

공장 자동화 전용 데이터베이스 처리기의 설계 및 구현

박 성진, 김 성훈, 백 두권
고려대학교 전산학과

Design and Implementation of A Database Processor for Factory Automation

Seong-Jin Park, Seong-Hoon Kim, Doo-Kwon Baik
Dept. of Computer Science, Korea University

요 약

본 논문에서는 공장의 모든 생산 환경을 컴퓨터를 이용, 자동화하는 컴퓨터 통합 생산(CIM)의 작업장 통제 시스템을 비롯한 중앙 통제 제어 시스템, 자동 창고 시스템, 자동 운반기 시스템 등을 위한 데이터베이스 처리기의 프로토타입을 설계 및 구현하였다.

이를 위해 FA전용 데이터베이스 처리기(database processor)의 요구 기능을 명세하고, 처리기의 기본 구조 및 실시간 지원 기법들을 제시했으며 단순 구조 질의어로서 SSQL(Simple Structured Query Language)을 개발하였다.

I. 서 론

컴퓨터 산업과 각종 전자 산업의 급속한 발달로 사회 여러 분야에 관한 자동화 요구가 증가하고 있으며 특히 제조업체들이 직면하고 있는 경제적 문제 해결과 생산성 향상을 통한 경쟁력 강화를 위해 공장 자동화(FA : Factory Automation), 특히 다품종 소량 생산 공장에서의 자동화 요구는 많은 관심과 연구의 대상이 되고 있다.

공장 자동화란 제품의 수주에서 출하까지의 생산 활동과 생산 시스템 전체의 효율적인 관리 및 제어를 위한 행위로 궁극적인 목적은 생산성 향상과 제품의 질을 높이는 것이다. 이러한 공장 자동화는 최종적으로 컴퓨터 통합 생산을 목표로 하고 있는데 CIM(Computer Integrated Manufacturing)이란 제품의 설계, 생산, 관리, 판매 등 제품 생산에 관련된 전체 업무를 컴퓨터를 이용하여 자동화한 시스템을 말한다. 즉, 종래에 별도로 개발된 CAD/CAM, 공장 자동화, 생산 관리, 재고 관리 등의 기능을 통합한 것으로, 각 서브 시스템 사이에 서로 정보를 교환하여 이들 사이의 제어를 자동화함으로써 전체 제품 생산을 더욱 체계적이고 정확하게, 그리고 효율적으로 할 수 있도록 지원하는 시스템이다. 또한 FA의 요소인 FMS는 CIM의 하위의 개념으로 생산 제품에 따라 신속성있게 라인을 운영하여 생산성을 높일 수 있는 시스템으로 다품종 중·소량 제품의 유연하고 신속한 생산을 지원한다.

이러한 자동화 시스템을 구성하는 데는 실제 제조 작업을 자동적으로 수행할 수 있도록 해주는 많은 컴퓨터와 CNC(Computer Numerical Control) 제어기, PLC(Programmable Logic Controller), 로봇 등 필요하며, 이러한 자동화된 가공 기계들 사이에는 각 제조 단계마다 필요한 데이터가 있고 각 단계마다 교환하여야 하는 제어 정보가 많아지게 된다. 따라서 이러한 데이터와 제어 정보에 대한 효율적인 자동 관리 시스템 즉, 공장 자동화 전용 데이터베이스 처리기가 요구된다.

공장 자동화 전용 데이터베이스 처리기란 공장내에서 실제 제조 작업과 각 공정을 수행하는 중에 CNC, PLC, 로봇 등으로부터 발생하는 데이터베이스에 대한 요청을 실시간적으로 처리하는 시스템으로 주로 자동화된 가공 기계들 사이의 필요한 데이터(NC 프로그램 포함) 및 제어 정보를 관리한다.

일반적으로 FA환경에서의 데이터베이스 관리 처리기는 독립된 프로세스로서 통제용 프로세스나 통신용 프로세스로부터의 질의 및 데이터 조작에 대한 요청을 수신, 처리하며 작업 및 일정에 대한 스케줄링을 지원한다. 공장 자동화 데이터베이스 시스템 개발을 위한 방안으로는 다음과 같은 네가지 유형을 고려할 수 있다.

- ① 기존의 상업용 DBMS를 수정없이 그대로 사용
- ② 새로운 기능들을 첨가하기 위하여 기존의 상업용 DBMS자체를 수정하여 확장
- ③ 단점을 보완하기 위하여 기존의 상업적 DBMS 위에(DBMS 자체 변경없이) 새로운 소프트웨어 layer를 추가하여 확장
- ④ 요구되는 기능들을 구비한 새로운 형태의 DBMS 개발

일반적으로 공장 자동화 데이터베이스 시스템은 상업적 DBMS의 모든 기능을 필요로 하는 것은 아니며, 오히려 그러한 기능 외에 필요한 데이터에 대한 복잡한 계산이나 분석을 요구한다. 그리고, 외부의 실재 상태를 즉각적으로 반영할 수 있는 일관성 제한 조건, 처리시간 제한 조건 등을 만족해야 한다.

