

**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(51) Int. Cl.<sup>7</sup>  
G06F 15/163

(11) 공개번호 특2001-0043617  
(43) 공개일자 2001년05월25일

(21) 출원번호	10-2000-7012770		
(22) 출원일자	2000년11월14일		
번역문제출일자	2000년11월14일		
(86) 국제출원번호	PCT/US1999/10711	(87) 국제공개번호	WO 1999/60487
(86) 국제출원출원일자	1999년05월14일	(87) 국제공개일자	1999년11월25일
(81) 지정국	AP ARIPO특허 : 케냐 레소토 말라위 수단 스와질랜드 우간다 시에라리온 가나 감비아 짐바브웨 EA 유라시아특허 : 아르메니아 아제르바이잔 벨라루스 키르기즈 카자흐스탄 몰도바 러시아 타지키스탄 투르크메니스탄 EP 유럽특허 : 오스트리아 벨기에 스위스 독일 덴마크 스페인 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투갈 스웨덴 핀란드 사이프러스 OA OAPI특허 : 부르키나파소 베냉 중앙아프리카 콩고 코트디부와르 카메룬 가봉 기네 말리 모리타니 니제르 세네갈 차드 토고 기네비소 국내특허 : 알바니아 아르메니아 오스트리아 오스트레일리아 아제르바이잔 보스니아-헤르체고비나 바베이도스 불가리아 브라질 벨라루스 캐나다 스위스 중국 쿠바 체코 독일 덴마크 에스토니아 스페인 핀란드 영국 그루지야 헝가리 이스라엘 아이슬란드 일본 케냐 키르기즈 북한 대한민국 카자흐스탄 세인트루시아 스리랑카 라이베리아 레소토 리투아니아 룩셈부르크 라트비아 몰도바 마다가스카르 마케도니아 몽고 말라위 멕시코 노르웨이 뉴질랜드 슬로베니아 슬로바키아 타지키스탄 투르크메니스탄 터어키 트리니다드토바고 우크라이나 우간다 우즈베키스탄 베트남 폴란드 포르투갈 루마니아 러시아 수단 스웨덴 싱가포르 그레나다 가나 감비아 크로아티아 인도네시아 인도 시에라리온 유고슬라비아 짐바브웨		
(30) 우선권 주장	60/085,539 1998년05월15일 미국(US)		
(71) 출원인	트리디움 인코퍼레이티드 미국 버지니아주 (우편번호 : 23233) 리치몬드 웨스트레 파크웨이 3951 슈트 350		
(72) 발명자	프랭크제랄드엘. 미국버지니아주(우편번호:23233)리치몬드길스퍼로드1405 왈퀴스트대니제이.쥬니어 미국버지니아주(우편번호:23233)패터슨웜벌리드라이브11201 프랭크브라이언에스. 미국버지니아주(우편번호:23220)리치몬드하노버애버뉴1926 서브렛존더블유. 미국버지니아주(우편번호:23233)리치몬드체이스웰레슬레이드라이브#63311829 기오기스다니엘피. 미국버지니아주(우편번호:23060)글렌알렌와일드트리드라이브9019 록웰윌리엄지. 미국버지니아주(우편번호:23236)리치몬드에스티.레기스드라이브2609 올굿오티이. 미국버지니아주(우편번호:23116)메카닉스빌섬머힐로드10490 아담스로버트에이. 미국버지니아주(우편번호:23112)미드로티안왈넛크릭코트2705		
(74) 대리인	김명신, 이동기, 최민서		

심사청구 : 없음

## **(54) 컴퓨터 네트워크를 사용하는 이종 전기기계 시스템의 객체지향 제어를 위한 시스템 및 방법**

### **요약**

제어 시스템 방법론은 공통 객체 모델에 복수의 제어 시스템을 결합하기 위해 객체지향 소프트웨어를 사용한다. 객체지향 기술은 다양한 크기 및 구성의 시스템을 제어하고 감독하는데 사용하기 위한 멀티-벤더 개방형 시스템 환경에서의 분산형 자바 기반 애플리케이션을 구성하는데 사용된다. 이 시스템은 브라우저 클라이언트(101)(자바 가능 브라우저에서 실행하는)와 서버 클라이언트(121)를 모두 제공한다. 본 발명은 복수 국, 웹 BAS 브라우저(9110), 네트워크 프로세서(NP)(111) 및 필드 컨트롤러(FC)(112)를 갖는다. 바람직하게 NP 및 FC는 플러그인 자바 모듈 환경에서 구현된 자바 가상 머신이다. 정보는 공통 객체 모델로 되어, 시스템 전체에 걸쳐서 이용 가능하게 된다. 자바에 기초한 커스텀 프로그래밍 언어가 객체 생성을 위해 제공되고, 객체에 대한 접근은 다중 레벨 보안 프로토콜을 통해 제어된다. 데이터 흐름은 실시간 정보 동기화 관리자에 의해 관리된다.

### **대표도**

### **도9**

### **명세서**

본 출원은 1998년 5월 15일 출원된 미국 가출원 No. 60/085,539의 이점을 주장하고, 그의 전체 명세서는 참조상 여기에 포함된다.

이 명세서의 일부는 출원인이 저작권을 주장하는 요소를 포함한다. 출원인은 출원 파일의 복사본을 만들거나 출원에 대해 특허를 받는 과정에서 상기 요소의 복사에 반대하지 않지만, 저작권으로 보호된 요소에서 그 밖의 다른 모든 권한은 확보된다.

### **기술분야**

본 발명은 일반적으로 제어 시스템과 연관된 로컬 컨트롤러에서 가동되고, 인터넷과 같은 통신 네트워크를 사용하여 중앙 제어, 모니터링 및 갱신되는 모듈 객체지향 제어 소프트웨어를 사용하여 다양한 시스템을 제어 및 모니터링하는 시스템 및 방법에 관한 것이다. 이 시스템 및 방법들은 특히 빌딩 자동화 시스템(BAS) 애플리케이션에 유용하다.

### **배경기술**

과거에, 빌딩 자동화 시스템과 같은 제어 시스템은 통상 배타적인 단일 제조사 솔루션 또는 독립 시스템의 복잡한 결합물이었다. 상기 시스템들이 보다 효율적으로 결합될 수 있도록 표준을 규정하기 위한 산업계의 노력이 있었지만, 진실로 규칙적인 방법이 이용 가능하지는 않았다.

빌딩 자동화 시스템에 대한 BACnet / LonMark™ 필드버스 통신 표준의 발표는 제어 솔루션 전망에 변혁을 일으킬 수 있는 빌딩 자동화 시스템 아키텍처를 다시 생각하기 시작했다. 통상의 배타적인 빌딩 자동화 아키텍처는 양호하게 수행하지만, 설계 및 설치 비용이 비싸고, 재구성하기 곤란하며, 자립형 기구 서비스를 결합하기 위해 많은 소프트웨어를 종종 필요로 한다. 반대로, 분산형 BACnet / LonMark 필드버스 아키텍처에서 논리 프로세싱은 빌딩 제어 문제를 협력적으로 해결하기 위해 표준 소프트웨어를 통해 서로 통신하는 자동 멀티-벤더 모듈내에 포함된다. 그러나, 유지된 빌딩 자동화 아키텍처에서 제어 시스템에 추가되는 장치의 외부 인터페이스 특성은 BAS가 새로운 장치를 효율적으로 제어할 수 있기 전에 장치의 외부 인터페이스 특성의 데이터베이스 표현내에 입력되어야 한다.

이와 같은 방식으로 제어를 분산하는 것은 설치 배선을 최소화하고, 제어 소프트웨어를 최적화하고, 운영/유지 비용을 절감하며, 시스템 확장을 간편화할 것으로 기대된다. 오늘날의 자유 경쟁에 의한 세계 시장에서 사용자는 기구 관리 서비스의 변화에 재빨리 대응하기 위해 필드버스의 가용성, 재사용 가능성 및 비용면에서 효율적인 빌딩 제어 솔루션을 요구한다.

그러나 다른 분야에서는 빌딩 자동화 시스템 분야에 충분히 적용되지 않았던 컴퓨팅의 진전이 있었다. 예를 들어 시스템을 프로그램하는 비용 및 개발 요구를 간편하게 하기 위해 컴퓨터 시스템 표준이 개발되었다. 기본적인 기술은 Java™ 가상 머신(JVM)과 JavaBeans™과 같은 객체지향 소프트웨어를 포함한다.

객체지향 프로그래밍은 종래의 프로그래밍 메커니즘보다 양호한 현실 세계와 부합하는 소프트웨어 빌딩 블록으로서 현실 세계 실체를 보다 직접적으로 모형화하는 새로운 방법을 제공한다. 마이크로프로세서가 하드웨어 설계 및 기능성에 충격을 준 것처럼 이 새로운 소프트웨어 패러다임은 소프트웨어 분야에 미치는 충격이 점점 커진다.

객체지향 프로그래밍에서, 애플리케이션 프로그램은 독립 빌딩 블록 모듈(객체)로부터 생성된다. 애플리케이션 엔지니어는 새로운 프로그램을 생성할 때마다 처음부터 하기 보다는 라이브러리로부터 미리 작성된 애플리케이션 모듈을 빌려 올 수 있고, 이들을 간단하게 설치 및 사용할 수 있다. 객체 라이브러리를 사용함으로써 개발자는 적당한 객체를 서로 연결하고, 애플리케이션이 필요로 하는 커스텀 객체를 작성하는 것에 대해서만 집중할 수 있다. 다시 커스텀 객체는 이후의 사용을 위해 라이브러리에 추가될 수 있다.

또한 시스템을 결합하는 비용 및 개발 요구를 간편하게 하기 위한 노력으로 네트워킹 표준을 개발하였다. 빌딩 자동화 산업계에서 개발된 기본 기술은 미국냉난방공조기술자협회(ASHRAE) 빌딩 및 제어 네트워크(BACnet™) 및 ECHELON LONTALK™ 프로토콜(미국 캘리포니아주 팔로 알토의 Echelon Corporation에 의해

개발)을 포함한다.

한편, 컴퓨터 산업계에서는 통신 표준을 개발하기 위해 노력했지만, 이들 표준들은 빌딩 자동화의 특정 문제에 대한 적당히 효율적인 방법으로 적용되지는 않았다. 예를 들어 인터넷 / WWW / Java / CORBA 표준들의 목적은 어떤 종류의 프로그램을 사용하는지, 어떤 컴퓨터에서 실행되는지 또는 정보의 형식이 무엇인지는 고려하지 않고 사용자가 원하는 정보를 제공하기 위한 것이다.

하이퍼텍스트의 체계를 통해, 웹의 사용자는 전세계로부터 정보를 선택하고 검토할 수 있었다. 그러나, 기본적인 웹은 적당한 대화성, 즉 사용자와 애플리케이션 사이의 실시간, 동적 및 시각적인 대화가 부족하다. 자바는 이 결여된 대화성을 웹에 제공한다. 자바 가능 웹브라우저를 사용하여 사용자는 애니메이션과 대화형 애플리케이션을 마주할 수 있다.

패킷 전송 시간의 변화도와 인터넷을 통한 보증된 송달의 결여 때문에 인터넷에 대한 실시간 제어 시스템을 구현하기 위한 노력은 일반적으로 실패하였다.

결론을 말하면, 역사적으로 제어 시스템은 기본적으로 단일 제조자의 배타적인 기술에 기초를 두고 있었다. 이러한 아키텍처들을 표준화하기 위한 노력이 있었지만, 이들은 다만 제한적인 성공을 달성하였다. 본 발명자들은 제어 시스템의 분야에서 특히 빌딩 자동화 애플리케이션에서 사용하도록 적응된 자바 기반 공통 객체 모델을 결합하고, 인터넷과 같은 통신 네트워크 표준을 사용하여 중앙 제어 및 모니터링을 제공하는 고도로 집적화된 표준에 대한 요구가 있다고 믿는다.

