(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년09월30일
(11) 등록번호 10-1661784
(24) 등록일자 2016년09월26일

(51) 국제특허분류(Int. Cl.)
G06F 17/24 (2006.01) *G06F 17/21* (2006.01)
G06Q 50/10 (2012.01)
(21) 출원번호 10-2011-7022370
(22) 출원일자(국제) 2010년03월19일
심사청구일자 2015년02월17일
(85) 번역문제출일자 2011년09월23일
(65) 공개번호 10-2012-0009434
(43) 공개일자 2012년01월31일
(86) 국제출원번호 PCT/US2010/028028
(87) 국제공개번호 WO 2010/111147
국제공개일자 2010년09월30일
(30) 우선권주장
12/410,470 2009년03월25일 미국(US)
(56) 선행기술조사문헌
KR1020060080132 A*
Popov.K et al., An efficient incremental marshaling framework for distributed systems, FGCS v.21 no.5 pp.717-724, 2005.
US20070245353 A1*
US20090070765 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
요시다 고타로
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 인터내셔널 페이턴츠 마이크로소프트 코포레이션
(74) 대리인
제일특허법인

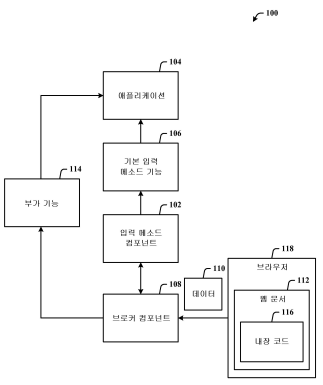
전체 청구항 수 : 총 15 항 심사관 : 경연정

(54) 발명의 명칭 **웹 브라우저를 통한 애플리케이션으로의 컨텐츠 입력 메소드**

(57) 요 약

개시된 아키텍처는 마크업 언어 페이지(예컨대, HTML)를 통해, 입력 메소드가 부착된 애플리케이션으로의 컨텐츠(예컨대, 텍스트)의 입력을 용이하게 할 수 있다. 이러한 방식으로, 플러그인 개발자는 웹 페이지 언어(예컨대, HTML)를 사용하여 임의의 UI 형태 및 동작을 구현할 수 있고, 통상의 HTML 페이지가 하듯이 백엔드 데이터소스와 통신할 수도 있으며, 적합한 기술을 통해 이를 구축할 수도 있다. 웹페이지는 웹 서버에 저장되고 온더 플라이로 클라이언트에 다운로드되기 때문에, 설치가 최소화되고 사용자가 자동으로 최신 버전을 사용할 수 있게 된다. 브로커 컴포넌트는 프로그래밍 언어로 작성되고 마크업 언어로 내장된 코드로부터 얻어지는 데이터를 수신하고, 애플리케이션에 임의의 부가 데이터를 제공한다.

대 표 도



명 세 서

청구범위

청구항 1

컴퓨터로 구현되는 컨텐츠 시스템으로서,

프로세서와,

상기 프로세서를 통해 동작가능한 입력 메소드 컴포넌트 - 상기 입력 메소드 컴포넌트는 기본 입력 메소드 기능(baseline input method functionality)을 적어도 하나의 애플리케이션에 제공함 - 와,

상기 프로세서를 통해 동작가능하고 상기 입력 메소드 컴포넌트로부터 웹 문서와 연관된 데이터를 수신하도록 구성된 제1 브로커 컴포넌트 - 상기 데이터는 상기 입력 메소드 컴포넌트에 관련된 제1 애플리케이션에 부가 기능을 추가하고, 상기 제1 브로커 컴포넌트에는 URL(uniform resource locator) 접두어(prefix)가 등록되고, 상기 URL 접두어는 상기 웹 문서 내에 있고, 상기 URL 접두어는 상기 제1 브로커 컴포넌트가 동작하는 운영 체제의 네이티브 URL 접두어 집합의 확장이고, 상기 운영 체제는 상기 URL 접두어를 등록하여 상기 네이티브 URL 접두어 집합을 확장할 수 있는 기능을 포함하고, 상기 URL 접두어를 등록하는 것은 상기 URL 접두어를 상기 운영 체제에 관한 제2 애플리케이션과 연관시키는 것을 포함함 - 와,

상기 프로세서를 통해 동작가능하고 상기 입력 메소드 컴포넌트로부터 웹 문서와 연관된 데이터를 수신하도록 구성된 제2 브로커 컴포넌트 - 상기 데이터는 상기 입력 메소드 컴포넌트에 관련된 제3 애플리케이션에 부가 기능을 추가함 -

를 포함하는 시스템.

청구항 2

제 1 항에 있어서,

상기 웹 문서는, 실행 시에 상기 데이터를 생성하는 내장 코드를 포함하는 웹 페이지이며, 상기 내장 코드는 프로그래밍 언어로 작성되고 마크업 언어로 내장되는 시스템.

청구항 3

제 1 항에 있어서,

상기 웹 문서는 웹 서버에 저장되고, 상기 웹 서버에서 업데이트되며, 상기 애플리케이션에 의한 업데이트 사용 요청 시 클라이언트 브라우저에 다운로드되고, 상기 웹 문서에 대한 업데이트는 상기 프로세서와 연관된 컴퓨팅 장치에 적용되는 업데이트와는 독립적인 것인 시스템.

청구항 4

제 1 항에 있어서,

상기 웹 문서는 웹 문서 컨트롤을 통해 상기 애플리케이션으로의 상기 데이터의 제어된 입력을 용이하게 하는 시스템.

청구항 5

제 1 항에 있어서,

상기 제1 브로커 컴포넌트는 URL(uniform resource locator) 프로토콜 핸들러를 사용하여 상기 웹 문서로부터 호출되는 시스템.

청구항 6

제 1 항에 있어서,

상기 입력 메소드 컴포넌트는, 상기 데이터가 라우팅될 애플리케이션 또는 관련 입력 메소드 컴포넌트를 상기 제1 브로커 컴포넌트에게 알려주는 식별자를, 통지를 통해 상기 제1 브로커 컴포넌트에 전송하는 시스템.

청구항 7

제 1 항에 있어서,

상기 제1 브로커 컴포넌트는 각각의 입력 메소드 컴포넌트에 연관된 다수의 식별자들을 수신하여 상기 식별자들을 등록하고, 마지막에 등록된 식별자의 입력 메소드 컴포넌트에 연관된 애플리케이션에 상기 데이터를 라우팅하는 시스템.

청구항 8

제 1 항에 있어서,

상기 부가 입력 메소드 기능은 텍스트 또는 복합 객체들 중 적어도 하나를 포함하는 시스템.

청구항 9

컴퓨터로 구현되는 컨텐츠 시스템으로서,

프로세서와,

상기 프로세서를 통해 동작가능하고 애플리케이션과 연관된 입력 메소드 컴포넌트 - 상기 입력 메소드 컴포넌트는 기본 입력 메소드 기능(baseline input method functionality)을 상기 애플리케이션에 제공하고, 상기 입력 메소드 컴포넌트는 일본어, 중국어 또는 아랍어 문자 중 적어도 하나를 사용하여 입력하는 것을 가능하게 함 - 와,

상기 프로세서를 통해 동작가능하고 상기 입력 메소드 컴포넌트와 연관된 브로커 컴포넌트 - 상기 브로커 컴포넌트는 상기 입력 메소드 컴포넌트로부터 통지를 수신하도록 구성되고, 상기 브로커 컴포넌트는 웹 페이지 내의 코드로부터 생성된 데이터의 형태로 상기 애플리케이션에 부가 기능을 제공하고, 상기 브로커 컴포넌트에는 URL(uniform resource locator) 접두어(prefix)가 등록되고, 상기 URL 접두어는 상기 웹 페이지 내에 있고, 상기 URL 접두어는 상기 브로커 컴포넌트가 동작하는 운영 체제의 네이티브 URL 접두어 집합의 확장이고, 상기 운영 체제는 상기 URL 접두어를 등록하여 상기 네이티브 URL 접두어 집합을 확장할 수 있는 기능을 포함하고, 상기 URL 접두어를 등록하는 것은 상기 URL 접두어를 상기 운영 체제에 관한 제2 애플리케이션과 연관시키는 것을 포함함 -

를 포함하는 시스템.

