



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년11월05일
(11) 등록번호 10-0992027
(24) 등록일자 2010년10월29일

(51) Int. Cl.

G06F 17/30 (2006.01) G06F 17/00 (2006.01)

(21) 출원번호 10-2007-7005937

(22) 출원일자(국제출원일자) 2005년09월14일

심사청구일자 2008년03월28일

(85) 번역문제출일자 2007년03월14일

(65) 공개번호 10-2007-0058504

(43) 공개일자 2007년06월08일

(86) 국제출원번호 PCT/EP2005/054569

(87) 국제공개번호 WO 2006/030015

국제공개일자 2006년03월23일

(30) 우선권주장

0420675.1 2004년09월17일 영국(GB)

0420679.3 2004년09월17일 영국(GB)

(56) 선행기술조사문헌

Cherry Shu 외, Building JSR 168-Compliant Portlets with Sun Java Studio Enterprise, 2004.5.21.

전체 청구항 수 : 총 10 항

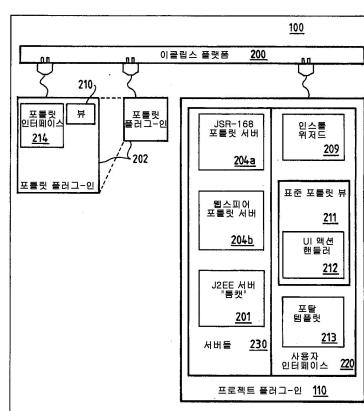
심사관 : 김태근

(54) 클라이언트 플랫폼에 포틀릿을 표시 및 설치하는 방법

(57) 요 약

포틀릿 컨텐츠를 디스플레이하기 위한 클라이언트 컴퓨터 시스템, 방법 및 소프트웨어 툴이 제공된다. 이 시스템은 사용자 인터페이스, 포틀릿 서버 및 포틀릿 뷰 툴을 포함한다. 사용자 인터페이스는 컴퓨터 시스템의 컴퓨트에 의해 제공된 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이 할 수 있다. 포틀릿 서버는 포틀릿 컨텐츠에 대한 요청들을 처리하고, 상기 서버에 설치된 하나 이상의 포틀릿(들)을 가진다. 포틀릿 뷰 툴은 포틀릿 서버에 설치된 포틀릿과 연관되며, 그 연관된 포틀릿을 제어한다. 웹 브라우저는 디스플레이의 뷰 윈도우에서 그 연관된 포틀릿으로부터의 컨텐츠를 디스플레이하는데 사용된다.

대 표 도 - 도2b



(72) 발명자

오마호니 키에란 요셉

아일랜드 퀸라니 파하 파하 이스트

스미스 찰스 로버트 에드워드

아일랜드 코 커스터리지 캐슬타운 더 애비뉴

1

특허청구의 범위

청구항 1

포틀릿(portlet) 컨텐츠를 디스플레이하는 클라이언트 컴퓨터 시스템에 있어서,

상기 컴퓨터 시스템의 컴포넌트에 의해 제공되는 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰(view) 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스;

포틀릿 컨텐츠에 대한 요청들을 처리하는 포틀릿 서버-상기 포틀릿 서버 상에는 하나 이상의 포틀릿(들)이 설치되어 있음-; 및

상기 포틀릿 서버에 설치된 단일 포틀릿과 연관되고, 그 연관된 단일 포틀릿을 제어하고 그 연관된 단일 포틀릿으로부터의 컨텐츠만을 뷰 윈도우에 디스플레이하기 위한 내장된 웹 브라우저를 동작시키는, 포틀릿 뷰 툴

을 포함하는 클라이언트 컴퓨터 시스템.

청구항 2

제 1 항에 있어서, 클라이언트 플랫폼을 동작시키기 위한 수단을 더 포함하며, 상기 사용자 인터페이스는 상기 클라이언트 플랫폼을 사용하여 등록된 플러그-인 컴포넌트들에 의해 제공된 뷰들을 디스플레이할 수 있는 것인 클라이언트 컴퓨터 시스템.

청구항 3

제 1 항에 있어서, 포틀릿 서버 상에 설치된 포틀릿과 각각 연관되는 복수의 포틀릿 뷰 툴을 더 포함하는 클라이언트 컴퓨터 시스템.

청구항 4

클라이언트 컴퓨터 시스템을 이용하여 디스플레이 페이지에 포틀릿 컨텐츠를 디스플레이하는 방법에 있어서,

상기 컴퓨터 시스템의 컴포넌트에 의해 제공된 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스를 제공하고;

클라이언트 컴퓨터 시스템에 포틀릿 컨텐츠에 대한 요청들을 처리하기 위한 포틀릿 서버-상기 포틀릿 서버 상에는 하나 이상의 포틀릿(들)이 설치되어 있음-를 제공하고;

디스플레이 페이지 상에 디스플레이하기 위한 단일 포틀릿을 선택하고;

상기 선택된 단일 포틀릿과 연관된 뷰 윈도우를 상기 디스플레이 페이지 상에 제공하여 상기 뷰 윈도우에서 내장된 웹 브라우저를 동작시키고;

상기 웹 브라우저를 통하여 상기 포틀릿 서버 상에 설치된 상기 선택된 포틀릿으로부터 웹 컨텐츠를 요청하며;

제공된 상기 뷰 윈도우의 디스플레이 페이지 상에서 상기 선택된 단일 포틀릿으로부터 수신된 웹 컨텐츠만을 디스플레이하는 것

을 포함하는 포틀릿 컨텐츠의 디스플레이 방법.

청구항 5

제 4 항에 있어서, 상기 클라이언트 컴퓨터 시스템에서 클라이언트 플랫폼을 동작시키는 것을 더 포함하며, 상기 사용자 인터페이스는 상기 클라이언트 플랫폼을 사용하여 등록된 플러그-인 컴포넌트들에 의해 제공된 뷰들을 디스플레이할 수 있는 것인 포틀릿 컨텐츠의 디스플레이 방법.

청구항 6

클라이언트 컴퓨터 시스템에 포틀릿을 설치하는 방법으로서, 상기 클라이언트 컴퓨터 시스템은 상기 컴퓨터 시스템의 컴포넌트에 의해 제공된 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스를 가지며,

상기 방법은,

포틀릿 컨텐츠에 대한 요청들을 처리하기 위한 포틀릿 서버를 제공하고;

단일 포틀릿에 대한 사용자 선택에 응답하여, 상기 포틀릿 서버에 상기 선택된 단일 포틀릿을 설치하며;

상기 선택된 단일 포틀릿과 연관된 포틀릿 뷰 툴-상기 포틀릿 뷰 툴은 상기 툴과 연관된 단일 포틀릿으로부터의 컨텐츠만을 뷰 윈도우에 디스플레이하기 위한 내장된 웹 브라우저를 동작시키고 상기 연관된 단일 포틀릿을 제어하도록 동작하는 것임-을 생성하는 것

을 포함하는 포틀릿 설치 방법.

청구항 7

클라이언트 컴퓨터 시스템을 사용하기 위한 소프트웨어 툴을 기록한 컴퓨터 판독가능 기록 매체에 있어서,

상기 클라이언트 컴퓨터 시스템은,

상기 컴퓨터 시스템의 컴포넌트에 의해 제공되는 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스; 및

포틀릿 컨텐츠에 대한 요청들을 처리하기 위한 포틀릿 서버-상기 포틀릿 서버 상에는 하나 이상의 포틀릿(들)이 설치되어 있음-를 포함하고,

상기 소프트웨어 툴은,

상기 포틀릿 서버에 설치된 단일 포틀릿과 연관되고, 상기 연관된 단일 포틀릿을 제어하고 그 연관된 단일 포틀릿으로부터의 컨텐츠만을 뷰 윈도우에 디스플레이하기 위한 내장된 웹 브라우저를 동작시키도록 동작하는, 포틀릿 뷰 툴

을 포함하는 것인 컴퓨터 판독가능 기록 매체.

청구항 8

하나 이상의 포틀릿 서버(들)을 사용하여 클라이언트 컴퓨터 시스템에 포틀릿을 설치하는 소프트웨어 툴을 기록한 컴퓨터 판독가능 기록 매체로서,

상기 소프트웨어 툴은,

하나 이상의 포틀릿을 각각 포함하는 하나 이상의 파일들을 식별하는 데이터를 제공하고,

설치용 단일 포틀릿의 선택에 응답하여, 포틀릿 서버로 하여금 상기 선택된 단일 포틀릿을 설치하도록 지시하고, 상기 선택된 단일 포틀릿과 연관된 포틀릿 뷰 툴을 생성하도록 동작하며,

상기 포틀릿 뷰 툴은 그 연관된 단일 포틀릿으로부터의 컨텐츠만을 디스플레이하기 위한 내장된 웹 브라우저를 동작시키고 그 연관된 단일 포틀릿을 제어하도록 동작될 수 있는 것인, 컴퓨터 판독가능 기록 매체.

청구항 9

제4항 내지 제6항 중 어느 한 항에 기재된 방법을 수행하기 위한 컴퓨터 실행가능한 명령들을 기록한 컴퓨터 판독가능 기록 매체.

청구항 10

제7항 또는 제8항에 기재된 소프트웨어 툴을 포함하는 데이터 프로세싱 시스템.

명세서

기술분야

[0001] 본 발명은 데이터 프로세싱 분야에 관한 것으로, 보다 자세하게는 클라이언트 플랫폼에 포틀릿들을 표시 및 설치하는 것에 관한 것이다.

