

객체지향 제조관리 시스템 평가를 위한 객체지향 시뮬레이터 개발

Development of an Object-Oriented Simulator for Evaluating Object-Oriented CIM S/W

백준걸*, 오훈언*, 신현준*, 김성식*, 이홍철*

Jun-Geol Baek, Hoon-Eon Oh, Hyun-Jun Shin, Sung-Shick Kim, Hong-Chul Lee

Abstract

Current CIM systems need to correspond flexibly to the frequent change of demands of the customers and the fast changing informations about the production environment. The demands have brought forth CIM systems developed using object-oriented technology, with reusability, expansibility, and flexibility for the system components.

Due to the fact that the OO-CIM system has been developed based on an incomplete data, the constructed system must be implemented to the actual surroundings to see how pertinent it is. Hence, this paper presents an OO-simulator as a means to evaluate the pertinency and the efficiency of the developed CIM system. The OO-simulator can determine the problems likely to occur when a developed CIM system is implemented to the actual site and evaluate the efficiency beforehand. Such properties will decrease the cost of CIM system development and increase the reliability of the system.

This paper presents a framework for an OO-simulator composed of a virtual factory component embodying the characteristics of a virtual factory, a connector component for the interface between the CIM system and the simulator, a configuration component for modeling the constituent structure of the CIM system, and a timer component in charge of the time advance for the simulation.

1. 서론

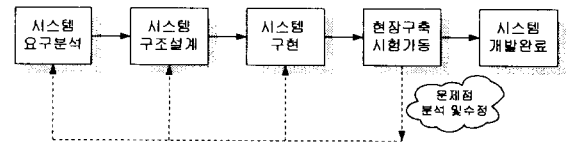
제조기업으로부터 다양한 제품을 보다 빠른 시간, 보다 싼 비용으로 공급받기를 원하는 소비자의 요구와 다양한 정보가 급속하게 변화하는 시장환경은 제조환경의 혁신을 요구하게 되었으며 이러한 요구에 대응하기 위한 방법으로 대부분의 제조기업들은 컴퓨터를 이용한 제조관리 시스템을 구축하고 있다. 그러나 지속적으로 변화하는 제조환경에 대응하기 위해서는 제조관리 시스템 자체가 이러한 변화를 수용할 수 있는 형태로 구축되어야 하는데 현재 대부분의 제조기업에서 사용되는 제조관리 시스템은 소프트웨어 제작기법 상 구조적 방법론을 이용하여 개발된 시스템으로써 기능중심의 고정된 형태를 지니게 되어 기업환경의 변화를 쉽게 반영할 수 없는 단점을 지닌다.

최근 들어 변화에 대한 대응의 근본적 한계를 갖는 구조적 방법론의 단점을 보완하기 위한 방법으로 새로운 시스템 개발 방법론인 객체지향 방법론이 등장함에 따라 많은 시스템 개발자들이 객체지향 방법론을 이용한 시스템 개발에 노력을 기울이고 있으며, 이러한 노력은 제조관리 시스템의 개발로까지 이어지고 있다. 객체지향 방법론을 이용한 제조관리 시스템의 개발은 제조관리 프로세스를 객체지향 분석을 통해 분석한 후 분석된 내용을 근거로 객체지향 설계 및 구현을 수행하는 것으로써 변화하는 환경에 대처할 수 있는 시스템 개발이 가능하다는 장점을 지닌다.[4]

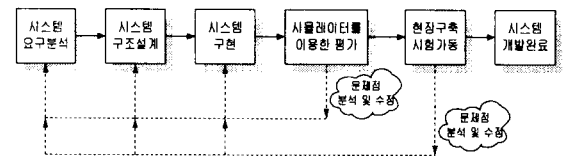
그러나 객체지향 방법론을 이용한 제조관리 시스템 개발 시 여전히 발생하는 문제점은 개발된 시스템이 현재의 제조환경을 얼마만큼 잘 반영한 것인가를 확신할 수 없다는 것이다. 대부분의 기업이 과거 데이터나 전문가 의견과 같은 불확실한 자료를 바탕으로 시스템을 개발한 후 개발된 시스템을 제조현장에 구축하여 시험 가동함으로써 시스템의 타당성 및 효율성을 검증하게 되고 검증과정에서 발견된 오류를 수정한 후 최종적인 개발을 완료하게 되는데 이러한 개발방법은 시스템의 개발기간과 비용을 증가시키는 문제점을 지니게 된다.

따라서 본 연구에서는 개발된 객체지향 제조관

리 시스템을 평가하기 위한 방법으로 시물레이터를 제안한다. 본 연구에서 제안하는 시물레이터를 이용하게 되면 제조관리 시스템을 실제 제조공장에 구축하기 전 그 성능을 평가할 수 있으며 이에 따라 시스템 개발에 소요되는 시간과 비용을 절감할 수 있게 된다. 즉, 기존의 제조관리 시스템 개발이 <그림 1>과 같은 절차로 진행되는 것에 비해 본 연구에서 제안하는 시물레이터를 이용한 제조관리 시스템 개발은 <그림 2>와 같은 절차를 통해 진행됨으로써 시스템 개발기간과 비용의 단축뿐만 아니라 개발되어지는 시스템의 신뢰성을 향상시킬 수 있게 된다.



<그림 1> 기존 제조관리 시스템 개발 절차



<그림 2> 시물레이터를 이용한 제조관리 시스템 개발 절차

객체지향 제조관리 시스템의 성능을 평가하기 위한 도구로 사용되는 시물레이터는 다음과 같은 특성을 지녀야 한다.

- 시물레이터는 객체지향 제조관리 시스템이 대상 제조환경에서 효율적으로 구동되는지를 평가하는 도구로 사용되어야 한다.
- 객체지향 제조관리 시스템은 제조환경의 변화에 유연하게 대처할 수 있는 시스템이므로 시물레이터 또한 이러한 변화를 수용할 수 있어야 한다.
- 시물레이터는 제조관리 시스템과 실시간으로 정보를 교환할 수 있어야 한다. (제조관리 시스템에서 결정된 정보는 시물레이터에 전달되어야 하며, 시물레이터에서 발생된 이벤트는 제조관리

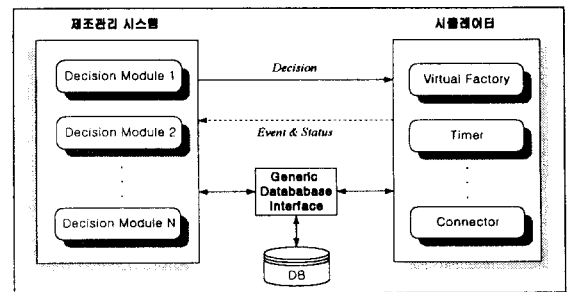
- 시스템에 전달되어야 한다.)
- 시뮬레이터와 제조관리 시스템은 필요한 정보를 공유해야 한다. (공통 데이터 인터페이스를 이용한 시스템 구축이 수행되어야 한다.)
- 시뮬레이터는 제조공장의 특성을 나타낼 수 있어야 한다.

최근 들어 시뮬레이션을 이용하여 제조시스템의 특성을 분석하는 연구가 폭넓게 진행되고 있다. Haddock[11]은 유연생산시스템(FMS)의 설계와 운용을 위해 시뮬레이션을 이용하였는데 시뮬레이션 생성기(simulation generator)를 사용하여 대상 시스템에서 필요로 하는 구성요소를 생성하고 이를 토대로 시뮬레이션을 수행함으로써 시스템의 성능을 평가할 수 있도록 하였다. Yih 등[17]은 유연생산시스템(FMS)에서 최적의 스케줄링 규칙을 선택하기 위한 방법으로 인공지능(AI), 유전자 알고리즘(genetic algorithm)과 시뮬레이션을 통합한 프레임워크를 제시하였다. 그러나 Haddock[11]과 Yih 등[17]의 연구에서 볼 수 있듯이 대부분의 연구가 대상이 되는 제조시스템의 성능(performance)을 평가하기 위한 도구로 시뮬레이션을 사용하고 있기 때문에 본 연구에서 목적으로 하는 시뮬레이터와는 근본적인 차이를 지니며 고정된 프레임워크를 갖는 시스템이기 때문에 변화에 대응하기 힘들다는 단점을 지닌다.

