(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년05월07일
(11) 등록번호 10-0827280
(24) 등록일자 2008년04월28일

(51) Int. Cl.
G06F 15/16 (2006.01)
(21) 출원번호 10-2005-7005849
(22) 출원일자 2005년04월04일
심사청구일자 2006년04월12일
번역문제출일자 2005년04월04일
(65) 공개번호 10-2005-0055746
(43) 공개일자 2005년06월13일
(86) 국제출원번호 PCT/GB2003/004201
국제출원일자 2003년10월01일
(87) 국제공개번호 WO 2004/031882
국제공개일자 2004년04월15일
(30) 우선권주장
2,406,713 2002년10월04일 캐나다(CA)
(56) 선행기술조사문헌
java servlet programming
전체 청구항 수 : 총 8 항

(73) 특허권자
인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드
(72) 발명자
엔쥐 조아나
캐나다 온타리오 엘3알 9씨7 유니온빌 4 캡틴 프란시스 드라이브
카푸르 로히트
캐나다 온타리오 엘3비 4이1 리치몬드 힐 31 코버르그 크레센트
니굴 레호
캐나다 온타리오 엠2알 3브이4 토론토 에이피알 305 623 핀치애비뉴 웨스트
(74) 대리인
김태홍

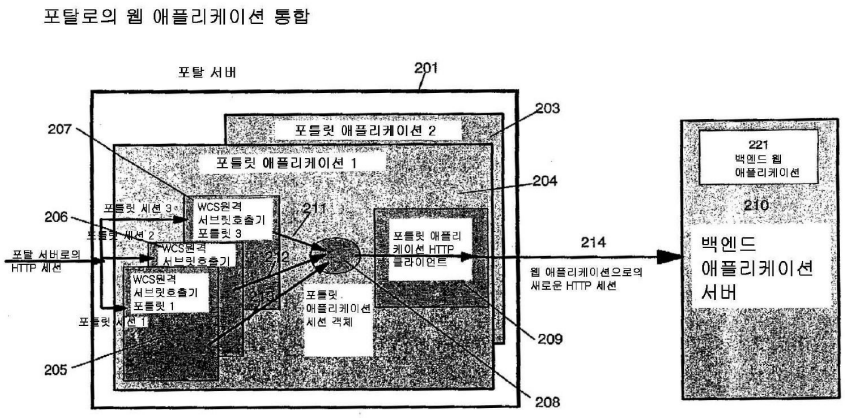
심사관 : 노영철

(54) 포탈 서버로부터 세션 정보를 릴레이하기 위한 방법 및장치

(57) 요 약

본 발명은 웹 애플리케이션에 액세스하려는 사용자의 요청에 응답하는 관련된 포틀릿들의 집합을 관리하는 웹 포탈 시스템을 위한 다음을 포함하는 장치 및 방법을 제공한다: 관련된 포틀릿들의 사용자 요청으로부터의 파라미터들을 저장하기 위한 포틀릿 애플리케이션 세션 객체; 및, 상기 관련된 포틀릿들과 상기 웹 애플리케이션 사이에서 통신하여 상기 관련된 포틀릿으로부터 수신된 사용자 요청들을 상기 웹 애플리케이션에 전달하기 위한, 상기 포틀릿 애플리케이션 세션 수단에 링크된 포틀릿 애플리케이션 통신 클라이언트.

대표도



특허청구의 범위

청구항 1

적어도 하나의 포틀릿 애플리케이션을 포함하고, 사용자 세션 정보를 릴레이하기 위하여 관련된 포틀릿들의 집합을 관리하기 위한 포탈 서버 시스템으로서, 포틀릿은 포탈 페이지의 가시적인 활성 구성요소인, 포탈 서버 시스템에 있어서,

상기 포틀릿 애플리케이션은:

웹 애플리케이션에 액세스하기 위한 사용자의 요청에 응답하여, 제1 포틀릿(205)을 개시하기 위한 수단과;

상기 제1 포틀릿에 대한 상기 사용자를 위한 포틀릿 애플리케이션 세션 오브젝트(208)를 생성하기 위한 수단과;

상기 요청으로부터 파라미터들을 저장하기 위한 수단(308)과;

상기 웹 애플리케이션에 액세스하기 위한 상기 사용자의 추가 요청에 대하여, 상기 제1 포틀릿과 연관되는 추가 포틀릿들을 생성하기 위한 수단과;

상기 저장된 파라미터들을 사용하여 상기 추가 포틀릿들을 상기 포틀릿 애플리케이션 세션 오브젝트와 관련시키기 위해, 상기 포틀릿 애플리케이션 세션 오브젝트에 의해 제어되는 포틀릿 애플리케이션 세션 오브젝트 데이터 저장부(302)와;

상기 제1 포틀릿 및 상기 추가 포틀릿들로부터 수신된 사용자 요청들을 상기 웹 애플리케이션으로 전달하기 위하여, 상기 포틀릿 애플리케이션 세션 오브젝트 및 상기 웹 애플리케이션과 통신하기 위한 포틀릿 애플리케이션 통신 클라이언트(209)를 생성하기 위한 수단

을 포함하고,

상기 사용자 세션 정보는, 상기 사용자 세션 정보를 상기 웹 애플리케이션의 대응 세션에 매핑하기 위한 사용자 세션 정보를 포함하는 것인, 포탈 서버 시스템.

청구항 2

제1항에 있어서, 상기 포틀릿 애플리케이션 통신 클라이언트는 사용자 세션 정보를 저장하는 것인, 포탈 서버 시스템.

청구항 3

제1항에 있어서, 사용자 요청들로부터의 데이터 및 지시들을 상기 관련된 포틀릿들에 저장하기 위해, 포틀릿 요청 파라미터 맵이 이용되는 것인, 포탈 서버 시스템.

청구항 4

제1항에 있어서, 상기 사용자 세션 정보는, 상기 사용자 세션 정보를 상기 웹 애플리케이션의 대응 세션에 매핑하기 위하여,

사용자 ID;

사용자 적격(credential)들;

언어 우선순위(preference)들;

세션 타임아웃 정보;

세션 ID

를 포함하는 세트로부터 선택되는 것인, 포탈 서버 시스템.

청구항 5

제1항에 있어서, 상기 포틀릿 애플리케이션 통신 클라이언트는, 상기 통신 클라이언트가 상기 웹 애플리케이션

용 데이터 및 지시들을 제공할 수 있도록 상기 관련된 포틀릿들로부터의 요청들을 저장하는 요청 버퍼를 갖는 것인, 포탈 서버 시스템.

청구항 6

제5항에 있어서, 상기 포틀릿 애플리케이션 통신 클라이언트는, 상기 포틀릿 애플리케이션 통신 클라이언트가 상기 웹 애플리케이션용 데이터 및 지시들을 제공할 수 있도록, 상기 관련된 포틀릿들의 포틀릿 요청 파라미터 맵들로부터의 요청들을 저장하는 요청 버퍼를 갖는 것인, 포탈 서버 시스템.

청구항 7

웹 애플리케이션에 액세스하기 위한 사용자 요청에 응답하여 관련된 포틀릿들의 집합을 관리하여 상기 관련된 포틀릿들과 상기 웹 애플리케이션 간에 사용자 세션 정보를 릴레이하기 위한 포탈 서버 시스템용 방법으로서,

웹 애플리케이션에 액세스하기 위한 사용자의 요청에 응답하여, 제1 포틀릿(205)을 개시하는 단계와;

상기 제1 포틀릿에 대한 상기 사용자를 위한 포틀릿 애플리케이션 세션 오브젝트(208)를 생성하는 단계;

상기 요청으로부터 파라미터들을 저장하는 단계;

상기 웹 애플리케이션에 액세스하기 위한 상기 사용자의 추가 요청에 대하여, 상기 제1 포틀릿과 연관되는 추가 포틀릿들을 생성하는 단계;

상기 저장된 파라미터들을 사용하여 상기 추가 포틀릿들을 상기 포틀릿 애플리케이션 세션 오브젝트와 관련시키기 위해, 상기 포틀릿 애플리케이션 세션 오브젝트에 의해 제어되는 포틀릿 애플리케이션 세션 오브젝트 데이터 저장부(302)를 이용하는 단계;

상기 제1 포틀릿 및 상기 추가 포틀릿들로부터 수신된 사용자 요청들을 상기 웹 애플리케이션으로 전달하기 위하여, 상기 포틀릿 애플리케이션 세션 오브젝트 및 상기 웹 애플리케이션과 통신하기 위한 포틀릿 애플리케이션 통신 클라이언트(209)를 생성하는 단계

을 포함하고,

상기 사용자 세션 정보는, 상기 사용자 세션 정보를 상기 웹 애플리케이션의 대응 세션에 매핑하기 위한 사용자 세션 정보를 포함하는 것인, 포탈 서버 시스템용 방법.

청구항 8

컴퓨터상에서 실행되어 제7항의 방법을 수행하기 위한 컴퓨터 프로그램 코드를 기록한 컴퓨터 판독가능 기록매체.

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

명 세 서

기 술 분 야

<1> 본 발명은 인터넷에 관한 것으로, 더 상세하게는 향상된 능력을 웹 사이트에 제공하는 웹 애플리케이션에서 포탈들(portals) 및 포틀릿들(portlets)을 생산하고 사용하기 위한 방법 및 장치에 관한 것이다.

배 경 기 술

<2> 월드 와이드 웹(World Wide Web)은 그래픽 정보를 사용자에게 보내어, 인터넷 상의 통신에 패러다임의 변화를 가져왔다. 웹의 출현으로 통신가능성(communicability) 및 광역 접속성(broad connectivity)을 증가하기 위한 요구가 있었고 이 요구들은 여전히 존재한다.

<3> (웹 포탈로 미리 알려진) 포탈은 인터넷 공간에서 주요 패러다임 변화를 가져왔다. 이메일, 포럼, 서치엔진, 데이터베이스 또는 다른 정보들과 같은 리소스들의 배열 또는 서비스들을 제공하는 웹 사이트는 포탈로 생각될 것이다. 최초의 포탈은 온라인 서비스들이었을 것이다. 최초에, 인터넷 서핑을 하는 사용자들은, 조합되고 다양한 사이트로부터의 들어오는 정보를 제공하는 웹 페이지를 월드 와이드 웹에서 볼 수 있었지만, 집합의 구조(aggregation's constitution)는 사용자에게 투명하였다. 전형적인 웹 브라우저를 사용하는 사용자는 응집한(cohesive) 웹 페이지가 디스플레이되는 것을 본다. 보이는 웹 사이트와 관련되지 않는 다양한 웹 사이트들로부터의 페이지의 서로 다른 부분들의 근원(origination)은 쉽게 명확해지지 않는다. 이 부분들은 포틀릿들로 불린다.

<4> 포틀릿들은 눈에 보이는 활동적인 구성들이고 이 구성들을 종단 사용자들이 포탈 페이지에서 보게 된다. 데스크탑 PC의 윈도우와 유사하게, 각 포틀릿들은 브라우저의 일부 또는 결과를 디스플레이하는 개인 디지탈 장치 스크린의 일부를 갖는다.

<5> 사용자의 관점에서, 포틀릿은 콘텐트(content) 채널 또는 사용자가 가입하고, 개인 포탈 페이지에 추가하며, 또한 개인화된 콘텐트를 보여주도록 배열하는 애플리케이션이다.

<6> 콘텐트 제공자의 관점에서, 포틀릿은 콘텐트를 사용가능하게 만드는 수단이다.

<7> 포탈 관리자의 관점에서, 포틀릿은 포탈로 등록되어, 사용자들이 가입할 수 있는 콘텐트 컨테이너이다.

<8> 포탈의 관점에서, 포틀릿은 페이지들 중 하나로 렌더링(rendering)되는 구성이다.

<9> 기술적 관점에서, 포틀릿은 하나의 코드 또는 포탈 서버 상에서 실행되고 포탈 페이지들에 포함되어야 하는 콘텐트를 제공하는 작은 애플리케이션이다. 간단하게는, 포틀릿은 포탈 내에서 동작하는 자바 서브릿(JAVA™ servlet)일 수 있다.

<10> (전형적으로 월드 와이드 웹의 서로 다른 장소로부터 나온) 주어진 페이지의 각 부분(포틀릿)은 동일한 페이지의 또 다른 부분(포틀릿)과 협력하여 사용자의 서핑 또는 페이지 액세스를 위한 더 높은 성능을 달성할 수 있다. 따라서, 포탈은 복수의 채널을 경유하여 복수의 정보 소스들로의 다중 사용자들을 위한 액세스의 한 점(single point)이 된다.

<11> 포틀릿들은 기업과 고객간의 비지니스, 기업 간의 비지니스 등의 다양한 비지니스 모델들에 적용될 수 있다. 포탈 패러다임을 빨리 채택하기 위한 열쇠는 무결절성(seamless) 방식으로 기존의 웹 애플리케이션 데이터를 포탈 프레임워크(framework)로 통합하는 능력을 강화하는 것이다.

<12> 그러나, 포탈로의 무결절성 웹 애플리케이션 통합을 위하여 다양한 기술적 장애물들이 존재한다.

<13> 사용자들이 포탈 페이지에 액세스하는 경우, 각 사용자에 대한 원시 HTTP 요청 포탈 서버로 향한다. 각 포틀릿들은 포틀릿 세션으로 불리는 자신의 독립된 세션을 갖는다. 포틀릿이 주어진 웹 애플리케이션으로부터 나오는 정보를 렌더링할 필요가 있는 경우, 웹 애플리케이션의 뷰 포인트로부터 하나의 응집된 HTTP 세션으로서, 다양한 포틀릿들로부터 주어진 웹 애플리케이션까지 생성되는 다중 HTTP 요청을 유지하는 메커니즘이 존재하지 않는다. 또한, 다중 포틀릿 세션 사이의 세션 정보를 웹 애플리케이션 세션에 릴레이하는 메커니즘은 존재하지 않는다. 웹 애플리케이션이 정확하게 렌더링하기 위하여 포틀릿들로부터의 세션 정보는 웹 애플리케이션에 포워딩될 필요가 있다. 포틀릿 세션으로부터 웹 애플리케이션까지의 포워딩에 필요한 중요한 세션 정보의 예는 위치 정보, 사용자 에이전트, 및 세션 타임아웃 정보를 포함한다.

<14> 종래 기술에서는 다음의 포탈 가공물들(artifacts)이 기존의 웹 애플리케이션과 함께 협력하는 방법에 관계된 한계들이 있다. 포탈 아키텍처로의 웹 애플리케이션 통합의 구현은 잘 정의되지 않는다. 이 엔터티들은 다음을 포함한다:

<15> 포탈에 대한 원시(original) HTTP 요청;

<16> 포탈 내의 포틀릿 세션;

<17> 관련된 웹 애플리케이션에 대한 포탈로부터의 HTTP 요청.

<18> 서로 다른 사용자들이 포탈 페이지에 액세스하는 경우, 각 사용자에 대한 원시 HTTP 요청은 포탈 서버(a)를 향한다. 또한 각 사용자에 대한 원시 HTTP 세션은 전체적으로 포탈 서버가 소유한다. 각 포틀릿들은 포틀릿 세션으로 불리는 독립적인 세션을 갖는다. 포틀릿이 주어진 웹 애플리케이션(b)으로부터 오는 정보를 렌더링할 필요가 있는 경우, 전형적으로 다음의 기술적 장애물들이 존재한다:

<19> i. 포틀릿이 백엔드 웹 애플리케이션에 대한 또는 백엔드 웹 애플리케이션으로부터의 HTTP 요청들 및 응답들을 생성하는 기존의 메커니즘이 없다.

<20> j. 백엔드 웹 애플리케이션(및 웹 애플리케이션의 세션)에 대한 다중 요청들 및 응답들을 호출 포틀릿(및 포틀릿 세션)에 대한 다중 요청들 및 응답들에 정확히 매핑하도록 관리하는 기존의 메커니즘이 없다. 각각(포틀릿 및 웹 애플리케이션 모두)은 그에 따른 사용자 세션을 유지한다.

<21> 다중 포틀릿들이 동일한 웹 애플리케이션을 호출하는 경우, 동일한 웹 애플리케이션 세션 내에서 이러한 다중 포틀릿들의 요청들을 취급하는 웹 애플리케이션으로는 상기 문제는 어려워진다.

<22> k. 다중 포틀릿 세션들과 웹 애플리케이션 세션 사이의 세션 정보를 릴레이(relay)하는 기존의 메커니즘은 없다.

<23> 동일한 포틀릿 애플리케이션 내의 잘 정의된 포틀릿들의 세트가 백엔드의 하나의 웹 애플리케이션과 협력하는 경우, 애플리케이션으로부터 렌더링된 정보가 포틀릿들의 포탈 정보의 설정과 일치하도록 참여하는 모든 포틀릿들은 정확한 세션 정보를 백엔드의 웹 애플리케이션에 검색 및 포워딩할 수 있어야 한다. 그러한 설정의 예는 위치(locale) 정보, 특정 액세스의 사용자 에이전트(agent) 등을 포함한다. 예를 들면, 웹 애플리케이션이 송

신한 응답들은 그것을 디스플레이하는 포탈 서버 내의 포틀릿과 동일한 위치를 사용하고 잇어야 한다.

<24> 포탈 사용자의 자격이 백엔드 웹 애플리케이션에 의하여 요구되지 않는 한번의 사인온(single sign on)에 대한 기존의 메커니즘은 없다. 이것은 중요한 요구사항이다. 포틀릿들은 서로 다른 근원들 및 식별 요구사항들을 가지기 때문에, 이러한 메커니즘이 없다면 사용자가 웹 페이지의 한 부분에서 동일한 웹 페이지의 다른 부분으로 이동하는 경우 사용자의 자격이 요구될 것이다.

<25> 주어진 포틀릿 애플리케이션과 관련된 웹 애플리케이션 백엔드의 포틀릿들 사이에서 다중 요청들 또는 응답들의 동기화에 대한 기존의 메커니즘은 없다.

<26> 종래 기술은 동일한 애플리케이션 내의 다중 포틀릿들이, 동적으로 정의되지 않은 다양한 통합 웹 애플리케이션뿐만 아니라, (동일한 콘텍스트를 공유하는) 하나의 또 다른 웹 애플리케이션과 협력할 수 있는 방법에 관한 한계가 있었다.

<27> 동일한 '콘텍스트'를 동적으로 공유함으로써 협력하는 다중 포틀릿들을 포함하는 하나의 사용 시나리오는 개념적으로 한계를 예시할 것이다.

<28> 동일한 포탈 웹 페이지상에 디스플레이되는 3개의 포틀릿들로:

<29> - 제1 포틀릿은 계정들의 리스트를 디스플레이함으로써 계정 요약을 보여준다

<30> - 제2 포틀릿은 주어진 계정들의 미결 요금 청구서 리스트를 보여준다.

<31> - 제3 포틀릿은 주어진 계정의 주문 기록 요약을 보여준다.

<32> 제2 및 제3 포틀릿들은 개념적으로 제1 포틀릿에 동적으로 종속되어 있고, 미결 요금 청구서(제2 포틀릿) 및 주문 기록(제3 포틀릿)을 반영하여 제1 포틀릿의 계정 리스트로부터 선택된 계정과 동기화된다.

<33> 종래 기술의 한계들:

<34> i. 협력적으로 동작할 포틀릿 애플리케이션 내의 포틀릿들의 서브 그룹핑을 정의하는 메커니즘은 존재하지 않는다.

<35> j. 주어진 포틀릿 애플리케이션 내의 포틀릿들의 서브 그룹 사이에서 공유되는 (동적으로 변화될 수 있는) 콘텍스트를 정의하는 메커니즘은 존재하지 않는다: 여기서의 콘텍스트의 예는 포틀릿 1에서 선택된 계정이고, 그 계정 선택은 동적으로 변화될 수 있다.

<36> k. 콘텍스트의 변화를 동적으로 검출하는 메커니즘은 존재하지 않는다: 상기 예의 포틀릿 1의 계정 리스트의 하나의 계정부터 또 다른 계정까지의 선택 변화의 예.

<37> l . 동일한 콘텍스트를 공유하는 포틀릿들의 서브 그룹 내의 참여하는 각 포틀릿들에 대하여 미리 정의된 액션(또는 응답들)을 등록(register)하는 메커니즘은 존재하지 않는다: 콘텍스트가 (포틀릿 1에서 하나의 계정 선택으로부터 또 다른 것으로) 변화되는 경우 미결의 요금 청구서(포틀릿 2의 액션)의 리스트를 디스플레이하는 예.

<38> m. 동적 콘텍스트를 관련된 통합 웹 애플리케이션들로 릴레이하는 메커니즘은 존재하지 않는다.

<39> 종래 기술에는 포틀릿 애플리케이션 내의 포틀릿들의 그룹을 위한 리프레쉬(refresh) 시퀀스를 정의하는 메커니즘이 존재하지 않는다.

<40> i. 오늘날, 디스플레이되고 있는 포틀릿들의 주어진 세트 리프레쉬 순서를 포탈 디자이너가 특정하는 규정(provision)은 없다.

<41> 상기 시나리오에서, 제2 및 제3 포틀릿들이 자동적으로 리프레쉬되도록, 포탈 디자이너는 제1 포틀릿(계정 리스트)을 첫번째로 리프레쉬하고, 제2 포틀릿을 두번째로 리프레쉬하는 것 등을 원할 것이다. (포틀릿이 배치된 경우) 정의된 액션들은 정확한 시퀀스로 일어난다.

<42> 포탈의 아키텍처에는 비지니스 룰(rule)에 기초한 포틀릿들의 집합(aggregation) 및 사용자들의 롤(role)을 포함하는 사용자의 프로파일링 정보를 지원하는 잘 정의된 메커니즘이 부족하다.

<43> i. 비지니스 룰에 기초하여 사용자마다 포탈 리소스들의 집합을 정의하는 메커니즘이 존재하지 않는다.

