공정계획을 위한 분산객체의 응용

- Application of Distributed Objects for CAPP -

김준국*
Kim Jun Kook
이홍희*
Lee Hong Hee

Abstract

As the market and the organizations of an enterprise expand globally, the rapid and accurate communication gets more important for product manufacturing. The manufacturing information flow among the designers, the process planners and the shop floors is characterized and modelled. Its methods are constructed using distributed objects. Their introduction to the network-based CAPP system offers speed and safety for the system and makes the reconstruction and distribution of the application programs easy. The manufacturing processes are generated based on the feature information of a designed part, then the manufacturing resources are selected using the process planning logic which is implemented by distributed objects. The databases and distributed objects are integrated under the recent internet environments. The developed system makes it possible to manipulate and to transfer the process planning and manufacturing data everywhere in the world.

1. 서 론

시장과 생산조직이 지리적으로 널리 분산되면서 원거리에 있는 생산조직 간의 신속하고 안정적인 생산관련정보의 교환이 제품개발의 기간단축과 비용절감에 매우 중요한 요소가 되었다. 최근 전산환경의 발전에 따라 전 세계의 모든 생산자들이 인터넷을 이용하여 필요한 정보를 검색하고 교환할 수 있게 되었으며, 제품정보를 분산되어있는 다수의 공장과 협력업체들로 효율적으로 신속히 전달하여 생산하는 분산생산시스템도 구현할 수 있게 되었다. 이러한 분산 생산시스템은 인터넷상에서 각 지역의 설계와 생산정보

* 인하대학교 산업공학과

를 통합함으로써 가능해지며 다음의 두 가지 요건을 만족해야한다. 첫째로, 방대한 제품 정보를 인터넷상에서 신속하게 처리해야 한다. 즉, 정보처리속도를 최소화시켜야 인터넷상에서 설계와 생산의 실시간 통합이 가능해진다. 설계에서 납품에 이르는 전 생산과정에 걸친 모든 제품정보가 통합되기 위해서는 설계정보, 공정계획정보, 생산정보 등의 데이터베이스가 분산시스템의 특성에 맞게 효율적이고 빠르게 처리될 수 있도록 설계되고 구현되어야 하며, 데이터처리량이 많고 복잡한 업무에도 대처할 수 있는 응용프로그램이 구축되어야 한다. 둘째로, 시스템이 환경변화에 유연하게 대처해야 한다. 분산생산시스템은 각지의 다양한 요구에 곧바로 적응하여야 하고 생산시스템의 변화에 따른 잦은 수정에도 신속하고 용이하게 대응해야 한다.

본 연구에서는 분산생산시스템에서 필요한 정보의 신속한 처리 및 시스템의 융통성 확보를 위해 분산객체기술을 공정계획에 도입한다. 비즈니스 객체들이 네트워크로 분산되어있을 때 이를 분산객체(Distributed Object)라고 하며, 이 분산객체들이 모여서 전체 시스템이 구성된다. 이들간에는 뚜렷한 클라이언트와 서버의 개념이 없으며 시스템의 기능은 여러 객체들간의 상호작용으로 구현된다. 이러한 분산객체시스템은 클라이언트와 서버의 능력을 분산시킴으로써 통신망 내의 병목을 해소시주고, 시스템 변경 시 변경이 필요한 객체만 골라서 작업할 수 있으므로 유지보수에도 용이한 장점을 가진다. 이러한 분산객체를 관리하는 미들웨어로 Microsoft사의 DCOM(Distributed Common Object Model)을 이용해서 비즈니스 로직을 구현한다. 먼저 설계정보와 제품정보, 그리고 생산능력에 관한 정보들을 저장하고 처리할 수 있는 데이터베이스를 구축한다. 그리고 분산객체를 이용하여 각종 정보를 인터넷상에서 조회하고, 제품을 제작하기 위한 공정 계획을 수립하는 프로세스를 설계해 인터넷 클라이언트인 웹브라우저로 실제 작업을 수행하는 시스템을 구현한다.

2. 연구배경 및 이론적 고찰

2.1 분산객체시스템

분산환경이란 네트워크환경을 기반으로 원격지에 있는 시스템간의 협동작업을 통해 서로의 자원을 공유하거나 필요한 정보를 주고받는 등의 상호작용을 할 수 있는 환경을 말한다. 분산환경의 설계와 구축에는 여러 가지 고려사항이 철저히 분석되어야 한다. 현재의 네트워크장비와 인터넷의 발전은 여러 가지 새로운 전산환경을 제공하여 실시간으로 대용량의 다양한 정보를 제공한다.

분산환경에 있어서 클라이언트/서버 구조는 2-tier 클라이언트/서버 구조에서 3-tier 클라이언트/서버 구조로 변화하고 있다. 2-tier 클라이언트/서버 모델은 서버는 데이터만 가지고 클라이언트가 프레젠테이션 로직과 비즈니스 로직을 가지는 형태이다. 2-tier 클라이언트/서버 모델은 개발하기가 쉽고 단순하다는 이점이 있지만, 이러한 2-tier 방식의 클라이언트/서버 모델은 LAN 범위를 넘어서 배치될 때는 클라이언트의 배치, 관리 및 확장이 어려워진다. 또한 많은 클라이언트가 한 서버로 접근할 때 많은 양의 정보가 네트워크를 통하여 전송되기 때문에 네트워크의 병목현상이 생긴다. 이러한 문제를 해결

한 것이 3-tier 모델이다. 3-tier 클라이언트/서버 시스템은 확장성, 견고성, 유연성이 뛰어나고, 또한 여러 경로의 데이터를 통합할 수 있다. 3-tier 응용프로그램은 데이터 베이스와 직접 상호작용하지 않고, 그 대신 클라이언트는 서버 상에서 비즈니스 로직을 호출하고, 이 비즈니스 로직이 클라이언트를 대신하여 데이터베이스를 검색한다. 따라서 비즈니스 로직이 변경되더라도 서버만 변경해주면 되기 때문에 응용프로그램을 네트워크 상에서 관리하기가 용이하다.