배경기술

[0002]

월드 와이드 웹(world wide web)은 인터넷의 멀티미디어 정보 검색 시스템이다. 웹 환경에서, 클라이언트 머신들은 하이퍼텍스트 전송 프로토콜(HTTP)을 이용하여 웹 서버들과 통신한다. 이 웹 서버들에 의해 사용자들은 하이퍼텍스트 마크업 랭귀지(HTML)로서 알려진 표준 페이지 디스크립션 랭귀지(standard page description language)를 이용하여 텍스트, 그래픽, 이미지, 사운드, 비디오 등과 같은 파일들에 액세스한다. HTML은 기본적인 문서 포맷팅을 제공하고, 개발자로 하여금 다른 서버들 및 파일들로의 하이퍼링크들로서 알려진 접속들을 지정하게 한다. 인터넷 패러다임에서, 서버에 대한 네트워크 경로는 네트워크 접속을 정의하는 특정 sintax(syntax)를 가지는 URL(Uniform Resource Locator)에 의해 식별된다. 소위 웹 브라우저, 예를 들어, 클라이언트 머신에서 동작하는 애플리케이션들인, 네스케이프 내비게이터(네스케이프 내비게이터는 네스케이프사(Netscape Communications Corporation)의 등록 상표임) 또는 마이크로소프트 인터넷 익스플로러(마이크로소프트와 인터넷 익스플로러는 마이크로소프트사의 상표임)는, 사용자들로 하여금 URL을 통하여 링크의 명세(specification)에 의해 정보를 액세스하고, 다른 HTML 페이지들간을 내비게이트하게 한다.

[0003]

웹 브라우저의 사용자가 링크를 선택하는 경우에, 클라이언트 머신은 서버 머신이 위치되는 특정 네트워크 IP(인터넷 프로토콜) 어드레스에 호스트명(URL에서 사용)을 매핑하기 위하여 네이밍(naming) 서비스에 대한 요청을 생성한다. 네이밍 서비스는 그 요청에 응답할 수 있는 IP 어드레스를 리턴시킨다. IP 어드레스를 이용하여, 웹 브라우저는 서버 머신에 대한 접속을 구축한다. 서버 머신이 이용 가능한 경우에, 이 서버 머신은 웹 페이지를 리턴시킨다. 사이트내에서의 추가적인 내비게이션을 용이하게 하기 위하여, 웹 페이지는 통상적으로 "앵커(anchors)" 또는 "링크"로서 알려진 하나 이상의 하이퍼텍스트 레퍼런스("HREF; hypertext reference")를 포함한다.

[0004]

포탈은 일반적으로 다양 다른 소스들로부터의 컨텐츠를 모으는 웹 애플리케이션이고, 포탈 웹 페이지 내에 컨텐츠를 나타내고, 사용자들에게 맞춤화된 컨텐츠를 제공하기 위하여 복잡한 개인화 양상(sophisticated personalisation features)을 가질 수도 있다. 포탈 애플리케이션은 케이트웨이에 하나 이상의 백엔드(backend) 소프트웨어 애플리케이션을 제공할 수 있고, 종종 별도의 포탈 서버에 제공될 수 있다.

[0005]

포탈 서버는 통상적으로 웹 컨텐츠를 하나 이상의 포틀릿에 배열한다. 포틀릿은 동적 웹 컨텐츠를 처리 및 생성하는 포틀릿 컨테이너에 의해 관리되는, 웹 컴포넌트이다. 종종 프로그먼트로 불리는 이 컨텐츠는, 포탈에 의해 모아져서 다른 포틀릿들로부터의 컨텐츠와 함께 포탈 페이지를 형성한다. 포틀릿에 의해 생성된 컨텐츠는, 포틀릿에 대한 사용자 구성에 따라 하나의 사용자로부터 또 다른 사용자로 변할 수도 있다.

[0006]

포탈은 계층내의 서버에 배열된 웹 페이지들의 세트에 대하여 내비게이션 프레임워크(framework)를 제공한다. 이 프레임워크는 서버에서 이용할 수 있는 페이지들의 계층을 통한 내비게이션을 허용하는 사용자 인터페이스를 제공한다. 이러한 내비게이션을 제공하는 사용자 인터페이스는, 주제(theme)로서 알려져 있다. 각 페이지는 0 또는 그 이상의 포틀릿을 포함할 수도 있으며, 페이지 배열은 디자인 또는 관리 툴을 이용하여 미리 결정되어 구성된다.

[0007]

표준 서버측 포탈에 대하여, 클라이언트 웹 브라우저는 단일 페이지상의 몇몇 포틀릿의 접합된 출력을 조망하는데 사용된다. 사용자들은 포틀릿들에 의해 예를 들어 폼(form) 또는 이하의 링크들을 제공함으로써 생성된 컨텐츠와 상호작용하여, 포탈에 의해 수신된 포틀릿 동작들을 생성한다. HTTP 요청이 포탈 서버에 의해 수신되는 경우에, 포탈 서버는 그 요청이 포탈 페이지와 연관되는 임의의 포틀릿들에 타겟화된 동작을 포함하는지 여부를 판정한다. 만일 포틀릿에 타겟화된 동작이 있는 경우에, 포탈은 포틀릿 컨테이너로 하여금 포틀릿을 생성하게 하여 그 동작을 처리한다. 포틀릿은 그 동작을 처리하고 새로운 포탈 페이지에 포함될 컨텐츠 프래그먼트를 생성한다. 또한, 요청된 포탈 페이지 상의 모든 다른 포틀릿들은 컨텐츠 프래그를 리프레쉬하고 그 컨텐츠 프래그를 포탈에 제공한다. 그 포탈은 각각의 윈도우에 타이틀 및 제어 버튼을 부가하여 포틀릿 윈도우에 각각의 포틀릿 컨텐츠 프래그먼트를 패키지화한다. 이를 종종 각각의 공급된 컨텐츠 프래그를 '래핑(wrapping)'한다고 하며, 프래그먼트를 래핑하는데 사용되는 부가적인 마크업은 '스킨(skin)'으로 불린다. 이 스키는 포틀릿을 편집(edit) 또는 구성(configure)과 같은 특정 모드에 배치하거나 또는 포틀릿의 디스플레이 상태를 공통 윈도우 임 시스템에서 보여질 수 있는 것과 같은 최대화된 또는 최소화된 비주얼 상태로 변경하는데 사용될 수도 있는 제어 버튼들을 포함할 수도 있다. 그 후, 포탈은 클라이언트에 전송하기 위하여 포틀릿 윈도우들을 완성된 포탈 페이지로 모은다. 웹 브라우저는 클라이언트의 디스플레이 스크린상에 코드를 제공한다.

[0008]

포탈에 의해 서빙되는 페이지로의 내비게이션은, 사용자가 포탈 홈 페이지에 대한 URL을 사용하여 또는 특정 폐

이지들에 대한 URL 링크들을 통하여, 또는 포틀릿의 인스턴스에 대한 URL 링크들을 통하여 개시하는 경우에 주제(theme)에 의해 제공되는 내비게이션 프레임워크를 통하여 이루어 질 수 있으며, 후자의 경우에, 포탈은 그러한 포틀릿을 포함하는 완성된 페이지를 서브업한다.

[0009] 많은 회사들이 그들의 사업 조건을 충족시키기 위하여 포틀릿들의 개발에 많은 투자를 행하였다. 현재, 이들은 단지 클라이언트 웹 브라우저만을 이용하여 액세스할수 있는 서버측 애플리케이션이며, 단지 일반적으로 네트워크의 일부 품을 통하여 포탈 서버에 접속되어 있다. 이는 네트워크에 접속되지 않으면서 사용자들이 이러한 동일한 포틀릿들을 계속해서 이용할 수 있다는 점에서 큰 이점을 가진다. 이는 클라이언트측 상의 배치를 위한 독립형 애플리케이션들로서 포틀릿들을 재기록함으로서 달성될 수 있다. 그러나, 이는 모든 포틀릿들을 독립형 애플리케이션들로서 동작시키기 위하여 이러한 모든 포틀릿들을 변경해야 하므로, 서버로부터 클라이언트 측으로의 이동을 방지하려 할 때 비용이 많이 들고 많은 시간이 소요된다. 본 발명은 이러한 문제점들을 해결하려는 것이다.

발명의 상세한 설명

[0010] 본 발명의 제 1 양태에 따르면, 클라이언트 플랫폼에 포틀릿을 설치하는 방법이 제공된다. 이 클라이언트 플랫폼은 컴퓨터 시스템의 컴포넌트에 의해 제공되는 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스를 가진다. 포틀릿 컨텐츠에 대한 요청들을 처리하는 포틀릿 서버가 제공된다. 포틀릿의 사용자 선택에 응답하여, 선택된 포틀릿은 포틀릿 서버에 설치되며, 선택된 포틀릿과 연관되는 포틀릿 뷰 툴이 생성된다. 이 포틀릿 뷰 툴은 뷰 윈도우 내의 상기 툴과 연관된 포틀릿으로부터의 컨텐츠를 디스플레이하기 위한 웹 브라우저를 동작시키고 그 연관된 포틀릿을 제어하도록 사용될 수 있다.

[0011] 또한, 이 방법은, 하나 이상의 포틀릿을 각각 포함하는 하나 이상의 파일들을 제공하는 단계; 이러한 파일들의 리스트를 사용자 선택에 대하여 디스플레이하는 단계; 및 사용자 선택을 검출하는 단계 중 임의의 단계를 포함할 수도 있다.

[0012] 본 발명의 제 2 양태에 따르면, 포틀릿 서버가 제공된 클라이언트 플랫폼에 포틀릿을 설치하는 소프트웨어 툴이 제공되어 있다. 이 툴은 하나 이상의 포틀릿 서버(들)을 가진 클라이언트 컴퓨터 시스템에서 포트릿을 설치하기 위한 소프트웨어 툴을 제공하도록 사용될 수 있으며, 이 툴은 하나 이상의 포틀릿을 각각 포함하는 하나 이상의 파일을 식별하는 데이터를 제공하도록 사용될 수 있다. 설치용 포틀릿의 선택에 응답하여, 툴은 포틀릿 서버에게 선택된 포틀릿을 설치하도록 지시하며, 그 선택된 포틀릿과 연관된 포틀릿 뷰 툴을 생성한다. 포틀릿 뷰 툴은 이와 연관된 포틀릿으로부터의 컨텐츠를 디스플레이하기 위한 웹 브라우저를 동작시키고, 그 연관된 포틀릿을 제어하도록 사용될 수 있다.