청구항 10

컴퓨터로 구현되는 컨텐츠 방법으로서,

복수의 입력 메소드 컴포넌트로부터 애플리케이션에 기본 입력 메소드 기능(baseline input method functionality)을 제공하는 단계와,

부가 애플리케이션 기능에 관련된 내장 코드를 포함하는 웹 페이지에 액세스하는 단계와,

상기 부가 애플리케이션 기능을 얻기 위해 상기 웹 페이지 내의 상기 내장 코드를 실행하는 단계 - 상기 웹 페이지는 URL(uniform resource locator) 접두어(prefix)를 포함하고, 상기 URL 접두어는 상기 애플리케이션이 동작하는 운영 체제의 네이티브 URL 접두어 집합의 확장이고, 상기 URL 접두어는 브로커에 등록되고, 상기 운영 체제는 상기 URL 접두어를 등록하여 상기 네이티브 URL 접두어 집합을 확장할 수 있는 기능을 포함하고, 상기 URL 접두어를 등록하는 것은 상기 URL 접두어를 상기 운영 체제에 관한 제2 애플리케이션과 연관시키는 것을 포함함 - 와,

상기 애플리케이션에 상기 부가 애플리케이션 기능을 추가하는 단계

를 포함하는 방법.

청구항 11

제 10 항에 있어서,

스크립트 언어로 상기 내장 코드를 작성하는 단계와, 마크업 언어로 상기 코드를 상기 웹 페이지에 내장하는 단계를 더 포함하는 방법.

청구항 12

제 10 항에 있어서,

후속 실행을 위해 상기 웹 페이지 내의 상기 내장 코드를 업데이트하는 단계와, 업데이트된 부가 애플리케이션 기능을 상기 업데이트된 내장 코드에 기반하여 상기 애플리케이션에 인가하는 단계를 더 포함하는 방법.

청구항 13

제 10 항에 있어서,

상기 브로커에게 상기 기본 입력 메소드 기능을 제공하는 상기 복수의 입력 메소드 컴포넌트 중 적어도 하나를 등록하는 단계와, 상기 부가 애플리케이션 기능을 상기 등록에 기반하여 상기 브로커를 통해 상기 애플리케이션으로 라우팅하는 단계를 더 포함하는 방법.

청구항 14

제 10 항에 있어서,

상기 웹 페이지와의 상호작용에 응답하여, 상기 부가 애플리케이션 기능을 상기 애플리케이션에 자동으로 추가하는 단계를 더 포함하는 방법.

청구항 15

제 10 항에 있어서,

추적 정보를 사용하여 다수의 애플리케이션들 중에서 상기 애플리케이션과의 상호작용을 추적하는 단계와,

상기 추적 정보를 통지를 통해 브로커에게 전송하는 단계와,

상기 추적 정보에 기반하여 상기 부가 애플리케이션 기능을 해당 애플리케이션에 추가하는 단계를 더 포함하는 방법.

발명의 설명

배 경 기 술

[0001] 입력 메소드가 써드 파티 컴포넌트들을 입력 시스템에 추가하여 특정 시나리오에서의 입력 경험을 확대할 수 있도록, 써드 파티 프로그램들은 플러그인 기능(plug-in capability)을 가진 입력 메소드들을 요구한다. 또한 이는 입력 메소드 코어 바이너리(input method core binaries)를 출시한 후, 서비스로서 보다 많은 입력 기능들을 추가하는 확장성을 제공할 수 있게 한다. 이러한 플러그인 컴포넌트들은 로컬 컴퓨터 또는 원격 서버 안에 있을 수 있는 백엔드 데이터소스(backend datasources)와의 통신을 위한 사용자 인터페이스(UI) 형태와 동작을 필요로 한다. 게다가, 이러한 플러그인 컴포넌트들은, 다운로드 될 수 있고 설치가 용이해야하며 새로운 버전이 출시되면 업데이트 되어야 한다.

발명의 내용

과제의 해결 수단

[0002] 다음은, 본원에 기술된 새로운 실시예들의 기본적인 이해를 돕기 위해서 간략화된 요약이 제공된다. 이러한 요약은 전반적인 개요가 아니며, 핵심/필수 구성요소를 밝히거나 그 범위를 한정하기 위한 것도 아니다. 이후에 제공되는 보다 상세한 설명에 대한 서문으로서, 몇몇 컨셉을 간략화된 형태로 제공하는 것이 그 유일한 목적이다.

[0003] 개시된 아키텍처는 마크업 언어 페이지(markup language pages, 예컨대, HTML)를 통해, 입력 메소드가 부착된 애플리케이션으로의 컨텐츠(예컨대, 텍스트)의 입력을 용이하게 할 수 있다. 이러한 방식으로, 플러그인 개발자는 웹 페이지 언어(예컨대, HTML)를 사용하여 임의의 사용자 인터페이스(UI) 형태 및 동작도 구현할 수 있고, 통상의 웹페이지가 적합한 기술을 통해 하듯이 백엔드 데이터소스와 통신할 수 있다. 웹페이지는 웹 서버에 저장되고 온더 플라이(on-the-fly)로 클라이언트 브라우저에 다운로드되기 때문에, 설치가 최소화되고 사용자가 자동으로 최신 버전을 사용할 수 있게 된다.

[0004] 일 구현예로, 웹 페이지의 HTML에 내장된 코드로부터 인스트럭션을 수신하는 브로커 컴포넌트(broker component)가 구현된다. 클라이언트 브라우저(또는 웹 페이지를 처리하는 다른 적절한 애플리케이션)는 웹 페이지를 수신하고, HTML에 내장된 코드(예컨대, 자바스크립트와 같은 스크립트)를 실행한다. 따라서, 브로커 컴포넌트는 URL 프로토콜 핸들러 메커니즘(URL protocol handler mechanism)에 의해 HTML 페이지의 코드로부터 호출될 수 있다. 나아가, 브로커 컴포넌트는 클라이언트 시스템의 일부로 기본적으로 설치될 수도 있다.

[0005] 예를 들어, 플러그인 개발자가 소프트웨어적인 키보드 기능을 입력 메소드 및/또는 관련 애플리케이션에 추가하기 위해 입력 메소드 플러그인 컴포넌트를 구현하기를 원한다고 가정하자. 개발자는 'a'부터 'z'까지의 키 버튼을 보여주는 코드를 가진 HTML 페이지를 작성할 수 있다. 사용자가 HTML 페이지를 열고 페이지 상의 버튼을 선택할 때, 클라이언트 컴퓨터에서 실행되고 있는 애플리케이션으로 해당 문자가 입력된다. HTML 페이지이기 때문에, 개발자들은 원하는 시각적 외향에 따라 임의로 버튼을 배치할 수 있다. 그 후에, 플러그인에 보다 많은 기능들을 추가하고자 할 때(예컨대, 문자가 아닌 단어를 입력하는 버튼을 더 추가), 개발자는 단순히 웹 서버에 있는 HTML 페이지를 업데이트하면 된다. 사용자는, 예를 들어, 클라이언트 브라우저를 통해 자동으로 새로운 버전(업데이트)을 수신하게 된다.

[0006] 상기 및 관련 목적을 달성하기 위해, 예시적인 양태들이 다음의 설명 및 첨부 도면과 관련하여 본원에서 설명된다. 이러한 양태들은 본원에 개시된 원리를 실시할 수 있는 다양한 방식을 나타내고 있으며, 이 모든 양태 및 그 등가물은 특허청구된 대상의 범위 내에 속하는 것으로 보아야 한다. 그 외의 효과 및 새로운 특징들은 도면과 함께 살펴볼 때, 다음의 상세한 설명으로부터 명백해질 것이다.

도면의 간단한 설명

[0007] 도 1은 개시된 아키텍처에 따른, 컴퓨터로 구현되는 컨텐츠 시스템을 도시한다.