제조시스템의 변화에 대응하기 힘든 기존 시뮬레이션의 단점을 보완하기 위한 방법으로 객체지향 방법론을 이용한 시뮬레이터 개발이 활발히 진행되고 있다. Bruno와 Agarwal[8]은 컴퓨터 통합생산 시스템(CIM)을 객체지향 방법론을 이용하여 모델링하는 연구를 수행하였는데 이 연구에서는 컴퓨터 통합생산 시스템에서 필요로 하는 구성요소를 객체(object)로 정의하고 정의된 구성요소를 시뮬레이션을 통해 검증하였다. Barcio 등[5]은 SMOUCHES(State Machines for Object-Oriented Construction Hierarchical Engineering Specifications)를 이용하여 제조시스템을 모델링하고 객체지향 분석과 설계과정을 거쳐 필요한 객체를 정의하였으며 정의된 객체들 사이의 상태전이(state transition)를 분석하기 위한 방법으로 시뮬레이션을 이용하였다. Nadoli와 Biegal

[14]는 IMSAT(Intelligent Manufacturing Simulation Agents Tool)를 이용하여 지적 대리인(intelligent agent)을 생성하고 이를 이용하여 계층적 생산시스템(hierarchical manufacturing system)을 모델링하는 연구를 수행하였다. 이 연구의 특징은 생산시스템의 구성요소를 객체로 정의하고 정의된 객체들의 상태를 분석하는 대리인(agent)을 통해 시뮬레이션을 진행함으로써 정보(information)와 의사결정(decision-making)을 분리하였다는 점이다. 또한 Nakano 등[16]은 ROPSIII라는 객체지향 시뮬레이터를 개발하여 새로운 공장의 설계 시 시뮬레이션을 이용함으로써 설계된 공장의 성능을 평가할 수 있도록 하였다.

위에서 기술한 연구들은 구조적 방법론이 지닌 한계점을 극복하기 위해 객체지향 방법론을 이용하여 개발된 시뮬레이션을 제시하고 있지만 대부분의 연구가 제조시스템을 모델링하고 평가하는데 목표를 두고 있으므로 본 연구에서와 같이 제조시스템의 효율적인 운용을 위해 개발된 제조관리 시스템을 평가하는 시뮬레이터와는 근본적인 차이를 갖는다. 본 연구에서 제시하는 시뮬레이터는 앞에서 기술한 특성을 만족시키는 시스템으로 개발되어야 하는데 이를 위해 본 연구에서는 객체지향 제조관리 시스템의 변화를 수용할 수 있도록 객체지향 방법론을 이용하여 시뮬레이터를 개발하였으며 제조관리 시스템과의 정보 공유를 위해서 공통 데이터 인터페이스를 이용하였다. 본 연구에서 제안하는 객체지향 시뮬레이터는 <그림 3>과 같은 형태로 제조관리 시스템과 연결되어 제조관리 시스템의 성능을 평가하게 된다.



<그림 3> 제조관리 시스템과 시뮬레이터의 연결관계

본 논문의 2장에서는 시물레이션의 대상이 되는 객체지향 제조관리 시스템의 특징에 대해서 기술하고, 3장에서는 본 연구에서 제안하는 객체지향 시물레이터에 대해 기술한다. 4장에서는 시물레이터를 이용하여 객체지향 제조관리 시스템의 평가를 수행한 적용사례에 대해 기술하고 마지막으로 5장에서는 본 연구의 결론을 기술한다.

2. 객체지향 제조관리 시스템(KU-CIM)

소비자의 다양한 요구변화에 의한 제품의 수명주기(life cycle) 단축과 제조관련 정보의 급속한 변화는 이러한 변화에 유연하게 대처할 수 있는 제조관리 시스템을 필요로 하게 되었다. 그러나 현재 사용되고 있는 제조관리 시스템의 대부분은 기능(function) 분석과 데이터 흐름(data flow) 분석을 통해 개발된 기능 중심의 구조적인 시스템으로서 대상 제조시스템의 환경이 변화하는 경우 제조관리 시스템 전체를 점검하여 수정해야 하기 때문에 변화에 대응하는 시간과 노력이 많이 소요되는 문제점을 지닌다. 또한, 특정 제조시스템을 대상으로 개발된 제조관리 시스템을 다른 제조시스템에 적용하는 것이 사실상 불가능하기 때문에 여러 가지 형태의 제조시스템에서 공동으로 사용할 수 있는 단일 패키지(single package)를 개발한다는 것은 매우 어려운 현실이다.[4]

최근 들어 구조적 방법론의 한계를 극복하기 위한 방법으로 대두된 객체지향 개발 방법론은 실세계(real world)를 상호 작용하는 별개의 객체(discrete objects) 집합으로 간주하고 전체 시스템을 객체 클래스로 분할(decomposition)하여 분석하고 설계하는 모델링 방법을 사용함으로써 기계, 제품, 부품, 작업자 등 제조환경의 물리적 엔터티(entity)를 개념적 불일치 없이 쉽게 소프트웨어에 반영할 수 있도록 하였다.[15] 따라서 객체지향 방법론을 적용하여 구현한 소프트웨어는 실세계를 정확하게 묘사하는 객체들로 시스템을 모델링할 수 있으며, 정의된 객체를 재사용할 수 있다는 장점을 지닌다.

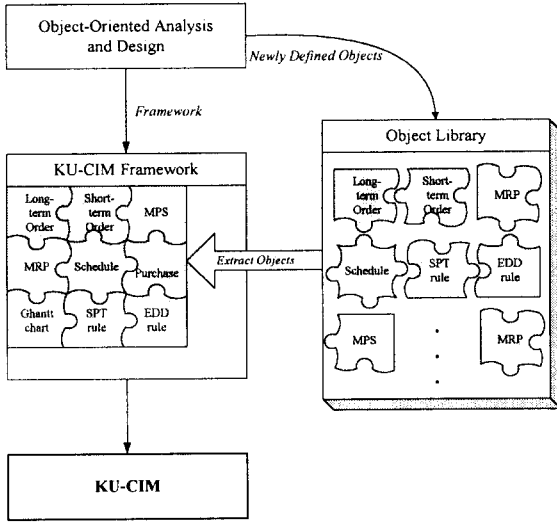
객체지향 방법론을 이용한 제조관리 시스템의 개발은 여러 가지 제조시스템의 특성을 분석하여 공

통적으로 사용할 수 있는 프레임워크를 개발한 후 대상이 되는 제조시스템의 특성을 반영한 구성 객체(component objects)를 정의하고 커스터마이징(customize)시키는 방법을 주로 사용한다.[4] 이러한 개발 방법은 여러 가지 제조시스템에서 공동으로 사용할 가능한 제조관리 시스템을 개발할 수 있는 기반을 제공하고, 제조시스템에서 발생하는 변화에 대해서는 해당되는 객체만 수정하면 다른 객체의 수정 없이 개발된 시스템을 재사용할 수 있는 장점을 제공한다.

고려대학교에서 객체지향 방법론을 이용하여 개발한 제조관리 시스템인 KU-CIM[3]은 제조 관리 시스템의 기능 중 생산관리 부분을 중심으로 개발된 것으로서 객체지향 분석 및 설계를 통해 제조관리 시스템의 공통적 특성을 나타내는 프레임워크를 정의하고 대상 제조 시스템에서 필요로 하는 기능과 알고리즘을 객체로 정의하여 객체 라이브러리(object library)를 구축한 후 <그림 4>와 같은 방법으로 제조관리 시스템을 개발하는 것이다. 따라서 KU-CIM은 대상이 되는 제조시스템의 환경이 변화하더라도 객체 라이브러리에서 필요한 객체들을 다시 추출하여 조합을 재구성하거나 필요한 객체만을 새롭게 정의함으로써 시스템을 재구성할 수 있기 때문에 변화에 유연한(flexible) 시스템 개발이 가능하다는 장점을 지닌다.

지금까지 객체지향 방법론을 이용한 시스템 구현에 대해 많은 연구가 진행되어왔는데 이러한 연구에서 밝힌 바와 같이 객체지향 설계는 경험적인 요소가 많은 부분을 차지하기 때문에 설계를 위해서는 많은 시간과 노력이 소요되며 설계된 결과의 재사용과 확장이 어렵다는 문제점이 대두되었다.[7][10][12] 따라서 KU-CIM은 객체지향 설계 단계에서의 효율성을 높이기 위한 방법으로 디자인 패턴을 이용한 시스템 설계를 실시하였는데, 디자인 패턴(design pattern)[10]은 경험적 설계 방법을 패턴화하여 저장하고 추후 재사용할 수 있는 방법을 제시한 것으로서 시스템 개발시간을 단축할 수 있는 장점을 지닌다. 또한 KU-CIM은 워크플로우(workflow)[13]를 이용하여 상호 연관된 개별 객체간의 연결을 도모하고 실행순서를 제어함으로써 제조시스템의 환경 변

화에 따른 제조관리 시스템의 변경(customize)을 용이하게 하고 새로운 기능 추가에 따른 확장성을 보장할 수 있도록 하였다.

