(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0108485
(43) 공개일자 2008년12월15일

(51) Int. Cl.
G06Q 50/00 (2008.03)
(21) 출원번호 10-2008-7023478
(22) 출원일자 2008년09월25일
심사청구일자 없음
번역문제출일자 2008년09월25일
(86) 국제출원번호 PCT/US2007/001985
국제출원일자 2007년01월24일
(87) 국제공개번호 WO 2007/114884
국제공개일자 2007년10월11일
(30) 우선권주장
11/278,364 2006년03월31일 미국(US)

(71) 출원인
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
콕스, 숀, 디.
미국 98052-6399 워싱톤주 레드몬드 원 마이크로소프트 웨이
(74) 대리인
양영준, 백만기

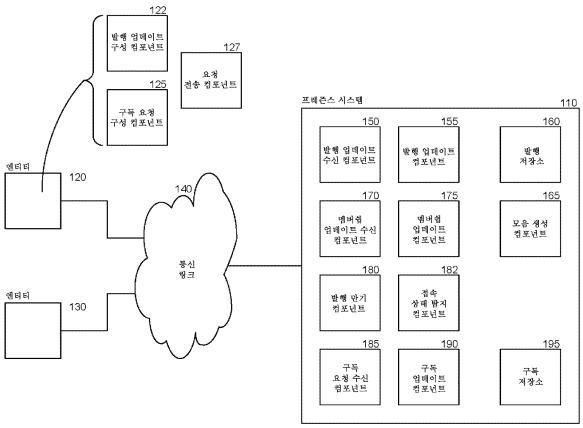
전체 청구항 수 : 총 20 항

(54) 멤버 리스트를 갖는 모음들로 발행되는 프레즌스 정보로의 액세스를 수정하는 방법 및 상기 방법을 수행하게 위한 명령어들을 포함하는 컴퓨터 판독가능 매체

(57) 요 약

발행의 모음을 갖는 프레즌스 서버의 계산 효율성 및 네트워크 효율성을 증가시키기 위한 방법 및 시스템이 제공된다. 프레즌스 시스템은, 비용이 많이 드는 처리를 필요로 하지 않으면서 풍부한 프레즌스 정보를 제공하는 것을 가능하게 하는 여러 기법들을 사용한다. 우선, 프레즌스 시스템은 단일의 프레즌스 업데이트 요청으로 발행 업데이트의 묶음을 받아들인다. 마찬가지로, 프레즌스 시스템은 구독 요청의 묶음을 받아들인다. 다음으로, 프레즌스 시스템은, 발행하는 사용자가 바뀌지 않은 프레즌스 정보를 계속해서 리프레시할 필요가 없게 하는 발행 업데이트 요청에 대한 새로운 만료 유형을 지원한다. 마지막으로, 프레즌스 시스템은 특정 프레즌스 모음으로 액세스하고자 하는 사용자들의 개개 리스트가 아닌 멤버쉽 그룹을 포함하는 액세스 리스트를 허용한다.

대 표 도



특허청구의 범위

청구항 1

멤버 리스트(member list)를 갖는 모음들(collections)로 발행되는(published) 프레즌스 정보(presence information)로의 액세스를 수정하는 시스템으로서,

프레즌스 정보를 발행하고자 하는 요청을 수신하는 요청 수신 컴포넌트(receive request component)(150) -상기 요청은 멤버 리스트가 수정될 모음을 명시(specify)하는 모음 식별자(collection identifier)와 액세스가 수정될 멤버 그룹을 포함함-; 및

상기 요청에 응하여, 명시된 멤버 그룹의 명시된 모음으로의 액세스를 수정하는 액세스 수정 컴포넌트(modify access component)(175)

를 포함하는 액세스 수정 시스템.

청구항 2

제1항에 있어서, 상기 멤버 그룹은 도메인(domain)을 명시하는 액세스 수정 시스템.

청구항 3

제1항에 있어서, 상기 멤버 그룹은 상기 모음 소유자의 연락처 리스트를 명시하는 액세스 수정 시스템.

청구항 4

제1항에 있어서, 상기 멤버 그룹은 연합된 멤버 서버들(federated member servers)을 명시하는 액세스 수정 시스템.

청구항 5

제1항에 있어서, 상기 멤버 그룹은 상기 프레즌스 시스템의 외부에 있는 엔티티에 의해 정의되는 멤버 그룹을 명시하는 액세스 수정 시스템.

청구항 6

제1항에 있어서, 상기 요청은 액세스가 수정될 둘 이상의 멤버 그룹을 포함하는 액세스 수정 시스템.

청구항 7

제1항에 있어서, 상기 요청은 액세스가 수정될 둘 이상의 모음을 포함하는 액세스 수정 시스템.

청구항 8

제7항에 있어서, 상기 모음들은 서로 다른 발행하는 사용자들에 속하는 액세스 수정 시스템.

청구항 9

제1항에 있어서, 상기 요청은 상기 멤버 그룹에게 상기 모음으로의 액세스가 허가될 것임을 명시하는 액세스 수정 시스템.

청구항 10

제1항에 있어서, 상기 요청은 상기 멤버 그룹에게 상기 모음으로의 액세스가 거부될 것임을 명시하는 액세스 수정 시스템.

청구항 11

제1항에 있어서, 구독하는 사용자로부터 프레즌스 정보를 구독하고자 하는 요청을 수신하는 구독 요청 수신 컴포넌트(receive subscription request component)를 더 포함하는 액세스 수정 시스템.

청구항 12

제11항에 있어서, 상기 모음들과 상기 멤버 리스트는 상기 구독하는 사용자의 아이덴티티에 기초하여 달라지는 상세 수준(a level of detail)을 갖는, 구독하는 사용자들에 사용가능한 프레즌스 정보의 계층 구조(hierarchy)를 생성하는 액세스 수정 시스템.

청구항 13

제1항에 있어서, 상기 요청은 SIP 프로토콜을 이용하는 액세스 수정 시스템.

청구항 14

제1항에 있어서, 상기 시스템의 일부 컴포넌트들은 프레즌스 서버에 위치하며, 다른 컴포넌트들은 프레즌스 클라이언트에 위치하는 액세스 수정 시스템.

청구항 15

멤버 리스트를 갖는 모음들로 발행되는 프레즌스 정보로의 액세스를 수정하는 방법으로서,

프레즌스 정보를 발행하고자 하는 요청을 수신하는 단계(150) -상기 요청은 멤버 리스트가 수정될 모음을 명시하는 모음 식별자 및 액세스가 수정될 멤버 그룹을 포함함-; 및

상기 요청에 응하여, 명시된 멤버 그룹의 명시된 모음으로의 액세스를 수정하는 단계(175)

를 포함하는 액세스 수정 방법.

청구항 16

제15항에 있어서, 상기 멤버 그룹은 도메인을 명시하는 액세스 수정 방법.

청구항 17

제15항에 있어, 상기 요청은 액세스가 수정될 둘 이상의 멤버 그룹을 포함하는 액세스 수정 방법.

청구항 18

멤버 리스트를 갖는 모음들로 발행되는 프레즌스 정보로의 액세스를 수정하기 위한 방법을 수행하는 명령어들을 포함하는 컴퓨터 판독가능 매체로서,

상기 방법은,

프레즌스 정보를 발행하고자 하는 요청을 프레즌스 서비스에 전송하는 단계(127) -상기 요청은 멤버 리스트가 수정될 모음을 명시하는 모음 식별자 및 액세스가 수정될 멤버 그룹을 포함함-;

를 포함하고,

상기 요청은 상기 프레즌스 서비스에게 상기 요청에 응하여 명시된 멤버 그룹의 명시된 모음으로의 액세스를 수정할 것을 지시하는 컴퓨터 판독가능 매체.

청구항 19

제18항에 있어서, 상기 멤버 그룹은 모음 소유자의 연락처 리스트를 명시하는 컴퓨터 판독가능 매체.

청구항 20

제18항에 있어서, 상기 요청은 액세스가 수정될 둘 이상의 모음을 포함하는 컴퓨터 판독가능 매체.

명 세 서

배 경 기 술

<1> 사용자의 가능함 상태(availability status of user)와 같은 프레즌스 정보(presence information)를 제공하는 데에 프레즌스 서버(presence server)가 점점 더 많이 사용되고 있다. 사용자의 프레즌스 정보는 그 사용자의

현재의 "프레즌스 상태"를 식별한다. 사용자는 다른 사용자가 자신과 가장 잘 통신하는 방법을 결정할 수 있도록 자신의 프레즌스 정보를 사용하게 할 수 있다. 예를 들면, 프레즌스 정보는 사용자가 인스턴스 메시징 서버에 로그온 했는지("온라인") 또는 로그오프 했는지("오프라인")의 여부를 나타낼 수 있다. 프레즌스 정보는 또한 사용자의 가능함에 관해 좀 더 상세한 정보를 제공할 수 있다. 예를 들면, 사용자가 온라인일지라도, 그 사용자는 회의로 인해 자신의 컴퓨터에서 멀리 있을 수 있다. 이런 경우, 프레즌스 상태는 "온라인"이고 "회의중"을 나타낼 수 있다.

<2> 인스턴트 메시징의 정황에서, 발행하는 사용자("발행자(publisher)")는 자신의 프레즌스 정보를 프레즌스 서비스에 제공하고 이 프레즌스 정보를 구독하는 사용자("구독자(subscribers)")에 제공할 수 있다. 따라서, 프레즌스 서비스는 구독자/발행자 모델을 이용하여, 프레즌스 서비스의 발행하는 사용자와 구독하는 사용자에 프레즌스 정보를 제공할 수 있다. 사용자의 프레즌스 정보가 바뀔 때마다, 그 사용자의 컴퓨터 시스템에 의해 그 변경사항이 프레즌스 서비스에게 통지되고, 이어서 구독하는 사용자에게 그 변경사항을 통지한다. 이후 구독하는 사용자는 의도되는 참여자의 프레즌스 정보에 기초하여 발행하는 사용자와 가장 잘 연락할 수 있는 방법을 결정할 수 있다. 예를 들면, 발행하는 사용자가 현재 화상 회의 중이라는 것을 프레즌스 정보가 나타낸다면, 구독하는 사용자는 발행하는 사용자에게 전화를 걸기 보다는 인스턴트 메시지를 전송하기로 할 수 있다. 그러나, 구독하는 사용자가 발행하는 사용자에게 전화를 걸어 통화할 필요가 있다면, 구독하는 사용자는 언제 전화를 걸 수 있을지를 알기 위해 발행하는 사용자의 프레즌스 정보를 모니터링 할 수 있다. 발행하는 사용자의 프레즌스 정보가 화상 회의가 끝났음을 나타낸다는 것을 구독하는 사용자가 알아차리면, 구독하는 사용자는 이제 전화를 걸 수 있다.