도 2는 웹 문서를 통해 컨텐츠 입력을 용이하게 하는 컴포넌트들과 엔티티들 간의 통신의 흐름도를 도시한다.

도 3은 애플리케이션에 부가 기능을 추가하기 위한 제 1 클라이언트 배치를 도시한다.

도 4는 애플리케이션에 부가 기능을 추가하기 위한 제 2 클라이언트 배치를 도시한다.

도 5는 브라우저를 통해 애플리케이션으로 컨텐츠를 입력하는 컨텐츠 메소드를 도시한다.

도 6은 도 5의 메소드의 추가적인 양태들을 도시한다.

도 7은 도 5의 메소드의 추가적인 양태들을 도시한다.

도 8은 개시된 아키텍처에 따른, 브라우저를 통해 애플리케이션으로의 컨텐츠 입력을 용이하게 하는 컴퓨팅 시스템의 블록 다이어그램을 도시한다.

도 9는 브라우저를 통해 애플리케이션으로의 컨텐츠 입력을 용이하게 하는 컴퓨팅 환경의 개략적인 블록 다이어그램을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0008] 개시된 아키텍처는 마크업 언어 페이지(예컨대, HTML)를 통해, 입력 메소드가 부착된 애플리케이션으로의 컨텐츠(예컨대, 텍스트 문자열, 객체, 오디오, 비디오 등)의 입력을 용이하게 할 수 있다. 이러한 방식으로, 플러그인 개발자는 웹 페이지 언어(예컨대, HTML)를 사용하여 임의의 사용자 인터페이스(UI) 형태 및 동작을 구현할 수 있고, 통상의 HTML 페이지가 하듯이 백엔드 데이터소스와 통신할 수도 있으며, 적합한 기술을 통해 이를 구축할 수도 있다. 웹페이지는 웹 서버에 저장되고 온더 게플라이로 클라이언트에 다운로드되기 때문에, 설치가 최소화되고 사용자가 용이하 그리고 자동으로 최신 버전을 사용할 수 있게 된다.

[0009] 브로커 컴포넌트는 클라이언트 상에 구현되며, 웹 페이지의 마크업 언어로 내장된 코드의 실행 결과인 데이터를 수신한다. 부가적인 브로커 컴포넌트를 클라이언트 시스템에 추가함으로써, 해당 클라이언트 애플리케이션을 위해 보다 많은 데이터 유형이 클라이언트 시스템에서 처리될 수 있다.

[0010] 이제부터 도면을 참조하며, 도면 전체에 걸쳐 유사한 참조 부호가 유사한 구성 요소를 참조하는 데 사용된다. 다음의 설명에서는, 설명의 목적으로, 본 발명에 대한 전반적인 이해를 돕기 위해 상세한 많은 설명이 제공된다. 그러나, 이러한 상세한 설명 없이도 새로운 실시예가 실행될 수 있음은 자명하다. 다른 경우들에는, 본 발명의 설명을 용이하게 하기 위해 공지된 구조 및 장치들이 블록 다이어그램의 형태로 도시된다. 이는, 특허청구된 대상의 사상과 범위에 속하는 모든 변경, 등가물 및 대체물을 포함하기 위한 것이다.

[0011] 도 1은 개시된 아키텍처에 따른, 컴퓨터로 구현되는 컨텐츠 시스템(100)을 도시한다. 시스템(100)은, 애플리케이션(104)에 부착되어 기본 입력 메소드 기능(106)을 이 애플리케이션(104)에 제공하는 입력 메소드 컴포넌트(102)와, 웹 문서의 데이터(110)를 수신하여 결과적으로 데이터(110)에 의해 제공되는 부가 기능(114, 예컨대, 입력 메소드 기능)을 추가하는 브로커 컴포넌트(108)를 포함하며, 이 부가 기능(114)은 웹 문서(112)에 내장된 코드(116)의 처리를 통해 간접적으로 애플리케이션(104)에 제공된다.

[0012] 웹 문서(112)는, 실행 시에 데이터(10)를 생성하는 내장 코드(116)를 포함하는 웹 페이지일 수 있다. 내장 코드(116)는 적절한 어떠한 프로그래밍 언어(예컨대, 스크립트 언어(scripting language))로도 작성될 수 있고, HTML을 비롯한 마크업 언어로 내장될 수 있다. 웹 문서(112)는 웹 서버에 저장되고 클라이언트 브라우저(118)에 다운로드됨으로써, 클라이언트 브라우저(118)가 데이터(110)와 결국, 부가 입력 메소드 기능(114)을 생성하도록 내장 코드(116)를 처리하게 할 수 있다.

[0013] 일반적으로, 웹 문서(112)는, 웹 문서(112)와 사용자의 상호작용에 기반하여 애플리케이션(104)을 위한 (데이터(110)로서) 부가 기능(114)을 자동으로 보충함을 용이하게 할 수 있다. 이후에, 부가 기능(114)의 생성물(예컨대, 텍스트, 이미지 등)이 애플리케이션(104)에 제공된다. 브로커 컴포넌트(108)는 URL(uniform resource locator) 프로토콜 핸들러 메커니즘을 사용하여 웹 문서(112)로부터 호출될 수 있다. 부가 기능(114)은 텍스트, 또는 복합 객체, 예를 들어, 이미지, 오디오 및/또는 비디오 중에서 적어도 하나를 포함한다.

[0014] 다른 방식을 고려하면, 컴퓨터로 구현되는 컨텐츠 시스템(100)이, 애플리케이션(104)에 연계되어 기본 입력 메소드 기능(106)을 이 애플리케이션(104)에 제공하는 입력 메소드 컴포넌트(102)와, 입력 메소드 컴포넌트(102)에 연계되어 이 입력 메소드 컴포넌트(102)로부터 통지를 수신하는 브로커 컴포넌트(108)를 포함하며, 브로커 컴포넌트(108)는 애플리케이션(104)에 웹 페이지(예컨대, 웹 문서(112))의 코드(116)로부터 생성되는 데이터의 형태로 추가 기능(114)을 제공한다.

[0015] 브로커 컴포넌트(108)는 웹 페이지의 코드를 처리하는 브라우저로부터 데이터(110)를 수신한다. 브라우저(118)

에서 브로커 컴포넌트(108)를 호출하는 웹 페이지의 코드(116)를 실행할 때, 데이터(110)로부터 부가 기능(114)이 생성된다. 브로커 컴포넌트(108)는 입력 메소드 컴포넌트(102)에 연계된 식별자를 수신한다. 브로커 컴포넌트(108)는 식별자를 등록하고, 등록된 식별자에 따라 데이터(110)를 라우팅한다. 브로커 컴포넌트(108)는 특정 데이터 유형의 데이터(110)의 사용을 용이하게 한다. 또한, 브로커 컴포넌트(108)는 데이터(110)를 문자, 단어, 구 또는 복합 객체 중 적어도 하나의 형태로 애플리케이션(104)에 입력가능하게 하며, 이 애플리케이션(104)은 언어 관련 애플리케이션(linguistics application)이다.

[0016] 도 2는 웹 문서를 통해 컨텐츠 입력을 용이하게 하는 컴포넌트들과 엔티티들 간의 통신의 흐름도를 도시한다. 애플리케이션(104)은 클라이언트 컴퓨터에서 실행되는 프로그램일 수 있으며, 입력 메소드 컴포넌트(102)가 애플리케이션(104)에 부착되어 있다. 브로커 컴포넌트(108)는 클라이언트 컴퓨터 상에서 설치되고 실행되는 네이티브 코드일 수 있다. 본 예에서는, 웹 문서가 플러그인 개발자에 의해 제작될 수 있는 HTML 페이지(202)임이 다이어그램(200)에 도시된다. HTML 페이지(202)는 스크립트 언어로 작성된 도 1의 내장 코드(116)를 포함하여, 예컨대, 브라우저가 이 코드(116)를 실행하면, 그 결과인 데이터는, 입력 메소드 컴포넌트(102)에 의해 애플리케이션(104)에 제공되는 부가 입력 메소드 기능을 형성하게 된다. 다른 기술들을 사용하여 HTML에 내장된 프로그램 코드를 작성할 수 있음을 알 것이다.