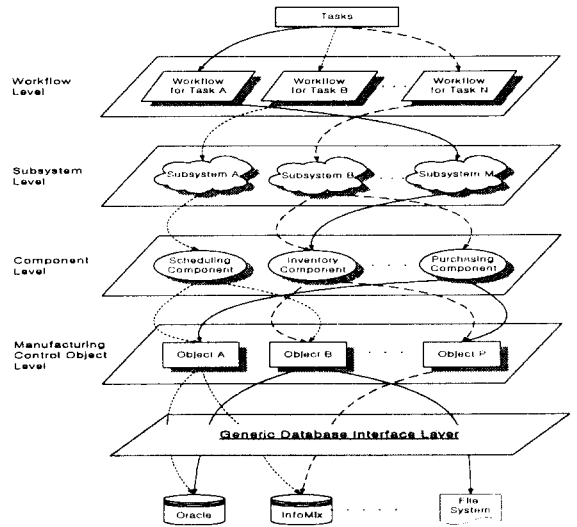


<그림 4> 객체지향 방법론을 이용한 KU-CIM 구축

KU-CIM은 제조관리 시스템의 고유한 특성을 살리기 위한 방법으로 객체지향 설계와 함께 기능 중심 설계(function driven design)를 수행함으로써 생산관리 기능을 담당하는 생산관리 객체를 추출하였고, 데이터 중심 설계(data driven design)를 통해 생산관리 객체에서 필요로 하는 정보를 전달하거나, 임시 저장소 역할을 수행하는 생산정보 객체를 추출하였다. KU-CIM은 객체의 종류를 생산관리 객체 그룹과 생산정보 객체 그룹으로 분리함으로써 생산관리 업무 수행 상 필연적으로 발생하는 데이터 공유로 인한 데이터 일관성(consistency) 문제를 해결하였는데 기존의 연구에서는 생산관리 객체에 데이터를 은닉(hiding)함으로써 다른 객체로부터의 데이터 접근을 어렵게 하고 데이터 일관성을 유지하기 어렵게 하는 문제점을 지니고 있었다.

객체지향 분석 및 설계를 통해 정의된 KU-CIM의 프레임워크는 <그림 5>와 같은 구조를 지니며 각 계층별 설명은 다음과 같다.

- 워크플로우 계층 (Workflow level) : 서브시스템의 실행 순서를 제어하여 제조관리 시스템에서 필요로 하는 작업(task)을 수행하는 계층
- 서브시스템 계층 (Subsystem level) : 제조 시스템의 업무에 따라 분류되어지는 계층
- 컴포넌트 계층 (Component level) : 서브시스템에서 정의한 업무를 세분화하여 실질적인 기능을 정의하는 계층
- 객체 계층 (Control object level) : 정의된 기능을 수행하기 위한 함수(function) 또는 알고리즘을 정의한 객체 클래스 계층



<그림 5> KU-CIM 프레임워크

KU-CIM에서 업무의 흐름은 <그림 5>에서와 같이 제조관리 시스템에서 필요로 하는 작업(task)에 따라 해당되는 워크플로우가 실행되고, 워크플로우는 요구되는 작업을 수행하기 위한 서브시스템을 선택하여 제어하게 된다. 또한 각 서브시스템에서는 실제로 수행할 업무를 정의한 컴포넌트를 선택하여 제어하며 컴포넌트 계층에서는 함수와 알고리즘을 정의한 생산관리 객체를 제어하여 업무를 수행하게 된다.

<그림 5>와 같이 계층적 구조를 갖는 KU-CIM은 상위 계층일수록 큰 단위의 객체로 구성되고 하위 계층일수록 작은 단위의 객체로 설계됨으로써 제

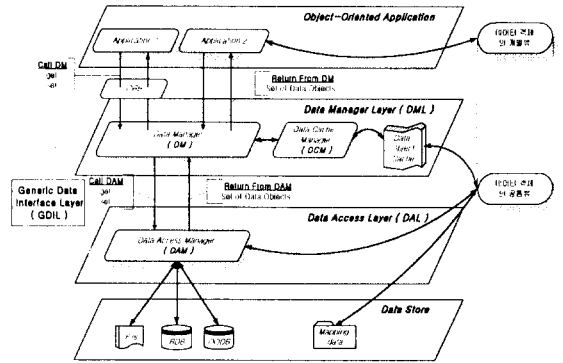
조환경의 변화에 의한 객체의 수정과 기능의 추가에 의한 객체의 삽입을 기존의 시스템 구조에 미치는 영향을 최소화하면서 가능하도록 설계되었다. 즉, 실제의 기능과 알고리즘을 수행하는 생산관리 객체들을 하위 계층에 두고 이들을 제어하는 상위계층을 추가로 정의함으로써 변화에 의한 수정과 추가 작업이 최하위 계층인 생산관리 객체 계층에서만 이루어지도록 설계하여 제조관리 시스템 전체에 미치는 영향을 최소화할 수 있도록 하였다.

앞에서도 언급한 바와 같이 KU-CIM에서는 객체지향 분석 및 설계와 함께 기능 중심 설계와 데이터 중심 설계를 병행함으로써 생산관리 객체와 생산정보 객체를 분리하여 데이터의 일관성을 보장하는 설계를 수행하였는데 <그림 5>에서와 같이 생산관리 객체에서 생산정보 객체로의 접근을 공통 데이터 인터페이스 계층을 통해 가능하도록 설계하고 생산관리 객체와 생산정보 객체, 데이터 저장소(데이터베이스, 파일시스템 등) 사이의 인터페이스를 단일화하였다. 이로 인해 데이터 저장소는 생산관리 객체와 생산정보 객체로부터 캡슐화(encapsulation)되어 종속성을 제거할 수 있고 공통 데이터 인터페이스의 구축을 통해 모든 생산정보 객체에서 사용 가능한 인터페이스를 제공함으로써 시스템 구축 시간을 단축할 수 있도록 하였다.

공통 데이터 인터페이스는 생산정보 객체로부터 데이터 저장소를 은닉하기 위해 생산정보 객체의 개별 뷰(specific view)를 공통 뷰(generic view)로 전환하는 일반화 패턴(generalizing pattern)을 이용하여 <그림 6>과 같은 구조로 개발되었다.[1][2] 공통 데이터 인터페이스의 개발은 실제 제조현장에 존재하는 이기종 데이터베이스를 KU-CIM에서 코드의 수정 없이 사용할 수 있도록 하며 생산정보 객체와 데이터 저장소 사이의 인터페이스 부분을 공통 데이터 인터페이스 계층으로 집중시킴으로써 인터페이스의 중복으로 발생하는 문제점을 해결할 수 있도록 하였다.

앞에서 언급한 바와 같이 KU-CIM은 객체지향 제조관리 시스템의 프레임워크를 개발하고 제조관리 시스템에서 필요로 하는 기능과 알고리즘을 객체로 구현함으로써 시스템의 재사용성과 유연성, 확장

성을 높일 수 있도록 하였다. 또한 공통 데이터 인터페이스를 통해 생산관리 객체와 생산정보 객체, 데이터 저장소를 독립시킴으로써 데이터의 일관성을 유지할 수 있고 데이터베이스의 종류에 상관없이 애플리케이션을 수행할 수 있도록 하였다. KU-CIM에 관한 자세한 내용은 김창욱 등[2]과 전진 등[3]의 연구를 참조하기 바란다.



