

## 실시간 분산 공정 제어 시스템

이도영    윤창진    권태웅<sup>2)</sup>  
 금성통신(주) 연구소

Real-Time Distributed Industrial  
 Process Control System

Do Young Lee, Chang Jin Yoon, Tae Woong Jeon  
 Gold Star Tele Communication R&D Laboratory

## Abstract

This article surveys techniques and issues related to real time process control system developed for industrial control applications. It covers the system architecture and software engineering issues such as the design of data structures, scheduling of asynchronous task activities, management of shared resources, handling of interrupt and implementing an user friendly man-machine interface. Also problems associated with implementing a real-time system that supports dynamic configuration of data base is addressed.

## I. 서론

제어 이론 분야에 있어서 시스템 모델링 및 최적 제어 이론을 포함한 상당한 양의 지식의 축적이 이루어져 왔다. 그러나 이러한 정교한 제어 이론들을 매우 복잡하고 다양하면서도 서로 유기적인 기능들을 요구하는 산업 공정 제어 운영에 실제로 적용하기 위해서는 컴퓨터/소프트웨어 공학, 시스템 공학 등으로부터 여러가지 시스템 설계 기법들을 종합적으로 도입 응용 하여야 한다.<sup>1)</sup> 한편 최근의 급격한 전자기술의 진보는 16비트, 32비트 슈퍼 마이크로 프로세서의 출현과 그의 고성능화, 저가격화를 가져왔고 이에 따라 각종 산업 공정 제어 분야에 마이크로 컴퓨터 기술을 이용한 분산 처리 시스템의 구현을 가능케 하고 있다.<sup>2)</sup>

공정 제어 시스템의 산업 응용 분야는 계속 다양, 복잡해져서 이제는 데이터 수집을 통한 단순한 감시 제어를 넘어, 각종 데이터의 분석 기록, 에너지 최적 관리,

사용자와의 다양한 입출력 서비스,<sup>3)</sup> 운영 환경 변화에 따르는 데이터 베이스의 온라인 생산, 변경 등의 기능이 요구되고 있다. 제어 시스템의 주요한 또하나의 요구 기능은 시스템 내부, 혹은 외부로부터 비동기적으로 발생하는 각종 이벤트(event)들의 실시간 처리가 가능해야 하는 점이다.<sup>5)</sup> 따라서 오늘날의 제어 시스템은 요구되는 시스템 수행 능력 (System performance)을 만족하면서 동시에 융통성 (flexibility), 재구성도 (configurability)가 뛰어난 구성을 갖도록 설계 되어야 한다.

본 논문은 각종 산업 설비의 제어 감시 및 관리를 위하여 개발된 분산 제어 시스템의 구성과, 개발에 적용되었던 설계 및 구현 방식을 기술하였다. 시스템 측면에서는 전체 하드웨어 및 소프트웨어의 분산 구성방식, subsystem 간의 통신 방식 등을 기술하였다. 소프트웨어 측면에서는 비동기적인 여러 task 들의 병렬 수행, 각종 event 들의 실시간 처리, 실시간 데이터 베이스 구성 및 관리, 인간-기계 인터페이스 (MMI) 등을 기술하였으며 구현된 시스템의 응답 시간 (response time), 처리 능력 (Throughput), 신뢰성도 다루었다.

## II. 시스템 기능 개요

실시간 제어 시스템은 각종 이벤트 (event) 들에 의해 기능이 수행되는 시스템이다. 시스템에서 감지되는 이벤트는 그 발생원인 (source)에 따라 크게 3가지로 분류할 수 있다.

- 1) 현장 (field)의 각종 상태 변화
- 2) 운영자 (operator)에 의한 시스템 제어명령 입력
- 3) 타이머 (timer)에 의한 주기적인 혹은 time-delayed 기능 수행

위의 이벤트들은 시스템에 의해 처리됨에 따라서 다시 여러가지 시스템 내부적인 이벤트 (triggered event)들이 발생, 수행하게 된다. 이벤트의 처리 내용도 그 수행

결과에 따라서 다음과 같이 3가지로 분류할 수 있다.