<44> 예: 모든 십대 포탈 사용자들은 포틀릿들의 하나의 그룹을 보고, 모든 연장자 포탈 사용자들은 포틀릿들의 또 다른 그룹을 본다.

<45> j. 런타임에 동적으로 수행되는 룰 기반 및 사용자 기반 포틀릿들의 집합을 위한 메커니즘이 존재하지 않는다.

<46> 관련된 통합 백엔트 웹 애플리케이션들은 포탈 레벨의 비지니스 룰 및 사용자 프로파일 정보를 공유하지 않는다.

<47> 룰 또는 사용자 구분(segmentation)이 포탈과 통합 백엔드 웹 애플리케이션 사이에서 일치하도록 하는, 통합 웹 애플리케이션에서의 비지니스 룰 또는 사용자 구분 정보의 공유는 없다. 예를 들면, 십대의 연령 범위를 정의하는 룰이 있는 경우, 그 룰은 일관성을 위하여 통합 웹 애플리케이션에 가시적이고(visible) 적용할 수(applicable) 있어야 한다.

<48> **전문용어**

<49> **포틀릿**

<50> 포틀릿은 종단 사용자들이 포탈 웹 페이지들 내에서 볼 수 있는 가시적인 능동적(active) 구성이다. 데스크탑 PC의 윈도우와 유사하게, 각 포틀릿은 포틀릿의 특정 정보를 디스플레이하는 브라우저 또는 PDA(개인 디지털 기기) 스크린의 일부를 소유한다.

<51> **포틀릿 애플리케이션**

<52> 포틀릿은 또한 포틀릿 애플리케이션에서 함께 그룹화될 수 있다. 포틀릿 애플리케이션들은 웹 아카이브(archive) 파일들(WAR)을 사용하여 분배 및 배치된다. 표준 웹 애플리케이션 배치 디스크립터(deployment descriptor)로의 포틀릿의 특정한 확장이 있다.

<53> **포틀릿 메시지**

<54> 포틀릿 액션들 및 포틀릿 메시지들을 사용하여 2개의 포틀릿 사이에서 통신을 하기 위하여 포틀릿 메시지들이 사용된다. 송신하는 포틀릿은 포틀릿 액션을 생성하고, 액션을 URL로 인코딩한다. 예컨대, 업무를 수행하려는 사용자가 URL을 써 넣은 경우 액션 리스너(listener)가 호출되고 필요한 데이터를 송신하는 포틀릿 메시지를 송신한다.

<55> **포틀릿 세션**

<56> 포틀릿 세션은 각 사용자 및 각 포틀릿 인스턴스에 대하여 세션 정보를 유지하도록 로그온하는 각 사용자 및 각 포틀릿 인스턴스에 대하여 생성된다.

발명의 상세한 설명

<57> 본 발명의 다양한 실시예들은 종래 기술의 하나 이상의 결점들을 언급한다.

<58> 본 발명은 관련된 포틀릿들에 대한 포탈로부터의 세션 정보를 백엔드 웹 애플리케이션들에 릴레이하는 방법을 제공한다. 이것은 세션 정보를 포탈로부터 백엔드 웹 애플리케이션까지 릴레이를 가능케 하여, 웹 애플리케이션이 포탈에 의해 제공되는 세션 정보에 따라 동일하게 행동하는 것을 가능하게 만든다.

<59> 본 발명의 실시예는 사용자 세션 정보를 저장하기 위한 사용자 세션 정보 저장부(매핑 테이블)를 제공한다. 관련된 포틀릿들은 포틀릿들에 대한 사용자 요청으로부터 데이터 및 명령들을 저장하는 포틀릿 요청 파라미터 맵들을 갖는다. 포틀릿 요청들로부터의 파라미터들은 상기 포틀릿으로부터 웹 애플리케이션 백엔드까지 HTTP 요청들에 대하여 릴레이된다.

<60> 위치, 세션 타임아웃 정보 등의 중요한 세션 정보는 이제 백엔드 웹 애플리케이션에 릴레이될 수 있다.

<61> 본 발명의 실시예로서, 이제 포틀릿 서버와 백엔드 웹 애플리케이션 사이의 공통 세션 데이터의 공유를 위한 세션 릴레이를 구현할 수 있으므로, 포탈 서버의 응답들과 동기화된 백엔드 웹 애플리케이션 응답들로서, 백엔드 웹 애플리케이션은 포탈 서버로부터 정보를 수신하여, 백엔드 웹 애플리케이션에 의하여 그 정보가 이용되는 것을 가능케 한다.

<62> 본 발명의 하나의 실시예는 웹 애플리케이션에 대하여 사용자에게 웹 포탈을 디스플레이하는, 다음을 포함하는 장치를 제공하는데, 상기 웹 포탈은 복수의 관련된 포틀릿들을 디스플레이하고, 상기 사용자의 액세스가 가능한 정보를 서로 공유한다: 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하기 위한 포탈 서버; 관련된 포틀릿들의 집합을 관리하기 위한 포탈 서버 상에서 동작하는 포틀릿 애플리케이션; 상기 포틀릿 애플리케이션은 다음을 포함한다: 웹 애플리케이션에 액세스하려는 사용자의 요청들에 대하여 포틀릿들을 시작하는 수단;

포틀릿들에 대한 포틀릿 애플리케이션 세션을 관리하는 수단; 및, 포틀릿 애플리케이션 세션 객체와 포틀릿들을 관련시키기 위하여 사용자의 요청들로부터의 파라미터들을 저장하기 위한, 포틀릿 애플리케이션 세션 객체가 제어하는 포틀릿 애플리케이션 세션 객체 데이터 저장부를 제공한다.

<63> 본 발명의 장치는 포틀릿 애플리케이션 세션 객체와 웹 애플리케이션 사이에서 통신을 하여 관련된 포틀릿들로부터 수신된 사용자 요청을 웹 애플리케이션에 전달하기 위하여 포틀릿 애플리케이션에서 포틀릿 애플리케이션 통신 클라이언트를 포함한다. 포틀릿 애플리케이션은 공통 키를 포틀릿 애플리케이션 세션 객체와 관련된 각 포틀릿에 할당할 수 있다.

<64> 본 발명의 또 다른 실시예는 복수의 사용자들에게 웹 애플리케이션에 대한 웹 포탈을 디스플레이하는, 다음을 포함하는 장치를 제공하는데, 상기 웹 포탈은 사용자의 액세스가 가능한 정보를 공유하는 복수의 포틀릿들을 디스플레이한다: 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하기 위한 포탈 서버; 복수의 각 사용자들에 대하여 관련된 포틀릿들의 집합을 관리하기 위한, 복수의 각 사용자들에 대하여 포탈 서버 상에서 동작하는 포틀릿 애플리케이션; 상기 각 포틀릿 애플리케이션은 다음을 포함한다: 웹 애플리케이션에 액세스하려는 복수의 사용자 중 한 명의 요청들에 대하여 포틀릿들을 시작하는 수단; 포틀릿들에 대한 포틀릿 애플리케이션 세션 객체를 관리하는 수단; 및, 포틀릿 애플리케이션 세션 객체와 포틀릿들을 관련시키기 위하여 사용자의 요청들로부터의 파라미터들을 저장하기 위한, 포틀릿 애플리케이션 세션 객체가 제어하는 포틀릿 애플리케이션 세션 객체 데이터 저장부.

<65> 본 발명의 또 다른 실시예는 복수의 웹 애플리케이션들에 대하여 사용자에게 웹 포탈을 디스플레이하는 다음을 포함하는 장치를 제공하는데, 상기 웹 포탈은 사용자의 액세스가 가능한 정보를 공유하는 복수의 관련된 포틀릿들을 디스플레이한다: 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하기 위한 포탈 서버; 포틀릿 서버 상에서 동작하기 위한 복수의 웹 애플리케이션들과 각각 관계있는 복수의 포틀릿 애플리케이션들, 상기 각 포틀릿 애플리케이션은 관련된 포틀릿들의 집합을 관리하는 것으로 적합화된다; 상기 각 포틀릿 애플리케이션은 다음을 포함한다: 복수의 웹 애플리케이션 중 하나에 액세스하려는 사용자의 요청에 대하여 포틀릿들을 시작하는 수단; 포틀릿들에 대한 포틀릿 애플리케이션 세션 객체를 관리하는 수단; 및, 포틀릿 애플리케이션의 포틀릿들을 포틀릿 애플리케이션 세션의 포틀릿 애플리케이션 세션 객체와 관련시키기 위하여 사용자의 요청들로부터의 파라미터들을 저장하기 위한, 포틀릿 애플리케이션 세션 객체가 제어하는 포틀릿 애플리케이션 세션 객체.

<66> 본 발명의 장치의 또 다른 태양은 포틀릿 애플리케이션 세션 객체로 다중 웹 애플리케이션들에 접속하도록 적합화된 사용자 세션 정보 테이블을 포함한다.

<67> 본 발명의 또 다른 실시예는 웹 애플리케이션에 대하여 사용자에게 웹 포탈을 디스플레이하는, 다음을 포함하는 장치를 제공하는데, 상기 웹 포탈은 사용자의 액세스가 가능한 정보를 서로 공유하는 복수의 관련된 포틀릿들을 디스플레이한다: 웹 애플리케이션으로의 액세스를 제공하기 위한 웹 포탈을 동작하는 포탈 서버; 관련된 포틀릿들을 관리하기 위한, 포탈 서버 상에서 동작하는 포틀릿 애플리케이션; 상기 포틀릿 애플리케이션은 다음을 포함한다: 웹 애플리케이션에 액세스하려는 사용자의 요청에 대하여 제1 포틀릿을 시작하는 수단; 사용자를 위하여 제1 포틀릿에 대한 포틀릿 애플리케이션 세션 객체를 생성하는 수단; 요청으로부터의 파라미터들을 저장하는 수단; 웹 애플리케이션에 액세스하려는 사용자의 추가적인 요청에 대하여 제1 포틀릿과 관련된 추가적인 포틀릿들을 생성하는 수단; 추가적인 포틀릿들을 포틀릿 애플리케이션 세션 객체와 관련시키기 위하여 저장된 파라미터를 사용하기 위한, 포틀릿 애플리케이션 세션 객체가 제어하는 포틀릿 애플리케이션 세션 객체 데이터 저장부; 및, 포틀릿 애플리케이션 세션 객체와 웹 애플리케이션이 통신을 하여 제1 및 추가적인 포틀릿들로부터 수신한 사용자 요청들을 웹 애플리케이션에 보내기 위한 포틀릿 애플리케이션 통신 클라이언트(HTTP 클라이언트)를 생성하는 수단.

<68> 상기 장치는 포틀릿 애플리케이션 세션 객체와 웹 애플리케이션 사이에서 통신을 하여 관련된 포틀릿들로부터 수신된 사용자 요청들을 웹 애플리케이션에 보내기 위하여 포틀릿 애플리케이션에서 포틀릿 애플리케이션 통신 클라이언트를 포함할 수 있다.

<69> 바람직하게는, 상기 포틀릿 애플리케이션은 포틀릿 애플리케이션 세션 객체와 관련된 각 포틀릿에 공통 키를 할당한다.

<70> 포틀릿 애플리케이션 세션 객체로 다중 웹 애플리케이션에 접속하도록 적합화된 사용자 세션 정보 테이블이 제공될 수 있는 장점이 있다.

<71> 본 발명의 또 다른 실시예는 웹 애플리케이션에 대하여 사용자에게 웹 포탈을 디스플레이하는 다음을 포함하는

장치를 제공하는데, 상기 웹 포탈은 사용자의 액세스가 가능한 정보를 서로 공유하는 복수의 관련된 포틀릿들을 디스플레이한다: 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하는 포탈 서버; 관련된 포틀릿들의 집합을 관리하기 위하여, 포탈 서버 상에서 동작하는 포틀릿 애플리케이션; 상기 포틀릿 애플리케이션은 다음을 포함한다: 웹 애플리케이션에 액세스하려는 사용자의 요청들에 대하여 포틀릿들을 시작하는 수단; 포틀릿들에 대한 포틀릿 애플리케이션 세션 객체를 관리하는 수단; 및, 포틀릿들을 포틀릿 애플리케이션 세션 객체와 관련시키기 위하여 사용자 요청들로부터의 파라미터들을 저장하기 위한, 포틀릿 애플리케이션 세션 객체가 제어하는 포틀릿 애플리케이션 세션 객체 데이터 저장부를 포함한다.

<72> 본 발명의 또 다른 태양은 웹 포탈에서 복수의 관련된 포틀릿들 사이에서 정보를 공유하는, 다음을 포함하는 방법을 제공한다: 복수의 관련된 포틀릿들 각각에 대하여 포틀릿 데이터 저장부에 액세스를 부여하는 단계; 복수의 관련된 포틀릿들 각각이 포틀릿 데이터 저장부에 데이터를 기록(write)하고 포틀릿 데이터 저장부로부터 저장된 데이터를 읽도록 허용하는 단계를 포함한다.

<73> 상기 방법은, 상기 관련된 포틀릿들이 데이터 프로세싱 시스템 상에서 동작하도록 적합화된 포틀릿 애플리케이션에 의하여 관리되고; 상기 포틀릿 데이터 저장부는 포틀릿 애플리케이션의 관련된 포틀릿들 사이에서 데이터의 교환을 허용하여, 데이터 저장부에서 관련된 포틀릿들에 의하여 데이터의 판독 및 기록을 제어하는 포틀릿 애플리케이션 세션 객체에 의하여 관리되는 포틀릿 애플리케이션 스토리지를 포함하는 시스템을 유리하게 제공할 수 있다.

<74> 본 발명의 또 다른 태양은 웹 포탈에서 관련된 다중 포틀릿들 사이의 정보 공유를 위한 다음을 포함하는 장치를 제공한다: 관련된 다중 포틀릿들을 관리하기 위한 포틀릿 애플리케이션; 포틀릿 애플리케이션 데이터 저장부; 포틀릿들이 서로 데이터를 교환할 수 있도록 관련된 다중 포틀릿들에 데이터 저장부에 대한 판독/기록 액세스를 부여하는 수단을 포함한다.

<75> 본 발명의 또 다른 태양은 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한 포탈 상에서 동작가능한 다음을 포함하는 포틀릿 서버를 제공한다: 관련된 다중 포틀릿들을 관리하기 위한 수단; 포틀릿 애플리케이션 세션 객체를 관리하기 위한 수단; 관련된 포틀릿들이 서로 데이터를 교환할 수 있도록 관련된 다중 포틀릿들에 데이터 저장부에 대한 판독/기록 액세스를 부여하는 포틀릿 애플리케이션 세션 객체가 관리하는 포틀릿 애플리케이션 데이터 저장부를 제공한다.

<76> 본 발명의 또 다른 태양은 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한 포탈 서버 상에서 동작이 가능한 다음을 포함하는 포틀릿(애플리케이션) 서버를 제공한다: 관련된 다중 포틀릿들을 관리하기 위한 수단; 포틀릿 애플리케이션 세션 객체를 생성 및 관리하기 위한 수단; 관련된 포틀릿들이 서로 데이터를 교환할 수 있도록 관련된 다중 포틀릿들에 데이터 저장부에 대한 판독/기록 액세스를 부여하는 포틀릿 애플리케이션 세션 객체가 생성하고 관리하는 포틀릿 애플리케이션 데이터 저장부를 제공한다.

<77> 유리하게는, 상기 포틀릿 애플리케이션은 공통 키를 포틀릿 애플리케이션 세션 객체와 관련된 각 포틀릿에 할당한다.

<78> 본 발명의 또 다른 태양은 사용자의 액세스가 가능한 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한 포탈 서버 상에서 동작이 가능한 다음을 포함하는 포틀릿 애플리케이션을 제공한다: 관련된 다중 포틀릿들을 관리하기 위한 포틀릿 애플리케이션 수단; 사용자에 대한 포틀릿 애플리케이션 세션 객체를 관리하기 위한 포틀릿 애플리케이션 수단; 포틀릿 애플리케이션 객체로의 액세스를 제어하기 위하여 관련된 각 포틀릿에 키를 부여하는 포틀릿 애플리케이션 수단을 제공한다.

<79> 본 발명의 또 다른 태양은 사용자의 액세스가 가능한 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한 포탈 서버 상에서 동작이 가능한 다음을 포함하는 포틀릿 애플리케이션을 제공한다: 관련된 다중 포틀릿들을 관리하기 위한 포틀릿 애플리케이션 수단; 사용자에 대한 포틀릿 애플리케이션 세션을 생성 및 관리하기 위한 포틀릿 애플리케이션 수단; 사용자를 위하여 포틀릿 애플리케이션 세션 객체에 대한 키를 생성 및 관리하는 포틀릿 애플리케이션 수단; 포틀릿 애플리케이션 객체으로의 액세스를 제어하기 위하여 관련된 각 포틀릿에 키를 부여하는 포틀릿 애플리케이션 수단을 제공한다.

<80> 유리하게는, 하나의 포틀릿 애플리케이션이 각 사용자에 할당되고, 하나의 키가 각 사용자에 대하여 각 포틀릿 애플리케이션에 대한 각 포틀릿 애플리케이션 객체들에 개별적으로 할당된다.

<81> 본 발명의 또 다른 태양은 웹 애플리케이션에 대하여 사용자에게 웹 포탈을 디스플레이하기 위한 다음을 포함하는 장치를 제공한다: 사용자에게 웹 애플리케이션으로의 액세스를 제공하는 웹 포탈을 동작하기 위한 포탈

서버; 포탈 서버 상에서 동작하기 위한, 관련된 포틀릿들의 집합을 관리하기 위한 포틀릿 애플리케이션; 사용자를 위한 관련된 포틀릿들에 대한 포틀릿 애플리케이션 세션 객체; 포틀릿 애플리케이션 세션 객체에 의하여 제어되는 포틀릿 애플리케이션 세션 객체 데이터 저장부; 관련된 포틀릿과 웹 애플리케이션 사이에서 통신하여 관련된 포틀릿들로부터 수신된 사용자 요청들을 웹 애플리케이션에 보내기 위한 포틀릿 애플리케이션 데이터 저장부에 링크된 포틀릿 애플리케이션 통신 클라이언트; 상기 통신 클라이언트는 통신 클라이언트가 웹 애플리케이션에 동기화되도록 관련된 포틀릿들로부터의 요청들을 저장 및 동기화하기 위한 요청 버퍼를 가진다.

<82> 바람직하게는, 상기 포틀릿 애플리케이션 통신 클라이언트는 네트워크 상의 요청들을 포함하는 정보를 웹 애플리케이션에 송신하고 웹 애플리케이션으로부터의 요청들에 대한 응답들을 포함하는 정보를 수신하도록 적합화된다.

<83> 본 발명의 또 다른 태양은 웹 애플리케이션에 대하여 사용자에게 웹 포탈을 디스플레이하기 위한 다음을 포함하는 장치를 제공한다: 웹 애플리케이션으로의 사용자의 액세스를 제공하도록 웹 포탈을 동작하기 위한 포탈 서버; 관련된 포틀릿들의 집합을 관리하기 위한, 포탈 서버 상에서 동작하기 위한 포틀릿 애플리케이션; 사용자를 위한 관련된 포틀릿들에 대한 포틀릿 애플리케이션 세션 객체; 포틀릿 애플리케이션 세션 객체에 의하여 제어되는 포틀릿 애플리케이션 세션 객체 저장부; 관련된 포틀릿들과 웹 애플리케이션 사이에서 통신하여 관련된 포틀릿들로부터 수신된 사용자 요청들을 웹 애플리케이션에 보내기 위하여 포틀릿 애플리케이션 데이터 저장부에 링크된 포틀릿 애플리케이션 통신 클라이언트; 상기 통신 클라이언트는 통신 클라이언트가 웹 애플리케이션에 나열되어 생성하도록 관련된 포틀릿들로부터의 요청들을 저장하고 나열하기 위한 요청 버퍼를 갖는다.

<84> 바람직하게는, 상기 포틀릿 애플리케이션 통신 클라이언트는 네트워크 상의 요청들을 포함하는 정보를 웹 애플리케이션 또는 웹 애플리케이션 서버에 송신하고 웹 애플리케이션으로부터의 요청들에 대한 응답들을 포함하는 정보를 수신하도록 적합화된다.

<85> 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하는데 적합화된 포탈 서버를 위한 본 발명의 또 다른 태양은 관련된 포틀릿들의 집합을 관리하기 위한, 포탈 서버 상에서 동작하는 포틀릿 애플리케이션을 갖는다. 상기 포틀릿 애플리케이션은 다음을 포함한다: 웹 애플리케이션에 액세스하려는 사용자의 요청들에 대하여 포틀릿들을 시작하는 수단; 포틀릿들에 대한 포틀릿 애플리케이션 세션 객체를 관리하는 수단; 및, 포틀릿들과 포틀릿 애플리케이션 세션 객체를 관련시키기 위하여 사용자의 요청들로부터의 파라미터들을 저장하기 위한, 포틀릿 애플리케이션 세션 객체가 제어하는 포틀릿 애플리케이션 세션 객체 데이터 저장부. 상기 장치는 다음을 포함한다: 관련된 포틀릿들과 웹 애플리케이션 사이에서 통신을 하여 관련된 포틀릿으로부터 수신된 사용자 요청들을 웹 애플리케이션에 보내기 위한 포틀릿 애플리케이션 데이터 저장부에 링크된 포틀릿 애플리케이션 통신 클라이언트(HTTP 클라이언트); 사용자 세션 정보를 대응하는 웹 애플리케이션의 세션에 매핑하기 위하여, 상기 포틀릿 애플리케이션 통신 클라이언트는 다음의 사용자 세션 정보의 세트로부터 선택된 정보를 포함하는 사용자 세션 정보를 저장하는 사용자 세션 정보 저장부(매핑 테이블)을 갖는다: 사용자 ID, 사용자 적격, 언어 우선순위(preference), 세션 타임아웃 정보, 세션 ID, 등.

<86> 바람직하게는, 상기 세션 타임아웃 정보는 포탈 서버 및 웹 애플리케이션의 세션 타임아웃 정보를 포함한다.

