

CIM 구축 tool 로서의 네트워크 상의 분산시물레이션

김성식 배경한

고려대학교 공과대학 산업공학과

Distributed Simulation on a Network as a CIM developing tool

Sung-Shick Kim Kyoung-Han Bae

Dept. of Industrial Eng.

Korea University

ABSTRACT

This study proposes a CIM shell that is consisted with computers connected by a LAN. Since this system excludes production hardwares, physical movements of devices are represented by computer simulation. On the other hand, softwares that dictate system operation are real, and consume physical time. Thus the shell becomes half-real-half-simulation that requires events synchronization mechanism. This study shows the building procedure of such shell as well as an example of application of the shell.

1. 서론

CIM은 컴퓨터를 이용하여 생산시스템, 생산관리시스템, 판매 시스템 등의 기업을 운영하는데 필요한 부분시스템들을 통합한 시스템[4]이다. 이와같이 여러 분야가 합쳐진 거대한 시스템이므로 시스템을 구축할 때 방대한 비용이 소모되며, 시스템 구축과 시스템 구축용 모형설계가 어렵다. 그리고, 시스템 구축 시에 시행착오가 발생할 가능성이 많고, 한번 시행착오가 발생하면 막대한 비용이 소모되는 위험이 있다.

그러므로, 시스템 구축을 용이하게하고, 시행착오를 최소화하여 이로인한 비용 발생을 최소화하며, 시스템을 구축할 때에 발생하기 쉬운 호환성 문제를 해결할 수 있는 CIM 구축도구가 필요하다. 그리고, 본 연구는 그러한 도구를 제시한다.

본 연구에서 제안하는 CIM 구축도구는 컴퓨터 배치, LAN, 데이터베이스 등의 CIM의 핵심은 실제시스템과 같이 구축하고, 통신망에 연결될 물리적 장비의 동작은 그 장비를 통제하는 컴퓨터에서 이산형 시물레이션을 이용하여 모의로 처리하는 것이다.

이 도구를 기초로 하여 운영소프트웨어를 개발및 개선 보정해 나가며 이것이 완성되면 장비를 구축된 통신망에 직접 연결하여 자동화공장을 구축하게된다.

본 연구의 아이디어는 실제 시스템의 운영소프트웨어는 모형을 이용하여 만들고 실제 시스템을 구축할 때는 이 모형을 그대로 옮기는 것이다. 즉, 시스템의 구축, 검증, 수정 보완은 모형에서 하며, 모형 CIM이 만족하는 수준에 이르면 이 모형을 그대로 현장에 이식하고 여기에 실제로 작업할 장비를 장착하여 CIM을 구축하는 것이다.

이러한 모형 CIM에서 장비의 동작은 이산형 시물레이션으로 처리하기 때문에 장비의 동작을 묘사하는 이벤트를 발생시키는 시간은 실제 동작시간에 비하면 매우 짧다. 그러나, 시스템을 운영하는 소프트웨어들은 실제상황이기 때문에 물리적으로 실제 시간을 소모하며 동작한다. 이 경우에 발생하는 문제는 모형에서 발생하는 사건의 차례를 실제의 차례와 동일하게 일치시키는 것이다. 이 문제는 Parallel-Processing과 multi-computer Processing 등에서 일어나는 문제이며 컴퓨터구조 선정[3], 이산형 분산시물레이션을 대상으로 이루어진 연구[2][5]는 많지만 본 연구모형과 같이 이산형 이벤트작업과 실시간 작업이 혼합된 분산시물레이션을 대상으로 한 연구는 Ray[6]의 컴퓨터 통신망의 성능평가를 위한 연구를 제외하면 보고된 연구가 없는 실정이다. 본 연구는 이에 대한 연구를 뒤에 제시하고 있다.

이 도구를 이용하면 운영시스템에 대한 충분한 검토가 이루어진 후에 요소장비를 연결하므로 시행착오로 인한 경제적 손실을 최소화할 수 있으며, 방대한 크기의 시스템일지라도 적은 공간에서 구축이 가능하므로 기존에 시스템을 운용하고있더라도 시스템의 운영에 방해를 주지않는다.