비록 여러 상업용 DBMS가 각광을 받고 있지만 기존의 DBMS를 사용할 경우 여러가지 제한 사항이 존재하여 성능면에서 문제가 된다. 예를 들면 사용자가 원시 코드를 가지고 있지 못함으로 인해 버퍼와 배열 크기 등에 대한 제한과 한계 범위를 설정해야만 하고, 방대한 기능으로 인한 많은 메모리를 차지하여 효율이 떨어지며 특히, 실시간 기능을 제공하기 위해서는 가급적 병행 제어 및 회복 기법 등을 단순화시킬 필요가 있다. 따라서, 공장 자동화 전용 데이터베이스를 위해 요구되는 기능만 제공하며 고유하고 독자적인 구조를 갖는 새로운 데이터베이스 관리 시스템인 FA전용 데이터베이스 처리기의 설계, 개발이 요구된다.

외국의 경우에는 Oracle과 같은 기존의 상업용 DBMS(Database Management System)를 사용하여 공장 자동화 데이터베이스를 구축한 사례가 있으나 국내에서는 FA 혹은 FMS(Flexible Manufacturing System) 환경이 미비한데다 더구나 기존 DBMS를 사용하는 사례는 높은 비용으로 인해 중소기업의 실정에는 적합하지 않으므로 거의 없는 실정이다. 본 논문에서 제안한 FA전용 데이터베이스 처리기는 데이터 접근 요청에 대한 실시간 접근을 제공하며 자동적인 데이터베이스 관리 및 강력한 보고서 기능을 갖는 시스템이다. 즉, 일반적인 데이터베이스 시스템보다 실시간성과 자동 질의에 중점을 둔 시스템이다.

II. FA전용 데이터베이스 처리기의 설계

1. 설계시 고려 사항

FA전용 데이터베이스 처리기를 설계할 때에는 일반적인 DBMS가 제공해야 할 기능에 첨가하여 데이터베이스에 대한 요구의 빠른 수행과 자동적인 질의의 효율성에 중점을 두고 설계하여야 한다.

기존의 DBMS 설계시 데이터의 무결성(integrity), 시스템의 비용 절감, 사용자 인터페이스 등이 주요한 고려 사항이만 공장 자동화에 사용되는 데이터에 대한 처리는 실시간 데이터베이스를 필요로하므로 메모리 사용과 연산 횟수의 최소화가 주된 고려 사항이 된다.

실시간 데이터베이스란 데이터베이스 시스템에 실시간 시스템의 시간 제한을 첨가한 것으로 데이터 일치성(consistency)뿐만 아니라 시간 제약성을 보장해야만 한다. 즉, 각 트랜잭션은 올바르게 수행되어야할 뿐만 아니라 제한 시간안에 수행이 완료되어야 한다. 구체적인 기능은 다음과 같다.

(1) 데이터에 대한 실시간성 지원 기능

공장 자동화를 위한 데이터베이스가 가장 우선적으로 갖추어야 하는 기능으로는 “현재 상태의 정확한 표현” 즉, 실시간성이라고 할 수 있다. 현재의 전체 시스템이 처해 있는 상황에 대한 모든 정보를 실시간적으로 유지하여야 한다는 것이다. 실시간성이라는 기능의 의미는 두가지로 생각해 볼 수 있는데 첫째는 “알고 싶을 때 알고 싶은 정보를 즉시 알 수가 있다”는 의미이고 둘째는 “요구된 시간 내에 어떠한 사건을 확실하게 처리할 수 있다”는 의미이다. 첫째 항목의 기능은 생산 관리 체계의 합리화를 위

해서 각 생산 셀(Cell)에 대한 기초 데이터의 수집과 축적을 의미한다. 두번째 기능은 FMS와 같은 다기능 가공 시스템에서 실시간성이 가능하면 생산 시뮬레이션이 가능하게 되고, 이에 따라 제품의 수주가 발생하였을 때 정확한 납기를 계산해 낼 수가 있어서 요구된 시간 내에 주어진 일을 처리할 수 있게 된다는 것이다.

(2) 자동적인 데이터베이스 관리 기능

일반적인 사용자와의 대화식 모드를 통한 질의 처리 기능 이외에도 데이터베이스에 대한 제어기들의 요청이 오면 자동적으로 그 요구를 해석, 원하는 데이터를 넘겨주는 자동 모드 기능이 필요하다. 공장 자동화의 특성상 사용자보다는 제어기들로부터 공정이 진행되는 과정에서 데이터베이스에 대한 요청이 빈번하게 발생하며 이러한 요구는 데이터베이스 내의 데이터에 대한 질의나 데이터의 변경 등을 포함할 수 있으므로 이에 대한 적절한 수행 기능이 지원되어야 한다.

(3) 데이터의 특성 지원 기능

데이터베이스의 물리적 구조에 있어서 필드당 문자수, 레코드당 문자수, 파일당 레코드수 등은 각 제어기에서 필요로 하는 데이터 특성 특히 데이터의 크기 등을 만족시킬 수 있어야 한다. 예를 들면 데이터베이스의 저장되는 데이터중에는 NC 프로그램과 같은 대용량의 데이터도 포함될 수 있어야 한다.

