

제조 시스템 운용시스템 개발을 위한 객체 지향 기법

송영조[○], 홍성수
서울대학교 전기공학부

Object-Oriented Programming for the Development of Manufacturing System Operation System

Youngjo Song, Seongsu Hong
School of Electrical Engineering Seoul National University

요 약

본 논문에서는 확장성과 재사용성을 갖춘 제조 시스템 운용 시스템 개발을 위해 객체 지향 기법을 사용한다. 객체 지향 기법 중에서 소프트웨어 개발 방법론인 ROOM (Real-Time Object-Oriented Method)을 이용하여 모델링하고, 실제 목적 운영체제에서 사용할 수 있는 실행화일을 생성하는 방법을 제안한다. 본 연구에서는 제조 시스템 운용 소프트웨어 개발을 위해 ROOM에 기초한 CASE Tool인 "ObjecTime"을 사용한다. ObjecTime의 개발환경은 ROOM에 기초를 둔 모델링으로부터 실제 시스템에서의 실행화일의 생성에 이르는 모든 과정을 지원한다. 이런 기능으로 인해 제조 시스템 운용 시스템 개발을 위해 모델링 과정에서부터 실제 프로그램을 만드는 과정 사이에 발생할 수 있는 오류를 최소한으로 줄일 수 있다.

1. 서론

최근 들어 제조 시스템은 점점 다양해지는 소비자의 기호를 반영하기 위해 유연성이 요구되고 있으며 또한 경쟁력을 유지하기 위해 높은 생산성이 요구되고 있다. 이러한 요구 조건의 변화에 따라 메카트로닉스 기술과 소프트웨어 기술을 바탕으로 FMS(Flexible Manufacturing System)에 관한 많은 연구가 진행되어 왔다. 특히 전체 시스템을 운용 제어하는 운용 소프트웨어는 FMS의 가장 핵심적인 부분이며 시스템의 성능에 큰 영향을 줄 수 있다. 아울러 운용 소프트웨어는 계속 발전하는 운용 기술과 생산 설비의 변화에 따라 최소의 시간과 경비로 대응할 수 있어야 한다. 이를 위해 FMS 운용 소프트웨어 분야에서는 확장성, 호환성, 재사용성 그리고 이식성을 보장할 수 있는 객체 지향 기법이 도입되었다.

Smith[1]는 FMS의 다양성과 복잡성으로부터 운용 소프트웨어의 '재사용성'의 필요성을 보이며 FMS 제어를 위한 객체 구조를 제안하였다.

Adiga[2][3]는 제조 시스템의 운용을 위한 객체 구조를 제안하고 이를 Objective-C를 이용하여 BLOCS-M이라는 제조 시스템용 시뮬레이션 패키지로 구현하였다. 이는 객체 지향 기법의 특징을 이용하여 대상 시스템을 다양하게 구성할 수 있으며, 이의 예로 반도체 생산 시스템의 모델링에 적용하였다. 또 Bodner[4] 등도 제조 시스템을

위한 객체를 설계하고, 이를 이용한 시뮬레이터를 개발하여 반도체 제조나 FMS의 deadlock회피를 위한 모델링에 적용하였다.

본 연구에서는 확장성과 재사용성을 보장하는 제조 시스템 운용 소프트웨어의 개발과정에서 사용하는 객체 지향 기법을 논의하고자 한다. 2절에서는 제조시스템의 요구조건과 제조시스템 운용 소프트웨어의 기본 구성을 간략히 소개하고, 3절에서는 운용 소프트웨어 개발에 이용하는 객체 지향 프로그램 기법에 대해 설명한다. 그리고 4절에서는 CASE Tool인 ObjecTime의 모델링과 실행환경에서 TargetRTS와 실행화일 생성에 대해 구체적으로 설명한다. 5절에서는 본 연구의 결론을 내린다.

2. 제조 시스템의 요구 조건과 제조 시스템 운용 소프트웨어의 기본 구성

IMOS는 유연 생산 시스템(FMS: Flexible Manufacturing Systems)을 위한 운용소프트웨어이다. 유연 생산 시스템은 생산품의 다양성뿐만 아니라 제조기계의 교체 등과 같은 환경의 변화에 대처해야 한다. 우선, 본 연구에서 대상으로 하는 제조분야의 FMS의 구조는 Fig.1과 같고 다음의 네 가지의 구성요소로 이루어진다.

- + 운용 컴퓨터 : 통신망과 데이터베이스 시스템을 포함
- + Machine : 머시닝 센터, 선반, 셋업 스테이션 기타

- + 자동창고(AS/RS) : 팔렛, 치구와 소재를 포함
- + AGV 또는 기타 물류 운반 장치 (컨베이어 시스템 등)

이때, 여기서 Machine으로 표현된 것들은 그 역할에 따라 머시닝 센터, 선반, 셋업 스테이션 등이 될 수 있다. 각 구성 요소에는 호스트와의 통신과 구성요소의 모니터링 및 제어를 위한 컴퓨터(PC)가 포함되며, 머시닝 센터인 경우에는 ATC(Automatic Tool Changer), APC(Automatic Pallet Changer)등이 포함되는 하나의 시스템으로 생각할 수 있다. 선반인 경우에는 다수의 스테이션과 로보트 등으로 구성되는 하나의 시스템으로 가정할 수 있다.