### **발명의 상세한 설명**

따라서, 본 발명의 일반적인 목적은 BACnet, LonMark, CORBA, 자바 및 인터넷 시스템을 하나의 공통 객체 모델로 결합하는 객체지향 소프트웨어를 사용하는 시스템 및 방법을 제공하는데 있다. 바람직한 실시예에서 본 발명은 복수의 산업계 표준을 지원하는 멀티-벤더 개방형 시스템 환경에서 적당히 분산된 애플리케이션을 구성하는 것에 대한 키로서 객체 기술을 사용한다. BACnet / LonMark / 인터넷 / CORBA 아키텍처는 본 발명에서 조합되고, 분산형 제어 및 정보 관리를 실시하기 위해 자바 객체지향 소프트웨어 개념이 적용된다. 메시지(이벤트)는 데이터 및 기능성을 모두 포함하도록 객체 기술을 사용하는 자동 모듈 사이에서 전달된다. 이러한 기능적인 객체들은 재사용을 위해 복제되고, 다른 객체의 작업으로 구축하는 보다 복잡한 기능을 형성하기 위해 그룹화된다. 웹 BAS 서버의 내비게이션 도구들은 자동 필드버스 모듈에 의해 제어되고 있는 프로세스의 개관을 제공하고, 제한되지 않은 정보 흐름을 제공한다.

본 발명의 다른 목적은 미리 정의된 객체와 사용자 정의 형성 객체가 모두 단일 제어 시스템에 존재하는 제어 시스템 및 방법을 제공하는데 있다. 객체 수정, 생성 및 관리를 위해 개별화된 프로그래밍 언어가 제공된다. 미리 정의된 객체 세트가 또한 제공된다.

본 발명의 다른 목적은 분산형 시스템에서 시스템 데이터의 무결성을 유지하고, 아카이빙(archiving) 및 동기화 기술을 통해 분산형 시스템에 걸쳐서 요구에 따라 시스템 데이터를 분산하는 제어 시스템 및 방법을 제공하는데 있다. 각각의 데이터의 마스터 카피는 단일 가상 머신에 유지되는 것이 바람직하고, 동기 및 비동기 캐시를 갖는 시스템이 제어 시스템의 객체가 현재 시스템 데이터에 접근하는 것을 보장하기 위해 사용된다. 시스템 데이터는 지속성을 제공하기 위해 아카이빙된다.

본 발명의 다른 목적은 한 장치와 연관된 객체 특성이 그 장치의 외부 인터페이스 특성을 반영하기 위해 동적으로 갱신되는 제어 시스템 및 방법을 제공하는데 있다. 이를 달성하기 위해 제어 시스템의 장치들은 그들의 외부 인터페이스 특성과 그에 따라 갱신된 연관된 객체 특성에 대해 조회를 받는다. 이것은 미리 프로그램된 특정한 외부 인터페이스 표현을 필요로 하지 않으면서 장치들의 관리를 제공한다.

본 발명의 이러한 목적 및 다른 목적들은 부록 및 도면을 포함하는 본 명세서를 살펴보면 명백하다.

### **도면의 간단한 설명**

도 1은 클라이언트, 국, 외래 장치 및 원격 호스트를 포함하는 객체지향 제어 시스템의 바람직한 일반적인 아키텍처이다.

도 1A는 객체지향 제어 시스템의 여러 가지 형태의 국들을 나타낸다.

도 2는 클라이언트가 LonMark™ 필드버스를 감독하는 단일 필드 컨트롤러에 접근하는 필드 컨트롤러 시스템의 형태로 객체지향 제어 시스템의 구성에 대한 대체 실시예를 보여준다.

도 3은 클라이언트가 LonMark™ 필드버스를 감독하는 단일 웹 BAS 서버에 접근하는 소형 웹 BAS 서버 시스템의 형태로 객체지향 제어 시스템의 구성에 대한 대체 실시예를 보여준다.

도 4는 웹 BAS 서버가 네트워크 프로세서, 필드 컨트롤러 및 BACnet 장치를 직접 감독하는 2레벨 시스템의 형태로 객체지향 제어 시스템의 구성에 대한 바람직한 실시예를 보여준다.

도 5는 웹 BAS 서버가 필드 컨트롤러를 감독하는 네트워크 프로세서를 감독하는 3레벨 시스템의 형태로 객체지향 제어 시스템의 구성에 대한 대체 실시예를 보여준다.

도 6은 대형 또는 소형 시스템의 집합이 원격 호스트를 통해 접속되고 감독되는 분산형 시스템의 형태로 객체지향 제어 시스템의 구성에 대한 대체 실시예를 보여준다.

도 7은 객체지향 제어 시스템에 대한 지속적인 실시간의 정보 동기화 관리자의 바람직한 실시예에 대한 설명이다.

도 8은 객체지향 제어 시스템의 객체 클래스 계층의 개관이다.

도 9는 객체지향 제어 시스템의 바람직한 실시예의 국과 클라이언트 사이의 통신을 기술한다.

도 10은 객체지향 제어 시스템의 바람직한 실시예의 국과 클라이언트 사이의 통신을 추가로 기술한다.

도 11은 객체지향 제어 시스템의 바람직한 실시예의 아키텍처를 기술한다.

## 실시예

객체지향 제어 시스템(100)의 바람직한 실시예는 여러 가지 크기 및 구성의 시스템들의 분산형 제어에 대한 솔루션을 제공한다. 바람직하게, 도 1에 도시된 바와 같이 클라이언트(101), 국(102)(도 1A 참조) 및 외래 장치(105)를 포함하는 일반적인 아키텍처가 제공된다.

상기 시스템에서 클라이언트(101)는 브라우저 클라이언트(120)와 풀 클라이언트(121)를 포함한다. 클라이언트(101)는 인터넷 또는 인트라넷을 통해 시스템에 대한 접근을 제공한다. 클라이언트(101)는 이 접근을 제공하기 위해 국의 서버 소프트웨어에 의존한다. 브라우저 클라이언트(120)는 승인된 자바 가능 브라우저로부터 시스템에 대한 완전한 사용자 접근을 제공한다. 풀 클라이언트(121)는 브라우저 클라이언트(120)의 모든 기능에 덧붙여 애플리케이션 인에이블러 도구를 제공한다.

국(102)은 웹 BAS 서버(WBS)(110), 네트워크 프로세서(NP)(111) 및 필드 컨트롤러(FC)(112)를 포함한다. 국(102)은 통상 3가지 역할 중 하나 이상을 수행한다.

1. 국(102)은 시스템에 외래 장치(108)를 결합한다. WBS(110), NP(111) 및 FC(112)가 이 역할을 수행할 수 있다.
2. 국(102)은 다른 국(102)을 감독한다. 이 역할에서, 국(102)은 시스템에 감독하는 국(102)을 결합한다. 또한 감독은 모니터링 및 라우팅과 같은 요구를 포함한다. WBS(110)과 NP(111)가 이 역할을 수행할 수 있다.
3. 국(102)은 로컬 시스템의 마스터가 될 수 있다. 이 역할에서 국은 모든 구성 정보에 대한 지속적인 백업을 제공한다. WBS(110)과 FC(112)가 이 역할을 수행할 수 있다.

결합기의 역할에서 국(102)은 시스템의 나머지가 예측 가능한 동작으로 이용할 수 있도록 정보를 객체로 가져온다. 감독 역할에서 국(102)은 다른 국(102)을 감독하고, 국 사이의 시스템 무결성을 검사하고, 데이터와 이벤트 정보를 라우팅하는 것을 포함한다.

마스터로서 기능할 때, 국(102)은 시스템의 기본 마스터로서 정의된다. 기본 마스터는 모든 구성 정보의 백업을 제공한다.

외래 장치(105)는 시스템과 호환되는 장치를 포함하지만, 다운로드 가능한 자바 기반 객체를 사용하여 직접 제어되지는 않는다. 2개의 특정한 형태의 외래 장치(105)는 이러한 장치에 대한 산업계에 보급된 표준을 추종하는 BACnet™ 장치(123)와 LonMark 장치(122)이다. 원격 호스트(108)는 제어 시스템의 사이트 이외의 한 사이트에서 시스템의 몇몇 기능을 제공한다. 원격 호스트(108)는 시리얼 모뎀을 포함하는 TCP/IP를 지원하는 통신연결에 의해 잡에 연결될 수 있다.

국(102)은 도 9 및 도 10에 도시된 것처럼 통신한다. 브라우저 클라이언트(120)와 풀 클라이언트(121)는 하이퍼텍스트 전송 프로토콜(HTTP)과 자바 원격 메소드 요청(RMI)을 사용하는 제어 네트워크(106)를 통해 웹 BAS 서버(110), 네트워크 프로세서(111) 및 필드 컨트롤러(112)를 포함하는 국(102)과 통신한다. 웹 BAS 서버(110)와 필드 컨트롤러(112)는 객체지향 제어 시스템(100)의 객체의 저장 및 객체에 대한 접근을 위해 객체 데이터베이스(150)를 추가로 포함한다.

다양한 사용자를 위한 객체지향 제어 시스템(100)의 바람직한 아키텍처는 도 11에 도시되어 있다.

LonMark™ 장치(122)의 분산형 제어 시스템 관리를 허용하기 위해서 각각의 LonMark™ 장치(122)와 연관된 객체 특성에 표현되는 특정한 장치 형태에 대한 외부 인터페이스가 필요하다. 외부 인터페이스 정보는 예를 들어 특정한 장치 형태에 대해 이용 가능한 네트워크 변수를 포함할 수 있다. 과거에는 시스템에 부가된 각각의 새로운 LonMark™ 장치(122) 또는 LonMark™ 장치(122)의 외부 인터페이스에 대한 변화를 대하여 분산형 제어 시스템이 LonMark™ 장치(122)를 관리하기 위해서는 분산형 제어 시스템의 사용자가 먼저 외부 인터페이스 정보를 LonMark™ 장치(122)와 연관된 객체 특성에 프로그램 또는 입력할 필요가 있었고, 결과적으로 새로운 또는 수정된 외래 장치(105)를 온라인 및 분산형 제어 시스템에 의해 제어될 수 있도록 하는데 지연이 발생하였다. 객체지향 제어 시스템(100)은 장치의 외부 인터페이스 특성을 반영하기 위해 LonMark™ 장치(122)와 연관된 객체 특성의 동적 LonMark™ 장치 갱신을 제공함으로써 상기 제한을 극복한다. 이를 달성하기 위해 객체지향 제어 시스템(100)은 LonMark™ 장치(122)에 그들의 외부 인터페이스 특성을 조회한다. 다음으로 조회를 받는 LonMark™ 장치(122)와 연관된 유효한 객체 특성이 조회에 응답하여 수신된 정보에 따라 갱신되고 구성된다. 따라서, 이 동적 장치 외부 인터페이스 갱신 기능은 사용자에 의한 이전에 미리 프로그램된 특정한 외부 인터페이스 표현을 필요로 하지 않으면서 LonMark™ 장치(122)의 관리를 제공하고, 이에 의해 시간을 절약하고, LonMark™ 장치(122) 객체 특성을 구성하는데 있어서 오류의 가능성을 절감할 수 있다.

국(102)의 네트워크 프로세서(111)와 필드 컨트롤러(112) 형태가 NP(111) 또는 FC(112)에 부착된 시스템의 특정한 제어를 위해 자바 객체를 사용하여 프로그램될 수 있는 자바 가상 머신(JVM)을 구현하는 것이 바람직하다. 이러한 JVM의 프로그래밍은 객체 사이에서 링크를 부가, 삭제 또는 재구성함으로써 실시간으로 조정될 수 있다. 각각의 객체는 자바를 사용하여 구현되는 것이 바람직하고, 특정한 제어 기능에 대응하며, 미리 정해진 링크 세트의 사용하여 다른 객체에 연결될 수 있다. 예를 들어 가용 객체는 기능 발생기, 아날로그 및 디지털 입력 및 출력, 다상 출력, 경보 객체, 가용 국 서비스 객체 및 커스텀 사용자 프로그램화 객체를 포함한다. 이들 객체의 각각은 관련된 JVM에 연결된 장치 또는 장치들에 대한 가상 프로그램화 컨트롤러를 생성하기 위해 바람직한 방법으로 조합될 수 있다.

바람직한 실시예에서, 미리 정의된 객체는 단일 객체지향 제어 시스템(100)의 사용자 정의 객체와 공존한다. 객체 생성을 용이하게 하기 위해 제어 애플리케이션에 대해 개별화되고 간략화된 자바 언어에 기초

한 개별화된 프로그래밍 언어가 제공되는 것이 바람직하다. 개별화된 프로그래밍 언어는 객체지향 제어 시스템(100)의 프로그램 노드에 대하여 프로그램 명령을 명시하는데 사용된다. 게다가 프로그래밍 에디터가 개별화된 프로그래밍 언어를 사용하는 프로그램 노드의 사용자 프로그래밍에 대한 개발 도구로서 제공된다.