[0017] 플러그인 개발자는 브로커 컴포넌트(108)를 개발하여, 사용자가 로컬 클라이언트 컴퓨터에 이 브로커 컴포넌트(108)를 설치하게 할 수 있다. 입력 메소드 컴포넌트(102)에서 특정 이벤트가 발생하면, 브로커 컴포넌트(108)가 입력 메소드 컴포넌트(102)로부터 통지(204)를 수신한다. 또한, 브로커 컴포넌트(108)는 URL 프로토콜 핸들러(206)에 의해 HTML 페이지(202)로부터 호출될 수 있다.

[0018] 데이타는 웹 브라우저의 웹 문서로부터 브로커 컴포넌트, 입력 메소드 컴포넌트, 마지막으로 애플리케이션으로 한쪽 방향으로 흐를 수 있다. 브로커 컴포넌트는 어떤 입력 메소드 컴포넌트 인스턴스가 데이터를 전송하는 지 식별자(티켓 또는 토큰이라고도 함)를 사용하여 확인한다. 브로커 컴포넌트는 입력 메소드 컴포넌트의 식별자들을 미리 알고 있다. 입력 메소드 컴포넌트로부터 브로커 컴포넌트로 식별자가 전송된다.

[0019] 일 구현예로, 애플리케이션이 사용자 상호작용으로부터 포커스를 얻을 때마다(또는 애플리케이션 윈도우가 포어그라운드로 이동), 그 애플리케이션에 연계된 입력 메소드 컴포넌트가 그 식별자(이벤트 통지라고도 함)를 브로커 컴포넌트로 전송한다. 임의의 애플리케이션에 부착된 각 입력 메소드 컴포넌트는 그 식별자를 브로커 컴포넌트로 언제든지 전송할 수 있다. 브로커 컴포넌트는 최종으로 등록된 식별자를 기억하고 있다.

[0020] 언제든지(입력 메소드 컴포넌트로부터 브로커 컴포넌트에 전송된 이벤트와 동기화되지 않을 때), 브로커 컴포넌트가 웹 문서로부터 호출될 수 있다. 호출되면, 브로커 컴포넌트는 최종 등록된 식별자를 사용하여 연계된 입력 메소드 컴포넌트 인스턴스의 위치를 파악한다.

[0021] 즐겨찾는 URL 리스트에서의 선택에 대한 응답, 입력 메소드 컴포넌트는 열린 특정 웹 문서를 가지고 웹 브라우저를 시작을 비롯한 가능한 모든 방식에 의해, 웹 문서는 웹 브라우저에 미리 열려져 있다. 일단 웹 문서가 열리면, 웹 문서는 모든 입력 메소드 컴포넌트 인스턴스들을 제공한다. 따라서, 입력 메소드 컴포넌트 인스턴스와 웹 문서는 1 대 1 관계일 필요가 없다. 예를 들어, 데스크탑에 도시된 소프트웨어적인 키보드의 단일 인스턴스가 원하는 어떤 애플리케이션과도 또는 다수의 애플리케이션들과도 동작하도록 사용될 수 있다.

[0022] 동작 시에, 특정 이벤트가 발생하면, 입력 메소드 컴포넌트(102)가 브로커 컴포넌트(108)에게 통지(204, 식별자)를 전송한다. 브로커 컴포넌트(108)는 이러한 이벤트를 트리거로서 갖는 HTML 페이지(202)를 호출하거나 호출하지 않을 수 있다. 이 이벤트들은 "티켓" 정보를 포함한다. 이러한 티켓 정보는 애플리케이션(104)의 핸들이거나, 입력 메소드 컴포넌트(102)의 COM(component object model) 인터페이스 포인터이거나, 또는 브로커 컴포넌트(108)가 추후에 적절한 애플리케이션(애플리케이션(104)) 및/또는 입력 메소드를 관리하기 위해 사용할 수 있는 다른 메커니즘들일 수 있다. 즉, 클라이언트 컴퓨터는 다수의 상이한 입력 메소드 컴포넌트들 및 호환 브로커 컴포넌트들을 특수 목적을 위해 사용할 수 있다. 본원의 실시예에서는, 다수의 브로커 컴포넌트들 또는 각 클라이언트 애플리케이션에 대해 적합한 상이한 브로커를 가진 단일 입력 메소드 컴포넌트의 시나리오에 대해서도 설명하고 있다.

[0023] HTML 페이지(202)는 웹 브라우저에서 열리고, 이 브라우저에 의해 렌더링될 수 있다. 개발자는 원하는 어떤 HTML 페이지도 작성할 수 있다. 등록된 URL 프로토콜 핸들러(206)가 적절한 파라미터들을 가진 HTML 페이지(202)로부터 호출된다. 이에 대해서, 브로커 컴포넌트(108)가 시작되고, HTML 페이지(202)로부터 이 파라미터들을 수신한다. 브로커 컴포넌트(108)가 통지(204)와 함께 전송된 티켓 정보에 대해 알고 있으므로, 이 브로커 컴

포넌트(108)는 애플리케이션(104) 및 입력 메소드 컴포넌트(102)와 통신하여, HTML 페이지(202)의 데이터로부터 기인한 부가 입력 메소드 기능(208)을 통지(240)에서 수신된 티켓 데이터에 기반하여 추가한다.

[0024] 일단 브로커 컴포넌트(108)가 클라이언트 컴퓨터에 설치되면, 클라이언트 컴퓨터에 적용되는 업데이트 없이도, 개발자가 웹 서버에 저장된 HTML 페이지(202)에 내장된 코드를 변경하거나 개선시킴으로써 입력 메소드 컴포넌트 기능을 변경하거나 개선할 수 있다. 이로써, 부가 기능을 자동으로 업데이트하고 배치하는 편리하고 효율적인 아키텍처가 제공된다.

[0025] 애플리케이션(104)에 제공된 컨텐츠가 문자 형태의 컨텐츠로 한정되지는 않음을 알 것이다. 가장 간단한 브로커가 있는 가장 단순한 형태에서는, 컨텐츠가 텍스트 문자열일 수 있다. 보다 복잡한 구현에서는, 브로커 컴포넌트(108)가 HTML 페이지(202)에서 호출될 때, 이 브로커 컴포넌트(108)가 특정 애플리케이션에 메타데이터를 전송하고, 이 메타데이터를 수신한 애플리케이션(104)이 복잡한 객체들(예컨대, 이미지)을 자신에게 입력할 수 있도록 설계될 수 있다. 결국, 애플리케이션(104)과 브로커 컴포넌트(108) 간의 "계약"에 기반한다. 앞서 언급한 대로, 다수의 상이한 브로커 컴포넌트들이 특수한 목적으로 사용될 수 있다. 따라서, 여러 가지 다른 데이터 유형들이 사용자 시스템에 추가될 수 있다.

[0026] HTML 및 웹 브라우징 기술은 이미 서버 측에서 사용되고 있다. 웹 서버에서 HTML의 스크립트를 사용하고, 사용자가 웹 페이지를 브라우징할 때, HTML 페이지가 서버에서 클라이언트 브라우저로 다운로드된다. 클라이언트 브라우저는 내장된 스크립트 코드를 실행하여 데이터를 생성한다. 따라서, 사용자는 클라이언트를 설치할 필요가 없고, 단지 서버의 HTML 페이지를 열면 된다. 브로커 컴포넌트는 네이티브 코드로서 클라이언트에 설치될 수 있다. 클라이언트 브라우저에 이미 다운로드된 웹 페이지 문서 안의 프로그램 코드는, 클라이언트가 서버로부터 업데이트된 해당 웹 페이지 문서를 다운로드할 때마다 자동으로 업데이트된다.

