(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.[7]
G06F 13/00
G06F 15/16

(11) 공개번호 10-2004-0101537
(43) 공개일자 2004년12월02일

(21) 출원번호 10-2004-7016784
(22) 출원일자 2004년10월19일
번역문 제출일자 2004년10월19일
(86) 국제출원번호 PCT/US2003/012104
(87) 국제공개번호 WO 2003/090090
(86) 국제출원출원일자 2003년04월18일
(87) 국제공개일자 2003년10월30일

(30) 우선권주장 60/373,893 2002년04월19일 미국(US)
10/417,028 2003년04월15일 미국(US)

(71) 출원인 컴퓨터 어소시에이츠 싱크, 인코포레이티드
미국, 뉴욕 11749, 아일랜디아, 원 컴퓨터 어소시에이츠 플라자

(72) 발명자 보딘,그레고리,엘.
미국 60047 일리노이주 롱 그로브 알에프디 8030

앤더슨,앤드류,제이.
미국 80021 콜로라도주 브룸필드 아파트먼트 422 리지 파크웨이 11837

캐인,존,피.
미국 11780 뉴욕주 세인트 제임스 매도우 게이트 이스트 5

(74) 대리인 주성민
백만기
정은진

심사청구 : 없음

(54) 기존의 시스템 관리 제품들 또는 소프트웨어 솔루션들과인터페이스하는 시스템 및 방법

요약

상이한 시스템 제품들 또는 소프트웨어 솔루션들과의 일관된 인터페이스를 제공하는 시스템 및 방법은, 상이한 시스템 제품들 또는 소프트웨어 솔루션들(계기들)이 상주하는 하나 이상의 시스템 유닛들 또는 머신들을 포함한다. 요청을 서비스할 수 있는 계기를 포함하는 특정 시스템 유닛 또는 머신과 관련되는 계기 게이트웨이에 대해 서비스 요청이 이루어질 수 있다. 계기 게이트웨이는 요청된 서비스를 특정한 그 계기로부터 호출하기 위해 특정한 그 계기와 관련되는 계기 스폰서를 선택한다.

대표도

도 1

색인어

일관된 인터페이스, 계기 게이트웨이, 계기 스폰서

명세서

기술분야

본 명세서는 일반적으로 데이터 리소스 관리에 관한 것으로, 보다 구체적으로는, 기존의 컴퓨터 시스템 관리 제품들 또는 소프트웨어 솔루션들과의 일관된 관리 인터페이스를 제공하는 시스템 및 방법에 관한 것이다.

배경기술

컴퓨터 시스템 관리는 기업 경영의 중요한 요소이며 소프트웨어 시장에서 가장 빠르게 성장하는 세그먼트들 중의 하나이기도 하다. 점차 많은 양의 제품들(예를 들어, 소프트웨어 및 하드웨어)이 이용되고 구매됨에 따라, 더 큰 책임성(accountabilty)에 대한 필요성이 증가한다. 신용력을 입증하기 위한 비용 제어 필요성의 증가와 함께, 이용 중인 하드웨어 및 소프트웨어를 획득하기 위해 이루어진 막대한 지출로 인해 증가된 책임성이 필요해졌으며, 이로 인해 시스템 관리에 기업 경영의 보다 결정적인 역할이 부여되었다.

상이한 도구들의 집합을 이용하는 종래의 시스템 관리 접근들은, 이러한 개별적 관리 도구들의 사용과 관련된 비용이 일반적으로, 관리될 필요가 있는 하드웨어 및 소프트웨어에 행해진 초기 투자를 초과하는 관리 비용을 초래하기 때문에, 비용-비효율적임이 입증되었다. 사용되는 각각의 도구들이 상이한 능력을 가지며 이들 중 몇몇은 관리 목적을 달성하기 위해 사용되어야 하므로, 사용되는 도구 수의 감소는 대개 불가능하다. 이 문제를 해결하기 위한 종래의 다른 접근들로 인해, 궁극적으로 기존 제품들의 새로운 버전에 시간에 걸쳐 (예를 들어, 일반적으로 표준 준수에 대한 고객 요청에 응답하여) 기재되고 구현된 ISD(Interface Standard Definitions) 또는 ISP(Interface Standard Proposals)가 조성되어 왔다.

<발명의 개요>

본 명세서는 기존 기술로의 표준 인터페이싱 기술이 실현될 수 있는 메커니즘을 설명한다. 일 실시예에서는, 기존 제품들 또는 관련 기술들에 대한 새로운 통신 프로토콜, 메시징 구성체(messaging constructs), 또는 다른 중요 업데이트의 도입 또는 개발없이, 표준들에 기초한, 일관된 인터페이스들이 이들 제품들 및 기술들에 제공된다.

따라서, 상이한 제품들 또는 소프트웨어 솔루션들과의 일관된 인터페이스를 제공하는 시스템 및 방법은, 상이한 시스템 제품들 또는 소프트웨어 솔루션들(계기들;instruments)이 상주하는 하나 이상의 시스템 유닛 또는 머신을 포함할 수 있다. 요청을 서비스할 수 있는 계기(instrument)를 포함하는 특정 시스템 유닛 또는 머신과 관련되는 계기 게이트웨이에 대해 서비스 요청이 이루어질 수 있다. 계기 게이트웨이는 요청된 서비스를 특정한 그 계기로부터 호출하기 위해 특정한 그 계기와 관련되는 계기 스폰서(instrument sponsor)를 선택한다. 계기 스폰서는 요청자와 서비스 요청을 완수할 수 있는 특정 계기간에 제품-특정 인터페이스 (product-specific interface)를 제공한다. 계기 스폰서는, 메시지 페이로드(message payload)의 적절한 인터페이스 및 계기 게이트웨이를 통해, 서비스 결과를 요청자에게 리턴할 수 있다. 예시적 일 실시예에서, 메시지 페이로드는 XML 메시지를 구비할 수 있다.