(4) 강력한 보고서 기능

특정 시점에서의 공정 상태, 작업 진행 상황 등을 명확하게 보여줄 수 있는 강력한 보고서 기능이 필요하다. 예를 들면 각 셀의 부품 수량이라든가, 공정의 진척도 등에 대한 그래픽 디스플레이 등과 같은 기능이 필요하다.

(5) 데이터 회복 및 동시성 제어 지원 기능

기존의 보조기억 장치 데이터베이스를 기반으로 하는 방법이 아닌 실시간 데이터베이스를 지원하는 효율적인 회복 및 동시성 제어 기법의 개발이 필요하다.

(6) 정형화된 질의 양식 제공 기능

물품의 소재 및 수량 체크, 공정 현황 파악 등과 같은 빈번히 사용되는 질의 형태에 대해서는 보다 손쉬운 정형화된 질의 양식이 제공되어야 한다.

2. FA전용 데이터베이스 처리기의 실시간 지원 방안

일반적으로 실시간성을 제공하기 위한 데이터베이스 시스템 개발 방안으로는 기존의 DBMS보다 빠른 반응 시간을 제공하는 새로운 데이터베이스 시스템 구조를 개발하는 방안, 실시간성을 위해 장애가 되는 장치(예, 디스크)를 고성능의 장치로 대체하는 방안, 기존 DBMS의 특성 중의 일부 기능을 제한하는 방안 등의 세가지로 나누어 생각해 볼 수 있다.

공장 자동화 전용 데이터베이스 처리기에서는 두번째 및 세번째 방안을 통합해 실시간 데이터베이스를 개발한다. 즉, 실시간성을 제공하기 위하여 주기억 장치 데이터베이스를 구성하고 동시성 제어 및 회복 기법의 기능을 단순화하였다.

(1) 주기억 장치 데이터베이스

FA전용 데이터베이스 처리기에서는 공장 자동화 환경에서의 데이터 실시간 접근성 요구를 만족시키기 위해 주기억 장치 데이터베이스 시스템(main-memory database system)을 구현했다. 주기억 장치 데이터베이스 시스템은 데이터베이스 자체가 주기억 장치내에 상주함으로써 인접한 기존의 보조 기억 장치(주로 디스크)의 데이터베이스에 대한 액세스로 인한 시간상의 오버헤드를 감소시킬 수 있는 시스템이다. 주기억 장치 데이터베이스는 데이터베이스 전부 혹은 대부분을 저장하는 형태로 구분될 수 있는데 본 연구는 주기억 장치에 데이터베이스 전부를 상주시키는 경우만을 고려한다. 이는 데이터베이스의 대부분을 주기억 장치에 저장하는 방안은 기존의 버퍼 크기를 단순히 확장하여 대용량의 버퍼풀을 이용함으로써 디스크 액세스 횟수를 감소시킬 수 있지만 주기

억 장치에 없는 데이터는 결국 디스크로의 액세스를 필요로하므로 기존의 효율적인 액세스 기법을 사용하더라도 데이터베이스 전체가 주기억 장치에 있는 경우보다 응답 시간이 떨어지기 때문이다. 주기억 장치 데이터베이스 시스템은 공장 자동화의 실시간성 요구의 중요성과 주기억 장치 가격의 하락을 고려할때 바람직하다. 그리고, 디스크와 같은 보조 기억 장치는 데이터베이스의 안정된 백업 버전들을 저장하기 위해 사용한다.

(2) 인덱스 기법

주기억 장치를 위한 인덱스 구조와 디스크 지향 인덱스 구조는 상이하다. 디스크 지향 인덱스 구조의 주요 목표는 디스크 액세스 회수와 디스크 공간의 최소화이다. 하지만 주기억 장치를 위한 인덱스 구조는 데이터베이스가 주기억 장치에 위치하므로 디스크 액세스 시간을 최소화하는 대신에 가능하면 적은 메모리를 차지하면서 전체 수행 시간을 감소시키는 것이 목표이다.

디스크의 데이터에 대한 일반적인 액세스 방식은 임의 그리고 순차 키 액세스를 제공하는 B+ tree 기법이지만 전체 데이터베이스가 주기억 장치에 상주하는 FA전용 데이터베이스 처리기에서는 AVL(혹은 다른 이진) 트리 기법을 사용한다.

(3) 동시성 제어

실시간 접근성을 제공하기 위해 수정하는 곳을 제한하여 타 모듈은 판독만 가능하도록 함으로써 동시성 제어 기능을 단순화하며, 방안으로는 로킹 방법보다는 낙관적 동시성 제어 방법을 사용한다.

(4) 회복 기법

주기억 장치 상주 데이터베이스에서 가장 중요하게 간주되는 회복 기법은 논리적 오류나 시스템 오류, 고장 등으로 인해 소멸성 기억 장치(volatile storage)인 주기억 장치의 데이터베이스의 정보가 유실되지 않도록 고장을 탐지하고 그 고장이 발생하기 이전의 상태로 복원시키는 것을 말한다. 고장이 발생했을때 주기억 장치 데이터베이스의 내용이 디스크에 있는 마지막 백업용 데이터베이스와 일치하지 않을 수가 있다. 이에 대한 아주 간단한 회복 구조로는 디스크에 있는 백업용 데이터베이스를 먼저 재로딩하고 그것에 대해 트랜잭션 로그(log)를 현시점까지 적용하는 방법이 있다. 이때, 데이터베이스가 주기억 장치에 상주하기때문에 회복을 위한 디스크로의 로그 입/출력 시간이 주요 문제점이 된다. 본 FA전용 데이터베이스 처리기에서 고려하는 회복 기법으로는 다음과 같은 방안들이 있다.

