

(19) 대한민국특허청(KR)

(12) 공개특허공보(A)

(51) Int. Cl.⁶
H04L 12/407

(11) 공개번호 특1999-0051702
(43) 공개일자 1999년07월05일

(21) 출원번호	10-1997-0071070
(22) 출원일자	1997년 12월 19일
(71) 출원인	한국전자통신연구원 정선중
(72) 발명자	대전광역시 유성구 가정동 161번지 유인원 대전광역시 대덕구 비래동 136-1 금성백조 103-1005 이근영 대전광역시 유성구 어은동 99 한빛아파트 117-204 김동진 대전광역시 유성구 가정동 236-1 남궁한 대전광역시 유성구 어은동 99 한빛아파트 128-303 신영무, 최승민
(74) 대리인	신영무, 최승민

심사청구 : 있음

(54) 분산객체 시스템에서 원격객체 생성 및 접근방법

요약

본 발명은 분산 객체 환경에서 클라이언트와 동일한 지역에 있는 객체를 생성하고 접근하는 방법과 동일한 방법으로 스텐브(Stub)를 생성하고 스텐브에 접근하면 스텐브에서 객체 중개자(CORBA: Common Object Request Broker Architecture)를 통하여 원격 시스템의 스켈톤(skeleton)과 연결하고 스켈톤에서 목적 객체를 생성하고 접근한다. 그러므로 본 발명은 클라이언트에서 지역 객체를 이용하는 것과 동일한 방법과 효과를 그대로 유지하면서 스텐브를 이용하여 원격 시스템의 객체를 생성하고 접근하도록 하였다.

대표도

도5

명세서

도면의 간단한 설명

도 1은 이 발명을 구체화하기 위하여 사용되는 범용 컴퓨터인 자료 처리 시스템의 블록 도표.
도 2는 도 1의 메인 메모리(main memory)에서 클라이언트가 목적 객체에 접근하는 상태도.
도 3은 도 1의 각기 다른 시스템들이 통신 라인(Communication Line)으로 연결되고 도 2의 클라이언트와 객체가 다른 시스템에서 동작하는 경우에 클라이언트에서 목적 객체에 접근하는 상태도.
도 4는 도 3에 도시한 클라이언트에서 목적 객체에 접근하기 위하여 필요한 개체들을 열거하고 그들 간에 연결 관계를 나타낸 구조도.
도 5는 도 4에 도시한 클라이언트에서 목적 객체에 접근하는 경로를 나타낸도면.
도 6은 도 5의 경로를 통하여 클라이언트에서 목적 객체를 생성하는 과정을 나타낸 도표.
도 7은 도 5의 경로를 통하여 클라이언트에서 목적 객체의 메소드에 접근하는 과정을 나타낸 도표.
도 8은 도 6의 클라이언트에서 목적 객체를 생성하는 과정을 차례로 기술한 흐름도.
도 9는 도 7의 클라이언트에서 목적 객체의 메소드에 접근하는 과정을 차례로 기술한 흐름도.

<도면의 주요부분에 대한 부호의 설명>

11: 생성자	12: 메소드
13: 클라이언트	14: 객체

- | | |
|------------|--------------|
| 15: 스칼톤 | 18: 스텐브 |
| 20: CORBA | 21: 매소드 이름 |
| 22: 매개 변수값 | 23: 매개 변수 타입 |
| 26: 서버 요구 | |

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 분산 객체 시스템에서 원격 객체 생성 및 접근 방법에 관한 것으로서, 특히 분산 객체 환경에서 클라이언트 시스템의 클라이언트와 서버 시스템의 객체간에 통신하는 방법으로 분산 처리 기술 분야, 객체 기술 분야, 클라이언트/서버 기술 분야 및 통신 기술 분야에 속하는 분산 객체 시스템에서 원격 객체 생성 및 접근 방법에 관한 것이다.