도면의 간단한 설명

본 발명의 보다 완전한 이해를 위해, 첨부 도면들과 함께, 다음의 상세한 설명을 참조한다.

도 1은 예시적 일 실시예를 구현하는데 사용될 수 있는 예시적 시스템을 도시한다.

도 2는 예시적 일 실시예를 구현하는데 사용될 수 있는 예시적 방법을 도시한다.

실시예

도 1은 예시적 일 실시예를 구현하는데 사용될 수 있는 예시적 시스템(10)을 도시한다. 시스템(10)은 복수의 시스템 유닛들 또는 머신들(12a-12n)을 포함한다. 4개의 이러한 시스템 유닛(12a-12n)이 도시되어 있지만, 이는 단지 예시적 목적을 위한 것이며 적당한 임의 갯수의 시스템 유닛(12)이 사용될 수 있다. 이 예의 경우, 시스템 유닛들(12a-12n)은 유사하게 구성될 수 있다. 그러나, 시스템 유닛들(12a-12n) 각각에 제공되는 시스템 제품들, 소프트웨어, 또는 계기는, 예를 들어, 데이터 저장 머신과 같은 상이한 제품들 또는 솔루션들일 수 있다.

각각의 시스템 유닛(12a-12n)은 계기 게이트웨이(14)를 포함할 수 있다. 계 기 게이트웨이(14)는 주로, 소정 시스템 유닛 또는 머신(12a-12n)상의 모든 계기 스폰서(16a-16n)로의 액세스를 제공할 수 있는 디스패처(예를 들어, 객체 요구 매개자;object request broker)로서 동작한다. 본질적으로, 계기 게이트웨이(14)는, 그를 통해 계기 스폰서(16a-16n)로의 액세스가 획득될 수 있는 중심 도관(central conduit)으로 동작한다. 3개의 이러한 계기 스폰서(16a-16n)가 도시되어 있지만, 본 명세서가 이에 한정되는 것은 아니며, 적당한 임의 갯수의 계기 스폰서(16)가 사용될 수 있다. 각각의 계기 스폰서(16a-16n)는 소정 시스템 유닛 또는 머신(12a-12n)상의 계기 게이트웨이(14)에 대한 하위 에이전트로서 동작할 수 있다. 계기 스폰서(16a-16n)는 주로, 그를 통해 계기 스폰서들이 작용하도록 설계된 특정 시스템 제품들 또는 소프트웨어 솔루션들의 능력에 대한 액세스가 획득될 수 있는, 계기 게이트웨이(14)로의 표준 인터페이스를 노출시키는 기능을 한다. 각각의 계기 스폰서(16a-16n)는 지식 제공자의 실행가능한 인스턴스 생성(executable instantiation)으로 동작할 수 있으며 기존의 시스템 제품들을 클라이언트와 인터페이스할 수 있게 한다. 이와 같이, 각각의 계기 스폰서는 실질적인 지식 생성자 또는 서비스 제공자로서 동작할 수 있다.

지식 제공자는, 소정의 시스템 관리자 포트렛(portlets)에 의해 요청되는 정보 및 서비스로의 액세스를 제공하기 위해 포털 서버로부터 호출될 수 있는 애플리케이션이다. 포트렛은, 포털 사용자가 다양한 시스템 관리-관련 태스크들에 액세스할 수 있도록 하는 포털 윈도우이다. 포털 서버는 포털 서버 애플리케이션이 실행 중인 머신일 수 있다.

기초 지식 제공자는, 계기층(18a-18n;instrumentation layer)의 단일 애플리케이션에 의해 제공되는 하나 이상의 적당한 인터페이스들에 대한 직접 호출에 의해 서비스들로의 액세스를 제공하는 지식 제공자이다(예를 들어, 기초 지식 제공자는 단일 계기 스폰서(16a-16n)를 호출할 수 있다). 복합 지식 제공자는, 계기층(18a-18n)의 복수의 단일 애플리케이션들에 의해 제공되는 하나 이상의 적당한 인터페이스들에 대한 직접 호출에 의해 서비스들로의 액세스를 제공하는 지식 제공자이다(예를 들어, 복합 지식 제공자는 다수의 계기 스폰서들(16a-16n)을 호출할 수 있다).

각각의 계기층(18a-18n)은, 클라이언트 애플리케이션(20)이 (예를 들어, 계기 스폰서(16a-16n)를 통해) 인터페이스할 수 있는 기존의 시스템 제품 또는 소프트웨어 솔루션으로 기능할 수 있다. 이와 같이, 계기층(18a-18n)의 시스템 제품들 또는 소프트웨어 솔루션들의 기능들이 관련 제품들, 소프트웨어 또는 계기들의 실제 태스크들을 수행하도록 포트렛을 통해 호출될 수 있다. 계기 스폰서는, 클라이언트의 서비스 요청을 완수하기 위해 적절한 계기층 또는 제품들(18a-18n)을 호출할 수 있다.