<3> 특정 사용자는 또한 다수의 장치로부터 프레즌스 정보를 발행할 수 있다. 예를 들면, 사용자는 랩톱 컴퓨터, 데스크톱 컴퓨터 및 셀룰러 폰의 프레즌스 서비스에 동시에 로그온될 수 있다. 프레즌스 정보는 각 장치상에서의 사용자의 상태를 나타낼 수 있다. 이것은, 예를 들면 사용자가 셀룰러 폰을 지닌 채 회의실로 걸어가고 있으며 자신의 데스크톱 컴퓨터로부터는 멀리 떨어져 있을 경우에 특히 유용하다. 구독하는 사용자가 발행하는 사용자와 연락하기를 원하는 경우, 이 프레즌스 정보는 셀룰러 폰으로 전화를 하는 것이 사용자의 데스크톱 컴퓨터에서 수신되는 인스턴트 메시지보다 이 사용자와 연락을 취하는 더 효과적인 방법이라는 것을 나타낼 수 있다. 사용자 장치들 각각은 또한 사용자의 다른 장치들 상에서의 프레즌스 정보를 구독할 수 있으며, 이는 발행하는 사용자가 현재 어느 장치에서 가장 활성인지를 장치들이 결정할 수 있게 한다.

<4> 인스턴트 메시징 시스템 및 기타 실시간 통신 시스템의 인기가 점점 증가하기 때문에, 프레즌스 서비스는 증가하는 사용자 수를 지원할 필요가 있다. 게다가, 이들 시스템은 "프레즌스 정보"의 점점 복잡해지는 정의를 지원할 필요도 있다. 예를 들면, 프레즌스 정보는 다음 여러 달 동안의 발행자의 가능함을 명시(specify)하는 달력 정보를 포함할 수 있다. 그 결과, 프레즌스 서비스는 통상적으로 효율적인 방식으로 프레즌스 서비스를 제공하도록 발전되었다. 불행히도, 통상적인 프레즌스 모델은 프레즌스 정보를 업데이트할 때 집중적인(intense) 계산을 필요로 한다. 따라서, 더 많은 발행자와 구독자가 추가됨에 따라, 필요한 계산을 수행하기 위해 추가의 프레즌스 서버가 필요하게 된다.

<5> 초기의 프레즌스 시스템은 모든 프레즌스 정보를 단일의 XML 문서에 둠(place)으로써 프레즌스 정보의 발행을 처리했다(handle). 사용자의 프레즌스 정보 중 일부가 변경되면, 발행자는 문서의 라인을 가리켜(point to) 그것을 새로운 프레즌스 정보로 대체하거나, 또는 XML 서브트리 내의 노드를 가리켜 그것을 또 다른 XML 서브트리로 대체해야 했다. 이러한 종류의 업데이트는 프레즌스 서버가 계산적으로 비용이 많이 드는 XML 파싱을 수행하는 것을 필요로 했다. 사용자와 프레즌스 업데이트의 수가 증가함에 따라, 프레즌스 서버에서의 XML 파싱의 부담(burden)이 압도적으로 되었다. 이러한 프레즌즈 정보 발행 모델로는 또한 구독자가 프레즌스 문서의 일부만을 구독하는 것이 불가능하다. 게다가, 문서의 각 부분은, 구독자 또는 프레즌스 서버가 문서의 어느 부분들이 최근에 변경되었는지를 쉽게 알(tell) 수 있도록 버전화되어 있지 않다. 이 모델은 또한 다른 구독자에게 프레즌스 문서의 다른 버전을 노출하거나 또는 문서의 각 부분에 대해 상이한 권한 정보를 명시하는 것을 허용하지 않는다.

<6> 계산적으로 비용이 좀 덜 드는 좀 더 효율적인 프레즌스 시스템이, 발명의 명칭이 "ORGANIZING PRESENCE INFORMATION INTO COLLECTIONS OF PUBLICATIONS"이고 미국 출원번호 제11/190,503호(대리인 정리 번호 41826-8225US)인 관련 출원에 설명되어 있으며, 이 관련 출원은 참조함으로써 그 전체가 본 명세서에 포함된다. 이 관련 출원은 발행의 모음(collections of publications)에 기초하여 프레즌스 정보를 관리하기 위한 방법 및 시스템을 설명한다. 프레즌스 모음 시스템은 발행자의 발행 모음을 생성한다. 발행은 유형과 값을 포함하며, 발행이 만료되어야만 하는 때와 같은 속성을 수반할 수 있다. 예를 들면, 유형 "상태" 및 값 "온라인"을 갖는 발

행은, 발행하는 사용자가 현재 온라인임을 구독하는 사용자에게 나타낼 수 있다. 발행자는 소정의 구독자에게 사용가능하게 될 발행 모음과, 다른 구독자들에게 사용가능하게 될 또 다른 발행 모음을 정의할 수 있다. 예를 들면, 하나의 모음은 대중(public)에게 보일 수 있는 반면, 또 다른 모음은 발행하는 사용자의 동료에게만 보일 수 있다. 각 모음은 그 모음 내의 정보를 구독하는 것이 허용되는 사용자의 리스트를 포함할 수 있다. 프레즌스 모음 시스템은 발행자가 모음 내의 각각의 발행을 업데이트하는 것을 허용할 수 있다. 업데이트를 수신하면, 프레즌스 서버는 계산적으로 비용이 많이 드는 XML 문서의 파싱에 의존할 필요 없이 신속하게 프레즌스 정보를 업데이트할 수 있다. 발행의 각 모음은 또한 정보의 카테고리를 포함할 수 있다. 카테고리는 다수의 모음에 이르는(span) 발행의 유형이다. 예를 들면, 카테고리 "상태"는, 대중에게 사용가능한 모음에서는 그 값이 "가능하지 않음"인 발행과, 동료에게 사용가능한 모음에서는 그 값이 "John과 회의중"인 발행을 지닐 수 있다. 발행은 또한 인스턴스 식별자를 명시할 수 있다. 인스턴스 식별자는 동일한 모음 내의 유사한 정보들과 구별짓게 한다. 예를 들면, 사용자는 자신의 상태가 랩탑에 있음을 나타내는 "랩탑" 인스턴스를 갖는 발행 "상태"를 지닐 수 있으며, 사용자의 상태가 셀룰러 폰에 있음을 나타내는 "셀룰러 폰" 인스턴스를 갖는 또 다른 발행 "상태"를 지닐 수 있다.

<7> 프레즌스 모음 시스템이 종래 기술에 비해 개선되었음에도 불구하고, 네트워크 자원과 프레즌스 서버에는 더 큰 부담이 있을 수 있다. 예를 들면, 많은 모음 내의 많은 발행 인스턴스에 대한 개별적인 업데이트를 프레즌스 서버로 전송하는 발행하는 사용자는 프레즌스 서버로의 많은 양의 네트워크 트래픽을 생기게 한다. 마찬가지로, 업데이트되는 각 발행에 대해 개별적인 통지를 수신하는 구독하는 사용자도, 네트워크 서버로부터 구독하는 사용자가 로그온된 장치 또는 장치들로의 많은 양의 네트워크 트래픽을 생기게 한다. 통상적으로, 프레즌스 발행은 정해진 시간 내에 만료하도록 설정되며, 그 발행이 발행된 상태를 계속 유지하기 위해서는 발행하는 사용자에 의해 반드시 주기적으로 리프레시(refresh)되어야만 한다. 변경되지 않았던 많은 발행을 리스레시해야 하는 의무(need)는 서버에 불필요한 부담을 준다. 마지막으로, 사용가능한 상세의 정도가 변하는 모음을 서로 다른 구독하는 사용자에게 추가하는 것은, 더 광범위한 구독자 그룹에 프레즌스 정보를 노출시키는 기회를 만든다. 한 모음으로 액세스하는 사용자의 리스트가 꽤 커질 수 있으며, 이 리스트가 증가함에 따라 특정 구독자가 어느 모음을 구독해야만 하는지를 점검하는 계산 비용이 증가하기만 하며, 이것은 프레즌스 서버에 추가의 부담을 줄 수 있다.

발명의 상세한 설명

<8> 발행의 모음을 갖는 프레즌스 서버의 계산 효율성 및 네트워크 효율성을 증가시키기 위한 방법 및 시스템이 제공된다. 프레즌스 시스템은, 비용이 많이 드는 처리를 필요로 하지 않으면서 풍부한 프레즌스 정보를 제공하는 것을 가능하게 하는 여러 기법들을 사용한다. 우선, 프레즌스 시스템은 단일의 프레즌스 업데이트 요청으로 발행 업데이트의 묶음(batches)을 받아들인다. 마찬가지로, 프레즌스 시스템은 구독 요청의 묶음을 받아들인다. 다음으로, 프레즌스 시스템은, 발행하는 사용자가 바뀌지 않은 프레즌스 정보를 계속해서 리프레시할 필요가 없게 하는, 발행 업데이트 요청에 대한 새로운 만료 유형을 지원한다. 마지막으로, 프레즌스 시스템은 특정 프레즌스 모음으로 액세스하고자 하는 사용자들의 개개 리스트보다는 멤버쉽 그룹을 포함하는 액세스 리스트를 받아들인다.

<9> 본 요약은 이하의 상세한 설명에서 더 설명될 일련의 개념을 단순화된 형태로 소개하기 위해 제공된다. 본 요약은 청구되는 내용의 주요 특징 또는 핵심 특징을 식별하고자 하는 것이 아니며, 청구되는 내용의 범위를 제한하기 위해 사용되고자 하는 것 또한 아니다.