<그림 6> 공통 데이터 인터페이스 계층 구조

3. 객체지향 시뮬레이터

3.1 객체지향 시뮬레이터의 필요성

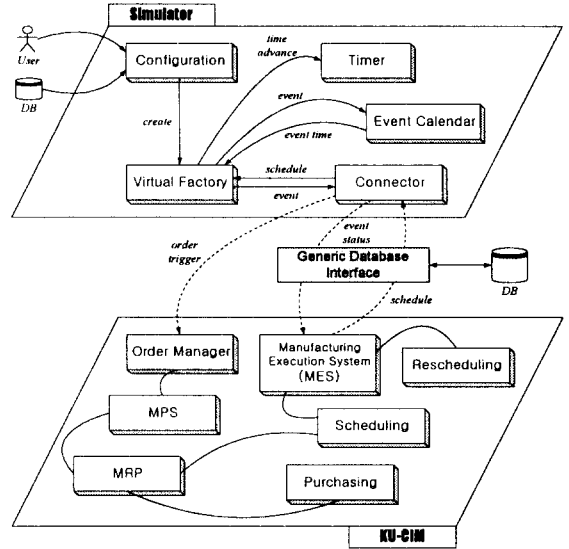
2장에서 설명한 객체지향 제조관리 시스템인 KU-CIM은 객체지향 분석 및 설계를 통해 시스템의 프레임워크를 구축하고 제조시스템에서 필요로 하는 기능과 알고리즘을 객체로 구현함으로써 실제 시스템을 구현하게 되는데 여기서 발생하는 문제점은 과연 구현된 객체가 필요한 기능을 제대로 수행하는가에 대한 의문과 구축된 프레임워크가 제조시스템에서 필요로 하는 기능을 정확히 수행하는지에 대한 의문이 발생한다는 것이다. 일반적으로 개발된 시스템을 검증하기 위해 사용되는 방법은 시스템을 실제 제조 현장에 구축하여 시험 가동함으로써 시스템에 대한 검증작업을 수행하고 검증과정에서 발생하는 문제점을 수정하여 최종적인 시스템 개발을 완료하는 것이다. 그러나 이러한 시스템 검증 방법은 시스템 개발기간과 비용을 증가시키는 문제점을 지니게

된다. 따라서 이러한 문제점을 해결하기 위한 방법으로 시뮬레이션을 이용한 시스템 검증방법이 대두되고 있지만 이들은 다음과 같은 문제점을 지닌다.

- 대부분의 연구에서 시뮬레이션은 대상 제조시스템의 성능(performance)을 평가하기 위한 목적으로 사용되었다.
- 대부분의 시뮬레이션은 스케줄링과 같은 의사 결정 모듈을 지니고 있으며 이를 통해 제조시스템의 운용을 결정하는 방법을 사용하고 있기 때문에 단순한 형태의 의사 결정만이 이루어지며, KU-CIM과 같이 복잡한 의사 결정 모듈을 포함하는 시스템과의 연결을 통한 시스템 검증이 불가능하다.
- KU-CIM의 검증을 위해서는 KU-CIM과 시뮬레이터와의 정보공유가 필수적인데 생산정보 객체와 공통 데이터 인터페이스를 사용하는 KU-CIM과 기존의 시뮬레이터와는 정보의 공유가 불가능하다.
- KU-CIM은 특성상 사용자와의 상호작용을 필요로 하며 이를 위해서 시뮬레이션의 시간 전진은 실세계 시간(real world time)과 가상 시간(virtual time)이 공존하는 혼합시뮬레이션(hybrid simulation) 형태로 진행되어야 한다.
- 전통적인 시뮬레이션은 변화에 대응하기 힘든 구조를 지니기 때문에 KU-CIM과 같은 객체지향 제조관리 시스템의 변화를 수용하기 어렵다.

따라서 본 연구에서는 객체지향 제조관리 시스템인 KU-CIM의 평가를 위해 객체지향 시뮬레이터를 제안한다. 본 연구에서 제안하는 객체지향 시뮬레이터는 KU-CIM의 변화에 유연하게 대처할 수 있고, 공통 데이터 인터페이스를 통해 KU-CIM과의 정보 공유를 빠르고 정확하게 수행함으로써 효율적인 시스템 검증을 수행할 수 있도록 하였다. KU-CIM에서는 제조시스템을 운용하기 위해 필요로 하는 의사결정을 수행하고 시뮬레이터에서는 KU-CIM에 의해 결정된 결과에 따라 가상적인 작업 진행을 수행하여 그 결과를 KU-CIM에 통보함으로써 KU-CIM의 성능을 평가하게 되는데 KU-CIM과

객체지향 시뮬레이터와의 연결관계는 <그림 7>과 같다.



<그림 7> KU-CIM과 시뮬레이터와의 연결관계

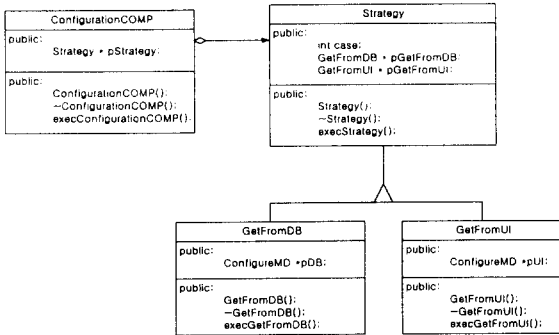
3.2 객체지향 분석 및 설계

재사용이 가능한 객체지향 소프트웨어를 설계하는 것은 매우 어려운 일이며 객체지향 분석과 설계를 통해 대상 시스템을 적절한 규모의 서브시스템으로 나누고 필요한 객체를 추출함으로써 새로운 객체의 추가나 기존 객체의 수정이 용이하도록 하는 일은 매우 중요하다.

본 연구에서는 시뮬레이터의 객체지향 분석을 위해 Jacobson[12]이 제안한 유즈케이스(use case) 모델을 사용하였으며 객체지향 설계를 위해서는 Booch, Jacobson, Rumbaugh가 각자의 방법론을 통합하여 제안한 모델링 언어인 UML (unified modeling language)을 사용하였다.[6][9] 또한 추출된 객체간의 관계를 파악하고 설계의 재사용과 유연성을 높이기 위한 방법으로 디자인 패턴(design pattern)[10]을 사용하였다.

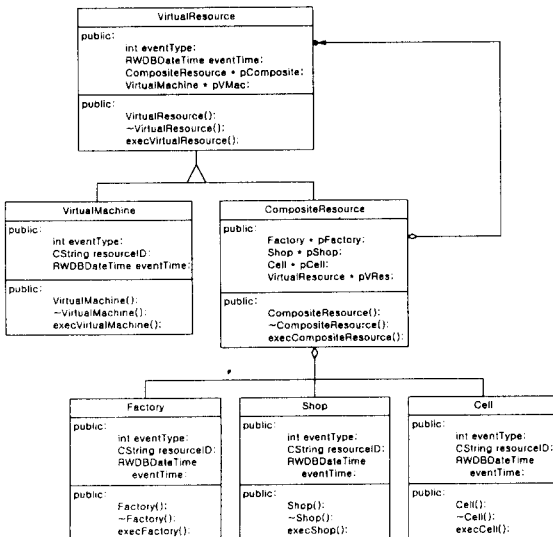
유즈케이스(use case) 모델은 대상이 되는 시스템에서 해야될 일을 묘사한 모델링 기술로서 근본적

들의 관계를 규명할 때 사용되었으며 <그림 10>과 같이 나타낼 수 있다. <그림 10>에서 컨피규레이션 컴포넌트 객체는 제조시스템의 물리적 정보를 데이터베이스 객체와 사용자 인터페이스 객체를 통해 입력받을 수 있도록 설계된 것이다.



<그림 10> 스트러티지 패턴을 이용한 설계

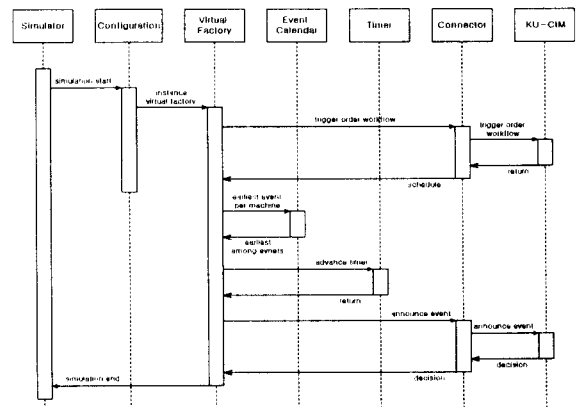
컴포지트 패턴은 부분(part)-전체(whole)로 구성되는 계층적 구조를 표현하기 위한 디자인 패턴으로 본 연구에서는 제조시스템의 계층적 구성요소를 표현할 때 사용하였으며 <그림 11>에서와 같이 제조시스템의 구성요소를 공장(factory)-샵(shop)-셀(cell)-기계(machine)와 같은 계층적인 형태로 나타낼 수 있다.