본 연구의 CIM 구축도구를 사용하여 CIM을 구축할 때는 다음

과 같은 절차를 따른다.[1]

절차 1 : 공장 운용방법을 결정한다.

절차 2 : 상세설계를 바탕으로 장비동작을 분석하고, 공장에서 사용할 컴퓨터의 역할을 결정한다.

절차 3 : LAN을 이용하여 컴퓨터를 결합한다.

절차 4 : 데이터베이스를 선정하고 구축한다.

절차 5 : Master Clock, 운용프로그램, 장비동작모의프로그램 등 분산시물레이션을 이용하여 운영시스템 모형을 완성한다.

절차 6 : 장비를 결합하여 시스템을 완성한다.

CIM을 구축하는 단계들 중에서 가장 중요하고 작업이 많은 단계는 절차 5이다. 이 단계에서는 관련 소프트웨어들을 개발하는데는 많은 시간과 노력이 필요하다. 그런데, 모듈화 설계가 잘 된 표준화된 소프트웨어가 있다면 소프트웨어를 개발하는데 소모되는 시간과 노력을 절감할 수 있다. 저자는 본 연구와 관련하여 FMS용 소프트웨어를 모듈화하여 개발하고 있으며 실제 회사에 적용할 때는 최소한의 수정으로 적용가능하도록 설계되어 있다. 장비를 결합하는 단계인 절차 6에서는 구축된 통신망에 연결 가능한 자동화 장비가 도입되지 못하더라도 시스템을 완성할 수 있다. 물론, 적합한 장비가 준비되어 결합되어야 하지만, 기계가 자동화되지 못하여 인간이 직접 조작해야 하는 경우에는 인간이 그 기계를 담당하는 컴퓨터로부터 작업명령을 받아서 작업을 수행하고 작업결과를 해당 컴퓨터에게 보고하는 방법을 취하여 CIM이 목표하는 효과를 거둘 수 있다. 자동화되어질 장비는 회사의 자금계획을 고려하여 순차적으로 도입할 수 있으므로 무리한 초기의 장비투자를 피할 수 있다.

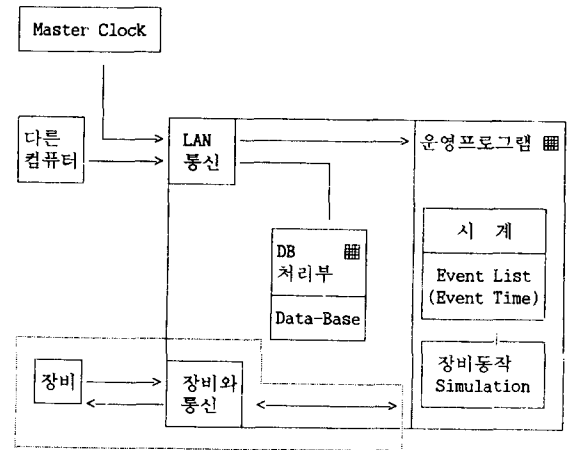
본 연구는 서론에 이어 2장에서는 CIM 시스템 구축용 분산시물레이션의 구성과 특성에 관하여 설명하고, 3장에서는 분산시물레이션을 구축할때 가장 필요한 시간동기화 방법에 대하여 논한다. 4장에서는 본 연구에서 제안하는 CIM 구축도구를 사용하여 개발한 FMS에 관한 사례를 소개하고 5장에서 결론을 맺는다.

2. 분산시물레이션의 구조

CIM 개념 하의 시스템은 기업의 전략 수립에서부터 대일정계획, 소일정 계획의 생산기획기능, CAE, CAD/CAM, CAPP 등의 직접 생산기능, 공장 자체의 물류 관리와 실행, 공구 관리와 활용, NC 프로그램 관리, 전송 및 실행 등의 최말단의 가공에 관련된 기능이 모두 통합되어야만 진정한 의미의 CIM[7]이라고 한다.

본 연구에 설명하고자하는 시물레이션의 구조에 이들을 모두 포함시킬 경우 내용이 방대한 반면 산만해진다. 따라서 본 연구에서는 CIM의 축소판인 FMS를 대상으로하며 이를 CIM으로 확장하여도 동일한 개념 적용이기 때문에 내용에는 큰 차이가 없다.