실 시 예

<18> 발행의 모음을 갖는 프레즌스 서버의 계산 효율성 및 네트워크 효율성을 증가시키기 위한 방법 및 시스템이 제공된다. 프레즌스 시스템은, 비용이 많이 드는 처리를 필요로 하지 않으면서 풍부한 프레즌스 정보를 제공하는 것을 가능하게 하는 여러 기법들을 사용한다. 우선, 프레즌스 시스템은 단일의 프레즌스 업데이트 요청으로 발행 업데이트의 묶음을 받아들인다. 예를 들면, 발행하는 사용자는, 프레즌스 서버에 단일의 업데이트 요청을 전송함으로써 다수의 모음의 다수의 정보 카테고리를 발행할 수 있다. 마찬가지로, 프레즌스 시스템은 구독 요청의 묶음을 받아들인다. 예를 들면, 구독하는 사용자는, 단일의 구독 요청을 프레즌스 서버에게 전송함으로써 다수의 모음의 다수의 정보 카테고리에 대한 구독을 요청할 수 있다. 다음으로, 프레즌스 시스템은, 발행하는 사용자가 바뀌지 않은 프레즌스 정보를 계속해서 리프레시할 필요가 없게 하는 발행 업데이트 요청에 대한 새로운 만료 유형을 지원한다. 예를 들면, 발행하는 사용자의 집 전화 번호는 거의 바뀌지 않으므로, 프레즌스 시스템은 사용자가 이 정보를 한 번 발행한 후, 매 시간마다 한번씩 이 정보를 재발행하기보다는 이 정보가 바뀌

지 않는다면 업데이트하지 않는 것을 가능하게 한다. 마지막으로, 프레즌스 시스템은 특정 프레즌스 모음으로 액세스하고자 하는 사용자들의 개개 리스트보다는 멤버쉽 그룹을 포함하는 액세스 리스트를 받아들인다. 예를 들면, 발행하는 사용자는, 각 동료(coworker)를 개별적으로 리스트하지 않고 멤버쉽 그룹을 모음에 적용함으로써 자신의 동료 모두로의 액세스를 허가(grant)할 수 있다.

<19> 발행 배칭(publication batching)

<20> 한 실시예에서, 프레즌스 시스템은 단일의 요청으로 다수의 프레즌스 발행을 받아들인다. 발행은 다수의 카테고리, 인스턴스 및 모음에 이를 수 있다. 예를 들면, 발행 카테고리 "상태" 및 "위치"를 갖는 발행하는 사용자는, 대중에 액세스가능한 모음의 "가능하지 않음"이라는 상태 및 "마이크로소프트 본사"라는 위치, 그리고 동료들에 액세스가능한 모음의 "존과 회의중"이라는 상태 및 "빌딩 40/회의실 5"라는 위치를 발행할 수 있다. 발행하는 사용자는 또한 각 모음 내의 동일한 정보의 다수의 인스턴스를 발행할 수 있다. 예를 들면, 카테고리 "전화 번호"는 집 전화의 "집", 직장 전화의 "직장", 및 셀룰러 전화 번호의 "모바일"의 인스턴스들을 지닐 수 있다. 사용자는 동일한 발행 요청으로 이들 인스턴스 각각에 대한 값들을 발행할 수 있다.

<21> 한 실시예에서, 프레즌스 시스템은 단일의 요청으로 발행하지 않는 발행(unpublishing publication)과 발행하는 발행(publishing publication)을 받아들인다. 예를 들면, 회의에 참석할 예정인 발행하는 사용자는 자신의 데스크톱 컴퓨터에서 로그 오프할 수 있으며 이로 인해 데스크톱과 관련된 발행 인스턴스를 발행되지 않게 하며, 사용자는 자신의 셀룰러 폰에서 활성화될 수 있으며 이로 인해 셀룰러 폰에 대한 새로운 인스턴스가 추가될 필요가 있게 된다. 프레즌스 시스템은 단일의 업데이트 요청으로 데스크톱 발행의 비발행과 셀룰러 폰 발행의 발행 둘 다를 받아들일 수 있다. 예를 들면, 회사의 네트워크 프레즌스 시스템은 단일의 요청으로 이 두 장치들로부터 발행 요청을 받고 이 업데이트를 글로벌 프레즌스 서비스로 전달할 수 있다.

<22> 한 실시예에서, 프레즌스 시스템은, 세션 개시 프로토콜(Session Initiation Protocol:SIP)과 SIMPLE(SIP for Instant Messaging Presence Leveraging Extensions) 프로토콜에 대한 확장을 이용하여 다수의 프레즌스 업데이트를 전송한다. SIP는 애플리케이션-층 제어 프로토콜이며, 장치들은 이 SIP를 이용하여 장치들 서로를 발견하며, 장치들간에 세션을 확립하고, 수정하고, 종료할 수 있으며, SIP는 "RFC 3261"에서 기술된 인터넷 제안 표준이다. RFC 3261은 인터넷 www.ietf.org/rfc/rfc3261.txt에서 얻을 수 있으며, 참조함으로써 그 전체가 본 명세서에 포함된다. SIP 표준은 구현자가 커스텀 거동을 정의하는 확장(extension)을 추가하는 것을 가능하게 한다. 프레즌스 시스템은, 발행될 프레즌스 정보를 명시(specify)하는 <publications> 태그들의 리스트를 포함하는 태그 <publish> 및 발행되지 않을 프레즌스 정보를 명시하는 <publication> 태그들의 리스트를 포함하는, 유사한 <unpublish> 태그를 추가한다. 각각의 발행은 그것이 적용되는 카테고리, 인스턴스 및 모음을 식별한다. 일부 실시예에서, 프레즌스 시스템은 별도의 <unpublish> 태그를 갖기보다는 정보를 발행하지 않는다는 표시로서 <publish> 태그 내의 발행의 0이라는 만료 값(expiration value)을 받아들인다.

<23> 한 실시예에서, 프레즌스 시스템은 단일의 요청으로 다수의 발행하는 사용자에 대한 업데이트들을 받아들인다. 예를 들면, SIP 프로토콜이 사용되는 경우, 각 발행은 그 발행이 적용되는 사용자의 URI를 포함할 수 있다. 이것은 서비스들이 많은 수의 사용자를 대신하여 정보를 발행하는 것을 가능하게 한다. 예를 들면, 익스체인지 이메일 서버는 그 서버를 사용하는 사용자들 모두를 위해 달력 정보를 발행할 수 있다. 또 다른 예로서, 셀룰러 폰 사업자는 셀룰러 폰 고객들의 위치에 대한 정보를 프레즌스 서버에 발행할 수 있다. 이들 두 가지 예 모두에서, 이전에는 대규모의 네트워크 트래픽을 발생시켰던 많은 수의 사용자에 대한 정보가 단일 요청으로 프레즌스 서버에 발행될 수 있다.

<24> 한 실시예에서, 프레즌스 시스템은 단일의 요청으로 서로 다른 속성들을 갖는 발행 요청들을 받아들인다. 발행하는 사용자는 한 모음의 발행에 대해 또 다른 모음의 발행과는 상이한 만료 정책(expiration policy)을 명시할 수 있다. 예를 들면, 발행 "위치"에 대해 상세한 정보를 갖는 모음에서, 상세한 상태 정보(예를 들면, "회의실 2에 있음")는 자주 바뀌고 또 다른 모음의 덜 상세한 정보(예를 들면, "시애틀에 있음")보다 더 빨리 만료될 수 있다. 또 다른 예로서, 발행하는 사용자는 발행의 한 유형에 대해 동일한 요청의 또 다른 유형과는 상이한 통지 정책을 명시할 수 있다. 예를 들면, 통지 정책은, 소정의 카테고리에서의 업데이트들은 통지를 전혀 생성해서는 안 된다는 것을 나타낼 수도 있다.

<25> 구독 배칭(subscription batching)

<26> 한 실시예에서, 프레즌스 시스템은 단일의 요청으로 다수의 카테고리에 대한, 한 구독하는 사용자로부터의 구독을 받아들인다. 예를 들면, 발행하는 사용자의 상태와 위치에 관심이 있는 구독하는 사용자는 카테고리 리스트

에 "상태"와 "위치"를 포함하는 단일의 구독 요청을 포함할 수 있다.

<27> 한 실시예에서, 프레즌스 시스템은 단일의 요청으로 다수의 발행하는 사용자에 대한, 구독하는 사용자로부터의 구독을 받아들인다. 예를 들면, 구독하는 사용자는 단일의 요청으로 사용자 A와 사용자 B에 대한 다수의 카테고리를 구독할 수 있다. 구독하는 사용자가 자신의 연락처 리스트에 있는 사용자들 각각에 대한 프레즌스 정보를 구독하고 있는 경우, 모든 구독이 단일의 요청으로 행해질 수 있기 때문에, 이것은 상당한 자원을 절약할 수 있게 한다. 마찬가지로, 프레즌스 시스템은 또한 단일의 요청으로 서로 다른 사용자들에 대한 서로 다른 카테고리를 받아들일 수 있다. 예를 들면, 구독하는 사용자는 단일의 요청으로 사용자 A에 대한 카테고리 1 및 2와, 사용자 B에 대한 카테고리 3 및 4를 구독할 수 있다.

<28> 한 실시예에서, 프레즌스 시스템은 구독 요청이 구독을 생성하지 않고 프레즌스 정보를 검색(retrieve)하는 것을 가능하게 한다. 예를 들면, 사용자는 정보가 변경될 때 통지를 수신하지 않고 발행하는 사용자에 대한 프레즌스 정보를 한 번 요청하기를 원할 수 있다. 사용자는 이 요청이 명시된 카테고리의 프레즌스 정보의 현재 값에 대한 질의이지 구독을 생성하기 위한 요청이 아니라는 지시와 함께 구독 요청을 전송할 수 있다.

<29> 한 실시예에서, 프레즌스 시스템은 새 구독을 추가할 때 동일한 요청에 있는 기존의 구독의 제거를 받아들인다. 예를 들면, 구독하는 사용자가 발행하는 사용자의 위치를 아는 것에는 더 이상 관심이 없고 발행하는 사용자의 현재 전화 번호를 추적하기를 원한다면, 구독하는 사용자는, 구독하지 않는 카테고리의 리스트에 카테고리 "상태"를 포함하고 구독하는 카테고리의 리스트에 카테고리 "전화 번호"를 포함하는, 단일의 요청을 구성할 수 있다.