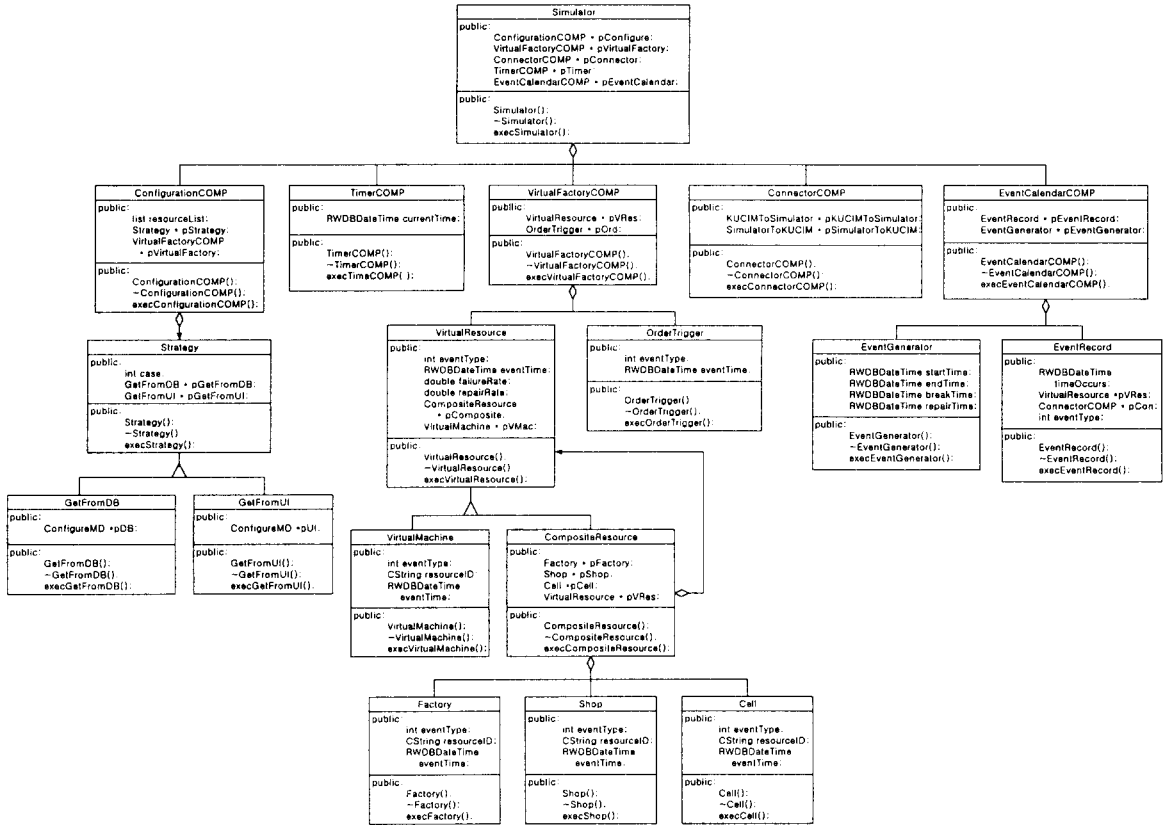
<그림 11> 컴포지트 패턴을 이용한 설계

객체지향 분석 및 설계를 통해 객체들이 추출되면 추출된 객체들을 바탕으로 객체들 사이의 상속(inheritance), 집단화(aggregation), 연관화(association) 관계를 분석하여 전체 시스템의 프레임워크를 설계하고 객체들간의 실행순서를 파악해야 하는데 이 때 사용되는 기술이 클래스 다이어그램(class diagram)과 실행순서 다이어그램(sequence diagram)이다. 본 연구에서 제시하는 시뮬레이터의 프레임워크는 <그림 12>와 같은 클래스 다이어그램으로 표현되어질 수 있으며 제시된 프레임워크에서 상위 클래스인 컴포넌트 객체들은 시뮬레이터 구조안에 집단화 관계로 묶이게 되고, 하위 클래스인 기본 객체들은 상위 클래스인 컴포넌트 객체로부터 상속 관계와 연관화 관계로 연결됨으로서 시스템의 재사용성, 확장성, 유연성을 높일 수 있도록 설계되었다.

<그림 13>은 시뮬레이터 컴포넌트 객체간의 실행순서 다이어그램을 나타낸 것인데 이것은 시뮬레이션이 진행되는 동안 객체들 간의 실행순서와 객체가 활성화(active)되는 기간 등에 대한 분석을 실시한 것으로써 객체지향 프로그램에서 전체 시스템의 제어 흐름을 한눈에 파악할 수 있도록 하는 기능을 제공한다.



<그림 13> 컴포넌트간 실행순서 다이어그램



<그림 12> 시뮬레이터 클래스 다이어그램

3.3 객체지향 시뮬레이터 구성 요소

객체지향 분석 및 설계에 의해 정의된 시뮬레이터 구성 객체들은 객체들간의 관계성 규명에 의해 <그림 12>에서와 같이 상위계층인 컴포넌트 객체들의 집합으로 정의될 수 있다. 시뮬레이터를 구성하는 컨피규레이션(configuration) 컴포넌트, 가상 공장(virtual factory) 컴포넌트, 이벤트 캘린더(event calendar) 컴포넌트, 커넥터(connector) 컴포넌트, 타이머(timer) 컴포넌트에 대한 설명은 다음과 같다. 각각의 컴포넌트에 대한 설명은 해당 컴포넌트의 목적, 역할, 다른 컴포넌트와의 관계, 클래스 다이어그램, 실행순서 다이어그램으로 나타내어진다.

(1) 컨피규레이션(configuration) 컴포넌트

- 목적 (Intent)

시물레이션 대상이 되는 제조 시스템의 물리적 환경 정보를 입력받는다. KU-CIM의 변화에 대해 유연하게 대처할 수 있는 객체지향 시물레이터의 특징으로써 제조 시스템의 물리적 환경 변화에 유연하게 대처할 수 있도록 하기 위해 제안되었다.

- 역할

제조시스템의 특성을 반영한 가상공장(virtual factory)을 구현하기 위해서는 제조시스템의 물리적 환경을 알아야 한다. 컨피규레이션 컴포넌트에서는 두 가지 방법에 의해 환경정보를 입력받을 수 있다

록 스트러티지 패턴[10]을 사용하였는데 첫번째 방법은 사용자 화면을 통한 드래그 앤드 드롭(drag and drop) 방식의 사용자 입력방법이고 두번째 방법은 데이터베이스에 저장되어 있는 환경정보를 읽어들이는 방법이다. 컨피규레이션 컴포넌트에서는 입력받은 정보를 바탕으로 시뮬레이션을 수행할 가상 공장을 생성하도록 지시하고 시뮬레이션을 수행하는데 필요한 파라미터를 초기화한다. 컨피규레이션 컴포넌트를 이용한 시뮬레이터의 구현은 대상 제조 시스템의 변화에 유연하게 대처할 수 있는 시스템으로의 구현을 가능하게 하며 외부로부터 가상 공장을 캡슐화(encapsulation)할 수 있기 때문에 시뮬레이터의 변화를 최소화할 수 있다.

(2) 가상공장(virtual factory) 컴포넌트

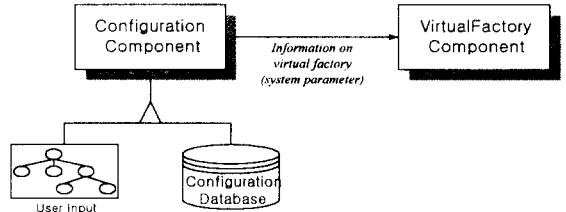
● 목적 (Intent)

컨피규레이션 컴포넌트로부터 전달받은 제조시스템의 환경정보를 바탕으로 가상 공장을 생성하고 생성된 가상공장 내의 가상기계에서는 KU-CIM에서 수립된 생산계획에 따라 작업을 진행하여 이벤트를 생성한다.

● 역할

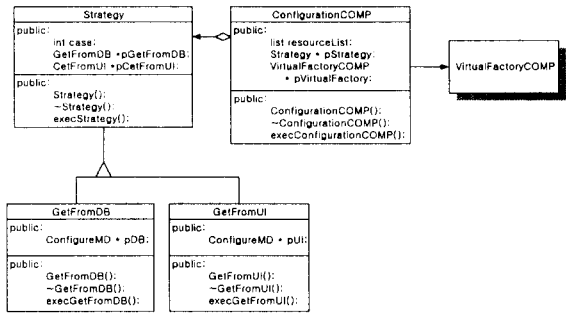
가상공장 컴포넌트는 컨피규레이션 컴포넌트로부터 제조시스템의 물리적 환경 정보를 전달받아 가상공장을 생성한다. 가상공장 컴포넌트에서는 제조시스템의 계층적 구조를 묘사하고 제조시스템의 물리적 변화에 쉽게 대처할 수 있도록 콤포지트 패턴을 이용하였다.[10] 가상공장 컴포넌트에 의해 생성된 가상기계에서는 KU-CIM에서 수립된 스케줄을 바탕으로 작업을 실시하여 이벤트를 생성하고 이를 이벤트 캘린더 컴포넌트에 전달한다.