본 명세서와 여기에 첨부된 부록에 걸쳐서 용어 "노드"는 객체지향 제어 시스템(100)에 의해 제공된 특정 객체와 그들의 특성을 인용하기 위해 사용된다. 객체지향 제어 시스템(100)의 바람직한 실시예에서 "노드"는 객체라는 것을 알 수 있다. 노드의 프로세싱 및 기능적인 동작은 프로그램 가능한 연관된 특성에 의해 정해진다. 객체지향 제어 시스템(100)을 포함하는 대부분의 소프트웨어 애플리케이션은 노드의 형태로 제공된다.

객체지향 제어 시스템(100)에 의해 객체 특성 세트가 제공된다. 미리 정의된 객체 세트가 객체지향 제어 시스템(100)에 의해 제공된다.

게다가 객체지향 제어 시스템(100)은 사용자가 명령과 시스템의 소프트웨어 애플리케이션의 타이밍 및 동작을 제어할 수 있는 칼렌더, 스케줄 및 로그를 제공한다. 칼렌더는 시스템 동작과 관련하여 특별히 다루어져야 하는 날들의 세목을 허용한다. 스케줄은 프로그램 명령이 특정 시간에 실행되도록 허용한다. 로그는 객체지향 제어 시스템(100)에 걸쳐서 정보의 공유를 위해 버퍼에 시스템 정보를 수집하는 노드 및 서비스 세트이다.

로그는 정보의 지능형 아카이빙을 지원하는 데이터베이스에 지속적으로 저장된다. 특정 형태의 로그는 다음을 포함하지만, 이에 한정되지는 않는다.

1. 수치 상대값과 출력 스트링값을 수집하고 저장하는 제어 로그. 제어 로그는 다른 노드로부터 정보 출력을 수집하기 위해 제어 엔진에서 실행한다.
2. 오류 및 운영자가 행한 변경을 로그하는 서비스 로그. 서비스 로그는 국(102)에 대해 1회 발생한다.
3. 객체지향 제어 시스템(100)에 대해 시스템 외부에서 얻어진 로그 정보를 포함하는 외래 로그.

부가적으로, 로그-서비스 소프트웨어 애플리케이션은 모든 로그에 대해 하이퍼텍스트 전송 프로토콜(HTTP)을 통해 웹 기반 접근을 제공한다.

객체 또는 노드에 대한 접근은 다중 레벨 보안 프로토콜(바람직하게는 8레벨 프로토콜)을 통해 제어되는 것이 바람직하고, 객체는 다양한 레벨의 세목으로 표시를 용이하게 하는 "컨테이너"에서 그룹화될 수 있다. 즉, 객체 동작을 모니터링, 제어 또는 프로그래밍하는 사람은 즉각적인 요구에 따라 다양한 레벨의 세목으로 연결된 객체를 표시할 수 있다. 객체지향 제어 시스템(100)에 접근하고, 그에 한정되지는 않지만 로그에서 필요한 초기 암호와 객체에 대한 프로그램 가능한 보안 그룹을 포함하는 개개의 객체에 접근하여 수정할 수 있는 사람을 제어하기 위해 몇가지 기능이 제공된다.

특히, 보안 모델은 유효한 암호를 갖는 시스템에 로그인하는 사용자를 필요로 한다. 각각의 사용자 객체는 8개 보안 그룹의 8개 접근 레벨(보안 승인)에 대하여 승인이 할당되거나 거절될 수 있다. 이러한 보안 그룹은 서로 독립적이며, 그들이 의미하는 것은 국부적인 문제이다. 각각의 객체는 또한 이들 8개 보안 그룹의 조합에 할당될 수 있다. 이 방식에서 다른 많은 정의들이 동시에 지정될 수 있다.

데이터베이스에 정의된 각각의 사용자 객체는 시스템에 로그인하기 위해 요구되는 사용자명과 암호를 갖는다. 이러한 사용자 객체의 각각은 8개 보안 그룹의 보안 승인에 대하여 승인이 주어지거나 거절될 수 있다.

시스템에 대하여 정의된 8개 보안 그룹이 존재한다. 이들은 서로 독립적이고, 이들이 의미하는 것은 국부적인 문제이다. 각각의 객체는 특성 보안 그룹을 통해 이들 8개 보안 그룹의 조합에 할당될 수 있다. 기본값으로 각각의 객체는 보안 그룹 0(일반)에 할당된다. 이 방식에서 다른 많은 보안 승인 동시에 지정될 수 있다. 기본 보안 그룹은 다음과 같다.

0 = 일반

1 = hvac

2 = 보안

3 = 라이프 세이프티

4 = 그룹4

5 = 그룹5

6 = 그룹6

7 = 그룹7

보안 승인은 각각의 사용자에게 주어진 기본 권한이다. 이들은 제한 없이 운영자 판독, 운영자 기록, 관리자 판독, 관리자 기록, 표준 명령, 인식 경보, 긴급 명령 및 관리자 명령을 포함한다.

사용자의 객체에 대한 접근 권한은 객체의 보안 그룹 특성에 표시되거나 검사되는 각각의 그룹에 대하여 자신의 권한을 조합함으로써 정해진다. 부가적으로, 사용자가 컨테이너에 대한 권한을 가지면, 몇몇 권한은 그의 자(子)에게 적용된다. 사용자가 컨테이너에 대한 운영자 판독 권한을 가지면, 그 사용자는 작업 공간 관점에서 자를 볼 수 있다. 사용자가 컨테이너에 대한 관리자 기록 승인을 가지면, 그 사용자는 연결, 절단, 복사, 사복, 삭제 및 개명을 포함하는 컨테이너에 대한 많은 기능을 수행할 수 있다.

객체 보안은 사용자가 그 각각의 특성의 운영자 또는 관리자 특성으로의 지정과 객체에 할당하는 보안 그룹의 조합에 의해 정해진다. 사용자는 그 시스템에 지정된 8개 보안 그룹의 어떤 조합에도 어느 한 객체

를 할당할 수 있다. 이것은 특성 보안 그룹을 통해 행해진다. 기본적으로 각각의 객체는 보안 그룹 0 (일반)에 할당된다.

시스템의 각각의 특성은 운영자 또는 관리자 특성의 어느 하나로 지정되었다. 이것은 사용자가 객체의 특성을 검토하거나 편집하기 위해 어떤 보안 승인을 가져야 하는지를 결정한다.

운영자 보안 승인은 객체의 최하위 레벨 기능만을 검토하거나 편집할 필요가 있는 시스템 운영자로서 필요한 기능으로 정의된다.

관리자 보안 승인은 시스템을 구성하기 위해 필요한 기능으로 지정된다. 이 사람은 시스템의 모든 기능을 검토하거나 편집할 필요가 있다. 관리자 편집은 객체를 생성하고, 특성을 부가하는데 덧붙여 객체를 연결하는 것을 포함한다.

객체지향 제어 시스템(100)은 사용자 객체를 통해 새로운 사용자를 지정하기 위해 사용자 서비스를 제공한다. 사용자 서비스는 각각의 사용자에 대해 사용자 보안 그룹과 승인이 시행되는 것을 보장한다. 새로운 사용자 객체는 기본 사용자명과 암호를 포함하는 새로운 사용자에 대해 형성된다. 사용자는 자신의 암호를 변경할 수 있다. 사용자 또는 객체들은 하나 이상의 보안 그룹에 할당된다. 각각의 사용자 보안 승인은 변경될 수 있다.

객체는 제한 없이 다음 연결, 정상, 트리거, LonTalkLocal, LonTalkNetwork, UI, 복합 또는 외부 중 하나를 사용하여 동적으로 서로 연결된다.

또한, 객체지향 제어 시스템(100)은 객체 또는 노드에 대해 소프트웨어 애플리케이션 프로그램에 의해 수행된 동작들을 중재하기 위해 BACnet™ 명령 우선주의를 사용한다. BACnet™ 명령 우선주의는 특정한 객체 또는 노드를 명령할 수 있는 애플리케이션 프로그램에 서로 다른 우선순위 레벨을 할당하고, 해당 노드에서 그 노드에 대한 우선주의를 노드의 객체 특성으로 저장함으로써 달성된다. 각각의 노드 또는 객체는 그의 명령 우선주의 레벨에 따라 소프트웨어 애플리케이션 프로그램으로부터 수신된 명령을 따른다.

클라이언트(101), 서버(110) 및 네트워크 프로세서(111)는 마이크로소프트 윈도우 95 또는 윈도우 NT 운영체제를 실행하는 인텔 펜티엄™ 기반 개인용 컴퓨터와 같은 개인용 컴퓨터를 사용하여 구현되는 것이 바람직하다. 필드 컨트롤러(112)는 자바 가상 머신으로 동작할 수 있는 제한된 성능, 저가의 단일 보드 컴퓨터를 사용하여 구현될 수 있다. 예를 들어 소위 네트워크 컴퓨터가 이런 목적으로 사용될 수 있다.

본 발명의 아키텍처는 다른 많은 시스템 구성을 지원한다. 보다 일반적인 구성의 몇몇은 필드 컨트롤러 시스템(도 2), 소형 웹 BAS 서버 시스템(도 3), 2레벨 시스템(도 4), 3레벨 시스템(도 5) 및 분산형 시스템(도 6)을 포함한다.

도 2를 참조하면, 필드 컨트롤러 시스템은 클라이언트가 복수의 LonMark 장치(122)에 연결된 LonMark 필드버스(107)를 감독하는 단일 필드 컨트롤러(112)에 접근하는 본 발명의 구성이다.

도 3은 클라이언트(브라우저 클라이언트(120), 풀 클라이언트(121) 또는 양쪽 모두)가 복수의 LonMark 장치(122)에 연결된 LonMark 필드버스(107)를 감독하는 단일 웹 BAS 서버(110)에 접근하는 소형 웹 BAS 서버 시스템을 보여준다. 도 4에 도시된 바와 같이 2레벨 시스템은 네트워크 프로세서(111), 필드 컨트롤러(112) 및 BACnet 장치(123)의 임의의 조합을 직접 감독하는 웹 BAS 서버(110)로 구성된 시스템이다. 또한 웹 BAS 서버(110)가 직접 필드 컨트롤러(112)를 감독하는 네트워크 프로세서(111)를 감독하는 3레벨 시스템이 제공될 수 있다. 마지막으로 도 6은 원격 호스트(108)를 추가로 포함하는 분산형 시스템으로 구성되고, 대형 또는 소형 시스템의 집합이 모뎀(109)을 사용하여 원격 호스트(108)를 통해 연결되고 감독되는 객체지향 제어 시스템(100)의 대체 실시예를 보여준다.

이 시스템은 사무실, 상업, 건강관리, 교육 및 산업 설비를 성공적으로 운영하고 관리하는데 필요한 많은 서비스들을 자동화하는 시스템을 구성, 전개 및 실행하는 아키텍처를 포함한다.

시스템의 데이터 흐름은 분산형 객체지향 제어 시스템(100)의 목적을 달성하는데 핵심이 되는 각 국(102)에서의 지속적이고 실시간의 정보 동기화 관리자(PRISM)(200)에 관리되는 것이 바람직하다.