대부분의 3-tier 응용프로그램에서의 중간계층은 클라이언트에서 비롯되는 다양한 비즈니스 트랜잭션에서 사용되는 객체들의 집합으로서 구현된다. 비즈니스객체들이 네트워크로 분산되어 있을 수 있는데 이를 분산객체라고 한다. 이들 분산객체들은 자신에 접근할 수 있는 공개된 인터페이스를 가지고 있는데, 이들의 모임들이 전체 시스템을 구성하는 부분으로 사용된다고 하여 컴포넌트라고 한다. 이들간에는 뚜렷한 클라이언트/서버의 개념이 없으며 여러 컴포넌트간 다단계 상호작용으로 기술되며, 이러한 시스템들끼리도 상호작용을 하므로 N-tier시스템이 된다. 각 컴포넌트들은 상대적으로 작은 기능들을 수행한다. 하나의 컴포넌트는 비즈니스 로직을 보다 효율적으로 구현하기 위해 다른 컴포넌트들을 호출할 수 있다. 컴포넌트기반 응용프로그램으로서 중간계층을 설계할 때는 다음과 같은 이점을 얻을 수 있다. 첫째로, 컴포넌트를 재 사용할 수 있다. 소스코드의 재사용에 초점을 맞추고 있는 객체지향언어와는 달리 컴포넌트를 이진코드로 재사용한다. 둘째로, 클라이언트가 데이터와 기능을 쉽고 안전하게 액세스할 수 있다. 클라이언트는 컴포넌트들에게 자신들을 대신하여 기능을 수행하라고 요청한다. 서버에 위치한 컴포넌트는 응용프로그램들의 논리의 세부사항들을 캡슐화 한다. 클라이언트는 요청을 수행하기 위해 액세스되고 있는 데이터베이스를 알아야 될 필요가 없다. 또한 요청이 실행을 위해 다른 컴포넌트에게 전송되었는지 응용프로그램에게 전송되었는지 확인할 필요가 없다. 캡슐화는 내용, 보안, 데이터에 대한 미리 설계된 인터페이스를 통한 액세스를 제공하여 수많은 응용프로그램으로부터 들어오는 통제되지 않은 액세스를 제거한다. 셋째로, 업그레이드가 용이하다. 컴포넌트기반시스템은 새로운 클라이언트를 구성하고, 추가적인 중간계층 컴포넌트를 추가하며, 기존 컴포넌트를 재사용하여 빠르게 응용프로그램을 재구성할 수 있다. 또한 클라이언트를 변경하지 않고도 컴포넌트들을 업데이트 할 수 있다. N-tier 컴포넌트 클라이언트/서버 모델은 3-tier 클라이언트/서버 모델의 확장형으로 3-tier 모델의 장점과 단점을 모두 지니면서 컴포넌트의 캡슐화로 안정성을 확보하고 비즈니스 로직의 부하분산이라는 추가적인 장점을 가지고 있다.

분산객체는 생성된 이후에 응용프로그램에 관계없이 독립적으로 존재하여 사용되어져야 하고 분산환경에서 객체호출이 이루지는 경우 동시적인 제어가 필요하다. 분산객체 시스템은 분산객체의 생성, 소멸, 저장, 복원과 같은 관리가 필요하다. 또한 분산객체의 위치 투명성을 보장하는 분산객체 이름관리가 필요하다. 즉, 어떤 분산된 환경 하에서도 객체의 이름을 통해 그 환경에 접근할 수가 있어야 한다는 것이다. 분산객체의 수정과 이에 따른 일관성유지가 필요하며, 분산객체의 소유권관리와 이에 따른 접근제어가 가능해야 한다. 이러한 분산객체가 가져야할 기능들을 수행하도록 호출 및 서비스를 관리하는 역할을 객체지향 미들웨어가 맡는다.

DCOM은 현재 널리 사용되는 Visual C++이나 Visual Basic 시리즈 등으로 개발이용이해 분산객체 분야에서 매우 우수한 산업표준의 하나라고 할 수 있다. DCOM은 컴포넌트기반 컴퓨팅환경을 지원하기 위해 도입한 COM(Component Object Model) 컴포넌트인 ActiveX 컴포넌트들이 분산환경에서 네트워크를 통해 다른 컴포넌트와 직접 통신할 수 있도록 하는 컴포넌트 프로토콜이다. 내부구조는 COM과 동일하며 DCOM은 언어중립적이기 때문에 ActiveX 컴포넌트를 작성하는 모든 언어를 사용하여 DCOM 응용프로그램을 만들 수 있다.[1][2]

2.2 인터넷기반 공정계획과 분산객체 모델

인터넷기반 공정계획시스템은 분산되어 있는 다수의 생산조직의 공정계획자들이 마치 한 장소에서 서로 대화를 하며 하나의 공정계획작업을 수행하는 것과 같은 효율성을 이루기 위한 것이다. 최근 전산환경의 발전에 따라 이러한 시스템을 실제로 구현할 수 있는 기반이 마련되었다. 인터넷기반 공정계획시스템의 운영에 있어서 설계와 생산자원에 관한 정보의 효과적 전달은 매우 중요하다. 여기서 설계정보는 공작물을 이루는 특징형상에 대한 정보로서 공작물의 형상, 치수, 공차, 표면 거칠기, 재질 등이며, 생산자원에 관한 정보는 기계와 공구의 제작가능 형상과 치수, 공차, 표면 거칠기, 각종 절삭조건 등이다. 이러한 정보교환은 설계자와 공정계획자, 그리고 공정계획자와 생산자간에 이루어지며, 항상 일관되게 각 생산분야로 전달되어야하고, 따라서 하나의 데이터베이스로 통합하여 관리하는 것이 필요하다.