● 컴포넌트간 관계



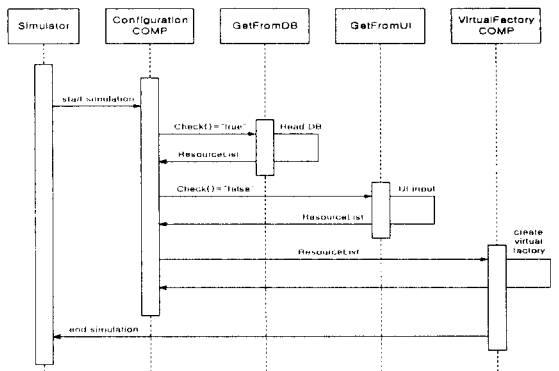
<그림 14> 컨피규레이션 컴포넌트 관계도

● 클래스 다이어그램



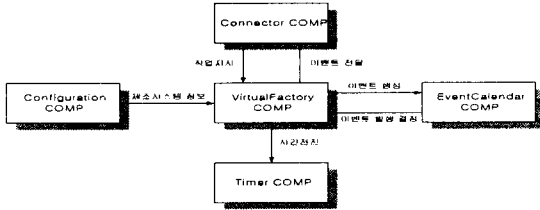
<그림 15> 컨피규레이션 컴포넌트 클래스 다이어그램

● 실행순서 다이어그램



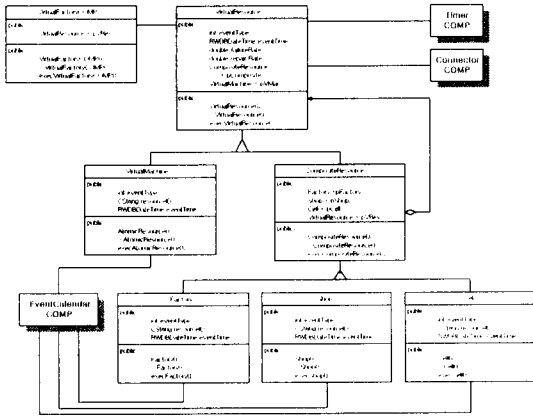
<그림 16> 컨피규레이션 컴포넌트 실행순서 다이어그램

● 컴포넌트간 관계



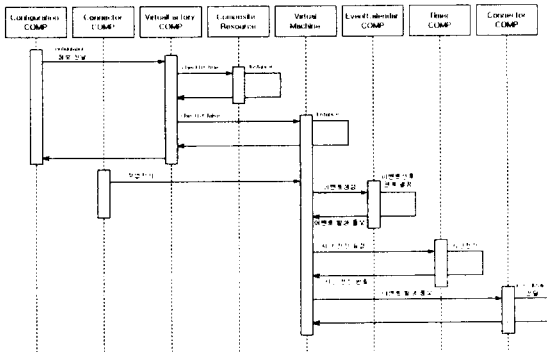
<그림 17> 가상공장 컴포넌트 관계도

● 클래스 다이어그램



<그림 18> 가상공장 컴포넌트 클래스 다이어그램

● 실행순서 다이어그램



<그림 19> 가상공장 컴포넌트 실행순서 다이어그램

(3) 이벤트 캘린더(event calendar) 컴포넌트

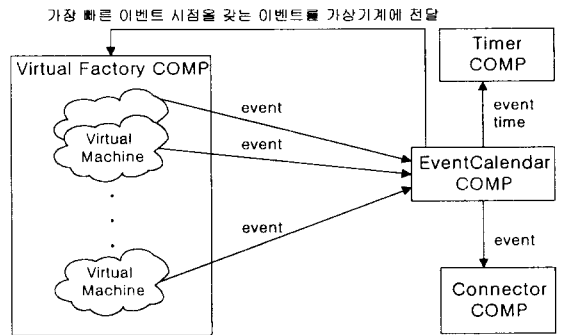
● 목적(Intent)

가상기계에서 발생된 이벤트를 총괄하여 관리하고 이벤트가 제조시스템의 운용시간 동안에만 발생되도록 조절한다.

● 역할

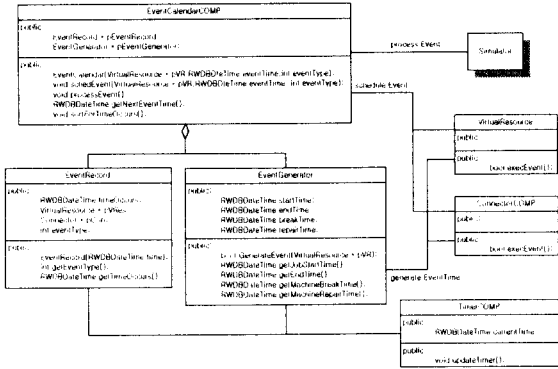
이벤트 캘린더 컴포넌트는 시물레이션이 진행되는 동안 발생된 이벤트를 총괄하여 관리한다. 즉, 이벤트 캘린더 컴포넌트는 여러 개의 가상기계에서 발생하는 이벤트를 전달받아 이벤트 리스트를 작성하고, 가장 빠른 발생 시간을 갖는 이벤트를 선택하여 해당 이벤트가 발생된 가상기계에 전달한 후 리스트에서 삭제한다. 또한 새로운 이벤트가 발생할 때 이벤트 발생시간은 제조시스템에서 실제로 작업이 가능한 시간이어야 하므로 워크캘린더(work calendar) 정보를 참조하여 이벤트의 발생을 제어한다.

● 컴포넌트간 관계



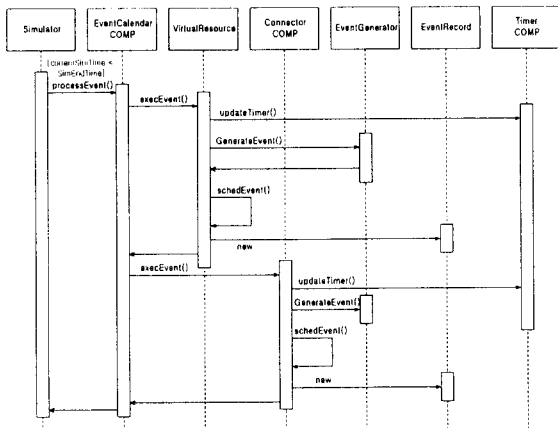
<그림 20> 이벤트 캘린더 컴포넌트 관계도

● 클래스 다이어그램



<그림 21> 이벤트 캘린더 컴포넌트 클래스 다이어그램

● 실행순서 다이어그램



<그림 22> 이벤트 캘린더 컴포넌트 실행순서 다이어그램

(4) 커넥터(connector) 컴포넌트

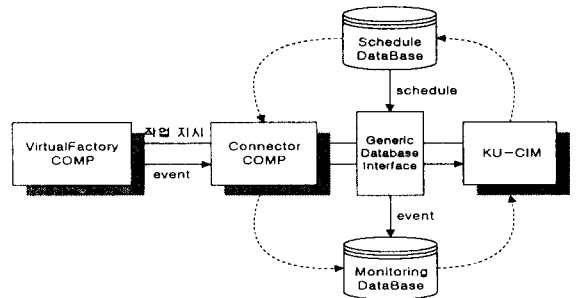
● 목적(Intent)

시뮬레이터가 KU-CIM과 유기적으로 연계되어 시뮬레이션을 진행할 수 있도록 인터페이스 기능을 제공한다.

● 역할

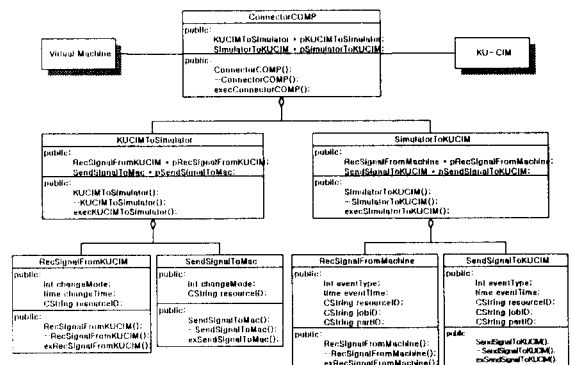
커넥터 컴포넌트는 시뮬레이터와 KU-CIM 사이의 정보 흐름을 담당하는 인터페이스로서 KU-CIM에서 수립된 스케줄을 기반으로 가상기계에 작업지시를 전달하고 가상기계에서 발생한 이벤트를 KU-CIM에 전달하는 역할을 수행한다. KU-CIM에 전달된 이벤트 정보는 KU-CIM에서 의사결정을 위한 기본정보로 사용된다.