CIM 구축도구로서 분산시물레이션 모형을 꾸밀 경우 컴퓨터가 갖추어야할 소프트웨어의 구성은 시간소모의 과정과 기능 별로 나눌 수 있다. 시간소모의 과정에 의하여 소프트웨어를 분류하면 ①물리적 시간을 소모하며 실행되는 부분, ②시물레이션으로 사건을 발생시키는 부분, ③사건발생 동기화를 위한 시간처리부분으로 나눌 수 있다. 또한, 기능 별로 소프트웨어를 분류하면 ①LAN 통신프로그램, ②자신에게 부여된 임무(공장 통제임무, 작업장 통제임무)를 수행하는 운용프로그램, ③Database와 이의 관리를 담당하는 프로그램으로 나눌 수 있다. 이들의 관계를 표시하면 그림 1 과 같다.



※ ㉑ : 다른 컴퓨터에게 통신을 보내는 통신 프로그램.

※ 점선 안은 실제 장비를 결합하였을 경우의 시스템의 구성.

그림 1 : 분산시물레이션 모형의 구성

그림 1 에서 운영프로그램 내에 설치된 시계는 시물레이션에서 현재의 시간을 표시하며 모든 사건의 차례를 정하는 기준이 된다. Master Clock은 공장 전체의 흐름을 통제하는 컴퓨터에 별도의 프로그램으로 설치하며, 여기서 시계를 실시간으로 전진시킬 것인가, 아니면 시물레이션에서 발생한 다음 사건의 이벤트시간으로 전진시킬 것인가를 결정하고, 결정된 현재시간을 모든 컴퓨터에게 통신을 이용하여 알린다.

장비에서 발생하는 사건들(작업의 시작과 종료, 고장의 발

과 회복, 공구의 파손과 교환 완료)을 모의로 표현하여 발생시키는 시뮬레이션 프로그램은 운영프로그램 내에서 부프로그램(sub-program)으로 만들어진다. 이 프로그램은 고장발생과 회복, 공구의 파손과 같이 이벤트를 자체에서 발생시키는 경우와 기계의 작업과 같이 운영프로그램의 명령과 통제부분에서 작업을 명령받아서 이벤트를 발생시키는 경우로 나뉘는데 전자의 경우는 해당 이벤트 시간이 되었을 때 새로운 이벤트를 발생시키고, 후자의 경우에는 운영프로그램 내부의 명령에 의하여 이벤트를 발생시킨다. 이들은 이벤트리스트에 기록되며 이벤트시간은 LAN을 통하여 Master Clock에게 전달된다.

운영프로그램에서 시뮬레이션을 제외한 나머지 부분(job scheduling, Sequencing, NC 프로그램 관리 및 Down-loading, 공구관리, DB 검색 및 갱신, 통계처리, 작업명령, ...)들은 모두 실제 CIM에서 사용하는 것이며, 통신프로그램, DB 관련 프로그램

들 역시 실제 상황에서 운영되는 프로그램들이다. 그러므로, 이들의 작동은 물리적 실시간을 소모한다.

그림 1에서 보듯이 CIM 시스템을 운영하려면 한 컴퓨터 내에서 여러 프로그램이 동시에 구동되어야하므로 시뮬레이션 모형은 Multi-tasking 또는 Multi-processing 하에서 운영되어야 한다. 그리고, 각 프로세스는 서로 긴밀한 대화를 나누므로 내부통신(IPC, Inter-Process Communication)이 필수적이다. 그러므로, 이를 지원해주는 OS가 필요하다. 통상, FMS를 구축할 때는 단일 프로세스를 지원하는 MS-DOS상에서 인터럽트를 사용하여 한번에 하나의 프로세스를 처리하는 방법을 사용하고 있으나 이는 시스템이 간단한 경우에 제한적으로 사용할 수 있을 뿐이며 CIM 개념 하의 시스템을 구축할 때는 이 방법의 사용은 현실적이지 못하다. Multi-tasking, Multi-processing, 내부통신을 행할 수 있는 OS는 UNIX[8]가 대표적인 시스템이다.