- 1) 현장 설비 제어 (field control)
- 2) 운영자에게 각종 정보 출력 (CRT display, 프린터 output)

3) 시스템 데이터 베이스의 수정 (상태 변화 보존)  
 따라서 실시간 제어 시스템은 외부 혹은 내부적으로 발생하는 각종 이벤트들을 주어진 시간내에 처리하여 그에 따라 시스템의 데이터 베이스를 수정 관리하고 분석하여 필요시 현장 제어 및 운영자에게 정보를 제공하는 시스템이라 할 수 있다. 본 장에서는 위에서 언급된 시스템 모델을 바탕으로 실제 산업 설비 공정 제어 운영을 위하여 개발된 시스템의 기능을 기술하였다.

제어 시스템의 가장 중요한 기능은 현장에서 감지되는 각종 원시 데이터 (raw data)를 사용자가 이해하거나 시스템 자체의 응용 소프트웨어가 처리할 수 있도록 논리적인 정보로 제공하는 것이다. 이를 위하여서는 원시 데이터는 여러가지 감지기 (sensor)들로 부터 디지털 신호로 입력 전송되어 시스템에 의해 엔지니어링 수치로 변환, 분석, 상태 체크 혹은 누적 (accumulate)되어 데이터 베이스에 저장되어야 한다.

공정 제어 시스템에 있어서 현장 정보 (field information)는 포인트 (point)를 기본 단위 개념으로 하여 구성되어 감시 혹은 제어 하게 된다. 포인트는 감시 제어 되는 데이터의 성격에 따라 다음의 5가지로 분류된다.

- 상태 표시 포인트 (on/off status indicator)
- 발정 제어 포인트 (on/off control point)
- 아날로그 입력 포인트 (analog input)
- 아날로그 적산 포인트 (accumulator)
- 아날로그 제어 포인트 (analog adjust point)

이러한 포인트들을 감시 제어 하기 위하여 포인트의 상태는 단순 상태 표시 (indication), 정상 상태 (normal), 비정상 (off-normal), 경보상태 (alarm), 불량상태 (trouble) 등으로 분석 처리된다.

각종 포인트의 데이터가 전송되면 그이전에 전송 기억된 데이터와 비교 분석되어 상태 혹은 수치에 있어서의 변화가 체크된다 시스템에서의 모든 상태 변화를 COS (chance of State)라 한다. COS가 발생하면 여러가지 방법으로 사용자에게 이를 알려야 한다. 그 중의 하나로서 COS 발생과 동시에 CRT 화면상에 해당 포인트의 위치와 상태가 디스플레이되며 프린터에 그 내용이 기록된다. 또한 상황감시판 (map board)의 해당 포인트의 적색 램프가 켜지거나 깜박임으로써 전체적인 상황을 한눈에 볼 수 있게 된다.

포인트의 상태 변화는 사용자에게 이를 알리는 것 외에도 시스템내의 각종 자동 제어 및 관리 기능을 활성화 (activate)시킨다. 그중의 하나가 연동제어 (interlo-

cking)로서 master로 할당된 포인트의 상태 변화가 감지되면 이에 대응하여 slave로 할당된 포인트들을 제어 하는 기능이다.

한편 주기적으로 수행되는 기능으로서 경향감시 (trend log)와 가동시간/사용량 적산 (runtime/consumption totalization)기능이 있다.

경향 감시는 일정한 시간 간격으로 포인트의 상태를 샘플링하여 프린터로 출력하여 시간의 흐름에 따른 포인트의 상태 변화를 감시 하는 기능이며 적산은 특정 설비의 가동시간 혹은 사용량을 누적 감시하는 기능이다.

이외에도 특정 설비 혹은 공정 제어에서 요구되는 응용 기능으로서 에너지 관리, 최적 전력 제어, 화학 공정제어 등이 있으며 이와 관련된 모든 데이터 베이스는 현장설비, 환경조건, 요구조건 들이 바뀔때 따라 쉽게 적용될 수 있도록 온라인 생산, 변경, 추가, 삭제가 가능하여야 한다.

위에서 기술한 시스템 기능의 관점에서 데이터의 흐름을 petri-net로 표현한 것이 그림1 이다.

