



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2011년02월14일  
(11) 등록번호 10-1013046  
(24) 등록일자 2011년01월28일

(51) Int. Cl.

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

(21) 출원번호 10-2006-7022055

(22) 출원일자(국제출원일자) 2005년04월13일

심사청구일자 2008년02월29일

(85) 번역문제출일자 2006년10월24일

(65) 공개번호 10-2007-0007158

(43) 공개일자 2007년01월12일

(86) 국제출원번호 PCT/EP2005/051624

(87) 국제공개번호 WO 2005/106710

국제공개일자 2005년11월10일

(30) 우선권주장

10/838,837 2004년05월04일 미국(US)

(56) 선행기술조사문헌

US06067565 A1\*

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

인터내셔널 비지네스 머신즈 코퍼레이션

미국 10504 뉴욕주 아몬크 뉴오차드 로드

(72) 발명자

헨더슨 로데릭 찰스

미국 노쓰캐롤리나주 27502 아팩스 켈리 글렌 드  
라이브 1805

리 용채

미국 노쓰캐롤리나주 27511 캐리 올데 트리 드라  
이브 107

셀로이 토마스 프란시스

미국 노스캐롤리나주 27617 랄레이 애로우 크릭  
드라이브 아파트#108

(74) 대리인

송승필, 김태홍

전체 청구항 수 : 총 9 항

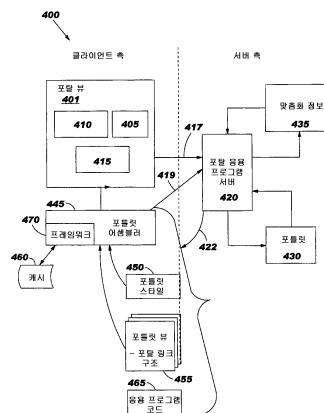
심사관 : 안병일

(54) 클라이언트 측 포틀릿의 프리패치 및 캐시 방법, 시스템 및 컴퓨터 프로그램 제품

(57) 요약

논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 포탈(portal) 정보의 프리페치(prefetch) 및 캐시(cache) 방법, 시스템 및 컴퓨터 프로그램 제품이 제공된다. 사용자가 사용자의 웹 브라우저에서 특정 포틀릿 뷰(portlet view)가 디스플레이되기를 요청하기 전에, 서버로부터 포틀릿 정보를 프리페치한다. 포틀릿 정보는 포틀릿 뷰에 디스플레이되는 콘텐츠 데이터, 콘텐츠 데이터가 어떻게 디스플레이될 것인지 나타내는 메타 데이터 또는 콘텐츠 데이터와 메타 데이터 둘 다의 조합을 포함한다. 프리페치하는 단계는 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 측에서 수행된다. 클라이언트는 프리페치된 포틀릿을 저장한다. 프리페치된 포틀릿 정보가 저장되면, 포틀릿 뷰에서 링크의 클릭과 같은 사용자 요청시, 상기 방법은 저장된 포틀릿 정보를 검색하고 대응하는 포틀릿 뷰에서 포틀릿 정보를 디스플레이한다. 사용자의 시각으로부터, 포틀릿 뷰는 새로운 콘텐츠 정보, 다른 포맷의 미리 보여진 콘텐츠 정보 또는 새로운 콘텐츠 정보와 다른 포맷으로 표현된 오래된 콘텐츠 정보의 조합을 디스플레이할 수 있다.

## 대표도 - 도4



## 특허청구의 범위

### 청구항 1

논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램(client application)에서 캐싱 동작을 수행하는 방법으로서,

포탈의 복수의 포틀릿들(portlet)의 각 포틀릿에 대하여 어느 포틀릿 정보가 프리페치될지를 결정하는 단계;

상기 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트에서 상기 포탈의 복수의 포틀릿들에 대하여 서버로부터 상기 포틀릿 정보를 프리페치(prefetch)하는 단계로서, 상기 프리페치하는 단계는 상기 포틀릿 정보가 요청되기 이전에 수행되는 것인, 상기 프리페치하는 단계;

클라이언트 컴퓨터에 상기 프리페치된 포틀릿 정보를 저장하는 단계;

사용자의 요청시 상기 프리페치된 포틀릿 정보를 디스플레이하는 단계

를 포함하고,

상기 어느 포틀릿 정보가 프리페치될지를 결정하는 단계는,

복수개의 포틀릿 링크 엔트리들 -상기 포틀릿 링크 엔트리들 각각은 대응하는 포틀릿 뷰를 나타내는 데이터를 포함함- 의 계층형 배열을 포함하는 포탈 링크 구조를 구성하고,

대응하는 포틀릿 뷰가 현재 디스플레이되고 있는 엔트리인 현재의 포틀릿 링크 엔트리를 추적(tracking)하며,

어느 대응하는 포틀릿 뷰가 프리페치되어야 하는지를 결정하기 위해, 상기 현재의 포틀릿 링크 엔트리와 상기 복수개의 포틀릿 링크 엔트리들 내의 또 다른 포틀릿 링크 엔트리 간의 간격을 결정하는 것을 포함하는 것인,

논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

### 청구항 2

삭제

### 청구항 3

삭제

### 청구항 4

제 1 항에 있어서,

상기 프리페치(prefetch)하는 단계는 상기 서버에서 포탈 응용 프로그램 서버로부터 포틀릿 정보를 요청하는 것을 더 포함하는 것인, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

### 청구항 5

제 1 항에 있어서,

상기 디스플레이하는 단계는 웹 브라우저를 통해 상기 프리페치된 포틀릿 정보를 디스플레이하는 것을 더 포함하는 것인, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

### 청구항 6

제 1 항에 있어서,

상기 저장하는 단계는 상기 프리페치된 포틀릿 정보를 캐시(cache)저장하는 것을 더 포함하는 것인, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

## 청구항 7

제 1 항에 있어서,

상기 클라이언트 컴퓨터에 프레임워크 코드를 다운로드하는 것을 더 포함하며, 상기 프레임워크 코드는 클라이언트 컴퓨터에서 실행될 때 상기 프리패치하는 단계 및 저장하는 단계를 수행하는 유틸리티 함수를 제공하는 것인, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

## 청구항 8

제 1 항에 있어서,

상기 프리패치된 포틀릿 정보는 브라우저의 전체 표시 영역을 차지하는 HTML 페이지의 정보보다 작은 용량을 갖는 것인, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

## 청구항 9

제 1 항에 있어서,

상기 디스플레이하는 단계는 프리패치된 포틀릿 정보를 사용자의 요청시 검색된 포틀릿 정보와 조합하는 단계를 더 포함하는 것인, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 방법.

## 청구항 10

논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 시스템으로서, 제1항, 제4항, 제5항, 제6항, 제7항, 제8항, 또는 제9항 중 어느 한 항에 따른 방법의 각각의 단계를 수행하는 각각의 수단을 포함하는 시스템.

## 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**

논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 캐싱 동작을 수행하는 컴퓨터 판독 가능한 기록매체로서, 제1항, 제4항, 제5항, 제6항, 제7항, 제8항, 또는 제9항 중 어느 한 항에 따른 방법의 각각의 단계를 수행하는 각각의 프로그램 코드를 기록한 컴퓨터 판독 가능한 기록매체.

**청구항 20**

삭제

**청구항 21**

삭제

**청구항 22**

삭제

**청구항 23**

삭제

**청구항 24**

삭제

**청구항 25**

삭제

**청구항 26**

삭제

**청구항 27**

삭제

**명세서****기술분야**

[0001] 본 발명은 주로 개선된 정보 처리 방법, 시스템 및 컴퓨터 프로그램 제품에 관한 것으로, 보다 상세하게는 네트워크 컴퓨터 환경에서 유리한 포틀릿(portlet)의 프리페치(prefetch) 및 캐시(cache) 방법, 시스템 및 컴퓨터 프로그램 제품에 관한 것이다.

**배경기술**

[0002] “World Wide Web”(WWW)의 인기가 증가함에 따라, 웹 브라우저 또는 브라우저들의 사용이 보다 일반화되었다. 예를 들어, Netscape, Inc.가 시판하고 있는, Netscape Navigator<sup>®</sup>로 알려진 웹 브라우저는 웹을 통해 응용 프로그램을 동작하여 데이터를 뷰잉하는 편리한 방법을 제공할 수 있다. 웹을 통해 이용할 수 있는 일부 응용 프로그램은 C 또는 Java<sup>®</sup>와 같은 모언어로 쓰여질 수 있기 때문에, 사용자와 높은 레벨의 상호 작용을 제공할 수 있다. 특히, 이 유형의 모언어로 쓰여진 응용 프로그램은 특히 집중적인 사용자 상호 작용을 요청하도록 쓰여질 수 있다. 사용자와 응용 프로그램 간의 상호 작용 레벨이 증가함에 따라, 웹 브라우저와 응용 프로그램 간의 통신도 증가할 수 있다. 통신의 이러한 증가는 이용할 수 있는 네트워크 대역폭을 감소시켜, 사용자에게 의해 인식되는 응용 프로그램의 응답 시간 지연을 야기할 수 있다. 예를 들어, 사용자가 응용 프로그램에 의해 관리되는 데이터를 요청할 때마다, 사용자는 일반적으로 웹을 거쳐 응용 프로그램에 송신될 요청, 데이터를 검색하기 위한 응용 프로그램 및 요청된 데이터에 응답하기 위한 응용 프로그램을 기다려야 한다.