● 컴포넌트간 관계



<그림 23> 커넥터 컴포넌트 관계도

● 클래스 다이어그램



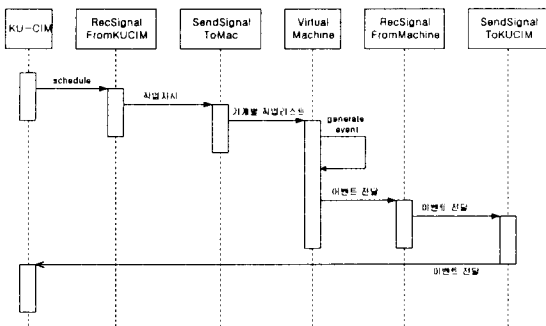
<그림 24> 커넥터 컴포넌트 클래스 다이어그램

(5) 타이머(timer) 컴포넌트

● 목적 (Intent)

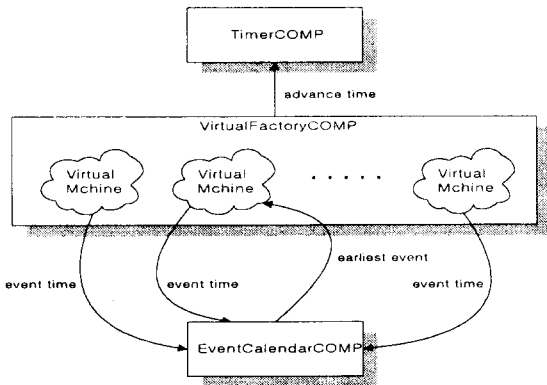
시물레이션의 진행을 위한 시간 전진(time advance)을 담당하며, 타이머 컴포넌트에서 시간의 전진은 실제 세계 시간과 가상 시간이 공존하는 혼합 시물레이션으로의 진행되어야 한다.

● 실행순서 다이어그램



<그림 25> 커넥터 컴포넌트 실행순서 다이어그램

● 컴포넌트간 관계

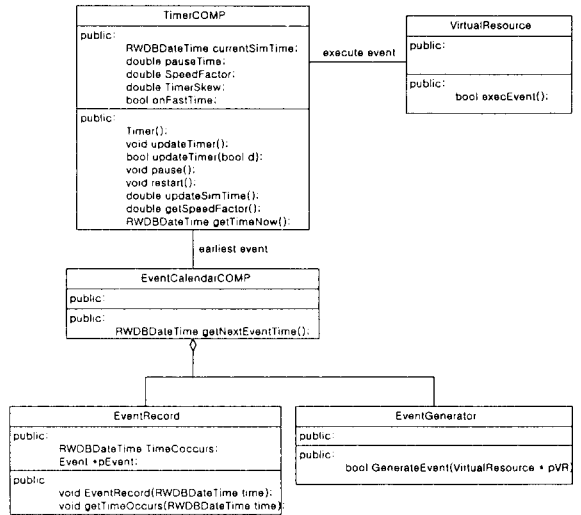


<그림 26> 타이머 컴포넌트 관계도

● 역할

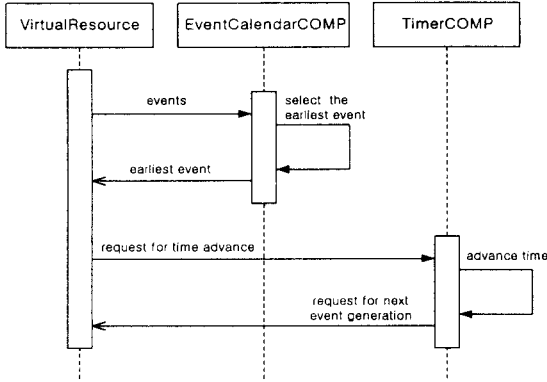
시물레이션의 진행을 위해 시물레이션 시간을 이벤트 단위로 전진시키는 기능을 수행한다. 본 연구에서 제안하는 시물레이터는 실제 세계 시간(real-world time)과 가상시간(virtual time)이 공존하는 혼합(hybrid) 시물레이션이므로 타이머 컴포넌트는 이를 지원할 수 있도록 설계되어야 한다. 다시 말하면 KU-CIM에서 의사결정이 진행되는 동안과 사용자가 시스템과 상호작용(interaction)을 수행하고 있는 동안, 그리고 데이터베이스에 접근하여 작업을 수행하는 동안은 시간전진이 실제 세계 시간으로 진행되어야 하고 시물레이션이 진행되는 동안의 시간전진은 가상시간으로 진행되어야 한다.

● 클래스 다이어그램



<그림 27> 타이머 컴포넌트 클래스 다이어그램

● 실행순서 다이어그램



<그림 28> 타이머 컴포넌트 실행순서 다이어그램

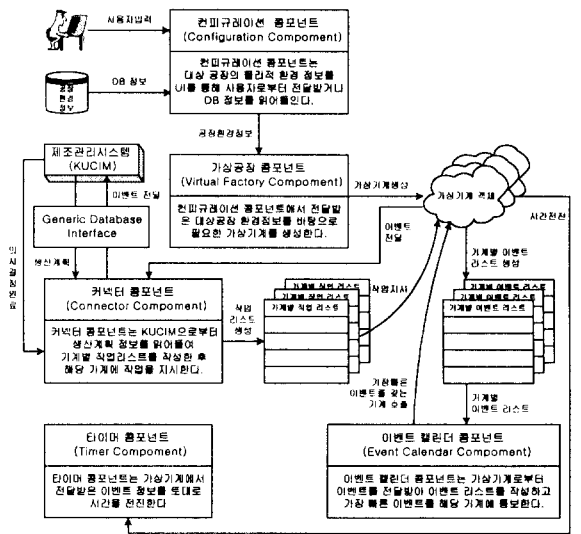
4. 적용 사례

본 연구에서는 자동차 부품 중의 하나인 트랜스미션(transmission)을 생산하는 P공장을 대상으로 개발된 객체지향 제조관리 시스템인 KU-CIM을 평가하기 위해 객체지향 시뮬레이터를 사용하였다. 대상 제조시스템인 P공장은 개별공정(job shop) 형태의 생산 방식을 취하는 9 대의 기계로 구성되어지며 10여 종의 부품을 생산하는 시스템이다. 본 연구에서 제시하는 객체지향 시뮬레이터는 KU-CIM을 다음과 같은 관점에서 평가하였으며 이로 인해 KU-CIM이 실제 제조현장에 구축되지 않은 상태에서 시스템의 성능을 검증할 수 있도록 하였다.

- KU-CIM에서 수립된 생산계획의 타당성 검증
 - 수립된 생산 계획의 적용 가능성 평가
 - 수립된 생산 계획이 최적인지 평가
 - 수립된 계획에 의한 납기 준수율 평가
 - 생산계획 수립 알고리즘의 평가
- KU-CIM에서 수립된 재계획의 타당성 검증
 - 수립된 재계획의 적용 가능성 평가
 - 수립된 재계획이 최적인지 평가
 - 수립된 재계획에 의한 납기 준수율 평가
 - 재계획 수립 알고리즘의 평가

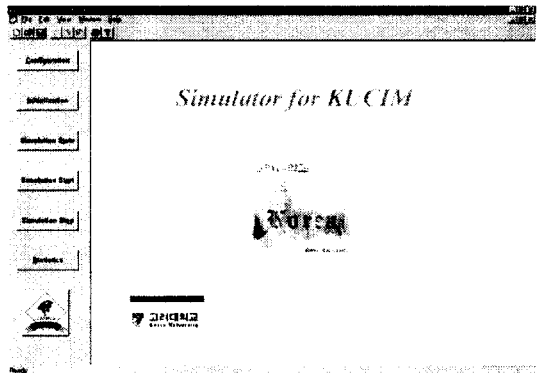
- 이상상황 발생 시 KU-CIM에서 수립된 의사결정의 효율성 검증
 - 수립된 의사결정의 타당성 평가

본 연구에서 개발한 객체지향 시뮬레이터는 앞에서 설명한 컴포넌트들을 기반으로 KU-CIM을 평가하기 위한 시뮬레이션을 진행하게 되는데 시뮬레이션이 진행되는 시나리오는 <그림 29>와 같다.



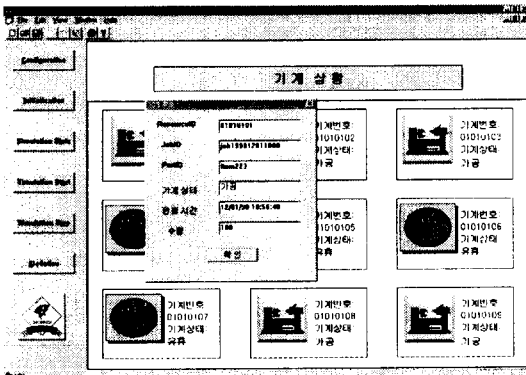
<그림 29> 시뮬레이터 진행 시나리오

개발된 객체지향 시뮬레이터는 <그림 30>과 같으며 화면 왼쪽에 있는 버튼을 이용하여 시뮬레이션을 진행하게 된다.