지금까지 언급한 Distributed Hybrid-Simulation인 CIM구축도구의 성질을 언급하면 다음과 같이 정리할 수 있다.

① CIM 시스템의 운영상황을 정확하게 표현하므로 CIM의 특성을 동일하게 갖는다.

② CIM 시스템의 통신형태는 한 컴퓨터에서 여러 다른 컴퓨터에게 작업을 요구하는 경우와 순서에 입각하여 한 컴퓨터에서 작업이 완료된 후에 다른 컴퓨터로 작업이 이전되는 경우가 있으며, 전자의 경우에는 여러 컴퓨터가 서로 긴밀한 관계(mutually dependent)를 가지고 작업을 하는 병렬형 시뮬레이션(

Parallel Simulation)[3]이 된다.

③ 장비의 동작을 제외한 모든 작업은 연속적인 시간이 흐르는 실제작업이며 장비의 동작은 이벤트 시간이 흐르는 시뮬레이션 작업이므로 시간동기화 작업이 필요하다.

④ 실제시스템으로 전환할때는 Master Clock과 장비동작시뮬레이션프로그램을 제거하고 실제장비를 연결하는 작업만 하면되므로 모형시스템을 실제 시스템으로 전환하기 쉽다.

⑤ CIM 시스템을 정확하게 표현하므로 실제 시스템을 구축하지않고도 문제의 도출과 해결이 쉽다.

⑥ 모형 개발에 사용된 H/W와 S/W를 실제 시스템에서 직접 사용하므로 CIM 시스템을 구축할때 장비구입비용 외에 여타의 비용이 발생하지 않는다.

3. 시간동기화 방법

본 연구에서 개발하는 분산시뮬레이션 모형은 이산적으로 흐르는 이벤트시간과 연속적으로 흐르는 실제 시간이 혼합된 분산 모형이므로 현재시간을 판단하기가 어렵다. 이런 모형에서 단일 표준화된 현재시간을 정하지않는다면 시스템 내의 시간은 혼란 상태가 되어 모형을 전혀 구동할 수 없게된다. 그러므로, 시스템을 대표하는 현재시간을 판단할 수 있는 규칙(rule)을 발견하고, 이 규칙을 이용하여 현재시간을 인지하고 시스템 내의 모든 컴퓨터에게 동기화시키는 알고리즘을 개발할 필요가 있다.

컴퓨터에서 이벤트 작업을 수행할 경우에는 이벤트의 종료시점을 컴퓨터 내의 현재시간으로 정한다. 그리고, 시스템 내의 모든 컴퓨터에서 실시간 작업을 할 경우에는 흐르고있는 시간은 모두 같다. 그러므로, 시스템 내의 각각의 컴퓨터에서 흐르고있는 시간이 서로 다른 경우는 한대 이상의 컴퓨터가 이벤트 작업을 수행하는 경우이다. 이런 경우에 대비하여 시스템을 대표하는 현재시간을 정의할 필요가 있다. 정의되어지는 현재시간은 모든 컴퓨터의 작업시간을 준수할 수 있어야한다. 그러므로, 현재시간은 다음과 같은 방법으로 판단한다.

방법 1 : 현재시간은 시스템에 속하는 각각의 컴퓨터에서 전진하고있는 시간 중에서 가장 작은 시간이다. 전진하고있는 시간에 공존하는 물리적인 실시간과 시뮬레이션에 의하여 발생되는 이벤트 시간의 크기를 비교하기위하여 기준시간 단위를 정하며, 기준시간 단위에 의하여 이벤트 시간을 발생시킨다.

방법 2 : 현재시간에 시스템내에서 실시간 작업이 없을 경우에는 이벤트 종료시간 중에서 가장 작은 시간을 다음의 현재시

간으로 정한다.

방법 3 : 시스템의 어느 한 곳에서 실시간 작업이 진행되고 있는 경우에는 실시간에 의하여 흐르고 있는 시간의 점들이 현재시간이 된다.

방법 4 : 시스템의 어느 한 곳에서 실시간 작업이 진행되고 있는 동안에는 발생되어진 이벤트들의 시간도 연속적으로 흐르게 한다.