동작시에, 사용자 또는 클라이언트 애플리케이션(20)은 하나 이상의 시스템 유닛들(12a-12n)상의 기존 시스템 제품, 소프트웨어 솔루션, 또는 관련 기술과의 인터페이스를 시도한다. 예시적 일 실시예의 경우, 시스템(10)은 시스템 관리자에게 상이한 많은 시스템 기술들(저장, 보안, 네트워크 제어 등)에 대해 그리고 상이한 플랫폼들에 대해 시스템 관리를 시청하고(view), 구현하며, 보고하고, 분석하 며, 모니터링하고, 자동화하기 위한 공통 인터페이스를 제공할 수 있는 웹-기반 포털로 간주될 수 있다. 애플리케이션(20)은, 관련 제품에 주문 중인 서비스에 대한 요청의 특징을 설명하는 표준 메시지를 생성할 수 있다. 예를 들어, 애플리케이션(20)은 수행 중인 요청을 설명하기 위해 XML(Extensible MArkup Language) 메시지를 생성할 수 있다. 이 메시지는 일반적으로 메시지 페이로드(또는 XML 메시지의 경우, 'XML 메시지 페이로드')라 할 수 있다. XML 메시지 페이로드의 내용은, 관련된 시스템 제품, 소프트웨어 솔루션, 또는 계기와의 트랜잭션 또는 그에 대한 서비스 요청과 관련된 조회 및 포매팅 기준(query and formatting criteria)을 결합하여 구성하는 XML 태그들 및 값들을 포함할 수 있다. 제품에 대한 트랜잭션 또는 서비스 요청을 포함할 수 있는 메시지 페이로드는 인터페이스될 제품이 활성화되어 있는 시스템 유닛(12a-12n)으로 전달될 수 있다. 예를 들어, 메시지 페이로드는 애플리케이션(20)으로부터 인터넷(예를 들어, 네트워크(19))을 통해, 예를 들어, 표준형 메시지에 대한 HTTP 1.1 또는 일반적인 것보다 좀더 안전한 핸들링을 요하는 메시지 페이로드에 대한 HTTPS 1.1과 같은, 표준 메시지 전송 프로토콜을 사용해 시스템 유닛(또는 머신;12a-12n)상의 포트로 전달될 수 있다. 애플리케이션(20)으로부터 전달된 메시지 페이로드를 수신하는 시스템 유닛(12a-12n)상의 포트는 계기 게이트웨이(14)에 결합될 수 있으며 그에 의해 서비스될 수 있다.

네트워크(19)는 사용자(예를 들어, 20)를 하나 이상의 시스템 유닛들 또는 머신들(12a-12n)에 결합시킬 수 있는 공중망 및/또는 사설망의 적절한 임의 조합을 포함할 수 있다. 예시적 일 실시예에서, 네트워크(19)는 사용자(20)를 하나 이상 의 시스템 유닛들 또는 머신들(12a-12n)에 결합시키는 인터넷 및 적절한 임의의 LAN(local area networks), MAN(metropolitan area networks), 또는 WAN(wide area networks)을 포함한다.

계기 게이트웨이(14)는, 애플리케이션(20)으로부터의 메시지 페이로드의 요청을 서비스하기 위해 어떤 계기 스폰서(16a-16n)가 선택되어 호출될 것인지 판정하기 위해 인바운드(inbound) 메시지 페이로드를 검사할 수 있다. 바람직하게도, 각각의 계기 스폰서(16a-16n)는 요청된 태스크(들)를 수행하기 위해 호출될 수 있는 계기층(18a-18n)의 특정 시스템 제품, 소프트웨어 솔루션, 또는 계기와 관련될 수 있다. 그 다음, 선택된 계기 스폰서(16a-16n)는 요청된 태스크(들)을 수행하기 위한 적절한 시스템 제품, 소프트웨어 솔루션 또는 계기를 호출할 수 있다. 특히, 선택된 계기 스폰서(16a-16n)는 요청된 태스크(들)를 수행하는데 필요한 임의의 제품-특정 인터페이스(들)을 제공하도록 구성될 수 있다.

상술한 바와 같이, 계기 게이트웨이(14)는 클라이언트 애플리케이션(20)으로부터의 요청을 서비스하기에 적절한 계기 스폰서(16a-16n)를 선택하고 호출할 수 있다. 예시적 일 실시예의 경우, 각각의 계기 스폰서(16a-16n)는 실행가능한 소프트웨어일 수 있으며, DLL(Dynamic Link Library); COM(Component Object Model) 오브젝트[예를 들어, VB(Visual Basic), OLE(Object Linking and Embedding), OCX(OLE Custom Control), ActiveX 등]; 자바 클래스; VB 스크립트(VBScript)의 포맷들 중 하나 이상을 가진 외부 프로세스(out-of-process) 애플리케이션 또는 내부 프로세스(in-process) 애플리케이션으로 실행될 수 있다. 계기 스폰서(16a-16n) 각각의 능력들 및 속성들에 관한 일관된 정보가 (나타내지 않은) 중앙에서 액세스가능한 등록 기관(centrally accessible registration authority)에 저장될 수 있다. 등록 기관은 계기 스폰서(16a-16n)에 관한 등록 정보를 UDDI(Universal Description, Discovery and Integration) 디렉토리 및 다른 웹-기반 서비스 디렉토리에 저장하기 위한 접촉점으로 동작할 수 있다.

등록 정보는 계기 스폰서(16a-16n) 각각에 대한 디지털 서명(digital signature)을 포함할 수 있다. 디지털 서명은, 계기 게이트웨이(14)에 의해 로딩 또는 호출 중인 특정한 계기 스폰서가 변경되지 않았음을 보증하기 위하여, 소정 플랫폼 또는 시스템 유닛(12a-12n)상의 계기 게이트웨이(14)에 의해 (예를 들어, 옵션으로) 사용될 수 있다. 그에 따라, 예시적 일 실시예의 경우, 마이크로소프트 부호 코드 메커니즘(Microsoft sign code mechanism)이 계기 스폰서(16a-16n)를 지원하는 DLL, COM 오브젝트, 자바 클래스 등의 디지털 부호화에 사용될 수 있다.