① 선완료 트랜잭션

다수의 트랜잭션에 걸친 로그 입/출력의 속도를 높이기 위해 선완료 트랜잭션(pre-committed transaction) 개념을 적용한다. 임의의 트랜잭션이 완료할 준비가 되어 있을때 완료 레코드(commit record)를 로그 버퍼(log buffer)에 기록하고 완료 레코드를 실제로 디스크에 기록하기전에 그 트랜잭션의 모든 잠금(lock)을 풀어준다. 이렇게 완료전에 모든 잠금을 해제하여 선완료 트랜잭션의 dirty data에 대한 판독을 가능하게 함으로써 블러킹 시간을 단축시킨다.

② 체크 포인트

시스템 고장으로 인한 회복 시간을 줄이기 위한 방안중의 하나로 정기적으로 데이터베이스를 안전 장치(stable storage)에 체크포인트(check-point)한다. 알고리즘은 마지막 체크포인트후에 갱신된 데이터들을 디스크의 데이터베이스에 백업하는 것이다. 체크포인트를 되도록 자주하는것이 회복시간을 단축시켜주지만 너무 잦은 체크포인트는 데이터처리의 지연을 발생시키므로 trade-off를 고려한다.

③ 로그 압축

메모리 관리와 회복 시간의 절약을 위해 저장되는 로그 정보의 양을 효율적으로 줄인다. 로그 정보로는 데이터 조작에 대한 연산 자체를 기록함으로써 회복시 정보 해석의 복잡성과 그로 인한 실행시간이 오래 걸리는 논리적 정보가 아닌 변경된 모든 데

이타의 값을 기록하는 물리적 정보를 저장한다.

④ 롤백 데이터

오류로 인하여 트랜잭션이 완료되지 않았을 경우 트랜잭션을 롤백(rollback)시키기 위하여 그 트랜잭션으로 인해 갱신된 데이터의 이전 값을 주기억 장치내의 롤백 영역에 일시적으로 저장한다.

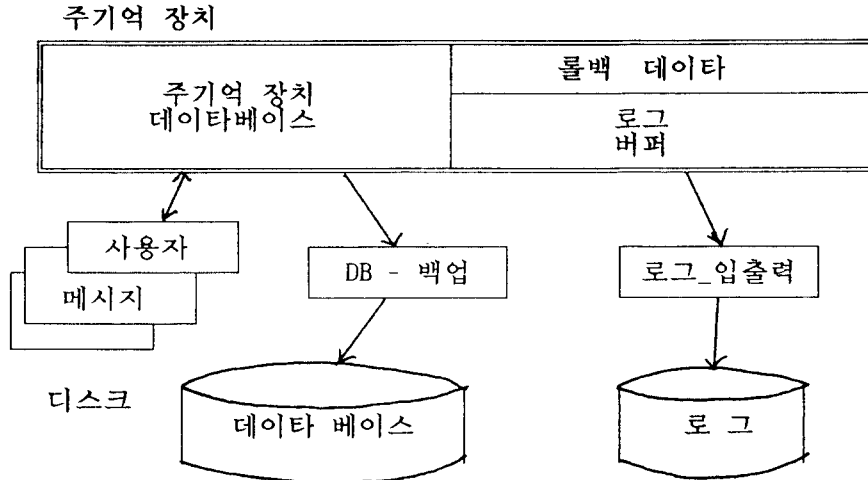


그림1. 기본 구조

그림 1은 주기억 장치 데이터베이스와 관련하여 FA전용 데이터베이스 처리기에서 적용하는 고장 회복 메카니즘 기본 구조이다. 주기억 장치는 크게 전체 데이터를 갖는 데이터베이스와 로그 정보를 저장하는 로그 버퍼, 수행중인 트랜잭션 대상 데이터의 원래 값을 저장하는 롤백 데이터로 구분할 수 있다. 주기적인 체크포인트에 의한 데이터베이스의 백업 저장은 DB-백업이 수행하고, 일정 단위의 디스크 로그 정보 판독 및 출력은 로그_입출력이 수행한다.

3. FA전용 데이터베이스 처리기의 구조

FA전용 데이터베이스 처리기의 전체적인 구조는 크게 상부 구조와 하부 구조로 구분되며, 이들은 각각 독립적인 기능을 담당하는 여러 개의 모듈로 구성된다.

(1) FA전용 데이터베이스 처리기의 상부 구조

FA전용 데이터베이스 처리기의 상부 구조는 데이터베이스에 대한 요청을 받아들이는 부분으로 사용자 및 제어기와의 인터페이스 모듈과 받아들이는 질의어의 분석 모듈로 구분한다.

① 인터페이스 모듈

FA전용 데이터베이스 처리기는 데이터베이스에 대한 접근 형태로 사용자와의 대화식 모드(interactive mode)와 각 제어기와의 자동 모드(automatic mode)의 두가지 질의 모드를 지원한다.