종래의 기술은 서버 시스템에서 목적 객체를 먼저 생성하여 대기하였다가 클라이언트에서 목적 객체의 메소드를 호출하면 수행하였다. 이 방법의 단점은 목적 객체가 클라이언트와 관계없이 먼저 생성되므로 목적 객체의 생성에 필요한 매개변수 값을 전달받지 못한다. 또한 기존의 방법으로는 이름은 동일하나 매개 변수가 다른 여러 메소드를 갖는 목적 객체를 클라이언트에서 호출하지 못하는 문제점이 있었다.

발명이 이루고자하는 기술적 과제

객체는 자신의 고유한 성질이나 모습을 표현하는 속성과 요구에 의하여 반응하는 동작을 기술하는 행위로 구성된 식별 가능한 실체이다. 행위는 객체 자신을 생성하는 생성자와 생성된 객체가 동작하는 메소드로 구분한다.

한 프로그램 영역에서 객체는 다른 객체를 생성하고 객체 메소드를 호출하여 동작하도록 한다. 새로운 객체의 생성은 그 객체의 생성자를 호출하면서 매개변수에 값을 전달하므로 이루어진다. 객체의 동작은 객체의 메소드를 호출하면서 매개변수에 값을 전달하므로 행동한다. 객체는 객체의 이름과 동일한 이름을 갖는 생성자와 객체의 이름과 다른 이름을 갖는 메소드를 가질 수 있으며, 생성자 및 메소드는 동일한 이름이라도 매개변수의 형태가 다르면 중복된 이름의 생성자와 메소드도 가질 수 있다.

본 발명은 한 시스템에서 다른 시스템에 있는 목적 객체의 생성자를 선택하여 생성하고 목적 객체에 이름이 동일한 메소드가 있으면 매개변수 형태를 구분하여 호출하므로 원격 목적 객체에 접근하는 방법이 지역 객체에 접근하는 방법을 그대로 유지하도록 하였다.

본 발명에서 분산 객체 환경을 지원하기 위하여 사용되는 객체 중개자(CORBA: Common Object Request Broker Architecture)는 객체 관리 그룹(OMG: Object Management Group, Inc.)에서 상업적으로 이용 가능한 제품에 근거하여 객체 기술을 이용한 분산 환경의 단일 구조 상에서 소프트웨어 구성 요소의 재사용, 신속한 개발, 용이한 사용, 상호 운용 및 이식을 위하여 목적 관리 구조(OMA: Object Management Architecture) 참조 모델을 정의한 것이다. 객체 중개자는 분산 처리 시스템의 종류나 그 시스템의 설치 위치에 관계없이 분산 응용 객체 간에 통신하도록 허용한다. 객체 중개자는 객체간에 통신을 위한 객체 요구 중개자(ORB: Object Request Broker)를 기반으로 객체 집단이 연결되며 클라이언트의 요구 사항을 응용 객체에 전달하는 인터페이스를 표준 규격으로 정의한 것이다.

발명의 구성 및 작용

이하 첨부된 도면을 참조하여 본 발명을 설명하면 다음과 같다.

도 1은 본 발명을 구체화하기 위하여 사용되는 범용 컴퓨터인 자료 처리 시스템의 블록 도표로서, 본 발명은 각기 다른 운영체제를 갖춘 다양한 범용 컴퓨터 시스템에서 동작될 수 있다. 자료 처리 시스템은 본체에 CPU(2)와 메인 메모리(Main Memory: 3)를 포함하며 시스템 버스(System Bus: 4)로 연결된 확장 슬롯(Expansion Slot: 10)을 통하여 그래픽 어댑터(Graphic Adapter: 5), 통신 보드(Communication Board: 6), 디스크 컨트롤러(Disk Controller: 9)등과 연결된다. 입력 장치인 키보드 마우스 등은 CPU(2)에 연결되며, 출력 장치인 그래픽 디스플레이(1)는 그래픽 어댑터(5)에 연결된다. 디스크(disk: 8)는 디스크 컨트롤러(Disk Controller: 9)에 연결되고, 다른 시스템과 통신을 위한 통신 라인(Communication Line: 7)은 통신 보드(6)에 연결된다. 이 발명에서 구현한 코드의 일부는 메인 메모리(3)에 적재되며 나머지는 다른 시스템의 메인 메모리(3)에 적재되어 서로 통신 라인을 통하여 정보를 교환하며 작업을 수행한다.