<30> 한 실시예에서, 프레즌스 시스템은 단일 요청으로 다수의 구독하는 사용자를 대신하여 구독 요청을 받아들인다. 예를 들면, 회사는 그 회사 내의 구독하는 사용자들로부터의 개개의 구독 요청을 수신하여 단일의 배치 구독 요청으로서 이 요청들을 글로벌 프레즌스 서버로 전달하는 로컬 프레즌스 서버를 지닐 수 있다.

<31> 한 실시예에서, 프레즌스 시스템은 SIP 및 SIMPLE 프로토콜에 대한 확장을 이용하여 다수의 프레즌스 구독을 전송한다. 프레즌스 시스템은 구독할 카테고리의 리스트를 명시하는 <categoryList> 태그를 포함하는 태그 <batchSub>를 추가한다. <batchSub> 태그는 또한 카테고리가 구독되고 있는 다수의 사용자의 리스트를 명시하는 <adhocList> 태그를 명시할 수 있다. 프레즌스 시스템은 또한 구독하지 않고 있는 사용자 및 그 사용자의 카테고리의 리스트를 명시하는 태그를 또한 포함하는 <batchUnsub> 태그를 추가한다. 시스템은 또한 빈 본문(empty body)을 지니는 구독 요청이, 구독하는 사용자가 자신이 구독해 왔던 모든 카테고리 및 사용자를 리스트하기보다는 구독하지 않아야 하는 모든 사용자와 카테고리를 나타내게 할 수 있다.

<32> 한 실시예에서, 프레즌스 시스템은 구독하는 사용자가 관심이 있는 모든 카테고리에 대한 명시적인 구독을 요구함으로써 구독하는 사용자를 위해 원치 않는 프레즌스 정보를 필터링할 수 있다. 예를 들면, 발행하는 사용자는 수백 가지 카테고리의 정보를 발행할 수 있지만, 구독하는 사용자는 발행하는 사용자의 현재 상태만을 알기를 원할 수 있다. 구독하는 사용자는 카테고리 "상태"의 구독을 나타내는 구독 요청을 전송할 수 있으며, 사용자는 그 카테고리에 대한 통지만을 수신할 것이다. 이것은 프레즌스 서버에 의해 제공되는 정보의 양이 증가함에 따라, 구독하는 사용자 자신이 관심이 없는 통지로 인해 압도되는(overwhelmed) 것을 방지한다.

<33> 만료 모델

<34> 한 실시예에서, 프레즌스 시스템은 발행하는 장치가 프레즌스 서버에서 로그 오프할 때 만료되는 발행을 받아들인다. 이전의 시스템에서, 발행하는 사용자는 정해진 지속 시간(예를 들면 한 시간) 동안 정보를 발행하고, 이 정보가 계속 발행된 상태를 유지하기 위해 발행의 만료가 가까워지면 그 정보를 리프레시하거나 또는 재발행해야 했다. 프레즌스 시스템은 그 정보를 발행했던 장치가 오프라인이 될 때까지 프레즌스 정보가 계속 발행된 상태를 유지할 수 있게 하는 새로운 만료 유형을 정의한다. 이것은, 예를 들면, 셀룰러 폰과 같은 장치가 그 셀룰러 폰이 온라인인 동안만 유효한 정보의 인스턴스를 발행했을 때 사용된다. 셀룰러 폰이 턴 오프 또는 오프라인이 되면, 프레즌스 서버는 그것을 탐지하여 그 장치가 온라인인 동안에만 관련되는 정보를 제거할 수 있다. 장치가 온라인일 때, 프레즌스 정보가 만료되는 것을 막기 위해 변경되지 않았던 프레즌스 정보를 계속 업데이트하는 것이 더 이상 필요하지 않다.

<35> 한 실시예에서, 프레즌스 시스템은 발행하는 사용자가 더 이상 온라인이 아닐 때 만료되는 발행을 받아들인다. 이 만료 유형은 사용자가 온라인인 동안에는 다수의 장치에 적용되나 사용자가 오프라인일 때에는 더 이상 관련되지 않는 프레즌스 정보에 유용하다. 프레즌스 시스템은 임의의 장치를 통해 사용자가 더 이상 로그온 되어 있지 않을 때 사용자가 오프라인임을 탐지할 수 있다. 프레즌스 시스템은, 장치가 프레즌스 서버와 접속을 끊

거나 또는 사용자가 더 이상 장치를 사용하지 않는다는 것을 장치가 프레즌스 서버에 알릴 때, 사용자가 더 이상 그 장치에 로그온 되어 있지 않다는 것을 탐지할 수 있다. 예를 들면, 공용의 인터넷 키오스크는, 사용자가 자신을 식별하는 USB 동글(dongle)을 삽입함으로써 인터넷에 액세스할 수 있게 할 수 있다. 이 동글이 제거될 때, 인터넷 키오스크는, 프레즌스 서비스와 같은 사용자가 접속했던 임의의 서비스에게 그 사용자가 더 이상 그 장치를 사용하지 않음을 알릴 수 있다. 사용자가 온라인인 동안 그 정보는 사용자로부터의 리프레시 요청이 없어도 계속 발행된 상태로 남아있으므로, 이것은 프레즌스 서버가 추가의 업데이트 요청을 처리하는 것을 면하게 해 줄 수 있다.

<36> 한 실시예에서, 프레즌스 시스템은 만료하지 않는 발행을 받아들인다. 정적(static)이라 불리는 이 만료 유형은, 사용자의 전화 번호 또는 주소와 같이 사용자가 온라인인지 아닌지의 여부에 관련된 정보에 유용하다. 이 만료 유형은 또한 사용자를 대신하여 정보를 발행하는 서비스에 의해 사용될 수 있다. 예를 들면, 다수의 사용자를 위해 달력 정보를 발행하는 익스체인지 서버는 사용자의 달력 정보를 발행하기 전에 그 사용자가 온라인인지 아닌지의 여부에 관해 알 필요가 없고, 그 정보는 사용자의 현재 로그온 상태에 상관없이 만료되어서는 안된다. 그러므로, 이러한 서비스는 발행에 대해 정적 만료 유형을 명시함으로써 정보를 발행할 수 있고, 프레즌스 서버가 명시적인 비발행 요청을 수신하지 않는다면 발행된 정보는 비발행되지 않을 것이다.

<37> 멤버쉽 그룹

<38> 한 실시예에서, 프레즌스 시스템은 사용자 그룹을 포함하는 모음 멤버쉽 리스트를 받아들인다. 이전에는, 프레즌스 정보의 특정 모음에 액세스하는 사용자가 명시적으로 리스트되었으며, 디폴트 모음은 그 어느 리스트에도 없는 사용자에 적용되는 정보를 포함하였다. 프레즌스 시스템은 사용자 그룹이 명시되는 것을 가능하게 하며, 이것은 사용자가 큰 사용자 그룹에 대한 모음으로 액세스할 수 있는지의 여부를 결정하는 계산적 곤란함을 줄여준다. 모음 멤버쉽 리스트는 프레즌스 정보가 발행될 때 명시될 수 있고, 또는 별도의 요청으로 독립적으로 명시될 수 있다. 멤버쉽 그룹의 한 유형인 "연락처 리스트"는 발행하는 사용자의 연락처 리스트에 있는 임의의 구독하는 사용자가 그 그룹에 할당된 모음에 있는 정보를 볼 수 있다는 것을 명시한다.

<39> 한 실시예에서, 프레즌스 시스템은 발행하는 사용자와 동일한 회사 내의 사용자들을 포함하는 멤버쉽 그룹을 받아들인다. 이 멤버쉽 그룹은, 사용자의 모음 멤버쉽 리스트 내의 단일 엔트리가 동일한 회사의 임의의 구독하는 사용자가 특정 모음으로 액세스할 수 있어야 함을 명시하는 것을 가능하게 한다. 예를 들면, 마이크로소프트와 같은 큰 회사의 경우, 이것은 모음이, "동일한 회사" 멤버쉽 그룹을 명시하는 단일 엔트리를 멤버쉽 리스트 내에 포함하는 동료들에 대한 추가의 상세사항으로 생성되는 것을 가능하게 한다. 프레즌스 서버는, 외부 도메인 컨트롤러에 질의함으로써와 같이 사용자의 회사가 발행하는 사용자와 동일한지의 여부를 점검함으로써, 특정의 구독하는 사용자가 모음으로 액세스할 수 있는지의 여부를 판정한다. 이것은 프레즌스 서버가 외부 도메인 컨트롤러가 이미 포함하고 있는 것과 동일한 사용자의 리스트를 복제하는 것을 방지한다.

<40> 한 실시예에서, 프레즌스 시스템은 공중 클라우드(public cloud) 내의 사용자를 포함하는 멤버쉽 그룹을 받아들인다. 공중 클라우드는 다른 조직에 의해 제공되는 외부에서 가능한 사용자 리스트로서 정의된다. 공중 클라우드 멤버쉽 그룹은 프레즌스 시스템의 한 조작자가 다른 프레즌스 시스템의 조작자들과 파트너쉽을 지닐 때 유용하다. 예를 들면, 마이크로소프트 사의 인스턴트 메시징 소프트웨어를 이용하는 프레즌스 서버가 AOL, 야후, 및 공중 클라우드 멤버쉽 그룹을 이용하는 다른 인스턴트 메시지 플랫폼과 인터랙트할 수 있다. 모음 멤버쉽 리스트에 공중 클라우드 멤버쉽 그룹을 명시함으로써, 발행하는 사용자는 모음 멤버 리스트에 명시적으로 각 멤버들을 리스트할 필요 없이 다른 프레즌스 시스템의 멤버가 구독하고 볼 수 있는 프레즌스 발행을 정의할 수 있다.

<41> 한 실시예에서, 프레즌스 시스템은 연합된 멤버쉽 그룹(federated membership group)을 받아들인다. 연합된 멤버쉽 그룹은, 그 연합된 멤버쉽 그룹에서 다수의 엔티티가 프레즌스 정보로 유사하게 액세스해야만 하는 사용자들의 리스트를 정의하는 그러한 것이다. 예를 들면, 마이크로소프트사와 인텔사와 같은 두 회사는 연합된 멤버쉽 그룹에서 사용가능한 그 피고용인 리스트를 작성할 수 있다. 연합된 멤버쉽 그룹 유형이 있는 발행을 수신하는 프레즌스 서버는 명시된 연합 멤버쉽 서버를 참조하여 어느 사용자가 연합된 그룹의 멤버인지를 판정한다. 이후 발행하는 사용자는 연합 그룹이 구독할 수 있는 발행을 명시할 수 있다.