현재시간을 정의할 수 있는 일반적인 방법을 정리하였다. 그런데 이들 정리만을 이용하여 시간동기화처리를 할 수는 없다. 그 이유는 다음과 같이 정리된다.

① MASTER CLOCK은 어느 컴퓨터에서 이벤트를 발생시키는지를 판단하는 능력이 없다. 그러므로, 시스템 내에서 실시간 작업이 전혀 없거나 실시간 작업이 종료된 직후에 Master Clock은 어느 한 컴퓨터가 이벤트를 발생시키는지를 판단하지 못하고 현재시간을 전진시키는 경우가 있다.

② 한 컴퓨터에서 발생시킨 이벤트시간보다 더 작은 이벤트시간을 나중에 발생시키는 경우가 있다.

이러한 문제들은 ①의 경우에는 컴퓨터에서 이벤트작업이 발생할 경우에 Master Clock에게 이벤트 발생 신호를 항상 보내고 Master Clock이 이 신호를 받으면 현재시간을 전진시키지 않고 기다리고, ②의 경우에는 컴퓨터에서 자신이 보유하는 이벤트시간들을 올림차순으로 정리하여 Master Clock이 현재시간을 알릴 때마다 최소의 이벤트시간을 Master Clock에게 알리는 방법으로 해결할 수 있다.

4. 분산시뮬레이션 개발 사례

고려대학교 산업공학과는 1990년에 상공부의 지원을 받아 FMS (K.U.FMS) 를 앞에서 언급한 CIM 구축도구를 사용하여 구축 중이다. K.U.FMS는 그림 2에서 보듯이 작업장(Cell) 2 개, 자동창고시스템, 무인자동운반차시스템으로 구성되어있으며 공장 통제용 컴퓨터는 Workstation 급인 MIPS 시스템을 사용하고, 나머지 컴퓨터는 IBM-PC 호환기종인 80386 시스템을 사용하고있다. 그리고, 통신시스템은 BUS 구조를 가진 Ether-Net을 이용하여 개발된 TCP/IP 통신시스템을 사용한다.

머시닝센터로 이루어진 작업장 1은 Conveyor가 부품 운반을 담당하며, Conveyor와 기계 사이에서는 자동팔렛교환기(Automatic Pallet Changer)를 사용하여 부품이 장착된 팔렛을 교환한다. CNC선반으로 이루어진 작업장 2에 대한 물류의 흐름은 무

인자동운반차를 이용하며, 무인자동운반차는 버퍼에 부품을 공급하고 버퍼로부터 제품을 받아서 필요한 장소에 운반한다. 그리고, 물류용 로봇이 버퍼와 기계 사이의 부품흐름을 담당한다. 무인자동운반차는 이를 담당하는 컴퓨터에서 통제하고, Conveyor 시스템의 통제는 작업장 1 통제용 컴퓨터에서 담당한다. 자동창고는 통제컴퓨터에게 입출고명령을 받아서 부품을 출고하고 제품을 입고한다. 부품이 작업장 2로 운반될 때는 무인자동운반차에게 부품을 올리고, 작업장 1로 운반될 때는 Conveyor의 I/O 장소에 부품을 올린다. 제품을 입고할 때도 출고할 때와 같은 물류장치를 이용한다.

그러므로, 작업장 1을 통제하는 컴퓨터는 CNC 선반과 로봇의 작동을 시뮬레이션으로 표현하였으며, 로봇이 선반에 작업물을 공급하는 과정과 선반이 로봇으로부터 작업물을 받는 과정, 선반이 제품을 로봇에게 옮기는 과정과 로봇이 선반에서 제품을 인수받는 과정은 각각의 모듈화된 시뮬레이션에 의하여 정확하게 표현하였다. 작업장 2를 통제하는 컴퓨터는 머시닝센터와 Conveyor의 작동을 시뮬레이션으로 표현하였다. Conveyor는 물품의 운반 뿐만아니라 버퍼의 역할도 하므로 운영 프로그램 내에서 실시간작업에 의하여 Conveyor의 관리를 담당하고 Conveyor의 작동만 시뮬레이션에 의하여 처리하였다. 자동창고를 통제하는 컴퓨터는 자동창고의 입출고 작업을 시뮬레이션으로 표현하였다.