[0013] 본 발명의 또 다른 양태에 따르면, 클라이언트 플랫폼을 사용하여 등록된 플러그-인(plug-in)들에 의해 기여된 뷰(view)들의 디스플레이를 허용하는 사용자 인터페이스, 설치된 포틀릿으로부터 웹 컨텐츠를 서빙업하기 위한 하나 이상의 포틀릿 서버, 포탈 서버에 설치되는 포틀릿과 연관되는 포틀릿 뷰 플러그-인, 및 웹 브라우저 모듈을 포함하는 포탈 뷰 툴을 포함하는 클라이언트 컴퓨터 시스템이 제공되며, 여기서 포틀릿 뷰 플러그-인은 포틀릿 서버 상의 포틀릿 인스턴스에 대한 id(identification) 데이터 포함하여, 포틀릿 뷰 툴을 이용하여 클라이언트 플랫폼을 뷰잉하도록 구성되며, 이러한 뷰는 웹 브라우저 모듈을 이용하여 포틀릿로부터 웹 컨텐츠를 디스플레이한다.

[0014] 또한, 클라이언트 컴퓨터 시스템은 각각 하나 이상의 포틀릿을 포함하는 하나 이상의 파일을 포함한다.

[0015] 본 발명의 또 다른 양태에 따르면, 클라이언트 컴퓨터 시스템을 이용하여 디스플레이 페이지상에 포틀릿 컨텐츠를 디스플레이하는 방법이 제공된다. 이 방법은 이 컴퓨터 시스템의 컴포넌트에 의해 제공되는 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스를 제공하는 단계; 및 하나 이상의 포틀릿(들)이 설치되는 포틀릿 서버를 제공하여 포틀릿 컨텐츠에 대한 요청들을 처리하는 단계를 포함한다. 디스플레이용 포틀릿의 선택시에, 뷰 윈도우가 디스플레이 페이지에 제공되며, 웹 브라우저는 그 디스플레이 페이지에서 동작한다. 웹 컨텐츠가 웹 브라우저를 통하여 선택된 포틀릿으로부터 요청되고, 제공된 뷰 윈도우의 디스플레이 페이지에 디스플레이된다.

[0016] 본 발명의 또 다른 양태에 따르면, 클라이언트 컴퓨터 시스템의 컴포넌트에 의해 제공되는 컨텐츠를 각각 디스플레이하는 하나 이상의 뷰 윈도우(들)을 포함하는 페이지를 디스플레이하는 사용자 인터페이스; 및 하나 이상의 포틀릿(들)이 설치되어 포틀릿 컨텐츠에 대한 요청들을 처리하는 포틀릿 서버를 구비하는 클라이언트 컴퓨터

시스템에 대하여 사용되는 소프트웨어 툴이 제공된다. 소프트웨어 툴은, 포틀릿 서버상에 설치되는 포틀릿과 연관되는 포틀릿 뷰 툴을 구비하며, 이러한 포틀릿 뷰 툴은 그 연관된 포틀릿을 제어하고 그 연관된 포틀릿으로부터의 컨텐츠를 뷰 윈도우에 디스플레이하기 위한 웹 브라우저를 동작시키는데 사용될 수 있다.

[0017] 종래 기술 시스템들에 사용되는 포탈 집합(aggregation) 모델은 사용되지 않는다. 그 대신에, 별도의 뷰 윈도우가 각각의 포틀릿과 연관된다. 웹 브라우저는 이들의 뷰 각각에서 동작한다. 포틀릿 뷰의 브라우저는 그 포틀릿과 연관되는 페이지를 내비게이트하는 것에 관한 것이므로, 포틀릿의 출력을 디스플레이한다. 그 후, 사용자와 포틀릿 사이의 모든 통신은 HTTP를 이용하여 삽입된 브라우저를 통하여 행해진다. 포틀릿의 모드 또는 상태를 변경하기 위한 요청들은 포틀릿 뷰 툴을 통하여 행해질 수 있으며, 이들은 그 포틀릿과 연관된 브라우저를 특정 URL로 리다이렉팅(redirecting)함으로써 포틀릿에 전달된다.

[0018] 본 발명의 바람직한 실시형태들에 의하면 변경되지 않은 상태의 포틀릿을 리치 클라이언트 플랫폼(스마트 클라이언트, 패트(fat) 클라이언트 또는 씩(thick) 클라이언트로도 지칭됨)의 윈도우 내부에서 동작하게 하며, 이러한 리치 클라이언트 플랫폼은 서버에 의해 수행되는 프로세싱이 거의 없거나 또는 없는 상태에서 대부분의 프로세싱을 수행하는 클라이언트이다. 포틀릿의 제어는 포틀릿 마크업으로부터 클라이언트 UI로 이동된다. 사용자의 관점에 있어서, 포틀릿과 임의의 다른 클라이언트측 애플리케이션 사이에는 차이가 없다. 사용자는 포틀릿이 서버측 포틀릿으로서 배치되는 경우에 이들이 포틀릿과 상호작용하는 것과 정확히 동일한 방식으로 포틀릿과 상호작용한다. 포틀릿들은 단순한 위저드 시스템을 이용하여 클라이언트에 설치되어 단순하게 제거될 수 있다. 따라서, 포틀릿들에 대한 클라이언트측 지원이 제공되고, 이는 포틀릿들을 리치 클라이언트 환경에서 국부적으로 동작시키고, 별도의 서버 및 네트워크 접속에 대한 필요성을 제거한다. 이에 의해 이미 기록되어 있는 포틀릿 코드를 임의로 변경하지 않고 서버측 포틀릿들에 투자하는 사업을 허용할 수 있다.

실시 예

[0027] 다음으로, 본 발명의 바람직한 실시형태를 첨부된 도면들을 참조하여 단지 예로서 설명한다.

[0028] 이하의 설명에서, 레퍼런스는 도면의 부분을 형성하는 첨부된 도면들에 부여되고, 이는 하나의 구현예를 예시한다. 다른 구현예들이 이용될 수 있으며, 본 발명의 범위를 벗어나지 않고 구조적 및 기능적 변경이 행해질 수도 있음을 이해해야 한다.

[0029] 클라이언트측 접근방식에서, 별도의 뷰 윈도우가 각각의 포틀릿과 연관되며, 웹 브라우저는 이러한 뷰 윈도우들의 각각 내부에서 동작한다. 포틀릿 뷰의 브라우저는 그 포틀릿과 연관되는 페이지로 내비게이트하는 것에 관한 것이므로, 포틀릿의 출력을 디스플레이한다. 그 후, 사용자와 포틀릿 사이의 모든 통신은 HTTP를 이용하여 삽입된 브라우저를 통하여 행해진다. 포틀릿의 모드 또는 상태를 변경시키기 위한 요청들은 클라이언트 플랫폼 UI를 통하여 행해지며, 이러한 요청들은 그 포틀릿과 연관된 브라우저를 특정 URL로 리다이렉트시킴으로써 그 포틀릿에 전달된다.

[0030] 본 발명의 바람직한 실시형태는, 소스 코드 빌딩 블록들, 플러그-인 프레임웍 및 애플리케이션 툴 발전의 진행을 원활하게 하는 동작 예들을 제공하는 이클립스(Eclipse) 기술에 기초한 클라이언트 플랫폼을 사용한다. 이는 플러그-인으로 불리는 모듈들을 생성, 통합, 배치하는 환경이며, 이러한 환경은 컴퓨팅 환경의 범위에 걸쳐서 사용되는 툴일 수도 있다.

[0031] 이클립스 플랫폼은 워크벤치 사용자 인터페이스(UI) 및 툴 빌더들이 새로운 가능성을 부가하기 위하여 이용하는 공통 도메인-독립형 사용자 상호작용 패러다임들의 세트를 정의한다. 이 플랫폼은 툴 빌더들에 의해 확장될 수 있는 표준 뷰들의 세트로 개시한다. 툴 빌더들은 새로운 뷰들을 부가하고, 새로운 도메인-특정 능력을 협존 뷰들에 플러그인할 수 있다.

[0032] 툴은 워크스페이스로 불리는 이클립스의 파일 공간에서 파일들을 동작시키는 플러그-인으로서 기록되며, 워크벤치에 그 툴-특정 UI를 표면화한다. 이클립스 플랫폼이 착수되는 경우에, 사용자에게는 이용가능한 플러그-인들의 세트로 구성된 통합된 발전 환경을 제공하는 워크벤치 윈도우 디스플레이 페이지가 제공된다. 이클립스는 Linux™ 및 Windows™에 포함되는 동작 시스템들의 넓은 범위에서 동작한다(마이크로소프트 및 윈도우는 미국, 다른 나라들 또는 이를 모두에 있어서 마이크로소프트사의 상표이며, 리눅스는 미국, 다른 나라들 또는 이를 모두에 있어서 리눅스 토발즈(Linus Torvalds)의 상표이다.).

[0033] 이클립스 플랫폼은 Object Technology International Inc에 의해 기고된 이클립스 플랫폼 기술 개요에 개시되어 있으며, 이는 예를 들어 www.eclipse.org에서 찾을 수 있으며, 참고로 여기서 통합되어 있다.