<그림 30> 객체지향 시뮬레이터

컨피규레이션 버튼은 제조시스템의 물리적 환경 정보를 입력받는 것으로서 데이터베이스를 통하여 입력받을 것인지 아니면 사용자 화면을 통해 사용자로부터 직접 입력받을 것인지를 선택할 수 있다. 초기화(initialization) 버튼은 시물레이션을 진행할 기간과 시스템 파라미터 값을 입력받을 때 사용된다. 초기화가 끝난 후 시물레이션 시작 버튼을 누르면 시물레이션이 진행되는데 주문입력, MPS 수립, MRP 수립, 생산계획 수립 등과 같은 의사결정은 KU-CIM에 의해 수행되고 시물레이터는 수립된 스케줄에 의해 작업을 진행하여 이벤트를 생성해 낸다. 시물레이션이 진행되는 동안 가상기계에서의 작업진행 상황은 <그림 31>과 같은 형태로 나타나게 되는데 해당 기계를 사용자가 선택하면 그 기계에 대한 자세한 정보가 나타난다.



<그림 31> 시물레이션 진행상황

Simulation Results

Machine ID	Utilization	납기완수율	No of tardy	No of break	Mean RPT
01010101	98.10	98.00	5	2	13
01010102	91.24	97.20	6	2	15
01010103	92.20	96.35	7	1	21
01010104	93.50	95.15	8	1	17
01010105	91.12	97.42	6	1	8
01010106	95.50	93.80	9	0	0
01010107	98.45	91.36	12	0	0
01010108	94.05	94.55	8	1	12
01010109	96.55	92.10	10	0	0

OK

<그림 32> 시물레이션 수행 결과

시물레이션이 완료된 후 통계량(statistics) 버튼을 누르면 <그림 32>와 같은 시물레이션 수행결과가 나타나게 되는데 시스템 개발자는 이 결과를 토대로 KU-CIM에 대한 평가를 수행하게 된다.

5. 결 론

본 연구에서는 객체지향 제조관리 시스템의 성능을 평가하기 위한 방법으로 객체지향 시물레이터를 제안하였다. 일반적으로 개발된 제조관리 시스템을 평가하기 위해서는 실제 제조현장에 시스템을 구축하여 시험 가동하는 방법을 주로 사용하는데 이러한 방법은 시스템 개발기간과 비용을 증가시키는 문제점을 나타낸다. 따라서 본 연구에서는 개발된 제조관리 시스템을 현장에 구축하기 전 시물레이터를 이용하여 평가할 수 방법을 제안하였다.

객체지향 제조관리 시스템의 성능을 평가하기 위한 시물레이터는 제조관리 시스템과 유기적으로 연결되어 정보를 교환할 수 있어야 하며 객체지향 제조관리 시스템이 제조환경의 변화에 유연하게 대처하기 위해 개발된 시스템이므로 이를 평가하기 위한 시물레이터도 제조환경의 변화에 유연하게 대처할 수 있어야 한다. 본 연구에서 제안하는 시물레이터는 변화에 유연하게 대처할 수 있도록 객체지향 방법론을 이용하여 설계되었으며 제조관리 시스템과의 유기적인 연결관계 유지와 원활한 정보교환을 위해 공통 데이터 인터페이스를 사용하였다. 객체지향 시물레이터를 구현하기 위해서 본 연구에서는 유즈케이스 다이어그램, UML, 디자인 패턴, 클래스 다이어그램, 실행순서 다이어그램 등을 이용하는 객체지향 분석 및 설계를 수행하였으며 계층적 구조를 갖는 시물레이터 프레임워크를 개발하여 제조환경의 변화에 유연하게 대처할 수 있도록 하였다.

본 연구에서 제시하는 시물레이터는 기계 계층에서의 시물레이션을 수행하고 있기 때문에 기계의 수가 증가하는 경우 시물레이션의 진행이 느려지는 문제점이 발생한다. 따라서 상위계층(공장-샵(shop)-셀(cell) 계층)에서 기계계층의 특성을 추상화(abstraction)하여 나타낼 수 있는 시물레이터의 개발로 시물레이션 수행시간을 단축시킬 필요가 있

으며 시뮬레이터가 표현하는 작업진행 상태의 수준을 조절할 수 있는 시뮬레이터의 개발이 필요하다.

참고문헌

- [1] 권일명, 김창욱, 전진, 김성식, "데이터 객체의 일반화를 통한 공통 데이터 인터페이스의 구현", *대한산업공학회/한국경영과학회 '98 춘계학술발표 논문집*, 1998
- [2] 김창욱, 이홍철, 최인찬, 김성식, "KU-CIM : 객체지향형 CIM 구축 프로젝트", *대한산업공학회/한국경영과학회 '97 춘계학술발표 논문집*, 1997, pp. 561~564
- [3] 전진, 백준걸, 김창욱, 김성식, "KU-CIM Project에서의 객체지향 설계", *한국경영과학회/대한산업공학회 '97춘계학술대회 발표논문집*, pp 565~568
- [4] Adiga, S., *Object-oriented Software for Manufacturing Systems*, Chapman & Hall, 1993
- [5] Barcio, B. T., Ramaswamy, S., and Barber, K. S., "An Object-Oriented Modeling and Simulation Environment for Reactive Systems Development", *The International Journal of Flexible Manufacturing Systems*, Vol. 9, pp 51-80, 1997
- [6] Booch, G., Jacobson. I., and Rumbaugh, J., *UML Distilled Applying the Standard Object Modeling Language*, Addison Wesley, 1997
- [7] Booch,G, *Object-Oriented analysis and Design*, Addison-Wesley,1994
- [8] Bruno, G. and Agarwal, R., "Modeling the Enterprise Engineering Environment", *IEEE Transactions on Engineering Management*, Vol. 44, No. 1, pp 20-30, 1997
- [9] Eriksson, H. E and Penker, M., *UML Toolkit*, John Wiley and Sons Ltd, 1997
- [10] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns : Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995
- [11] Haddock, J., "A Simulation Generator for Flexible Manufacturing Systems Design and Control", *IIE Transactions*, Vol. 20, No. 1, pp 22-31, 1988
- [12] Jacobson, I., *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.
- [13] Lawrence, P., *WORKFLOW HANDBOOK 1997*, John Wiley and Sons Ltd, 1997
- [14] Nadoli, G. and Biegel, J. E., "Intelligent Manufacturing-Simulation Agents Tool (IMSAT)", *ACM Transaction on Modeling and Computer Simulation*, Vol. 3, No. 1, pp42-65, 1993
- [15] Najmi, A. and Lozinski, C., "Managing factory productivity using object-oriented simulation for setting shift production targets in VLSI manufacturing", *Proceedings of Autofact '89 Conference*, 3:1 -3:14.
- [16] Nakano, M., Sugiura, N., Tanaka, M., and Kuno, T., "ROPSII: Manufacturing System Simulator with Object-Oriented Simulation Language", *New Directions in Simulation for Manufacturing and Communications*, pp 493~498, 1994
- [17] Yih, Y., Rabelo, L. C., and Jones, A., "A Hybrid AI/Simulation Framework for FMS Scheduling", *New Directions in Simulation for Manufacturing and Communications*, pp 204-210, 1994

● 저자소개 ●



백준걸

1993년 고려대학교 산업공학과(공학사)

1995년 고려대학교 산업공학과(공학석사)

현 재 고려대학교 산업공학과 박사과정



오훈연

1997년 한성대학교 산업공학과(공학사)

1999년 고려대학교 산업공학과(공학석사)

현 재 (주)한미데이터 연구원



신현준

1995년 고려대학교 산업공학과(공학사)

1997년 고려대학교 산업공학과(공학석사)

현 재 고려대학교 산업공학과 박사과정



김성식

1972년 고려대학교 기계공학과(공학사)

1974년 고려대학교 산업공학과(공학석사)

1976년 S.M.U 산업공학과(공학석사)

1979년 S.M.U 산업공학과(공학박사)

현 재 고려대학교 산업공학과 교수



이홍철

1983년 고려대학교 산업공학과(공학사)

1988년 University of Texas Arlington 산업공학과 (공학석사)

1993년 Texas A&M University 산업공학과(공학박사)

현 재 고려대학교 산업공학과 조교수