도 2는 도 1의 한 시스템에 있는 메인 메모리(3)에서 객체간에 동작하는 상태를 자세하게 표시한 그림이다. 클라이언트(13)에서 지역 객체(14)의 생성자(11)와 메소드(12)에 접근하는 것을 나타낸다.

도 3은 본 발명을 설명하기 위하여 도 2의 클라이언트(13)가 클라이언트 시스템의 메인 메모리(3)에서 동작하고 객체(14)는 서버 시스템의 메인 메모리(3)에서 동작하고 있으며, 클라이언트(13)에서 목적 객체(14)의 생성자(11)와 메소드(12)에 접근하는 가상의 그림이다.

도 4는 도 3의 클라이언트와 객체가 실제로 연결하기 위하여 필요한 하부 구조의 준비과정을 표시한 그림이다. IDL 화일(16)은 클라이언트(13)에서 객체(14)에 접근하기 위한 접속 형식을 정의한다. IDL 컴파일러(compiler: 17)는 IDL 화일(16)로 클라이언트(13)와 CORBA(20)를 연결하는 스텐브(stub: 18) 및

CORBA(20)와 객체(14)를 연결하는 스켈톤(Skeleton: 15)을 생성한다. CORBA(20)는 서로 다른 시스템에 있는 스텐브(18)와 스켈톤(15)간에 전달할 정보를 통신 라인을 통하여 전달한다.

도 5는 목적 객체에 접근하는 경로를 도시한다. 클라이언트(13)에서 스텐브(18)에 접근하면 스텐브(18)에서 클라이언트 시스템의 객체 중개자(CORBA: 20)에 접근하며 클라이언트 시스템의 객체 중개자(20)는 서버 시스템의 객체 중개자(20)와 연결되고 서버 시스템의 객체 중개자(20)는 스켈톤(15)에 접근하며 스켈톤(15)은 객체(14)에 접근하는 경로를 나타낸다.

도 6은 목적 객체를 생성하는 과정을 도시한다. 클라이언트(13)에서 매개 변수를 지정하여 스텐브(18)의 생성자(11)를 호출한다. 스텐브 생성자(11)는 객체 중개자(20)와 연결하고 목적 객체의 생성자 이름(21), 매개변수 값(22) 및 매개 변수 타입(23)으로 요구(request)를 만들어 객체 중개자(20)에게 전달한다. 클라이언트 시스템의 객체 중개자(20)는 서버 시스템의 객체 중개자(20)와 연결하고 요구를 서버 시스템의 객체 중개자(20)에게 전달한다. 서버 시스템의 객체 중개자(20)는 스켈톤(15)을 생성하고 스켈톤(15)의 인 보우크 메소드를 호출하여 서버 요구(Server Request: 26)를 전달한다. 스켈톤(15)의 인 보우크 메소드에서는 서버 요구의 생성자 이름(21)과 매개변수 타입(23)을 확인하고 전달된 매개변수 값(22)을 지정하여 목적 객체 생성자(11)를 호출한다. 목적 객체 생성자는 객체(14)를 초기화하고 클라이언트(13)에서 호출된 역순으로 복귀한다.

도 7은 목적 객체의 메소드에 접근하는 과정을 도시한다. 클라이언트(13)에서 매개 변수를 지정하여 스텐브(18)의 메소드(12)를 호출한다. 스텐브 메소드(12)는 목적 객체의 메소드 이름(21), 매개변수 값(22) 및 매개 변수 타입(23)으로 요구를 만들어 객체 중개자(20)에게 전달한다. 클라이언트 시스템의 객체 중개자(20)는 요구를 서버 시스템의 객체 중개자(20)에게 전달한다. 서버 시스템의 객체 중개자(20)는 스켈톤(15)의 인보우크 메소드를 호출하여 서버 요구(26)를 전달한다. 스켈톤(15)의 인보우크 메소드에서는 서버 요구(26)의 객체 메소드 이름(21)과 매개변수 타입(23)을 확인하고 전달된 매개변수 값(22)을 지정하여 목적 객체 메소드(12)를 호출한다. 객체(14)의 호출된 메소드(12)를 수행하고 클라이언트(13)에서 호출된 역순으로 복귀한다.