<87> 본 발명의 또 다른 태양은 포탈에서 관련된 포틀릿들의 집합을 관리하고, 웹 애플리케이션으로의 액세스를 사용자에 제공하는 서버 상에서 동작하기 위한, 포틀릿 애플리케이션을 제공한다; 상기 관련된 포틀릿들은 사용자 요청들로부터의 명령들 및 데이터를 포틀릿들에 저장하는 포틀릿 요청 파라미터 맵들을 갖는다; 사용자를 위한 관련된 포틀릿들에 대한 포틀릿 매플리케이션 세션 객체; 포틀릿 애플리케이션 세션 객체에 의하여 제어되는 포틀릿 애플리케이션 세션 데이터 저장부; 관련된 포틀릿들과 웹 애플리케이션 사이에서 통신하여 관련된 포틀릿들로부터 수신된 사용자 요청들을 웹 애플리케이션에 전달하기 위한 포틀릿 애플리케이션 데이터 저장부에 링크된 포틀릿 애플리케이션 통신 클라이언트(HTTP 클라이언트); 상기 통신 클라이언트는 상기 통신 클라이언트가 웹 애플리케이션에 데이터 및 명령들을 제공할 수 있도록 관련된 포틀릿들의 포틀릿 요청 파라미터 맵들로부터 요청들을 저장하기 위한 요청 버퍼를 갖는다.

<88> 본 발명의 또 다른 태양은 관련된 포틀릿들과 웹 애플리케이션 사이에서 통신하여 관련된 포틀릿들로부터 수신된 사용자 요청들을 웹 애플리케이션에 전달하기 위하여 포탈 애플리케이션 데이터 저장부에 링크된 포틀릿 애플리케이션 통신 클라이언트(HTTP 클라이언트)를 제공한다; 사용자 세션 정보를 대응하는 웹 애플리케이션의 세션에 매핑하기 위하여, 상기 포틀릿 애플리케이션 통신 클라이언트는 다음의 사용자 세션 정보의 세트로부터 선택된 정보를 포함하는 사용자 세션 정보를 저장하기 위한 사용자 세션 정보 저장부(매핑 테이블)을 갖는다: 사용자 ID, 사용자 적격, 언어 우선순위, 세션 타임아웃 정보, 세션 ID 등.; 상기 세션 타임아웃 정보는 포탈 서

버 및 웹 애플리케이션의 세션 타임아웃 정보를 포함한다.

<89> 바람직하게는, 상기의 예는 웹 애플리케이션이 포탈 서버보다 먼저 타임아웃하는 경우 사용자를 재인증함으로써 포탈 서버 및 웹 애플리케이션 사이의 세션 타임아웃을 매칭하기 위한 포틀릿 애플리케이션 통신 클라이언트에 대한 동기화 수단을 포함한다.

<90> 본 발명의 또 다른 태양은 사용자의 액세스가 가능한 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한 포탈 서버 상에서 동작이 가능한 포틀릿 애플리케이션을 제공하는데, 상기 포탈 서버는 관련된 포틀릿들이 서로 메시지를 보낼 수 있도록 하기 위하여 다음을 포함하는 메시징 수단을 제공한다: 관련된 다중 포틀릿들을 관리하기 위한 포틀릿 애플리케이션 수단; 콘텍스트 이름들을 설명하는 포틀릿 디스크립터를 각각 갖는 관련된 포틀릿; 콘텍스트 값을 정의하는 대응하는 콘텍스트 이름들을 갖는 포틀릿들의 협력 그룹들을 포함하는 상기 관련된 포틀릿들; 마스터 포틀릿과 하나 이상의 슬레이브 포틀릿을 각각 포함하는 상기 포틀릿들의 그룹; 상기 포틀릿들의 그룹 각각은 공통적인 콘텍스트 이름들을 공유한다; 마스터 포틀릿의 콘텍스트 값들의 변화를 마스터 포틀릿의 슬레이브 포틀릿들에 방송 통신(broadcasting communicating)하기 위한 포탈 서버내의 수단; 마스터 포틀릿의 콘텍스트 값들에 매칭하도록 슬레이브 포틀릿들의 콘텍스트 값들을 방송으로 변화하기 위한 위한 포탈 서버의 수단을 제공한다.

<91> 본 발명의 또 다른 태양은 사용자의 액세스가 가능한 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한 포탈 서버 상에서 동작 가능한 다음을 포함하는 포틀릿 애플리케이션을 제공하는데, 상기 포탈 서버는 포틀릿 리프레쉬 능력을 갖는다: 관련된 다중 포틀릿들을 관리하기 위한 포틀릿 애플리케이션 수단; 포틀릿 디스크립터를 각각 가지는 관련된 포틀릿; 포틀릿들에 대한 리프레쉬 우선순위 설명을 각각 포함하는 포틀릿 디스크립터; 포틀릿들의 협력 그룹들을 포함하는 관련된 포틀릿들; 마스터 포틀릿 및 하나 이상의 슬레이브 포틀릿을 포함하는 포틀릿들의 그룹; 리프레쉬 우선순위의 순서에 따라 포틀릿들을 리프레쉬하기 위한 포틀릿 애플리케이션 수단 내의 수단을 제공한다.

<92> 본 발명의 또 다른 태양은 사용자의 액세스가 가능한 웹 포탈에서 관련된 다중 포틀릿들을 호스팅하기 위한, 포탈 서버 상에서 동작 가능한 다음을 포함하는 포틀릿 애플리케이션을 제공하는데, 상기 포탈 서버는 포틀릿 리프레쉬 능력을 갖는다: 포틀릿들의 협력 그룹들을 포함하는 관련된 포틀릿들; 관련된 다중 포틀릿들을 관리하기 위한 포틀릿 애플리케이션 수단; 포틀릿 디스크립터를 각각 가지는 관련된 포틀릿; 포틀릿에 대한 리프레쉬 우선순위 설명, 및 그 그룹의 포틀릿이 멤버인 포틀릿들의 그룹에 대한 리프레쉬 설명 우선순위를 각각 포함하는 포틀릿 디스크립터; 마스터 포틀릿 및 하나 이상의 슬레이브 포틀릿을 각각 포함하는 상기 포틀릿들의 그룹; 우선순위의 순서에 따라 포틀릿들을 리프레쉬하기 위한 포틀릿 애플리케이션 수단 내의 수단; 그룹 리프레쉬 우선순위의 순서에 따라 포틀릿들의 협력 그룹들을 리프레쉬하기 위한 포틀릿 애플리케이션 수단 내의 수단을 제공한다.

<93> 상기 마스터 포틀릿들은 슬레이브 포틀릿들보다 더 높은 우선순위를 갖는다.

<94> 바람직하게는, 상기 포틀릿 애플리케이션은 그룹 우선순위의 순서에 따라 먼저 그룹들을 리프레쉬하고, 그 후에 우선 순위에 따라 각 그룹 내에서 리프레쉬한다.

<95> 본 발명의 또 다른 태양은 사용자에게 웹 애플리케이션에 대한 웹 페이지 세션을 디스플레이하는 다음을 포함하는 장치를 제공하는데, 상기 웹 페이지 세션은 복수의 관련된 협력 포틀릿들을 디스플레이한다: 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하기 위한 포탈 서버; 포탈 서버 상에서 동작하고, 관련된 포틀릿들의 집합을 관리하기 위한 포틀릿 애플리케이션; 룰 데이터베이스에 액세스하는 액세스 수단; 포틀릿들, 페이지들, 페이지 그룹들의 세트들을 사용자들에게 디스플레이하는 것을 제어하는 룰을 포함하는 룰; 사용자(정보 특성들)에 의하여 제공되는 정보에 기초하여 사용자에게 디스플레이되는 포틀릿들, 페이지들 및 페이지 그룹들의 세트를 선택하는 선택 수단을 제공한다.

<96> 본 발명의 또 다른 변형에서, 선택 수단은 플러거블 룰 엔진, 룰 데이터베이스, 및 포틀릿들, 페이지들, 페이지 그룹들을 선택하고 선택된 것들을 사용자에게 디스플레이하는데 룰을 적용하는 포틀릿 애플리케이션 집합 엔진을 포함한다.

<97> 본 발명의 또 다른 태양은 사용자에게 웹 애플리케이션에 대한 웹 페이지 세션을 디스플레이하는 다음을 포함하는 장치를 제공하는데, 상기 웹 페이지 세션은 서로 사용자의 액세스가 가능한 정보를 공유하는 복수의 관련된 협력 포틀릿들을 디스플레이한다: 웹 애플리케이션으로의 액세스를 제공하도록 웹 포탈을 동작하기 위한 포탈 서버; 포탈 서버 상에서 동작하고, 관련된 포틀릿들의 집합을 관리하기 위한 포틀릿 애플리케이션; 룰 데이터베

이스을 액세스하는 롤 액세스 수단; 사용자의 롤에 기초하여 포틀릿들, 페이지들, 페이지 그룹들의 세트들을 사용자들에게 디스플레이하는 것을 제어하는 룰을 포함하는 롤; 사용자의 식별된 롤에 기초하여 사용자에게 디스플레이될 포틀릿들, 페이지들 및 페이지 그룹들의 세트를 선택하는 롤 선택 수단을 제공한다.

<98> 본 발명의 다른 태양들은 다음을 포함하는 제품(article)을 제공한다: 컴퓨터 판독 가능한 신호 베어링(bearing) 매체; 상술한 본 발명 실시예들의 방법을 수행하도록 적합화된 매체 상에 저장된 컴퓨터 프로그램 코드 수단을 제공한다.

<99> 본 발명의 다른 태양들은 다음을 포함하는 제품을 제공한다: 컴퓨터 판독 가능한 신호 방위 매체; 상술한 본 발명 실시예의 장치를 구현하도록 적합화된 매체 상에 저장된 컴퓨터 프로그램 코드 수단을 제공한다.

<100> 상기 매체는 자기적, 광학적, 생물학적 및 원자력 데이터 스토리지 매체들로 구성되는 그룹들로부터 적절하게 선택될 수 있다.

<101> 상기 매체는 변조된 캐리어 신호일 수 있다.

<102> 상기 신호는 네트워크 상에 전송일 수 있다.

실 시 예

<122> 이 섹션은 본 발명의 바람직한 실시예들을 기술한다.

<123> **A.1 포탈 및 웹 애플리케이션들의 통합 가능성**

<124> 도 2는 웹 포탈 서버로 본 발명의 사용을 예시하는 본 발명의 바람직한 실시예를 예시한다.

<125> **A.1.1 포틀릿 애플리케이션 HTTP 클라이언트**

<126> (백엔드 웹 애플리케이션에 HTTP 요청들을 만드는) 포틀릿은 포틀릿 애플리케이션 HTTP 클라이언트(209)를 사용하여 백엔드 웹 애플리케이션 서버(210)에서 실행되는 백엔드 웹 애플리케이션에 HTTP 접속을 개방한다. 백엔드 웹 애플리케이션은 포틀릿 애플리케이션 HTTP 클라이언트(209)가 다중 요청 및 응답들, 쿠키 취급 및 한번의 사인온(SSO) 로직 상에서 세션 지원을 제공하는 것을 필요로 한다. 동일한 포틀릿 애플리케이션 내의 모든 포틀릿들은 동일한 포틀릿 애플리케이션 HTTP 클라이언트 객체(209)를 사용하여, 하나 이상의 백엔드 웹 애플리케이션들에 접속한다. 각 포틀릿 애플리케이션(204)에는 하나의 포틀릿 애플리케이션 HTTP 클라이언트(209)가 있다.

<127> **A.1.2 포틀릿 애플리케이션 세션**

<128> 포틀릿 애플리케이션 세션 객체(208)는 주어진 포틀릿 애플리케이션의 모든 포틀릿들에 공유될 수 있는 통합된 데이터 저장 객체이다. 이 객체는 각 사용자 및 각 포틀릿 애플리케이션에 존재한다. 포틀릿 애플리케이션 세션 객체(208)는 기반 구성을 제공하여, 주어진 포틀릿 애플리케이션의 다중 포틀릿들은 독립적인 (포틀릿 세션들(204,205,206)로 불리는) 사용자 세션들을 가질 것이고, 동일한 포틀릿 애플리케이션 세션을 공유하고, 하나의 웹 애플리케이션 세션을 가진 백엔드 애플리케이션 서버(210) 상의 웹 애플리케이션과 통신할 것이다.

<129> **A.1.3 포틀릿 애플리케이션 세션 콘텍스트**

<130> 포틀릿 애플리케이션 세션 콘텍스트는 각 사용자 및 각 포틀릿 애플리케이션의 정보를 제공한다. 이것은 동일한 포틀릿 애플리케이션(204, 203) 내의 모든 포틀릿들은 이제 그것들 사이의 공통 정보를 공유하는 방법을 가질 수 있다는 것을 의미한다.

<131> **A.1.4 세션 릴레이 메커니즘(320)**

<132> 세션 릴레이 메커니즘은 포탈 서버에 있는 원시 HTTP 세션으로부터 백엔드 HTTP 세션으로 포틀릿 애플리케이션의 HTTP 클라이언트에 의하여 생성되는 정보의 패싱(passing)을 가능하게 한다. 이 메커니즘은 다음의 기반 구성을 사용한다.

<133> **쿠키 테이블(305) 및 쿠키 룩업 키**

<134> 쿠키 테이블(205, 사용자 세션 정보 테이블)은 포탈 서버 쿠키들을 백엔트 웹 애플리케이션 세션 쿠키들로 매핑하기 위한 주요 엔터티이다. 포탈 서버로의 HTTP 요청들의 쿠키와 하나의 주어진 웹 애플리케이션으로의 포틀릿 애플리케이션 HTTP 클라이언트의 쿠키의 매핑 관계는 일대일이다. 그러나, 주어진 포틀릿 애플리케이션

HTTP 클라이언트는 독립된 세션들을 유지하는 각 웹 애플리케이션으로 서로 다른 웹 애플리케이션들에 HTTP 요청들을 만들 수 있다. 그것에 관하여, 포탈 서버 세션 쿠키 및 백엔드 웹 애플리케이션들의 쿠키 사이의 매핑은 (다중 백엔드 웹 애플리케이션 서버들 때문에) 일대다(one to many)일 수 있다.

<135> 도 13은 이 매핑을 나타내는데, 여기서 몇 개의 아이템들이 예시된다:

<136> RQ1: 사용자의 에이전트(브라우저)의 HTTP 요청으로부터 포탈 서버로의 쿠키

<137> RQA: 포틀릿 HTTP 애플리케이션 클라이언트의 HTTP 요청으로부터 웹 애플리케이션 A로의 쿠키

<138> RQB: 포틀릿 HTTP 애플리케이션 클라이언트의 HTTP 요청으로부터 웹 애플리케이션 B로의 쿠키

<139> 포틀릿 애플리케이션 HTTP 클라이언트(209)는 이 테이블을 사용하여 백엔드 웹 애플리케이션 서버(210)에서 실행되는, 백엔드 웹 애플리케이션으로 매칭하는 쿠키를 룩업(look up)한다.

<140> 쿠키 매핑 테이블(305)의 존재는 포탈 서버 세션이 만료하는 경우 백엔드 웹 애플리케이션 세션의 자동 만료(automatic expiration)를 가능하게 한다.

<141> **쿠키 룩업 키**

<142> 포틀릿 애플리케이션 HTTP 클라이언트(209)는 각 포틀릿 애플리케이션마다 생성된다. 쿠키 룩업 키는 동일한 포틀릿 애플리케이션 내의 모든 포틀릿들에 액세스가 가능한 포탈 애플리케이션 세션 객체에 저장된다. 쿠키 룩업 키는 백엔드 웹 애플리케이션의 HTTP 세션으로 포탈 서버의 HTTP 세션을 매칭하는 것을 담당한다.

<143> 쿠키 룩업 키의 사용은 동일한 HTTP 클라이언트 키를 공유하는 주어진 포틀릿 애플리케이션의 모든 포틀릿들이 현재 로그인된 사용자에 대한 백엔드 웹 애플리케이션 정보의 세트를 정확하게 검색하고 포워딩하는 것을 가능케 하여, 동일한 포틀릿 애플리케이션 내의 모든 포틀릿들이 동기화되어, 사용되고 있었던 백엔드 웹 애플리케이션을 업데이트할 수 있다. 종단 사용자는 다중 포틀릿들을 통하여 백엔드 웹 애플리케이션의 통합된 뷰를 본다는 것이 효과이다.

<144> **포틀릿 요청 파라미터 맵**

<145> 포틀릿 요청 파라미터 맵(308)은 각 포틀릿, 각 포틀릿 서버 세션 마다 생성된 공유 애플리케이션 세션 데이터 저장부에 저장된 메모리 객체에 존재한다. 포틀릿 요청 파라미터 맵은 인입되는 사용자 요청으로부터의 모든 요청 파라미터들을 특정 포틀릿에 저장하는데 사용된다.

<146> **A.2 포틀릿들의 동적 콘텐트 동기화**

<147> **A.2.1 동적 콘텍스트 정의 템플릿**

<148> 도 5는 백엔드 웹 애플리케이션으로의 포탈의 통합을 예시한다. 도 5에 대한 참조는 다음을 위하여 유용할 것이다:

<149> 동적 콘텍스트 정의 템플릿(503)은 각 동적 콘텍스트 그룹에 대하여 다음을 정의한다.

<150> - 콘텍스트 및 콘텍스트의 타입(이전의 예에서, 콘텍스트는 계정 ID이다.)

<151> - 정의된 콘텍스트의 값을 변화시킬 수 있는 마스터 포틀릿

<152> - 정의된 콘텍스트가 변화되는 경우 통지될 슬레이브 포틀릿(들)

<153> - 콘텍스트 변화 통지시 슬레이브 포틀릿(들)의 등록된(registered) 응답(또는 액션)

<154> - 옵션으로 슬레이브 포틀릿들의 리프레쉬 시퀀스를 정의한다.(마스터는 항상 주어진 그룹 내에서 최초로 리프레쉬된다.)

<155> 하나의 동적 콘텍스트 정의 템플릿(503)은 하나 이상의 동적 콘텍스트 그룹(들)을 포함할 수 있다. 그러나 각 동적 콘텍스트 그룹은 단지 다음을 가질 수 있다:

<156> - 하나의 마스터 포틀릿

<157> - 하나의 정의된 콘텍스트

<158> - 하나 이상의 슬레이브 포틀릿들

<159> 주석: 주어진 포틀릿은 각 그룹에서 서로 다른 롤들을 가지고 하나 이상의 동적 콘텍스트 그룹들에 참가할 수 있다.

<160> **A.2.2 동적 콘텍스트 포틀릿 그룹핑 생성 도구**

<161> 이 도구(501)는 동적 콘텍스트 정의 템플릿(503)에서 리드인(read in)하고, 그에 대응하여 포틀릿 배치 디스크립터들(502)를 업데이트함으로써 특정된 정의에 따라서 모든 동적 콘텍스트 그룹들을 위한 동적 콘텍스트 마스터 포틀릿 및 슬레이브 포틀릿들을 생성한다.

<162> **A.2.3 동적 콘텍스트 그룹**

<163> 동적 콘텍스트 그룹은 동일한 콘텍스트를 공유하고 하나의 동적 콘텍스트 그룹 하에서 그룹핑되는 포틀릿들의 서브세트이다. 주어진 포틀릿은 하나 이상의 동적 콘텍스트 그룹에 속할 수 있다.

<164> 동적 콘텍스트 그룹의 정의 다큐먼트 인스턴스(504)는 특정한 동적 콘텍스트 그룹의 동적 콘텍스트를 정의하는데 사용된다.

<165> 동적 콘텍스트 마스터 포틀릿

<166> 동적 콘텍스트 마스터 포틀릿은

<167> - 콘텍스트 상태 변화를 검출 및

<168> - 모든 슬레이브 포틀릿들에 콘텍스트 상태 변화를 통지하는 것을 담당한다.

<169> 동적 콘텍스트 슬레이브 포틀릿(들)

<170> 동적 콘텍스트 슬레이브 포틀릿들은 다음을 수행한다:

<171> - 마스터 포틀릿에 의하여 통지된 콘텍스트 변화를 위한 풀링(pulling)

<172> - 콘텍스트 변화의 통지시 대응하는 백엔드 웹 애플리케이션을 지향하는 등록된 액션을 수행

<173> 동적 콘텍스트 모델들

<174> 서로 관련된 포틀릿들을 위하여 사용될 수 있는 동적 콘텍스트 모델들에는 2가지 타입들이 있다:

<175> **A.2.4 동기 모델**

<176> 도 14에 나타나는 동기 모델에서, 마스터 포틀릿(101)은 동적 콘텍스트 마스터 포틀릿의 콘텍스트 상태 변화를 슬레이브들(1701~1703)에 알려준다. 모든 슬레이브들은 마스터의 콘텍스트 상태 변화와 동기화하기 위하여 미리 정의된 응답에 기초한 액션들을 수행할 것이다.

<177> 동기 모델 도면

<178> **A.2.5 연쇄 모델**

<179> 도 1에 표시된 연쇄 모델에서, 마스터 A(101)의 상태 변화가 슬레이브 A(102)의 응답 액션을 초래하고, 슬레이브 A는 또한 마스터 포틀릿 B이며, 콘텍스트 B의 상태 변화를 가져오고, 슬레이브 B(103)의 콘텍스트 변화 응답을 초래하며, 슬레이브 B는 또한 동적 콘텍스트 그룹 C의 마스터 포틀릿이고, 슬레이브 C의 액션 응답을 초래한다.

<180> **A.2.6 포틀릿 트랜잭션 관리자:**

<181> 도 15를 참조하는 시퀀스 인식 포탈 집합 엔진 확장에서, 포틀릿 트랜잭션 관리자(1802)는 포틀릿 요청들, 응답들 및 세션들의 생성을 포함하는 포틀릿들의 런타임 리프레쉬 시퀀스를 관리하는 일을 담당하는 구성이다.

