중심의(user friendly)의 모델링 개발 환경을 갖추어야 한다.
- 2) FMS 동작의ダイナミック한 속성이 정적인 데이터와 프로세스 계획 양쪽 모두에 의해 쉽게 정의되어야 한다.
- 3) MHS, THS(Tool Handling System)이 고려되어야 한다.
- 4) FMS의 복잡한 컨트롤/스케줄링 로직을 시뮬레이션하기 위해 real dynamic control이나 스케줄링 프로세스가 시뮬레이션 시스템안에서 구현되어야 한다.
- 5) 올바른 의사결정을 지원하기 위해 지능적 분석 및 평가가 구현되어야 한다.

4. 결론

본 연구에서는 객체지향 시뮬레이션의 개념과 FMS를 위한 객체지향 시뮬레이션에 대한 연구들을 추상화(abstraction)과정을 중심으로 살펴보았다. 객체지향 시뮬레이션은 시뮬레이션 모델링과 구현에 있어 프로그램상에서의 자연스러운 전이, 모듈화된 디자인, 계층적 구조를 통한 소프트웨어의 재사용성과 확장성, GUI환경구현의 용이성, 사용의 편리함과 같은 장점을 지니고 있다. 이런 장점들은 시스템의 domain analysis를 통한 추상화(abstraction)를 통해서 이루어진다. 단 단계에 걸친 추상화는 생산시스템의 표현능력을 극대화 시켜줄 수 있다. 시스템 구현 언어에 따라서 그 표현방식이 조금씩 상이하지만 기존의 연구들은 재사용성의 확보와 사용의 편의성이라는 목적을 가지고 있으며 이는 앞으로도 객체지향시뮬레이션이 지향해야 될 점이라고 생각할 수 있다. 또한, 복잡한 FMS의 행동양식을 표현하기 위한 효과적인 방법론에 대한 연구가 더욱 필요하며 GUI 구현에 대한 연구 또한 병행되어야 할 것이다. 본 논문을 토대로 추후의 연구에서는 객체지향언어인 smalltalk를 이용해 생산시스템을 위한 객체지향시뮬레이터를 개발하고자 한다.

<References>

- [1] A. Najmi and S.J. Stein "Comparison of Conventional and Object-Oriented Approaches for Simulation of Manufacturing Systems" IIE Integrated Systems Conference & Society for Integrated Manufacturing Conference Proceedings. pp.471-476, 1989.
- [2] D.C. Pegden, R.E. Shannon and R.P. Sadowski "Introduction to Simulation Using SIMAN" McGraw Hill 1990.
- [3] A. Goldberg and D. Robson "Smalltalk-80" ADDISON WESLEY 1989.
- [4] D.P. Bischak and S.D. Roberts "Object-Oriented Simulation" Proceedings of the 1991 Winter Simulation Conference. pp.194-203.
- [5] B. Pete and D. Love "Expanding the Capabilities of Manufacturing Simulators Through Application of Object-Oriented Principles" Journal of Manufacturing System. Vol. 13, No. 6, 1994.
- [6] J.R. Drolet and M. Moreau "Development of an Object-Oriented Simulator for Material Handling System Design" Computers & Industrial Engineering. Vol. 23, No. 1-4, pp.249-252, 1992.
- [7] J.P. Shewchuk and T.C. Chang "An Approach to Object-Oriented Discrete Event Simulation of Manufacturing Systems" Proceedings of the 1991 Winter Simulation Conference. pp. 302-311, 1991.
- [8] O.M. Ulgen and T. Thomasma "Simulation Modeling in an Object-Oriented Environment Using Smalltalk-80" Proceedings of the 1986 Winter Simulation Conference. pp. 474-484, 1986.
- [9] Russell E. King and Kyung Sup Kim "Agtalk: An Object-Oriented Simulator for AGV Systems" Computers & Industrial Engineering. Vol. 28, No 3, pp.575-592, 1995.
- [10] Objectshare "VisualWorks: Application Developer's Guide" ObjectShare 1998.
- [11] K. Venkatesh and M. Zhou "Object-Oriented Design of FMS Control Software Based on Object Modeling Technique and Petrinets" Journal of Manufacturing Systems. Vol. 17, No. 2, pp.118-136, 1998.
- [12] S.C. Karacal and J.H. Mize "A Formal Structure for Discrete Event Simulation Part II: Object-Oriented Software Implementation for Manufacturing Systems" IIE Transactions, Vol.30, No. 3, pp.217-226, 1998.
- [13] J.T. Lin, K.C. Yeh and L.C. Sheu "A Context-Based Object-Oriented Application Framework for Discrete Event Simulation" Computers and Industrial Engineering Vol. 30, No. 4, pp.579-597, 1996.
- [14] J.H. Mize, H.C. Bhuskute, D.B. Pratt and M. Kamath "Modeling of Integrated Manufacturing Systems Using an Object-Oriented Approach" IIE Transactions. Vol. 24, No. 3, pp.14-26, 1992.
- [15] C.B. Basnet, P.A. Farrington, D.B. Pratt, M. Kamath, S.C. Karacal and T.G. Beaumar "Experiences in Developing an Object-Oriented Modeling Environment for Manufacturing Systems" Proceedings of the 1990 Winter Simulation Conference. pp.477-481,1990.
- [16] H.C. Bhuskute, M.N. Duse, J.T. Gharpure, D.B. Pratt, M. Kamath and J.H. Mize "Design and Implementation of a Highly Reusable Modeling and Simulation Framework for Discrete Part Manufacturing Systems" Proceedings of the 1992 Winter Simulation Conference. pp.680-688, 1992.
- [17] T. Thomasma and J. Madsen "Object Oriented Programming Language for Developing Simulation Related Software" Proceedings of the 1990 Winter Simulation Conference. pp. 482-485, 1990.
- [18] T. Thomasma and O.M. Ulgen "Hierarchical, Modular Simulation Modeling in Icon-Based Simulation Program Generators for Manufacturing" Proceedings of the 1988 Winter Simulation Conference. pp.593-600, 1988.
- [19] O.M. Ulgen, T. Thomasma and Y. Mao "Object Oriented Toolkits for Simulation Program Generators" Proceedings of the 1989 Winter Simulation Conference. pp.593-600, 1989.
- [20] S. Adiga and W.T. Lin "Object-Oriented Simulation of Manufacturing Systems" IIE Integrated Systems Conference & Society for Integrated Manufacturing Conference Proceedings. pp.489-494, 1989.
- [21] S. Adiga and C.R. Glassey "Object-Oriented Simulation to Support Research in Manufacturing Systems"