PRISM(200)은 도 7에 기술된 바와 같이 아카이빙 및 동기화 기술의 사용을 통해 객체지향 제어 시스템(100) 전체에 걸쳐서 시스템 데이터의 무결성을 유지한다. 객체지향 제어 시스템(100)과 연관된 데이터는 시스템 데이터, 자바 애플리케이션 및 제어 엔진(JACE), 노드 및 특성으로 카테고리화된다. 게다가, 객체지향 제어 시스템(100)의 각각의 데이터는 또한 그의 라이프 사이클에 따라 카테고리화된다. 임시 데이터는 호스트 가상 머신(VM)의 존속 기간 동안 존재한다. 지속적인 데이터는 호스트 VM의 존속 기간을 넘어서 존재하고, 전원 장애 또는 차단 동안에도 계속 남아있다. 객체지향 제어 시스템(100)에 존재하는 가상 머신은 풀 클라이언트(121)와 연관된 풀 사용자 인터페이스(206), 브라우저 클라이언트(120)와 연관된 브라우저 사용자 인터페이스(208), Taz(210) 및 하나 이상의 JACE(212)를 포함하지만, 이에 한정되지는 않는다. 지속적인 데이터는 비휘발성 기억장치에 그것을 저장함으로써 아카이빙된다. 현재의 바람직한 실시예에서, FLASH(204)와 자기 디스크(202) 비휘발성 기억장치 매체(즉, "Pstore")가 아카이빙된 지속적인 데이터를 저장하는데 사용된다. JACE(212)는 자신들과 연관된 임시 및 지속적인 데이터의 마스터 카피를 유지한다. 동기화 및 비동기화된 캐시는 객체가 현재의 시스템 데이터에 접근했음을 보장하기 위해 PRISM(200)의 제어 하에서 사용된다. 각각의 가상 머신은 해당 가상 머신에 의해 요구된 모든 데이터를 포함하는 적어도 하나의 캐시를 갖는다. 캐시는 가상 머신의 캐시가 그들의 감독 국(102)의 캐시와 일치되거나 이 캐시를 갱신하는데 사용되는 일치 방법으로 최신으로 유지된다. 다음에 국(102)은 서로 자신들의 캐시와 일치한다. 현재의 바람직한 실시예에서 현재의 데이터에서 불일치를 결정하기 위해 체크섬이 사용된다. 각각의 감독 국(102)은 감독 국에서 저장된 노드에 대한 현재의 가상 머신 캐시 데이터를 얻고, 다음에 조회를 받는 가상 머신으로부터 수신된 캐시 데이터와 캐시 내용에 대한 체크섬을 계산한다. 만일 체크섬이 일치하지 않으면, 불일치를 검출되고, 이 경우에 감독 국(102)은 그에 한정되지는 않지만 보다 새로운 데이터를 반영하기 위해 캐시를 갱신하고, 보다 새로운 데이터를 선택하기 위해 타임스탬프 비교를 수행하거나 또는 경보 또는 오류 표시를 올리는 것을 포함하는 문제를 해결하기 위한 단계를 밟는다.



현재의 최선의 실시예에서 단일 웹 BAS 서버(110) 대신에 객체지향 제어 시스템(100)은 복수의 웹 BAS 서버(110)를 포함하고, 복수의 웹 BAS 서버(110)의 각각은 자체적으로 지속적인 데이터의 데이터베이스를 갖는다. 현재의 최선의 실시예에서, 객체는 데이터가 아카이빙되는 특정한 웹 BAS 서버(110)의 물리적인 위치와 관계 없이 지속적인 데이터에 접근한다. 국(102)은 다른 국(102)이 또한 로컬 웹 BAS 서버(110)에 연결되는지에 관계 없이 다른 국(102)과 통신할 수 있다.

이전에 지적한 바와 같이, 모든 형태의 설비에서 객체에 대한 광범위한 요구를 지원하기 위해 본 발명의 바람직한 실시예는 소프트웨어 모듈이 제어 시스템의 어느 곳에서든지 쉽게 설치되고 사용되는 것을 허용하는 자바 모듈 환경을 제공한다. 이 방식에서 객체는 요구되는 곳에 설치되어 최선의 실시간 성능을 제공할 수 있다.

이 시스템은 어떤 지원 플랫폼으로 인도될 수 있도록 플랫폼 독립 방식으로 시스템을 구현하기 위해 자바 클래스 파일로 구성되는 것이 바람직하다. 본 발명에 따른 코어 클래스 계층은 이러한 유연성이 가능한 노드 및 플랫폼을 갖는 시스템을 제공한다. 계층은 서로 다른 플랫폼에 대하여 서로 다른 인터페이스 및 프로토콜에 대한 드라이버를 구성하기 위한 유연한 방식으로 지원된다. 코어 클래스 계층에 부가하여, 사용자 객체는 이들이 계층적으로 동작을 이어받고, 예측 가능하게 수행하도록 계층으로 구성된다. 바람직한 실시예에서 달성된 계층은 도 8에 도시되어 있다.

현재의 바람직한 실시예에서 객체지향 제어 시스템(100)의 사용자는 그에 한정되지는 않지만 제어, 애플리케이션, 사용자 인터페이스, 컨테이너, LonMark™, BACnet, 관리자 및 서비스와 같은 객체 카테고리를 포함하는 복수의 객체 카테고리과 연관된 객체를 형성하고 조작할 수 있다.

게다가 제어 객체 카테고리는 추가로 아날로그 입력, 아날로그 출력, 바이너리 입력, 바이너리 출력, 비교, 논리, 루프, 계산, 함수 발생기 및 계산기와 같은 서브카테고리를 포함한다.

게다가 애플리케이션 객체 카테고리는 추가로 스케줄, 칼렌더, 프로그램, 아날로그 로그, 바이너리 로그 및 정수 로그와 같은 서브카테고리를 포함한다.

게다가 사용자 인터페이스 객체 카테고리는 추가로 막대 그래프, 불린 이미지, 바운드 이미지, 덤퍼, 팬, 핫 스폿 이미지, 스펙트럼, 텍스트 및 타이머 플롯과 같은 서브카테고리를 포함한다.

게다가 컨테이너 객체 카테고리는 추가로 복합 및 총칭 서브카테고리를 포함한다.

게다가 LonMark™ 객체 카테고리는 추가로 Snvt 스위치 Mux, Snvt 스위치 Demux, DemoFcu, Leviton 스위치, Leviton Sw 481, Action Instr A0, Action Instr A1, Action Instr D0 및 로컬 Lon 노드와 같은 서브카테고리를 포함한다.

게다가 BACnet 객체 카테고리는 추가로 장치, 아날로그 입력, 아날로그 출력, 바이너리 입력 및 바이너리 출력과 같은 서브카테고리를 포함한다.

게다가 서비스 객체 카테고리는 추가로 제어 엔진, UI 엔진, 사용자 접근, 국간 연결, HTTPD 서비스, 웹 텍스트, 웹 뷰, 미디어, 프로그램 디버그, 다이얼업, Lonworks, Lon 통신, 네트워크 변수 폴 스케줄러, 네트워크 변수 관리자, BACnet 서버, BACnet 클라이언트, BACnet 폴, BACnet 트랜스포트, BACnet 네트워크, BACnet 이더넷, 오류 아카이브, 오류 포워드 및 메일과 같은 서브카테고리를 포함한다.

게다가 관리자 객체 카테고리는 추가로 사용자 서브카테고리를 포함한다.

객체지향 제어 시스템(100)에 의해 제공된 이러한 객체들의 목적과 사용은 그에 한정되지는 않지만 주로 이들 객체 카테고리 및 서브카테고리의 각각과 연관되는 것으로 이해된 것을 포함한다.

실시간 환경에서 수용 가능하게 수행하기 위해 객체에 대한 제어 엔진이 예측 가능하게 수행해야 하지만, 객체의 실행 순서에 대한 제어를 사용자에게 제공해야 한다. 본 발명은 이러한 기능을 제공한다. 현재의 바람직한 실시예에서 객체 특성은 실행 순서 및 실행 주기를 지정하기 위해 제공된다. 현재의 최선의 실시예에서 5개의 서로 다른 실행 주기가 지정될 수 있다. 각각의 실행 주기는 100ms의 배수로 연관된 객체 특성을 편집함으로써 조정될 수 있다. 현재의 최선의 실시예에서 3개의 서로 다른 실행 순서가 지정될 수 있다.

원격 사이트에 사이트의 성능에 대한 정보를 제공하기 위하여 시스템은 그와 같이 하여 프로그램될 때마다 실시간 정보를 원격 사이트에 이메일을 전송하기 위해 프로그램 가능 이메일 서비스를 제공한다.

시스템은 객체 특성을 모두 표준 데이터 형태와 같은 접근 가능한 요소로서 제공한다. 이것은 사용자 및 프로그램 객체가 어떤 특성을 효율적으로 사용하도록 허용한다. 다음에 이들 특성은 다른 도구에 의한 조작을 위해 XML(eXtended Markup Language)로서 노출된다.

이 시스템은 사용자가 몇몇 애플리케이션에 대한 커스텀 객체를 형성하도록 허용하는 수단을 필요로 한다. 이러한 요구에 대응하기 위해 사용자 접근 및 제어를 위해 프로그램 객체의 시스템을 노출하는 스크립트 언어가 정의된다. 이 언어는 간단하고, 표준 객체 연결을 사용하고, 라이브러리에 대한 접근을 가지며, 완전한 디버거를 제공한다.

본 발명은 또한 새로운 방식으로 제어 시스템에 인터넷의 복합 문서 패러다임을 적용한다. 이것은 보통의 URL(Universal Resource Locator)를 통해 리소스의 범용 가시성을 제공한다. 이러한 리소스는 내비게이션이 리소스의 하이퍼링크된 웹을 형성하는 것을 허용하기 위해 서로 링크된다.

리소스는 접근성 및 유효성을 보장하기 위해 시스템에 의해 관리된다. 시스템의 리소스는 클래스 파일, 실행 가능한 DLL, 미디어, 서식, 특성 파일, 데이터베이스와 로그 및 제 3의 소프트웨어를 포함한다.

제어 시스템은 설비 관리자, 애플리케이션 엔지니어 및 정보 전문가를 포함하는 많은 사람이 사용 가능해야 한다. 일반적으로 시스템은 객체 및 정보를 접근하기 위한 개별화된 브라우저 인터페이스를 포함하는 새롭고 독특한 제어 인터페이스를 제공한다. 이 인터페이스는 작업 공간 디스플레이, 작업 공간 에디터,

특성 디스플레이, 링크 디스플레이 및 도움말 기능을 포함한다. 사용자는 어떤 객체 또는 시스템에 명령을 발송할 수 있고, 상기 제어 인터페이스를 통해 동작 특징 및 특성을 변경할 수 있다.

일반적인 워크스테이션에 정보 관리 및 실시간 전문가 제어 시스템을 조립함으로써 본 발명의 BAS 아키텍처는 간단화되고, 고객은 정보 및 제어 모두에 대한 단일 포인트의 접근을 가질 수 있다. 웹 BAS 서버 플랫폼은 모든 설비 관리 정보 및 빌딩 제어 애플리케이션에 대하여 자바 가능 웹 브라우저에 기초한 일반적인 그래픽 사용자 인터페이스를 제공한다. 부가적으로 이러한 개방형 단일 소프트웨어 환경은 어떤 산업계 표준 브라우저를 사용하여 인터넷을 통하여 정보/필드버스 네트워크 및 제어 시스템 애플리케이션에 대한 일정한 구성 도구를 제공한다.

여기에 개시된 바와 같이 웹 BAS 서버 플랫폼은 자바 가능 웹브라우저 인터페이스를 통해 지원된 일반적인 객체 모델 애플리케이션 환경에서 서로 다른 BACnet, LonMark 및 인터넷 / CORBA를 결합하기 위한 BAS 산업계의 제 1 소프트웨어 기술이다. 이 실시예는 또한 BACnet / LonMark / 인터넷 / CORBA 시스템 네트워크의 입안, 설계, 구성, 설치 및 유지에서 BAS 필드 계약자 채널을 지원하기 위한 종합 네트워크 관리 도구이다.

기존화 가능한 웹 BAS 서버 소프트웨어가 BACnet / LonMark 필드버스에 직접 부착된 단일 사용자 프론트엔드 플랫폼으로 소형 빌딩에 사용될 수 있다. 대향 BAS 설비에 대하여 웹 BAS 서버는 인터넷을 통해 연결된 많은 네트워크 프로세서를 감독하고, 제한되지 않는 원격 웹브라우저 운영자 인터페이스 사용자를 지원하도록 구성될 수 있다. 기업 고객에 대하여 웹 BAS 서버는 IBM 기업 정보 기반구조 및/또는 그루페쉬나이어 개인 / 전원 시스템 아키텍처에 BAS 아키텍처를 결합하도록 구성될 수 있다.

본 발명의 한 가지 중요한 특징은 BAS 컨트롤러 모듈(네트워크 프로세서)의 개발이다. 종래의 우선주의 BAS 컨트롤러 플랫폼은 복수의 빌딩 서비스 애플리케이션을 지원하기에는 너무 작은 메모리, 속도 및 처리능력을 갖는다. 따라서 사용자의 실질적인 문제를 해결 가능한 독립적인 제품으로 다시 분류하지만 복잡한 솔루션을 갖지 않는 개별적인 컨트롤러를 HVAC 및 접근 제어와 같은 각각의 애플리케이션이 제공된다. 이에 따라 호환되지 않는 자립형 빌딩 서비스 솔루션 사이의 결합이 매우 적고, 설치 비용이 증가한다.