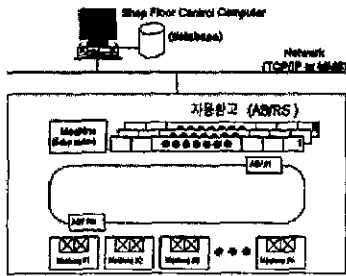


Fig.1 FMS Layout

3. 객체 지향 프로그램 기법

본 연구에서는 IMOS(Intelligent Manufacturing Operating System: 지능형 제조 운영 시스템)를 개발하기 위해 모델링과 프로그래밍 단계에서 객체지향(Object-Orientation)의 접근방식과 객체 지향 방법론인 ROOM[5](Real-Time Object-Oriented Modeling)을 도입한다.

ROOM은 이벤트 기반 분산 실시간 시스템(Event driven distributed real-time system)에 적합하게 개발된 방법론이다. ROOM의 기본 개념은 모델링 방향(modeling dimensions)과 추상화 단계의 체계(abstraction levels paradigm)로 대표된다. 모델링 방향은 시스템의 각 요소에 대한 구조, 그에 상응하는 시스템의 동적 특성, 상속성으로 나뉜다. 추상화 단계의 체계는 시스템의 상위 개념을 기술하는 개요 단계(schematic level)와 실제 구현에 대한 내용을 나타내는 상세 단계(detail level)로 구성된다. 개요 단계에서의 중요한 모델링 단위로서는 행위자(actor)가 있다. 행위자는 특정한 기능을 수행하는 객체(object)이다. 서로 독립적으로 동시에 수행 가능하며 메시지를 통하여 통신한다. 통신을 하는 행위자간에는 구조(structure)가 존재한다. 이러한 모델링 단위는 객체 지향(object-oriented)개념에 따라 클래스로 정의된다. 클래스는 행위자 클래스(actor class), 데이터 클래스(data class), 프로토콜 클래스(protocol class)가 있다.

IMOS 개발에서는 위에서 설명한 ROOM에 기초한 CASE Tool을 사용하였다. 소프트웨어 개발에 있어서 주요 핵심이 되는 문제는 지속적으로 고품질의 생산품을 만들어 내는 것이다. 특히 통신산업, 우주산업, 제조분야의 CIM과 같이 첨단산업에서 요구하는 방대하고 복잡한 소프트웨어 시스템을 개발, 유지하는 것은 체계적 관리 없이는 불가능하다. 이러한 상황에 대처하기 위해 CASE tool이 등장하게

되었고 그 중에서 본 연구에서는 ObjecTime을 사용한다. ObjecTime은 앞에서 설명한 것과 같이 실시간 시스템을 모델링하기에 적합한 ROOM(Real-Time Object-Oriented Method)에 기초하고 있다. ObjecTime의 개발환경은 ROOM에 기초를 둔 모델링으로부터 실제 시스템에서의 실행화일의 생성에 이르는 모든 과정을 지원한다. 이 과정은 마우스의 드래그 앤 드롭(drag-and-drop), C++ 코드의 자동 생성으로 간결화 된다. 즉, 프로그래머는 자기가 생각한 모델에서 프로그램을 만드는 과정에서 발생할 수 있는 오류를 최소한으로 줄일 수 있다.

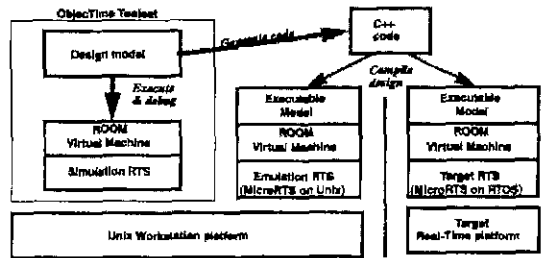


Fig.2 ObjecTime Model and Execution Environment

4. TargetRTS와 실행화일 생성

본 연구에서는 실제환경의 FMS에 IMOS를 적용하기 위해서 ObjecTime으로 설계된 디자인 모델을 바탕으로 C++코드를 자동생성하고 수정작업을 거쳐 IMOS의 실행화일을 만들어야 한다. 실행화일은 ObjecTime에서 제공하는 TargetRTS와 함께 목적시스템의 실시간운영시스템(Real-Time Operating System)에서 동작하게 된다. Fig.2은 이러한 ObjecTime 모델링과 실행환경의 관계를 보여준다. ObjecTime에서 라이브러리로 제공되는 TargetRTS는 ROOM 메소드를 지원하고 ROOM 모델을 실행하는데 여러가지 서비스를 제공하여 운영체제와 ObjecTime 모델을 연결하는 역할을 한다. 이런 TargetRTS를 실시간 운영체제에서는 ROOM virtual machine이라고도 하고, ObjecTime은 다양한 플랫폼의 TargetRTS를 제공하고 다른 플랫폼에 대해서도 이식이 가능하다. 본 연구에서는 Windows NT를 목적 시스템의 운영체제로 선택하였고, 실행화일은 ObjecTime에서 자동생성된 C++코드가 Windows NT에 맞게 이식된 TargetRTS에서 제공하는 서비스를 받아서 실행된다.