㉠ 대화식 모드

대화식 모드는 일반적인 데이터베이스 시스템에서 제공되는 질의 형태로서 본 시스템에서는 관계형 데이터베이스의 가장 범용적인 질의어인 SQL(Structured Query Language)을 모델로하여 SQL형태의 질의어 SSQL을 사용한다. SSQL은 기본적인 구조가 SQL과 거의 동일하며 데이터베이스 정의, 삽입, 삭제, 갱신 등의 기본적인 기능을 제공하며 기본적인 명령어의 축약 형태도 제공한다.

```

<< SSQL list >>
## create table tablename { fieldname (type[n]), ... , ... } ;
## cret //
## delete table tablename ;
## delt //
## insert data tablename { data1, data2, ..., ... } ;
## insd //
## delete data tablename { record number } ;
## deld //
## delete from tablename
where condition ;
## update data tablename [ recordnumber ] [ filename ], newdata ;
## ud //
## update tablename
set fieldname = newvalue, ... , ...
where condition ;
## display table tablename ( dt ) ;
## display table list ( dtl ) ;
## display table list and type ( dtlt ) ;
## load ;
## save ;

```

㉠ 자동 모드

각 공정 사이에 동적으로 발생하는 다양한 데이터베이스 접근 요청에 대한 자동적인 처리 형태를 제공한다. 자동 모드는 자동화의 특성상 반드시 필요한 데이터베이스 질의 형태로서 제어기들로부터의 메시지를 수신하여 SSQL로 번역하는 번역기가 지원되어야 한다. 번역기는 독립적인 제어기들로부터 데이터베이스에 대한 질의 요청이 접수되면 적절한 SSQL로의 변환을 통해 요구 조건에 부합하는 데이터를 처리, 결과를 해당 프로세스에 보낼 수 있게 한다.

대화식 모드에서는 질의의 결과를 테이블 형태로 화면에 디스플레이하는데 비해서 자동 모드에서는 질의 및 데이터 조작에 대한 메시지의 수행 여부와 그 결과값에 대한 데이터만을 테이블 형태가 아닌 배열 형태로 전달 한다. 이 자동 모드의 사용은 제어기들로부터 빈번하게 요구되는 질의 형태의 작업을 효율적으로 처리할 수 있게 한다.

㉡ 질의어 분석 모듈

인터페이스 모듈로부터 전달된 SSQL를 분석하여, FA전용 데이터베이스 처리기가 이 해할 수 있는 형태로 변환한 후에 질의어 처리 모듈로 넘겨주는 역할을 한다. SSQL을 토큰 리스트(token list)로 분리하여 데이터 조작문인 경우엔 토큰 리스트를 질의어 처리 모듈에 직접 전달하며 질의 구문인 경우엔 토큰 리스트를 다시 트리플 리스트(triple list)로 변환하여 넘겨준다. 이 트리플 리스트를 이용하여 SELECT 와 WHERE절을 해석하여 뷰(view)를 생성한다.

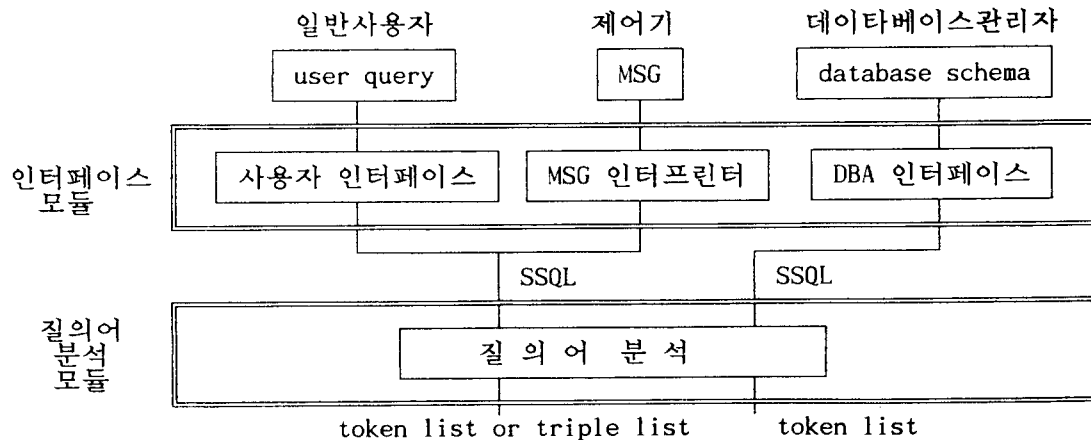


그림2. FA전용 데이터베이스 처리기의 상부 구조

(2) FA전용 데이터베이스 처리기의 하부 구조

FA전용 데이터베이스 처리기의 하부 구조는 받아들인 질의의 요구 사항에 대한 실질적인 작업을 수행하는 부분으로 질의어의 내용에 따라서 DML, DDL 처리, 질의어 처리 등의 질의어 처리 모듈과 고장 회복을 위해 각 처리에 대한 회복 정보를 저장하는 고장 회복 모듈로 구분된다.