설계대상 공작물은 각종 특징형상으로 구성된다. 따라서 공작물의 특징형상데이터를 분산환경하의 공정계획자들이 동시에 관찰하고, 설계상의 변화가 즉시 공정계획자에게 전달될 수 있어야 한다. 공정계획자로부터 작성된 공정계획은 생산현장으로 전달되어 각 가공기계에 할당된다. 생산 현장에서도 공정계획의 수정이 필요할 경우 이를 알려 즉각 조치되도록 하여 수정된 공정계획으로 작업이 신속히 이행되도록 해야 한다. 지금까지 설계자와 공정계획자, 그리고 공정계획자와 생산자간의 정보전달은 분리되어 이루어져왔으며, 공정계획시스템 또한 하나의 독립된 컴퓨터에서 작동하였기 때문에 분산환경에 적합한 기능을 구현하지 못했다. 공정계획 연관 정보전달에 있어서 작업간의 연계성이 높아서 분산환경에서 작업하기는 더욱 어렵다. 기존에는 분산환경에 적응하기 위해 도면을 파일로 전송하는 방법을 이용해 왔다. 그러나 이렇게 파일단위로 설계변경정보를 전달할 경우 공동작업을 수행하고 있는 다른 작업자가 자동적으로 알 수 있는 방법이 없으며, 또한 설계의 타당성 여부를 검증하기 위한 생산자원 정보교환도 이루어져야 하기 때문에 다른 기능과의 연결도 필요했다. 따라서 작업자와 작업자 사이에서 동적으로 설계정보나 공정계획의 변화를 공유할 수 있으며 그 타당성도 확인할 수 있는 시스템이 필요하게 되었다.

본 연구의 목적은 기업의 생산조직이 분산되어 있는 환경에서 인터넷기반 공정계획시스템 구축에 N-tier 분산객체모델을 적용시키는데 있다. 공정계획시스템은 공정생성, 공정검색 및 수정, 특징형상 정보관리, 생산자원 정보관리 등과 같은 기능을 포함한다. 정보관리 기능들을 인터넷상에서 수행하기 위해서는 다음과 같은 조건이 갖춰져야 한다.

① 별도의 프로그램 없이 웹브라우저만으로 공정계획 수립이 가능해야 한다. ② 공정계획 관련 정보들의 업데이트가 실시간으로 반영되어야 한다. ③ 데이터나 비즈니스 로직에 대한 보안이 확실해야 한다. ④ 비즈니스 로직의 변경을 각지에 분산되어있는 시스템에 최대한 빨리 반영해, 시스템 정비기간을 최소화시켜야 한다. ⑤ 업무량이 증가하거나 일시에 집중될 시에도 시스템의 성능저하를 최소화시켜야 한다.

본 연구에서는 공정계획시스템의 기본기능이 되는 특징형상과 생산자원 데이터의 관리를 위해서 우선적으로 각 데이터들을 효과적으로 표현하고 데이터베이스화하기 위해 데이터를 모델링 한다. 본 연구에서는 밀링머신을 대상으로 가공가능 특징형상, 기계, 공구 등의 데이터를 모델링 한다. 공정계획시스템이 인터넷에서 효과적으로 구현되기 위한 조건을 만족시키기 위하여 분산객체 모델을 적용한다. 인터넷상에서 공정계획을 수립하는 단계를 세분화하여 공정을 생성시키고, 생성된 공정에 공구, 기계와 같은 생산자원을 할당하는 단계별로 모델링된 데이터들을 이용하여 각각의 기능을 정의하고, 분산객체로 구현하며, 모델링된 데이터들을 데이터베이스화하기 위해 세부사항을 정의한다. 컴포넌트 개발에는 Microsoft Visual Basic, DBMS는 Oracle Enterprise Server, 웹서버로는 Internet Information Server, 그리고 DCOM 서버로는 Microsoft Transaction Server 등을 이용해 시스템을 구축한다.

3. 공정계획을 위한 데이터 모델링

3.1 특징 형상의 표현

본 연구에서는 가공을 위한 특징형상만을 다룬다. 여기서 특징형상이란 일련의 공정들에 의해 생성되며 설계나 제작의 관점에서 어떤 특별한 의미를 갖는 기하학적 형상 요소를 말하며, 이 특징형상을 통하여 가공할 부품의 설계 정보가 공정계획으로 전달된다. 특징형상의 표현에는 설계단계에서의 기하학적인 정보뿐만 아니라 다른 특징형상과의 관계 및 실제로 가공하기 위해서 요구되는 기술적인 정보들도 포함되어야 한다. 하나의 제품은 여러 개의 특징형상의 조합으로 표현되며 특징형상은 기하학적인 정보에 의해 다른 특징형상과의 관계를 나타낼 수 있고, 기술적인 정보에 의해 가공정보를 표시할 수 있다. 특징형상의 정보는 기하학적인 정보와 기술적인 정보로 나타낼 수 있다. 기하학적 정보는 너비, 길이, 깊이, 반지름 등으로 표현되는 형상의 크기와, 제품 내에 기본이 되는 면이나 축 등에 대해 표현되는 특징형상의 상대적인 위치, 그리고 제품 내에 특징형상의 기준 축이 향하는 방향 등으로 표현된다. 기술적인 정보는 표면거칠기와 공차, 재질 등으로 표현된다.[3][4]