도 8은 목적 객체를 생성하는 흐름도이다. 클라이언트에서 서버 시스템에 생성할 목적 객체를 결정한다. 클라이언트에서 목적 객체와 연결할 스텐브 생성자를 호출하여 스텐브를 생성하고 매개변수 값을 전달한다. 스텐브 생성자에서 객체 중개자를 호출하여 서버 시스템을 찾고 목적 객체의 생성자 이름과 매개변수 값 및 매개변수 타입으로 요구를 만들어 객체 중개자에게 전달한다. 클라이언트 시스템의 객체 중개자는 서버 시스템의 객체 중개자에게 요구를 전달한다. 서버 시스템의 객체 중개자는 목적 객체와 연결할 스켈톤 생성자를 호출하여 스켈톤을 생성하고 인보우크 메소드를 호출하여 서버 요구를 전달한다. 스켈톤의 인보우크 메소드에서 서버 요구에 있는 목적 객체의 생성자 이름과 매개변수 타입을 확인하고 목적 객체 생성자를 호출하여 목적 객체를 생성하고 서버 요구에 있는 매개변수 값을 전달한다. 목적 객체 생성자에서 매개변수 값으로 객체를 초기화하고 호출된 역순으로 클라이언트에 복귀한다.

도 9는 목적 객체 메소드를 호출하는 흐름도이다. 클라이언트에서 스텐브 메소드를 호출하여 매개변수 값을 전달한다. 호출된 스텐브 메소드에서 목적 객체의 메소드 이름과 매개변수 값 및 매개변수 타입으로 요구를 만들어 객체 중개자에게 전달한다. 클라이언트 시스템의 객체 중개자는 서버 시스템의 객체 중개자에게 요구를 전달한다. 서버 시스템의 객체 중개자는 스켈톤의 인보우크 메소드를 호출하여 서버 요구를 전달한다. 스켈톤의 인보우크 메소드에서 서버 요구에 있는 목적 객체의 메소드 이름과 매개변수 타입을 확인하고 목적 객체의 메소드를 호출하여 서버 요구에 있는 매개변수 값을 전달한다. 호출된 목적 객체의 메소드를 수행하고 클라이언트로 복귀한다.

발명의 효과

본 발명은 접근할 객체가 지역에 있거나 원격에 있거나 동일한 방법으로 접근하도록 하였다. 본 발명으로 인하여 한 시스템의 클라이언트에서 다른 시스템의 원격 객체를 생성하고 접근함은 물론, 원격 객체에서 동일한 이름을 갖는 생성자나 동일한 이름을 갖는 메소드를 클라이언트에서 접근하도록 함으로 지역 객체처럼 사용하도록 하였다.

(57) 청구의 범위

청구항 1

클라이언트에서 서버 시스템에 생성할 목적 객체를 결정하는 단계와,

상기 클라이언트에서 목적 객체와 연결할 스텐브 생성자를 호출하여 스텐브를 생성하고 매개변수 값을 전달하는 단계와,

상기 스텐브 생성자에서 객체 중개자를 호출하여 서버 시스템을 찾고 목적 객체의 생성자 이름과 매개변수 값 및 매개변수 타입으로 요구를 만들어 객체 중개자에게 전달하는 단계와,

상기 클라이언트 시스템의 목적요구 연결자는 서버 시스템의 객체 중개자에게 요구를 전달하는 단계와,

상기 서버 시스템의 객체 중개자는 목적 객체와 연결할 스켈톤 생성자를 호출하여 스켈톤을 생성하고 인보우크 메소드를 호출하여 서버 요구를 전달하는 단계와,

상기 스켈톤의 인보우크 메소드에서 서버 요구에 있는 목적 객체의 생성자 이름과 매개변수 타입을 확인하고 목적 요구 연결자를 호출하여 목적 객체를 생성하고 서버 요구에 있는 매개변수 값을 전달하는 단계와,

상기 목적 객체 생성자에서 매개변수 값으로 객체를 초기화하고 호출된 역순으로 클라이언트에 복귀하는 단계를 포함하여 이루어진 것을 특징으로 하는 분산 객체 시스템에서 원격 객체 생성방법.