[0027] 브로커 컴포넌트는 데이터 유형을 정의한다. 데이터 유형이 정의되고 연계된 브로커가 설치되면, 일반적으로 브로커는 자주 업데이트될 필요가 없게 된다. 그러나, 그 데이터 유형의 데이터는 다양할 수 있고 그 데이터 유형에 대한 데이터의 (웹 페이지에서 코딩된) 생성기는 동적으로 변경될 수 있다. 따라서, 데이터의 생성기는 예컨대, 서버 측에 저장되어 있는 마크업 언어(예컨대, HTML) 및 스크립트 언어(예컨대, 자바 스크립트)에 의해 서술되므로, 이 생성기가 손쉽게 변경될 수 있다. 앞서 언급한 바대로, 마크업 언어는 HTML로 제한되지 않으며, 브라우저에서 렌더링될 수 있는 적절한 기술이면 된다.

[0028] 이용 가능한 운영 체제의 기능으로 URL 프로토콜 핸들러가 있다. 이 기술은 URL을 실행 파일("exe")에 매핑할 수 있다. 예를 들어, URL은 "http:", "mailto:", 또는 "ftp:" 등으로 시작되는 포맷 문자열을 가질 수 있다. 운영 체제는 이러한 문자열을 넘기고, 이 문자열의 접두어(prefix)를 검사하고, 등록된 실행가능 애플리케이션으로 제어를 넘기는 메커니즘을 포함할 수 있다. 예를 들어, URL이 "http:"로 시작되면, 운영 체제는 브라우저 실행파일을 실행시킬 것이다. 또 다른 예를 들면, URL 문자열이 "mailto:"로 시작되면, 운영 체제는 메시징 프로그램(예컨대, 이메일)을 실행시킨다.

[0029] 운영 체제는 브로커 컴포넌트가 이러한 기능을 확대하여 임의의 접두어 문자열이 임의의 실행 파일과 관련하여 등록 및 정의될 수 있도록 하는 기능을 더 포함할 수 있다. 이러한 메커니즘이 사용되어 브로커 컴포넌트에 원하는 접두어 문자열을 등록하게 된다.

[0030] HTML 페이지에서, 사용자가 이런 접두어를 가진 URL을 열면, 운영 체제는 등록된 브로커 실행 파일을 실행한다. 브로커 컴포넌트가 호출될 때, 이 브로커 컴포넌트는 명령어 라인 파라미터를 사용하여 그 접두어로 시작되는 전체 URL 문자열을 수신한다. 예를 들어, HTML 페이지가 "<prefix>:xyz"를 포함하고, 여기서 <prefix>가 "http", "ftp", "mailto" 등인 경우, 브로커 컴포넌트는 명령어 라인 파라미터로 "<prefix>:xyz"의 문자열을 수신한다. 각 HTML 페이지가 스트링 "xyz"를 브로커 컴포넌트로 넘기기를 원한다면, HTML 페이지는 "<prefix>:xyz"를 포함하게 된다. 이후에, 브로커 컴포넌트가 실행되어 문자열 "<prefix>:xyz"를 수신하여, 공지된 "<prefix>" 부분을 제거하여 문자열 "xyz"를 추출할 수 있게 된다.

[0031] 브로커 컴포넌트가 문자열 "xyz"를 수신할 때, 브로커 컴포넌트 및 애플리케이션은 문자열을 송수신하기 위한 프로토콜을 사용한다. 사용가능한 아주 기본적인 프로토콜로 하나의 윈도우에서 다른 윈도우로 문자를 전송하기 위한 (Windows™ 운영 체제를 위한) WM_CHAR를 들 수 있다. 브로커 컴포넌트가 WM_CHAR 메시지(예컨대, 파라미터 x를 가진 WM_CHAR, 파라미터 y를 가진 WM_CHAR, 파라미터 z를 가진 WM_CHAR 순)를 애플리케이션에 전송하면, 이 애플리케이션은 키보드 입력 "x", 다음에 "y", 다음에 "z"를 수신한다. 따라서, 모든 애플리케이션들이 이러한 메시지를 수신 시 처리하여, 애플리케이션이 처리하고 있는 문서로 "xyz"를 입력할 수 있다. 이것이 기본적

인 브로커 모델이다. 원한다면, 많은 다른 브로커 컴포넌트들이 다른 프로토콜로 구현되고 등록될 수 있다.

[0032] 고급 브로커 모델에서는, 사진, 비디오, 이미지, 사운드 등이 애플리케이션에 전송될 수 있다. 이러한 경우, 사용자 애플리케이션 및 브로커는 최적화된 운영 동작을 위한 사전 정의된 사설 프로토콜을 사용할 수 있다. 보다 구체적인 구현예로, 특정 클라이언트 애플리케이션을 위한 커스텀 브로커(custom broker)를 개발할 수도 있다.

[0033] 클라이언트 기기에는 많은 애플리케이션들이 있으며, 각 애플리케이션은 부착된 입력 메소드 컴포넌트와 함께 동작하지만, 브로커 컴포넌트는 단일 인스턴스이다. 입력 메소드 컴포넌트는 브로커 컴포넌트에게 브로커 컴포넌트가 메시지를 전송할 애플리케이션을 알려준다. 사용자가 많은 이용가능한 애플리케이션들 중 하나에서 키를 누르고 있으면, 입력 메소드 컴포넌트는 사용자가 상호작용하고 있던 애플리케이션을 기억하여, 브로커 컴포넌트에 그 애플리케이션을 위한 운영 체제 핸들을 전송한다. 그 후에, 브로커 컴포넌트는 그 핸들을 사용하여 메시지를 전송한다. 이는 단지 일례일 뿐, 다른 방법들을 사용하여 메시지가 전송될 애플리케이션을 식별할 수 있다.

[0034] 개시된 아키텍처는 키보드에 일반적으로 표시되지 않는 문자들을 사용하는 언어들(예컨대, 일본어, 중국어, 아랍어, 라틴어 등)에 대한 특별 적용 가능성을 가진다. 또 다른 일 구현예로, 키보드를 통해서는 불가능한, 입력 장치(예컨대, 마우스, 키패드 등)에 의한 한 번의 선택으로 단어 및/또는 구를 입력하는 데에 개시된 아키텍처가 사용될 수 있다.

[0035] 도 3은 애플리케이션에 부가 기능을 추가하기 위한 제 1 클라이언트 배치(300)를 도시한다. 예를 들어, 제 1 클라이언트 배치(302)에서, 제 1 입력 메소드 컴포넌트(304)는, 제 1 입력 메소드 컴포넌트(304)가 제공하는 기본 입력 메소드 기능의 이용을 위해서 제 1 애플리케이션(306)과 함께 사용된다. 임의의 시점에서, 제 1 입력 메소드 컴포넌트(304)는 제 1 브로커 컴포넌트(308)로의 통지를 통해서 제 1 브로커 컴포넌트(308)에 식별자를 등록한다. 개시되면, 제 1 웹 페이지(310)가 제 1 브로커 컴포넌트(308)를 호출한다. 제 1 웹 페이지(310)는 프로그래밍 언어(예컨대, 자바스크립트)로 작성되고 HTML과 같은 마크업 언어로 내장된 내장 코드(312)를 포함한다. 또는, 다른 적절한 웹 문서 언어들이 사용될 수도 있다.

[0036] 클라이언트 브라우저는 다양한 목적으로 웹 페이지를 저장하는 네트워크 서버(도시되지 않음)로부터 제 1 웹 페이지(310)를 다운로드한다. 제 1 웹 페이지(310)는 클라이언트 브라우저에서 열리고, 클라이언트 브라우저는 내장 코드(312)를 실행한다. 코드(312)의 실행으로 프로토콜 핸들러를 통해 제 1 브로커 컴포넌트(308)로 전송되는 데이터가 출력된다. 이후에, 제 1 브로커 컴포넌트(308)는 제 1 애플리케이션(306)에서의 입력과 사용을 위한 제 1 부가 입력 메소드 기능(310)을 추가하기 위한 제 1 애플리케이션(306)으로 데이터를 라우팅한다.