<42> 한 실시예에서, 프레즌스 시스템은 단일 요청으로 멤버쉽 리스트 변경의 묶음을 받아들인다. 예를 들면, 발행하는 사용자는 단일의 요청으로 동일한 회사의 멤버쉽 그룹의 구독하는 사용자에게 한 모음으로의 액세스를 허가하고, 공용 클라우드 멤버쉽 그룹의 사용자들을 그 모음에서 제거시킬 수 있다. 발행하는 사용자는 단일의 요청으로 다수 모음의 멤버쉽 리스트 뿐만 아니라 특정 모음의 멤버쉽 리스트에 있는 다수의 멤버쉽 그룹 및 개

개의 사용자를 명시할 수 있다.

<43> 모음 모델과 멤버쉽 그룹을 조합함으로써, 발행하는 사용자는 프레즌스 정보 액세스의 계층 구조를 확립할 수 있다. 예를 들면, 사용자의 달력 정보는 동일한 회사 내의 동료들에게 상세하게 사용가능하도록 만들어질 수 있으며, 일반 대중에게는 한가한/바쁜 정보(사용자가 한가한 시간과 사용자가 바쁜 시간만을 나타냄)로서 사용가능하게 만들어질 수 있다. 이것은, 일반 대중의 멤버가, 발행하는 사용자가 하고 있는 것에 관한 개인적인 정보를 노출시키지 않고 그 사용자의 나머지 스케줄과 겹치지 않게 발행하는 사용자와 약속을 잡는 것을 가능하게 한다. 한편, 예를 들면, 사용자가 오늘 나중에 동료와 회의에 참가할 것이라는 것을 동료들이 알 수 있도록, 사용자가 하고 있는 것에 관해 좀 더 상세한 뷰가 동료들에게 제공된다.

<44> 도 1은 한 실시예에서 프레즌스 시스템의 컴포넌트들을 도시하는 블록도이다. 프레즌스 시스템(110)은 인터넷과 같은 통신 링크(140)를 통해 엔티티 장치(120 및 130)에 접속되어 있다. 엔티티 장치는 발행자 또는 구독자일 수 있는 엔티티에 대응한다. 엔티티 장치는 발행 업데이트 구성 컴포넌트(compose publication update component)(122), 구독 요청 구성 컴포넌트(compose subscription request component)(125) 및 요청 전송 컴포넌트(send request component)(127)를 포함한다. 발행 업데이트 구성 컴포넌트(122)는, 개개의 발행 업데이트의 묶음을 포함할 수 있는 발행된 프레즌스 정보에 대한 업데이트 요청을 구성한다. 구독 요청 구성 컴포넌트(125)는, 다수의 발행하는 사용자의 다수의 프레즌스 정보 카테고리에 대한 구독의 묶음을 포함할 수 있는 구독 요청을 구성한다. 요청 전송 컴포넌트(127)는 발행 업데이트 요청 및 구독 요청을 프레즌스 시스템(110)에 전송한다.

<45> 프레즌스 시스템은 발행 업데이트 수신 컴포넌트(receive publication update component)(150), 발행 업데이트 컴포넌트(update publications component)(155), 멤버쉽 업데이트 수신 컴포넌트(receive membership update component)(170), 멤버쉽 업데이트 컴포넌트(update memberships component)(175), 발행 만료 컴포넌트(expire publications component)(180), 접속 상태 탐지 컴포넌트(detect connect status component)(182), 구독 요청 수신 컴포넌트(receive subscription request component)(185), 구독 업데이트 컴포넌트(update subscriptions component)(190), 모음 생성 컴포넌트(create collection component)(165), 발행 저장 컴포넌트(publication store component)(160) 및 구독 저장 컴포넌트(subscription store component)(195)를 포함한다. 발행 저장 컴포넌트(160)는 발행자의 발행을 포함한다. 발행은 모음 생성 컴포넌트(165)에 의해 생성된 모음들로 정리되어(organized) 있다. 발행 업데이트 수신 컴포넌트(150)는 발행자로부터 하나 이상의 발행을 업데이트하고자 하는 요청을 수신했을 때 인보크된다. 발행 업데이트 수신 컴포넌트(150)는 발행 업데이트 컴포넌트(155)를 인보크하여, 요청에 의해 명시된 바대로 발행 저장소(160)에 발행을 추가하고, 발행 저장소(160)의 발행을 제거하고 또는 업데이트한다. 멤버쉽 업데이트 수신 컴포넌트(170)는 발행자로부터 발행 저장소(160)의 하나 이상의 모음의 멤버쉽 리스트를 업데이트하고자 하는 요청을 수신했을 때 인보크된다. 이 요청은 독립적인 메시지로 수신될 수 있고 또는 발행 업데이트 요청의 일부일 수 있다. 멤버쉽 업데이트 수신 컴포넌트(170)는 멤버쉽 업데이트 컴포넌트(175)를 인보크하여, 발행 저장소(160)의 모음에 대한 멤버쉽을 추가하고, 제거하고 또는 업데이트한다. 구독 요청 수신 컴포넌트(185)는 발행자의 하나 이상의 유형의 발행을 한 엔티티로부터 구독하고자 하는 요청을 수신했을 때 인보크된다. 구독 요청 수신 컴포넌트(185)는 구독 업데이트 컴포넌트(190)를 인보크하여 그 엔티티가 요청된 유형의 발행을 구독할 수 있게 한다. 구독은 발행 저장소(160)와 통합될 수 있는 구독 저장소(195)에 저장된다. 발행 만료 컴포넌트(180)는, 발행 저장소(160)에 있는 만료된 발행을 정리하기(clean up) 위해 프레즌스 시스템에 의해 주기적으로 인보크된다. 도 1에 도시되지는 않았지만, 엔티티 장치는, 모음과 그 멤버쉽을 정의하고, 발행 업데이트를 전송하고, 구독 요청을 전송하고, 발행에 대한 업데이트의 통지를 수신하는 프레즌스 시스템의 컴포넌트를 포함한다.

<46> 본 시스템이 구현된 컴퓨팅 장치는 중앙 처리 장치, 메모리, 입력 장치(예를 들면, 키보드 및 포인팅 장치), 출력 장치(예를 들면, 디스플레이 장치) 및 저장 장치(예를 들면, 디스크 드라이브)를 포함할 수 있다. 메모리와 저장 장치들은 본 시스템을 구현하는 명령어를 포함할 수 있는 컴퓨터 판독가능 매체이다. 게다가, 데이터 구조 및 메시지 구조는, 통신 링크 상의 신호와 같이 데이터 전송 매체를 통해 저장되거나 또는 전송될 수 있다. 인터넷, LAN, WAN, 포인트-대-포인트 다이얼 업 접속, 셀룰러 폰 네트워크 등과 같은 각종 통신 링크가 사용될 수 있다.

<47> 본 시스템의 실시예는 퍼스널 컴퓨터, 서버 컴퓨터, 핸드-헬드 또는 랩톱 장치, 멀티프로세서 시스템, 마이크로프로세서 기반 시스템, 프로그램가능한 가전제품, 디지털 카메라, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 상기 시스템들이나 장치들 중 임의의 것을 포함하는 분산 컴퓨팅 환경, 기타 등등을 포함하는 각종 운영 환경에서 구현될 수 있다. 컴퓨터 시스템은 셀룰러 폰, PDA, 스마트 폰, 퍼스널 컴퓨터, 프로그램가능한 가

전 제품, 디지털 카메라 등이 될 수 있다.

<48> 본 시스템은 일반적으로 하나 이상의 컴퓨터 또는 기타 장치에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행가능 명령어와 관련하여 기술될 것이다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 개체, 컴포넌트, 데이터 구조 등을 포함한다. 통상적으로, 프로그램 모듈의 기능은 각종 실시에에서 원하는 바에 따라 조합되거나 또는 분산될 수 있다.

<49> 도 2는 한 실시예에서 발행 업데이트를 전송하기 위한 SIP 프로토콜 확장을 도시하는 프로토콜 리스팅이다. 프로토콜은 정보를 발행하고자 하는 요청을 시그널링하는 SIP 동사 "PUBLISH"(210)로 시작한다. 프로토콜은 그 컨텐츠 유형과 같이 요청을 더 설명할 헤더(220)를 포함한다. 프로토콜은 "<publications>" 섹션(232)을 포함하는 "<publish>" 섹션(230)을 포함한다. "<publications>" 섹션은 발행될 발행(235 및 240)과, 발행되지 않을 발행(250)을 리스트한다. 발행(235)은 "status" 라 명명되는 카테고리, "laptop"이라 명명되는 카테고리의 특정 인스턴스, "1"로서 식별되는 (모음의 한 유형인) 컨테이너, "0"인 버전, SIP 어드레스 "sip:foo@bar.com"에 의해 식별되는 발행하는 사용자 및, 사용자가 오프라인으로 될 때 발행이 제거되어야 함을 나타내는 "user"라고 하는 발행의 만료 유형에 대한 프레즌스 정보를 명시한다. 명확하게 하기 위해, 일부 필드에서는 텍스트가 도시되었지만 숫자가 또한 사용될 수 있다. 예를 들면, "instance" 필드의 값은 텍스트 "laptop" 보다는 장치에 할당된 고유의 식별자를 포함할 수 있다. 또 다른 발행(240)은 동일한 카테고리 "status"를 명시하지만, 동일한 발행하는 사용자에 대해 상이한 컨테이너 "2"의 상이한 인스턴스 "desktop"을 명시한다. 프로토콜은 또한 만료 유형이 0임에 의해 본 예에서 나타나는 바와 같이 발행 저장소에서 제거될 발행(250)을 포함한다. 이 예에서는 발행이 발행되지 않을 것이라는 것을 나타내기 위해 "expireType" 필드가 오버로드되었지만, 프로토콜 내의 별도 필드 또는 별도의 XML 섹션이 또한 사용될 수 있다. 제거될 발행(250)은 카테고리 이름, 인스턴스 식별자, 컨테이너 식별자, 및 발행하는 사용자 주소에 의해 식별된다. 발행 내에 발행하는 사용자의 어드레스를 명시하는 것은, 사용자 외의 엔티티가 상술된 익스체인지 서버와 같이 사용자를 대신해서 프레즌스 정보를 발행하거나 발행하지 않는 것을 가능하게 한다.