계기 게이트웨이(14)가, 계기층(18a-18n)의 적절한 시스템 제품 또는 소프트웨어 솔루션을 호출하기 위해, 선택된 계기 스폰서(16a-16n)로 메시지 페이로드를 송신할 때, 계기 게이트웨이(14)는, 한가지 이상의 많은 방법으로, 일관된 메시지 정보를 선택된 계기 스폰서에 로딩할 수 있다. 그에 따라, 계기 게이트웨이(14)에 의해 사용되는 로딩 방법은 관련된 계기 스폰서(16a-16n)의 플랫폼 및 프로세싱 타입에 따라 달라질 수 있다. 예를 들어, 선택된 계기 스폰서(16a-16n)가 윈도우즈-기반의 플랫폼을 가진 외부 프로세스 메소드에 사용되어야 한다면, 계기 게이트웨이(14)는, 일관된 메시지 정보를 포함하는 적당한 COM 오브젝트의 인스턴스를 생성 함으로써(예를 들어, COM은 암시적 NT 생성 프로시저(implicit NT Create Procedure)를 구성할 수 있음), 선택된 계기 스폰서에 일관된 메시지 정보를 로딩할 수 있다. 또 하나의 예로서, 선택된 계기 스폰서(16a-16n)가 유닉스-기반의 플랫폼을 가진 외부 프로세스 메소드에 사용되어야 한다면, 계기 게이트웨이(14)는, 공유 라이브러리 및/또는 Fork나 Exec 명령을 통해, 선택된 계기 스폰서에 일관된 메시지 정보를 로딩할 수 있다.

그러나, 선택된 계기 스폰서(16a-16n)가 윈도우즈-기반 또는 유닉스-기반의 플랫폼을 가진 내부 프로세스 메소드에 사용되어야 한다면, 계기 게이트웨이(14)는, 정보를 포함하는 스레드(thread)를 생성하고 선택된 계기 스폰서(16a-16n)가 계기 게이트웨이(14)의 제어하에서 스레드(예를 들어, 다른 부분들과 독자적으로 실행될 수 있는 프로그램 부분)로서 실행될 수 있도록 스레드의 인스턴스를 생성함으로써, 선택된 계기 스폰서에 일관된 메시지 정보를 로딩할 수 있다.

상술한 바와 같이, 각각의 계기 스폰서(16a-16n)는 입력되는 메시지 페이로드(예를 들어, XML 메시지 페이로드)를 사용자(예를 들어, 클라이언트 애플리케이션(20))에 의해 요청 중인 시스템 또는 소프트웨어 기능을 식별하는 계기 게이트웨이(14)를 통해 수용할 수 있다. 또한, 각각의 계기 스폰서(16a-16n)는 계기 게이트웨이(14)를 통해 요청자(예를 들어, 애플리케이션(20))에게 메시지(예를 들어, XML 메시지 버퍼)를 출력할 수 있다. 구체적으로, 계기 게이트웨이(14)와 계기 스폰서들(16a-16n)간에서 메시지를 전달하는데 사용되는 메커니즘은 사용되는 계기 스폰서의 플랫폼과 프로세싱 타입에 따라 달라질 수 있다. 예를 들어, 계기 스폰 서(16a-16n)가 윈도우즈-기반 플랫폼의 외부 프로세스 메소드에 사용되어야 한다면, COM RPC(Remote Procedure Call) 메커니즘이 메시지를, 계기 게이트웨이(14)에 의해 시작되지만, 다른 관점에서, 마스터 에이전트(예를 들어, 계기 게이트웨이)의 서비스와는 별개로 실행되는, 프로세스로 송신하는데 사용될 수 있다. COM RPC를 사용해, 클라이언트 프로그램은 적절한 인수(arguments)를 가진 서버로 메시지를 송신할 수 있으며, 서버는 실행된 프로그램의 결과를 포함하는 메시지를 리턴할 수 있다. 또 하나의 예로서, 계기 스폰서(16a-16n)가 유닉스-기반 플랫폼의 외부 프로세스 메소드에 대해 사용되어야 한다면, 계기 스폰서는 'Fork' 또는 'Exec' 명령을 통해 호출될 수 있다. 그에 따라, 계기 스폰서는 'stdin' 필드로부터 입력 중인 메시지(예를 들어, XML 메시지)를 판독하고, 'stdout' 필드에 기입함으로써 응답 메시지를 출력할 수 있다. 그러나, 계기 스폰서(16a-16n)가 (예를 들어, 윈도우즈-기반 또는 유닉스-기반 플랫폼의) 내부 프로세스 메소드에 대해 사용되어야 한다면, 초기에, 핸들 또는 포인터가 인바운드 버퍼(예를 들어, char *inbuff)로 전달될 수 있으며, 또 하나의 포인터가 아웃바운드 버퍼(예를 들어, char *outbound)로 전달될 수 있다.