[0037] 도 4는 애플리케이션에 부가 기능을 추가하기 위한 제 2 클라이언트 배치(400)를 도시한다. 여기서는, 단일 브로커 컴포넌트(402)가 클라이언트 기기에 있는 두 애플리케이션들과 대응되는 두 입력 메소드 컴포넌트들에 대한 데이터 유형들을 지원한다. 제 2 입력 메소드 컴포넌트(404)는 제 2 애플리케이션(406)에 연계(예컨대, 부착)되어 있고, 제 3 입력 메소드 컴포넌트(408)는 제 3 애플리케이션(410)에 연계(예컨대, 부착)되어 있다. 제 2 입력 메소드 컴포넌트(404) 및 제 3 입력 메소드 컴포넌트(408)는 브로커 컴포넌트(402)에 등록되어 있고(통지 전송), 마지막에 등록되는 것이 (예컨대, 사용자 상호작용을 위해 포어그라운드에 있는) 활성화된 애플리케이션을 나타낸다.

[0038] 두 웹 페이지들(412), 제 2 웹 페이지(414) 및 제 3 웹 페이지(416)가 도시되어 있다. 제 2 웹 페이지(414)가 클라이언트 브라우저에 의해 다운로드되고 처리될 때, 브라우저는 브로커 컴포넌트(402)를 호출하는 내장 코드(418)를 실행시킨다. 클라이언트 브라우저에 의한 코드(418)의 실행으로 브로커 컴포넌트(402)로 전송되는 데이터가 생성되고, 마지막으로 등록된 입력 메소드 컴포넌트(예컨대, 제 2 입력 메소드 컴포넌트(404))에 기반한 부가 애플리케이션 기능으로서 데이터가 라우팅된다. 예를 들어, 제 2 웹 페이지(414)로부터 전송된 데이터는 부가 애플리케이션 기능(422)으로서 제 2 애플리케이션(406)으로 전송된다.

[0039] 이와 유사하게, 제 3 웹 페이지(416)가 클라이언트 브라우저에 의해 다운로드되고 처리될 때, 브라우저는 브로커 컴포넌트(402)를 호출하는 내장 코드(420)를 실행시킨다. 클라이언트 브라우저에 의한 코드(420)의 실행으로 브로커 컴포넌트(402)로 전송되는 데이터가 생성되고, 마지막으로 등록된 입력 메소드 컴포넌트(예컨대, 제 3 입력 메소드 컴포넌트(408))에 따라 부가 애플리케이션 기능으로서 데이터가 라우팅된다. 예를 들어, 제 3 웹 페이지(416)로부터 전송된 데이터는 부가 애플리케이션 기능(424)으로서 제 3 애플리케이션(410)으로 전송된다.

[0040] 브로커 컴포넌트가 단일 입력 메소드 컴포넌트(예컨대, 제 2 입력 메소드 컴포넌트(404)) 및 두 클라이언트 애플리케이션들을 지원하도록 개발된 경우도 본 대상 아키텍처의 사상에 속한다. 브로커 컴포넌트(402)와 마찬가

지로, 애플리케이션들(406 및 410) 및 연계된 입력 메소드 컴포넌트들(404 및 408)은 클라이언트 기반 애플리케이션들이다. 웹 페이지들(414 및 416)은 편리하고 효율적인 업데이트 배치를 위해 서버에 기반한다.

[0041] 본원은, 개시된 아키텍처의 새로운 양태들을 실행하고자 하는 방법론의 예들을 나타내는 일련의 순서도를 포함한다. 간단한 설명을 위해서, 순서도 또는 순서 다이어그램의 형태로 본원에서 도시된 하나 이상의 방법론은 일련의 액트로 도시되고 설명되었지만, 그 방법론들은 액트의 순서에 의해 제한되지 않으며, 액트들이 다른 순서로 실행되거나 본원에서 도시되고 설명된 다른 액트들과 동시에 실행될 수 있음을 알 것이다. 예를 들어, 당업자라면 방법론은 상태 다이어그램에서와 같이, 상호 연관된 일련의 상태 또는 이벤트로 표현될 수 있음을 알 것이다. 또한, 새로운 구현을 위해 방법론에 도시된 모든 액트들이 요구되는 것은 아니다.

[0042] 도 5는 브라우저를 통해 애플리케이션으로 컨텐츠를 입력하는 컨텐츠 메소드를 도시한다. 500에서, 기본 입력 메소드 기능이 애플리케이션에 제공된다. 502에서, 부가 애플리케이션 기능에 관련된 내장 코드를 포함하는 웹 페이지에 액세스한다. 504에서, 부가 애플리케이션 기능을 얻기 위해 웹 페이지의 내장 코드가 실행된다. 506에서, 부가 애플리케이션 기능이 애플리케이션에 추가된다.

[0043] 도 6은 도 5의 메소드의 추가적인 양태들을 도시한다. 600에서, 내장 코드는 스크립트 언어로 작성되어 웹 페이지에 마크업 언어로 내장된다. 602에서, 웹 페이지의 내장 코드는 일련의 실행을 위해 업데이트되고, 업데이트된 부가 애플리케이션 기능이 업데이트된 내장 코드에 기반하여 애플리케이션에 인가된다. 604에서, 기본 입력 메소드 기능을 브로커에게 제공하는 입력 메소드가 등록되고, 부가 애플리케이션 기능이 이 등록에 기반하여 브로커를 통해 애플리케이션으로 라우팅된다.

[0044] 도 7은 도 5의 메소드의 추가적인 양태들을 도시한다. 700에서, 웹 페이지와의 상호작용에 따라, 부가 애플리케이션 기능이 애플리케이션에 자동으로 추가된다. 702에서, 추적 정보를 사용하여 다수의 애플리케이션들 중에서 그 애플리케이션과의 상호작용이 추적된다. 704에서, 트래킹 정보는 통지를 통해 브로커 컴포넌트에 전송된다. 706에서, 추적 정보에 기반하여 부가 애플리케이션 기능이 해당 애플리케이션에 추가된다.

[0045] 본 출원에서 사용되는 바와 같이, 용어 "컴포넌트" 및 "시스템"은 컴퓨터-관련 엔티티, 예를 들어, 하드웨어, 하드웨어와 소프트웨어의 조합, 소프트웨어, 또는 실행 중인 소프트웨어를 말하기 위한 것이다. 예를 들어, 컴포넌트는 프로세서 상에서 실행 중인 프로세스, 프로세서, 하드 디스크 드라이브, (광, 고체 상태 및/또는 자기 저장 매체의) 다수의 저장 장치 드라이브, 객체, 실행 파일, 실행 스레드, 프로그램 및/또는 컴퓨터일 수 있지만, 이들로서 제한되지 않는다. 예시로서, 서버 상에서 실행 중인 애플리케이션 및 그 서버 모두 컴포넌트일 수 있다. 하나 이상의 컴포넌트가 프로세스 및/또는 실행 스레드 내에 존재할 수 있고, 컴포넌트가 하나의 컴퓨터 상에 로컬화되어 있을 수 있거나, 2개 이상의 컴퓨터 간에 분산되어 있을 수 있다. 본원에서 "예시적인"이라는 단어는 일례, 사례 또는 예시로서의 역할을 위해 사용될 수 있다. 본원에서 "예시적인" 것으로 기술된 양태 또는 설계가 반드시 다른 양태 또는 설계들보다 바람직하다거나 이점이 있는 것으로 해석되어서는 안 된다.

[0046] 이제, 도 8을 참조하면, 개시된 아키텍처에 따라 브라우저를 통해 애플리케이션으로의 컨텐츠의 입력을 용이하게 하는 컴퓨터 시스템(800)의 블록 다이어그램이 도시되어 있다. 개시된 아키텍처의 다양한 양태들에 대한 부가적인 상황을 제공하기 위해, 도 8 및 다음의 설명은, 이 다양한 양태들이 구현될 수 있는 적절한 컴퓨팅 시스템(800)에 대한 간략하고 개괄적인 설명을 제공하기 위한 것이다. 상기의 설명은 일반적으로 하나 이상의 컴퓨터 상에서 실행될 수 있는, 컴퓨터로 실행가능한 명령들과 관련하여 기술되어 있지만, 당업자라면 새로운 실시예도 다른 프로그램 모듈들과 함께 및/또는 하드웨어와 소프트웨어의 조합으로서 구현될 수 있음을 알 것이다.