<50> 도 3은 한 실시예에서 프레즌스 시스템의 발행 업데이트 컴포넌트의 처리를 도시하는 흐름도이다. 컴포넌트는 도 2에 있는 것과 같은 발행 요청이 수신될 때 발행 업데이트 수신 컴포넌트에 의해 인보크된다. 블록(310)에서, 컴포넌트는 발행 업데이트 수신 컴포넌트로부터 발행 업데이트 요청을 수신한다. 블록(320)에서, 컴포넌트는 요청에서 다음 발행을 선택한다. 블록(330)에서, 컴포넌트는 발행 저장소에서 선택된 발행에 의해 식별되는 발행하는 사용자에 대한 모음을 찾는다(locate). 판정 블록(340)에서, 발행이 존재하면, 컴포넌트는 블록(360)으로 계속되고, 그렇지 않을 경우 컴포넌트는 블록(350)으로 계속된다. 블록(350)에서, 이전에 존재하지 않았던 발행이 발행 저장소에 추가된다. 블록(360)에서, 발행 저장소에 이미 존재하는 발행은 요청의 정보로 업데이트된다. 판정 블록(370)에서, 요청에 발행이 더 있다면, 컴포넌트는 블록(320)으로 루핑하여 다음 발행을 선택하고, 그렇지 않으면 컴포넌트는 완료된다.

<51> 도 4는 한 실시예에서 구독 요청을 전송하기 위한 SIP 프로토콜 확장을 도시하는 프로토콜 리스팅이다. 프로토콜은 발행된 정보를 구독하고자 하는 요청을 시그널링하는 SIP 동사 "SUBSCRIBE"(410)로 시작한다. 이 프로토콜은 그 컨텐츠 유형과 같이 요청을 더 설명할 헤더(420)를 포함한다. 프로토콜은 또한 "<action>" 섹션(442)을 포함하는 "<batchSub>" 섹션(440)을 포함한다. "<action>" 섹션은 "<adhocList>" 섹션(445)과 "<categoryList>" 섹션(450)을 포함한다. "<adhocList>" 섹션(445)은 발행하는 사용자의 식별자를 명시하고 있는데, 구독하는 사용자는 이 발행하는 사용자의 발행에 대한 통지를 수신하기를 원한다. "<categoryList>" 섹션(450)은 발행하는 사용자 각각에 대해 구독하는 카테고리를 명시한다. 구독 요청은, 구독하는 사용자가 명시된 사용자 각각에 대해 명시된 카테고리 각각에 대한 통지를 수신하게 할 것이다. 도 4에 도시되지는 않았지만, 이 요청은 subscribe 섹션(440)과 유사한, 구독하지 않는 카테고리 및 사용자의 리스트를 또한 포함할 수 있다.

<52> 도 5는 한 실시예에서 프레즌스 시스템의 구독 업데이트 컴포넌트의 처리를 도시하는 흐름도이다. 컴포넌트는 도 4에 있는 것과 같은 구독 요청이 수신될 때 구독 업데이트 수신 컴포넌트에 의해 인보크된다. 블록(510)에서, 컴포넌트는 구독 요청 수신 컴포넌트로부터 구독 요청을 수신한다. 판정 블록(520)에서, 요청에 포함된 카테고리 필터가 더 있다면 이 컴포넌트는 블록(530)으로 계속되고, 그렇지 않으면 이 컴포넌트는 블록(550)으로 계속된다. 블록(530)에서, 컴포넌트는 요청에서 다음 카테고리 필터를 선택한다. 블록(540)에서, 컴포넌트는 구독 저장소의 카테고리 필터를 업데이트한다. 이후, 컴포넌트는 블록(520)으로 루핑하여 요청에 카테고리 필터가 더 있는지의 여부를 판정한다. 판정 블록(550)에서, 요청에 구독이 더 있다면, 컴포넌트는 블록(560)으로 계속되고 그렇지 않을 경우 컴포넌트는 블록(580)으로 계속된다. 블록(560)에서, 컴포넌트는 구독 요청에서 다

음 구독을 선택한다. 블록(570)에서, 컴포넌트는 구독을 구독 저장소에 추가한다. 구독은 사용자들의 리스트와 카테고리들의 리스트의 형태일 수 있어, 단일의 구독 섹션으로 다수의 카테고리가 다수의 사용자에게 구독될 수 있다. 이후 컴포넌트는 블록(550)으로 루핑하여 요청에 추가의 구독이 있는지의 여부를 판정한다. 판정 블록(580)에서, 요청에 비구독(unsubscription)이 더 있다면, 컴포넌트는 블록(590)으로 계속되고, 그렇지 않을 경우 컴포넌트는 완료된다. 블록(590)에서, 컴포넌트는 요청에서 다음 비구독을 선택한다. 블록(595)에서, 컴포넌트는 구독 저장소에서 명시된 구독을 제거한다. 모든 비구독이 처리된 후, 컴포넌트는 완료된다.

<53> 도 6은 한 실시예에서 모음 멤버쉽을 업데이트하기 위한 SIP 프로토콜 확장을 도시하는 프로토콜 리스팅이다. 도시된 프로토콜의 일부는 도 2에 도시된 SIP 발행 요청의 본문에 포함될 수 있으며, 또는 이것은 특별히 멤버쉽을 업데이트하기 위한 독립적인 메시지 유형의 일부일 수도 있다. 프로토콜은 멤버쉽이 수정될 모음을 식별하는 하나 이상의 "<container>" 섹션(615)을 포함하는 "<setContainerMembers>" 섹션(610)으로 시작한다. 각 컨테이너 섹션은 하나 이상의 멤버 엔트리(620, 630 및 640)를 포함한다. 도 6의 제1 멤버 엔트리(620)는 그 값이 특정 사용자의 SIP URI임을 나타내는 유형 "uri"를 명시한다. 액션 유형 "add"는 명시된 사용자가 명시된 모음으로 액세스하는 멤버로서 추가될 것이라는 것을 나타낸다. 제2 멤버 엔트리(630)는 발행하는 사용자와 동일한 도메인으로부터의 사용자의 멤버쉽이 수정될 것임을 나타내는 유형 "sameDomain"을 명시하며, 액션 유형 "remove"는 그 모음으로의 액세스가 제거될 것임을 나타낸다. 제3 멤버 엔트리(640)는 명시된 모음으로의 액세스가 발행하는 사용자의 연락처 리스트의 임의의 멤버에게 허가될 것임을 나타내는 유형 "contacList"와 액션 유형 "add"를 명시한다.

<54> 도 7은 한 실시예에서 프레즌스 시스템의 멤버쉽 업데이트 컴포넌트의 처리를 도시하는 흐름도이다. 이 컴포넌트는 도 6에 도시된 것과 같은 멤버쉽 업데이트 요청이 수신될 때 멤버쉽 업데이트 수신 컴포넌트에 의해 인보크된다. 블록(710)에서, 컴포넌트는 멤버쉽 업데이트 수신 컴포넌트로부터 멤버쉽 업데이트 요청을 수신한다. 블록(720)에서, 컴포넌트는 요청에서 다음 컨테이너 섹션을 선택한다. 블록(730)에서, 컴포넌트는 컨테이너 섹션 내의 다음 멤버 업데이트를 선택한다. 판정 블록(740)에서, 멤버 업데이트가 액션 유형 "add"를 포함한다면, 컴포넌트는 블록(750)으로 진행되고, 그렇지 않으면 컴포넌트는 블록(760)으로 진행한다. 블록(750)에서, 멤버 업데이트에 명시된 사용자 또는 그룹이 발행 저장소의 선택된 컨테이너의 멤버 리스트에 추가된다. 블록(760)에서, 멤버 업데이트에 명시된 사용자 또는 그룹이 발행 저장소의 선택된 컨테이너의 멤버 리스트에서 제거된다. 판정 블록(770)에서, 선택된 컨테이너 섹션에 멤버 업데이트가 더 있다면, 컴포넌트는 블록(730)으로 루핑하여 다음 멤버 업데이트를 선택하고, 그것이 아니라면 컴포넌트는 블록(780)으로 계속된다. 판정 블록(780)에서, 멤버쉽 업데이트 요청에 컨테이너 섹션이 더 있다면, 컴포넌트는 블록(720)으로 루핑하여 다음 컨테이너 섹션을 선택하고, 그렇지 않은 경우 컴포넌트는 완료된다.

<55> 도 8은 한 실시예에서 프레즌스 시스템의 발행 만료 컴포넌트의 처리를 도시하는 흐름도이다. 컴포넌트는 발행 저장소에서 실효하지 않은(stale) 발행을 제거하기 위해 프레즌스 시스템에 의해 주기적으로 인보크된다. 블록(810)에서, 컴포넌트는 발행 저장소에서 다음 발행을 선택한다. 판정 블록(820)에서, 선택된 발행의 만료 유형이 "static"인 경우, 컴포넌트는 블록(860)으로 계속되고, 그렇지 않은 경우 컴포넌트는 블록(830)으로 계속된다. 판정 블록(830)에서, 선택된 발행의 만료 유형이 "device"인 경우, 컴포넌트는 블록(833)으로 계속되고, 그렇지 않을 경우 블록(840)으로 계속된다. 판정 블록(833)에서, 선택된 발행을 발행한 장치가 오프라인이면, 컴포넌트는 블록(836)으로 계속되고, 그렇지 않을 경우 블록(860)으로 계속된다. 블록(836)에서, 컴포넌트는 발행 저장소에서 만료된 발행을 제거한다. 판정 블록(840)에서, 선택된 발행의 만료 유형이 "user"인 경우, 컴포넌트는 블록(843)으로 계속되고, 그렇지 않은 경우 컴포넌트는 블록(850)으로 계속된다. 판정 블록(843)에서, 선택된 발행을 발행했던 사용자가 임의의 장치를 통해 로그온되어 있지 않으면, 컴포넌트는 블록(846)으로 계속되고, 그렇지 않으면 컴포넌트는 블록(860)으로 계속된다. 블록(846)에서, 컴포넌트는 발행 저장소에서 만료된 발행을 제거한다. 판정 블록(850)에서, 선택된 발행의 만료 유형이 "duration"인 경우, 컴포넌트는 블록(853)으로 계속되고, 그렇지 않을 경우 블록(860)으로 계속된다. 판정 블록(853)에서, 선택된 발행에 대해 명시된 시간이 만료되었다면, 컴포넌트는 블록(856)으로 계속되고, 그렇지 않으면 블록(860)으로 계속된다. 블록(856)에서, 컴포넌트는 발행 저장소에서 만료된 발행을 제거한다. 판정 블록(860)에서, 발행 저장소에 발행이 더 있다면, 컴포넌트는 블록(810)으로 루핑하여 다음 발행을 선택하고, 그렇지 않으면 컴포넌트는 완료된다. 도시된 방법이 만료된 발행을 점검하고 이를 제거하는 폴링 방법(polling method)을 도시하고 있으나, 온라인 상태에서 오프라인 상태로 바뀌는 사용자 또는 장치의 이벤트가 만료된 발행을 바로 제거하게 하는 이벤트-드리븐 모델(event-driven model)과 같은 다른 방법이 사용될 수 있음을 당업자들을 이해할 것이다.