3.2 공정과 생산자원의 표현

공정은 하나의 특징형상을 가공할 수 있는 가공방법이다. 공정은 특징형상과 대응된다. 평면은 평면가공 공정들에 의해 가공되며, 포켓은 엔드밀 등의 공정들에 의해 가공된다. 어떤 특징형상을 생성해내는 가공공정은 그 특징형상의 크기, 각종 공차, 표면거칠기

등에 따라 decision tree 형태로 표현한다. 한 공작물의 가공공정은 순서를 가진 공정의 집합으로 구성되므로 공정을 노드로 하는 순서도의 형태로 나타내어진다. 이런 것을 공정경로라고 한다. 이러한 경로는 형상, 크기, 공작물 소재, 표면 거칠기 등 앞서 설명한 특징형상 정보에 의해 규정된다. 공정계획은 여러 개의 가능한 경로를 가지는 공정들의 순서로 구성되며, 각 공정에 대해 그 공정을 행할 수 있는 절삭기계와 공구들이 선택된다. 절삭기계의 정보는 절삭가능 공작물크기, 소요동력, 가공공차, 장착 공구 등으로 표현된다. 공구는 기계에 장착 가능한 절삭공구의 종류를 의미하는데, 가공기계에 장착되는 방법이나 공구의 회전방향 등에 따라 여러 종류로 분류가 되며, 절삭공구는 기하학적인 형상과 가공할 수 있는 특징형상을 표현한다.[5][6]

4. 분산객체를 이용한 공정계획시스템

4.1 공정계획 수립

공정계획의 전체적인 흐름은 Fig.1에서 나타내었듯이 공정생성단계, 절삭기계 선택단계, 절삭공구 선택 단계, 공정 재구성 및 저장 단계로 나누어진다. 먼저 설계단계에서는 미리 정의된 특징형상의 매개변수를 조정하여 부품을 특징형상의 조합으로 표현하여 설계하며, 공정계획을 위해서 CAD 데이터에서 특징형상을 인식하는 과정을 통해 특징형상 정보를 표현한다. 본 연구에서는 특징형상의 설계 및 인식에 관한 연구는 포

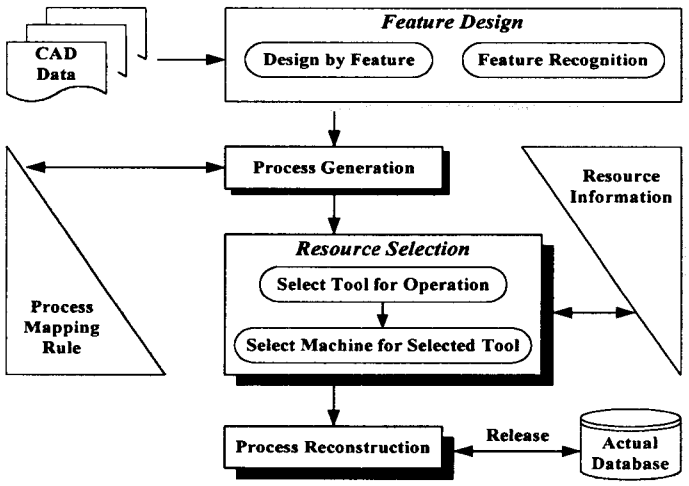

Fig. 1 공정계획 절차

함하지 않는다. 공정생성단계에서는 특징형상을 가공하는 공정을 process mapping rule로 표현된 공정계획 지식베이스를 이용하여 생성하는 것이다. 본 연구에서는 앞에서 설

명한 공정경로를 바탕으로 process mapping rule을 사전에 작성하고 CAD 데이터에서 얻어진 형상정보에 해당하는 공정을 찾아 공정을 생성시킨다. 절삭기계는 선택한 공정을 수행 가능한 기계의 리스트를 질의를 통해 생성시킨다. 절삭공구는 생산자가 입력한 정보를 바탕으로 구축된 데이터베이스에 생성된 공정에 적합한 공구를 질의를 통해 리스트를 추출하여 공정계획자가 직접 선택하도록 설계되었다.[5][6][7]

4.2 분산객체를 이용한 시스템 구축

네트워크를 이용하여 시스템을 개발하는 데 있어 가장 큰 어려운 점은 개개의 컴포넌트들을 변경, 배포, 관리하는 문제이다. 공정계획 지식베이스이나 공정계획 interface의 변경이 일어났을 때, 분산되어 있는 설계지와 생산지를 대상으로 이러한 컴포넌트의 변경을 동시에 이루는 것은 간단하지 않다. 이러한 컴포넌트의 배포와 관리 문제를 해결하기 위해 등장한 것이 앞에서 설명한 DCOM이고, 본 연구에서는 공정계획에 관련된 비즈니스로직 컴포넌트를 DCOM 구조를 따르는 ActiveX 컴포넌트로 구현했다. 본 시스템은 N-tier로 구성되어있으며 크게 3가지로 나누면 데이터베이스 계층, 비즈니스로직 컴포넌트 계층, 프레젠테이션 계층으로 나누어지며, Fig.2에서 각 계층간의 연결이 보여지고 있다.