이러한 제한은 본 발명에서 BAS 컨트롤러 프로세서 기술과 상호 동작하는 BACnet / LonMark 필드버스를 연결함으로써 극복된다. 이 새로운 고성능 기술을 사용하기 위해 본 발명은 이용 가능한 프로세싱 능력을 이용하는 애플리케이션 개발 기능을 사용한다. 따라서 결합과 복수의 빌딩 서비스 애플리케이션은 동일한 컴퓨팅 플랫폼에 의해 수행될 수 있다.

오늘날의 BAS 컨트롤러의 제한된 애플리케이션 개발 기능을 극복하기 위해 본 발명의 네트워크 프로세서는 객체지향 분석, 설계 및 프로그래밍에 기초한 자바 애플리케이션 환경을 사용한다. NP는 복수의 고수준 빌딩 서비스 애플리케이션을 결합하고, 멀티-벤더 필드버스 솔루션을 지원하기 위해 선(Sun)의 자바소프트 기술에 기초한 강력한 자바 객체/순서 엔진을 사용한다. 네트워크 프로세서 애플리케이션의 예는 대중적인 빌딩 서비스 제품을 결합하기 위해 글로벌 경보, 글로벌 스케줄링, 트랜딩/데이터 수집, 진단 서비스 및 요구 제한/전원 관리를 포함한다.

네트워크 프로세서(NP) 객체 소프트웨어 환경은 내장 컨트롤러 하드웨어 플랫폼에 제공되는 것이 바람직하다. NP 플랫폼 패키지는 멀티-벤더 LonMark 필드버스를 지원하고, 다른 NP 및 웹 BAS 서버에 대한 인터넷 동등 계층(peer-to-peer) 네트워킹을 제공한다(도 1 BACnet / LonMark / 인터넷 아키텍처 참조).

본 발명은 도면 및 부록에 도시된 바와 같이 필드 컨트롤러(FC) 플랫폼의 3가지 실시예를 제공한다. FC 플랫폼은 32비트 프로세서를 사용하는 것이 바람직하고, 에셀론(Echelon) 보조 프로세서를 통해 상호 동작 가능한 BASnet / LonMark 필드버스에 선택적으로 연결할 수 있다. 이러한 플랫폼은 애플리케이션에 대한 자바 객체/순서 엔진을 지원하는 실시간 운영체계를 사용한다. FC는 필드버스 위에 또는 FC의 로컬 RS 232 / 모뎀 포트 또는 비디오 모뎀 인터페이스를 통과하여 웹 BAS 서버에 의해 구성된다.

제 1 FC 패키지는 그루페쉬나이어에 의해 제조된 국부적으로 부착된 단일 블럭 입력/출력 모듈을 지원한다. 이 집적된 설계 방식은 최종적인 개인 및 상업용 컨트롤러 애플리케이션에 대해 매우 유연하며, 비용이 효율적이다. BACnet / LonMark 객체/순서 엔진은 그의 강력한 애플리케이션 환경에서 어떤 로컬 입력/출력 데이터 또는 어떤 필드버스 장치 데이터도 사용할 수 있다.

필드버스에 부착된 제 2의 32비트 필드 컨트롤러 플랫폼은 국부적으로 부착된 입력/출력을 지원하지 않지만, 집적된 저비용의 프로그램 가능 운영자 인터페이스 및 BACnet / LonMark 객체 엔진을 포함한다. 웹 BAS 서버 구성 도구는 필드버스 네트워크상의 어떤 노드 장치에서 제어 및 데이터 접근을 갖는 유연한 운영자 인터페이스를 프로그램할 수 있다.

필드 컨트롤러 패키지의 제 3 실시예는 웹 BAS 서버에 의한 모니터링과 원격 제어를 위해 소형 빌딩에 대한 저가의 인터넷 접속을 제공하기 위해 비디오 케이블 모뎀과 함께 결합된다.

시스템의 동작 요소들은 결합 및 용법을 간편화하기 위해 한 객체로서 각각 모형화된다. 이것은 사용자 프리젠테이션, 네트워크 관리, 네트워크 제어, 실시간 제어 및 장치 객체를 포함한다.

본 발명의 다른 특징으로서, 비교적 고가의 풀타임 인터넷 접속 또는 서버에서 클라이언트의 장거리 전화를 방지하기 위해 서버는 클라이언트가 다시 서버에 접속을 개방하는 간단한 다이얼업 통보 메커니즘을 사용한다. 동작에 있어서, 서버는 클라이언트에 부착된 모뎀에 전화를 걸고, 클라이언트가 응답하기 전에 호출을 끊는다. 클라이언트는 이것은 서버 요청 접속으로 인식한다. 호출에 응답하여 클라이언트는 저가의 월 서비스 요금을 갖는 로컬 ISP에 전화를 건다. 다음에 클라이언트는 인터넷을 통해 서버에 접속을 달성한다. 이 메커니즘은 ISP에 의해 부과된 월 요금으로 접근 요금을 제한함으로써 호출 또는 분당 요금을 방지한다.

따라서, 복수의 산업계 표준을 지원하는 멀티-벤더 개방형 시스템 환경에서 분산형 애플리케이션의 일반 객체 모델에 BACnet, LonMark, CORBA, 자바 및 인터넷 시스템을 결합하기 위해 객체지향 소프트웨어를 사



용하는 제어 시스템 및 방법이 도시되었다. 본 발명의 제어 시스템 및 방법은 단일 제어 시스템에서 미리 정의된 객체와 사용자 정의 객체를 모두 제공한다. 시스템 데이터의 집적 및 지속은 아카이빙 및 동기화 기술을 통해 유지되고 분산된다. 장치와 연관된 특성은 장치의 외부 인터페이스 특성을 반영하기 위해 동적으로 갱신된다.

본 출원은 1999년 5월 14일 출원된 Gerald Frank 등의 미국 특허 출원 "컴퓨터 네트워크를 사용하는 이종(異種) 전기기계 시스템의 객체지향 제어를 위한 시스템 및 방법"을 참조로서 포함한다.

### (57) 청구의 범위

#### 청구항 1

전자 네트워크를 통해 객체지향 제어 시스템에 대한 접근 및 제어를 제공하고, 이 시스템을 모니터링하는 클라이언트;

다운로드 가능한 자바 기반 객체를 사용하여 직접적으로 제어할 수 없는 외래 장치;

공통 객체 모델을 사용하여 객체지향 제어 시스템에 상기 외래 장치를 결합하고, 다른 국을 감독하며, 모든 구성 정보의 지속적인 백업을 제공하는 로컬 시스템의 마스터로서 기능하는 국;

상기 클라이언트, 외래 장치 및 국의 복수의 이종(異種) 구성을 지원할 수 있는 플렉시블한 개방형 아키텍처;

복수의 작업의 규칙적이고 적시의 실행을 조정하기 위한 실시간 동기화 관리자;

복수의 사용자 정의 객체;

복수의 객체 특성;

복수의 객체 카테고리;

상기 복수의 객체 카테고리에 따라 상기 객체 및 상기 특성의 사용자 생성 및 수정을 허용하기 위한 객체 생성 기구;

다중 레벨 보안 기구;

상기 사용자 정의 객체와 공존하는 미리 정의된 복수의 객체;

아카이빙(archiving)과 동기화를 통해 객체지향 제어 시스템 전체에 걸쳐서 시스템 데이터의 무결성을 유지하는 지속적인 실시간 정보 동기화 관리자;

객체의 코어 클래스 계층;

상기 객체의 실행에 대한 예측 가능한 순서를 제공하기 위한 제어 엔진; 및

원격 사이트에 실시간 데이터를 포함하는 이메일 통보를 전송하기 위한 프로그램 가능 메일 서비스를 포함하는 것을 특징으로 하는 객체지향 제어 시스템.

#### 청구항 2

제 1 항에 있어서,

상기 객체 생성 기구는 상기 외래 장치와 연관된 외부 인터페이스 객체 특성을 동적으로 갱신하거나 부가하기 위한 동적 장치 외부 인터페이스 갱신 기능을 추가로 포함하는 것을 특징으로 하는 객체지향 제어 시스템.

#### 청구항 3

제 1 항에 있어서,

객체지향 제어 시스템의 다른 구성요소와 물리적으로 함께 위치하지 않는 위치로부터 객체지향 제어 시스템에 대한 접근, 제어 및 모니터링을 허용하기 위한 원격 호스트를 추가로 포함하는 것을 특징으로 하는 객체지향 제어 시스템.

#### 청구항 4

전자 네트워크를 사용하는 객체지향 제어 시스템의 객체를 사용자가 제어하고 모니터링하는 것을 허용하는 접근수단;

복수의 외래 장치가 공통 객체 모델을 사용하는 객체지향 제어 네트워크에 의해 제어되고 모니터링되도록 하는 결합수단;

로컬 시스템의 마스터로서 기능하는 기능을 포함하는 한 국이 다른 국을 제어하고 모니터링하는 것을 허용하는 감독수단;

복수의 작업의 규칙적이고 적시의 실행을 조정하는 실시간 동기화 수단;

복수의 객체 카테고리에 따라 객체의 사용자 생성 및 수정을 허용하는 새로운 사용자 객체 생성 수단;

한 객체와 연관된 특정 정보에 대한 접근 또는 수정을 금지하는 다중 레벨 보안수단;

객체의 실행의 예측 가능한 순서를 제공하기 위한 제어 엔진 수단; 및

원격 사이트에 실시간 데이터를 포함하는 이메일 통보를 전송하기 위한 프로그램 가능 메일 수단을 포함

하는 것을 특징으로 하는 객체지향 제어 시스템.

#### 청구항 5

전자 네트워크를 통해 객체지향 제어 시스템에 대한 접근, 제어 및 모니터링을 제공하는 단계; 및

시스템 정보를 나타내고, 시스템 정보가 객체지향 제어 시스템의 전체에 걸쳐서 이용 가능하게 구성하기 위해 공통 객체 모델을 사용하는 객체지향 제어 시스템에 외래 장치를 결합하는 단계를 포함하는 것을 특징으로 하는 객체지향 제어 시스템을 사용하는 방법.

#### 청구항 6

제 5 항에 있어서,

객체지향 제어 시스템의 객체를 생성하거나 수정하기 위해 커스텀 프로그래밍 언어를 사용하는 단계를 추가로 포함하는 것을 특징으로 하는 방법.