이런 방법을 사용하여 장비의 동작을 모의로 표현하였고, 각 통제컴퓨터의 통신프로그램, 운영프로그램, 시간동기화 프로그램을 개발하고, Master Clock을 개발하여 공장통제컴퓨터에 설치하여 그림 1과 같은 분산시뮬레이션 모형을 완성하였다. 그리고, 현재는 실제 장비와 연결작업 중이다.

K.U.FMS의 운영소프트웨어 개발환경은 Multi-Processing이 가능한 UNIX 시스템과 XENIX 시스템을 혼합하여 사용하고 있고, 각 프로세스 간에는 이들 O.S가 지원하는 내부통신을 이용하여 대화를 하고있다.

5. 결론

본 연구에서 제안한 CIM 구축도구는 거대한 자동화공장의 운영시스템을 을 구축하는데 효과적인 방법이다. 그리고, 일반적인 시뮬레이션의 개념인 성능평가 기능과는 다르게 시스템 구축을 위하여 시뮬레이션을 사용한 점도 평가된다.

K.U.FMS를 구축하면서 얻은 경험적 교훈은 분산시뮬레이션 모형을 구축할 때 표준화 모듈로 구성된 소프트웨어를 설계하여

K.U. FMS

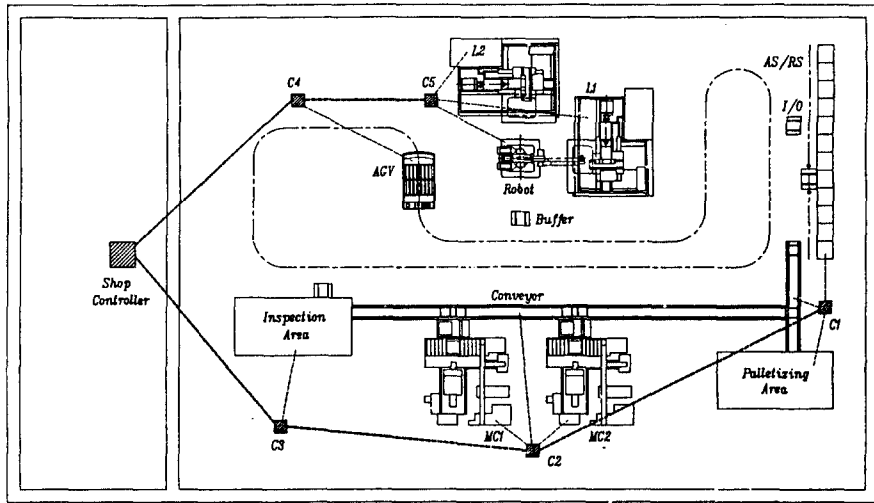


그림 2 : K.U.FMS의 구성

CIM의 형태에 따라 모듈조합만 바꾸면 즉시 사용할 수 있는 시스템이 개발되어야 할 것으로 생각된다.

참고문헌

- [1] 김성식, 배경한, "컴퓨터를 이용한 실제에 준하는 FMS구축", IE Interfaces 산업공학, 4권, 1호, 대한산업공학회, 1991.
- [2] Chandy, K. M., Misra, J., "Asynchronous Distributed Simulation via a Sequence of Parallel Computations", Commun. Assoc. Comput. Mach., pp. 198-206, Vol. 24, 1981.
- [3] Duncan, R., "A Survey of Parallel Computer Architectures", Computer, pp. 5-16, Vol. 23, no. 2, February 1990.
- [4] Groover, M. P., Automation, Production Systems, and Computer Integrated Manufacturing, Prentice-Hall, Inc., Engle wood Cliffs, N.J., 1987.
- [5] Misra, J., "Distributed Discrete-Event Simulation", Computing Survey, pp. 39-65, Vol. 18, No. 1, Mar. 1986.
- [6] Ray, A., "Networking for Computer-Integrated Manufacturing", IEEE Network, pp. 40-47, Vol. 2, No. 3, May 1988.
- [7] Scheer, A. W., CIM, Springer-Verlag Berlin Heidelberg, 1988.
- [8] Stevens, W. R., UNIX NETWORK PROGRAMMING, Prentice-Hall, Inc., N.J., 1990.