<56> 본 내용이 구조적인 특징 및/또는 방법론적인 액트에 특정한 언어로 설명되었지만, 첨부된 청구항에 정의된 내

용이 상술된 특정 특징 또는 액트를 반드시 제한할 필요가 없음을 이해할 것이다. 오히려, 상술된 특정 특징 및 액트는 청구항을 구현하는 예시적인 형태로서 개시된다. 예를 들면, 프레즌스 시스템에 의해 처리되는 요청이 SIP 프로토콜을 이용하여 설명되었지만, TCP(Transmission Control Protocol) 상의 커스텀 프로토콜과 같은 다른 프로토콜이 사용될 수도 있다. 일부 예에서, 단계들은 프레즌스 서버의 컴포넌트에 의해 수행되는 것으로서 설명되었으나, 단계들은 또한 프레즌스 서버와 인터랙트하는 엔티티의 컴포넌트에 의해 수행될 수도 있다. 이 시스템은 또한 단일의 중앙 프레즌스 서버는 없고 프레즌스 정보가 피어들에 의해 분산되어 저장될 수 있는 피어-대-피어 네트워크로서 구현될 수 있다. 따라서, 본 발명은 첨부되는 청구항에 의한 바를 제외하고는 제한되지 않는다.

도면의 간단한 설명

<10> 도 1은 한 실시예에서의 프레즌스 시스템의 컴포넌트들을 도시하는 블록도.

<11> 도 2는 한 실시예에서, 발행 업데이트를 전송하기 위한 SIP 프로토콜 확장을 도시하는 프로토콜 리스팅.

<12> 도 3은 한 실시예에서의 프레즌스 시스템의 발행 업데이트 컴포넌트의 처리를 도시하는 흐름도.

<13> 도 4는 한 실시예에서, 구독 요청을 전송하기 위한 SIP 프로토콜 확장을 도시하는 프로토콜 리스팅.

<14> 도 5는 한 실시예에서의 프레즌스 시스템의 구독 업데이트 컴포넌트의 처리를 도시하는 흐름도.

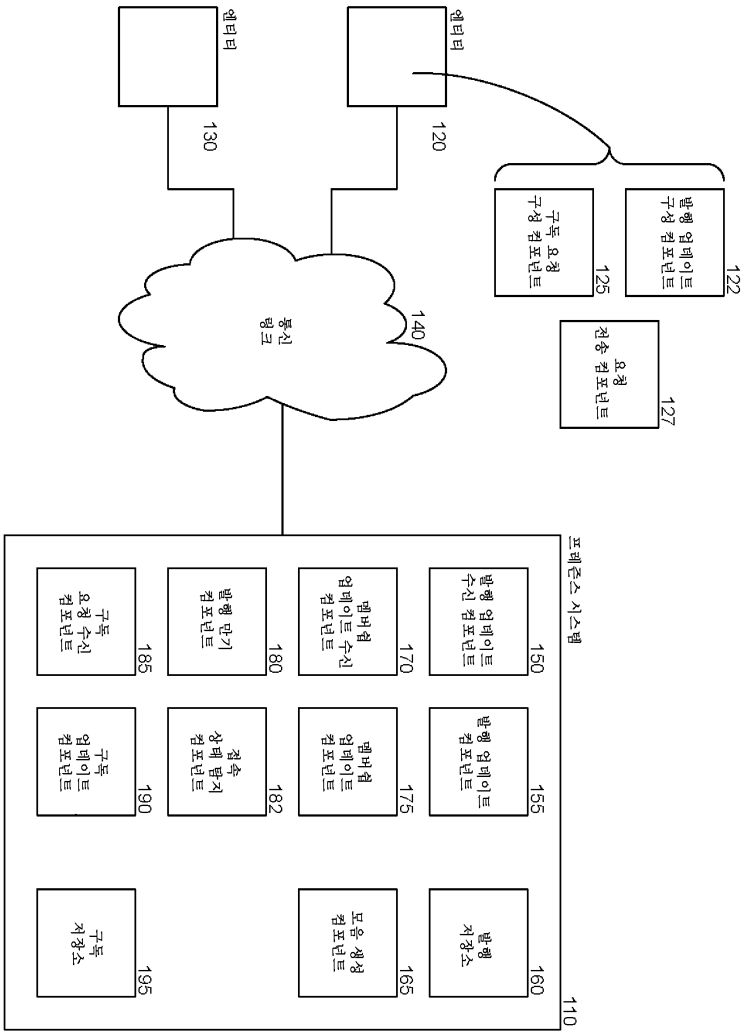
<15> 도 6은 한 실시예에서, 모음 멤버쉽을 업데이트하기 위한 SIP 프로토콜 확장을 도시하는 프로토콜 리스팅.

<16> 도 7은 한 실시예에서의 프레즌스 시스템의 멤버쉽 업데이트 컴포넌트의 처리를 도시하는 흐름도.

<17> 도 8은 한 실시예에서의 프레즌스 시스템의 발행 만료 컴포넌트의 처리를 도시하는 흐름도.

도면

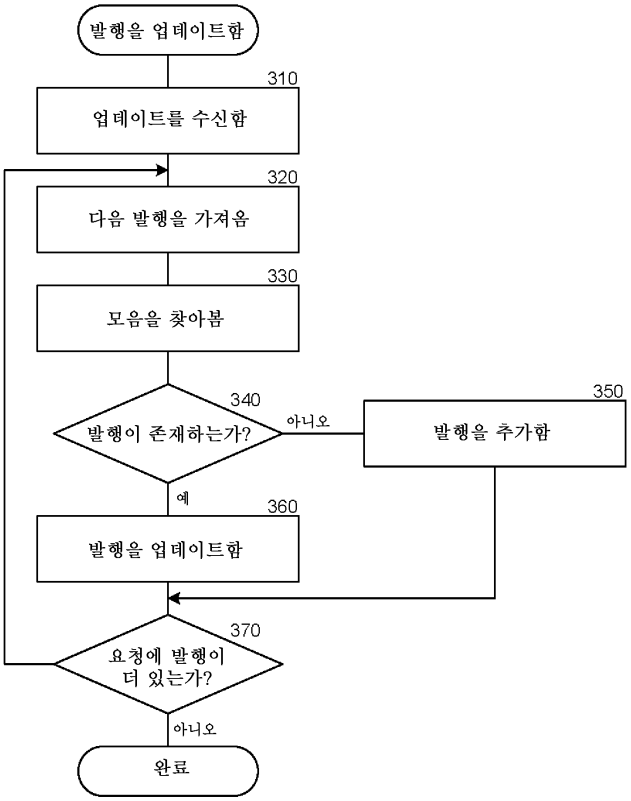
도면1


엔티티
130
엔티티
120
구독 요청 구성 컴포넌트
125
발행 업데이트 구성 컴포넌트
122
통신 링크
140
요청 전송 컴포넌트
127
프레즌스 시스템
구독 요청 수신 컴포넌트
185
발행 만기 컴포넌트
180
멤버쉽 업데이트 수신 컴포넌트
170
발행 업데이트 수신 컴포넌트
150
구독 업데이트 컴포넌트
190
결속 상태 탭지 컴포넌트
182
멤버쉽 업데이트 컴포넌트
175
발행 업데이트 컴포넌트
155
구독 저장소
195
모음 생성 컴포넌트
165
발행 저장소
160
110

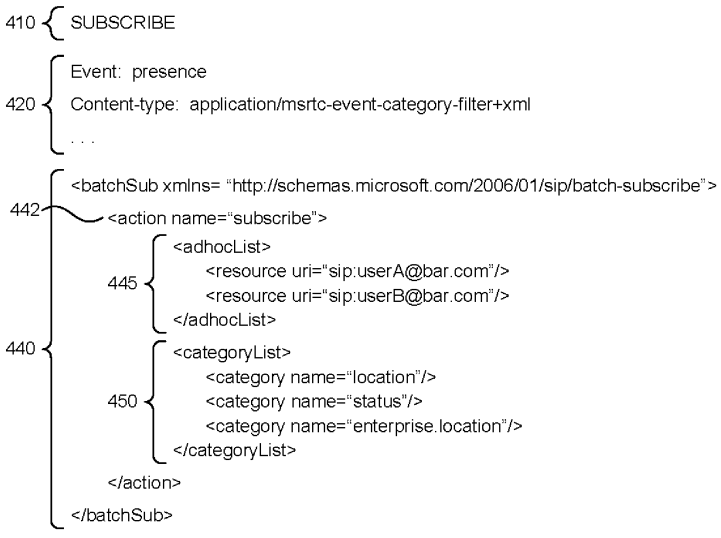
도면2


210
PUBLISH
220
Event: presence
Content-type: application/msrtc-presence-publish
. . .
230
<publish xmlns="http://schemas.microsoft.com/2006/09/sip/rich-presence">
232
<publications>
235
<publication categoryName="status" instance="laptop" container="1"
version="0" uri="sip:foo@bar.com" expireType="user"/>
240
<publication categoryName="status" instance="desktop" container="2"
version="0" uri="sip:foo@bar.com" expireType="device"/>
250
<publication categoryName="location" instance="laptop" container="2"
uri="sip:foo@bar.com" expireType="0"/>
</publications>
</publish>