① 질의어 처리 모듈

질의 처리 모듈은 질의어 분석 모듈에서 분석된 질의어를 처리하는 역할을 한다. 데이터 정의어 처리 루틴에서는 데이터베이스내의 릴레이션에 대한 스키마를 정의하는 사용자 질의어를 처리하는 역할을 하며 데이터 조작어 루틴은 데이터베이스내의 데이터를 검색하고, 수정, 그리고 저장하는 질의어를 처리하는 역할을 한다. 질의를 수행하기 위해 트리플 리스트를 이용, SELECT와 WHERE절을 해석하여 뷰를 생성한다.

② 고장 회복 모듈

데이터베이스에 대한 조작 및 질의를 수행하는 중에 일어날 수 있는 고장의 회복 기능을 담당하는 모듈이다. 각 트랜잭션이 실행될때마다 트랜잭션 단위로 고장 회복 정보(log)를 안전한 기억 장소에 저장한다.

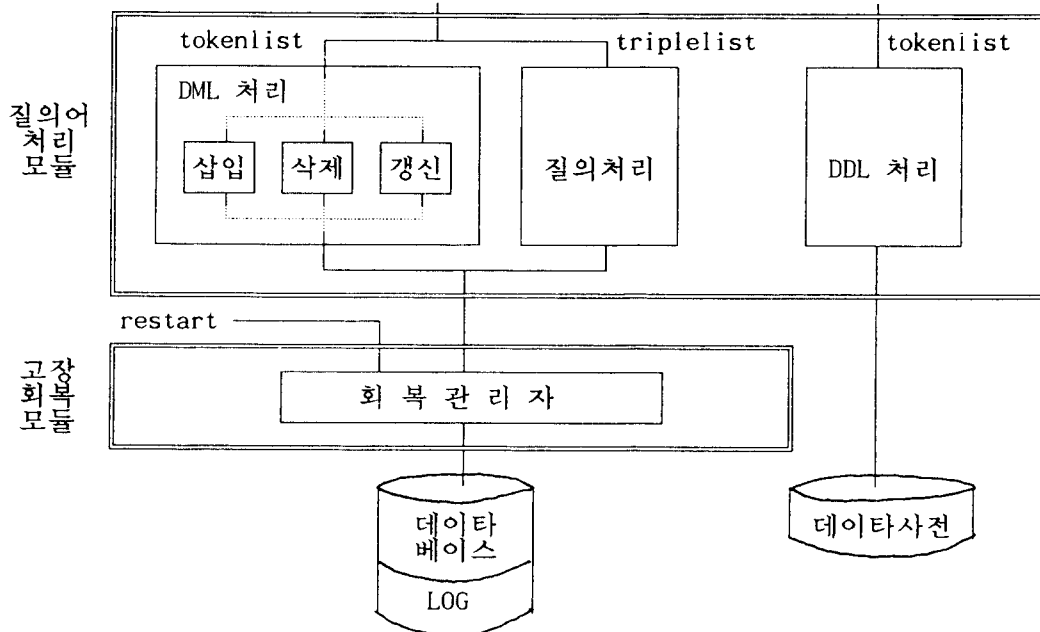


그림3. FA전용 데이터베이스 처리기의 하부 구조

Ⅲ. FA전용 데이터베이스 처리기의 구현

1. 구현 환경

(1) 개발 환경

FA전용 데이터베이스 처리기는 현재 컴퓨터에서 널리 사용되고 있고, 이식성이 뛰어난 UNIX 및 XENIX 환경에서 C 언어를 사용하여 구현하였다. 데이터베이스 모델은 범용적인 관계형 데이터베이스 모델을 사용하였으며, 일반 사용자와 제어기들이 새로운 질의어를 통해 FA전용 데이터베이스 처리기와 대화할 수 있도록 하였다.

(2) 시스템 구성

본 모형 공장에서는 TCP/IP Protocol을 사용한 Ethernet LAN을 이용하여 통제 컴퓨터들을 연결하며 토폴로지는 버스(bus)형으로 EXCELAN 카드를 사용한다. 컴퓨터 O/S는 UNIX(워크스테이션), XENIX(PC급)를 사용하며 Berkely Socket을 이용하여 통신 프로그램이 짜여졌다.

한편, 구축한 모형 공장은 FMS에서 일어나는 상황을 가능한 한 모두 포함시키기 위해 실험용 공장 배치를 다변화하였으며 모형 공장의 통제는 계층화된 전산망에 의하여 이루어지는데 각 기계 자체의 통제는 CNC 또는 PLC에 의하여 이루어지고 그 위의 계층에서는 PC가 이들을 통제하고 전산망 전체는 워크스테이션이 통제한다. IBM-PC 계열인 80386 PC가 통제하는 부분 시스템은 자동창고 시스템, AGV, 작업장 통제 시스템, 제품 검사 시스템이며 공장 전체를 통제하는 중앙 통제 제어 컴퓨터는 워크스테이션급인 MIPS 컴퓨터를 이용한다.

대상 모형 공장은 그림 4와 같다.