다음은 계기 스폰서(16a-16n)를 호출하기 위해 계기 게이트웨이(14)에 의해 사용될 수 있는 유닉스-기반 코드의 일례이다.

```
Loadlib( bebSubAjent.dll)
Beb.InvokeSub( "INIT")
wait for inbound data
if for bebAgent
        beb.InvokeSub("INVOKE",ib, ob);
        sendRply( ob );
        beb.InvokeSub("FREE", ob);
        continue;
```

다음은 계기 게이트웨이(14)로부터 메시지를 수용하고 응답하기 위해 계기 스폰서(16a-16n)에 의해 사용될 수 있는 코드의 일례이다.

```
fn: bebSubAject.dll
long __declspec( dllexport) __cdecl InvokeSub(
 PCHAR szFunction,
 ...
 )
{
 va_list argptr;
```

```
if ( stricmp( szFunction, "INITIALIZE") == 0 ){
 return ( ss.Initialize() );
}
else if ( stricmp( szFunction, "TERMINATE") == 0 ){
 return ( ss.Terminate() );
}
else if ( stricmp( sxFunction, "INVOKE") == 0 ){
 char* ibXml;
 char** obXml;
 va_start( argptr, szFunction );
 ibXml = va_arg( argptr, char* );
 obXml = va_arg( argptr, char** );
 va_end( argptr );
 return ( ss.Invoke( ibXml, obXml ) );
}
else if ( stricmp( szFunction, "FREE") == 0 ){
 char* obj;
 va_start( argptr, szFunction );
 obj = va_arg( argptr, char* );
 va_end( argptr );
 return ( ss.Free( obj ) );
}
return 400;
}
```

각각의 계기 스폰서(16a-16n)는 하나 이상의 메시지 페이로드(예를 들어, XML 메시지 페이로드)를 통해 계기 게이트웨이(14)로부터 호출될 수 있는 복수의 기능들 또는 메소드들을 수행할 수 있다. 예를 들어, 계기 스폰서(16a-16n)는 적어도 다음의 메소드들을 수행할 수 있다.

'Describe' 메소드는 자신에 관한 정보를 제공할 것을 에이전트에 요청하는데 사용될 수 있다. 예를 들어, 'Describe' 메소드가 계기 게이트웨이(14)로 송신되면, 계기 게이트웨이는 그 자신을 설명하는 메시지로 응답할 수 있는데, 현재 설치되어 있는 모든 계기 스폰서들은 계기 게이트웨이가 실행 중인 머신(시스템 유닛(12a-12n))상의 그 계기 게이트웨이와 관련되어 있으며, 모든 제품들 또는 소프트웨어 솔루션들은 계기 게이트웨이가 실행 중인 머신상에서 계기 게이트웨이가 검출할 수 있는 것들이다. 'Describe' 메소드가 계기 스폰서로 송신되면, 계기 스폰서는 자신을 설명할 수 있다. 다음은 이러한 Describe 메소드에 대한 코드의 일례이 다.

```
accepts
        <Method=describe/>
returns:
        <Version=string/>
        <ProdInterfaceInfo=(ProdInterface Type*)
        <PlatformInfo=string/>
        <CanIssueAsyncNotify=string/>                (y,n)
        <ReqDTDVersionID=(Version*) Version of Request
DTD this instrument sponsor supports
        <RespDTDVersionID=(Version*) Version of
Response DTD this instrument sponsor supports
        ProdInterfaceType(ProdName,Release,SubStat,
         Active)
                Name          CDATA Required
                Release       CDATA Required
                SubStat       Number (1,0)  (1=instrument
                sponsor installed, 0=instrument sponsor
                not installed)
                Active Number (1,0) (1=Active,
                0=Inactive)
        Version(VersionInfo)
                VersionInfo CDATA
```

'AsyncNotifySetDestination' 메소드는, 계기 스폰서(16a-16n)가 비동기 통지들을 송신할 수 있는 장소를 계기 스폰서(16a-16n)에 통보하는데 사용될 수 있다. 다음은 AsyncNotifySetDestination 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts <Method=AsyncNotifySetDestination>

        <AsyncNotifySetDestinationType=string/>
                                              (trap,http)
        <AsyncNotifySetDestinationTargets=string/>
                (machine namelip addr, port ; machine
                 namelip addr, port; etc. )
        <AsyncNotifySetDestinationPort=string/> (port)
  returns <AsyncNotifySetDestinationStatus=string/>
                                    (success, failure)
        <AsyncNotifySetDestinationReason=string/>
                                    (explanation for failure)
```

'ShowGrammar' 메소드는 에이전트로 인터페이스하는 메소드에 관한 정보를 제공할 것을 에이전트에 요청하는데 사용될 수 있다. 다음은 ShowGrammar 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts
        <Method=ShowGrammar/>
returns:
        <ResponseType=string/>      (URL, DTD, SCHEMA)
        <ResponseGrammar=string/>
```

'Init' 메소드은 시작할 것을 에이전트에 지시하는데 사용될 수 있다. 일반적으로, 이 메소드는, 계기 스폰서에 메시지 페이로드를 로딩할 때, 계기 게이트웨이에 의해 그 계기 스폰서로 송신될 수 있다. 다음은 Init 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts
        <Method=Term/>
returns:
        <InitStatus=string/>
```

'Term' 메소드는 셧다운(shutdown)할 것을 에이전트에 지시하는데 사용될 수 있다. 일반적으로, 이 메시지는, 계기 게이트웨이가 (예를 들어, 클라이언트 애플리케이션 또는 시스템 관리자에 의해) 계기 스폰서를 셧다운하도록 지시될 때, 계기 게이트웨이에 의해 계기 스폰서로 송신될 수 있다. 다음은 Term 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts

        <Method=Term/>
returns:
        <InitStatus=string/>
```