청구항 2

클라이언트에서 스템 메소드를 호출하여 매개변수 값을 전달하는 단계와,

상기 호출된 스템 메소드에서 목적 객체의 메소드 이름과 매개변수 값 및 매개변수 타입으로 요구를 만들어 객체 중개자에게 전달하는 단계와,

상기 클라이언트 시스템의 객체 중개자는 서버 시스템의 객체 중개자에게 요구를 전달하는 단계와,

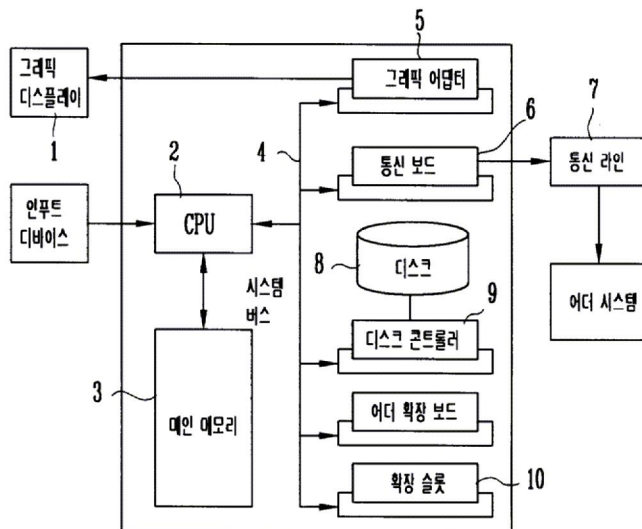
상기 서버 시스템의 객체 중개자는 스칼라의 인보우 메소드를 호출하여 서버 요구를 전달하는 단계와,

상기 스칼라의 인보우 메소드에서 서버 요구에 있는 목적 객체의 메소드 이름과 매개변수 타입을 확인하고 목적 객체의 메소드를 호출하여 서버 요구에 있는 매개변수 값을 전달하는 단계와,

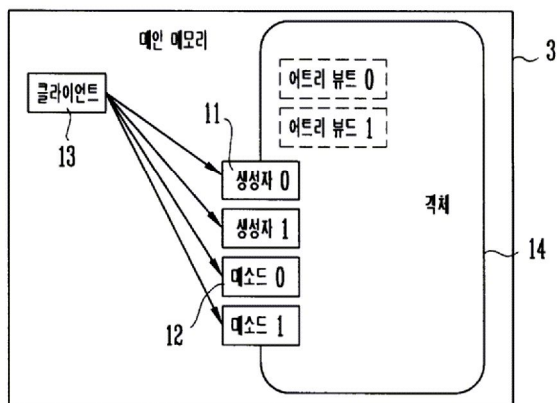
상기 호출된 목적 객체의 메소드를 수행하고 클라이언트로 복귀하는 단계를 포함하여 이루어진 분산 객체 시스템에서 원격 객체 접근방법.