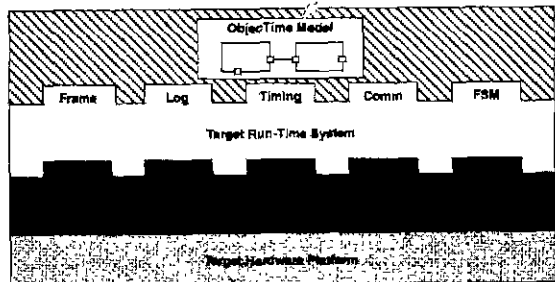


Fig.3 Models running in the TargetRTS

예를 들어 실제 메모리는 TargetRTS의 frame 서비스를 통해 복

적 시스템의 운영체제와 연결되며, 화일은 log서비스, IPC는 통신서비스를 통해 연결된다. Fig.3은 TargetRTS가 모델과 운영체제를 연결하여 모델이 목적 시스템의 운영체제에서 동작하는 것을 보여준다. Fig.4는 ObjecTime 디자인 모델로부터 실행화일을 만드는 과정을 설명해 준다. C++코드를 생성하고 실행화일을 생성하는 과정은 그림에서 보여진 것과 같이 여러단계의 과정을 거쳐 이루어진다. 본 연구에서는 Microsoft Visual C++ compiler를 사용하였다. 먼저, 1부터 3과정까지 ObjecTime의 Toolset을 이용하여 C++ 코드와 컴파일을 위한 Make화일을 생성하는 과정을 나타내고, 4부터 6과정까지는 생성된 Make화일을 이용하여 라이브러리인 TargetRTS와 링크하여 C++코드를 컴파일하고 최종적으로 원하는 플랫폼에서의 실행화일을 얻는 과정을 나타낸다. 이런 기능을 통해서 모델링으로부터 프로그램을 만드는 과정에서 발생할 수 있는 오류를 최소한으로 줄일 수 있다.

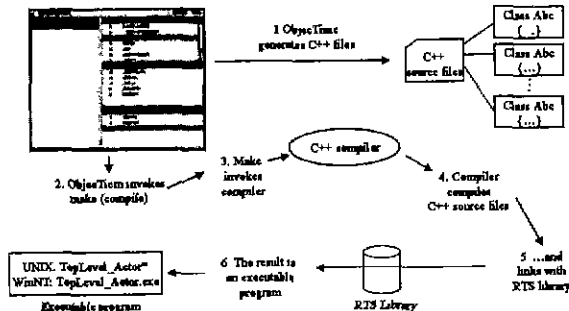


Fig.4 C++ Compilation Process

5. 결론

본 연구에서는 제조 시스템의 운용 시스템 개발에서 객체 지향 기법을 사용하여 시험용 FMS를 모델링하여 정의하였고, 이것을 대상으로 한 시험용 운용제어 소프트웨어의 C 코드를 CASE Tool인 ObjecTime을 이용하여 생성하고, 지금은 실행화일을 만들어 현재 시험 중에 있다.

향후에는 시스템 상태 모니터링 모듈과 그래픽 사용자 환경을 구축해서 IMOS에 포함시켜 제조 시스템 운용 소프트웨어를 완성하고, 실제 FMS에 설치할 것이다.

본 연구에서는 확장성과 재사용성을 보장하는 제조 운용 소프트웨어 개발을 위해 객체 지향 방법을 사용했다. 모델링으로부터 실행화일의 생성까지 지원하는 ROOM 방법론의 CASE Tool인 ObjecTime을 사용하여 프로그래머가 생각한 모델로부터 프로그램을 만드는 과정에서 발생할 수 있는 오류를 최소한으로 줄일 수 있다.

References

1. Smith, J. S. and Sanjay B. Joshi., 1992, Reusable Software concepts Applied to the Development of FMS Control Software. Int. J. Computer Integrated Manufacturing, Vol. 5, No. 3, pp.182-196
2. Adiga, S., 1989, Software Modeling of Manufacturing

Systems: A Case for an Object-Oriented Programming Approach. Annals of Operation Research, 17, pp.363-378

3. Adiga, S., 1993, Object-Oriented Software for Manufacturing System, Chapman & Hall.

4. Bodner, D. A. and S. Narayanan, 1992, Object-Oriented Simulation to Support Modeling and Control of Automated Manufacturing Systems. Proc. of Western Multiconference Society for Computer Simulation, pp.59-63

5. Selic, B., G. Gullekson and P. T. Ward, 1994, Real-Time Object-Oriented Modeling, John Wiley & Sons, Inc