'Configure' 메소드는 계기 스폰서의 구성가능한 설정들을 설정 또는 조회하는데 사용될 수 있다. 원래의 구성 값들이 계기 스폰서에 대해 설정될 때, 관련 플랫폼상의 그 계기 스폰서에 대한 후속적 재시동에서 이 값들은 고수된다. 특히, 다중 구성 설정이 단일 메시지에 시도되면, 관련된 계기 스폰서는 이들 구성 설정을 동시에 실행해야 한다. 새로운 구성값들을 설정하려는 시도들 중 하나라도 실패하면, 모든 값들은 Configure 메소드가 호출되기 전에 위치했던 구성값들로 되 돌려질 수 있다. 다음은 Configure 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts
        <Method=ConfigureSub/>
        <ConfigurePolarity=(ConfigPolarity*)
        <ConfigureSetting=(ConfigureVariable*)
returns:
        <ConfigureStatus=string/>    (success,
                failure)
        <ConfigureStatusReason=string/>    (explanation
                for failure)
ConfigurePolarity(Polarity)
        Polarity Number (1,0) (0=Get, 1=Set)
Polarity CDATA              (Valid values are get or set)
ConfigureVariable(VarName, VarValue)
        VarName CDATA
        VarValue CDATA
```

'Invoke' 메소드는, 계기 스폰서(16a-16n)가 사용자를 위해 수행할 수 있는 서비스(예를 들어, 정보 검색, 정책 관리, 동작 호출 등)를 요청하는데 사용될 수 있다. 다음은 Invoke 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts

        <Method=InvokeSub, InstallProd, DeinstallProd,
         InitProd, TermProd, CallProd>
        <method specific grammar>
 returns:
        <InvokeStatus=string/>
```

'Authenticate' 메소드는 자격 증명(credentials)을 계기 스폰서(16a-16n)로 송신하는데 사용될 수 있는데, 계기 스폰서는 사용자(예를 들어, 클라이언트 애플리케이션(20))에 의해 송신된 후속 서비스 요청을 그 계기 스폰서에 허용할 것인지의 여부를 판정하기 위해 인증할 수 있다. 바람직하게도, 계기 게이트웨이(14)를 인증하는 대신에, Authenticate 메소드는, 계기 게이트웨이를 통해 계기 스폰서에 제시된 요청의 서비스 요청자를 인증할 수 있다. 요청자가 인증되고 나면, 관련된 계기 스폰서들에 의해 MAC Addr 또는 IP Addr (또는 요청자를 식별할 수 있는 다른 토큰)이 국부적으로 캐시 또는 저장될 수 있다. 후속적으로, 계기 스폰서는 국부 캐시로부터 자격 증명을 검색하고 검색된 자격 증명 정보에 기초하여 액세스 판정을 내릴 수 있다. 그에 따라, 계기 게이트웨이(14) 및 디지털 부호의 모든 에이전트들은 계기 스폰서(16a-16n)들의 서명을 확인할 수 있다. 다음은 Authenticate 메소드에 사용될 수 있는 코드의 일례이다.

```
accepts
        <Method=Authenticate/>
        <UserID=userid string/>
        <Password=password/>
returns:
        <AuthenticateStatus=string/>
```

도 2는 예시적 일 실시예를 구현하는데 사용될 수 있는 예시적 메소드(100)를 도시한다. 단계 102에서, 클라이언트 또는 사용자(예를 들어, 20)는, 특정한 시스템 제품 또는 소프트웨어 솔루션에 대한 서비스 요청을 포함하는 메시지 페이로드(예를 들어, XML 메시지 페이로드)를 생성한다. 단계 104에서, 클라이언트 또는 사용자는 메시지 페이로드를 하나 이상의 시스템 유닛들 또는 머신들(12a-12n) 에 대한 포털과 관련되는 어드레스로 송신한다. 예를 들어, 이 어드레스는 사용자에 의해 특정되거나 형성 중인 요청의 타입에 기초하여 판정될 수 있다.

단계 106에서는, 서비스 요청을 포함하는 메시지 페이로드의 수신에 응답하여, (예를 들어, 각각이 특정한 시스템 유닛 또는 머신(12a-12n)과 관련되는) 하나 이상의 계기 게이트웨이(14)는 'Describe' 메소드 요청을 그 계기 게이트웨이와 관련되는 계기 스폰서(16a-16n)로 송신한다. 요청을 서비스하기에 적절한 시스템 제품 또는 소프트웨어 솔루션이 계기 스폰서(12a-12n)를 통해 이용가능한지를 판정하는 것에 응답하여, 단계 108에서는, 계기 게이트웨이(14)가 요청을 서비스할 그 계기 스폰서를 선택한다. 단계 110에서는, 선택된 계기 스폰서(12a-12n)가, 시스템 제품 또는 소프트웨어 솔루션에 대한 제품-특정 인터페이스를 통해, 계기층(18a-18n)으로부터 관련 시스템 제품 또는 소프트웨어 솔루션(계기)을 호출한다. 예를 들어, 적절한 시스템 제품 또는 소프트웨어 솔루션이 계기(18a)로서 계기층에 상주하면, 계기 게이트웨이(14)는 형성된 서비스 요청을 완수하기 위해 그 계기에 대해 설계된 적절한 인터페이스를 통해 그 계기를 호출하기 위해 (예를 들어, 계기(18a)와 직접적으로 관련되는) 계기 스폰서(16a)를 선택한다. 단계 112에서는, 선택된 계기 스폰서(16a-16n)가 서비스 결과를 포함하는 메시지 페이로드를 (예를 들어, 계기 게이트웨이(14)를 통해) 요청자에게로 송신할 수 있다.