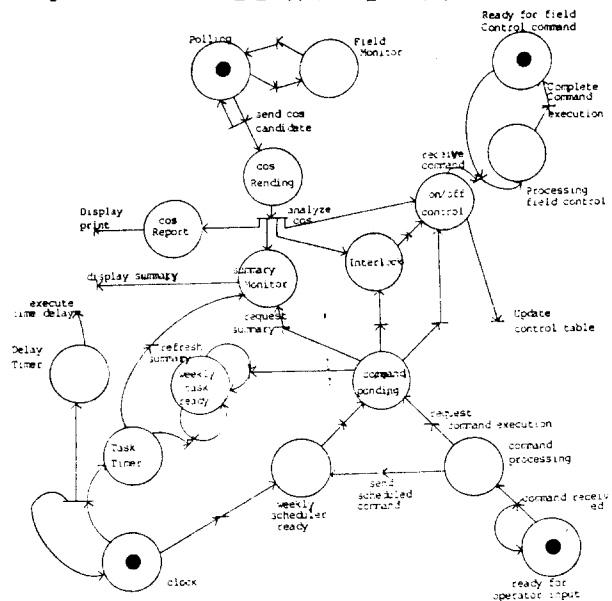


그림1. Functional Information Flow

### III. 시스템 구성

본 연구소에서 개발된 분산 제어 시스템의 전체 구성은 그림2 에 나타나 있는 바와 크게 3 부분으로 나누어져 있다. 즉 Host Computer 시스템, 데이터 수집/통신 시스템 (Data Concentrator), 원격 현장 처리장치 (Remote Terminal Unit)로 되어 있다.

Host Computer 시스템은 2대의 DC와 각각 통신선으로 연결되어 있어서 DC에 의해 수집된 각종 현장 데이터를 전송받고 또한 필요시 현장 설비 제어 명령을 DC에게 전송을 한다. Host Computer 시스템의 하드웨어 구성은 중앙 처리 장치 (CPU), 메모리, 입출력 제어 장치

(I/O Controller), 디스크 Controller, 통신 제어 장치 그리고 온전원과 입출력과 데이터 베이스의 저장 관리를 위한 각종 주변 장치들 (Color 그래픽 CRT, 프린터, 디스크 등)로 이루어져 있다.

감지하고 처리 수행해야 하는 실시간 제약 (real-time constraint)을 만족해야 하는 점이다. 이에 따라 실시간 제어 소프트웨어는 보통 여러개의 독립된 task들로 구성, 병렬 수행 (concurrent processing)

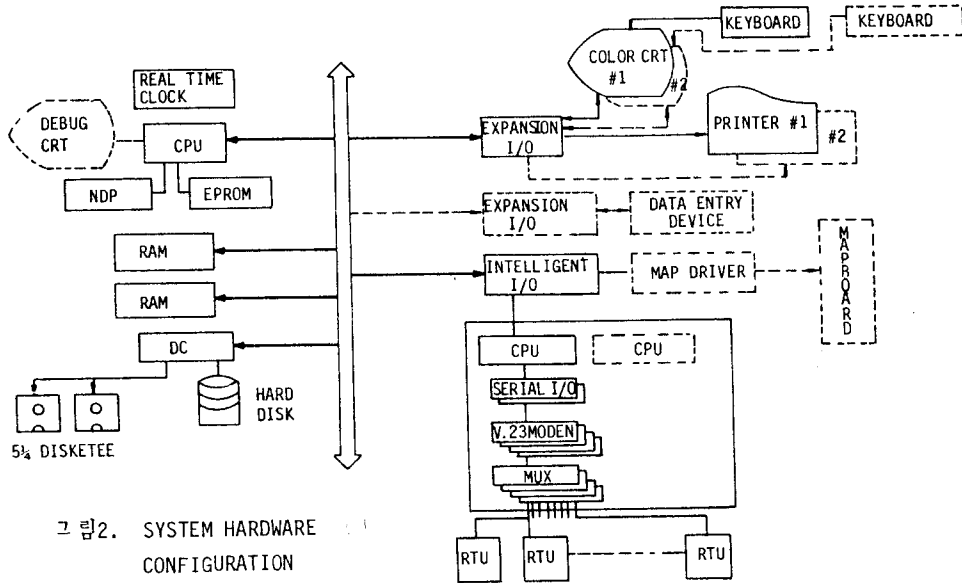


그림2. SYSTEM HARDWARE CONFIGURATION

데이터 수집/통신 시스템 (DC)은 각종 현장 설비 및 환경 데이터를 최대 48대의 원격 처리 장치(RTU) 들로부터 수집, 분석, 저장 하여 필요시 Host로 전송 하는 기능을 수행하도록 구성되어 있다.