<182> 1. 어떤 포틀릿 애플리케이션에 대하여 리프레쉬될 최초의 포틀릿은 주어진 사용자를 위하여 모든 포틀릿들 사이에서 최초로 리프레쉬될 포틀릿으로 정의된다. 주어진 페이지 내의 포틀릿들의 리프레쉬 시퀀스를 정의하는 기존의 메커니즘은 없다.

<183> 따라서, 우리는 런타임에서 동적으로 마스터 포틀릿을 식별할 수 있는 일정한 로직이 필요하다. 본 발명에서 우리는 각 포틀릿이 리프레쉬되는 모든 경우에 마크를 만드는 간단한 스크래치보드(scratchboard)를 사용한다. 최초에 포틀릿은 그것이 제1 또는 마스터 포틀릿임을 표시하는 마크를 스크래치보드에 만든다. 이 리스트상에 마크를 만든 이후의 포틀릿은 이미 다른 포틀릿이 스크래치보드 위에 마크를 만들었고 그 포틀릿이 마스터 포틀

릿이 아니라는 것 등을 알 수 있다. 다음에 포털 페이지는 리프레쉬되고, 이 리스트 상에 더블 마크를 만든 제1 포틀릿은 마스터 포틀릿이 된다. 그리고, 마스터 포틀릿은 모든 다른 포틀릿들의 마크를 제거함으로써 이 리스트 및 다음 요청을 위한 더블 마크들의 하나를 재개시한다. 이 알고리즘은 포틀릿들의 포탈 서버로부터 요청이 들어오는 모든 경우 우리가 동적으로 마스터 포틀릿을 검출할 수 있게 한다.

<184> 제1 포틀릿이 리프레쉬된 후에, 트랜잭션 관리자는 동적 콘텍스트 그룹의 마스터 및 슬레이브 포틀릿 매핑에서 미리 정의된 시퀀스로 다른 포틀릿들을 리프레쉬한다.

<185> **2. 시퀀스 소터(sorter):** 시퀀스 소터 모듈(1804)은 리프레쉬 시퀀스 순서로 포틀릿들을 소팅(sorting)하는데 사용된다. 시퀀스 소터는 포틀릿 배치 디스크립터를 사용하여 각 포틀릿의 리프레쉬 순서를 식별하고 요청 디스패칭 엔진에 대하여 포틀릿들을 소팅 아웃한다.

<186> **3. 시퀀스 인식 요청 디스패칭 엔진 확장:** 이 엔진(1805)은 요청들을 포틀릿에 디스패칭하고 포탈 집합 엔진에 우선하는데 사용된다. 엔진의 롤은 상거래 포탈 애플리케이션에서 모든 포틀릿에 대한 포틀릿 세션 뿐만 아니라, 관련된 포틀릿 요청 및 응답 객체들을 구성하는 것이다. 엔진은 트랜잭션 관리자에 의하여 실제적으로 포틀릿들을 리프레쉬하는데 사용된다.

<187> **4. 트랜잭션 관리자 캐싱 유닛:** 트랜잭션 관리자 캐싱 유닛(1806)은 포틀릿들이 응답 디스패칭 엔진에 의하여 리프레쉬되면서 포틀릿들에 의하여 생성된 응답들을 트랜잭션 관리자(1802)가 캐싱하는데 사용된다. 포탈 집합 엔진이 포틀릿 리프레쉬을 요청하는 경우, 캐싱된 응답들이 트랜잭션 관리자에 의하여 그것에 다시 송신하는 것이 필요하다. 이것은 인입하는 포털 요청 각각에 대한 더블 리프레쉬의 문제를 예방한다.

<188> **A.3 룰 기반 및 롤 기반 집합**

<189> 도 11은 본 발명의 바람직한 실시예의 룰 기반의 동적 집합 구성 구조 맵을 예시한다. 예시된 실시예의 구성들 및 구성들의 동작 설명은 다음과 같다:

<190> **포탈 리소스 변환 모듈**

<191> 포탈 리소스 변화 모듈(1015)은 포틀릿들, 페이지들 및 페이지 그룹들을 포함하는 포탈 리소스들의 세트를 외부 룰 엔진(1022)에 의하여 파싱(parsing) 및 처리될 수 있는 형태로 변환하는 것을 담당한다.

<192> 룰 **데이터베이스**

<193> 룰 데이터베이스(1001)는 포탈 집합 엔진(1006)을 위하여 비지니스 관리자가 정의한 룰들을 홀딩(holding)한다.

<194> **사용자 리소스 변화 모듈**

<195> 사용자 리소스 변환 모듈(1013)은 사용자 리소스들 및 다양한 사용자의 특성들을 외부 룰 엔진에 의하여 파싱 및 수행될 수 있는 형태로 변환한다.

<196> **플러거블 룰 엔진**

<197> 룰 엔진(1022)은 (본 발명의 실시예에서) 웹스피어™ 개인화 엔진(WEBSPHERE PERSONAIZATION ENGIN)과 같은 외부의, 플러거블(pluggable) 룰 엔진이고, 동적 룰 파싱 및 실행에 사용된다. 엔진의 실행은 비지니스 사용자 및 현재 사용자의 특성들에 의하여 정의된 비지니스 룰들에 기초하여 사용자가 볼 수 있는 포탈 리소스들의 세트를 생성한다.

<198> **포탈 롤 기반 개인화 엔진**

<199> 포탈 롤 기반의 개인화 엔진(1008)은 사용자 조직의 멤버쉽에 기초하여 사용자에게 액세스가 허용되지 않는 포탈 리소스들의 리스트 및 사용자에게 액세스가 허용되는 포탈 리소스들의 리스트를 추출하는데 사용되는 롤 기반 리소스 선택 모듈이다.

<200> 롤 기반 엔진(1008)은 롤 데이터베이스(1007)에 액세스함으로써 먼저 사용자의 조직을 본다. 사용자의 조직이 한번 판단되면, 사용자의 롤은 그 조직의 롤과 동일한 것이라고 생각된다. 그 후에 롤 기반 개인화 엔진(1008)은 이 조직에 비지니스 사용자의 액세스가 가능 및 불가능한 것으로 정의되었던 리소스들의 리스트를 추출한다. 이 리스트가 한번 판단 되었으면, 이 모듈은 또 다른 프로세싱을 위하여 포탈 집합 엔진의 집합화된 리소스 변환 서브시스템에 리스트을 포워딩(forwarding)한다.

<201> **룰 DB**

<202> 롤 DB(1007)는 포탈 서버를 위한 조직 데이터를 홀딩한다. 롤 DB는 다양한 사용자들에 대한 조직의 멤버쉽 정보 및 롤들에 기초하여 조직 멤버들이 액세스 가능 또는 불가능한 포탈 리소스들의 리스트를 홀딩한다.

<203> **리소스 변환 서브시스템을 집합화한 포탈 집합 엔진**

<204> 이 모듈(1004)은 룰 및 롤 기반 개인화 엔진들에 기초하여 현재의 사용자가 볼 수 있는 포탈 리소스들(이것은 포틀릿들, 페이지들, 및 페이지 그룹들을 포함한다)의 마스터 리스트를 생성하는 것을 담당한다. 이 모듈은 또한 실제적인 포탈 집합 엔진을 위한 어댑터(adapter)이다. 모듈의 역할은 마스터 리스트를 생성하는 것뿐만 아니라, 종단 사용자를 위한 최종 웹 사이트 생성을 위하여 실제적인 포탈 집합이 액세스 가능한 형태로 마스터 리스트를 변환하는 것이다.

<205> **파트 B: 동작 설명**

<206> B.1 포탈 및 웹 애플리케이션의 통합 인에이블링 설명

<207> B.1.1 전체적인 집합 구조도 및 흐름도

<208> 도 2, 도 3 및 도 4는 각각 포탈을 가진 웹 애플리케이션 통합, 통합 구조도, 및 통합 흐름도를 나타낸다.

<209> **B.1.2 상세한 설명**

<210> 도 2를 참조하면, 백엔드 웹 애플리케이션이 포탈 서버로 통합되는 경우, 백엔드 웹 애플리케이션(221)은 포틀릿들을 통하여 포탈 서버(201)로부터 요청들을 수신한다. 백엔드 웹 애플리케이션(221)은 응답들을 만드는 포틀릿에 응답들을 다시 보낸다.

<211> 웹 애플리케이션(221)로부터의 응답은 포탈 서버(201)의 포틀릿을 통하여 포틀릿에 액세스하는 사용자에게 렌더링(rendering)된다.

<212> 포탈 애플리케이션 HTTP 클라이언트(209)의 구현으로, 백엔드 웹 애플리케이션에 대한 다중 요청들 및 응답들이 백엔드 웹 애플리케이션에 의하여 응집한 세션들로서 인식된다. 포탈 애플리케이션 HTTP 클라이언트(209)는 백엔드 웹 애플리케이션(221)로의 HTTP 통신 접속들을 개방하는데 사용된다. 백엔드 웹 애플리케이션은 포탈 애플리케이션 HTTP 클라이언트(209)가 세션 지원, 쿠키 취급 및 한번의 사인온(SSO) 능력을 제공하도록 요구한다. 포탈 애플리케이션 HTTP 클라이언트(209)로, 포틀릿들은 웹 애플리케이션과 효율적으로 통신할 수 있다. (포틀릿 애플리케이션(205)와 같은) 포틀릿 애플리케이션 내의 모든 포틀릿들은 백엔드 웹 애플리케이션(221)의 하나의 포틀릿 애플리케이션 세션 객체(211)로 액세스하는 것을 필요로 하는데, 이것은 포탈 애플리케이션 HTTP 클라이언트(209)가 동일한 포털 애플리케이션 내의 모든 포틀릿들에 의하여 공유되어야 함을 의미한다.

<213> 이와 같은 공유가 가능하기 위하여, 우리는 주어진 포탈 애플리케이션 내의 모든 포틀릿들이 공유할 수 있는 통합된 세션 객체가 필요하다고 판단했다. 그러한 객체를 제공하기 위하여 본 발명은 포틀릿 애플리케이션 세션 객체(208)을 제공한다. 포틀릿 애플리케이션 세션 객체(208)는 상거래 포틀릿 애플리케이션에 의하여 생성되는 객체이다. 포틀릿 애플리케이션 세션 객체(208)는 (포틀릿 애플리케이션 1 내의 포틀릿들 204, 205, 206, 및 207과 같은) 주어진 포틀릿 애플리케이션 내의 모든 포틀릿이 액세스할 수 있다. 포틀릿 애플리케이션 세션 객체(208)가 없으면, 주어진 포탈 애플리케이션 내의 다중 포틀릿들은 모두 독립된 사용자 세션들을 갖게 되고 세션에 관련된 정보를 공유할 수 없을 것이다.

<214> 포틀릿 애플리케이션 HTTP 클라이언트(209)는 포틀릿 애플리케이션 세션(208)에 저장되고, 동일한 포틀릿 애플리케이션 내의 포틀릿들 사이에서 포틀릿 애플리케이션 세션 객체를 공유하는 것을 가능하게 만든다. 이 포틀릿 애플리케이션 세션 객체가 없다면, 포틀릿들이 백엔드 상의 하나의 웹 애플리케이션 세션과 통신하는 것이 가능하지 않을 것이다.

<215> 포탈 애플리케이션 세션 객체(208) 내에 저장되는 모든 데이터는 포탈 애플리케이션 세션 콘텍스트를 표현하고 각 사용자 및 각 포탈 애플리케이션 마다 존재한다.

<216> 포틀릿 애플리케이션 HTTP 클라이언트(209)가 백엔드 웹 애플리케이션(221)에 대한 모든 세션 정보를 홀딩한 후에는, 세션 정보가 도 3에 나타난 세션 릴레이 메커니즘(320)에 대한 기본으로 사용된다.

<217> 세션 릴레이는 전체 포탈 서버(201)에 특정된 (언어 정보, 사용자 에이전트 정보 등과 같은) 세션 정보가 백엔드 웹 애플리케이션(221)의 세션 정보로 릴레이되는 것을 가능케 한다. 이것은 송신된 원시 요청에 포함되는 모든 요청들에 따르는 데이터 표현을 백엔드 웹 애플리케이션(221)이 사용자에 의하여 포탈 서버에 전달할 수

있다는 것을 의미한다.

<218> 예를 들면, 사용자가 "프랑스어"로 설정된 디폴트 언어 위치를 가진, WAP(무선 애플리케이션 프로토콜) 가능한 이동 장치를 사용하여 포탈에 액세스하는 경우, 포탈 서버(201)로의 원시 HTTP 요청은 ITS 언어 파라미터를 "프랑스어"로, 또한 HTTP 헤더의 사용자-에이전트 필드를 "WAP"으로 설정하게 할 것이다. 세션 릴레이 메커니즘(320)은 이 정보를 웹 애플리케이션(221)로 릴레이하고 웹 애플리케이션은 프랑스어의 사용자 이동 장치에 디스플레이하기 적합한 프랑스어로 응답을 리턴한다. 세션 릴레이가 존재하지 않는 경우, 웹 애플리케이션은 디폴트 장치(예컨대, 인터넷 브라우저)에 적합한 디폴트 언어(예켠대, 영어)로 정보를 리턴할 것이다. 이 경우 사용자의 이동 장치와 호환되지 않으므로 ,사용자는 검색된 정보를 볼 수 없을 것이다.

<219> 도 3의 구조도에 있는 요소들을 참조하여, 열거한 단계들에 의하여 도 4의 프로세스 단계들이 표시될 것이다.

<220> 1. 단계 401, 컴퓨터 마우스를 사용하여 링크 또는 사용자의 웹 브라우저 상의 포틀릿에 디스플레이된 객체를 클릭함에 의한 인스턴스에 대하여, 사용자는 웹 포탈과 상호 작용한다. 각 포틀릿은 자신의 포틀릿 세션(310)(포틀릿 세션은 종래 기술임)을 가진다. 사용자 상호 작용의 일부로서, 원격 요청(306)은 백엔드 웹 애플리케이션(307)에서 만들어지고 있다.

<221> 2. 단계 403, 포틀릿 세션 내의 모든 파라미터들을 백엔드 웹 애플리케이션에 정확하게 패싱하기 위하여, 각 포틀릿 요청의 파라미터 리스트는 포틀릿 요청 파라미터 맵(#8, 308)에 저장된다. 파라미터들은 원격 백엔드 요청에 패싱된다.

<222> 3. 단계 404, 상거래 포틀릿은 HTTP 클라이언트 키(301)를 사용하여 포틀릿 애플리케이션 데이터 저장부(#4, 302)에 액세스함으로써 이미 기존의 포틀릿 애플리케이션 세션 객체(208) 및 포틀릿 애플리케이션 HTTP 클라이언트(303)가 존재하는지를 판단한다. 단계 405, 발견되지 않는 경우, 동일한 포틀릿 애플리케이션 내의 모든 포틀릿들을 위하여 새로운 것을 생성한다. (단계 407, 발견되는 경우, 기존의 것이 대신 사용될 것이다.)

<223> 4. 단계 406, 원시 포틀릿 세션으로부터의 각 사용자 자격이 쿠키 테이블(305)에 저장된다.

<224> 5. 단계 408, 쿠키 테이블(305)로부터의 사용자 자격 및 포틀릿 요청 파라미터 맵(308)에 미리 저장된 파라미터는 백엔드 웹 애플리케이션에 대한 새로운 HTTP 요청을 구성한다.

<225> 6. 단계 409, 원격 웹 애플리케이션(307)에 호출한다.

<226> 7. 단계 410, 원격 웹 애플리케이션(307)은 디스플레이할 포틀릿을 위한 호출에 대하여 응답을 리턴한다.

<227> **B.2 포틀릿들의 동적 콘텍스트 동기화**

<228> **B.2.1 개발 시간(development time) 설명**

<229> 백엔드 웹 애플리케이션과 함께하는 포탈 통합의 구조도를 나타내는 도 5를 참조하면, 포탈 개발자는 동적 콘텍스트 포틀릿 그룹핑 도구(501)을 사용하여 새로운 동적 그룹 정의 인스턴스(504)를 생성한다는 것을 알 수 있다. 이 인스턴스는 관련된 포틀릿들의 그룹이고 어떤 포틀릿들이 슬레이브이고 어떤 포틀릿이 슬레이브들의 마스터인지를 정의한다. 동적 그룹 정의의 필요한 요소는 동적 콘텍스트 그룹 정의 템플릿(503)에서 특정된다.

<230> 사용자는 동일한 도구(501)를 사용하여 기존의 동적 콘텍스트 그룹 정의를 업데이트한다.

<231> 사용자가 가장 최근의 동적 콘텍스트 그룹 정의를 제공한 후에, 동적 콘텍스트 포틀릿 그룹핑 도구(501)는 관련된 포틀릿 애플리케이션 배치 디스크립터(502)를 업데이트하여 그룹에 정의된 관계들을 반영한다.

<232> 도 6을 참조하면, 상기 프로세스의 포탈 통합 단계들을 표현하는 흐름도를 명확히 볼 수 있을 것이다.

<233> 사용자가 동적 콘텍스트 그룹을 생성(608) 또는 업데이트(609)하기를 원하는 경우, 사용자는 그룹핑 도구(501, 도 5)를 이용할 수 있다.

<234> 단계 601, 동적 콘텍스트 그룹핑 도구는 동적 콘텍스트 그룹 정의 템플릿(503)에 특정된 것에 기초하여 사용자 입력을 촉구하거나, 또는 동적 콘텍스트 그룹핑 도구를 업데이트하는 경우 기존의 동적 콘텍스트 그룹 인스턴스를 읽어서, 정의 템플릿(503)으로 인스턴스를 확인한다.

<235> 단계 603, 사용자는 동적 콘텍스트 그룹을 정의 또는 업데이트하는데 필요한 정보를 특정한다.

<236> 단계 605, 동적 콘텍스트 그룹 인스턴스(504)가 생성된다.

<237> 단계 606, 모든 관련된 포틀릿들의 배치 디스크립터가 업데이트된다.

<238> 동적 콘텍스트 그룹핑

<239> 도 7은 포틀릿들에 대한 동적 콘텍스트를 예시한다. 동적 그룹(701)은 마스터 포틀릿(704), 슬레이브 포틀릿(705), 및 슬레이브 포틀릿(707)을 포함한다.

<240> 그룹(703)은 마스터 포틀릿(705), 슬레이브 포틀릿(706), 및 슬레이브 포틀릿(707)을 포함한다.

<241> 동적 그룹(702)은 마스터 포틀릿(704) 및 슬레이브 포틀릿(708)을 포함한다.

<242> 포틀릿 애플리케이션 내의 포틀릿들이 표현하는 데이터가 백엔드 웹 애플리케이션 레벨에 동기화되면, 포틀릿들은 단지 웹 애플리케이션으로부터 데이터를 검색함에 의하여 데이터의 대등한(coordinated) 뷰를 전달한다. 그러나, 모든 포틀릿의 상호 작용들은 백엔드 웹 애플리케이션에서 변화를 초래하지 않는다. 동적 콘텍스트는 '유리(glass)에서' 동기화를 제공한다. 콘텍스트의 변화가 서로 다른 쿼리를 필요로 하는 경우 가장 효율적이다. 예컨대, 계정 리스트로부터 다른 계정을 선택하는 것은 선택된 계정과 함께 리프레쉬되는 요금청구서(invoice) 정보의 디스플레이를 필요로 한다.

<243> 종래 기술 시스템에서 포틀릿들은 일반적으로 서로 독립적이었다. 본 발명은 포틀릿들 서로의 관계들을 매핑하고 포틀릿 애플리케이션의 배치 및 구성시에 그것들의 의존관계를 명확히 하는 방법 및 장치를 제공한다. 포틀릿들은 변화될 필요가 없다.

<244> 포틀릿들 사이의 의존 관계는 마스터 및 슬레이브의 관계가 정의된 동적 콘텍스트 관계 템플릿(503)에 정의될 수 있다.

<245> 동적 콘텍스트 관계 템플릿(503)은 바람직하게 다음을 정의하는 XML 데이터 표현으로 인코딩된다:

<246> - 동적 콘텍스트 그룹을 구성하는 포틀릿들의 서브세트

<247> - 동적 콘텍스트 그룹의 마스터 포틀릿

<248> - 동적 콘텍스트 그룹의 슬레이브(들) 포틀릿

<249> - 슬레이브(들)가 콘텍스트 상태 변화시 수행해야 하는 슬레이브 액션

<250> - 동적 콘텍스트 그룹의 모든 구성들이 공유하는 콘텍스트

<251> 통적 콘텍스트 그룹 정의 인스턴스의 예는 다음과 같다:

<252> < DynamicContextGroup >

<253> < DynamicContextGroupName > OrderRelatedPortletGroup

<254> < /DynamicContextGroupName >

<255> < DynamicContextMasterPortlet >

<256> OrderItems

<257> < /DynamicContextMasterPortlet >

<258> < DynamicContext > itemName

<259> < /DynamicContext >

<260> < DynamicContextSlavePortlet >

<261> < DynamicContextSlavePortletName > UPSTracking

<262> < /DynamicContextSlavePortletName >

<263> < SlavePortletAction >

<264> http://inventoryserver.com/inStock/

<265> < /SlavePortletAction >

<266> < /DynamicContextSlavePortlet >

<267> < /DynamicContextGroup >

<268> < DynamicContextGroup >

<269> < DynamicContextGroupName > StockInventoryPortletGroup

<270> < /DynamicContextGroupName >

<271> < DynamicContextMasterPortlet >

<272> InStockInventory

<273> < /DynamicContextMasterPortlet >

<274> < DynamicContext > itemSKUnumber

<275> < /DynamicContext >

<276> < DynamicContextSlavePortlet >

<277> < DynamicContextSlavePortletEame > OrderedItems

<278> < /DynamicContextSlavePortletName >

<279> < SlavePortletAction >

<280> http://myserver.com/lastOrdered/

<281> < /SlavePortletAction >

<282> < /DynamicContextSlavePortlet >

<283> < /DynamicContextGroup >

<284> 하나의 동적 콘텍스트 그룹 정의는 하나의 인스턴스라는 것을 주목하라. 그러나, 다중 동적 콘텍스트 그룹 정의들은 3개의 포틀릿들로 구성되는 포틀릿 애플리케이션에서 2개의 포틀릿 세트들로 상기에서 정의된 다중 인스턴스들을 정의하기 위하여 하나의 파일로 통합될 수 있다.