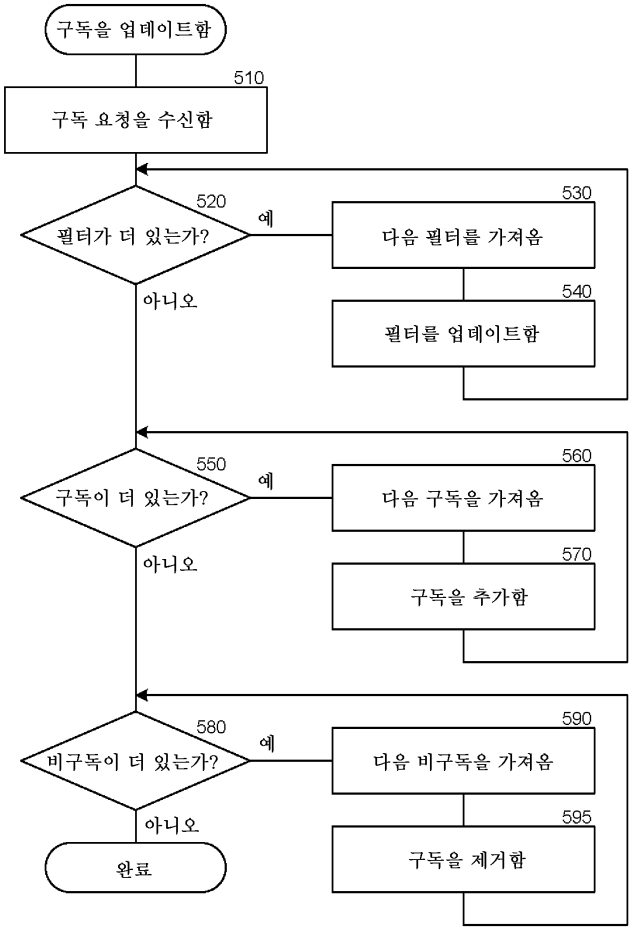
도면3


발행을 업데이트함
310
업데이트를 수신함
320
다음 발행을 가져옴
330
모음을 찾아봄
340
발행이 존재하는가?
아니오
350
발행을 추가함
예
360
발행을 업데이트함
370
요청에 발행이 더 있는가?
아니오
완료

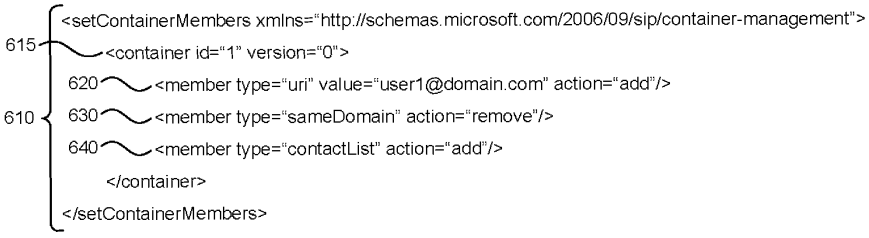
도면4


410
SUBSCRIBE
420
Event: presence
Content-type: application/msrtc-event-category-filter+xml
. . .
440
<batchSub xmlns= "http://schemas.microsoft.com/2006/01/sip/batch-subscribe">
442
<action name="subscribe">
445
<adhocList>
<resource uri="sip:userA@bar.com"/>
<resource uri="sip:userB@bar.com"/>
</adhocList>
450
<categoryList>
<category name="location"/>
<category name="status"/>
<category name="enterprise.location"/>
</categoryList>
</action>
</batchSub>

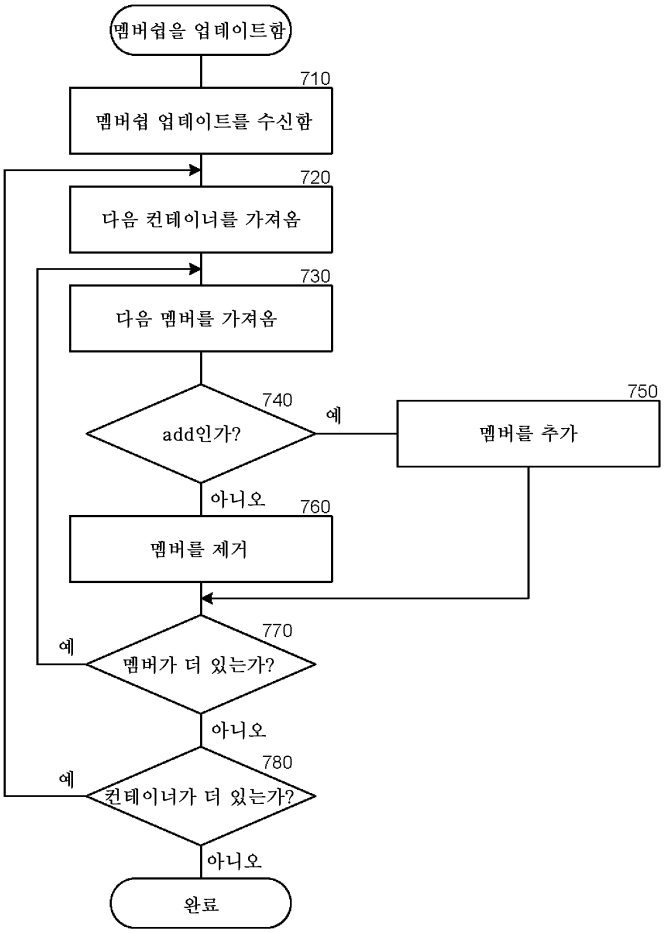
도면5


구독을 업데이트함
510
구독 요청을 수신함
520
필터가 더 있는가?
예
530
다음 필터를 가져옴
540
필터를 업데이트함
아니오
550
구독이 더 있는가?
예
560
다음 구독을 가져옴
570
구독을 추가함
아니오
580
비구독이 더 있는가?
예
590
다음 비구독을 가져옴
595
구독을 제거함
아니오
완료

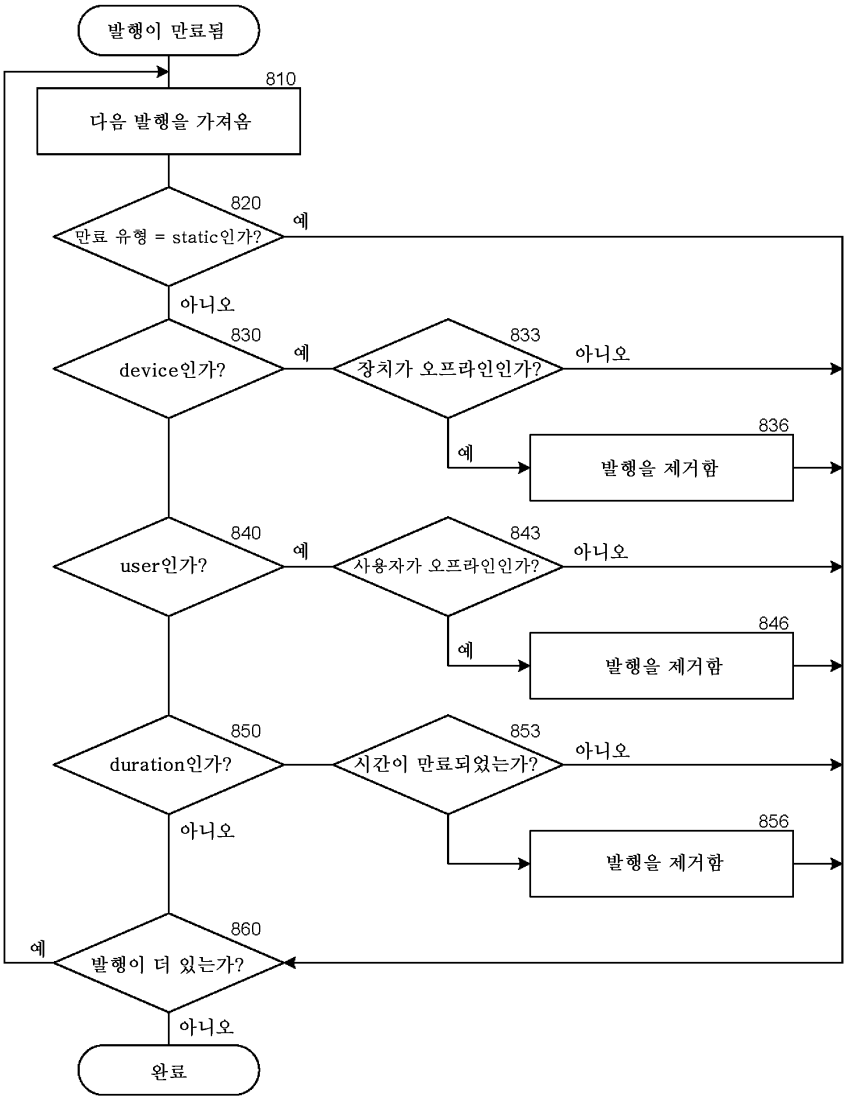
도면6


615
610
620
630
640
<setContainerMembers xmlns="http://schemas.microsoft.com/2006/09/sip/container-management">
<container id="1" version="0">
<member type="uri" value="user1@domain.com" action="add"/>
<member type="sameDomain" action="remove"/>
<member type="contactList" action="add"/>
</container>
</setContainerMembers>

도면7


멤버쉽을 업데이트함
710
멤버쉽 업데이트를 수신함
720
다음 컨테이너를 가져옴
730
다음 멤버를 가져옴
740
add인가?
예
750
멤버를 추가
아니오
760
멤버를 제거
770
예
멤버가 더 있는가?
아니오
780
예
컨테이너가 더 있는가?
아니오
완료

도면8


발행이 만료됨
810
다음 발행을 가져옴
820
만료 유형 = static인가?
예
아니오
830
device인가?
예
833
장치가 오프라인인가?
아니오
예
836
발행을 제거함
840
user인가?
예
843
사용자가 오프라인인가?
아니오
예
846
발행을 제거함
850
duration인가?
853
시간이 만료되었는가?
아니오
아니오
856
발행을 제거함
860
예
발행이 더 있는가?
아니오
완료