#### 청구항 7

제 5 항에 있어서,

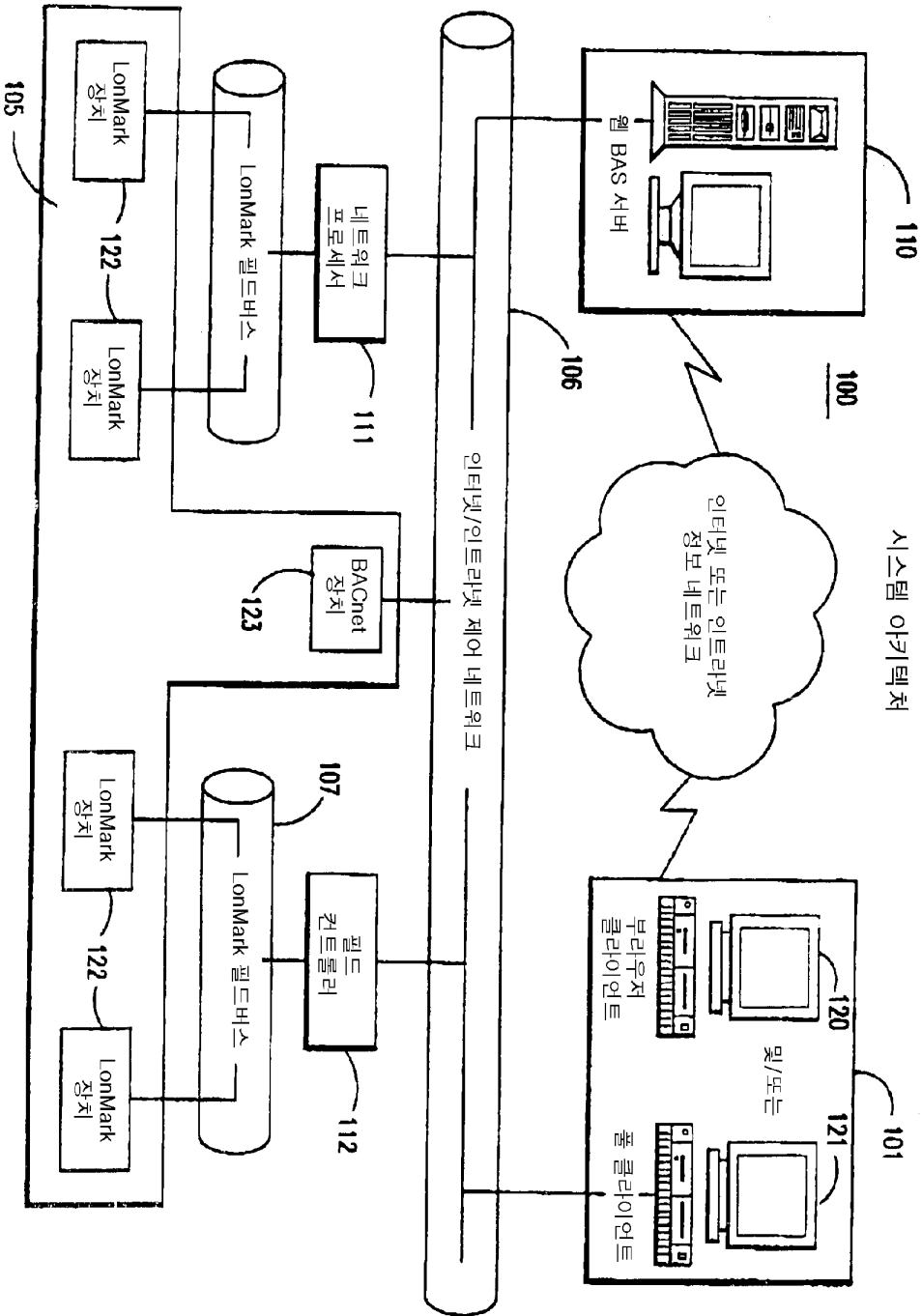
다양한 레벨의 세목으로 표시를 용이하게 하기 위해 컨테이너내에 객체를 그룹화하는 단계;

코어 클래스 계층에 따라 객체를 구성하는 단계; 및

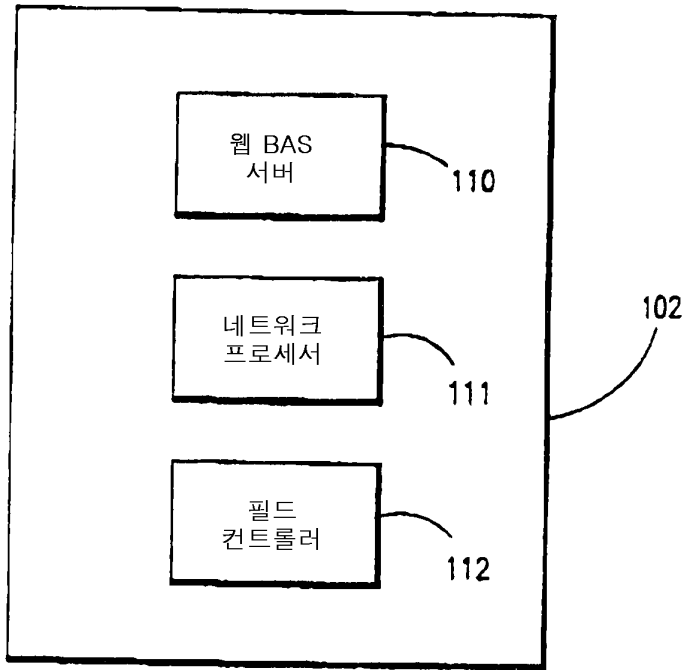
다중 레벨 보안 프로토콜의 사용을 통해 객체에 대한 접근을 제어하는 단계를 추가로 포함하는 것을 특징으로 하는 방법.