각각의 DC는 마이크로 프로세서와 메모리가 Host와 별도로 내장 되어 있어서 방대한 양의 현장 데이터의 수집에 빠르며 Host의 처리부담을 덜어주며 (Load distribution) Host 다운 혹은 Host와의 통신 두절시 자체적으로 데이터 수집 기능을 계속함으로써 시스템의 신뢰성을 높여주도록 구성되어 있다.

원격 현장 제어 장치 (RTU)는 8비트 마이크로 프로세서가 내장된 현장 설비 감시 제어 모듈로서 자체적으로 포인트 감시 수행 (Scanning)을 하며 또한 DC로부터 데이터 요청 혹은 현장 제어 명령 메시지를 받아 이를 처리할 수 있도록 구성되어 있다.

RTU는 최대 96대 까지 현장 설비 곳곳에 설치되어 DC와는 모뎀 통신 라인으로 연결되어 있어서 시스템의 분산 처리를 가능케 하고 있다.

#### IV 시스템 소프트웨어

제어 시스템의 가장 주요한 특징 중의 하나는 비동기적이며 (asynchronous) 동시발생적인 (concurrent) 각종 이벤트들을 주어진 시간내에

하게 되며 그와 동시에 필요시 task 간의 통신 및 동기화 (synchronization)가 가능하도록 설계 되어야 한다.

본 연구소에서 개발된 제어 시스템 소프트웨어는 그림3과 같이 8개의 subsystem과 이들을 지원하는 리얼타임 오퍼레이팅 시스템으로 구성되어 있으며 각각의 subsystem은 다시 요구되는 기능의 성격에 따라 서로 유기적인 다수의 task들로 이루어져 있다.

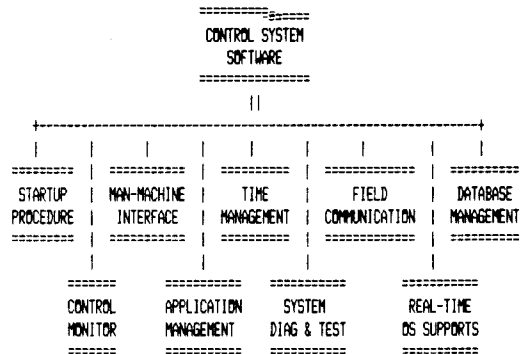


그림3. SOFTWARE CONFIGURATION

Startup procedure는 시스템이 처음 가동되어 정상운영 (normal operation)상태로 될 때까지의 초기화 과정을 수행하는 task로서 시스템 내의 모든 memory-resident task 들을 일정한 순서에 따라 초기

상태를 이루도록 하는 기능을 수행한다.

Man-machine 인터페이스는 쉽고 효율적인 사용자 운용 환경을 제공하는 subsystem으로서 복수 사용자의 동시 입력 처리 (multi-user supports)를 가능케 하는 console processor, 각종 정보를 사용자의 요구에 따라 CRT 화면 혹은 프린터로 출력하는 summary monitor, 다량의 데이터 입력을 용이하게 하는 DB 스크린 편집기 (DB screen editor), 이상 상태 (alarm status) 발생시 이를 우선 순위에 따라 CRT 혹은 프린터로 알려주는 COS reporter 등의 task들로 구성되어 있다.

타임 관리(time management) subsystem은 time delay 관리, 주기적 혹은 타임 스케줄링에 의해 수행되는 task들의 지원을 담당하는 소프트웨어로서 리얼타임 clock으로 부터 인터럽트를 받아 이를 시스템 내부적인 논리적 시간으로 처리하는 timer task와 각종 타임 delay 레이블을 관리하여 타임아웃시 이를 해당 task에 알려주는 delay timer로 구성되어 있다. 현장 통신 관리 (Field Communication Management) subsystem은 통신선으로 연결되어 있는 HOST, DC, RTU 사이에 데이터 혹은 제어명령의 송수신이 가능하도록 통신 프로토콜 (communication protocol)을 구현한 프로그램으로 각각의 프로세서 node들에 분산되어 있다. DBMS는 데이터베이스의 구성과 관리를 담당하는 subsystem으로서 본 장의 뒷부분에서 자세히 기술하였다. Control & Monitor는 현장으로 부터 감지되는 각종 상태 변화를 감시 분석하고 사용자 혹은 시스템 내부의 task들에 의해 요청 받은 각종 현장 제어 명령을 우선순위 (priority)에 의해 처리하여 주는 subsystem이다. 응용 관리 (application management) subsystem은 기본적인 감시제어기능을 바탕으로 에너지 최적제어, 연동제어, 경향감시 등과 같은 상의 레벨 기능을 수행하는 task들로 구성되어 있다.