[0003] 일반적인 응용 프로그램에 관하여 상기 언급한 문제점에 더하여, 일반적으로 “신생 웹 데스크톱(emerging web

desktop)” 이라고 불려지는 새로운 유형의 환경은 전술된 것과 유사하게 보다 큰 문제점을 야기할 수 있다. 구체적으로, 신생 웹 데스크톱은 사용자에게 일반적으로 포탈(portal)로 불려지는 것에 대한 접근을 제공할 수 있다. 포탈은 사용자가 웹 브라우저에 의해 표시되는 단일 스크린을 통해 다수개의 응용 프로그램에 접근하는 것을 허용할 수 있다. 예를 들어, 일부 포탈은 단일 스크린 상에서 사용자에게 날씨, 스포츠, 주식 정보 또는 기타 각종 데이터를 보여줄 수 있는 응용 프로그램에 사용자가 접근할 수 있게 한다. 각종 유형의 데이터의 각각은 일반적으로 포틀릿(portlet)에 의해 제어된다. 포틀릿은 포틀릿 사양(portlet specification)에서 지정된 인터페이스(interface) 및 행위(behavior)를 지지하고 포탈 응용 프로그램 서버에서 실행하는 컴퓨터 코드이다. 포틀릿 사양의 예로 IBM 포틀릿 API 및 JCP(Java Community Process)에 의해 정의된 Java 포틀릿 스펙의 JAVA™ Standardization Request를 포함한다.

[0004] 관리, 맞춤 및 전환과 같이 포탈을 관리하기 위해 필요한 많은 프로세싱들은 브라우저와 응용 프로그램 간에 이용할 수 있는 대역폭에 대해 보다 크게 요구할 수 있다. 이러한 이용할 수 있는 대역폭에 대해 증가되는 요구는 또한 사용자의 시각으로부터 응용 프로그램의 응답 시간의 감소로 표현될 수 있다.

[0005] 그들의 다른 단점 중에, 웹 브라우저를 포함한 종래의 접근 방법은 현재 캐시(cache)된 정보의 세분성(granularity)으로 인해 제한된 최적화 기회를 제공한다. 예를 들어, 하나의 종래의 접근 방법은 단지 전체 HTML(hypertext markup language) 페이지를 참조하는 URLs(uniform resource locator)을 캐시하거나, 또는 일반적으로 브라우저의 전체 표시 영역을 차지하는 전체 HTML 페이지 자체를 클라이언트에 캐시할 수 있다. 다른 종래의 접근 방법으로는 클라이언트와 서버 간의 응답 시간 지연을 어드레스(address)하지 않는 서버 측 캐싱을 포함한다.

[0006] 웹을 거쳐 데이터에 접근하는 웹 브라우저를 사용할 때 일어나는 일부 성능 문제를 해결하기 위해, 일반적으로 XForm으로서 불려지는 기술을 사용하는 것이 알려져 있다. 특히, XForm 기술은 웹 브라우저에서 표시될 데이터를 데이터가 어떻게 제시될 지와는 별개로 처리할 수 있다. 그러나, XForm 기술은 웹을 거쳐 응용 프로그램 또는 데이터에 접근하는 웹 브라우저를 사용할 때 일어날 수 있는 상기 언급된 일부 성능 문제를 해결하지 않는다.

### 발명의 상세한 설명

[0007] 다수의 양상 중에서, 대역폭 및 응답 시간 요건에 부과되는 증가 요구로 인해, 본 발명은 현존하는 웹 응용 프로그램의 동작을 증가시키기 위한 필요성이 있음을 인지한다. 또한 본 발명은 방법, 시스템 및 컴퓨터 프로그램 제품이 사용자가 요청하기 전에 사용자에게 의해 소비될 포틀릿 정보를 예상하고 그 포틀릿 정보를 검색할 필요성이 있음을 인지한다. 더욱이, 본 발명은 클라이언트측에서 실행하고, 사용자가 요청하기 전에 포틀릿 정보를 검색하고 클라이언트 컴퓨터에서 포틀릿 정보를 캐시하는 시스템, 방법 및 컴퓨터 프로그램 제품을 제공함에 의해 통상의 포탈 서버 기능의 부담을 덜어주는 유용성을 인지한다.

[0008] 본 발명의 일 양상에 따르면, 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 응용 프로그램에서 포탈(portal) 정보의 프리페치(prefetch) 및 캐시(cache) 방법이 제공된다. 사용자가 사용자의 웹 브라우저에서 특정 포틀릿 뷰(portlet view)가 디스플레이되기를 요청하기 전에, 상기 방법은 서버로부터 포틀릿 정보를 프리페치한다. 포틀릿 정보는 포틀릿 뷰에 디스플레이되는 콘텐츠 데이터(content data), 콘텐츠 데이터가 어떻게 디스플레이될 것인지 나타내는 메타 데이터(meta data) 또는 콘텐츠 데이터와 메타 데이터 둘 다의 조합을 포함한다. 프리페치하는 단계는 논리적으로 분리된 클라이언트/서버 컴퓨팅 환경의 클라이언트 측에서 수행된다. 클라이언트 측은 프리페치된 포틀릿을 저장한다. 프리페치된 포틀릿 정보가 저장되면, 상기 방법은 포틀릿 뷰에서 링크의 클릭과 같은 사용자 요청시 사용자가 그를 요청한 시간에 페치된 정보를 검색하기보다 저장된 포틀릿 정보를 검색하고 대응하는 포틀릿 뷰에서 포틀릿 정보를 디스플레이한다. 사용자가 활동적으로 서버로부터 새로운 정보가 검색되기를 요청하고 있지 않아 사용자의 시각으로부터 응용 프로그램의 응답 시간을 증가시킬 수 있을 때, 이러한 프리페치가 일어난다. 게다가, 사용자의 시각으로부터, 포틀릿 뷰는 새로운 콘텐츠 정보, 다른 포맷의 미리 보여진 콘텐츠 정보 또는 새로운 콘텐츠 정보와 다른 포맷으로 표현된 오래된 콘텐츠 정보의 조합을 디스플레이할 수 있다.

### 실시예

[0017] 이하, 본 발명의 바람직한 실시예와 이에 따른 도면을 참조하여 본 발명을 보다 상세하게 설명할 것이다. 그러나 본 발명은 다양한 형태로 구현될 수 있으며, 여기에 설명되는 실시예에 한정되어 파악되어서는 안 된다. 단지 본 실시예들은 본 발명의 개시가 명확하고 완전하며, 당업자에게 본 발명의 범위를 충분히 전달하기 위해 제

공되는 것이다.