[0047] 다양한 양태들을 구현하는 컴퓨팅 시스템(800)은 컴퓨터(802)를 포함하고, 이 컴퓨터(802)는 처리 장치(들)(804), 시스템 메모리(806) 및 시스템 버스(808)를 포함하고 있다. 처리 장치(들)(804)는 싱글 프로세서, 멀티 프로세서, 싱글 코어 장치 및 멀티 코어 장치를 비롯한 다양한 상용 프로세서들 중 어느 하나일 수 있다. 게다가, 당업자라면 새로운 방법들이, 미니컴퓨터, 메인프레임 컴퓨터뿐만 아니라, 개인 컴퓨터(예컨대, 데스크탑, 랩탑 등), 핸드헬드 컴퓨팅 장치, 마이크로프로세서 기반 또는 프로그램가능한 가전 제품 - 각각은 하나 이상의 연계 장치와 접속하여 동작 가능 - 등을 포함하는 다른 컴퓨터 시스템 구성들에서도 실시될 수 있음을 알 것이다.

[0048] 시스템 메모리(806)는 휘발성 메모리(VOL, 810)(예컨대, RAM(random access memory)) 및 비휘발성 메모리(NON-VOL, 812)(예컨대, ROM, EPROM, EEPROM 등)를 포함할 수 있다. 기본 입/출력 시스템(BIOS)은 비휘발성 메모리(812)에 저장될 수 있으며, 스타트업 중과 같은 때에 컴퓨터(802) 내의 컴포넌트들 간의 데이터 및 신호 전송을 용이하게 하는 기본 루틴을 포함한다. 또한, 휘발성 메모리(810)는 데이터를 캐싱하기 위한 정적 RAM을 비롯한

고속 RAM도 포함할 수 있다.

[0049] 시스템 버스(808)는, 처리 장치(들)(804)로의 메모리 서브시스템(806) 등을 포함하는 시스템 컴포넌트들에 대한 인터페이스를 제공한다. 시스템 버스(808)는, 각종 상용 버스 아키텍처 중 어느 하나를 사용하여 메모리 버스(메모리 컨트롤러를 갖거나 갖지 않음) 및 주변 장치 버스(예컨대, PCI, PCIe, AGP, LPC 등)와 추가적으로 상호 연결될 수 있는 몇몇 유형의 버스 구조 중 어느 하나일 수 있다.

[0050] 컴퓨터(802)는 저장 서브시스템(들)(814) 및, 저장 서브시스템(들)(814)을 시스템 버스(808) 및 소망하는 다른 컴퓨터 컴포넌트에 접속하는 저장 인터페이스(들)(816)를 더 포함한다. 예를 들어, 저장 서브시스템(들)(814)은 하나 이상의 HDD(hard disk drive), 자기 FDD(magnetic floppy disk drive) 및/또는 광 디스크 저장 드라이브(예컨대, CD-ROM 드라이브, DVD 드라이브)를 포함할 수 있다. 저장 인터페이스(들)(816)는 EIDE, ATA, SATA, 및 IEEE 1394 등의 인터페이스 기술들을 포함할 수 있다.

[0051] 메모리 서브시스템(806), 이동식 메모리 서브시스템(818)(예컨대, 플래시 드라이브 폼팩터 기술(flash drive form factor technology)) 및/또는 저장 서브시스템(들)(814, 예컨대, 광, 자기, 고체 상태)에는, 운영 체제(820), 하나 이상의 애플리케이션(822), 기타 프로그램 모듈(824) 및 프로그램 데이터(826)를 비롯한 하나 이상의 프로그램 및 데이터가 저장될 수 있다.

[0052] 하나 이상의 애플리케이션 프로그램(822), 기타 프로그램 모듈(824) 및 프로그램 데이터(826)는 도 1의 입력 메소드 컴포넌트(102), 애플리케이션(104), 기본 입력 메소드 기능(106), 브로커 컴포넌트(108), 부가 기능(114), 데이터(110), 브라우저(118), 웹 문서(112) 및 내장 코드(116), 도 2의 컴포넌트들과 엔티티들, 도 4의 컴포넌트와 엔티티들 및 도 5 내지 7의 메소드들에서 설명된 하나 이상의 액트 등을 포함할 수 있다.

[0053] 일반적으로, 프로그램들은 특정의 작업들을 수행하거나 특정의 추상 데이터 유형들을 구현하는 루틴, 메소드, 데이터 구조, 기타 소프트웨어 컴포넌트 등을 포함한다. 또한, 운영 체제(820), 애플리케이션(822), 모듈(824) 및/또는 데이터(826)의 전부 또는 일부도 휘발성 메모리(810) 등의 메모리에 캐싱될 수 있다. 개시된 아키텍처가 다양한 상용 운영 체제 또는 운영 체제들의 조합(예컨대, 가상 머신(virtual machines))에서 구현될 수 있음을 이해할 것이다.

[0054] 저장 서브시스템(들)(814) 및 메모리 서브시스템들(806 및 818)은 데이터, 데이터 구조, 컴퓨터로 실행가능한 명령어 등의 휘발성 및 비휘발성 저장을 위한, 컴퓨터로 판독가능한 매체로서 기능한다. 컴퓨터로 판독가능 매체는 컴퓨터(802)가 액세스할 수 있는 이용 가능한 매체라면 어느 것이라도 될 수 있고, 휘발성 및 비휘발성 매체, 이동식 및 비이동식 매체를 포함한다. 컴퓨터(802)를 위해, 이 매체는 어떠한 적절한 디지털 포맷으로도 데이터를 저장한다. 당업자라면 컴퓨터로 판독가능한 기타 유형의 매체로서, 개시된 아키텍처의 새로운 방법들을 수행하는, 컴퓨터로 실행가능한 인스트럭션을 저장하는 집 드라이브(zip drives), 자기 테이프, 플래시 메모리 카드, 카트리지 등이 사용될 수 있음을 알 것이다.

[0055] 사용자는 키보드 및 마우스를 비롯한 외부 사용자 입력 장치(828)를 사용하여 컴퓨터(802), 프로그램 및 데이터와 상호작용할 수 있다. 다른 외부 사용자 입력 장치들(828)로는 마이크, IR(infrared) 리모콘, 조이스틱, 게임 패드, 카메라 인식 시스템, 스타일러스 펜, 터치 스크린, 제스처 시스템(예컨대, 눈의 움직임, 머리의 움직임 등) 기타 등등이 있을 수 있다. 예를 들어, 컴퓨터(802)가 휴대용 컴퓨터인 경우에, 사용자는 터치패드, 마이크, 키보드 등을 비롯한 온보드 사용자 입력 장치(830)를 사용하여 컴퓨터(802), 프로그램 및 데이터와 상호작용할 수 있다. 이들 및 기타 입력 장치들은 I/O(input/output) 장치 인터페이스(들)(832)를 통해 시스템 버스(808)를 거쳐 처리 장치(들)(804)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스 등의 다른 인터페이스들에 의해 연결될 수도 있다. 또한, I/O 장치 인터페이스(들)(832)로 인해 프린터, 오디오 장치, 카메라 장치 등, 사운드카드 및/또는 온보드 오디오 처리 기능 등의 출력 주변 장치(834)가 사용이 용이해진다.

[0056] 하나 이상의 그래픽 인터페이스(들)(836, 통상 GPU(graphics processing unit)을 말함)은 컴퓨터(802)와 외부 디스플레이(들)(838, 예컨대, LCD, 플라즈마) 및/또는 온보드 디스플레이(840)(예컨대, 휴대용 컴퓨터의 경우) 간에 그래픽 및 비디오 신호를 제공한다. 그래픽 인터페이스(들)(836)는 컴퓨터 시스템 보드의 일부로서 제작될 수도 있다.