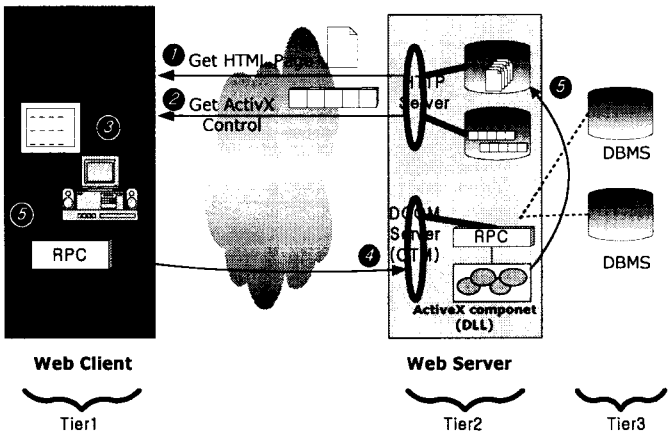

Fig. 2 시스템 계층간의 연결

4.2.1 데이터베이스 계층

데이터베이스는 공정계획을 수행하기 위한 입력정보가 들어가는 기초정보 데이터베이스와 공정계획절차가 진행된 후 실제 공정계획 결과가 저장되는 출력정보 데이터베이스로 구성되어 있다. 이 계층은 데이터베이스 입출력 컴포넌트들이 보내온 데이터베이스 사용자정보와 질의문(SQL; Structured Query Language)을 이용해 실제 질의를 수행하고, 출력질의일 경우 그 결과값을 레코드셋 형태로 데이터베이스 출력 컴포넌트에게 반환하고, 입력질의일 경우 그 값을 물리적으로 저장한다.

기초정보 데이터베이스는 공정계획에 필요한 입력정보가 저장되는 곳으로 process mapping rule, 특징형상정보, 절삭기계정보, 절삭공구정보가 이곳에 저장된다. Fig.3에 기초정보 데이터베이스의 E-R 다이어그램을 나타내고 있다. 기초정보 데이터베이스의

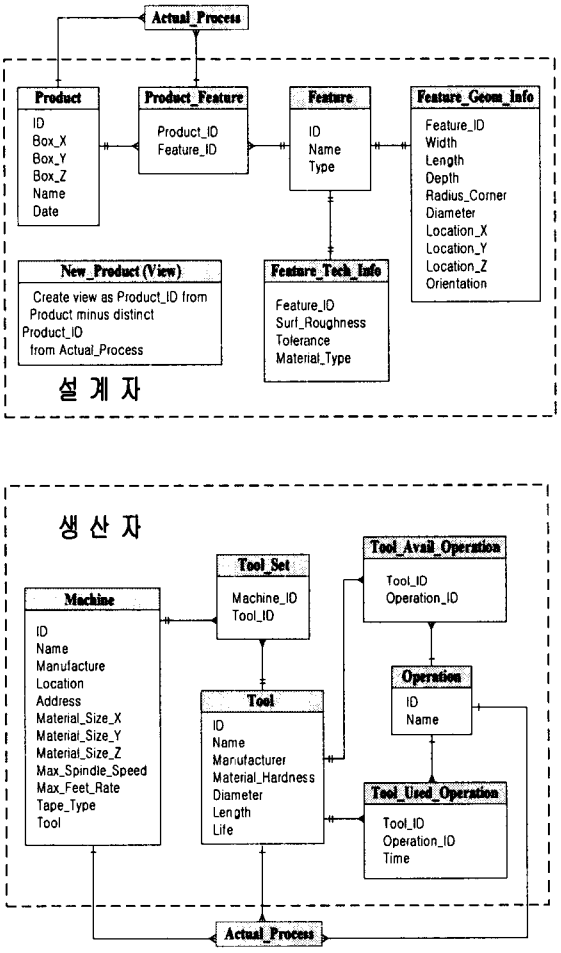

Fig. 3 기초정보 DB의 ER다이어그램

특징은 설계자들이나 공정계획자들이 주로 접근하며 읽기 작업이 쓰기 작업 보다 훨씬 빈번하게 일어나며 저장공간의 성장속도가 그리 빠르지 않다는 것으로, 출력대역폭이 넓고, 저장공간이 그리 크지 않은 서버에 설치할 수 있다. 기초정보 데이터베이스는 생산자 측 데이터를 관리하는 생산자원 데이터베이스와 설계자 측 데이터를 관리하는 설계정보 데이터베이스로 나뉜다.

출력정보 데이터베이스는 공정계획이 시작되면 중간결과물이나 배포될 최종 공정계획이 저장되는 곳이다. 출력정보 데이터베이스의 특징은 공정계획자들과 생산현장에서

주로 접근하며 쓰기 작업이 읽기 작업보다 빈번하게 일어나며, 저장 공간의 성장속도가 기초정보 데이터베이스보다 상대적으로 빠르다. 쓰기 작업의 특성상 회선이 안정적이고, 저장공간이 크거나 추가가 쉬운 서버에 설치하는 것이 바람직하다.[5][7]

4.2.2 비즈니스 로직 컴포넌트 계층

이 계층은 실제로 데이터베이스에 접근해서 공정계획자가 공정계획 작업을 수행하도록 지원하는 모든 로직을 구현하는 컴포넌트들이 들어있는 곳으로 데이터베이스 입출력 컴포넌트 층과 공정계획용 컴포넌트 계층으로 구별된다. Fig.4에 각 컴포넌트와 인터페이스들이 표시되어 있다. 각 컴포넌트들은 ActiveX DLL(Dynamic Link Library)로 구성되어 있다. 지금까지 인터넷에서 비즈니스로직을 구현하는 방법으로 CGI(Common Gateway Interface)나 각종 클라이언트나 서버 측 스크립트가 사용되어 왔다. 모두 클라이언트 아니면 서버에서 각 언어(C, Perl, etc.)나 스크립트(Java Script, VB Script)들을 컴퓨터에서 실행시킬 수 있는 이진코드로 해석해야하기 때문에 속도나 컴퓨터 자원활용 측면에서 비효과적이었다. 하지만 본 연구에서 사용한 DLL 컴포넌트는 이미 이진코드로 컴파일되어 있는 상태이기 때문에 스크립트언어에 비해 서버에 많은 요구가 단기간에 집중될 경우 속도나 서버의 자원활용 면에서 월등히 우수하다.