- [0018] 당업자에게 인식되는 바와 같이, 본 발명은 방법, 시스템 또는 컴퓨터 프로그램 제품으로서 구현될 수 있다. 따라서, 본 발명은 하드웨어 실시예, 소프트웨어 실시예 또는 소프트웨어와 하드웨어 양상을 조합한 실시예의 형태로 이루어질 것이다. 또한, 본 발명은 매체 내에서 구현되는 컴퓨터-사용 가능한 프로그램 코드를 갖는 컴퓨터-사용 가능한 저장 매체에 관한 컴퓨터 프로그램 제품의 형태로 이루어질 수 있다. 임의의 적합한 컴퓨터 판독 매체가 하드 디스크, CD-ROMs, 광학 저장 장치 또는 자기 저장 장치를 포함하여 이용될 수 있다.
- [0019] 컴퓨터 프로그램 코드 또는 본 발명에 따른 동작을 수행하는 “코드”는 JAVA<sup>®</sup>, Visual Basic<sup>®</sup>, TSQL, Perl 또는 다양한 다른 프로그래밍 언어와 같은 프로그램 언어에 기초한 객체로 쓰여질 수 있다. 본 발명의 소프트웨어 실시예는 특정 프로그래밍 언어의 구현에 좌우되지 않는다. 코드의 일부는 완전히 중간 서버에 의해 이용되는 하나 또는 그 이상의 시스템에서 실행할 수 있다.
- [0020] 코드는 완전히 하나 또는 그 이상의 서버에서 실행할 수 있거나, 또는 부분적으로는 서버에서, 클라이언트 장치 내 또는 통신 네트워크의 중간 지점에서 프록시 서버로서 부분적으로는 클라이언트에서 실행할 수 있다. 전자의 경우에 관하여, 도 1은 본 발명이 적합하게 구현될 수 있는 예시적인 시스템(100)의 도면이다. 시스템(100)은 워크스테이션(110A), 랩탑 컴퓨터(110B), 셀 폰(110C), 핸드헬드 컴퓨터(110D) 또는 컴퓨터 프로그램 코드를 실행할 수 있는 임의의 다른 컴퓨터 기반 장치와 같은 클라이언트 장치를 포함한다. 클라이언트 장치는 LAN, WAN 또는 다른 인트라넷과 같은 네트워크(130)를 통해 서버(120)에 연결될 수 있거나, 또는 ISP(Internet service provider)를 거쳐 인터넷을 통해 연결이 이루어질 수 있다. 본 발명은 TCP/IP-특정 또는 인터넷-특정이 아닌 것으로 이해하여야 한다. 본 발명은 다양한 유형의 컴퓨터 네트워크를 통해 다양한 전송 및 데이터 링크 프로토콜을 사용하여 구현될 수 있다.
- [0021] 본 발명에 따른 실시예들은 논리적으로 분리된 클라이언트 측/서버 측 컴퓨팅 환경에서 동작할 수 있고, 이하 때때로 클라이언트/서버 환경으로 언급된다. 클라이언트/서버 환경은 클라이언트 프로세스 또는 클라이언트, 서버 프로세스 또는 서버 및 서버로부터 서비스를 요청하는 클라이언트를 포함하는 계산 구조(computational architecture)이다. 일반적으로 클라이언트 및 서버 프로세스가 다른 기계 또는 동일한 기계에서 동작하더라도, 클라이언트/서버 환경은 프로세스 간의 구별을 유지한다. 따라서 클라이언트/서버 환경의 클라이언트 및 서버 측은 도 1에 도시된 바와 같이 논리적으로 분리되는 것으로 간주된다. 일반적으로, 클라이언트 및 서버 프로세스가 분리된 장치에서 동작할 때, 각 장치는 각각의 프로세스의 필요에 따라 맞춤화(customization)될 수 있다. 예를 들어, 서버 프로세스는 대량의 메모리 및 디스크 공간을 갖는 시스템에서 실행할 수 있고, 반면에 클라이언트 프로세스는 종종 하이 엔드 비디오 카드 및 대형 스크린 디스플레이에 의해 제공된 그래픽 사용자 인터페이스를 갖는 시스템에서 실행할 수 있다.
- [0022] 클라이언트는 서버로부터 웹 페이지 또는 포트릿 뷰와 같은 정보를 요청하는 웹 브라우저(web browser)와 같은 프로그램을 포함할 수 있다. 클라이언트의 예로 Netscape Navigator<sup>®</sup> 및 Internet Explorer<sup>®</sup>와 같은 브라우저를 포함한다. 일반적으로 브라우저는 웹 페이지, 웹 포탈, 포털릿, 응용 프로그램 및 웹 서버에 의해 제공되는 다른 자원들을 검색하고 보여주는 그래픽 사용자 인터페이스를 제공한다.
- [0023] 서버는 클라이언트로부터 요청에 응답하는 프로그램을 포함할 수 있다. 일부 서버의 예로는 International Business Machines Corporation's(IBM) family of Lotus Domino<sup>®</sup> servers, IBM's Websphere<sup>®</sup> servers, the Apache server 및 다른 적합한 서버들이 있다.
- [0024] 클라이언트 및 서버는 HTTP와 같은 표준 통신 모드를 사용하여 통신할 수 있다. HTTP 요청/응답 통신 모델에 따라, HTTP 요청은 클라이언트로부터 서버에 송신되고, HTTP 응답은 HTTP 요청에 응답하여 서버로부터 클라이언트에 송신된다. 동작에 있어서, 서버는 클라이언트가 접속을 열고, 웹 페이지, 포털릿 또는 다른 기타 정보와 같은 정보를 요청하기를 기다린다. 응답에 있어서, 서버는 요청된 정보의 사본을 클라이언트에 송신하고, 클라이언트에 접속을 닫고, 다음 접속을 기다린다. 서버가 하나 이상의 클라이언트로부터 요청에 응답할 수 있음을 이해할 것이다.
- [0025] 이하, 본 발명의 실시예에 따른 방법, 시스템 및 컴퓨터 프로그램 제품의 블록도 및 순서도를 참조하여 본 발명을 설명한다. 블록도 및 순서도의 각 블록과, 블록도 및 순서도의 블록들의 조합이, 컴퓨터 프로그램 코드에 의해 구현될 수 있음이 이해된다. 컴퓨터 프로그램 코드는 일반 목적 컴퓨터, 특정 목적 컴퓨터 또는 다른 프로그램 가능한 데이터 프로세싱 장치의 프로세서에 제공되어, 컴퓨터 또는 다른 프로그램 가능한 데이터 프로세싱



장치의 프로세서를 거쳐 실행하는 코드가 블록도 및/또는 순서도 블록 또는 블록들에 특정된 함수를 구현하는 수단을 만들 수 있도록 기계를 생산할 수 있다.

[0026] 컴퓨터 프로그램 코드 또는 명령어는 컴퓨터 또는 다른 프로그램 가능한 데이터 프로세싱 장치에 특정 방식으로 기능하도록 지시할 수 있는 컴퓨터 판독 가능한 메모리에 저장되어, 컴퓨터 판독 가능한 메모리에 저장된 명령어가 블록도 또는 순서도에 특정된 함수를 구현하는 명령 수단을 포함한 제조 품목을 생산할 수 있다.

[0027] 일련의 동작 단계가 컴퓨터 또는 다른 프로그램 가능한 장치에서 수행되도록 컴퓨터 프로그램 코드는 컴퓨터 또는 다른 프로그램 가능한 데이터 프로세싱 장치 상에 로드될 수 있으며, 컴퓨터 또는 다른 프로그램 가능한 장치에서 실행하는 명령어가 블록도 및/또는 순서도 블록 또는 블록들에 특정된 함수를 구현하는 단계를 제공하도록 프로세스가 구현된 컴퓨터를 제조할 수 있다.

[0028] 여기에 사용된 바와 같이, “웹 사이트”의 용어는 홈페이지를 불러오는 시작 파일을 포함한 파일들의 관련된 모음(collection)을 포함할 수 있다. 홈페이지로부터, 방문자는 웹 사이트에서 다른 파일 및 응용 프로그램에 접속할 수 있다. 대형 웹 사이트는 다수의 서버를 이용할 수 있으며, 서로 다르거나 그렇지 않을 수 있고, 지리적으로 분산되거나 그렇지 않을 수 있다. 예를 들어, International Business Machines Corporation의 웹 사이트(<http://www.ibm.com>)는 전세계 지역에서 다수의 웹 서버를 통해 분산된 수천개의 웹 페이지 및 파일로 구성된다.

[0029] 여기에 본 발명을 웹 페이지, 웹 서버 및 기타를 참조하여 설명한다 하더라도, 본 발명은 정보를 갖는 임의의 컴퓨터 디스플레이 가능한 문서에 실시될 수 있음을 이해할 것이다.

[0030] 당업자에게 알려진 바와 같이, 웹 페이지는 일반적으로 텍스트를 포함하고, 그래픽, 사운드, 애니메이션 및 비디오 데이터를 참조할 수 있는, HTML과 같은 표준 페이지 기술 언어를 통해 포맷된다. HTML은 기본 문서 포맷을 제공하고, 웹 콘텐츠 제공자가 일반적으로 강조된 텍스트로 명시되는 앵커(anchor) 또는 하이퍼텍스트 링크(hypertext link)를 다른 서버에 지정하는 것을 허용한다. 사용자가 특정 하이퍼텍스트 링크를 선택 또는 활성화할 때, 사용자의 클라이언트 장치를 구동하는 브라우저는 하이퍼텍스트 링크에 결합된 URL을 호출하는 어드레스를 읽고 해석하고, 그 어드레스에서 브라우저를 웹 서버에 접속하고, 하이퍼텍스트 링크에 관계된 파일의 HTTP 요청과 같은 요청을 만든다. 그 다음, 웹 서버는 디스플레이하는 웹 페이지를 해석하고 전달하는 클라이언트에 요청 파일을 송신한다. 여기에 사용되는 “사용자”의 용어는 소프트웨어 프로세스 또는 인간일 수 있다.

[0031] 웹 브라우저는 때때로 웹 포털 또는 포털로 불리는 것을 보여주는 데 사용될 수 있다. 당업자에게 이해되는 바와 같이, 웹 포털은 클라이언트 및 서버에 관련하여 상기에 기술된 동일한 통신 프로토콜에 따라 동작할 수 있고, 클라이언트는 포털 페이지 또는 포털 뷰를 보여주는 웹 브라우저를 포함하고, 서버는 때때로 웹 브라우저에 요청 정보를 공급하는 포털 응용 프로그램 서버로 언급된다.

[0032] 포털은 단일 프리젠테이션의 디스플레이 또는 때때로 정보의 집합으로 언급되는 다수 소스로부터 정보의 뷰를 나타낼 수 있다. 포털은 종종 달력 및 일정 리스트, 토론 그룹, 발표 및 보고, 뉴스, 추가, 검색, 이메일 및 주소록, 날씨, 지도, 쇼핑 및 기타와 같은 정보를 포함하고, 이들 모두는 다양한 소스 또는 응용 프로그램에 의해 포털에 제공될 수 있다.

[0033] 포털에 의해 제공되는 대부분의 정보는 사용자에게 의해 맞춤화될 수 있다. 예를 들어, My Lycos<sup>®</sup>와 같은 일부 포털은 사용자의 지역의 날씨 예보를 디스플레이하거나, 또는 사용자의 좋아하는 팀의 경기 점수를 디스플레이하도록 맞춤화될 수 있다. 또한, 맞춤화는 그 자체의 외관 및 분위기를 포함할 수 있다. 예를 들어, 일부 포털은 특정 스크린 테마를 사용하여 디스플레이되도록 맞춤화될 수 있다.