- International Journal of Production Research. Vol. 29 No. 12, pp.2529-2542, 1991.
- [22] U. Sreekanth, S. Narayanan, D.A. Bodner, T. Govindaraj, C.M. Mitchell and L.F. McGinnis "A Specification Environment for Configuring a Discrete - Part Manufacturing System Simulation Infrastructure" Proceedings of the 1993 IEEE International Conference on Systems, Man, and Cybernetics. pp.349-354, 1993.
- [23] B.P. Zeigler "A Knowledge-Based Simulation Environment for Hierarchical Flexible Manufacturing" IEEE Transaction on Systems Man and Cybernetics-Part A:Simulation and Humans. Vol. 26, No. 1, pp.81-89, 1996.
- [24] Y.R. Seong, T.G. Kim and K.H. Park "Performance Evaluation of a Parallel DEVS Simulation Environment P-DEVSIM++" 한국시뮬레이션 학회 논문지 제2권, 제 11호, 1993.
- [25] J.W. Rozenblit, T.G. Kim, and B.P. Zeigler "Towards an Implementation of a Knowledge-Based System Design and Simulation Environment" Proceedings of the 1988 Winter Simulation Conference. pp.226-230, 1988.
- [26] J.G. Macro, W.J. Davis and D.L. Setterdahl "Establishing an Object-Oriented Methodology for the Simulation and Control of Integrated Manufacturing Systems" Proceedings of the 1994 Winter Simulation Conference. pp.954-961, 1994
- [27] X.N. Li, H.B. Yuan, X.Y. Huang and E.H.M. Cheung "A New FMS Simulator with Object-Oriented Programming Techniques" Journal of Materials Processing Technology. Vol. 76, pp.238-245, 1998.
- [28] J.A. Joines and S.D. Roberts "Design of Object-Oriented Simulation in C++" Proceedings of the 1995 Winter Simulation Conference. pp.82-89, 1994.
- [29] D. Guo, D.H. Norrie and O.R. Fauvel "Object-Oriented Flexible Manufacturing System Simulation"
- [30] S. Narayanan, D.A. Bodner, U. Sreekanth, T. Govindaraj, L.F. McGinnis "Research in Object-Oriented Manufacturing Simulations: an assessment of the state of the art" IIE Transactions. Vol. 30 pp.795-810, 1998.
- [31] P. Solot and M.V. Vliet "Analytical Models for FMS Design Optimization: A Survey" The International Journal of Flexible Manufacturing Systems, Vol. 6, pp.209-223, 1994.