**도면**

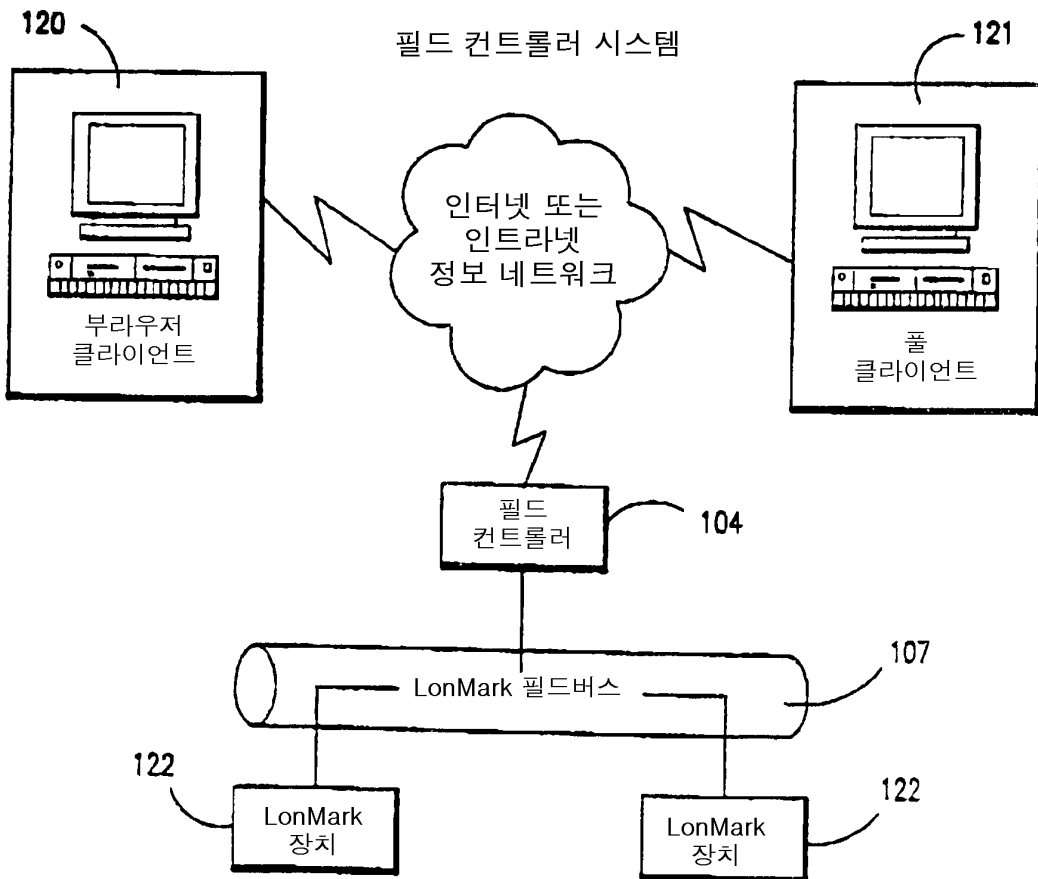
도면 1



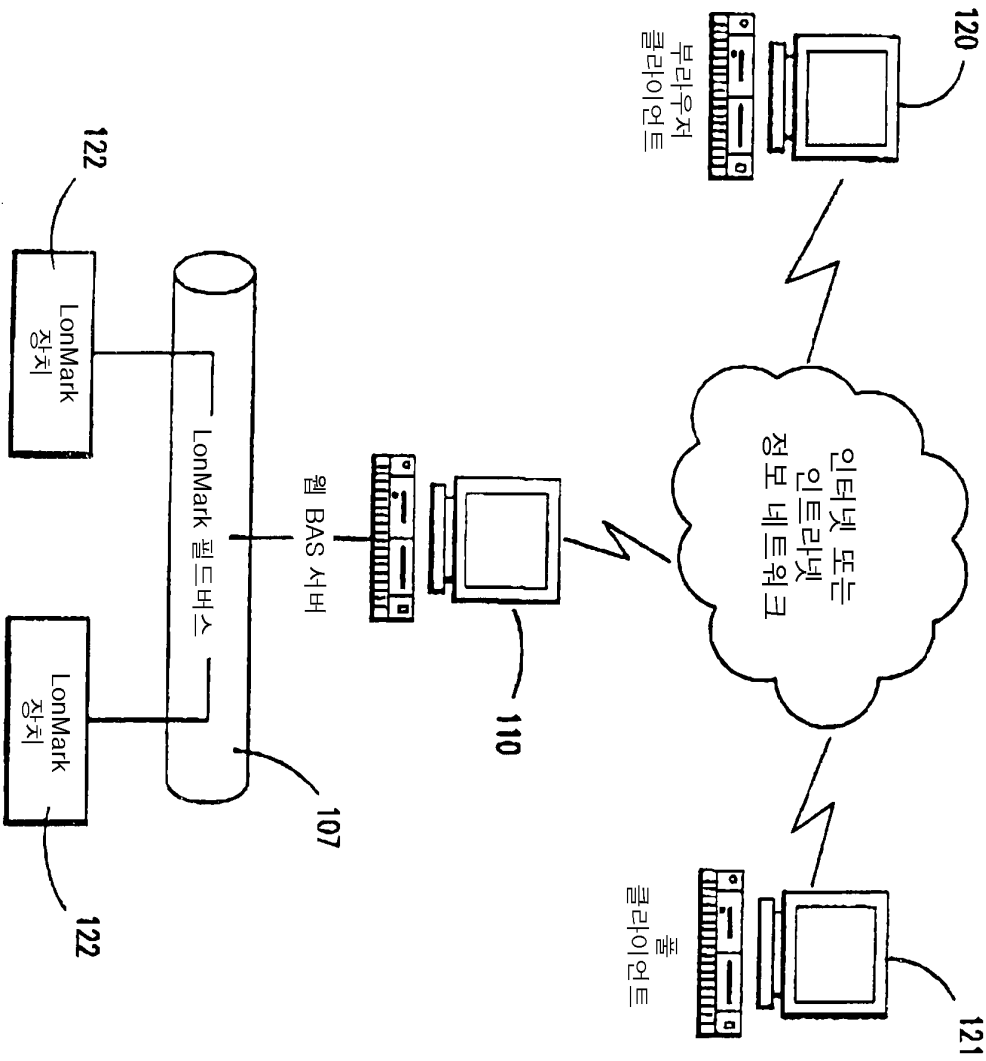
도면 1A



도면 2

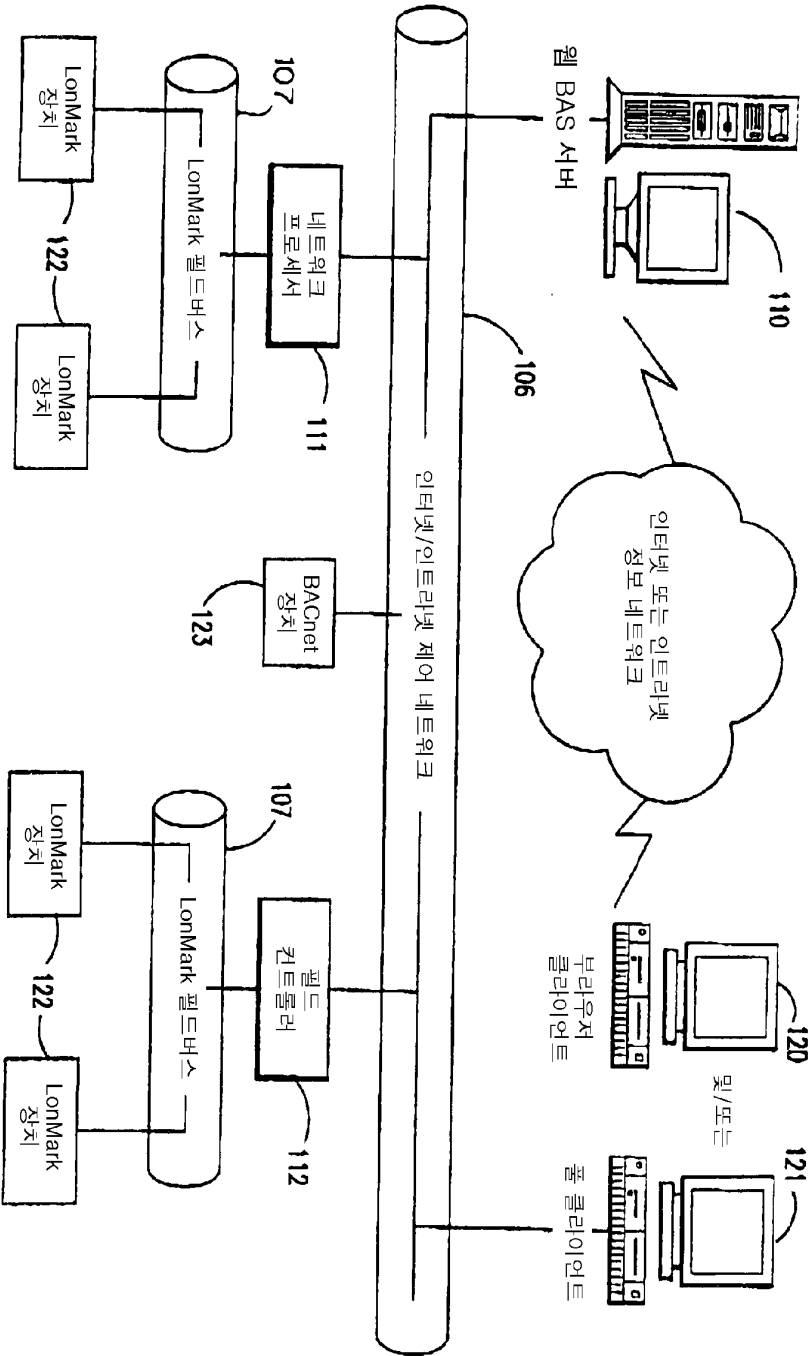


소형 웹 BAS 서버 시스템



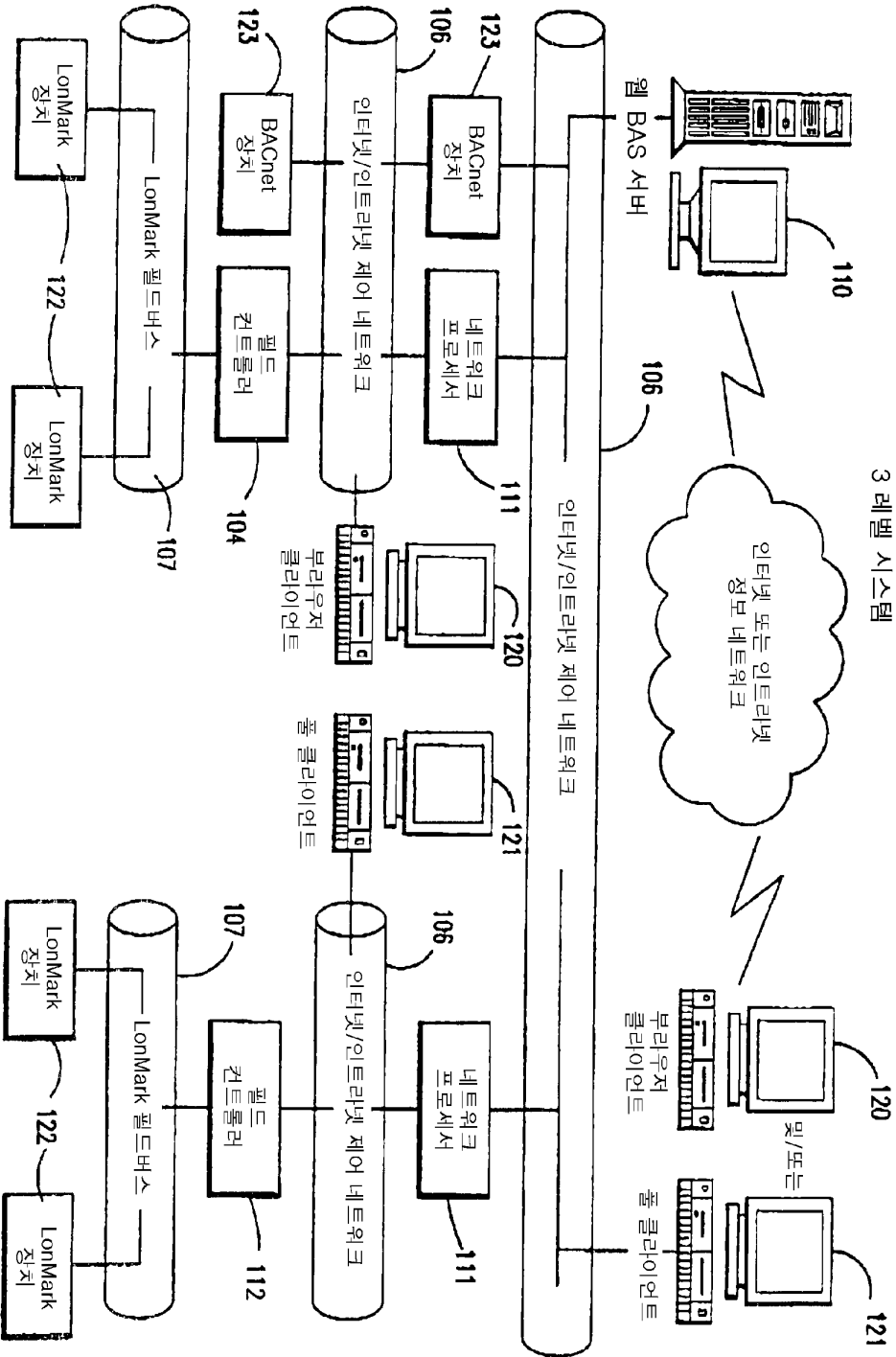
도면3

2 레벨 시스템