[0034] 또한 포털은 사용자가 관심이 있는 콘텐츠를 제공하고 맞춤화하거나, 또는 예상되는 광고자에게 정확한 인구 데이터를 제공하고 그들이 광고 콘텐츠를 적당한 판매 가능자에게 매치하도록 돕는 프로파일링(profileing)을 사용할 수 있다. 따라서 포털과, 최근 뉴스 서비스와 같은 다른 웹 제공자는 빈번히 사용자가 프로파일 형식을 완성하도록 요청한다. 예를 들어, 프로파일 형식은 사용자의 이름, 성, 나이, 주소, 소득 계층, 직업, 오락 활동, 취미 및 기타를 요청할 수 있다. 사용자 프로파일에 제공된 데이터는, 예를 들어, 사용자에게 대해 더 많은 것을 얻고 프린트, 이메일, 다이렉트 메일 광고 또는 기타에 사용자를 목표로 하기 위하여, 접속 정보의 대형 데이터베이스로부터 모집단(population)의 소형 세그먼트의 접속 정보를 선택하는 기술과 같은 데이터 마이닝(data mining)과 함께 사용될 수 있다. 사용자는 포털 페이지를 사용자가 관심있는 정보 또는 트랜잭션(transaction)을 보여주도록 맞춤화할 수 있다. 사용자의 시각으로부터, 포털은 사용자가 구독할 수 있는 콘텐츠 채널 또는 응용 프로

그램이다.

[0035] 포틀릿은 요청을 처리하고 동적 콘텐츠(dynamic content)를 생성하는 서버에서 실행되는 Java<sup>®</sup> 기반 웹 컴포넌트이다. 포탈은 정보 시스템에 표현 계층(presentation layer)을 제공하는 장착형(pluggable) 사용자 인터페이스 컴포넌트로서 포틀릿을 사용한다. 포틀릿 뷰는 포탈의 사용자가 포탈 페이지 내에서 보는 보이는(visible) 능동 컴포넌트에 대응한다. 데스크탑 컴퓨터에서 윈도우와 마찬가지로, 각 포틀릿 뷰는 포탈 뷰 내의 클라이언트 스크린의 일부에 배치되고, 관련 정보가 디스플레이된다. 포틀릿은 사용자에게 의해 보여지는 포탈 페이지의 내부에 포함된 콘텐츠를 생성한다. 포틀릿 자체는 많은 포틀릿 뷰들을 포함할 수 있고, 각각의 뷰는 포틀릿의 특정 상태를 나타낸다. 포틀릿은 일반적으로 서버에서 관리되는 데이터와 상호작용하여 사용자와 트랜잭션을 이룰 수 있다. 각각의 포틀릿 뷰는 트랜잭션의 다양한 상태에 대응할 수 있다. 예를 들어, 포틀릿을 통해 사용자는 주식을 팔 수 있다. 하나의 포틀릿 뷰는 사용자가 포지션을 갖고 있는 다양한 주식을 디스플레이하는 사용자의 증권 거래 계좌 상태를 나타낼 수 있다. 사용자가 그의 포트폴리오(portfolio)에서 그의 주식 중 하나를 클릭할 때, 트랜잭션을 제어하는 포틀릿은 첫 번째 포틀릿 뷰를 두 번째 포틀릿 뷰로 변환할 수 있다. 두 번째 포틀릿 뷰는 사용자가 팔 주식의 수와 사용자에게 의해 받아들일 수 있는 최소 판매 가격으로 참가하는 뷰를 나타낼 것이다. 포틀릿 뷰는 Sun Microsystem's Java<sup>®</sup> Server Pages technology와 같이 콘텐츠 제너레이션(generation)을 사용자 인터페이스와 분리하는 임의의 기술에 의해 개발될 수 있다.

[0036] 도 2는 본 발명에 따라 동작할 수 있는 웹 브라우저에 의해 디스플레이되는 예시적인 포탈 뷰(200)의 개략도이다. 도 2에 도시된 바와 같이, 포탈 뷰(200)는 포틀릿 뷰로 언급되는 다양한 정보의 다수의 뷰를 포함한다. 구체적으로, 포탈 뷰(200)는 각각 주식, 날씨 및 뉴스에 관련된 정보의 디스플레이에 관한 세 개의 포틀릿 뷰(205, 210, 215)를 포함한다. 도 2에 따르면, 포틀릿 뷰(205)는 사용자가 관심이 있는 주식 정보를 보여준다. 포틀릿 뷰(205)는 주식의 가격, 채권, 주식 거래 및 기타와 같은 정보를 디스플레이하도록 맞춤화될 수 있다. 포틀릿 뷰(210)는 사용자가 관심이 있는 특정 지리적 위치에 관련된 날씨 정보를 디스플레이하고, 또한 맞춤화될 수 있다. 포틀릿 뷰(215)는 사용자가 관심이 있는 뉴스 정보를 보여주고, 또한 맞춤화될 수 있다. 동작에 있어서, 사용자는 포탈 뷰(200)에 보여진 정보를 생성하는 임의의 포틀릿에 대해 업데이트된 정보 요구를 작성할 수 있다. 예를 들어, 사용자는 포틀릿 뷰(205)에 대응하는 포틀릿을 통해 업데이트된 주식 시세를 요청할 수 있고, 포틀릿 뷰(210)에 대응하는 포틀릿을 통해 업데이트된 날씨를 요청할 수 있고, 포틀릿 뷰(215)에 대응하는 포틀릿에 가장 최근의 뉴스로 특정 아이템에 관련된 정보를 요청할 수 있다. 또한, 사용자는 포틀릿을 통해 포틀릿 뷰(205, 210, 215)를 디스플레이하는 다른 포맷들을 요청할 수 있다. 어떤 경우, 업데이트된 정보의 요청은 포틀릿 뷰(205, 210, 215)의 적어도 하나의 내부에 디스플레이되는 정보의 일부를 변경시킬 요구를 작성할 수 있다.

[0037] 추가적인 응용 프로그램 코드의 제어시, 포틀릿에 관한 마크업(markup) 및 데이터와 같은 정보는 웹 브라우저와 같은 클라이언트 응용 프로그램에 의해 다음의 디스플레이로 집합될 수 있다. 특히, 논리적으로 분리된 클라이언트 측/서버 측 컴퓨팅 환경에서, 웹 브라우저에 의한 디스플레이를 위해 클라이언트 측에 보유된 정보는 환경의 서버 측에 위치한 포탈 응용 프로그램 서버로부터 웹 브라우저에 의해 요청된 정보로 집합될 수 있다. 따라서, 디스플레이되는 정보의 집합은 서버 측보다 클라이언트 측에 제공됨으로써, 포탈 응용 프로그램 서버에 로드를 감소시키고 웹 브라우저에 할당된 요청 대역폭을 감소시킬 수 있다. 일부 실시예에서, 웹 브라우저는 다수의 포틀릿을 지원하는 포탈에 접속하는 데 사용될 수 있고, 포틀릿의 각각은 원하는 정보를 디스플레이하도록 사용자에게 의해 맞춤화될 수 있다. 정보의 집합 동안에, 웹 브라우저는 단지 요청된 정보에 관련된 포탈의 업데이트된 디스플레이를 위해 필요한 정보를 요청하고, 업데이트된 디스플레이에 필요하지 않은 요청된 정보를 피할 수 있다. 필요하지 않은 요청된 정보를 피함으로써, 포탈 응용 프로그램 서버에 대한 로드가 더욱 감소된다. 클라이언트 응용 프로그램에 의해 디스플레이를 위한 정보의 집합 방법 및 장치는 그 전체가 여기에 참조용으로 사용되고, 2003년 6월 19자에 제출된 미국 특허 출원 제10/464910호의 “METHOD, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PORTLET AGGREGATION BY CLIENT APPLICATIONS ON A CLIENT SIDE OF A CLIENT/SERVER ENVIRONMENT”에 보다 상세하게 기술되어 있다.

[0038] 본 발명에 따른 실시예는 웹 브라우저와 같은 클라이언트 응용 프로그램에서 포틀릿 정보의 프리페칭(prefetching) 및 이후의 캐싱(caching)을 허용하는 기술을 제공한다. 특히, 본 발명은 포틀릿 뷰 간의 계층 관계를 정의하는 포탈 링크 구조(portal link structure)를 제공한다. 포탈 링크 구조는 정의된 범위 내 각 포틀릿 뷰에 참조들을 포함한다. 참조는 URL, 다른 포틀릿으로의 URL, 동적 링크 또는 포틀릿 내부로부터 다음에 접근하기 쉬운 포틀릿 정보에 어떻게 접속하는지 지시하는 다른 적합한 참조일 수 있다. 포탈 링크 구조는 초기에



포탈 응용 프로그램 서버에 의해 생성되고, 이후에 클라이언트에 의해 관리될 수 있다. 본 발명의 따른 일부 실시예에서는, 포탈 링크 구조가 클라이언트에서 초기에 생성되고 관리될 수 있다. 클라이언트는 어느 포털릿 참조들이 검색되었고 어느 포털릿 참조들이 현재 포탈 뷰로서 제공되고 있는지를 판단함으로써 포탈 링크 구조를 업데이트한다.