- [0034] 이클립스 플랫폼은 개별 플러그-인들이 배치되는 플러그인 폴더를 포함한다. 각각의 플러그 인은 플러그인 폴더 하에서 그 자신의 폴더에 설치된다. 플러그인은 플러그인 폴더에 속하는, `plugin.xml`로 불리는 XML 메니페스트(manifest) 파일에 개시되어 있다. 메니페스트 파일은 다른 플러그 인들에 대한 플러그-인의 상호접속을 선언하고, 이러한 파일이 플러그 인을 활성화할 것을 알 필요가 있는 것을 이클립스 플랫폼에 알린다. 플러그-인 메니페스트 파일들의 분석된 컨텐츠들은 플러그-인 레지스트리 API를 통하여 프로그램적으로 이용가능하게 되며, 분석된 플러그-인 명세는 플러그-인 레지스트리로 불리는 메모리내(in-memory) 레포지토리(repository)에 숨겨진다.
- [0035] 이클립스 플랫폼은 모든 이클립스 배치에 제공되는 플러그-인 관리 커널 및 임의의 코어 플러그 인들을 제공한다. 이러한 코어 플러그-인들의 id는 이클립스 플랫폼으로 하드코딩되며, 이 플랫폼은 이클립스의 각각의 동작 인스턴스에서 이러한 플러그-인들을 활성화할 것을 알고 있다. 한편, 난(non)-코어 플러그-인들은 다른 플러그-인들에 의해 요구되는 경우에 활성화된다.
- [0036] 플러그-인들 사이의 상호접속 모델은 단순하며, 플러그-인은 다른 플러그-인들이 기여할 수 있는 임의의 개수의 네이밍된 확장 포인트들 및 다른 플러그-인들이 다른 플러그-인들의 하나 이상의 확장 포인트들에 기여하는 임의의 개수의 확장들을 선언한다. 플러그-인의 확장들은 다른 플러그-인들에 의해 확장될 수 있다. 확장 포인트는 대응하는 API 인터페이스를 가질 수도 있다. 다른 플러그-인들은 이 확장 포인트에 대한 확장들을 통하여 이 인터페이스의 구현에 기여한다.
- [0037] 이클립스 플랫폼 플러그-인 메니페스트 명세 ; 이는 여기서 부록 1로서 통합되며, 플러그-인들을 정의하는데 사용되는 XML 엘리먼트들 및 속성들을 다큐먼트화한다. 각각의 플러그-인은 관련된 다른 플러그-인들의 메니페스트 파일들 내에서 플러그-인을 언급하는데 사용되는, 고유의 식별자(XML 속성 id)를 가진다. 또한, 고유의 식별자는 플러그-인의 동작 인스턴스에 액세스하도록 제공자-공급된 플러그-인 코드내에 사용될 수도 있다.
- [0038] 이클립스 플랫폼에 플러그-인을 설치하는 것은 2 가지 단계를 포함한다. 제 1 단계는 플랫폼의 "플러그인"디렉토리 하에서, 플러그-인을 구성하는 리소스들(메니페스트 파일, 자(jar) 또는 워(war) 파일 및 다른 리소스들)을 플러그-인에 대한 개별적인 폴더에 카피하는 단계를 포함한다. 제 2 단계는 플러그-인의 메니페스트 파일 예를 들어, 파싱된(parsed) 플러그-인 명세(specification)로부터 플러그-인 레지스트리로 데이터를 부가함으로써, 플러그-인을 등록하는 단계를 포함한다. 일단 상기 단계가 행해지면, 플러그-인은 일부 기능을 수행하도록 요청되는 경우에 이클립스 플랫폼에 의해 활성화될 수 있다.
- [0039] 이클립스 플랫폼 사용자 인터페이스(UI) 파라다임은 뷰(view)와 투시도에 기초한다. 뷰는 오브젝트들에 대한 정보를 디스플레이하는 윈도우이며; 투시도는 뷰들이 조직화될 수 있는 디스플레이 페이지 또는 캔버스를 제공한다. 플러그-인들은 플랫폼에 대한 뷰들에 기여할 수 있으며, 이러한 뷰들은 다른 투시도들로 조직화될 수 있다.
- [0040] 이클립스 워크벤치는 뷰 확장 포인트를 포함하는 복수의 확장 포인트들을 제공한다. 이러한 뷰 확장 포인트를 통하여 다른 플러그-인들은, 이러한 확장 포인트에 하나 이상의 확장부를 제공함으로써, 플랫폼에 대한 뷰들에 기여할 수 있다.
- [0041] 도 1 을 참조하면, 3 개의 다른 투시도(102a, 102b, 102c)를 가지는 클라이언트 플랫폼의 디스플레이의 간단한 일례가 도시되어 있다. 투시도(102a)는 가장 상부에 디스플레이되고, 5 개의 뷰(104)를 포함하며, 이 중 3 개의 뷰는 다른 클라이언트 플랫폼 컴포넌트들을 디스플레이하는 2 개의 난(non)-포틀릿 뷰 옆에포틀릿 컨텐츠를 디스플레이한다. 포틀릿 컨텐츠를 디스플레이하는 뷰들 각각은, 이클립스 뷰 확장 포인트의 확장부로서 포틀릿 뷰 플러그-인에 의해 기여된다. 포틀릿 뷰들을 이클립스 투시도에 부가하는 방법은 이후에 설명한다.
- [0042] 전제부에서 언급한 바와 같이, 포틀릿들은 통상적으로 포탈 서버들 내부에서 동작하며, 이 포탈 서버들은 페이지 상의 모든 포틀릿들로부터 컨텐츠를 모으고, 그 모아진 컨텐츠를 리턴시킨다. 본 발명에 있어서, 포탈 서버는 다른 포틀릿의 컨텐츠를 이용하여 하나의 포틀릿으로부터 컨텐츠를 모으지 않고 한꺼번에 단일 포틀릿에 의해 제공된 웹 컨텐츠를 리턴시킨다. 이와 같이 '포틀릿 서버'라는 용어는 표준 포탈 서버 프로세스가 변경되었음을 나타내는데 사용된다. 여기서 사용된 바와 같이, '포틀릿 서버'라는 용어는 요청시에 포틀릿으로부터, 마크업 문서들, 임의의 관련 파일들 및 스트립트들을 서빙하는 HTTP와 같은 프로토콜을 이용하는 서버 프로세스를 의미한다. 본 발명은 단일 포틀릿 서버를 이용하여 구현될 수도 있다. 그러나, 포틀릿들을 기록하기 위한 현존 몇몇 사실상(de-facto) 표준들이 존재한다. 바람직한 실시형태는 이러한 표준들 중 2 개(또는 그 이상) 즉, WebSphere™ 포틀릿 및 JSR(Java™ Specification Request)-168 표준을 지원한다. 웹스피어는 뉴욕주 아몬크시

의 인터내셔널 비지니스 주식회사의 등록 상표이다.

[0043] 바람직한 실시형태는 웹스피어 포탈 서버 및 JSR-168 포탈 서버의 변경된 버전들을 이용한 포틀릿 서버들을 제공한다. 이 서버 프로세스는 단일 포틀릿으로부터 즉 다른 포틀릿들로부터의 컨텐츠 수집 없이, 마크업 문서를 제공한다. URL들은 개별적인 포틀릿들을 어드레스화하고, 그 포틀릿에 대한 마크업 프래그먼트를 단독으로 리턴시킨다. 포탈 서버 코드는 임의의 주제들 또는 스킨들을 제공하거나 또는 포틀릿 컨텐츠 프래그먼트의 래핑을 수행하지 않기 위하여 컷 다운된다. 포탈 페이지들은, 디스플레이 페이지들에 따라 내비게이션을 제공하는 이를립스 내의 투시도들을 변경하기 위한 능력을 사용하여, 이를립스 투시도들에 의해 대체되며, 디스플레이 페이지에 디스플레이되는 포틀릿 컨텐츠에 대한 스킨은 그 대신에 아래에 설명된 바와 같이 플러그-인으로 구현되는 포틀릿 뷰 툴에 의해 이를립스 플랫폼에 제공된다.

[0044] 또한, 톰캣 애플리케이션 서버 코드, 웹스피어 및 JSR-168 포탈 서버 코드는, 클라이언트 컴퓨터에 필요하지 않은 서버 보안 양태(로그인/인증 등)와 같은 불필요한 기능을 포함하지 않도록, 가능한한 많이 '슬림 다운(slim down)'된다.

[0045] 포틀릿 서버들은 포틀릿들의 설치 뿐만 아니라 포틀릿 라이프사이클(예를 들어, Init, 서비스 및 파괴(destroy) 방법)을 지원하도록 구성된다. Init 방법에 의해 포틀릿을 초기화할 수 있고, 서비스 방법은 포틀릿이 그 컨텐츠를 제공하는데 요구되는 경우에 콜링되며, 파괴 방법에 의해 포틀릿들은 서비스를 받은 후에 파괴된다.

[0046] 바람직한 실시형태는 톰캣 J2EE(Java™ 2 Enterprise Edition) 애플리케이션 서버의 삽입된 버전을 이용하여 포틀릿 서버들을 호스팅한다. 톰캣은 자바서브릿의 오픈-소스 구현물이며, JavaServer Pages™ 기술은 아파치 소프트웨어 재단의 Jakarta 프로젝트(자바 및 모든 자바 기반 상표는, 미국, 다른 나라 또는 이를 모두에 있어서 선 마이크로시스템즈의 상표임) 하에서 개발된다. 또한, 톰캣 애플리케이션 서버는 데이터베이스 접속 및 EJB(Enterprise JavaBean)들과 같은 J2EE 서비스들을 지원한다. 웹스피어 애플리케이션 서버와 같은 다른 애플리케이션 서버들은, 당업자에 의해 인식될 수 있는 바와 같이, 동등하게 적절히 사용될 수 있다.