Task간의 통신은 메시지 교환 (message exchange) 방식으로 구현되었으며 정보의 교환이 아닌 단순한 task간의 동기화만 필요한 경우에는 semaphore를 이용하였다. 메시지 교환은 해당 task들이 같은 프로세서 내에서 수행되는 경우에는 mailbox로의 메시지 송신/수신에 의해 이루어졌으며 task들이 서로 다른 프로세서들에 분산되어 있으면 통신 port로의 송수신에 의해 이루어졌다. Mailbox, 통신 port, semaphore의 처리는 리얼타임 OS의 서비스를 통해 구현되었다.

그림4는 현장으로 부터 상태변화가 COS Processor task에 의해 감지되는 것을 중심으로 해서 task들이 메시지 송수신을 통해 triggering 되는 관계를 task structure chart로 나타낸 것이다.

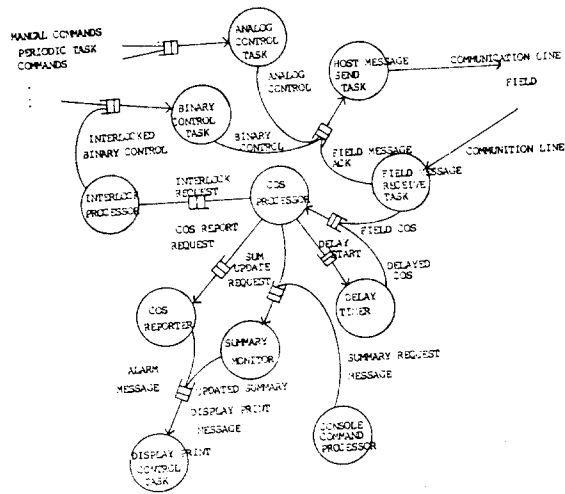


그림4. TASK STRUCTURE CHART

본 연구소에서 개발된 제어 시스템은 데이터 베이스를 중심으로 수행되는 시스템이다. 즉 각종 이벤트의 처리나 의사 결정들은 프로그램 내에 코드화 (code-wired) 되어 있는 것이 아니라 시스템 데이터베이스 내에 내장되어 있는 정보에 따라 수행된다. 뿐만 아니라 데이터베이스 자체도 시스템 수행에 따르는 내부의 상태변화, 현장상태변화, 혹은 사용자의 명령 입력에 의해 온라인으로 수정된다. 시스템 데이터베이스는 크게 다음의 3가지로 분류된다.

- 1) 현장 데이터베이스 (Field Database)
- 2) 오퍼레이터 데이터베이스 (Operator Database)
- 3) 응용 프로세스 데이터베이스 (Process Database)

현장 데이터베이스 (FDB)는 현장설비의 구성과 포인트의 감시제어 상태 정보를 저장하고 있으며 physical DB (PDB) 와 Logical DB (LDIS)의 2개의 계층 (layers)으로 구성되어 있다. PDB는 현장설비를 RTU 단위로 모델링하여 실제 포인트의 원시 데이터 및 통신상태 정보를 가지고 있다. LDB는 PDB의 원시데이터 (raw data)를 보다 논리적인 데이터로 분석 처리된 logical point들로 구성되어 있으며 각종 COS의 처리 및 응용 프로세스에 필요한 기본 정보 단위를 제공한다. FDB는 시스템 전반에 걸쳐 여러 task들에 의해 수시로 사용되는 DB로서 tree와 링크리스트 (linked list)의 복합

구조로 되어 있으며 여러가지 링크 포인터 테이블, 크로스 테이블 (cross table), 해시 테이블 (hash tables) 등이 다이내믹 생성, 수정되어 매우 효율적인 검색 (access), 수정 (update)이 가능하도록 되어 있다.

오퍼레이터 데이터 베이스는 입출력 장치구성 (I/O Device Configuration), 스테이션 구성 (Station Configuration), 패스워드, 서머리/그래픽 프레임 (Summary/graphic frames) 등의 정보를 내장하고 있어서 사용자의 입출력 기능을 운영환경에 맞게 온라인으로 구성 가능하게 되어 있다.