[0039] 도 3A는 포털릿 뷰(305, 310, 315) 내부에 포함된 참조를 도시한 계층도(300)이다. 예를 들어, 포털릿 뷰(305, 310, 315)는 도 2에서의 포털릿 뷰(205, 210, 215)에 제공된 포털릿 정보에 각각 대응할 수 있다. 일반적으로, 또한 링크로 불러지는 참조는, 실행시 특정 포털릿 뷰에 접속한다. 참조 7 및 참조 8은 포털릿 뷰(205) 내부에 포함된다. 예를 들어, 참조 7은 서버 어드레스에 대응할 수 있고, Ameriquest Mortgage Company에 관하여 더 많은 정보가 네트워크 클라우드(130)에 접속된 서버에서 검색될 수 있다. 이러한 정보로의 사용자 접근은 사용자가 하이라이트 링크(220)와 같은 강조된 링크를 클릭함으로써 디스플레이될 것이다. 참조 8은 주식 시세(225)와 같은 정보를 계속해서 검색하는 요청과 송신되는 파라미터를 이용하는 동적 링크에 대응할 수 있다. 포털릿 뷰(310)는 참조 9 및 10을 포함한다. 포털릿 뷰(315)는 참조 11 및 12를 포함한다. 포털릿 뷰 내부의 참조는 참조 7 및 8에 대해 설명된 바와 같이 사용자에게 디스플레이되거나, 또는 사용자에게 디스플레이되지 않을 수도 있다. 반복적으로 포털릿 정보 내부에 참조되는 각 참조를 판단하고, 그 다음 각 참조 내부에 포함된 포털릿 정보로부터 다음의 참조를 판단함으로써, 트리형의(tree) 참조들이 정의된다.

[0040] 하기에 보다 상세하게 설명되는 바와 같이, 본 발명은 트리형의 참조들을 구성하는 기술을 제공한다. 트리형의 참조들의 레벨 크기 및 수는 자바 스크립트로 쓰여지고 서버에 저장되거나 클라이언트에서 실행되는 응용 프로그램 코드와 같은 응용 프로그램 코드에 의해 구성될 수 있다. 트리형의 참조들을 구성하는 프레임워크(framework)의 응용 프로그램 코드 및 그의 사용은 도 4을 참조하여 보다 상세하게 설명될 것이다.

[0041] 도 3B는 본 발명에 따른 예시적인 포탈 링크 구조(330)를 도시하는 블록도이다. 포탈 링크 구조(330)는 도 3A에 도시된 트리형의 참조들에 기반한, 포털릿 링크 엔트리 P1(335)과 같은 포털릿 링크 엔트리의 계층형 배열(hierarchical arrangement)이다. 응용 프로그램 코드는 포털릿 링크 엔트리의 수와 포탈 링크 구조가 완성되는 범위를 결정한다. 일반적으로, 포털릿 링크 엔트리와 포털릿에 의해 생성된 각 포털릿 뷰 사이는 일 대 일 관계로 이루어진다.

[0042] 포털릿 링크 엔트리 P1(335)은 포털릿 뷰(305)에 대응한다. 포털릿 뷰(305)는 도 3A에 도시된 바와 같은 참조 7을 참조하므로, 포털릿 링크 엔트리 P1(305)은 참조 7에 의해 참조된 포털릿 뷰에 대응하는 포털릿 링크 엔트리 P7(337)에 연결될 것이다. 따라서, 포털릿 링크 엔트리(335, 337) 사이의 참조 관계는 포털릿 링크 엔트리(335)로부터 만입되는 포털릿 링크 엔트리(337)로 도 3B에 도시된다.

[0043] 도 3B에 도시된 계층형 배열 내부에 포털릿 링크 엔트리를 배열하고, 포털릿 링크 엔트리에 대응하는 어느 포탈 뷰가 디스플레이에 제공되고 있는지 탐색함으로써, 현재 제공되는 포털릿 엔트리와 제공되지 않는 포털릿 엔트리 간의 간격에 의해 필요할 것 같은 포털릿 정보가 판단될 수 있다. 예를 들어, P1과 P7 간의 간격이 1이고, P1과 P7의 자식 노드인 포털릿 링크 엔트리 P13와의 간격은 2일 것이다. P1이 현재 디스플레이되고 있다면, 사용자는 일반적으로 참조 13으로의 접속을 얻기 전에 참조 7을 클릭할 것이기 때문에, 1의 간격만 갖는 P7이 P13보다 먼저 프리패치될 것이다. 이 계층형 배열은 어느 포털릿 정보가 사용자에게 의해 요청될 것 같은지 예상하는 기술을 유리하게 제공한다.

[0044] 도 3C는 포탈 링크 구조(330) 내부의 예시적인 포털릿 링크 엔트리(350)를 도시하는 블록도이다. 명확화를 위하여, 포털릿 링크 엔트리(350)는 포탈 뷰(200) 내부에 제공되는 포털릿 뷰(305)에 대응하는 포털릿 링크 엔트리로 설명될 것이다. 포털릿 링크 엔트리(350)는 다양한 유형을 갖는 필드(352, 354, 356, 358, 360, 364)를 포함한다. 단순화를 위해 필드 유형은 도시되지 않는다. 각 필드는 다양한 필드 유형으로 구현될 수 있음을 당업자에 의해 인지되어야 한다. 캐시 가능한 필드(352), 포탈 응용 프로그램 서버 필드(354) 및 다른 포털릿 참조 필드(356)는 일반적으로 현재 포털릿 링크 엔트리의 콘텐츠에 관한 정보를 포함한다. 예를 들어, 캐시 가능한 필드(352)는 포탈 링크 엔트리에 관련된 콘텐츠가 캐시 가능한지 또는 아닌지의 명령어를 포함한다. 이 판단을 하는 데 사용되는 일부 인수로, 관련하는 포털릿 뷰가 동적 결과를 디스플레이하는지를 포함한다. 포털릿 뷰가 사용자에게 트랜잭션을 수행하는 데이터와 상호 작용하도록 촉구할 때, 동적 결과가 발생할 수 있다. 동적 결과를 디스플레이하는 링크에 의해 참조되는 포털릿 뷰는 일반적으로 캐시 가능하지 않을 것이다. 예를 들어, 포털릿에 의해 생성된 포털릿 뷰가 쇼핑 응용 프로그램이라면, 본 발명은 트랜잭션 동안에 누군가가 그 또는 그녀의 마음이 변하도록 사용자를 혼란시키는 것을 막기 위해, 포털릿 뷰를 통해 구매된 상품을 보여주는 포털릿 뷰를 캐시하지 않을 것이다.