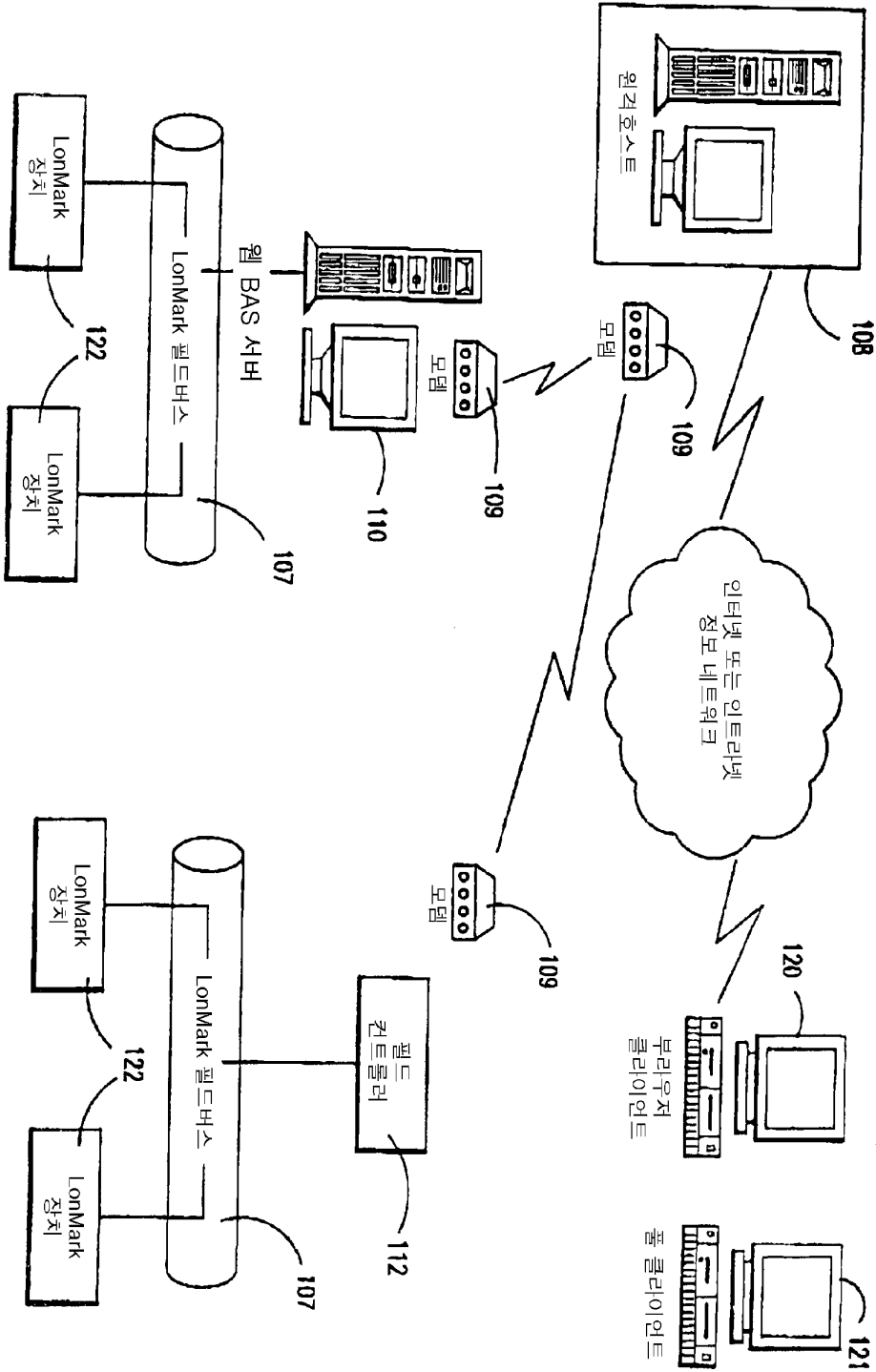
도면4

도면5



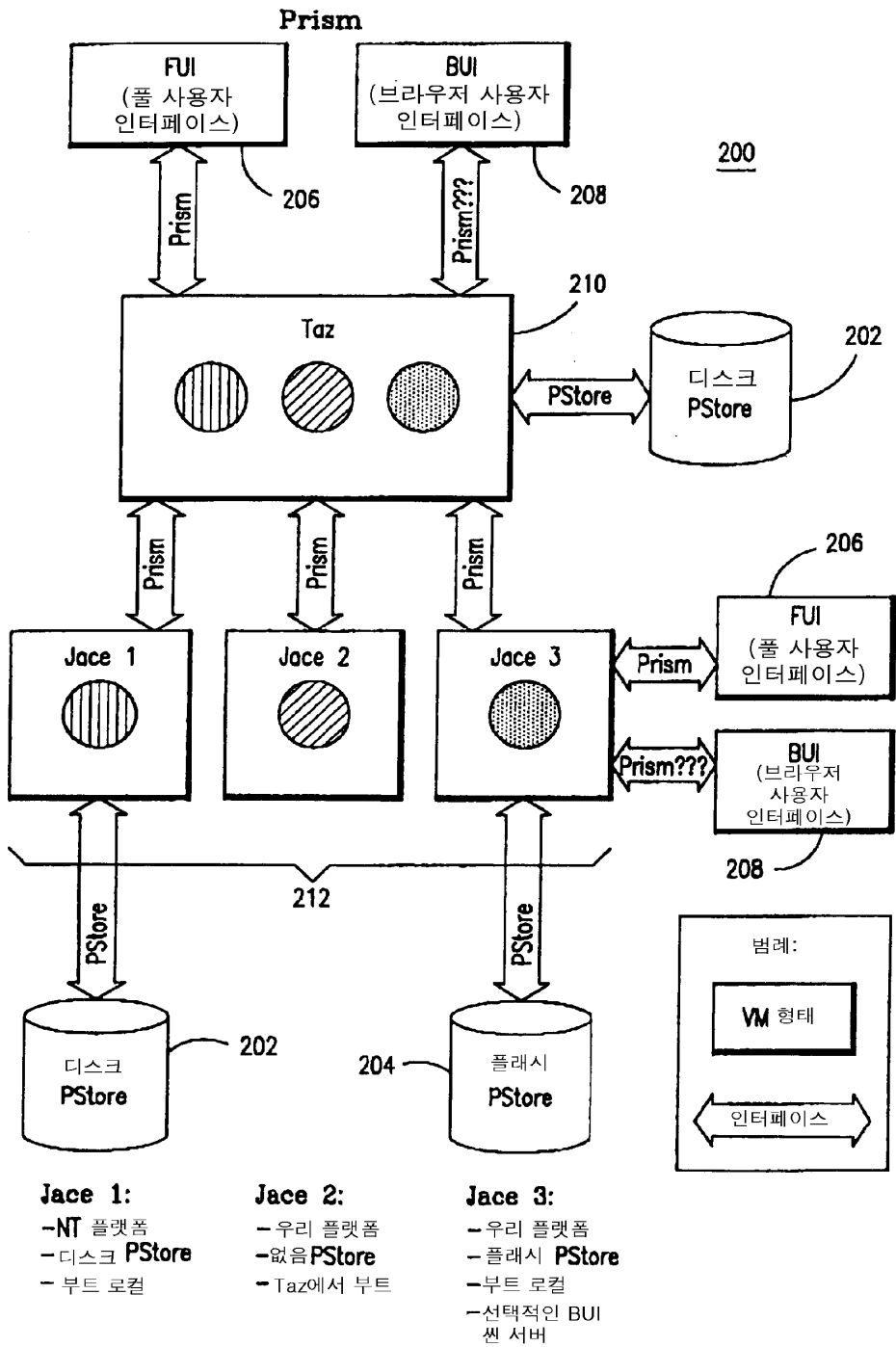


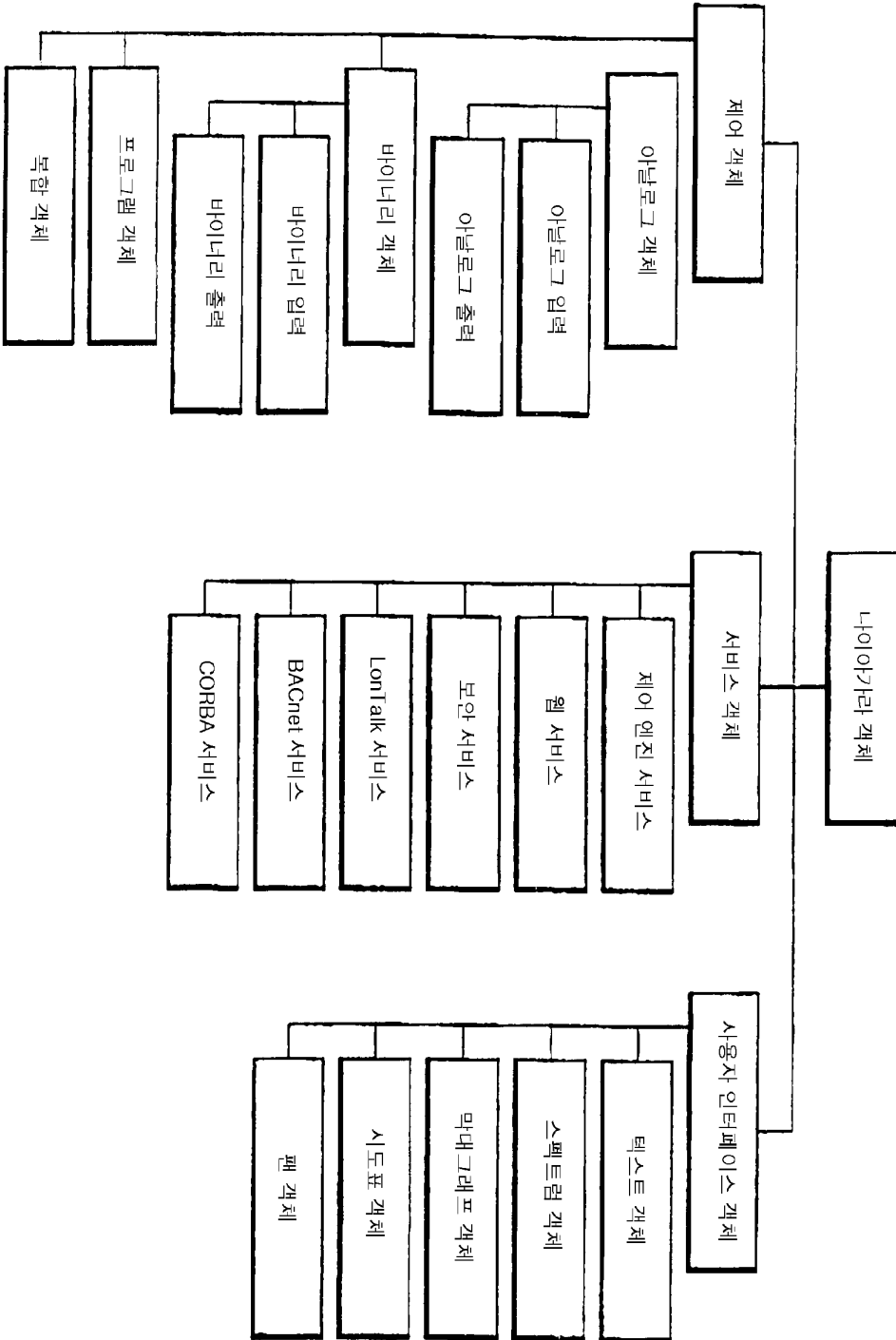
분산형 시스템 아키텍처



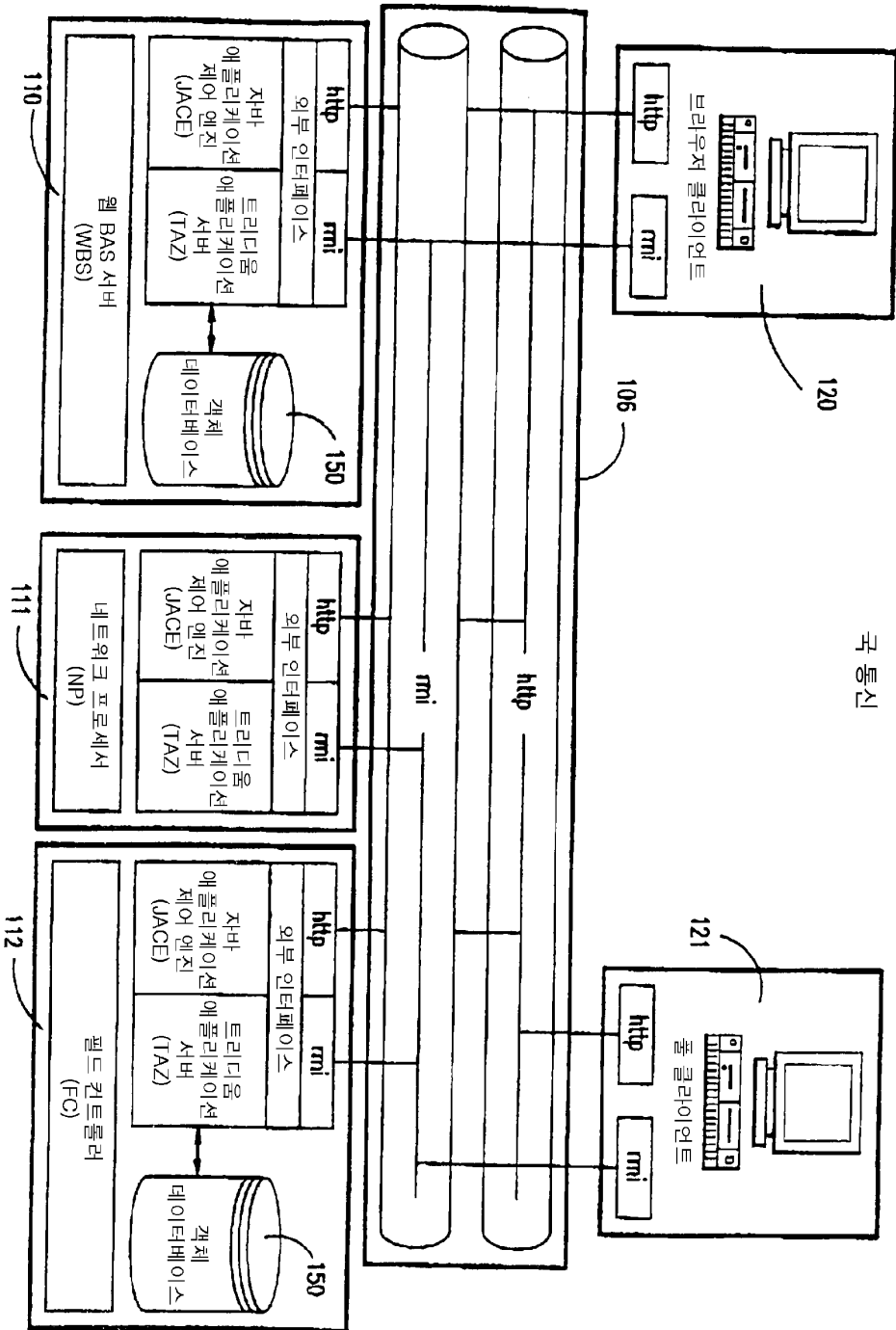
도면6

도면7

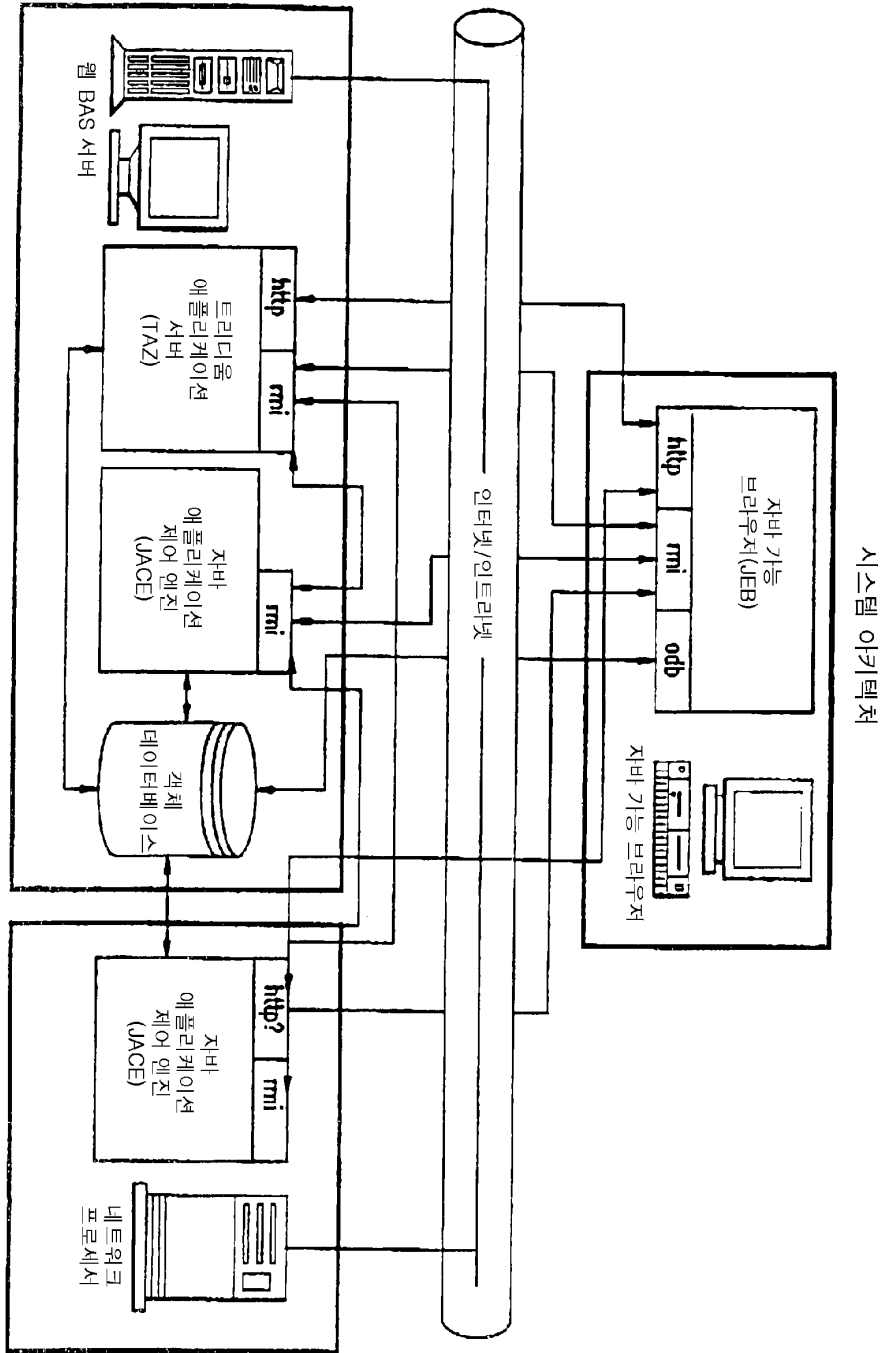




도면 6



도면 10



도면11