도면

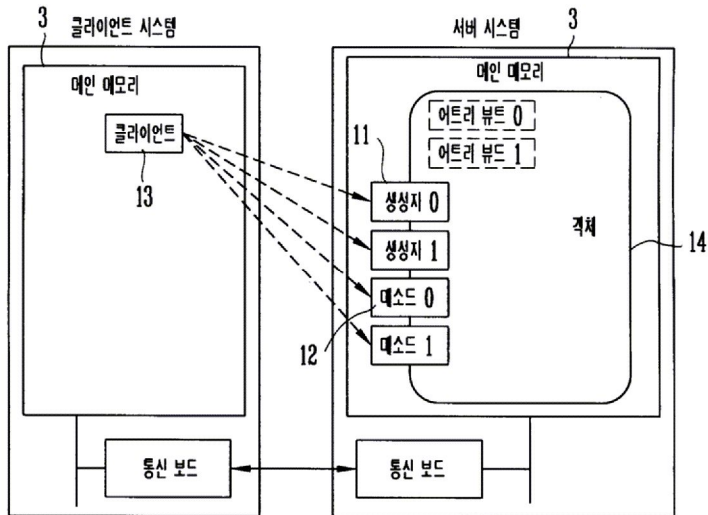
도면1



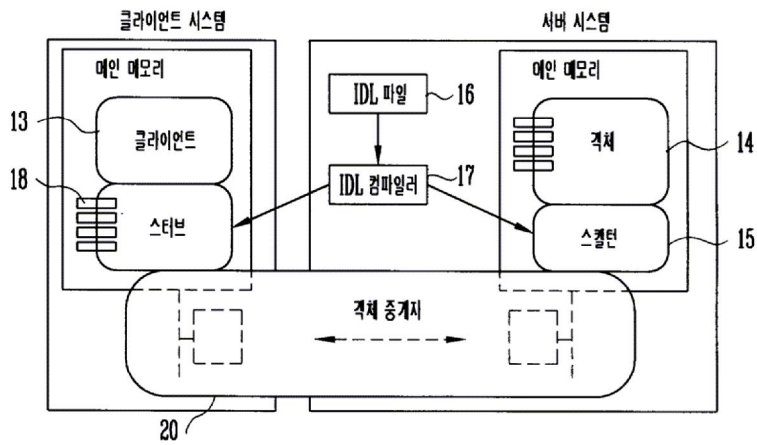
도면2



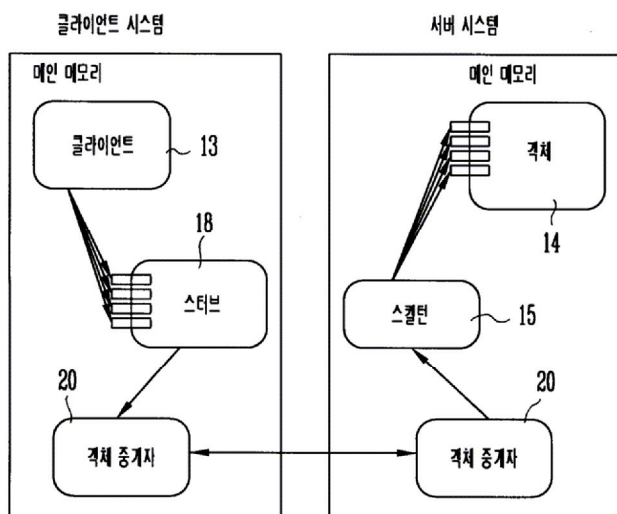
도면3