바람직한 실시예를 첨부 도면들에 도시하고 상기의 상세한 설명에 설명하였지만, 본 개시는 개시된 실시예에 한정되지 않으며, 다음의 청구항들에 의해 설명되고 정의된 본 발명의 정신을 벗어나지 않으면서, 다양한 재구성, 변형 및 대체가 가능하다는 것을 알 수 있다.

(57) 청구의 범위

청구항 1.
복수의 계기들과의 일관된 인터페이스를 제공하는 시스템에 있어서,

적어도 하나의 시스템 유닛;

상기 시스템 유닛들 중 하나 이상에 결합되는 계기 게이트웨이;

각각이 하나의 계기 게이트웨이에 결합되는 복수의 계기 스폰서들; 및

각각이 상기 복수의 계기 스폰서들 중 하나에 결합되는 복수의 계기들을 포함하는 인터페이스 제공 시스템.

청구항 2.
제1항에 있어서,

상기 복수의 계기들 각각은 개별 인터페이스를 통해 상기 복수의 계기 스폰서들 중 하나에 결합되는 인터페이스 제공 시스템.

청구항 3.
제1항에 있어서,

상기 시스템 유닛들 중 하나는 데이터 저장 머신을 포함하는 인터페이스 제공 시스템.

청구항 4.
제1항에 있어서,

상기 복수의 계기들은 복수의 시스템 제품들을 포함하는 인터페이스 제공 시스템.

청구항 5.
제1항에 있어서,

상기 복수의 계기들은 복수의 소프트웨어 솔루션들을 포함하는 인터페이스 제공 시스템.

청구항 6.
복수의 계기들과의 일관된 인터페이스를 제공하는 시스템에 있어서,

서비스 요청 메시지를 생성하는 수단;

상기 서비스 요청 메시지를 하나 이상의 시스템 유닛과 관련되는 게이트웨이로 송신하는 수단;

복수의 계기들 중 하나가 상기 서비스 요청 메시지에 포함된 서비스 요청을 완수하는 데 이용될 수 있는지를 판정하는 수단;

상기 계기가 이용될 수 있으면, 상기 계기와 관련되는 스폰서를 선택하는 수단; 및

상기 계기에 대해 상기 서비스를 호출하는 수단을 포함하는 인터페이스 제공 시스템.

청구항 7.
제6항에 있어서,

상기 복수의 계기들 중 상기 계기를 인터페이스를 통해 상기 스폰서에 결합시키는 수단을 더 포함하는 인터페이스 제공 시스템.

청구항 8.
제6항에 있어서,

상기 복수의 계기들은 복수의 시스템 제품들을 포함하는 인터페이스 제공 시스템.

청구항 9.
제6항에 있어서,

상기 복수의 계기들은 복수의 소프트웨어 솔루션들을 포함하는 인터페이스 제공 시스템.

청구항 10.
복수의 계기들과의 일관된 인터페이스를 제공하는 방법에 있어서,

하나 이상의 시스템 유닛과 관련되는 게이트웨이에서 서비스 요청 메시지를 수신하는 단계;

복수의 계기들 중 하나의 계기가 상기 서비스 요청 메시지에 포함된 서비스 요청을 완수하는 데 이용될 수 있는지를 판정하는 단계;

그렇다면, 상기 계기와 관련된 인터페이스를 포함하는 스폰서를 선택하는 단계; 및

상기 계기에 대해 상기 인터페이스를 통해 상기 서비스를 호출하는 단계를 포함하는 인터페이스 제공 방법.

청구항 11.
제10항에 있어서,

상기 서비스 요청 메시지를 생성하는 단계는 XML 메시지를 생성하는 단계를 포함하는 인터페이스 제공 방법.

청구항 12.
제10항에 있어서,

상기 호출된 서비스의 결과들을 XML 메시지를 사용해 서비스 요청자에게로 리턴하는 단계를 더 포함하는 인터페이스 제공 방법.

청구항 13.
제10항에 있어서,

상기 서비스 요청 메시지를 생성하는 단계는 사용자에 의해 수행되는 인터페이스 제공 방법.

청구항 14.
제10항에 있어서,

상기 복수의 계기들 중 상기 계기가 상기 서비스 요청 메시지에 포함된 서비스 요청을 완수하는데 이용될 수 있는지를 판정하는 단계는 상기 게이트웨이와 관련되는 복수의 계기 스폰서들에게로 설명(describe) 메시지를 송신하는 단계를 포함하는 인터페이스 제공 방법.

청구항 15.
제10항에 있어서,

상기 인터페이스는 상기 계기에 대해 제품-특정 인터페이스를 포함하는 인터페이스 제공 방법.

청구항 16.
제10항에 있어서,

상기 선택 단계는 상기 게이트웨이에 의해 수행되는 인터페이스 제공 방법.

청구항 17.
제10항에 있어서,

상기 호출 단계는 상기 스폰서에 의해 수행되는 인터페이스 제공 방법.

청구항 18.
제10항에 있어서,

상기 하나 이상의 시스템 유닛은 데이터 저장 머신을 포함하는 인터페이스 제공 방법.

청구항 19.
제10항에 있어서,

상기 게이트웨이는 계기 게이트웨이를 포함하는 인터페이스 제공 방법.

청구항 20.
제10항에 있어서,

상기 스폰서는 계기 스폰서를 포함하는 인터페이스 제공 방법.

청구항 21.
제10항에 있어서,

상기 복수의 계기들은 복수의 시스템 제품들을 포함하는 인터페이스 제공 방법.