데이터베이스 입출력용 컴포넌트층은 비즈니스로직 중에서도 공정계획 작업과 직접적인 관련이 없는 컴포넌트들로 공정계획용 컴포넌트층에서 보내오는 데이터베이스 접

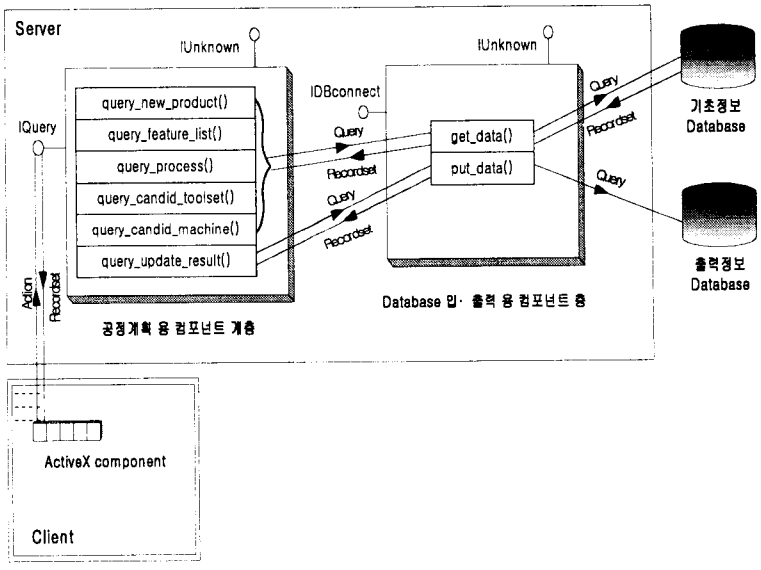

Fig. 4 비즈니스로직 콤포넌트와 인터페이스

근 사용자 정보와 데이터베이스 질의문들을 매개변수로 받아 데이터베이스 계층으로 질

의를 의뢰한 후 돌아오는 결과 값을 다시 공정계획용 컴포넌트들에게 반환한다. 데이터베이스 입출력 컴포넌트 층을 분리한 이유는 첫째로 여러 종류의 데이터베이스에 연결할 경우 데이터베이스용 미들웨어를 사용해야하는데 이 컴포넌트들이 동작하는 서버에는 반드시 데이터베이스용 미들웨어가 설치되어 있어야 하기 때문이다. 따라서 데이터베이스 입출력로직이 다른 컴포넌트와 통합되어있을 경우 다른 컴포넌트가 설치되어있는 서버에도 데이터베이스용 미들웨어를 설치해야 하기 때문이다. 둘째로, 이 컴포넌트들은 다른 공정계획용 컴포넌트들과 달리 하드웨어나 프로그램 개발용 API(Application Programming Interface)에 종속적이기 때문에, 공정계획로직상의 변경이 없더라도 하드웨어나 프로그램개발용 API가 업그레이드될 경우에 변경 사항이 생길 수도 있기 때문이다. 세째로, 데이터베이스에 접근하는 창구를 일원화하여 보안문제나 데이터베이스 접근관리를 효과적으로 할 수 있다. 네째로, 컴포넌트들을 세분화시킴으로 해서 융통성 있는 네트워크의 부하분산이 가능하다. 즉, 모든 데이터베이스는 이 컴포넌트들을 이용해 접근해야 하므로 병목이 생길 수 있는데 이럴 경우 이 컴포넌트들을 여러 군데로 분산시켜 병목을 해소할 수 있다.

공정계획용 컴포넌트계층은 실제로 공정계획 로직을 구현하는데 필요한 컴포넌트들로서, 새로 디자인된 부품의 발견, 부품을 구성하는 특징형상 발견, 발견된 특징형상과 공정계획 적용, 얻어진 공정에 적합한 절삭기계 추천, 절삭기계내의 공구선택 등의 기능들을 수행하여 공정계획자가 작업을 하는 클라이언트와 직접 교신하며 결과물을 클라이언트로 전송한다.

4.2.3 프레젠테이션 계층

이 계층은 공정계획자와 설계자, 그리고 생산현장에서의 시스템의 사용자들이 시스템에 접근해서 조작하는 계층으로 본 인터넷기반 시스템은 웹브라우저가 클라이언트 프로그램이 된다. 이 계층은 클라이언트에 존재하면서 공정계획 컴포넌트들과 교신하며 사용자가 원하는 작업을 원활히 수행할 수 있도록 지원한다. 기존의 HTML 문서로는 공정계획 작업을 수행하는데 필요한 기능을 구현하기가 쉽지 않기 때문에 본 시스템은 ActiveX 컨트롤 컴포넌트를 이용한 프레젠테이션 계층의 응용프로그램을 구현하였다. ActiveX 컨트롤도 컴포넌트 모델이 가지는 장점을 그대로 계승받으며, 역시 이진 코드로 구성되어 있기 때문에 속도가 빠르고 인터넷 브라우저 상에서 일반 어플리케이션이 가지는 모든 기능들을 구현할 수 있다.