응용 프로세스 데이터베이스는 여러가지 현장 설비의 자동 제어, 최적제어, 공정제어 및 각종 데이터의 분석, 기록, 관리를 하는데 필요한 정보로 구성되어 있어서 연동제어, 경향감시 (trend log), 가동시간/사용량 계산, 타임 스케줄링 및 에너지 최적 제어들의 기능을 수행하는데 사용된다.

응용 프로세스 DB는 여러가지 시퀀스/프로세스들을 모델링 한 것으로 볼 수 있으며 이에 수반되는 parameter/포인트들의 할당 정보, 제어상태 정보들을 갖고 있다.

제어 시스템은 비동기적인 다수의 task들이 병렬 수행하는 multi-tasking으로 운영되므로 데이터베이스의 동시 수행 제어 (concurrency control)가 필요하다. 이를 위하여 현장 데이터베이스 관리 프로그램들은 shared 라이브러리로 구성되어 있으며 어느 task든지 DB를 access 하고자 하는 경우 shared 라이브러리의 해당 모듈을 호출 (call)하도록 되어 있다. Shared 라이브러리 모듈의 호출은 반드시 동기화 모니터 (synchronization monitor)에 의해 허가를 받도록 되어 있어서 2개 이상의 task가 DB를 동시에 access 하는 것이 방지된다. 한편 응용 프로세스 DB와 오퍼레이터 DB는 그것을 주로 사용하는 해당 task들에 의해 관리되며 필요시 혹은 요청시 다른 task들에게 메시지 형식으로 전달된다 (Message passing).

제어 시스템의 데이터베이스 관리는 실시간 제약이 따르므로 운영 중에는 보통 메모리에 내장되어 있으며 필요시 디스크로의 온라인 백업이 수행된다.

특히 현장 데이터베이스는 HOST, DC, RTU에 각각 분산 구성되어 있어서 RTU로 부터 HOST로 전송되는 데이터의 양 (communication rate)을 줄이고 현장 제어 감시에 관련된 일의 분담 (load distribution)을 통해 전체 시스템의 성능

(performance) 및 신뢰성 (reliability)의 향상을 이룩 하였다.

## V. 결론

제어시스템의 성능을 향상시키는 방법은 여러가지가 있다. 시스템의 병렬, 분산구성, 데이터 구조의 제구성, task 간의 상호 수행 관계 (interaction) 최소화, 효율적인 task 스리칭 및 인터럽트 처리 등은 모두 시스템의 성능향상에 영향을 미치는 요인들이다. 또한 제어시스템의 소프트웨어는 매우 방대하고 복잡하며 산업설비의 제어는 인명과 경제적인 손실을 수반할 수 있으므로 신뢰성이 중요한 성능 중의 하나가 된다.

본 연구소에서 개발한 제어시스템은 하드웨어 및 소프트웨어의 분산 구성, 리얼타임 오퍼레이팅 시스템의 사용, 데이터 베이스를 중심으로 한 소프트웨어의 설계, 다양한 사용자 입출력 서비스들 통한 효율적이면서 편리한 사용자와의 인터페이스 방식등을 도입, 적용 개발하여 시스템의 성능 (performance), 신뢰성 (reliability), 사용의 편리성을 높였다. 시스템의 소프트웨어는 대부분이 고급 시스템 언어 (high level system language)인 C로 구현되었으며 하드웨어 인터페이스 및 매우 빠른 수행 속도를 요구하는 부분에 대해서는 PL/M 과 어셈블리어로 구현하였다.

## REFERENCE

1. Mack W. Alford, "A Requirements Engineering Methodology for Real-Time Processing Requirements", IEEE Trans. Software Eng., Vol. SE-3, No. 1, Jan. 1977.
2. Neil B. Corrigan and J. Denbigh Starkey, "A Concurrent General Purpose Operator Interface", IEEE Trans. Software Eng., Vol. SE-10, No. 6, Nov. 1984, pp. 738-748.
3. James L. Peterson, "Petri Nets", Computing Surveys, Vol. 9, No. 3, Sep. 1977.
4. D. M. Steelman, "Distributed Cement Plant Control - The Intelligent Approach", IEEE Trans. Industry Applications, Vol. IA-18, No. 2, March/April 1982, pp. 192-198.
5. Claudio Walter, "Control Software Specification and Design An Overview", IEEE

---

Computer Society, Feb. 1984, pp. 20-23.

6. Bernard I. Witt, "Communicating Modules: A Software Design Model for Concurrent Distributed Systems", IEEE Computer Society, Jan. 1985, pp. 67-77.