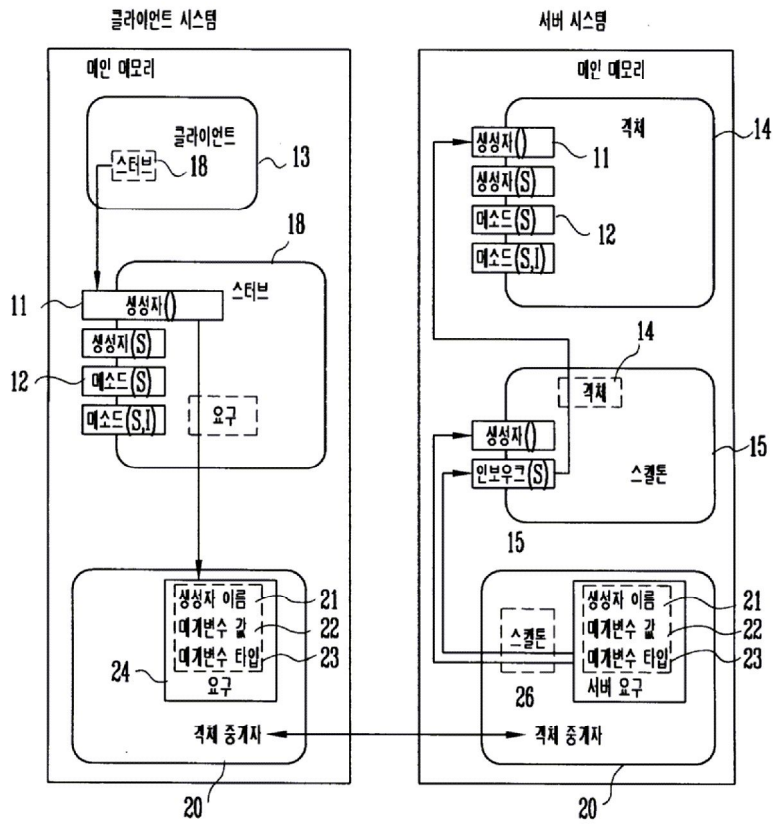
도면4



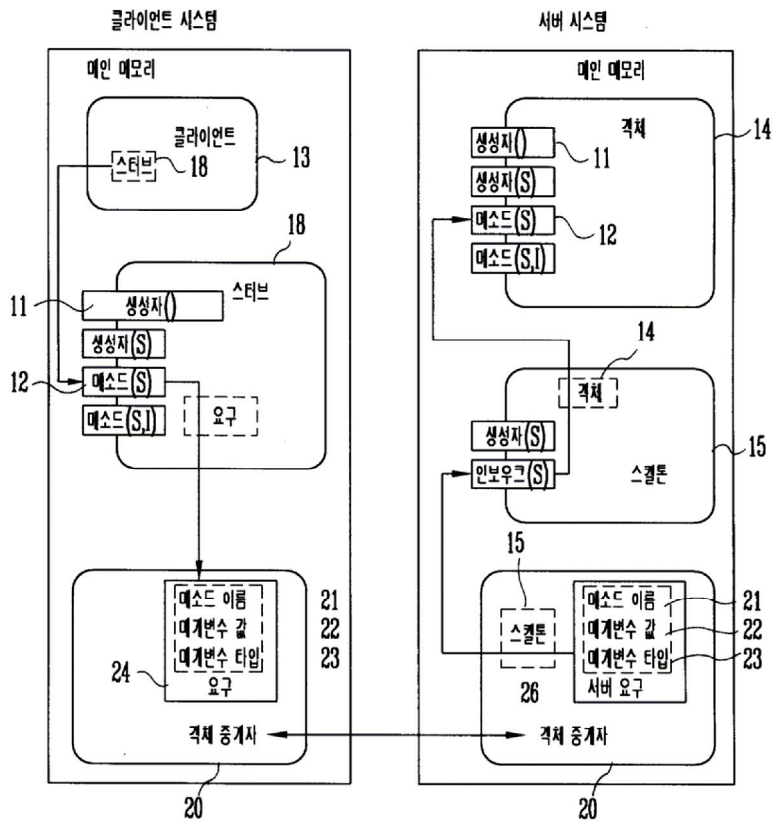
도면5



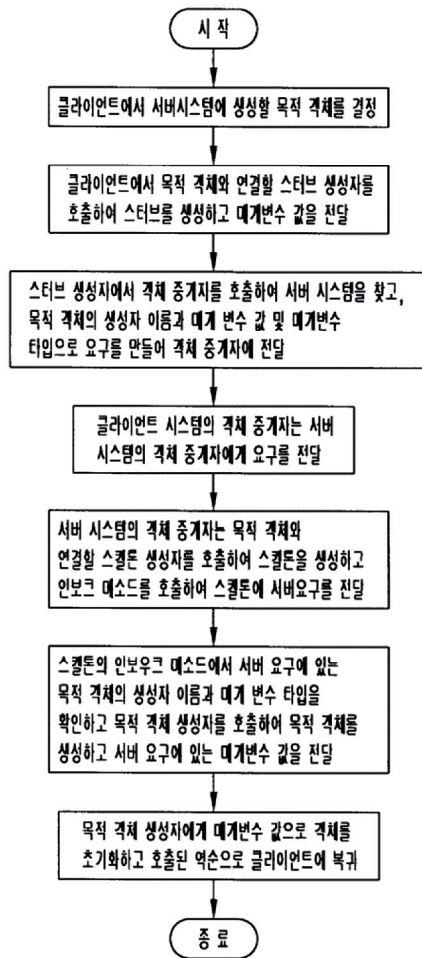
도면6



도면7



도면8



도면9