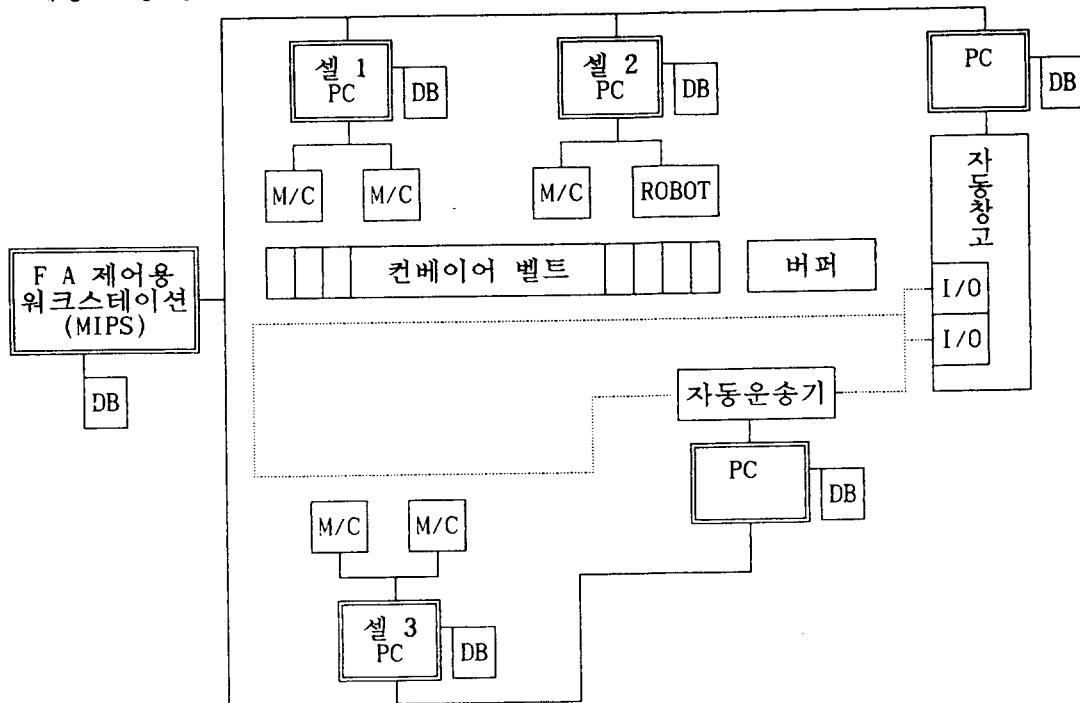


그림4. 시스템 구성도

① 공정 제어기(Cell Controller)

공장 작업 셀(Cell)의 모든 기계의 동작을 감시, 필요한 제어를 수행하며 동작의 이상시 경고 신호를 발생시키는 등 한 셀에서 일어나는 생산 공정을 자동적으로 제어/관리하는 시스템이다.

② 중앙 통제 제어기(Floor shop Controller)

공장 전체를 관할, 생산을 통제하는 컴퓨터로 주된 기능으로는 각 하부 시스템의 상태 점검, 주문 수령, 스케줄링, 자재 흐름 조정, 각 장치의 작업 준비 또는 실시 명령, 틀 정보 관리, 공장의 상황 보고 등이다.

③ 자동 창고(Automated Storage Retrieval System)

공장 자동화의 필수 요소로 단순한 물품 저장의 개념을 넘어 공장내의 물품의 흐름이 원활하도록 입고, 출고, 입하, 출하 등을 촉진한다.

④ 자동 운반기(Automated Guided Vehicles)

중앙 통제 제어기와 긴밀한 연락을 취하면서 공장 전체의 자재 흐름이 원활하게 이루어질 수 있도록 각 셀과 AS/RS 사이의 자재를 운반한다.

2. FA전용 데이터베이스 처리기의 상부 구조 구현

각 모듈에 대한 개략적인 기능만 기술하면 다음과 같다.

(1) 인터페이스 모듈

사용자 및 자동화된 공정 제어기로부터 질의어 문자열을 입력 받아 질의어 분석 모듈로 전달한다. 사용자로부터는 직접 SSQL 문장을 입력받지만 공정 제어기로부터는 특정한 메시지를 수신받아 인터프리터에 의한 적절한 SSQL로의 변환 단계를 거친다.

(2) 질의어 분석 모듈

질의어 분석 모듈은 어휘 분석(lexical analysis) 단계와 파싱(parsing) 단계로 구분된다. 어휘 분석 단계에서는 문자열 형태로 입력된 질의어를 어휘 규칙(lexical rule)에 따라 토큰으로 분리하는 역할을 한다. 파싱 단계에서는 분류된 토큰을 문법 규칙(grammar rule)에 따라 토큰의 의미적 오류를 검사한다. 오류가 검출될 경우 사용자에게 오류 메시지를 나타내고, 정상일 경우에는 질의어 처리 모듈내의 해당 루틴을 호출한다.