[0047] 도 2a는 바람직한 실시형태에 따른 시스템의 컴포넌트들 사이의 상호작용들의 개요를 나타낸다. 이 컴포넌트들은 톰캣 애플리케이션 서버(201), 애플리케이션 서버 인터페이스(API; Application server Interface)(205), 지원될 각 타입의 포틀릿에 대한 포틀릿 서버로서 여기서는 웹스피어 포틀릿 서버(204b), JSR-168 포틀릿 서버(204a), 포틀릿 인터페이스(203a, 203b)(이는 포틀릿 서버들의 일부분으로서 구현됨), 포틀릿 서버(204a)들 상에 설치된 JSR-168 포틀릿(206)들 중 하나와 연관된 JSR-168 포틀릿 뷰 플러그-인(202) 및 포틀릿 뷰 플러그-인(202)에 의해 기여된 클라이언트 뷰(도 2a에 도시되지 않음) 내에서 동작하는 웹 브라우저(207)를 포함한다. 이 웹 브라우저는 HTTP(208)를 이용하여 톰캣 서버(201)와 통신한다.

[0048] 도 2b는 포틀릿 뷰 툴들이 포틀릿 뷰 플러그-인들로 제공되는 바람직한 실시형태의 이를립스 플러그-인 구조를 나타낸다. 도시된 바와 같이, 클라이언트 시스템(100)은 복수의 포틀릿 뷰 플러그-인(202)(2 개 도시됨) 및 프로젝트 플러그-인(110)을 포함한다. 프로젝트 플러그-인은 클라이언트 시스템을 구현하는데 사용되는 코드를 포함하며, 이 코드는 복수의 UI 컴포넌트(220) 및 복수의 서버 컴포넌트(230)를 포함한다. (이들이 도 2b에서 단일 플러그-인(110)을 형성하는 것으로 도시되어 있지만, 이들은 선호도에 따라서 많은 별도의 플러그-인들로 구현될 수 있다.) UI 컴포넌트(220)들은 인스톨 위저드(209), 배선(wiring) 툴(215), UI 액션 핸들러(212)를 가지는 표준 포틀릿 뷰(211) 및 포탈 템플릿(213)을 포함한다. 표준 포틀릿 뷰 코드(211)는 포틀릿 컨텐츠를 뷰 윈도우에 래핑하고, 제어 아이콘들을 제공하고, 이와 연관된 포틀릿의 뷰를 디스플레이하는 경우에 웹 브라우저 등을 동작시키기 위하여 각각의 포틀릿 뷰 플러그-인에 의해 사용된다.

[0049] 포틀릿 인스톨 위저드 컴포넌트(209)는 포틀릿 인스톨 메뉴 및 포틀릿 뷰 플러그-인(202)들을 생성하는 인스톨 위저드의 구현물을 제공하며, 이에 대해서는 후술한다.

[0050] 표준 포틀릿 뷰(211)는 각각의 포틀릿 뷰 플러그-인에 의해 요구되는 코드를 제공하여 그 뷰를 구현한다. 이에 의해 포틀릿 뷰 툴들은 포틀릿 뷰들에 대한 프레임들, 뷰 프레임들에 사용하기 위한 포틀릿 제어 아이콘들을 제공하고, 각 포틀릿 뷰에서 웹 브라우저를 동작시킨다. UI 액션 핸들러(212)는 제어 액션을 나타내기 위하여 사용자들이 포틀릿 제어 아이콘들을 클릭하는 것에 응답하여 포틀릿 뷰 툴에 의해 실행되는 코드를 포함한다.

[0051] 포탈 템플릿 컴포넌트(213)는 사용자가 포틀릿 인스톨 위저드(209)를 선택 및 개방할 수도 있는 디폴트, 이니셜(initial) 투시도를 제공한다.

[0052] 클라이언트 위의 인스톨 위저드에 의해 설치된 각각의 포틀릿에는 그 자신과 연관된 포틀릿 뷰 플러그-인(202)

이 제공된다. 어느 포틀릿 서버에 포틀릿이 설치되는지, 상기 위저드에 의해 제공된 포틀릿 인터페이스 확장부(214)가 어느 포틀릿 인터페이스(203a, b)를 참조하는지에 대한 문의사항은 설치되는 포틀릿의 타입 즉, JSR-168 포틀릿 또는 웹스피어 포틀릿인지 여부에 따른다.

[0053] 도 2b 의 좌측의 포틀릿 뷰 플러그-인(202)은 2 개의 확장부 즉, 사용자 정의된 서버 확장 포인트(232)를 통하여 서버들을 접속하는 포틀릿 인터페이스 확장부(214) 및 이클립스 확장 포인트를 확장하는 뷰 확장부(210)인 컨텐츠를 나타내도록 확장되어 있다. 뷰(210)는 웹 브라우저(207)를 동작시키는데 요구되는 코드를 획득하는 표준 포틀릿 뷰(211)을 참조한다. 웹 브라우저(207)는 포틀릿 인스턴스와 연관된 플러그-인이 존재하는 포틀릿 서버로 부터 HTTP를 통하여 웹 컨텐츠를 수신하는데 사용될 수 있다. 포틀릿 인터페이스 확장부(214)는 그 포틀릿 서버에 대한 액세스 데이터를 포함하고, 뷰(210)로 하여금 관련 포탈 서버상의 연관 포틀릿 인스턴스와 통신하게 한다. 포틀릿 인터페이스 확장부(214)는 성명(name) 및 이와 연관된 포틀릿의 id 를 포함하며, 이를 이용하여 관련 포틀릿 인터페이스(203a 또는 203b)를 참조한다.

[0054] 포틀릿 서버들(204a 및 204b)은 포틀릿 컨텐츠에 대한 요청이 수신되는 경우에, 관련 포틀릿 서버(즉, 웹스피어 포틀릿에 대한 웹스피어 포틀릿 서버 및 JSR-168 포틀릿에 대한 JSR-168 포틀릿 서버)가 다른 포틀릿들로부터의 컨텐츠를 사용하여 컨텐츠를 모으지 않고, 타이틀 바 및 제어 버튼 등을 사용하여 그 컨텐츠를 포틀릿 윈도우로 패키지화하지 않고 그 개별 포틀릿의 컨텐츠만을 리턴시킨다.

[0055] 포틀릿의 뷰 플러그 인은 웹 브라우저를 사용하여 포틀릿의 구성 데이터에 따라 플러그-인이 제공하는 뷰 윈도우 내부에 이 컨텐츠를 디스플레이한다.

[0056] 도 2c 에 나타낸 바와 같이, 클라이언트 컴퓨터 시스템(100)은 이클립스 플랫폼의 워크벤치 확장 포인트(222)들에 플러그인되는 복수의 UI 확장부 및 프로젝트 플러그-인(110)에 의해 선언된 서버 확장 포인트(232)로 플러그인되는 서버 확장부를 포함한다.

[0057] 도 2a 및 도 2b 를 참조하여, 플랫폼 런타임 프로세스를 설명한다. 먼저, 이클립스 플랫폼은, 등록된 플러그-인들의 메니페스트 파일들을 판독하고, 메모리내 플러그-인 레지스트리를 설치하고, 등록된 플러그-인들을 개시하는 것을 시작한다. 이와 같이 프로젝트 플러그-인(110)은 초기화되고, 이클립스 플랫폼과 함께 사용될 수 있다. 또한, 임의의 등록된 포틀릿 뷰 플러그-인(202)들은 플랫폼에 의해 초기화된다. 사용자가 포틀릿의 뷰를 포함하는 페이지를 디스플레이하거나 또는 새로운 포틀릿을 설치할 때 까지는, 포탈 서버들(204a, 204b)에 특별한 것이 발생하지 않는다.

[0058] 사용자가 포틀릿의 뷰를 포함하는 디스플레이될 투시도를 선택하는 경우에, 연관된 플러그-인은 포틀릿의 마크업 컨텐츠를 획득해야 하므로, 이를 검색하기 위하여 콜을 행한다. 먼저, 연관된 플러그-인은 관련 포틀릿 서버(204a, 204b)가 개시되었는지를 알아보기 위하여 그 포틀릿 타입에 대한 인터페이스 오브젝트를 콜링한다. 포틀릿 서버가 개시되지 않은 경우에 포틀릿 인터페이스(203a, 203b)는 애플리케이션 서버(201)(API(205)를 통하여)에 알려 관련 포틀릿 서버(204a, 204b)를 개시한다. 포틀릿 서버가 개시되는 경우에, 설치된 포틀릿(206)들 모두가 초기화된다. 동작된 포틀릿과 연관된 포틀릿 뷰 플러그-인(202)은 이것과 연관된 포틀릿 인스턴스를 식별하는 파라미터를 포함한다. 포틀릿 뷰 툴은 웹 브라우저가 동작하는 뷰 윈도우를 생성하고, 포틀릿 인스턴스의 id 및 요구된 모드를 통과하는 서버 API를 통하여 포틀릿 인스턴스에 문의한다. 브라우저(207)는 HTTP 를 통하여 애플리케이션 서버(201)에 대한 URL 요청을 전송하고, 포틀릿에 의해 제공된 마크업을 포함하는 응답을 수신한다. 그 후, 브라우저(207)는 수신된 마크업에 따라 포틀릿 컨텐츠를 디스플레이한다. 이 포틀릿은 다른 페이지들, 문서들에 대한 링크들을 포함할 수도 있고, 사용자가 이들 중 임의의 것을 선택하는 경우에, 링크된 마크업이 HTTP를 이용하여 브라우저에 의해 검색된다.

[0059] 브라우저(207)와 포틀릿 서버(204a 또는 204b) 사이의 상호 작용은 사용자가 포틀릿(206)과 상호작용함에 따라 지속된다. 포틀릿 모드 또는 상태 변경 요청들 예를 들어, 편집, 구성 또는 헬프 모드는 사용자가 윈도우 프레임의 제어 아이콘들 중 하나 또는 포틀릿 뷰의 스킨을 클릭함으로써 초기화된다. 그 후, 포틀릿 뷰 툴은 어느 제어 버튼이 선택되었는지에 대하여 포틀릿 인터페이스(214)에 활성화된 뷰를 통지하는 UI 액션 핸들러(212)를 실행한다. 포틀릿 인터페이스(214)는 웹 브라우저에 뷰가 새로운 컨텐츠를 요청해야 하는 URL을 전달한다. 그 URL로부터 수신된 웹 컨텐츠에 의해 브라우저는 요청된 모드/상태에서 포틀릿(206)을 디스플레이할 수 있다.