<285> 제1 동적 콘텍스트 그룹에서, 포틀릿들 사이에 공유되는 동적 콘텍스트는 itemName이고, OrderedItems로 명명된 포틀릿은 동적 콘텍스트 마스터 포틀릿으로 서비스되고, 포틀릿 UPSTracking 및 InStockInventory는 동적 콘텍스트 슬레이브 포틀릿들로 서비스된다.

<286> 제2 동적 콘텍스트 그룹에서, 포틀릿들 사이에 공유되는 동적 콘텍스트는 itemSkuNumber이고, InstockInventory로 명명된 포틀릿은 동적 콘텍스트 마스터 포틀릿으로 서비스되고, 포틀릿 OrderedItems는 동적 콘텍스트 슬레이브 포틀릿들로 서비스된다.

<287> 각 동적 콘텍스트 마스터 포틀릿은 사용자 HTTP 요청을 관찰하고 동적 콘텍스트를 찾는다. 동적 콘텍스트가 요청에서 발견되는 경우, 동적 콘텍스트 포틀릿은 (HTTP 요청에서 이름 및 값 쌍의 파라미터인) 동적 콘텍스트를 슬레이브 포틀릿들에 송신한다.

<288> 예를 들면, 포틀릿 OrderedItems가 "PentiumIV"로 설정된 특성 ItemName을 가진 요청을 받은 경우, 포틀릿 OrderedItems는 동적 콘텍스트를 포틀릿 UPSTracking 및 InStockInventory에 보내어 값 "PentiumIV"를 가진 콘텍스트 ItemName은 이제 동적 콘텍스트에 설정되었음을 통지한다.

<289> 각 동적 콘텍스트 슬레이브 포틀릿은 동일한 동적 콘텍스트 그룹의 모든 슬레이브 포틀릿들에 대한 마스터의 통지를 듣는다. 마스터의 통지를 받으면, 동적 콘텍스트를 특성 'SlavePortletAction' 하의 동적 콘텍스트 그룹 정의 인스턴스에서 정의된 액션 URL에 동적 콘텍스트를 추가함으로써 대응하는 슬레이브 액션을 불러온다.

<290> 예를 들면, inStockInventory 포틀릿이 동적 콘텍스트 타입 "ItemName' 및 값 "PentiumIV"를 가진 OrdeItems 포틀릿으로부터 동적 콘텍스트를 받은 경우, inStockInventory 포틀릿은

<291> http://inventoryserver.com/inStock/itemName=PentiumIV URL로부터 데이터를 검색할 것이다.

<292> 동적 콘텍스트 그룹 정의 템플릿 코딩의 예는 다음과 같다:

```
<293> < xsd : schema
<294>    xmlns: xsd="http://www. w3.org/2001/XMLSchema"
<295>    xmlns : cep=
<296> "http://www. ibm. com/WebsphereCommerceEnabledPortal/DynamicContex
<297> tGroupDefinitionSchema" >
<298> < annotation >
<299> < documentation xml: lang="en" >
<300>       Schema for Websphere Commerce Enabled Portal Dynamic Context
<301> Group Definition
<302>       Copyright 2002 IBM Corporation
<303>    < documentation >
<304>       < /annotation >
<305>       < !-Dynamic Context Group Instance-- >
<306>       < xsd: elementname=DynamicContextGroup"
<307>          type="DynamicContextGroupDefinitionTemplate",
<308>          minOccurs="1/ >
<309> < !-Dynamic Context Group Definition Template Schema _
<310> < xsd :complexType name="DynamicContextGroupDefinitionTemplate" >
<311>    < xsd: sequence >
<312> < xsd:element name="DynamicContextGroupName"
<313> type="xsd: string"/ >
<314> < xsd: elementname="DynamicContextMasterPortlet"
<315> type="PortletName"/ >
<316>    < !- only one dynamic context per dynamic context group ->
<317>    < xsd: elementname="DynamicContext" type="ContextParameter"
<318> maxOccurs="1"/ >
<319>    < xsd: elementname="DynamicContextSlavePortlet"
<320> type="SlavePortlet"
<321>                                          minOccurs="1"/ >
<322>    < /xsd : sequence >
<323> < /xsd: complexType >
<324>    < xsd:complexType name="SlavePortlet" >
<325>       < xsd :sequence >
<326>    < xsd: elementname="DynamicContextSlavePortlet"
<327> type="PortletName"/ >
```

<328> < xsd: elementname="SlavePortletAction"type="xsd : string"/ >

<329> < xsd: elementname="SlavePortletRefreshPriority"

<330> type="xsd : decimal",

<331> minOccurs="0"/ >

<332> <!- master's context is in the slave action url if slave param

<333> map is absent -->

<334> < xsd: elementname="SlaveParamMapToContext"

<335> type="ContextParameter" ,

<336> minOccurs="0"/ >

<337> < /xsd:sequence >

<338> < /xsd:complexType >

<339> < xsd: simpleTypename="PortletName" >

<340> < xsd: string >

<341> < /xsd : simpleType >

<342> < !- name of the parameter in master's requesturl-- >

<343> < xsd: simpleType name="ContextParameter" >

<344> < xsd:string >

<345> < /xsd: simpleType >

<346> < /xsd: schema >

<347> **B.2.2 런 타임**

<348> 이 섹션은,

<349> 도 8 : 정의 인스턴스에 특정된 동적 콘텍스트를 위한 포틀릿 애플리케이션 개시, 및

<350> 도 9 : 동적 콘텍스트 포틀릿 그룹 런타임 흐름을 참조하면 잘 이해될 수 있을 것이다.

<351> 동적 콘텍스트의 런타임 태양들을 취급하는 2개의 중요한 구성이 있다:

<352> 1) 동적콘텍스트액션리스너(포틀릿 액션 리스너, 904) - 이것은 마스터 포틀릿에서의 동적 콘텍스트 변화를 듣는다. 모든 동적 콘텍스트 포틀릿 그룹 내의 마스터 포틀릿은 부속된 동적콘텍스트액션리스너를 갖는다.

<353> 2) 동적콘텍스트메시지리스너(포틀릿 메시지 리스너, 906) - 메시지 리스너는 특정 동적 콘텍스트가 정의되는 그룹의 마스터로부터 통지를 듣는다. 동적 콘텍스트 포틀릿 그룹 내의 모든 슬레이브 포틀릿은 부속된 동적콘텍스트메시지리스너를 갖는다.

<354> **런타임 흐름의 단계별 설명**

<355> 포틀릿 개시 시간(도 8: 단계 801), 모든 마스터 포틀릿들은 포틀릿 디스크립터(단계 802, 805)에 기초하여 정의된 동적 콘텍스트를 마스터 포틀릿의 액션 리스너에 추가한다(단계 806). 모든 슬레이브 포틀릿들에 대한 동적 콘텍스트 타입; 액션 URL; 파라미터 매핑 및 리프레쉬 시퀀스는 포틀릿 디스크립터(단계 802, 809)로부터 검색되고 슬레이브의 포틀릿 메시지 리스너에 추가될 것이다(단계 810).

<356> 1) 동적 콘텍스트 포틀릿 그룹의 마스터 포틀릿과의 사용자 상호 작용은 동적 콘텍스트의 변화를 초래한다(단계 901).

<357> 2) 마스터의 포틀릿 동적콘텍스트액션리스너는 사용자의 액션을 검출한다(단계 902).

<358> 3) 동적콘텍스트액션리스너는 마스터 포틀릿의 요청 객체내의 동적 콘텍스트에 대응하는 이름/값 쌍을 설정한다

(단계 904).

<359> 4) 마스터 포틀릿은 동적 콘텍스트의 값을 획득하고 동일한 동적 포틀릿 그룹 내의 모든 슬레이브 포틀릿들에게 값을 통지한다(단계 905).

<360> 5) 주어진 마스터에 대한 슬레이브 포틀릿과 관련된 동적콘텍스트메시지리스너는 통지(동적 콘텍스트의 값)를 수신한다(단계 906).

<361> 6) 동적콘텍스트메시지리스너는 슬레이브 포틀릿의 포틀릿 요청 객체 내의 동적 콘텍스트의 값을 설정한다(단계 907).

<362> 7) 슬레이브 포틀릿은 동적 콘텍스트의 값을 획득한다(단계 1008).

<363> 8) 슬레이브 포틀릿은 콘텍스트와 일부 파라미터 사이의 매핑이 특정된 경우, 주어진 슬레이브 포틀릿에 대하여 정의된 액션을 수정한다(단계 1009).

<364> 9) 매핑이 특정되지 않은 경우, 동적 콘텍스트의 이름/값 쌍이 슬레이브의 포틀릿 액션에 추가된다.

<365> 10) 슬레이브 포틀릿은 동적 콘텍스트 그룹 인스턴스 정의에 정의된 액션을 수행한다(단계 1011, 1012)

<366> **B.3 룰 기반 및 롤 기반의 동적 집합**

<367> 이 섹션에서는 다음과 같은 많은 도면들이 참조된다:

<368> 도 10: 롤 기반의 동적 집합 구성의 구조도

<369> 도 11: 룰 기반의 동적 집합 구성의 구조도; 및

<370> 도 12: 룰 기반의 동적 집합 흐름도

<371> 포틀릿 서버를 위한 롤 및 룰 기반의 동적 집합 구성들은 룰 및 롤 데이터베이스들 및, 각 롤과 룰에 대한 콘텐트(content)의 개념에 기초한다.

<372> 룰에 대한 콘텐트 그룹들은 도 10의 도시된 룰 DB 구성(1001)에서 유지된다. 유사하게 롤 콘텐트 그룹들은 도 10에 도시된 롤 DB 구성(1007)에 정의된다. 각 콘텐트 그룹은 특정한 롤 또는 룰 범위 아래로 떨어지는 것으로 평가된 사용자가 액세스하는 포탈 서버 리소스들의 세트로 구성된다.

<373> 이 방식에서 또 다른 주요 구성은 플러거블 룰 엔진(1022)이다. 이 엔진의 업무는 변환된 사용자 특성들을 읽고 사용자의 특성에 기초하여 특정한 미리 정해진 사용자 그룹의 맴버십 자격이 있는 사용자들의 세트를 런타임에서 동적으로 결정하는 것이다. 또한 이 엔진은 이 동적 사용자 그룹들의 세트를 롤 및 룰 DB에 정의된 콘텐트 그룹들의 세트로 매핑한다. 바람직하게 플러거블 룰 엔진은 이 업무를 관리하는 GUI를 갖는다. 도 16에 도시된 스크린 화면은 우리가 웹스피어 개인화 서버 엔진을 사용하여 업무들을 관리하는 방법을 예시한다.

<374> 예를 들면, 도 16은 "MaleTeen"으로 불리는 동적 그룹을 정의하고 16~19세 사이의 모든 남성 사용자를 이 그룹에 할당하는 방법을 예시한다.

<375> 도 17에서 나타나는 바와 같이, 특성들에 기초하여 십대 남자들인 것으로 동적으로 평가된 모든 사용자들이 이제 그들을 위하여 실행되는 "maleteenaction" 명령을 갖는데, 이 명령들은 동적 룰 및 롤 기반의 포탈 집합 엔진(1022)이 롤 DB(1007)로부터 십대 남자에 대한 콘텐트 리소스를 선택하도록 명령한다.

<376> 개발시에, 페이지들, 포틀릿들 등의 포탈 리소스들의 세트를 롤 및 룰 DB의 특정 콘텐트 그룹에 할당하는 것이 비지니스 관리자의 업무이다. 이것은 현재 룰 및 롤 DB를 직접적으로 로딩하는 SQL 스크립트를 사용함으로써 수행된다.

<377> **B.3.1 룰 기반 및 롤 기반의 동적 집합 런타임 가능 설명**

<378> 런타임에서 포탈 사용자에 대하여 실행하는 최초의 명령은 룰 기반 엔진에 대한 래퍼(wrapper) 명령이다. 이 명령은 실제적으로 사용자 특성들의 평가를 실제적인 플러거블 룰 엔진에 의하여 시작하는 프럭시(proxy)이다.

<379> 다음 단계에서 룰 엔진은 사용자의 특성들을 저장된 프로파일로부터 읽고, 사용자 리소스 변환 모듈을 이용함으로써 특성들을 이해될 수 있는 형태로 변환한다.

<380> 도 18은 "maleteengrp"로 불리는 콘텐트 그룹에 정의된 모든 포탈 리소스들을 룰 DB에서 선택하는

"MaleTeenAction"으로 불리는 새로운 액션을 생성하는 것을 예시한다.

<381> 도 17은 동적 사용자 특성들에 기초하여 "MaleTeens"를 분류하기 위하여 이전에 생성된 룰의 범위 아래로 떨어지는 모든 사용자에 대한 "maleteengrp"의 콘텐트들을 선택하도록 집합 모듈에 명령하는 동적 집합 모듈 명령을 생성하는 것을 예시한다.

<382> 도 17은 그 분류 아래로 떨어지는 일정한 사용자의 특성들로 주어진 사용자를 위하여 집합되는 콘텐트를 판단하는데 주어진 비지니스 룰(예컨대, maleteen 그룹을 구성하는 것을 정의하는 비지니스 룰)이 어떻게 영향을 미치는지를 예시한다.

<383> 사용자의 특성들을 읽은 후, 플러거블 룰 엔진은 이 사용자의 동적 그룹 멤버십을 도 18에 도시된 다양한 동적 그룹들을 위하여 정의된 룰들에 기초하여 평가한다.

<384> 이 사용자에 대한 동적 그룹들의 설정이 한번 확정되면, 룰 엔진은 도 18에 도시된 바와 같이, 동적 그룹을 위하여 정의된 액션 상에서 콘텐트 선택을 실행함으로써 이 사용자를 위한 관련된 포탈 콘텐트를 선택한다. 실행시에 액션들은 액션들에 대하여 정의된 콘텐트 그룹들로부터 포탈 리소스들의 세트를 룰 DB에 리턴한다.

<385> 다음의 실행 단계는 룰 엔진에 의하여 사용자에 할당되는 롤들의 평가이다. 롤 엔진은 (사용자 프로파일 특성들로부터 추출된) 조직의 멤버십을 사용하여 롤 DB로부터 사용자의 롤에 대한 콘텐트 리소스들의 세트를 추출한다. 이 리소스들은 이전의 세트에서 생성된 룰 기반 포탈 리소스들의 기존 리스트에 추가된다.

<386> 그 후, 리스트는 실행을 위하여 동적 포탈 집합 엔진에 포워딩된다. 동적 포탈 집합 엔진은 리스트에 의하여 식별되는 포탈 리소스들을 선택하여 현재의 사용자를 위한 디폴트 포탈 뷰를 설정완료(set up)한다.

<387> **요약**

<388> 1. 공통 백엔드 웹 애플리케이션의 통합 구현

<389> 포틀릿 애플리케이션 HTTP 클라이언트 및 포틀릿 애플리케이션 세션으로, 공통 백엔드 웹 애플리케이션 통합 모델을 갖는 것이 이제 가능하다. 이것은 동일한 포틀릿 애플리케이션 내의 다중 포틀릿들이 동일한 웹 애플리케이션 백엔드와 통신하는 것을 가능케 하는데 사용될 수 있다.

<390> 본 발명의 이런 구현은 다음을 가능하게 한다:

<391> ⅰ. 분리된 브라우저들을 런칭(launching)하지 않고, 또한 동일한 백엔드 웹 애플리케이션을 액세스하기 위하여 사용자 ID 및 암호에 대한 필요한 여러 번의 촉구없이도, 고유 포틀릿 통합을 가지고;

<392> ⅱ. 다중 요청들을 만들고 세션 관리로 백엔드 애플리케이션에/로부터의 응답들을 수신하는 것.

<393> 2. 간단한 도구화를 유도하는 간단한 공통 시스템

<394> 본 발명은 백엔드 서버에서 동작하는 기존의 웹 애플리케이션; 및 포틀릿 애플리케이션의 배치 디스크립터에서 단지 관련된 백엔드 웹 애플리케이션의 URL의 특정을 필요로 함으로써, 포틀릿 애플리케이션들을 통합하는 쉽고 빠른 방법을 제공한다. 이것으로, 통합의 공통 업무를 처리하는 도구를 만드는 것이 가능하다.

<395> 3. 포틀릿 애플리케이션 공유 공통 세션 내의 포틀릿들 및 세션 데이터

<396> 포틀릿 애플리케이션 세션 객체의 구현은 동일한 포틀릿 애플리케이션의 포틀릿들이 포틀릿 애플리케이션 사이에서 포틀릿 애플리케이션 내의 유일한, 동시에 포탈 서버의 원시 HTTP 세션의 정보와는 다른 공통 데이터를 공유하는 것을 가능케 한다. 이것은 동일한 포틀릿 애플리케이션 내의 포틀릿 사이에서 유일한 데이터의 공유를 쉽게 한다.

<397> 4. 포탈 세션 및 백엔드 세션의 공통 세션 데이터 공유

<398> 세션 릴레이의 구현은 포탈 서버와 백엔드 웹 애플리케이션 사이의 공통 세션 데이터의 공유를 가능하게 만든다. 이것은 백엔드 웹 애플리케이션이 포탈 서버로부터 정보를 수신하는 것을 가능하게 하고, 웹 애플리케이션의 비지니스 로직이 포탈 서버로부터 패싱된 정보를 이용할 수 있게 한다.

<399> 예를 들면: 백엔드 웹 애플리케이션이 단지 정보의 요약 버전을 보내는 포틀릿의 정상적인 뷰와는 다르게, 현재의 포틀릿 상태가 포틀릿의 최대화된 뷰를 디스플레이하는 경우, 백엔드 웹 애플리케이션은 정보를 수신하고 상세화된 비지니스 정보를 다시 보냄으로써 정보를 이용할 수 있다.

<400> 5. 포틀릿 애플리케이션 세션, 포틀릿 애플리케이션 세션 객체, 포틀릿 HTTP 클라이언트, 및 세션 릴레이 메커니즘을 가진 포탈 서버와 다른 응집한 백엔드 웹 애플리케이션 세션

<401> 백엔드 웹 애플리케이션은 여전히 포탈 서버의 세션을 가진 동일한 쿠키를 공유하면서, 이제 포탈 서버와 다른 세션을 저장할 수 있다. 백엔드 웹 애플리케이션은 이제 독립적이고 정확하게 동작할 수 있고, 포탈의 다양한 포틀릿들로부터 포틀릿 요청들을 인식하며, 백엔드 웹 애플리케이션과 응집한 세션을 인에이블링(enabling)할 수 있다.

<402> 6. 교차하는 포탈 서버 및 백엔드 웹 애플리케이션 상의 한번의 사인온

<403> 세션 릴레이 실행은 포탈 서버에 한번 로그인했던 사용자가, 관련된 백엔드 웹 애플리케이션에 로그인하기 위하여 사용자 자격을 다시 제출할 필요가 없도록 하는 한번의 사인온(sign-on) 능력을 제공한다. 이것은 포탈로의 HTTP 세션 및 포틀릿 HTTP 클라이언트부터 백엔드 웹 애플리케이션까지의 HTTP 세션 사이의 일대일 매핑을 가진 쿠키 테이블 수단에 의하여 가능하다.

<404> 7. 포탈 서버와 동기화된 백엔드 웹 애플리케이션 비해이비어(behavior)

<405> 세션 릴레이 실행은 포탈 세션으로부터의 세션 정보를 백엔드 웹 애플리케이션의 세션에 릴레이함으로써 백엔드 웹 애플리케이션의 비해이비어를 동기화함에 의하여 인에이블링된 무결절성(seamless) 통합을 가능하게 한다.

<406> 다음은 몇 가지의 예들이다:

<407> 포탈 서버에서 언어 및 위치 설정은 이제 백엔드 웹 애플리케이션에 패싱되어 이제 포탈 서버의 위치+언어 설정에 기초한 응답 메시지를 백엔드 웹 애플리케이션이 구성할 수 있다.

<408> 또 다른 예로, 포탈 서버로부터의 세션 만료 정보는 이제 백엔드 웹 애플리케이션 세션에 패싱되어 포탈 세션이 타임 아웃됨과 동시에 백엔드 웹 애플리케이션 세션이 타임 아웃될 수 있다. 백엔드 웹 애플리케이션은 이제 포탈 서버로부터 릴레이된 포탈의 상태 및 이벤트들에 응답할 수 있다.

<409> 8. 동일한 포탈 페이지 내에 동기화된 콘텐트

<410> 동적 콘텍스트 포틀릿 그룹핑은 동일한 동적 콘텍스트 그룹내의 포틀릿들 사이의 협력을 가능케 하여 비지니스 프로세스 및 정보 통합, 동기화를 달성할 수 있다.

<411> 각 포틀릿은 다중의 동적 콘텍스트 그룹들에 참여하는 것이 허용된다. 이것은 매우 개방되고 간단한 프로그래밍 모델을 포탈 관리자에 제공하여, 포탈 관리자는 포틀릿들을 동적 콘텍스트 포틀릿 그룹들로 그룹핑한다.

<412> 동적 콘텍스트 정의의 간단한 구조는 각 그룹핑을 위한 슬레이브 포틀릿들 및 동적 콘텍스트 마스터의 자동 생성을 위하여 만들어진 간단한 도구화(tooling)를 가능하게 한다.

<413> 동적 콘텍스트 정의 구현, 동적 콘텍스트 그룹, 마스터 포틀릿 및 (슬레이브 업무들, 슬래이브 콘텍스트 맵을 포함한) 슬레이브 포틀릿 구현은 본 발명의 장점들을 제공하는데 도움이 된다.