- [0045] 포탈 응용 프로그램 서버 필드(354)는 포탈 링크 엔트리에 대응하는 포틀릿 뷰를 생성하는 포틀릿을 제공하는 포탈 응용 프로그램 서버의 명령어를 포함한다. 다른 포틀릿 참조 필드(356)는 동일한 포탈 응용 프로그램 서버에 의해 제공된 포틀릿 참조들의 리스트를 포함하고, 이 포틀릿 링크 엔트리에 대응하는 포틀릿 뷰를 통해 접근하기 쉽다. DOM(document object model) 트리형 노드 필드(358)의 ID(identification)는 이 포틀릿 링크 엔트리에 대응하는 포틀릿 뷰를 디스플레이함으로써 대체될 DOM 내부의 위치를 지정한다. DOM은 스크립트가 문서의 콘텐츠, 구조 및 스타일에 접근하고 업데이트하는 것을 허용하는 W3C 문서 객체 모델 표준에 의해 특정되는 알려진 플랫폼 및 언어 중립 인터페이스이다.
- [0046] Javascript<sup>®</sup>와 같은 DOM-지원 프로그래밍 언어를 사용하여, 포틀릿 뷰 내부의 정보를 처리할 수 있다. 예를 들어, DOM 참조는 클라이언트에 의해 디스플레이되는 HTML 페이지 내로 요청 정보를 직접 삽입하도록 사용될 수 있다. DOM-지원 프로그래밍 언어의 다른 양상은 문서의 임의의 컴포넌트에 접근하고, 문서에 포함된 콘텐츠 정보를 업데이트하고, 독립적으로 정보를 처리하는 능력이다. 예를 들어, Javascript<sup>®</sup> 스크립트는 객체를 문서의 하나의 부분으로부터 다른 곳으로 이동하기 위해 사용될 수 있다. W3C 문서 객체 모델 표준에 대해 더 많은 정보를 위해서는, <http://www.w3.org/DOM/>에서 이용 가능한 W3C 아키텍처 도메인을 참조한다.
- [0047] 콘텐츠의 서버 ID 필드(360)는 서버 범위 내부의 콘텐츠의 유일한 ID를 포함한다. 이 ID는 포탈 응용 프로그램 서버로부터 관련된 콘텐츠를 검색하는 능력을 제공한다. 캐시된 콘텐츠에 링크 필드(362)는 포탈 응용 프로그램 서버보다 국부 캐시로부터 특정 콘텐츠를 검색하기 위해 캐시에서 국부 링크 또는 유일한 ID를 포함한다. 일반적으로, 이 필드는 콘텐츠가 프리페치되고 클라이언트에서 국부 캐시에 저장될 때까지 지정되지 않는다. 다른 포틀릿 링크 엔트리에 링크 필드(364)는 현재 링크 엔트리에 의해 접근하기 쉬운 다른 포틀릿 링크 엔트리로의 링크들의 리스트를 포함한다. 즉, 이 링크 필드(364)는 도 3A에 도시된 바와 같이 다른 포틀릿 뷰를 참조하는 포틀릿 뷰 사이의 탐색 관계와 유사하다.
- [0048] 도 4는 본 발명에 따른 포탈 링크 구조를 이용한 클라이언트/서버 상호작용(400)을 도시하는 블록도이다. 도 4에 따르면, 논리적으로 분리된 클라이언트 측/서버 측 컴퓨팅 환경에서, 클라이언트 측의 포탈(400)은 서버 측의 포탈 응용 프로그램 서버(220)와 동작한다. 포탈 뷰(401)는 웹 브라우저와 같이 클라이언트에 의해 디스플레이될 수 있고, 포탈 뷰(401) 내부에 디스플레이될 수 있는 포틀릿 뷰(405, 410, 415)를 포함한다. 사용자가 초기에 포탈 뷰(401)를 디스플레이하도록 하는 포탈에 접속할 때, 포탈 응용 프로그램 서버(420)에 요청이 송신되어 클라이언트에 제공할 포틀릿 뷰(405, 410, 415)를 검색한다. 포틀릿 응용 프로그램 서버(420)는 포탈 뷰(401)의 사용자에게 관련될 수 있는 맞춤화 정보(435)에 접속함으로써 정보(417)의 요청에 응답한다. 맞춤화 정보(435)는 또한 특정 포틀릿 뷰(405, 410, 415)를 맞춤화하는 정보를 포함할 수 있다. 예를 들어, 특정 포틀릿 뷰의 맞춤화 정보(435)는 포탈 뷰(401)에 대응하여 스킨의 유형 또는 포틀릿이 포탈 뷰에서 어떻게 보여지게 되는지에 관련된 정보를 포함할 수 있다. 또한, 맞춤화 정보(435)는 일반적으로 하나의 포틀릿 뷰에서 디스플레이되는 날씨 정보에 관련된 특정 위치, 하나의 포틀릿 뷰에서 보여지는 특정 주식 펀드 또는 사용자가 관심있는 특정 스포츠 팀과 같은, 사용자에게 의해 희망하는 정보 유형과 관련된 정보를 포함할 수 있다.
- [0049] 포탈 응용 프로그램 서버(420)는 또한 클라이언트 측의 포탈 뷰(401)에 전달될 각 포틀릿 뷰의 컴퓨터 프로그램 또는 포틀릿 코드(430)에 접속한다. 그러나, 하나의 포틀릿은 다수개의 포틀릿 뷰를 생성하여 포틀릿과 생성된 포틀릿 뷰 사이에 일 대 다수 관계가 존재할 수 있음이 인지된다. 포틀릿 코드(430)는 사용자에게 의해 소비할 수 있는 콘텐츠와 같은 포틀릿 정보를 생성한다. 포틀릿 정보는 소비할 수 있는 콘텐츠가 어떻게 디스플레이될 것인지를 나타내는 프리젠테이션 정보를 선택적으로 포함할 수 있다. 예를 들어, XML(extended markup language)에 특정된 데이터와 같은 마크업 데이터는 소비할 수 있는 콘텐츠를 나타내는 메타 데이터를 포함한다.
- [0050] 포탈 응용 프로그램 서버(420)는 사용자에게 의해 초기 뷰를 위해 클라이언트에게 전달되는 각 포틀릿 뷰를 분석한다. 이 실시예에서 이 분석 동안에, 포탈 응용 프로그램 서버(420)는 도 3A 및 도 3B에 관련하여 설명한 바와 같이, 포탈 링크 엔트리를 포함하는 포탈 링크 구조를 생성한다. 포탈 링크 구조는 또한 포탈 링크 구조에 참조된 포탈 정보를 나타내는 메타 데이터를 포함할 수 있다. 본 발명의 바람직한 실시예에서, 포탈 링크 구조를 형성하고 관리하는 데 있어서 분석의 제어를 허용할 수 있도록 프레임워크(470)가 제공된다. 예를 들어, 프레임워크(470)는 분석의 너비 및 포탈 링크 구조가 구성되는 범위를 조절하는 응용 프로그램 유틸리티 함수를 제공한다. 실행될 때, 프레임워크(470)는 포틀릿 정보의 캐싱 및 프리페칭을 유리하게 조정할 것이다.
- [0051] 포탈 응용 프로그램 서버(420)는 도시된 바와 같이 응답(422)에서 포틀릿 어셈블리(445), 포틀릿 스타일(450) 및 초기 포틀릿 뷰(405, 410, 415)를 다운로드한다. 포틀릿 스타일(450)은 클라이언트 측에 다운로드된 포틀릿

뷰의 스타일 시트나 스킨의 설계 또는 수집을 포함한다. 포틀릿 뷰와 함께, 포탈 링크 구조는 또한 클라이언트 측에 메타 데이터의 형태로 선택적으로 다운로드될 수 있다. 본 발명의 일부 실시예에서, 포탈 링크 구조의 분석 및 초기 생성은 초기 포틀릿 뷰가 다운로드된 후 클라이언트 측에서 수행된다. 어떠한 경우에도 포탈 링크 구조는 클라이언트 측의 메모리에 저장된다. 메모리는 RAM, 디스크, 테이프 또는 임의의 다른 적합한 저장 매체를 포함한다.

[0052] 또한, 프레임워크(470)는 또한 다운로드되는 응용 프로그램 코드(465)의 유틸리티를 제공한다. 그러한 응용 프로그램 코드는 프레임워크 내부의 함수를 이용하고, 포탈 링크 구조를 형성할 범위를 판단하는 알고리즘을 포함할 수 있다. 응용 프로그램 코드는 포탈의 사용자가 정보를 요청하기 전에 어느 포틀릿 뷰가, 얼마나 많은 포틀릿 정보가 패치(fetch)되는지, 그리고 패치된 정보가 소비할 수 있는 콘텐츠, 메타 데이터 또는 둘 다를 포함하는지를 판단하는 알고리즘을 포함할 수 있다. “프리패치”의 용어는 사용자가 그러한 포틀릿 정보를 요청하기 전에 포틀릿 정보를 패치하는 프로세스를 나타낸다. 바람직한 실시예에서, 응용 프로그램 코드(465)는 예를 들어 Javascript<sup>®</sup> 언어를 사용하여 쓰여지고, 포탈 아키텍처 내부에 맞추어진다. Javascript<sup>®</sup>의 응용 프로그램 코드(465) 개발은 유리하게 클라이언트에 구성된 Javascript<sup>®</sup> 플러그-인을 갖는 웹 브라우저가 아니라 임의의 추가적인 소프트웨어 없이, 서버에 저장된 코드가 클라이언트에 실행되는 것을 허용한다. 응용 프로그램 코드가 프레임워크의 함수로의 접근을 갖는 한, 다른 프로그래밍 언어가 응용 프로그램 코드(465)를 개발하도록 사용될 수 있음을 당업자에 의해 인지되어야 한다.

[0053] 포틀릿 어셈블러(445)는 클라이언트 측에서 호출되고, 클라이언트의 포틀릿 뷰(405)로서 포탈 뷰(401)에 포틀릿 정보를 포함하는 포틀릿 뷰(455)를 제공한다. 응용 프로그램 코드(465)는 또한 클라이언트 측에서 호출되어, 포탈 링크 구조의 포틀릿 링크 엔트리에 의해 지정된 무슨 포탈 정보가 프리패치될 것인지를 판단한다. 이 판단이 수행되면, 응용 프로그램 코드(465)에 의한 요청에 응답하여 포틀릿 어셈블러(445)는 포탈 응용 프로그램 서버(420)에 프리패치 요청(419)을 개시한다. 프리패치 요청(419)은 적합하게 예를 들어 클라이언트가 프리패치하기 원하는 포탈 링크 구조 내 포함된 포틀릿 뷰의 참조를 포함할 수 있다. 본 발명의 일부 실시예에서, 프리패치 요청(419)은 포탈 링크 구조 내 다수의 포틀릿 참조들을 포함한다. 본 발명의 일부 실시예에서, 프리패치 요청(419)은 포틀릿 뷰에 대응하는 소비할 수 있는 데이터만, 포틀릿 뷰에 대응하는 메타 데이터만, 또는 포틀릿 뷰에 대응하는 소비할 수 있는 데이터와 메타 데이터의 둘 다의 조합을 패치하는지를 나타내는 파라미터를 포함한다. 포틀릿 뷰는 볼 수 있는 브라우저 스크린의 일부이므로, 이 패치 가능한 데이터의 모두는 전체 볼 수 있는 브라우저 스크린을 차지하는 전체 HTML 페이지보다 적다.