[0060] 다음으로, 포틀릿의 설치를 도 3a, 3b, 3c를 참조하여 설명한다. 플랫폼(200)의 개시시에, welcome 투시도가, 통상적으로 도 3a 에 도시된 바와 같이 '개방 블랭크 포탈 템플릿'과 같은 옵션을 선택할 수 있는 제어 바/버튼을 이용하여 디스플레이된다. 블랭크 포탈 템플릿은, 사용자로 하여금 새로운 투시도들을 생성하게 하고 인스톨

포틀릿 위저드(도 3b 참조)를 이용하여 포틀릿들을 설치하게 하는 툴(이클립스 UI 의 '투시도' 확장 포인트의 확장부(213)로서 구현됨)이다. 인스톨 포틀릿 위저드(209)의 선택시에, 사용자에게는 클라이언트 시스템(100)의 특정 폴더(302)에서 포틀릿 웹 애플리케이션 아카이브(.war) 파일들의 파일 리스트(300)가 제공된다. 사용자가 이 리스트로부터 하나 이상의 포틀릿을 선택하는 경우에, 각 선택된 포틀릿의 .war 파일은 관련 포틀릿 서버 상의 위저드에 의해 설치된다. (단일 .war 파일은 복수의 포틀릿을 포함할 수 있고, 사용자는 인스톨용의 단일 .war 파일 내에 특정 포틀릿들을 선택할 수도 있음.) 각 선택된 포틀릿이 설치된 경우에, 위저드는 사용자에게 성공적인 인스톨 리포트를 제공하며, 이 사용자는 클라이언트 플랫폼을 재개시하여 사용된 특정 플랫폼에 따라 새로이 설치된 포틀릿들을 사용할 수도 있다. 또한 위저드는 각각의 설치된 포틀릿에 대하여 클라이언트 플랫폼 뷰를 제공하며, 이 메카니즘은 이후에 더 상세히 설명한다. 이와 같이 이러한 뷰들은 이용 가능한 뷰들의 플랫폼 리스트에 부가되며, 상기 플랫폼 리스트는 모든 뷰 확장부를 나타내며, 개발자 또는 사용자는 새로운 투시도들과 같은 새로운 디스플레이 페이지들을 설계할 수 있다.

[0061] 도 4a 를 참조하여, 새로운 투시도를 설계하기 위하여, 개시(400)시에, 사용자에게는 사용자가 블랭크 포탈 템플릿을 선택 및 개방(단계 402)할 수 있는 메뉴 페이지(이클립스 UI 확장 포인트 'ActionSets'의 확장부로서 구현됨)가 제공된다. 사용자는 상기 템플릿의 드롭 다운 메뉴로부터 '뷰 보기(show view)'옵션을 선택하며, 이는 이용 가능한 포틀릿 뷰 및/또는 클라이언트 플랫폼 컴포넌트 뷰들의 리스트로서 도시된다(단계 404). 리스트화된 뷰들 중 하나를 선택(단계 406)하면, 현재의 투시도에서 뷰가 예시되고 개방된다. 사용자는 생성되는 페이지에 더 많은 뷰들을 부가할 수 있고, 리사이징, 리포지셔닝 예를 들어 드래그 앤 드롭 등과 같은 기능들을 통하여 페이지상에 뷰들을 조직화하며(단계 4510), 도 1 에 나타낸 바와 같이 포틀릿들의 뷰들 옆에 난-포틀릿 클라이언트 플랫폼 컴포넌트들의 뷰들을 나타내는 투시도를 설계할 수 있다. 생성된 투시도는 포탈의 새로운 페이지로 전환하는 것과 유사한 방식으로, 필요에 따라, 리콜될 수 있는 미리 정의된 투시도(412)의 리스트에 저장될 수 있다. 투시도가 폐쇄되는 경우에, 투시도상의 뷰들도 폐쇄된다.

[0062] 도 4b 를 참조하여, 사용자가 저장된 투시도를 디스플레이하도록 요구하는 경우에, 요구된 투시도는 '투시도 개방' 옵션 하에서 드롭-다운 메뉴로부터 선택된다(단계 414). 그 후, 플랫폼(200)은 선택된 투시도의 일부인 뷰들을 가지는 모든 플러그-인들을 예시(단계 416)하고, 이들의 메니페스트 파일들을 판독한다. 이 메니페스트 파일들의 정보로부터, 플랫폼은 각각의 뷰 윈도우에 대하여 프레임들 및 임의의 필요한 툴 바 및 메뉴를 생성(단계 418)하고, 웹 브라우저를 각각의 예시된 포틀릿 뷰에 삽입하도록 포틀릿 뷰 툴을 이용한다. 포틀릿 뷰 플러그-인들에 의해 기여된 뷰들에 대하여, 각각의 관련 플러그-인이 활성화되어 그 삽입된 웹 브라우저를 통하여 이와 연관된 포틀릿의 컨텐츠에 대한 요청을 전송(단계 420)한다. 이 요청이 관련 포틀릿 서버로 향하게 하는 애플리케이션 서버로 이동한다. (이를 행함으로써, 애플리케이션 서버는 요청 URL 을 조사하고 컨테이너들 예를 들어 포틀릿 서버들에 대한 매핑 테이블을 검사함으로써 그 요청을 처리하기 위한 정확한 컨테이너를 판정한다.) 그 후, 투시도는 임의의 포틀릿 뷰들이 삽입된 웹 브라우저들을 통하여 수신된 컨텐츠를 나타내는 상태로 사용자에게 디스플레이된다.

[0063] 그 후, 사용자는 임의의 디스플레이된 뷰들과 상호작용할 수도 있다. 사용자가 예를 들어 뷰에 디스플레이된 URL 링크를 클릭함으로써 포틀릿 뷰와 상호작용하는 경우에, 삽입된 웹 브라우저는 HTTP 를 통하여 포틀릿 서버로부터 URL 을 가지는 컨텐츠를 요청한다. 포틀릿 서버는 어드레싱된 포틀릿으로부터 html 컨텐츠를 리턴시키고, 삽입된 웹 브라우저에 의해 디스플레이된다. 투시도의 뷰들의 나머지는 리프레시되지 않는다.

[0064] 다음으로, 도 5 를 참조하여, 애플리케이션 서버 상에 포틀릿을 설치하는데 포함되는 기능적 단계들을 설명한다. 플랫폼에는 포틀릿 인스톨 위저드 컴포넌트(209)가 제공되며, 이러한 컴포넌트는 사용자에 의해 선택되는 경우(단계 502)에, 사용자로 하여금 war 파일들의 리스트로부터 특정 포틀릿 war 파일을 선택하게 하는 사용자 인터페이스를 제공한다. war(web application archive) 파일은 포틀릿에 대한 정규 패키징 파일이다. 포틀릿의 사용자 선택에 응답하여(단계 504), 인스톨 위저드(209)는 war 파일로부터 포틀릿 탑입을 결정하고, 포틀릿 인터페이스(203)의 인스턴스를 생성하고, 인스톨 위저드가 아직 초기화되어 있지 않은 경우에 관련 포틀릿 서버(204a, 204b)를 초기화한다. 인스톨 위저드(209)는, 포틀릿 인터페이스(203a, 203b)를 통하여, 포틀릿 서버(204a, 204b)에 지시하여 포틀릿 .war 파일 (211)을 설치하며, 이에 의해 포틀릿 서버의 포틀릿(206)의 인스턴스를 새롭게 한다(단계 506). 이 프로세스에서, 포틀릿 서버는 포틀릿 파일(들)을 설치하고, 그 포틀릿들을 관리하는 포틀릿 컨테이너의 레지스트리에 포틀릿 세부사항들을 부가한다.

[0065] 포틀릿 .war 파일(211)의 성공적인 설치에, 인스톨 위저드(209)는 그 포틀릿과 연관되며 그 자신의 메니페스트 파일을 포함하는 새로운 포틀릿 뷰 플러그-인(202)를 생성한다(단계 508). 메니페스트 파일은 디스플레이 라벨들(예를 들어, 플러그-인 뷰의 타이틀)을 목록화하고, 뷰 플러그-인에 의해 요구되는 확장부를 제어할 뿐만 아

니라 2 개의 확장부 이클립스 UI 뷰 확장 포인트의 뷰 확장부 및 프로젝트 플러그-인(110)에 의해 제공되는 JSR-168 또는 웹스파이어 포틀릿 확장 포인트(232)를 확장시키는 JSR-168 또는 웹스파이어 포틀릿 인터페이스 확장부(214)를 선언한다. 포틀릿 인터페이스 확장부에 의해 포틀릿은 상기 포틀릿과 연관되어 있는 포틀릿이 어느 타입인지를 선언하고, 관련 포틀릿 서버 인터페이스(203a, 203b)를 제공하기 위하여 관련 포틀릿 서버 컴포넌트(204a, 204b)에 포함된 코드를 참조한다. 뷰 확장부는 뷰가 사용자에게 디스플레이되는 경우에 포틀릿 뷰 플러그-인이 동작하는 표준 포탈 뷰 코드를 참조한다. 이를 이용하여, 포틀릿 뷰 툴은 요구된 메뉴를 생성하고, 툴바들은 툴에 의해 포틀릿 뷰에 대한 프레임에 제공된 아이콘들을 인가한다. 일단 포틀릿 뷰 플러그-인이 이를 립스 플랫폼을 사용하여 등록되면, 사용자는 상술한 바와 같이 이클립스 투시도에 새로운 포틀릿의 뷰를 부가할 수도 있다.