<414> 9. 포틀릿들의 리프레쉬 시퀀스를 정의하는 능력

<415> 트랜잭션 관리자는 최초로 포틀릿들의 리프레쉬 시퀀스를 정의하는 능력을 제공한다. 포틀릿들의 리프레쉬 시퀀스를 정의하는 능력은 포탈/포틀릿 아키텍처를 사용하여 순차적 비지니스 로직의 적합한 구현을 가능하게 한다. 트랜잭션 관리자; 리소스 소터; 응답들의 캐싱(caching)은 본 발명의 장점들을 제공하는데 도움이 된다.

<416> 10. 룰 기반 및 롤 기반의 집합

<417> 양호한 레벨의 포탈 개인화는 현재 동적 집합에 의하여 달성될 수 있다. 본 발명에 따라 적용되는 포틀릿들, 페이지들 또는 페이지 그룹들의 형식적(formal) 개념이 없다는 점에서 정상적인 웹 애플리케이션들의 종래 기술 구현과는 명백하게 다르다. 양호한 레벨의 개인화는 포탈 시장이 활성화 되고, 양호한 레벨의 캠페인 목표에 대한 사용자의 필요 등이 도래하면서 점점 더 중요해질 것이다.

<418> 본 발명의 실시예들은 다음과 같은 많은 장점들을 제공한다.

<419> 1. 우리의 접근에 의하여 달성될 수 있는 개인화의 레벨은 훨씬 더 양호하게 정제(grained)된다. 포틀릿 운영 장치들은 오늘날 포탈 서버에 의하여 제공된다. 오늘날 사용가능한 포틀릿 운영 장치들은 고유(nature) 메뉴얼 구성에 의한다. 한번 구성되면, 정적이고 런타임에 변하지 않는다. 본 발명은 룰에 기초한 포탈 리소스들을

렌더링(rendering)하는 동적 능력을 제공한다.

<420> 2. 포탈 집합 모듈은 동적 엔터티이므로, 그것에 직접 룰 및 롤 엔진들을 묶는 것은 인간의 어떠한 간섭없이도 실시간의 동적 집합 능력들을 우리가 달성하게 한다.

<421> 3. 페이지들 및 페이지 그룹들과 같은 조잡한 포탈 리소스들의 개인화는 우리가 동적 레이아웃을 수행하게 한다.

<422> 4. 훨씬 더 효율적인 캠페인, 계약 등이 설정될 수 있다. 이것은 전자상거래 소매 및 B2B 조직들에 매우 중요하다.

<423> 5. 본 발명은 정상적인 콘텐트 개인화보다 매우 높은 정도의 개인화를 가능하게 한다. 예를 들면, 우리는 실제적으로 룰들에 기초한 웹페이지 전체의 섹션들을 불가능하게 할 수 있다. 이것은 보통의 개인화에 의하여는 수행될 수 없다. 또한, 동적 집합은 리소스가 관련되지 않은, 콘텐트인 정상적인 개인화의 영역에 적용되지 않는다.

도면의 간단한 설명

<103> 본 발명의 실시예들은 다음의 도면들을 참조하여, 예시하는 방법으로 설명될 것이다.

<104> 도 1은 동적 콘텍스트 연쇄 모델을 나타낸다.

<105> 도 2는 포탈로의 웹 애플리케이션 통합을 나타낸다.

<106> 도 3은 통합 구조도를 나타낸다.

<107> 도 4는 통합 흐름도를 나타낸다.

<108> 도 5는 웹 애플리케이션으로의 포탈 통합을 위한 구조도를 나타낸다.

<109> 도 6은 통합을 위한 흐름도를 나타낸다.

<110> 도 7은 포틀릿들에 대한 동적 콘텍스트 그룹들의 예를 나타낸다.

<111> 도 8은 정의 인스턴스에 정의된 동적 콘텍스트에 대한 포틀릿 애플리케이션 개시를 나타낸다.

<112> 도 9는 동적 콘텍스트 포틀릿 그룹의 런타임 흐름을 나타낸다.

<113> 도 10은 롤 기반의 동적 집합 구성의 구조 맵을 나타낸다.

<114> 도 11은 룰 기반의 동적 집합 구성의 구조 맵을 나타낸다.

<115> 도 12는 롤 기반의 동적 집합의 흐름도를 나타낸다.

<116> 도 13은 웹 애플리케이션들에 대한 포틀릿 요청들의 취급을 나타낸다.

<117> 도 14는 동기화 모델 도면을 나타낸다.

<118> 도 15는 시퀀스 인식 포탈 집합 엔진에 대한 흐름도를 나타낸다.

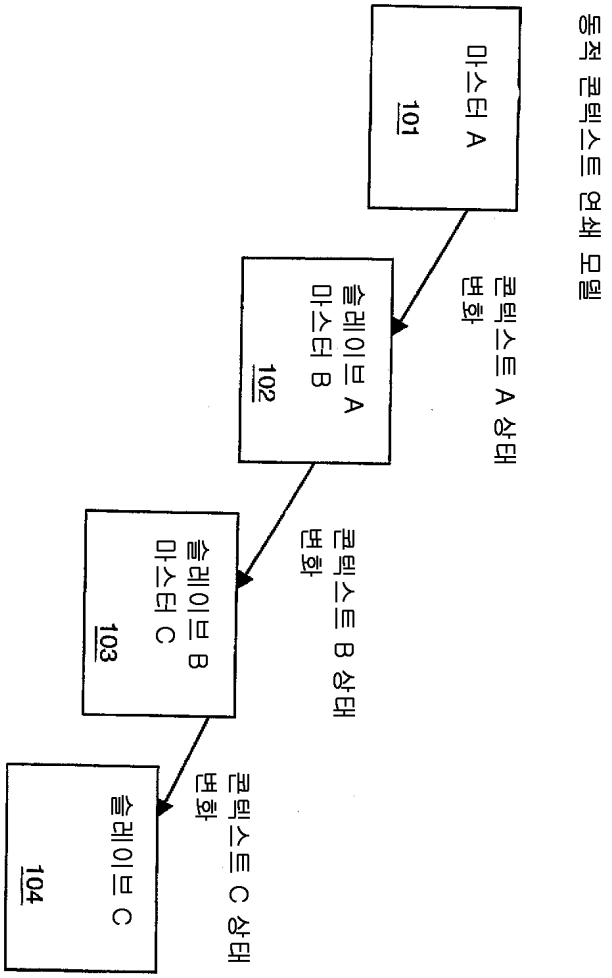
<119> 도 16은 "MaleTeen"으로 불리는 동적 그룹을 정의하고 사용자들을 그룹에 할당하는 것을 나타낸다.

<120> 도 17은 룰 데이터베이스 콘텐트 그룹 선택 액션을 동적 사용자 그룹에 할당하는 것을 나타낸다.

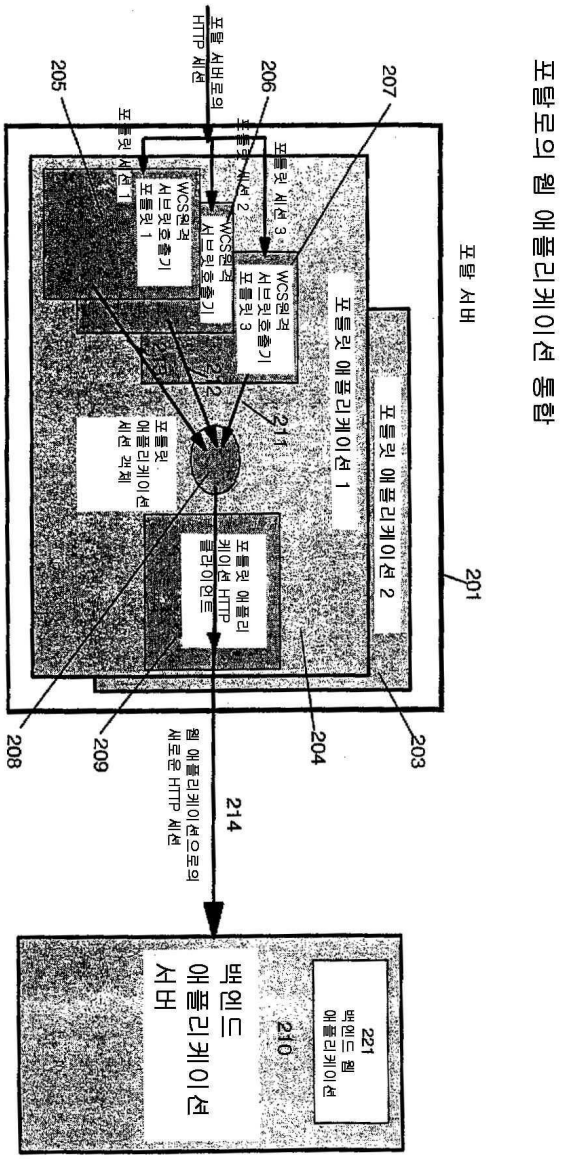
<121> 도 18은 "MaleTeenAction"으로 불리는 새로운 액션을 생성하는 것을 나타낸다.

도면

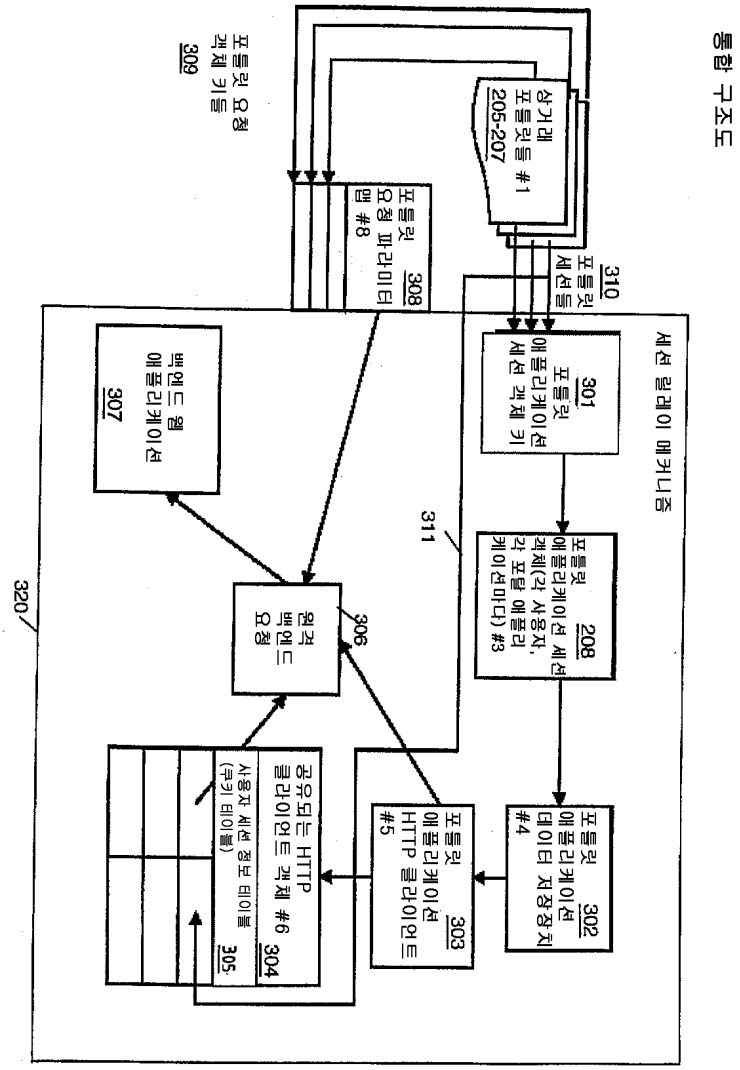
도면1


동적 콘텍스트 연쇄 모델
마스터 A
101
콘텍스트 A 상태 변화
슬레이브 A
마스터 B
102
콘텍스트 B 상태 변화
슬레이브 B
마스터 C
103
콘텍스트 C 상태 변화
슬레이브 C
104

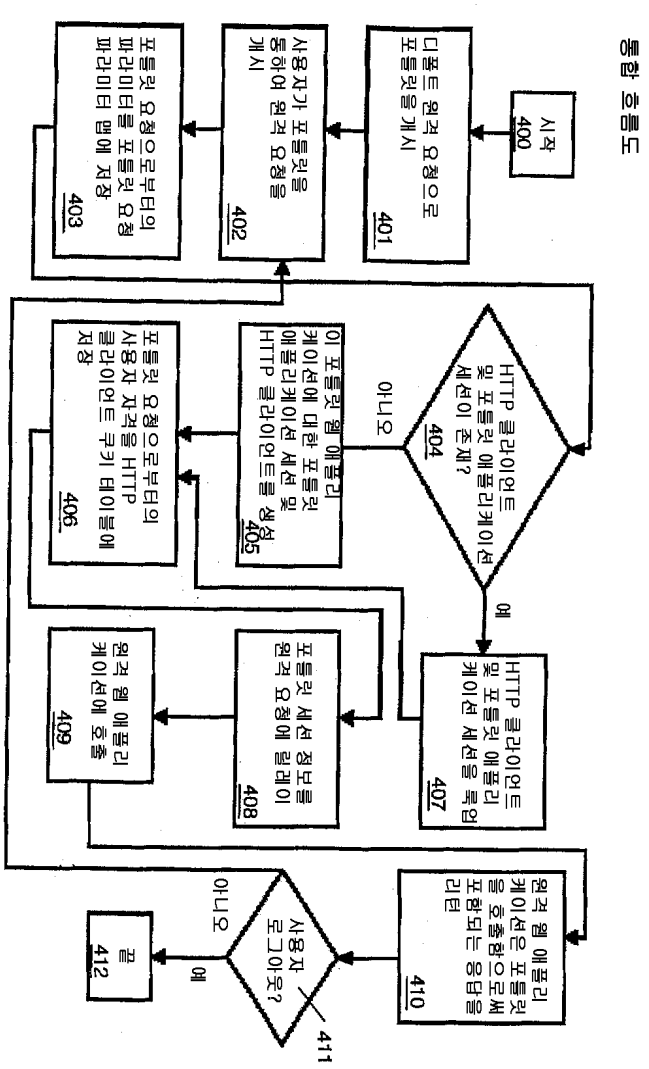
도면2


포탈로의 웹 애플리케이션 통합
포탈 서버
포탈 서버로의 HTTP 세션
205
206
207
포틀릿 세션 1
포틀릿 세션 2
포틀릿 세션 3
WCS원격 서브릿 호출기 포틀릿 1
WCS원격 서브릿 호출기 포틀릿 2
WCS원격 서브릿 호출기 포틀릿 3
포틀릿 애플리케이션 1
포틀릿 애플리케이션 2
211
포틀릿 애플리케이션 세션 객체
포틀릿 애플리케이션 HTTP 클라이언트
201
208
209
204
203
웹 애플리케이션으로의 새로운 HTTP 세션
214
백엔드 애플리케이션 서버
210
221
백엔드 웹 애플리케이션

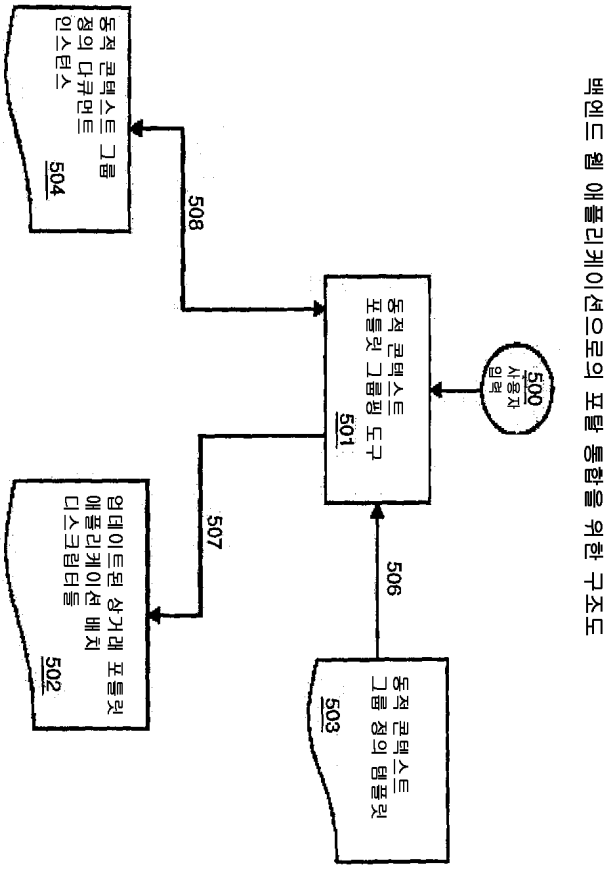
도면3


통합 구조도
세션 릴레이 메커니즘
320
311
310
포틀릿 세션들
삼각래 포틀릿들 #1 205-207
포틀릿 요청 파라미터 맵 #8 308
포틀릿 요청 객체 키들 309
301 포틀릿 애플리케이션 세션 객체 키
포틀릿 애플리케이션 세션 객체(각 사용자, 각 포틀 애플리케이션마다) #3 208
백엔드 웹 애플리케이션 307
원격 백엔드 요청 306
포틀릿 애플리케이션 데이터 저장 장치 #4 302
포틀릿 애플리케이션 HTTP 클라이언트 #5 303
공유되는 HTTP 클라이언트 객체 #6 304
사용자 세션 정보 테이블 (쿠키 테이블) 305

도면4

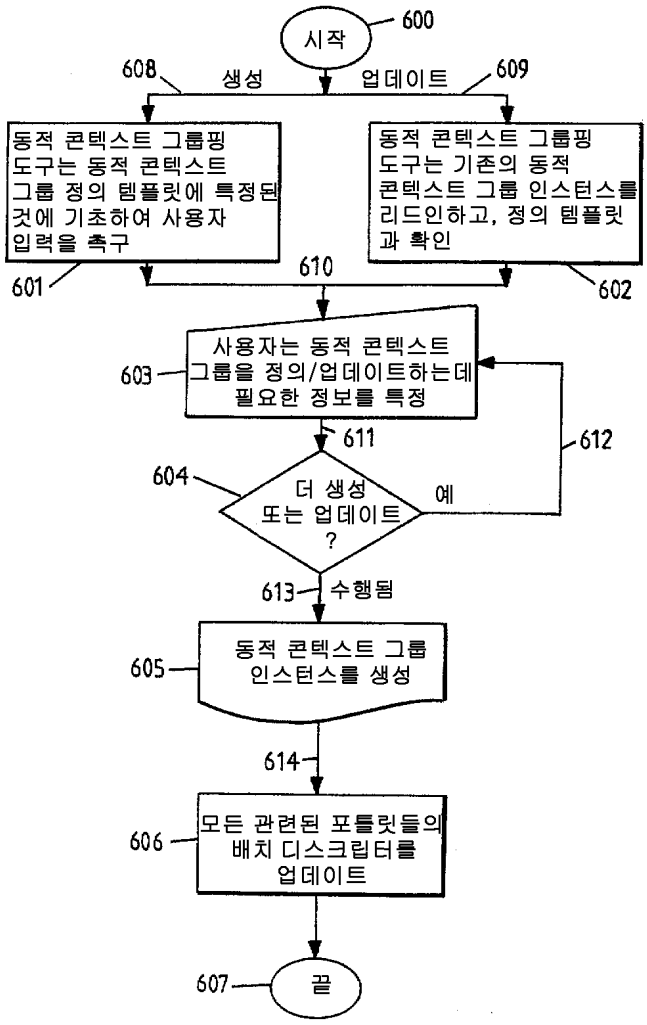

통합 흐름도
시작
400
디폴트 원격 요청으로 포틀릿을 개시
401
사용자가 포틀릿을 통하여 원격 요청을 개시
402
포틀릿 요청으로부터의 파라미터를 포틀릿 요청 파라미터 맵에 저장
403
HTTP 클라이언트 및 포틀릿 애플리케이션 세션이 존재?
404
아니오
이 포틀릿 웹 애플리케이션에 대한 포틀릿 애플리케이션 세션 및 HTTP 클라이언트를 생성
405
포틀릿 요청으로부터의 사용자 자격을 HTTP 클라이언트 쿠키 테이블에 저장
406
예
HTTP 클라이언트 및 포틀릿 애플리케이션 세션을 룩업
407
포틀릿 세션 정보를 원격 요청에 릴레이
408
원격 웹 애플리케이션에 호출
409
원격 웹 애플리케이션은 포틀릿을 호출함으로써 포함되는 응답을 리턴
410
사용자 로그아웃?
411
아니오
예
끝
412

도면5


백엔드 웹 애플리케이션으로의 포탈 통합을 위한 구조도
동적 콘텍스트 그룹 정의 다큐먼트 인스턴스
504
508
동적 콘텍스트 포틀릿 그룹핑 도구
501
500
사용자 입력
507
업데이트된 상거래 포틀릿 애플리케이션 배치 디스크립터들
502
506
동적 콘텍스트 그룹 정의 템플릿
503

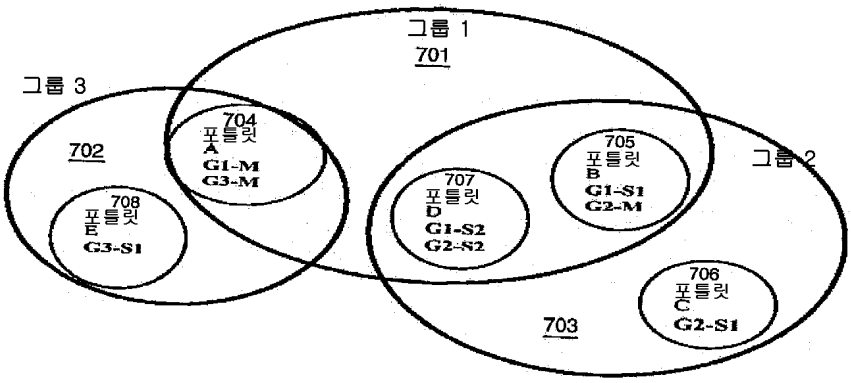
도면6

통합을 위한 흐름도



도면7

포틀릿들에 대한 동적 콘텍스트 그룹들의 예



동적 그룹 1:

마스터 포틀릿	G1-M:	포틀릿 A
슬레이브 포틀릿	G1-S1:	포틀릿 B (제1 우선순위)
슬레이브 포틀릿	G1-S2	포틀릿 D (제2 우선순위)

동적 그룹 2:

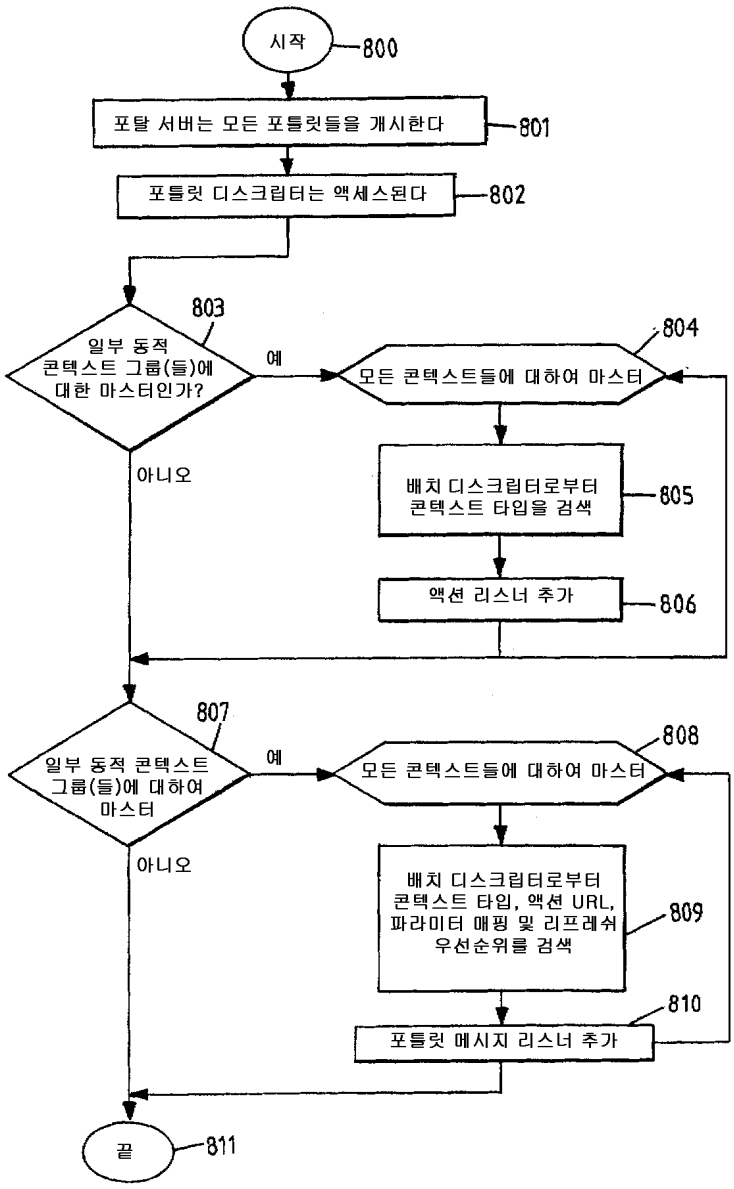
마스터 포틀릿	G2-M	포틀릿 B
슬레이브 포틀릿	G2-S1	포틀릿 C (제1 우선순위)
슬레이브 포틀릿	G2-S2	포틀릿 D (제2 우선순위)

동적 그룹 3:

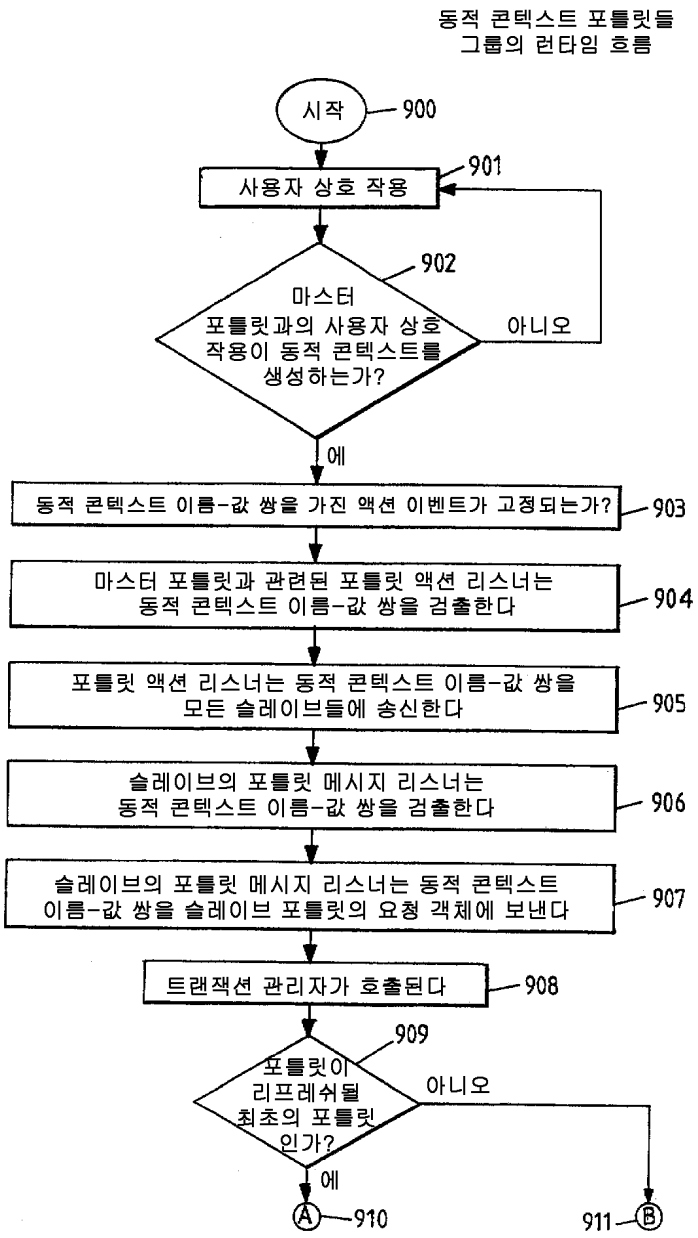
마스터 포틀릿	G3-M	포틀릿 A
슬레이브 포틀릿	G3-S1	포틀릿 E

도면8

정의 인스턴스에 특정된 동적 콘텍스트에 대한
포틀릿 애플리케이션 개시

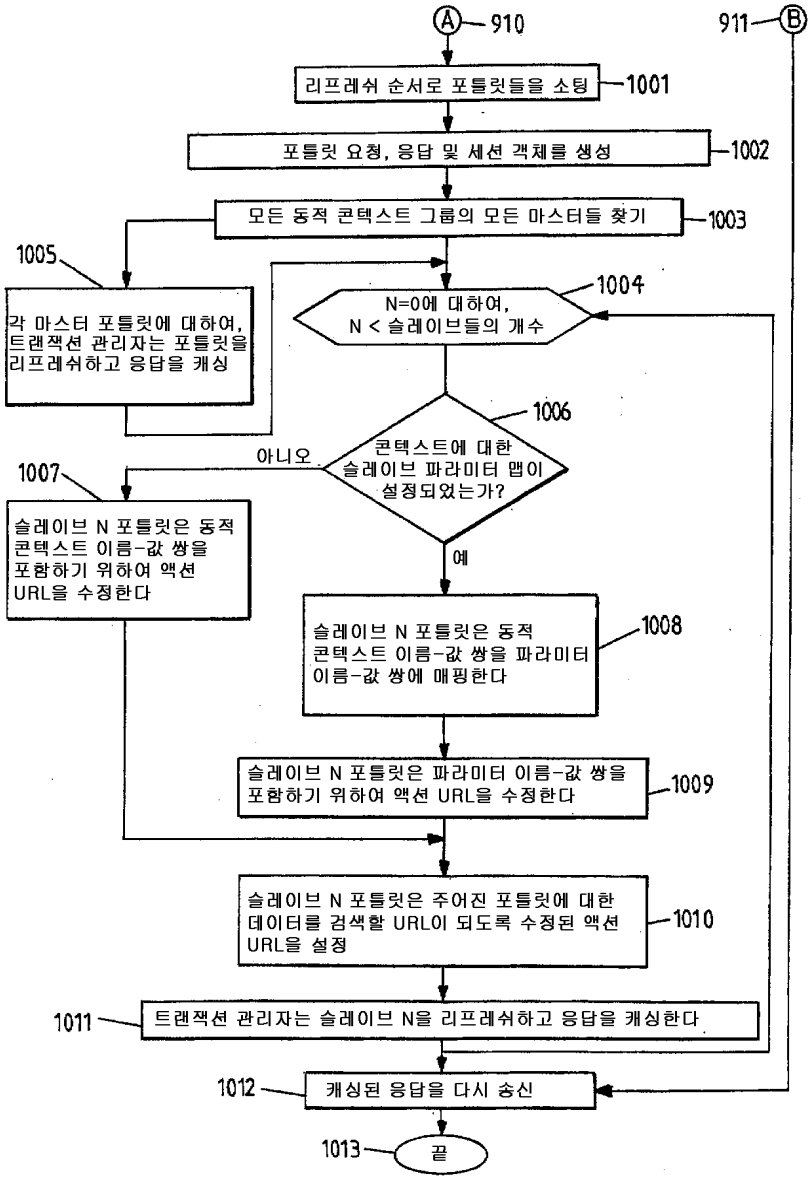


도면9a

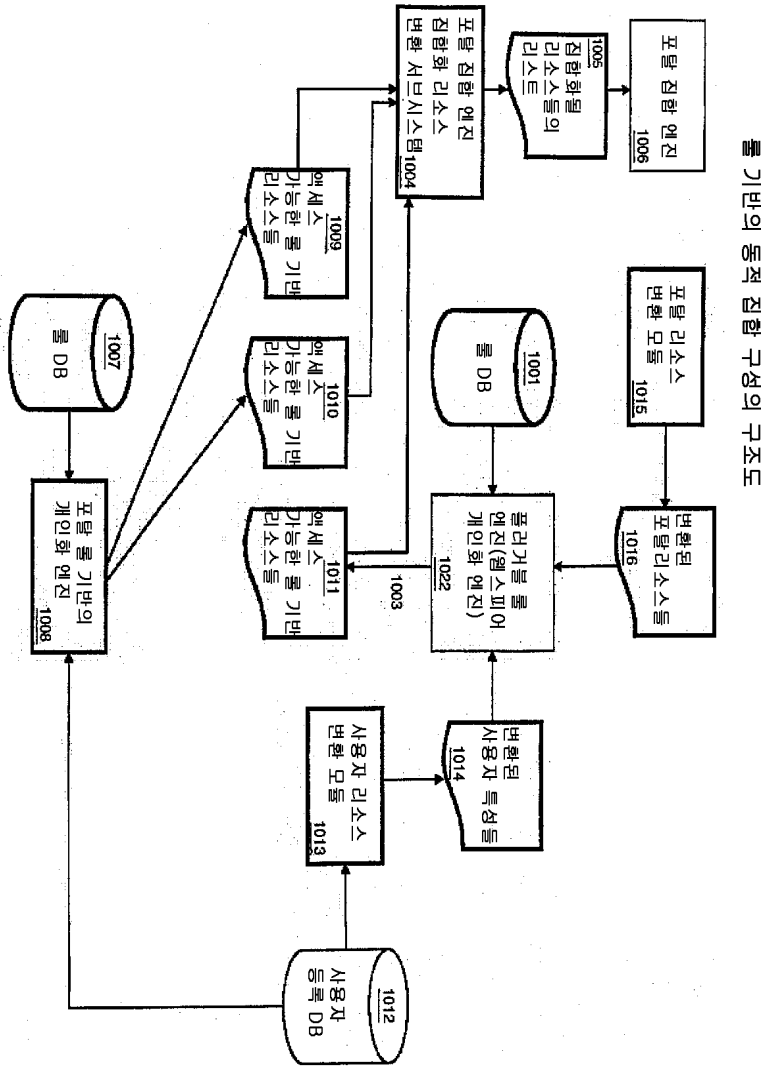

동적 콘텍스트 포틀릿들
그룹의 런타임 흐름
시작
900
사용자 상호 작용
901
902
마스터
포틀릿과의 사용자 상호
작용이 동적 콘텍스트를
생성하는가?
아니오
예
동적 콘텍스트 이름-값 쌍을 가진 액션 이벤트가 고정되는가?
903
마스터 포틀릿과 관련된 포틀릿 액션 리스너는
동적 콘텍스트 이름-값 쌍을 검출한다
904
포틀릿 액션 리스너는 동적 콘텍스트 이름-값 쌍을
모든 슬레이브들에 송신한다
905
슬레이브의 포틀릿 메시지 리스너는
동적 콘텍스트 이름-값 쌍을 검출한다
906
슬레이브의 포틀릿 메시지 리스너는 동적 콘텍스트
이름-값 쌍을 슬레이브 포틀릿의 요청 객체에 보낸다
907
트랜잭션 관리자가 호출된다
908
909
포틀릿이
리프레쉬될
최초의 포틀릿
인가?
아니오
예
A
910
911
B

도면9b

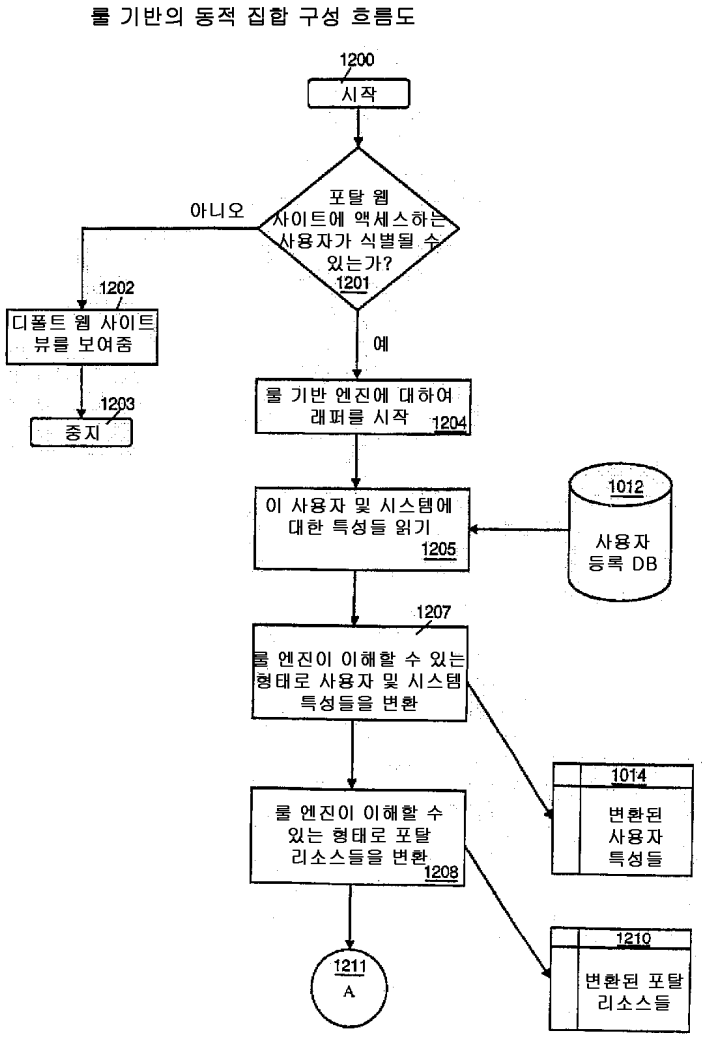
동적 콘텍스트 포틀릿들 그룹의 런타임 흐름



도면10

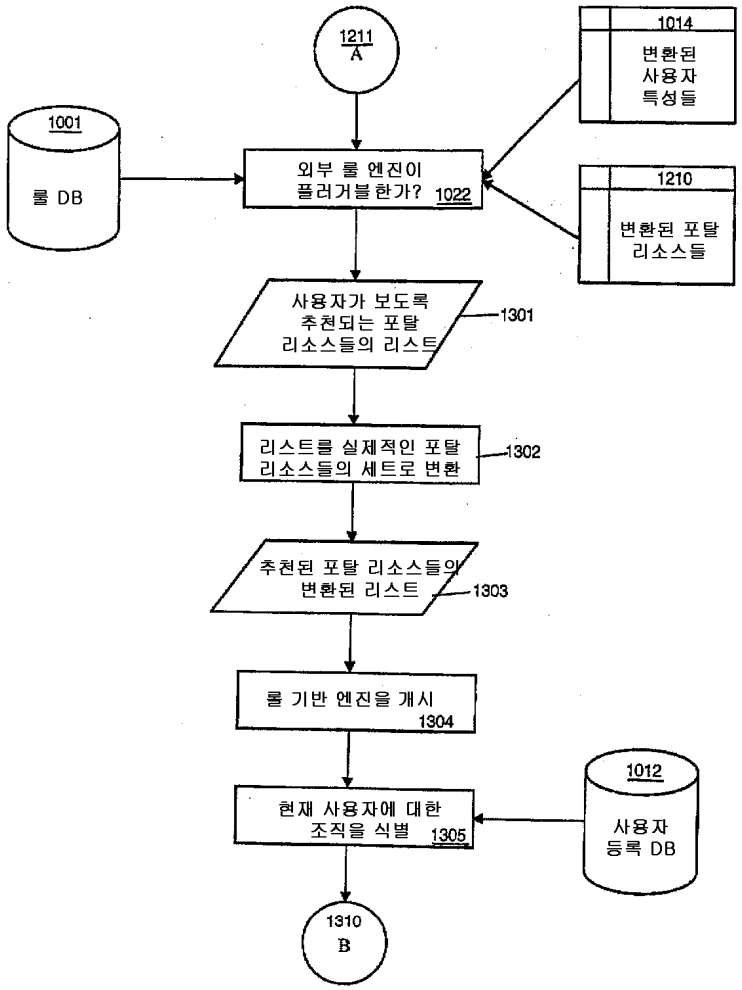

룰 기반의 동적 집합 구성의 구조도
포탈 집합 엔진 1006
집합 회원 리스트들의 리스트 1005
포탈 집합 엔진 변환 서브시스템 1004
액세스 가능한 룰 기반 리소스들 1009
액세스 가능한 룰 기반 리소스들 1010
액세스 가능한 룰 기반 리소스들 1011
룰 DB 1007
포탈 룰 기반의 개인화 엔진 1008
룰 DB 1001
포탈 리소스 변환 모듈 1015
변환된 포탈리소스들 1016
프러거블 룰 엔진(웹스피어 개인화 엔진) 1022
1003
변환된 사용자 특성들 1014
사용자 리소스 변환 모듈 1013
사용자 등록 DB 1012

도면11

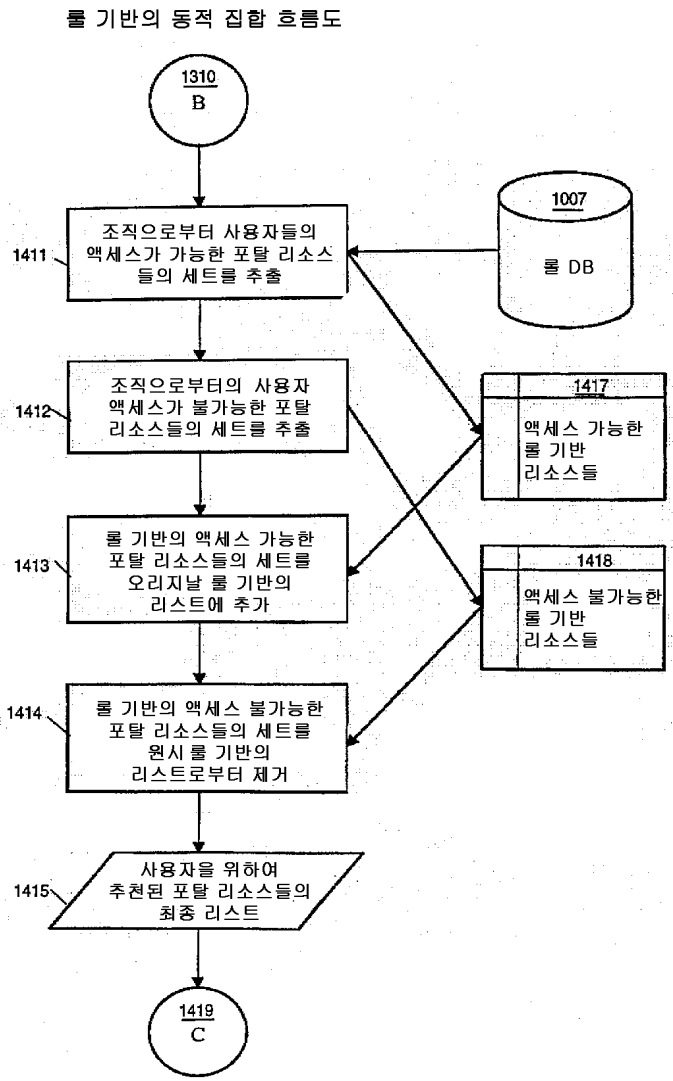

룰 기반의 동적 집합 구성 흐름도
1200
시작
포탈 웹 사이트에 액세스하는 사용자가 식별될 수 있는가?
1201
아니오
예
1202
디폴트 웹 사이트 뷰를 보여줌
1203
중지
룰 기반 엔진에 대하여 래퍼를 시작
1204
이 사용자 및 시스템에 대한 특성들 읽기
1205
1012
사용자 등록 DB
1207
룰 엔진이 이해할 수 있는 형태로 사용자 및 시스템 특성들을 변환
1014
변환된 사용자 특성들
룰 엔진이 이해할 수 있는 형태로 포탈 리소스들을 변환
1208
1210
변환된 포탈 리소스들
1211
A