청구항 22.
제10항에 있어서,

상기 복수의 계기들은 복수의 소프트웨어 솔루션들을 포함하는 인터페이스 제공 방법.

청구항 23.
컴퓨터-판독가능 매체에 구현된, 복수의 계기들과의 일관된 인터페이스를 제공하는 소프트웨어로서, 실행될 경우,

하나 이상의 시스템 유닛과 관련되는 게이트웨이에서 서비스 요청 메시지를 수신하고;

복수의 계기들 중 하나의 계기가 상기 서비스 요청 메시지에 포함된 서비스 요청을 완수하는 데 이용될 수 있는지를 판정하며;

그렇다면, 상기 계기와 관련된 인터페이스를 포함하는 스폰서를 선택하고;

상기 계기에 대해 상기 서비스를 호출하도록 동작할 수 있는 인터페이스 제공 소프트웨어.

청구항 24.
제23항에 있어서,

상기 복수의 계기들 중 하나를 상기 인터페이스를 통해 상기 스폰서에 결합시키도록 동작할 수도 있는 인터페이스 제공 소프트웨어.

청구항 25.
제23항에 있어서,

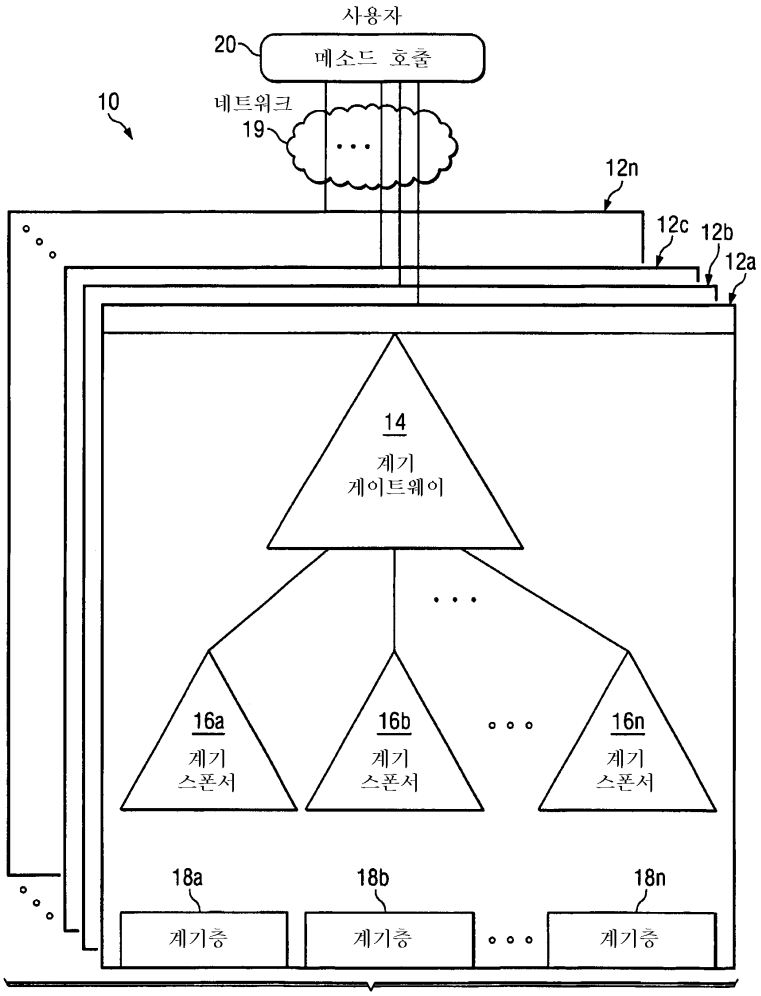
상기 복수의 계기들은 복수의 시스템 제품들을 포함하는 인터페이스 제공 소 프트웨어.

청구항 26.
제23항에 있어서,

상기 복수의 계기들을 복수의 소프트웨어 제품들을 포함하는 인터페이스 제공 소프트웨어.

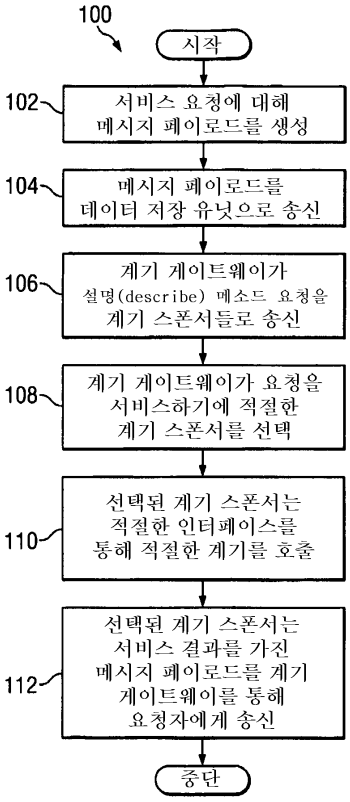
도면

도면1



계기 게이트웨이들과 계기 스폰서들이 채택된 머신들

도면2


100
시작
102
서비스 요청에 대해
메시지 페이로드를 생성
104
메시지 페이로드를
데이터 저장 유닛으로 송신
106
게기 게이트웨이가
설명(describe) 메소드 요청을
게기 스폰서들로 송신
108
게기 게이트웨이가 요청을
서비스하기에 적절한
게기 스폰서를 선택
110
선택된 게기 스폰서는
적절한 인터페이스를
통해 적절한 계기를 호출
112
선택된 게기 스폰서는
서비스 결과를 가진
메시지 페이로드를 게기
게이트웨이를 통해
요청자에게 송신
종단