[0066] 배열은 본 시스템의 포탈 서버(204)가 복수의 포틀릿로부터 포탈 페이지로 컨텐츠를 모으지 않는다는 점에서 서버측 포탈의 포틀릿 인스턴스의 배열과는 다르다. 그 대신에 각각의 포틀릿은 사용자의 이클립스 플랫폼 스크린에 디스플레이될 수 있는 그 자신의 포틀릿 뷰 윈도우를 가진다. 또한, 포탈 서버는 타이틀 및 제어 버튼을 포함하는 프레임을 각각의 윈도우에 부가하여 각 포틀릿 컨텐츠 프래그먼트를 포틀릿 윈도우에 패키화하지 않는다. 그 대신에, 포틀릿 뷰 툴은, 포틀릿 뷰 플러그-인 메니페스트에 따라, 타이틀바 및 제어 버튼을 포함하는 포틀릿 뷰 윈도우의 프레임을 제공한다.

[0067] 설치된 각각의 포틀릿에 대하여, 새로운 포틀릿 뷰 플러그-인이 인스톨 위저드에 의해 이클립스 플랫폼상에 설치된다. 이클립스 플랫폼은 각각 자신의 프레임, 제어 버튼 및 삽입된 웹 브라우저를 가지는, 별도의 윈도우 또는 뷰에 각각의 포틀릿 뷰 플러그-인을 디스플레이한다.

[0068] 따라서, 본 발명의 바람직한 실시형태에 의하면 변경되지 않은 포틀릿으로 하여금 포틀릿 마크업으로부터 클라이언트 UI로 이동하는 포틀릿을 제어하면서 리치 클라이언트 플랫폼의 윈도우 내부에서 동작하게 한다. 사용자의 관점에 있어서, 포틀릿과 임의의 다른 클라이언트측 애플리케이션 사이에 차이점은 없다. 사용자는 서버측 포틀릿으로서 배치되었던 상태로 포틀릿과 상기 클라이언트측 애플리케이션이 사용자와 상호작용하였던 방식과 정확히 동일한 방식으로 포틀릿과 상호작용할 수 있다. 포틀릿들은 간단한 위저드 시스템을 이용하여 클라이언트 플랫폼에 설치되고, 이와 유사하게 제거될 수 있다. 따라서, 포틀릿들에 대한 클라이언트측 지원이 제공되므로, 리치 클라이언트 환경에서 국부적으로 상기 포틀릿들을 동작시킬 수 있고, 별도의 서버 및 네트워크 접속에 대한 필요성을 제거할 수 있다.

[0069] 지금까지 기술한 본 발명의 실시형태들은, 적어도 부분적으로, 마이크로프로세서, 디지털 신호 프로세서 또는 다른 프로세싱 장치, 데이터 프로세싱 장치 또는 시스템과 같은 소프트웨어 제어된 프로그램가능한 프로세싱 장치를 이용하여 구현가능하며, 전술한 방법들을 구현하기 위한 프로그램가능한 디바이스, 장치 또는 시스템을 구성하는 컴퓨터 프로그램은 본 발명의 양태로서 간주될 수 있음을 이해해야 한다. 컴퓨터 프로그램은 소스 코드로서 구현될 수 있거나 또는 프로세싱 디바이스, 장치 또는 시스템 상에 구현하기 위하여 편집될 수 있거나 또는 예를 들어 오브젝트 코드로서 구현될 수도 있다. 또한, 이러한 컴퓨터 프로그램을 포함하는 제품도 본 발명의 양태로서 간주될 수 있다.

[0070] 다양한 기능들을 구현 또는 정보를 전달하기 위한 컴퓨터 프로그램이, 하나 이상의 DVD/CD-ROM 및/또는 플로피 디스크들과 같은 매체에 공급되어 예를 들어 하드 디스크에 저장될 수 있다. 또한, 데이터 프로세싱 시스템에 의해 구현될 수 있는 프로그램은, 예를 들어 원격통신 네트워크 및/또는 인터넷을 통하여 원격통신 매체 상에 공급되어 전자 신호로서 구현될 수도 있다. 무선 전화 네트워크를 통하여 무선 단말로서 동작하는 데이터 프로세싱 시스템에 대하여, 원격통신 매체는 컴퓨터 프로그램 및 데이터를 나타내는 적절히 암호화된 신호들을 운반하는 무선 주파수 반송파일 수도 있다. 옵션으로, 반송파는 광섬유 링크에 대한 광 반송파 또는 원격통신 시스템에 대한 임의의 다른 적절한 반송 매체일 수도 있다.

[0071] 본 발명이 전술한 예시적인 실시형태들에 관하여 기술되었지만, 본 발명은 이것들로 한정되지 않고, 본 발명의 범위 내에 포함되는 많은 허용가능한 변형 및 변경예가 존재할 수 있다는 점이 당업자에 의해 이해될 수 있다.

[0072] 본 발명의 범위는 임의의 신규한 양태 또는 여기서 개시되는 양태들의 결합을 포함한다. 본 출원인은 이 애플리케이션 또는 본 발명으로부터 획득되는 임의의 이러한 추가적인 애플리케이션들의 실행 동안에 이러한 양태들 또는 양태들의 결합으로 포뮬레이트될 수도 있음을 주목한다. 특히, 첨부된 청구항들을 참조하여, 독립 청구항들로부터의 양태들은 독립 청구항들의 양태와 결합될 수도 있으며, 각각의 독립 청구항들로부터의 양태들은 단지 청구항들에서 열거된 특정 결합물만이 아니라 임의의 적절한 방식으로 결합될 수도 있다.

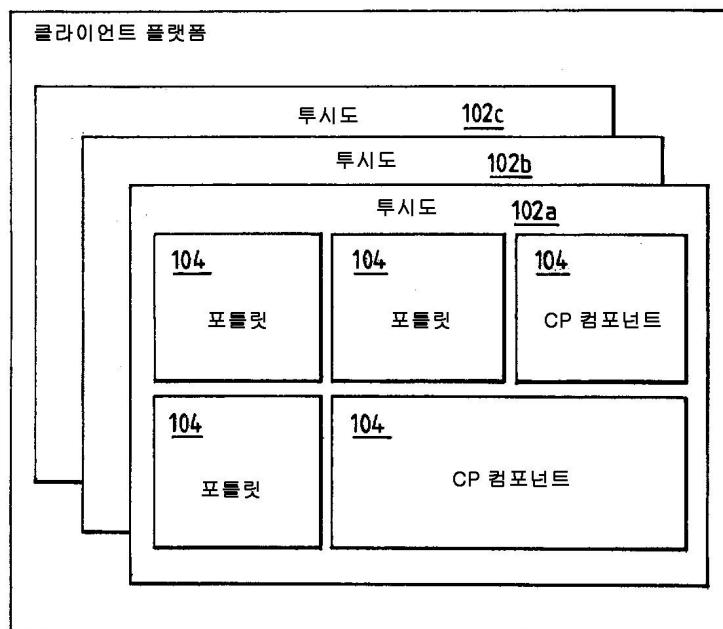
[0073] 의심을 피하기 위하여, 상세한 설명부 및 청구항들 전반에 걸쳐서 사용된 "포함(comprising)"이라는 용어는, "~로만 구성되는"의 의미로서 해석되어서는 안된다.

도면의 간단한 설명

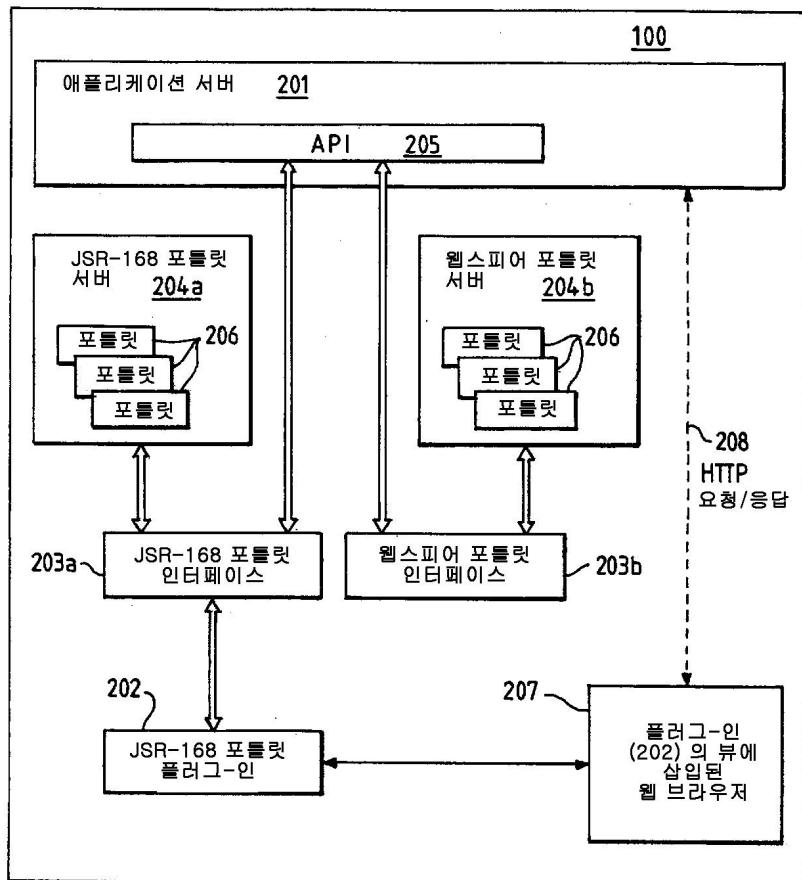
- [0019] 도 1 은 클라이언트 플랫폼의 디스플레이의 일례를 나타낸다.
- [0020] 도 2a 는 본 발명의 바람직한 실시형태에 따른 시스템의 컴포넌트들 사이의 상호작용의 개요를 나타낸다.
- [0021] 도 2b 는 도 2a 의 시스템이 구현될 수 있는 클라인트 플러그-인 구조 및 그 기능적 컴포넌트들을 나타낸다.
- [0022] 도 2c 는 바람직한 실시형태에 따른 클라이언트 컴퓨터 시스템의 확장 포인트 및 확장의 상호 관계를 나타낸다.
- [0023] 도 3a, 3b, 3c 는 본 발명의 바람직한 실시형태에 따라 디스플레이될 수도 있는 디스플레이 페이지들의 스트립샷들 또는 이러한 디스플레이 페이지들의 일부분들을 나타낸다.
- [0024] 도 4a 는 본 발명의 바람직한 실시형태에 따른 디스플레이 페이지를 설계하는데 포함되는 단계들을 나타낸다.
- [0025] 도 4b 는 본 발명의 바람직한 실시형태에 따른 저장된 디스플레이 페이지를 디스플레이하는데 포함되는 단계들을 나타낸다.
- [0026] 도 5 는 본 발명의 바람직한 실시형태에 따른 포틀릿의 설치에 포함되는 단계들을 나타낸다.