[0054] 프리패치 요청(419)에 응답하여, 포탈 응용 프로그램 서버(420)는 서버 측의 대응하는 포틀릿에 접근하고, 포틀릿으로부터 프리패치 요청에 따라 요청된 정보를 검색한다. 그 다음, 포탈 응용 프로그램 서버(420)는 포틀릿 어셈블러(445)에 요청된 정보를 다운로드한다. 포틀릿 어셈블러(445)는 캐시(460) 형태로 메모리 내에 다운로드된 정보를 저장한다. 캐시는 하드 디스크, 캐시 메모리, CD-ROMs, 광학 저장 장치 또는 자기 저장 장치를 포함한 임의의 적합한 컴퓨터 판독 가능한 매체를 포함할 수 있다. 포탈 사용자가 예를 들어 포틀릿 내부에 참조된 추가 정보를 요청하기 위해, 포탈 뷰 내부의 링크를 클릭하였을 때, 포틀릿 어셈블러(445)는 참조된 추가 정보에 관련된 포틀릿 링크 엔트리를 참조하고, 캐시된 콘텐츠에 링크 필드(362)가 유효 데이터를 포함하는지 체크한다. 그렇다면, 포틀릿 어셈블러는 유리하게 캐시(460)로부터 요청된 정보를 검색하고, 사용자가 그들의 캐시된 위치에 웹 사이트를 탐색한 것을 서버에 지시할 필요없이, 포틀릿 뷰에서 사용자에게 추가 정보를 제공한다. 그렇지 않으면, 포틀릿 어셈블러는 콘텐츠의 서버 ID(360) 필드를 이용하여 포탈 응용 프로그램 서버(420)로부터 추가 정보를 요청한다.

[0055] 응용 프로그램 코드(465)에 따라, 일반적으로 포틀릿 어셈블러(445)는 새롭게 제공되는 포틀릿 정보의 콘텐츠를 분석하여, 프리패치를 요청하는 임의의 참조들이 존재하는지를 판단할 것이다. 그렇다면, 프리패치 요청(419)은 상기 설명된 방법으로 만들어지고, 포탈 링크 구조는 제공 및 미제공된 포틀릿 뷰 간의 관계를 반영하기 위해 업데이트된다.

[0056] 포틀릿 뷰(405, 410, 415)보다 전에, 있다 해도 최소한의 콘텐츠를 갖는 골격 페이지가 초기에 다운로드될 수 있음을 당업자에 의해 인지되어야 한다. 본 발명의 이 양상에서, 프레임워크(470)는 예를 들어 백그라운드 HTTP 세션에 필요한 포틀릿 정보를 다운로드할 것이다. 그 다음, 프레임워크(470)는 골격 페이지와 다운로드된 응용 프로그램 코드(465)에 기반한 포탈 링크 구조를 구성할 것이다. 응용 프로그램 코드(465)는 포탈 링크 구조 메타 데이터에 접속하고 메타 데이터를 사용하여 포틀릿 뷰의 캐시를 프리패치하고 채우기 위해 프레임워크(470)를 사용할 것이다.

- [0057] 2003년 6월 19자에 제출된 미국 특허 출원 제10/464910호의 “METHOD, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR PORTLET AGGREGATION BY CLIENT APPLICATIONS ON A CLIENT SIDE OF A CLIENT/SERVER ENVIRONMENT” 에 보다 상세하게 기술된 바와 같이, 사용자에게 포틀릿 정보를 제공할 때, 포틀릿 어셈블리 코드 (445)는 서버 측으로부터 요청된 정보와 클라이언트 측에 저장된 정보를 집합한다.
- [0058] 도 5는 본 발명에 따른 프로세스(500)를 도시한 순서도이다. 도 5에 도시된 바와 같이, 단계 510에서 동작은 논리적으로 분리된 클라이언트 측/서버 측 컴퓨팅 환경의 서버 측에서 응용 프로그램 코드를 제공함으로써 시작할 수 있다. 이 응용 프로그램 코드는 일반적으로 JavaScript로 쓰여지고, 컴퓨팅 환경의 클라이언트 측에서 실행된다. 응용 프로그램 코드는 포탈 링크 구조의 크기 및 레벨 수와, 서버측으로부터 프리패치한 데이터의 크기 및 품질을 제어하는 프리패치 알고리즘을 포함한다. 포틀릿 링크 엔트리가 현재 포틀릿 뷰로 제공된다면, 응용 프로그램 코드는 포탈 링크 구조 내 포틀릿 링크 엔트리에 인접한 포틀릿 링크 엔트리를 대체하지 않도록 캐시 대체 수단을 특정할 수 있다. 응용 프로그램 코드는 프레임워크 코드를 이용한다. 프레임워크 코드는 분석의 너비, 포탈 링크 구조가 구성되는 범위를 제어하고, 포틀릿 정보를 프리패치 및 캐시하는 유틸리티 함수를 제공한다.
- [0059] 단계 520에서, 프로세스(500)는 포탈 링크 구조를 완성하기 위해 포틀릿 뷰 내부의 참조들을 분석한다. 이 분석은 컴퓨팅 환경의 서버 측에서, 클라이언트 측에서, 또는 둘 다에서 수행될 수 있다. 단계 525에서, 응용 프로그램 코드 및 포탈 링크 구조는 클라이언트 측에 다운로드된다. 단계 530에서, 클라이언트 측은 사용자가 정보를 요청하기 전에 포탈 링크 구조의 포틀릿 링크 엔트리 내 참조와 관련된 포틀릿 정보를 패치하는 응용 프로그램 코드를 호출한다. 단계 540에서, 프로세스(500)는 패치된 포탈 정보 내부의 참조를 반영하기 위해 포탈 링크 구조를 업데이트한다. 서버 측이 단계 520의 분석을 수행한다면, 포탈 링크 구조는 이 방법의 첫 번째 반복에 대한 유효 데이터를 포함할 것이다.
- [0060] 클라이언트가 단계 520을 수행하는 다른 방도의 실시예에서, 클라이언트는 비어있는 포탈 링크 구조를 수신하고, 단계 530 이전에 단계 540에서와 같이 그 속에 포탈 링크 엔트리를 배치할 것이다.
- [0061] 단계 550에서, 컴퓨팅 환경의 클라이언트 측에 접속하는 사용자의 요청시 프로세스(500)는 패치된 포탈 정보를 디스플레이한다. 단계 550 후에, 이후의 단계 530, 540 및 550의 반복을 위해 프로세스(500)는 단계 530으로 진행할 것이다. 사용자가 더 이상 이 포탈에 접속하기를 원하지 않을 때, 프로세스(500)는 종료된다.
- [0062] 따라서, 본 발명에 따른 실시예에서, 논리적으로 분리된 클라이언트 측/서버 측 컴퓨팅 환경의 서버 측에서보다 클라이언트 측에서 포틀릿 정보를 프리페칭 및 캐싱함으로써, 사용자에게 의해 보여지는 웹 응용 프로그램의 응답 시간은 개선된다.
- [0063] 당 기술 분야에서 통상의 기술을 갖는 자라면 본 발명의 기술적 사상 및 범위를 벗어나지 않고 본 발명의 개시에 따른 이점이 제공되도록 본 발명의 많은 대안 및 수정안을 형성할 수 있다. 따라서 도시된 실시예는 단지 예시의 목적으로 설명되고, 하기의 청구 범위에 의해 정의되는 본 발명을 한정하는 것으로 간주되어서는 안됨을 이해하여야 한다. 따라서 하기 청구 범위는 글자 그대로 설명된 요소의 조합 뿐만 아니라, 충분히 동일한 결과를 얻을 수 있는 동일한 방법으로 동일한 기능을 수행하는 모든 동등한 요소를 포함하는 것으로 인지되어야 한다. 그러므로 청구 범위는 상기 구체적으로 도시하고 기술한 것, 개념상 동등한 것 및 또한 본 발명의 요지를 내포하는 것을 포함하도록 이해되어야 한다.
- [0064] 본 발명을 전술한 실시예에 관련하여 설명하였다 하더라도, 당업자라면 본 발명이 이에 한정되지 않고 본 발명의 범위 내에서 다양한 변형 및 수정이 가능함을 이해할 것이다.
- [0065] 본 발명의 범위는 여기에 개시된 임의의 새로운 특징 또는 특징들의 조합을 포함한다. 그로 인해, 본 출원인은 이 출원 또는 이로부터 유도되는 임의의 출원의 진행 동안 새로운 청구 범위가 그러한 특징 또는 특징들의 조합에 공식화될 수 있음을 언급한다. 특히, 하기 첨부된 청구 범위에 관련하여, 단지 청구 범위에 열거된 특징 조합만이 아니라, 종속 청구항으로부터의 특징이 독립 청구항의 특징과 조합될 수 있고, 관련 독립 청구항으로부터의 특징이 임의의 적합한 방법으로 조합될 수 있다.
- [0066] 불명료함을 피하기 위해, 발명의 상세한 설명 및 청구 범위를 통해 여기에 사용된 “포함하다”의 용어는 “단지 ~으로 이루어지다”라는 의미로 해석되어서는 안된다.