5. 시스템 구현 및 결과

공정계획시스템의 주축이 되는 컴포넌트들을 DCOM 규격을 따르는 OTM(Object Transaction Monitor)에 등록시켜서 각 컴포넌트들이 동작하도록 해야한다. OTM은 분산객체용 응용프로그램서버로 서버측 컴포넌트를 교대로 실행하는 컨테이너 집합을 관리한다. 사용자가 컴포넌트를 호출하면 컨테이너는 필요한 기능을 구현하는 컴포넌트를 활성화시키고 컴포넌트에 요청을 전달한다. OTM은 컴포넌트를 자동으로 관리하고, 트

랜잭션, 견고성, 지속성, 보안등의 기능을 제공한다. 데이터베이스 입·출력 컴포넌트는 MTSDBConn 컨테이너에 공정계획 용 컴포넌트는 MTSQuery 컨테이너에 각각 패키징 되어 별도로 계층을 형성하고있다. 본 시스템에서는 두 개의 계층을 하나의 서버 내에 위치시켰으나 부하가 집중될 시에는 경우에 따라 MTSQuery 컨테이너를 다른 원격서버에 옮길 수도 있다. IUnknown 인터페이스의 메서드들인 AddRef, QueryInterface, Release와 OTM이 작동하는데 필요한 몇 가지 기본적인 메서드들이 인터페이스에 추가되어있다. 웹서버와 DCOM의 OTM이 정상작동되고 OTM내에 사용할 서버컴포넌트의 패키징이 끝나면 클라이언트로 사용되는 웹브라우저로 공정계획용 인터넷사이트에 접속한다.

본 시스템은 클라이언트로 웹브라우저 내에 ActiveX 컨트롤을 사용하고 있기 때문에 최초 접속 시 필요한 컨트롤을 클라이언트에서 자동으로 설치한다. ActiveX 컨트롤은 클라이언트 시스템 내에 자동으로 설치된다. 컨트롤을 다운로드한 후에 사용자 인증과정을 거치면 작업선택화면이 표시되고 사용목적에 맞는 기능을 선택하게된다. 공정계획자가 설계자로부터 입력된 특징형상의 각종 정보를 조회하거나, 설계자가 설계의 바탕이 되는 기초 특징형상의 변동사항이 생기거나 추가해야 할 형상이 생기면 즉시 정보를 업데이트시킬 수 있다. 우선 기초정보 데이터베이스에 접속하기 위한 사용자인증을 마친 후 부품을 선택하면 부품을 구성하는 형상들이 나열되고 각 형상을 선택하면 설계정보 조회화면에 선택된 형상에 대한 정보가 표시된다. 공정계획자가 새롭게 디자인된 부품을 선택해서 그 공정계획을 수립하기 위해 설계형상을 선택하면 공정계획을 수립하기 위한 그리드에 서버 컴포넌트 내에 query_process() 메서드를 활성화시켜 얻어진 공정들이 나열되며, 각 공정에 대한 공작기계와 공구에 대한 정보는 query_candid_machine()과 query_ candid_toolset() 메서드를 활성화시켜 그 결과 값들을 콤보박스형태로 제공하여 공정계획자가 선택할 수 있게 되어있다. 그리고 각 작업의 순서를 정해주고 공정저장 버튼을 누르면 공정결과가 정해진 순서에 따라 결과를 재구성해서 생산자가 조회해볼 수 있도록 저장된다. 이때 작업지시 여부를 물어 릴리즈를 선택하면 생산자가 조회가 가능해진다. 생산자가 공정계획 수립이 끝나서 작업지시된 부품을 선택하여 그 공정을 조회하면 부품선택 창에는 공정계획자가 작업을 마친 부품만 query_process() 메서드를 활성화시켜 조회된다. Fig.5에 공정계획 조회화면이 예로서 보여진다. 생산자 측에서 보유한 생산자원에 대한 변동사항이 생기면 즉시 차후 공정계획 수립에 반영되도록 하기 위해서 그 변동사항을 수정하는 기능의 화면이 제공되어 정보를 수정한 후 저장명령을 내리는 순간 즉시 query_update_result() 메서드를 활성화시켜 데이터베이스에 반영시킨다. Fig.6에서 이 기능의 화면을 보여주고 있다.

개발된 시스템은 웹브라우저만으로도 공정계획을 수행할 수 있으며, 데이터베이스 서버와 웹서버, DCOM서버를 통합하여 실시간으로 데이터의 변경사항이 적용되게 하였다. DCOM서버와 데이터베이스서버를 분리시키고 DCOM서버 내에서도 데이터베이스 입·출력 용 컴포넌트인 IDBConnect와 공정계획용 컴포넌트인 IQuery를 분리시켜 패키징하였기 때문에 필요할 경우 서로 다른 서버에 각 패키지를 나누어 설치하여, 부하를

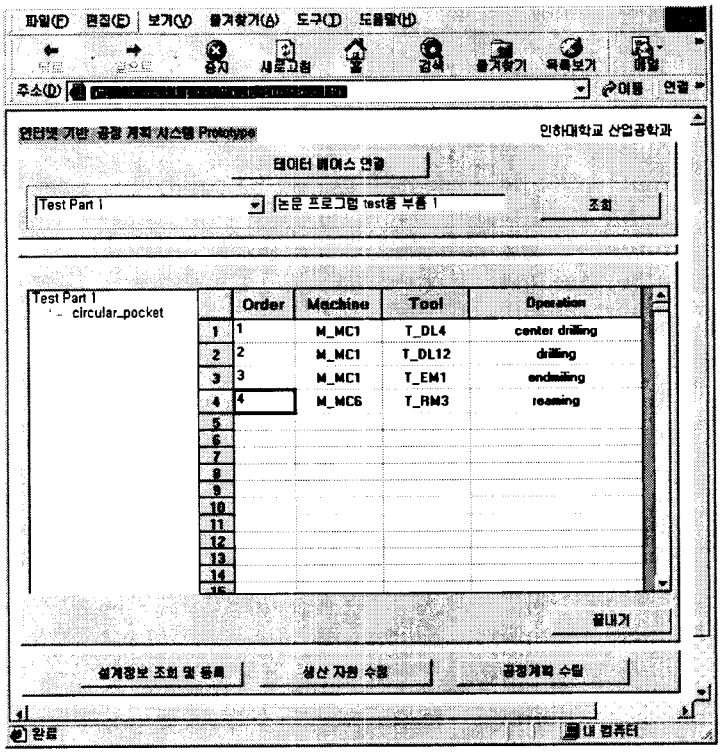

Fig.5 공정계획 조회

분산시켜 업무량의 증가하거나 일시에 집중될 시에도 시스템의 성능저하를 최소화시킬 것으로 기대된다. 또한 데이터베이스 접근 시 항상 인증 절차를 삽입하고, 비즈니스 로직을 DLL 이진코드로 컴파일하였기 때문에 데이터나 로직 유출에 대한 보안수준이 높을 것으로 기대된다.