도면12a

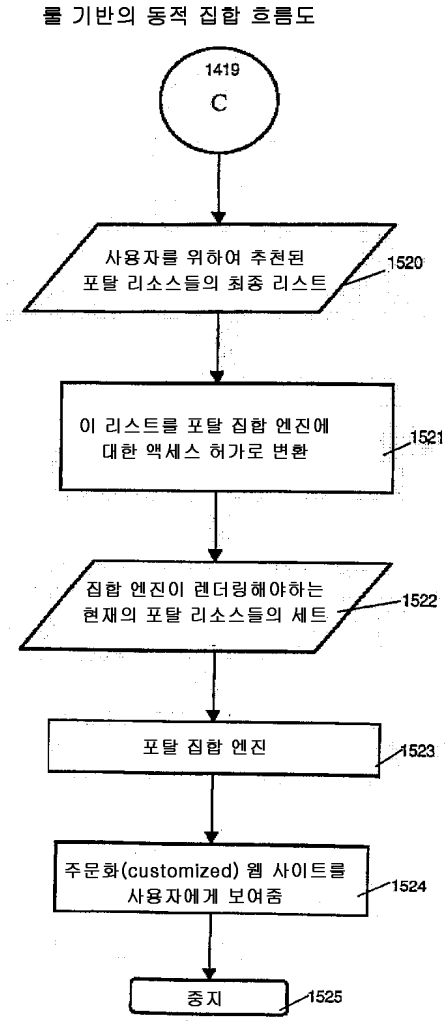
룰 기반의 동적 집합 흐름도



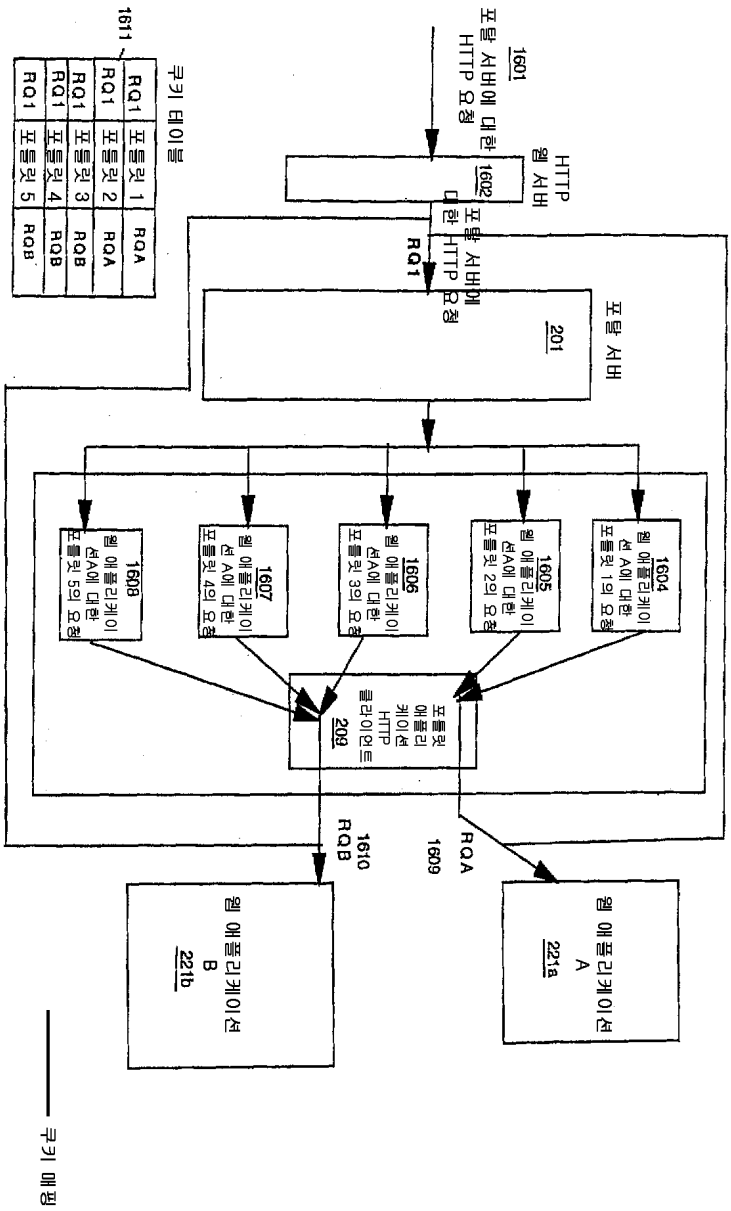
도면12b


롤 기반의 동적 집합 흐름도
1310
B
1007
롤 DB
1411
조직으로부터 사용자들의 액세스가 가능한 포탈 리소스들의 세트를 추출
1412
조직으로부터의 사용자 액세스가 불가능한 포탈 리소스들의 세트를 추출
1417
액세스 가능한 롤 기반 리소스들
1413
롤 기반의 액세스 가능한 포탈 리소스들의 세트를 오리지날 롤 기반의 리스트에 추가
1418
액세스 불가능한 롤 기반 리소스들
1414
롤 기반의 액세스 불가능한 포탈 리소스들의 세트를 원시 롤 기반의 리스트로부터 제거
1415
사용자을 위하여 추천된 포탈 리소스들의 최종 리스트
1419
C

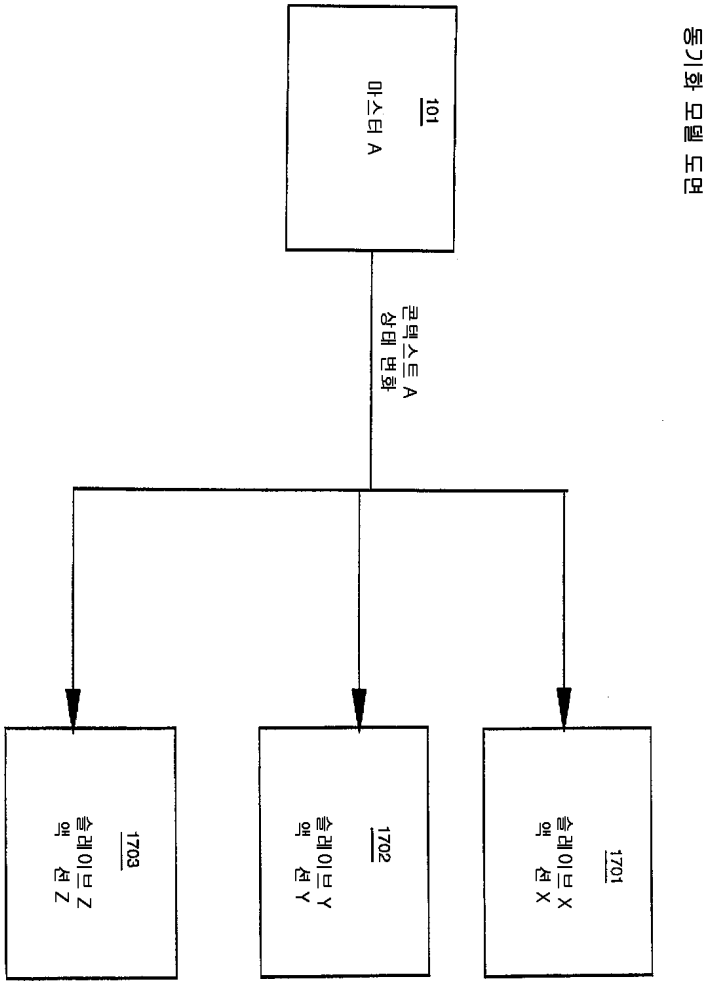
도면12c


롤 기반의 동적 집합 흐름도
1419
C
사용자를 위하여 추천된
포탈 리소스들의 최종 리스트
1520
이 리스트를 포탈 집합 엔진에
대한 액세스 허가로 변환
1521
집합 엔진이 렌더링해야하는
현재의 포탈 리소스들의 세트
1522
포탈 집합 엔진
1523
주문화(customized) 웹 사이트를
사용자에게 보여줌
1524
중지
1525

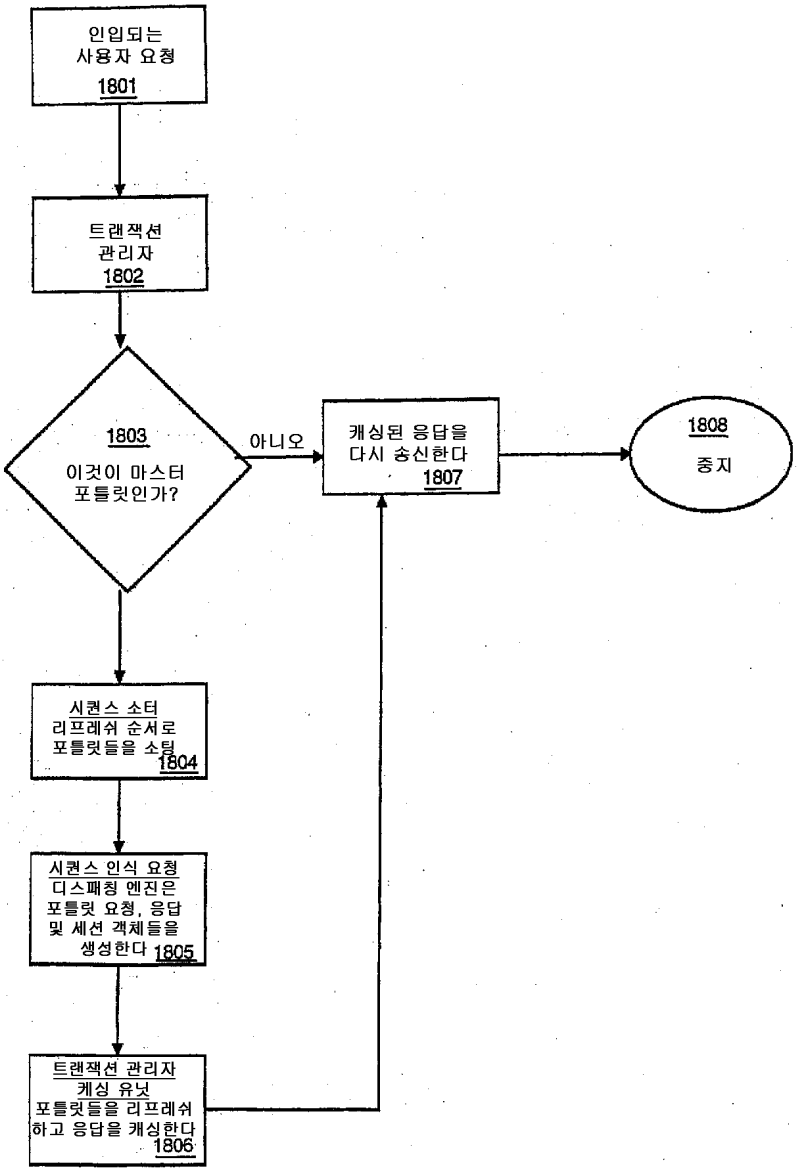
도면13


1611
쿠키 테이블
RQ1 | 포틀릿 1 | RQA
RQ1 | 포틀릿 2 | RQA
RQ1 | 포틀릿 3 | RQB
RQ1 | 포틀릿 4 | RQB
RQ1 | 포틀릿 5 | RQB
포탈 서버에 대한 HTTP 요청
1601
HTTP 웹 서버
1602
포탈 서버에 대한 HTTP 요청
RQ1
201
포탈 서버
1604 웹 애플리케이션A에 대한 포틀릿 1의 요청
1605 웹 애플리케이션A에 대한 포틀릿 2의 요청
1606 웹 애플리케이션A에 대한 포틀릿 3의 요청
1607 웹 애플리케이션A에 대한 포틀릿 4의 요청
1608 웹 애플리케이션A에 대한 포틀릿 5의 요청
포틀릿 애플리케이션 HTTP 클라이언트
209
RQA
1609
RQB
1610
웹 애플리케이션 A
221a
웹 애플리케이션 B
221b
쿠키 매핑

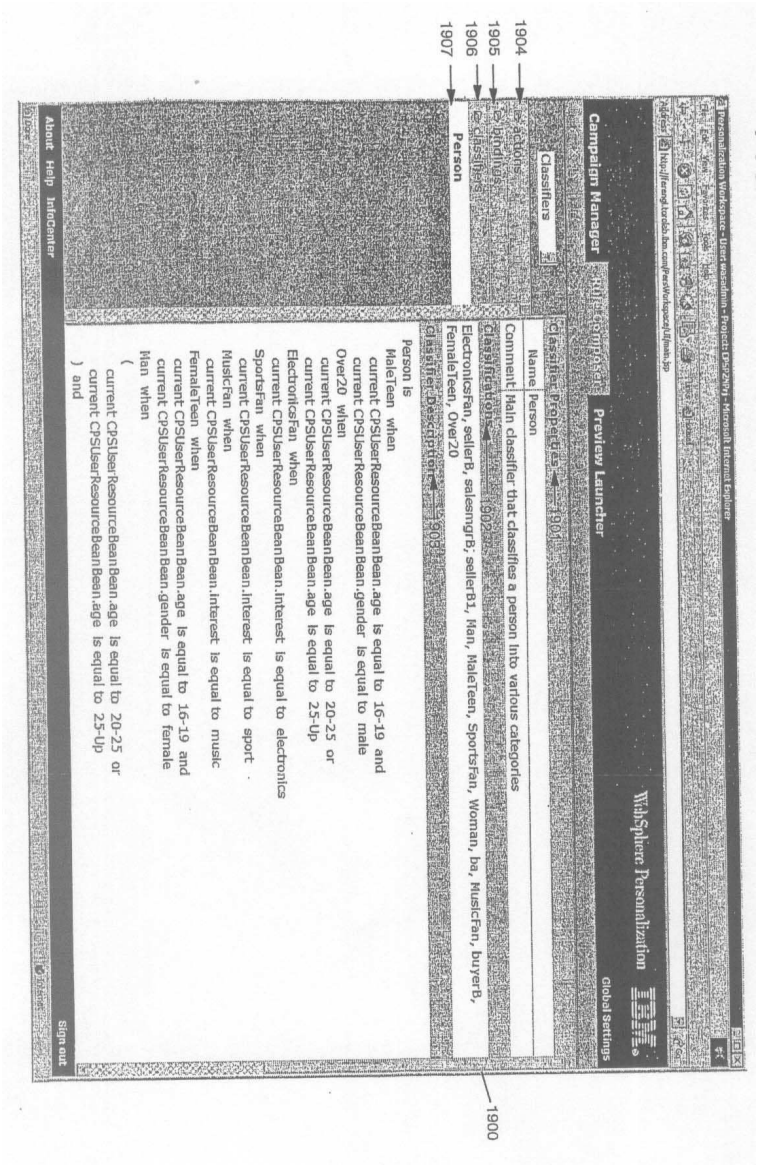
도면14


동기화 모델 도면
101
마스터 A
콘텍스트 A
상태 변화
1701
슬레이브 X
액 션 X
1702
슬레이브 Y
액 션 Y
1703
슬레이브 Z
액 션 Z

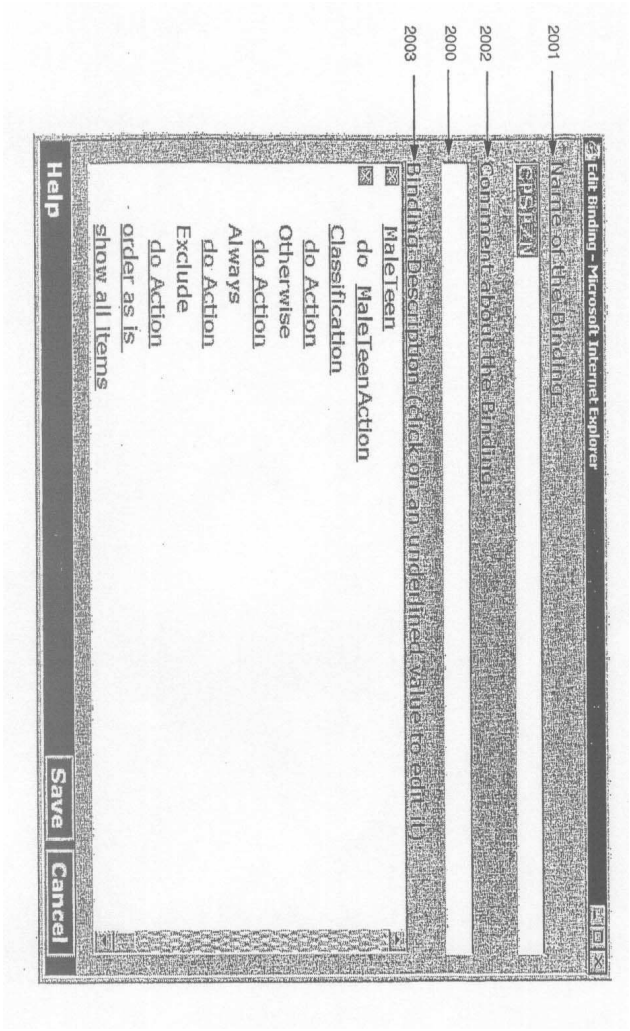
도면15


인입되는
사용자 요청
1801
트랜잭션
관리자
1802
1803
이것이 마스터
포틀릿인가?
아니오
캐싱된 응답을
다시 송신한다
1807
1808
중지
시퀀스 소터
리프레쉬 순서로
포틀릿들을 소팅
1804
시퀀스 인식 요청
디스패칭 엔진은
포틀릿 요청, 응답
및 세션 객체들을
생성한다 1805
트랜잭션 관리자
캐싱 유닛
포틀릿들을 리프레쉬
하고 응답을 캐싱한다
1806

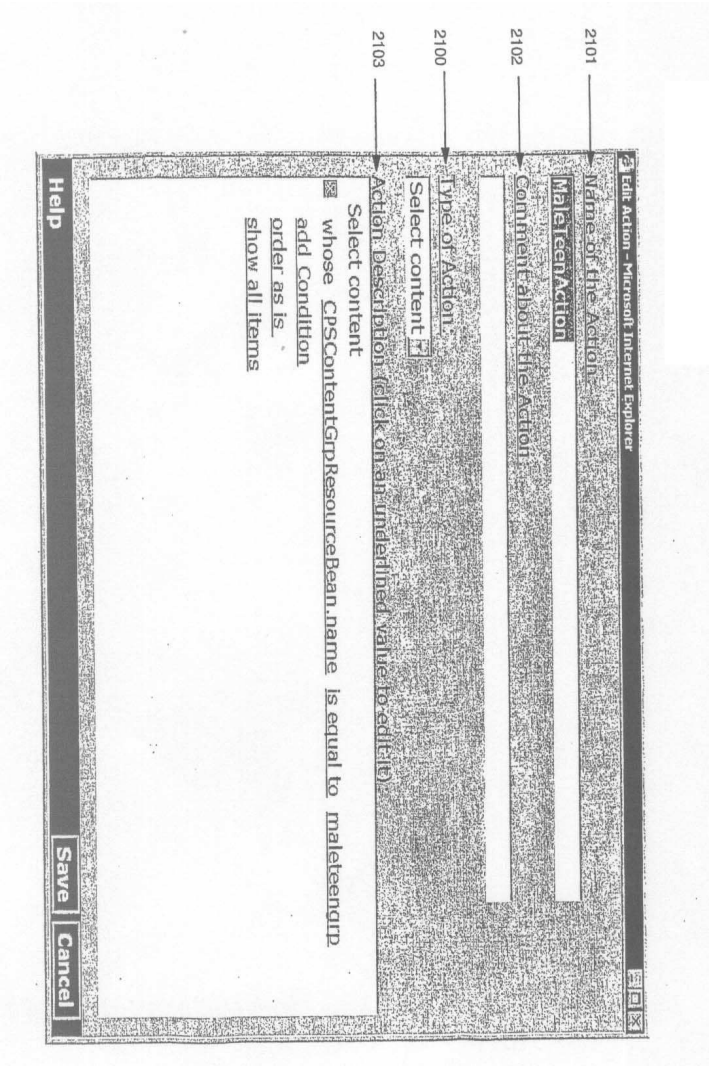
도면16


1904
1905
1906
1907
1900
WebSphere Personalization
Campaign Manager
Rule Composer
Preview Launcher
Global Settings
Classifiers
Actions
Attributes
Classifiers
Person
Classifier Properties
1904
Name Person
Comment Main classifier that classifies a person into various categories
Classification
1902
ElectronicsFan, sellerB, salesmgrB, sellerB1, Man, MaleTeen, SportsFan, Woman, ba, MusicFan, buyerB, FemaleTeen, Over20
Classifier Description
1903
Person is
MaleTeen when
current CPSUserResourceBeanBean.age is equal to 16-19 and
current CPSUserResourceBeanBean.gender is equal to male
Over20 when
current CPSUserResourceBeanBean.age is equal to 20-25 or
current CPSUserResourceBeanBean.age is equal to 25-Up
ElectronicsFan when
current CPSUserResourceBeanBean.interest is equal to electronics
SportsFan when
current CPSUserResourceBeanBean.interest is equal to sport
MusicFan when
current CPSUserResourceBeanBean.interest is equal to music
FemaleTeen when
current CPSUserResourceBeanBean.age is equal to 16-19 and
current CPSUserResourceBeanBean.gender is equal to female
Man when
(
current CPSUserResourceBeanBean.age is equal to 20-25 or
current CPSUserResourceBeanBean.age is equal to 25-Up
) and
About Help InfoCenter
Sign out

도면17


Edit Binding - Microsoft Internet Explorer
2001
Name of the Binding
EPISTAN
2002
Comment about the Binding
2000
2003
Binding Description (click on an underlined value to edit it)
MaleTeen
do MaleTeenAction
Classification
do Action
Otherwise
do Action
Always
do Action
Exclude
do Action
order as is
show all items
Help
Save
Cancel

도면18


Edit Action - Microsoft Internet Explorer
2101
Name of the Action
MaleTeenAction
2102
Comment about the Action
2100
Type of Action
Select content
2103
Action Description (click on an underlined value to edit it):
Select content
whose CPSContentGrpResourceBean.name is equal to maleteengrp
add Condition
order as is
show all items
Help
Save
Cancel