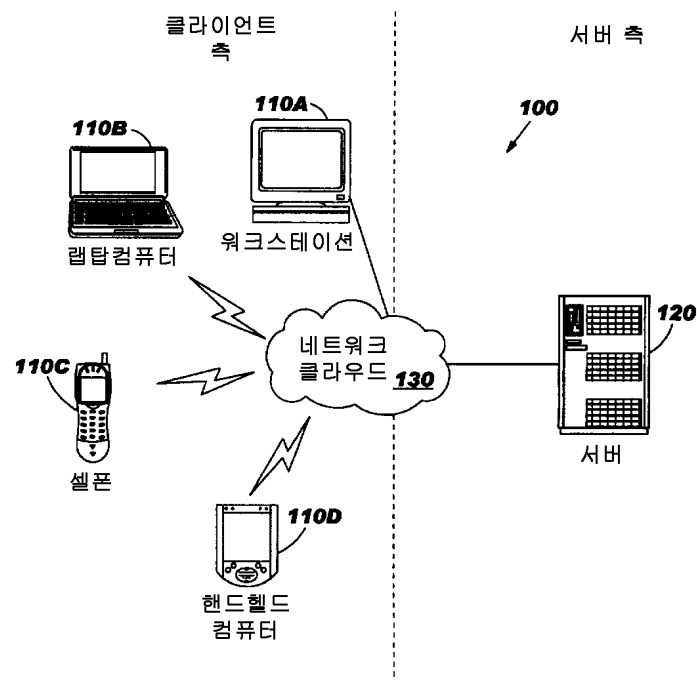


### 도면의 간단한 설명

- [0009] 하기 발명의 상세한 설명과 이에 따른 도면으로부터, 본 발명의 특징 및 효과에 더불어 본 발명을 보다 명백하게 이해할 수 있을 것이다.
- [0010] 도 1은 본 발명이 적합하게 구현될 수 있는 예시적인 시스템의 도면이다.
- [0011] 도 2는 본 발명에 따른 실시예를 이용하여 제공될 수 있는 예시적인 포탈 뷰의 도면이다.
- [0012] 도 3A는 포털릿 내 포함된 참조들을 도시한 계층도이다.
- [0013] 도 3B는 본 발명의 실시예에 따른 예시적인 포탈 링크 구조를 도시한 블록도이다.
- [0014] 도 3C는 본 발명의 실시예에 따른 도 3B의 포탈 링크 구조 내 예시적인 포털릿 링크 엔트리를 도시한 블록도이다.
- [0015] 도 4는 본 발명의 실시예에 따른 포탈 링크 구조에 관하여 클라이언트/서버 상호 작용을 도시한 블록도이다.
- [0016] 도 5는 본 발명의 실시예에 따른 프로세스를 도시한 순서도이다.

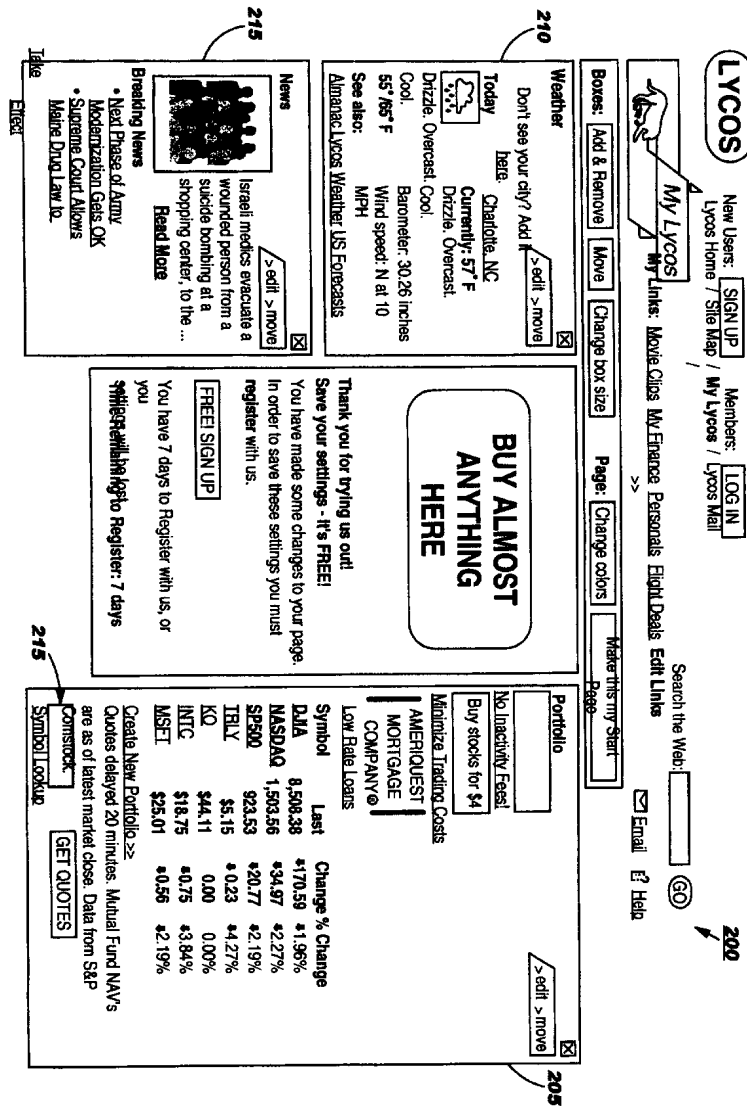
### 도면

#### 도면1

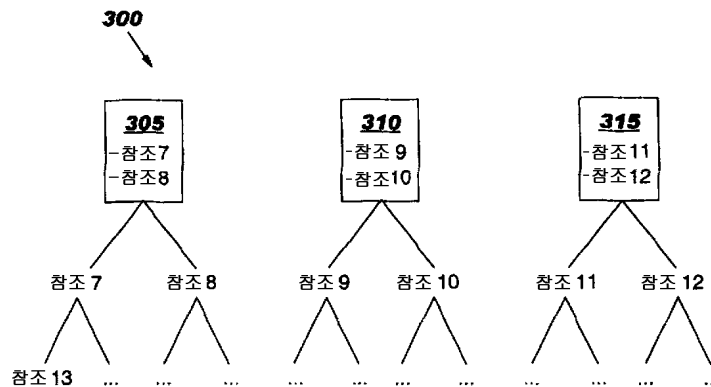




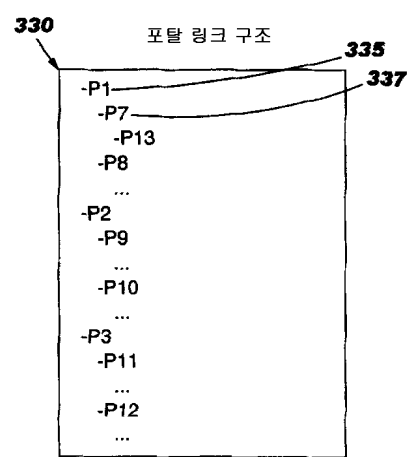
도면2



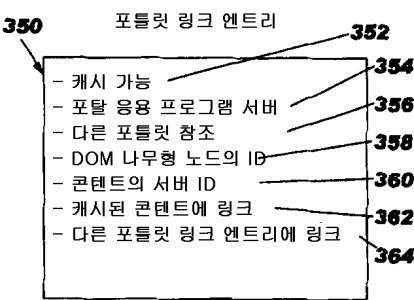
도면3a



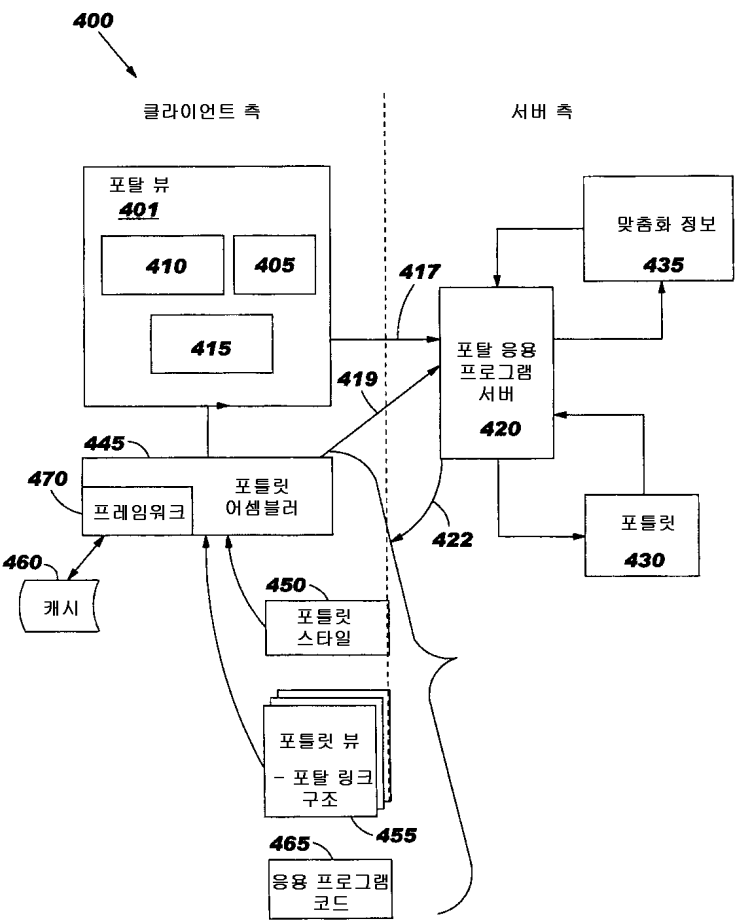
도면3b



도면3c



도면4



도면5