6. 결론

본 연구에서는 분산객체를 이용하여 공정계획 과정을 설계자, 공정계획자, 생산자들이 효과적으로 정보를 공유하며 수행할 수 있는 모델 및 방법을 제시하였다. 분산시스템에 중점을 두어 시스템을 개발하였다. 본 연구에서 제시된 공정계획시스템을 이용하면 각각의 비즈니스 로직을 이진 컴포넌트로 만들어 처리속도를 향상시키고, 각 컴포넌트를 서버에 분산 배치하는 방법을 이용해, 서버에 걸리는 부하를 분산시켜 정보처리 효율을 높임으로 해서 안정적인 서버의 운영이 가능해진다. 또한, 클라이언트 역시 웹브라우저에 삽입된 ActiveX 컴포넌트를 사용하기 때문에 공정계획을 수행할 별도의 응용프로그램 없이 어디서나 접근이 가능하게 되어, 광범위하게 산재한 생산조직 간에 신속한 정보

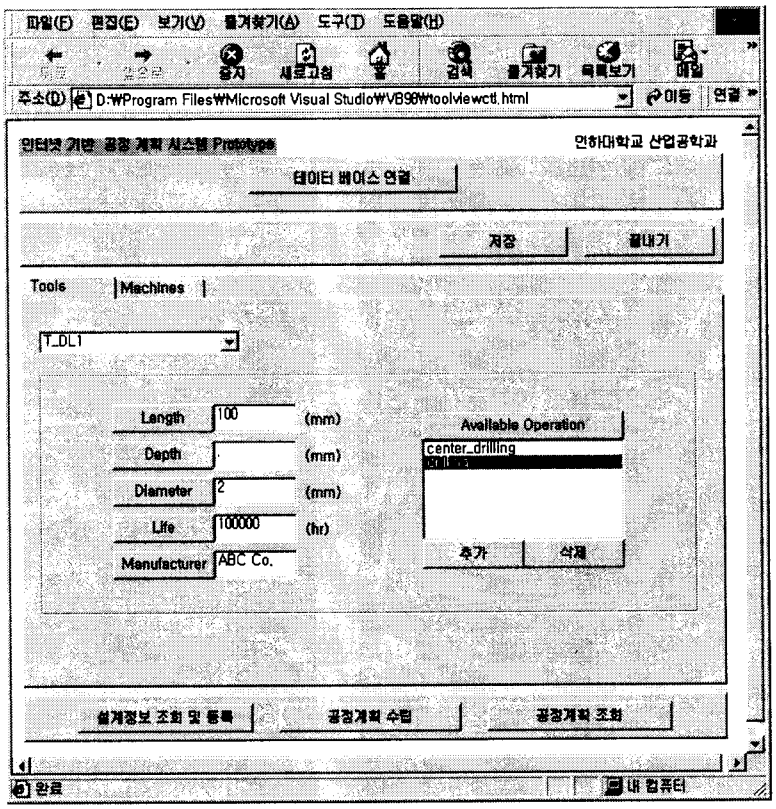

Fig.6 생산자원 정보 수정

교류가 가능해질 것이다. 하지만 시스템의 개발뿐만이 아니라 시스템을 운영하는 능력, 즉, 관리기법 또한 중요하다. 본 연구에서 제시한 시스템은 그 형태상 단일 컴퓨터시스템이나 2-tier 클라이언트 시스템보다 복잡하여 관리하기가 더욱 어렵기 때문에 효과적인 서버배치나 운영기법에 대한 연구도 따라야 할 것이다. 본 연구에서 제시한 시스템은 생산조직간에 존재할 수 있는 정보의 흐름 중 일부분만을 다루었기 때문에 다른 영역의 정보의 흐름을 처리하는 시스템의 개발이 있어야 할 것이고, 그러한 시스템과 본 연구에서 제시한 시스템과의 통합을 통해 인터넷이라는 가상공간을 기업활동의 주무대로 삼는 기업에게 더욱 더 많은 도움을 줄 것으로 기대된다.

후기

본 연구는 1998년도 인하대학교 연구비 지원에 의하여 수행되었음.

7. 참고문헌

[1] Whitten, J. L., L. D. Bentley, *System Analysis and Design Methods*, McGraw-Hill Inc., 1998.

[2]. Park, H. K., J. M. Baik, "Enhancing Manufacturing Product Development through Learning Agent System over Internet", *Computers & Industrial Engineering*, Vol37, 1999.

[3] Case, K., J. Gao, "Feature technology: an overview", *International Journal of Computer Integrated Manufacturing*, Vol.6, 1993.

[4] Chan, K. C., J. Nhieu, "A Framework for Feature-Based Applications", *Computers & Industrial Engineering*, Vol24, No.2, 1993.

[5] Chang, T., R. A. Wysk, and H. P. Wang, Computer-Aided Manufacturing, Prentice-Hall Inc, 1991.

[6] Lee, H., "A Generic Learning System for CAPP," Ph.D. Dissertation, The Pennsylvania State Univ., 1991

[7] Rembold, U., B. O. Nnaji, and A. Storr, Computer Integrated Manufacturing and Engineering, Addison-Wesley Inc., 1993.