도면

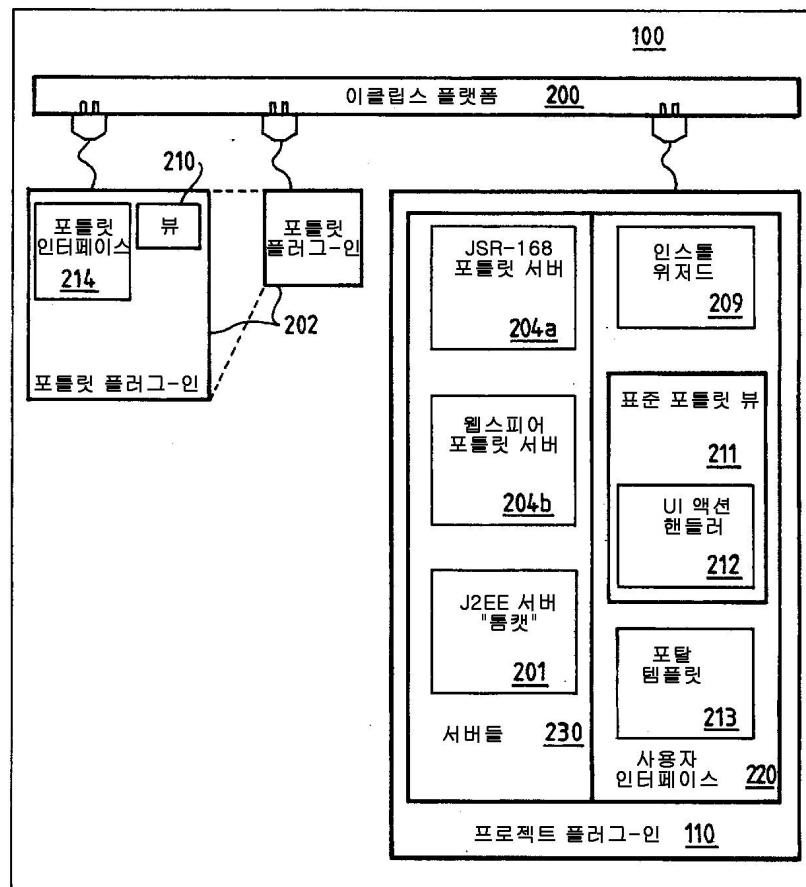
도면1



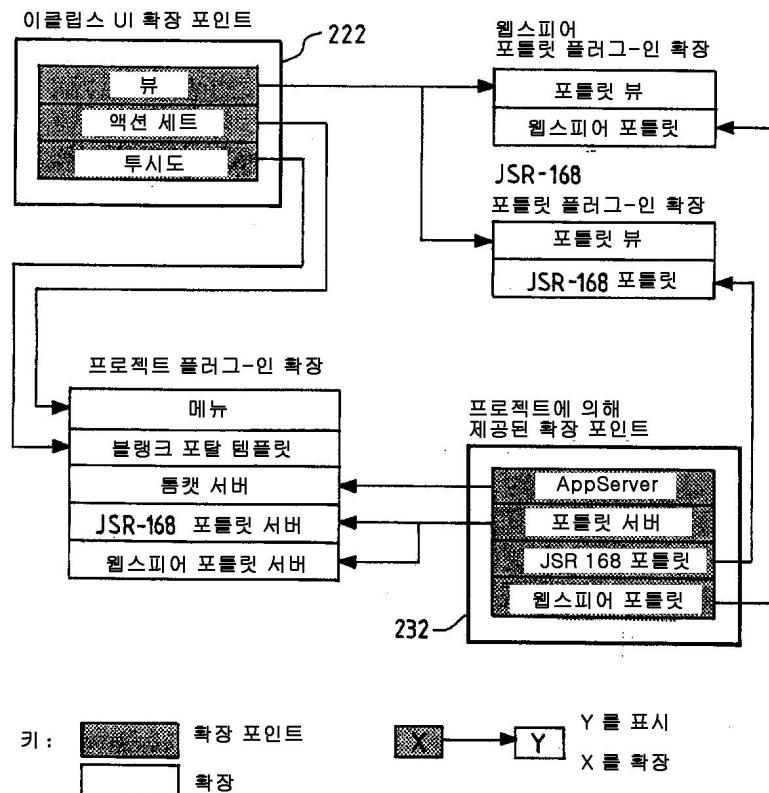
도면2a



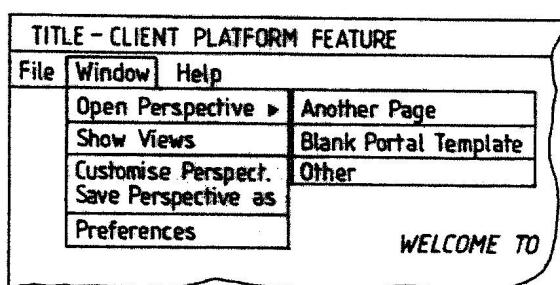
도면2b



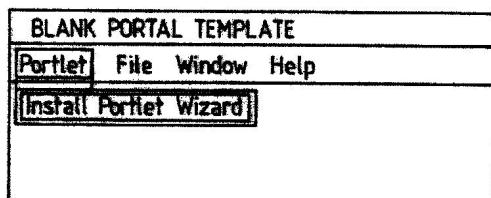
도면2c



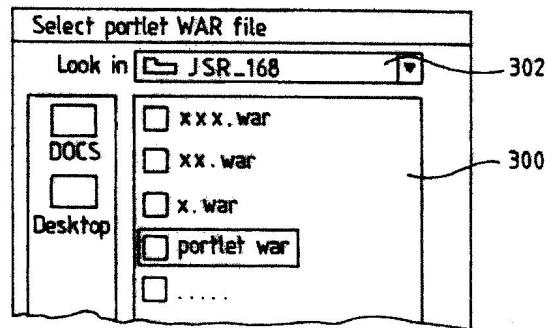
도면3a



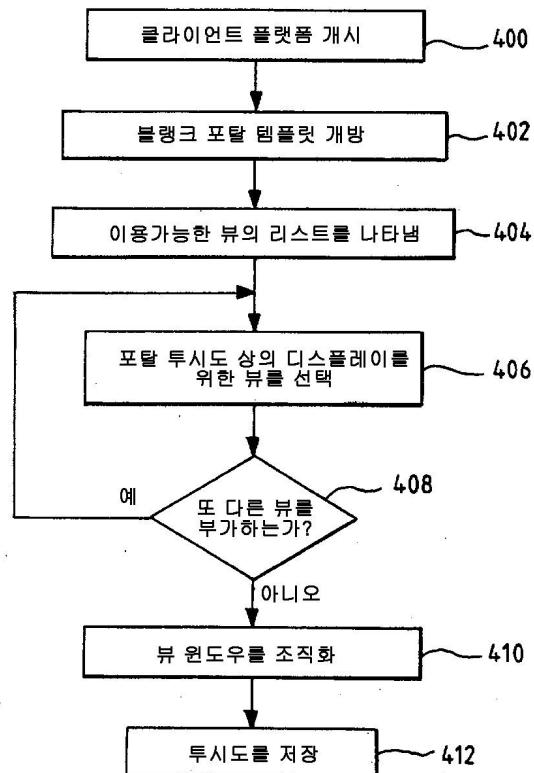
도면3b



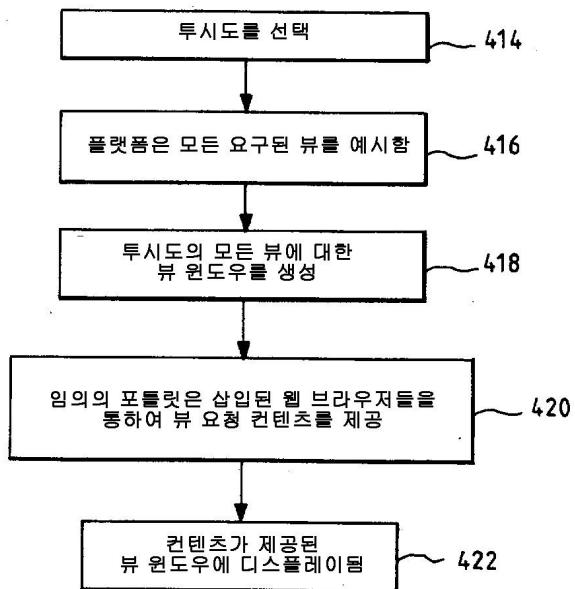
도면3c



도면4a



도면4b



도면5