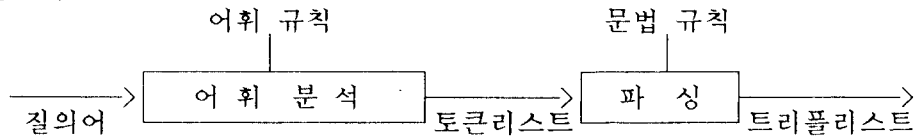


그림5. 질의어 분석 모듈의 구성도(질의어의 경우)

3. FA전용 데이터베이스 처리기의 하부 구조 구현

(1) 질의어 처리 모듈

데이터베이스에 대한 실질적인 요구사항이 수행되는 모듈로 삽입, 삭제, 갱신 등의 데이터 조작용 처리 모듈(DML 모듈)과 질의 처리 모듈, 데이터 정의어 처리 모듈(DDL 모듈)등의 서브 모듈들로 구분할 수 있다. FA전용 데이터베이스 처리기에서는 질의어의 효율적인 조건 탐색을 위해 토큰 리스트를 트리플 리스트로 변환하여 사용한다. 트리플 리스트의 예와 구조는 표 1.과 같다.

표 1. 토큰 리스트 및 트리플 리스트

select	0 , 1 , NULL
a. number	1 , 3 , NULL
b. name	2 , 2 , NULL
from	0 , 2 , NULL
a	1 , 0 , NULL
b	2 , 0 , NULL
where	0 , 3 , NULL
a. id	1 , 1 , NULL
=	0 , 6 , NULL
b. id	2 , 1 , NULL
and(or)	0 , 12(13), NULL
a. number	1 , 3 , NULL
>	0 , 7 , NULL
11	0 , 0 , 11

< 토큰 리스트 >

< 트리플 리스트 >

트리플의 구조 :

table number, field number, data

- 1. if (table number == 0 and data == NULL) then
 this triple means "keyword" or "operator"
 --- triple : (0 , x , NULL)
 token : select , where , = , > , ...

```

2. if ( table number == 0 and fieldnumber == 0 ) then
    this triple means "constant value"
    --- triple : ( 0 , 0 , X )
        token : "kim" , ...
3. if ( table number != 0 and fieldnumber != 0 and data == NULL )
    --- triple : ( X , Y , NULL )
        token : a.number , ...

```

(2) 고장 회복 모듈

로그 버퍼에는 갱신된 데이터값을, 롤백 데이터에는 기존의 값에 관한 정보를 기록한다. 트랜잭션 오류가 발생할 경우엔 롤백 데이터로부터 이전의 데이터값을 얻어 복귀시키고, 시스템 고장 등으로 인해 완료된 트랜잭션의 결과가 데이터베이스에 적용되지 못한 경우에는 로그 버퍼로부터 갱신된 데이터값을 얻어 재수행(redo)시킨다. 장치 고장에 대해서는 둘 이상의 장치에 데이터베이스를 저장시킴으로써 고장 발생시점의 상태로 회복시킬 수 있다.

IV. 결론

본 논문에서는 공장 자동화에 사용되는 데이터베이스의 특성과 그에 따른 요구 사항을 분석하였으며 실시간 접근성, 자동적인 데이터 관리, NC 프로그램과 같은 대용량 데이터 특성을 지원하는 FA전용 데이터베이스 처리기의 프로토타입을 설계, 구현하였다. 공장 자동화의 특성상 실시간성을 제공하기 위해 주기억 장치 상주 데이터베이스를 구축, 그에 대한 적절한 회복 기법과 동시성 제어 기법 등을 적용하였으며 대화식 질의 모드 이외에도 동적인 데이터베이스의 접근에 대한 자동적인 처리를 위해 메세지 번역을 포함하는 자동 질의 모드를 구현하였다.

추후 연구 과제로는 제안된 회복 및 동시성 제어 기법들의 보안과 현 시스템을 분산 환경에 적용시키기 위한 연구가 필요하다.

참고 문헌

- [1] Dr Paul Ranky, "The Design and Operation of FMS," North-Holland Publishing Company.
- [2] R.W. Yeomans, "Design Rules for A CIM System," North-Holland Publishing Company.
- [3] I. Burhan Turksen, "Computer Integrated Manufacturing," Springer-Verlag.
- [4] Jeffrey D. Ullman, "Database and Knowledge-base Systems," Computer Science Company.
- [5] Philip A. Bernstein, "Concurrency Control and Recovery in Database System," Addison-Wesley.
- [6] M. Singhal, "Issues and Approaches to Design of Real-Time Database Design," Addison-Wesley.
- [7] M. Singhal, "Issues and Approach to Design of Real-Time systems," ACM SIGMOD Record, Vol.17, No1, pp.19-33, 1988.
- [8] K. Lin and M. Lin, "Enhancing Availability in Distributed Real-Time Databases", ACM SIGMOD Record, Vol.17, No.1, pp.34-43, 1988.
- [9] L. Sha, R. Rajkumar, and J.P. Lehoczky, "Concurrency Control for Distributed Real-time Databases," ACM SIGMOD Record, Vol.17, No1, pp.82-98, 1988.
- [10] Dewitt D, Katz R, Shapiro L, Stonebraker M, and Wood D, "Implementation Techniques for Main Memory Database Systems," ACM SIGMOD Conference Proceedings, pp.1-8, 1984.
- [11] 이현길, 채종원, 김진호, "공정-제어 응용을 위한 실시간 데이터베이스 시스템의 개발," 한국정보과학회, '91 학술발표 논문집, 18권 1호, pp.7-10, 1991.
- [12] 고려대학교 부설 정보통신기술연구소, "국산자동화 기기를 사용한 FMS 개발," 중간보고서, pp.3-9, 1991.