[0057] 컴퓨터(802)는 하나 이상의 네트워크 및/또는 기타 컴퓨터들과의 유/무선 통신 서브시스템(842)을 통한 논리적 연결을 사용하여 네트워크화된 환경(예컨대, IP)에서 동작할 수 있다. 기타 컴퓨터로는 워크스테이션, 서버, 라우터, 개인용 컴퓨터, 마이크로프로세서-기반 엔터테인먼트 기기, 피어 장치(peer device) 또는 기타 통상의 네

트워크 노드가 있을 수 있으며, 일반적으로 컴퓨터(802)와 관련하여 설명된 구성요소들 중 다수 또는 그 전부를 포함한다. 논리적 접속은 LAN(local area network), WAN(wide area network), 핫스팟 등에 대한 유/무선 접속을 포함ㅎ할 수 있다. LAN 및 WAN 네트워킹 환경은 사무실 및 회사에서 흔하게 볼 수 있고, 인트라넷 등의 전사적(enterprise-wide) 컴퓨터 네트워크를 가능하게 해주며, 이들 네트워크 전부는 글로벌 통신 네트워크, 예컨대, 인터넷에 접속할 수 있다.

[0058] 네트워킹 환경에서 사용될 때, 컴퓨터(802)는 유/무선 네트워크, 유/무선 프린터, 유/무선 입력 장치(844) 등과 통신하기 위해, 유/무선 통신 서브시스템(842, 예컨대, 네트워크 인터페이스 어댑터, 온보드 트랜시버 서브시스템 등)을 통해 네트워크에 연결된다. 컴퓨터(802)는 모뎀을 포함할 수 있고, 또는 네트워크 상에 통신을 구축하는 기타 수단을 가질 수 있다. 네트워크 환경에서, 컴퓨터(802)에 관련된 프로그램 및 데이터는 분산 시스템에 연계되어 있는 원격 메모리/저장 장치에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이고, 컴퓨터들 간에 통신 링크를 설정하는 기타 수단이 사용될 수 있음을 이해할 것이다.

[0059] 컴퓨터(802)는, 예를 들어, 프린터, 스캐너, 데스크탑 및/또는 휴대용 컴퓨터, PDA(personal digital assistant), 통신 위성, 무선 검출가능 태그와 연관된 장비 또는 장소(예컨대, 키오스크, 신문 가판대, 화장실) 및 전화와의 무선 통신(예컨대, IEEE 802.11 공중파(over-the-air) 변조 기법)으로 배치되어 동작하는 무선 장치들을 비롯한, IEEE 802.xx 계열의 표준들을 비롯한 무선 기술을 사용하는 유/무선 장치들 또는 엔티티들과 통신하는 동작을 한다. 이것은 적어도 핫스팟을 위한 Wi-Fi(Wireless Fidelity), WiMax 및 Bluetooth™ 무선 기술들을 포함한다. 따라서, 이러한 통신은 종래의 네트워크와 같이 사전 정의된 구조거나 또는 단순히 적어도 두 개의 장치들 간의 애드혹 통신(ad hoc communication)일 수 있다. Wi-Fi 네트워크들은 안전하고 신뢰성 있는 고속의 무선 연결을 제공하기 위해 IEEE 802.11x(a, b, g 등)라고 하는 무선 기술들을 사용한다. Wi-Fi 네트워크는 컴퓨터들을 서로 연결시키고, 인터넷에 연결시키며, 또한 유선 네트워크들(IEEE 802.3-관련 매체 및 기능을 사용함)에 연결시키는 데 사용될 수 있다.

[0060] 이제 도 9을 참조하면, 브라우저를 통해 애플리케이션으로의 컨텐츠의 입력을 용이하게 하는 컴퓨팅 환경(900)의 개략적인 블록 다이어그램이 도시되어 있다. 환경(900)은 하나 이상의 클라이언트(들)(902)를 포함한다. 클라이언트(들)(902)는 하드웨어 및/또는 소프트웨어(예컨대, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 클라이언트(들)(902)는, 예를 들어, 쿠키(들) 및/또는 연관된 컨텍스트 정보(contextual information)를 가지고 있을 수 있다.

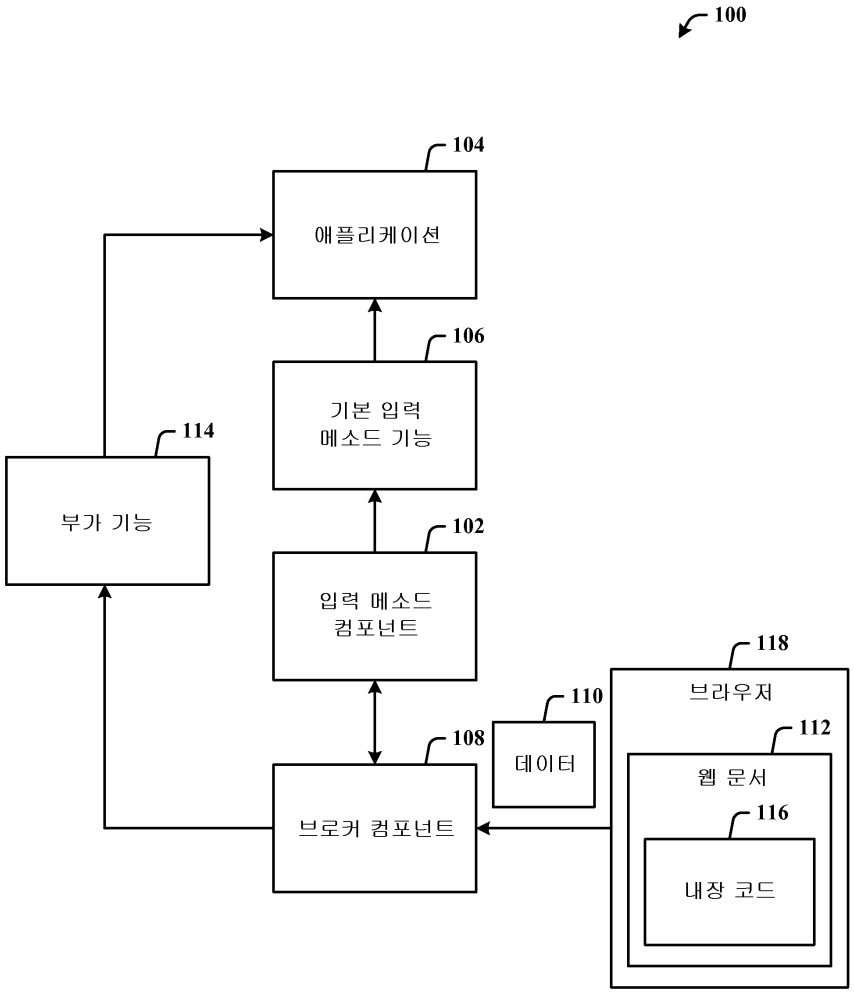
[0061] 또한, 환경(900)은 하나 이상의 서버(들)(904)도 포함하고 있다. 서버(들)(904)도 역시 하드웨어 및/또는 소프트웨어(예컨대, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 서버들(904)은, 예를 들어, 본 아키텍처를 이용하여 변환을 수행하는 스레드를 가지고 있을 수 있다. 클라이언트(902)와 서버(904) 간의 가능한 하나의 통신은 2개 이상의 컴퓨터 프로세스들 간에 전송되기에 적합한 데이터 패킷의 형태로 되어 있을 수 있다. 이 데이터 패킷은, 예를 들어, 쿠키 및/또는 연계된 컨텍스트 정보를 포함할 수 있다. 환경(900)은 클라이언트(들)(902)와 서버(들)(904) 간의 통신을 용이하게 해주는 데 이용될 수 있는 통신 프레임워크(906, 예컨대, 인터넷 등의 전세계 통신 네트워크)를 포함하고 있다.

[0062] 유선(광섬유 포함) 및/또는 무선 기술을 통해 통신이 용이해 질 수 있다. 클라이언트(들)(902)는, 로컬인 정보(예컨대, 쿠키(들) 및/또는 연계된 컨텍스트 정보)를 클라이언트(들)(902)에 저장하는 데 이용될 수 있는 하나 이상의 클라이언트 데이터 저장소(들)(908)에 연결되어 동작한다. 이와 유사하게, 서버(들)(904)는 서버들(904)에 로컬인 정보를 저장하는 데 이용될 수 있는 하나 이상의 서버 데이터 저장소(들)(910)에 연결되어 동작한다.

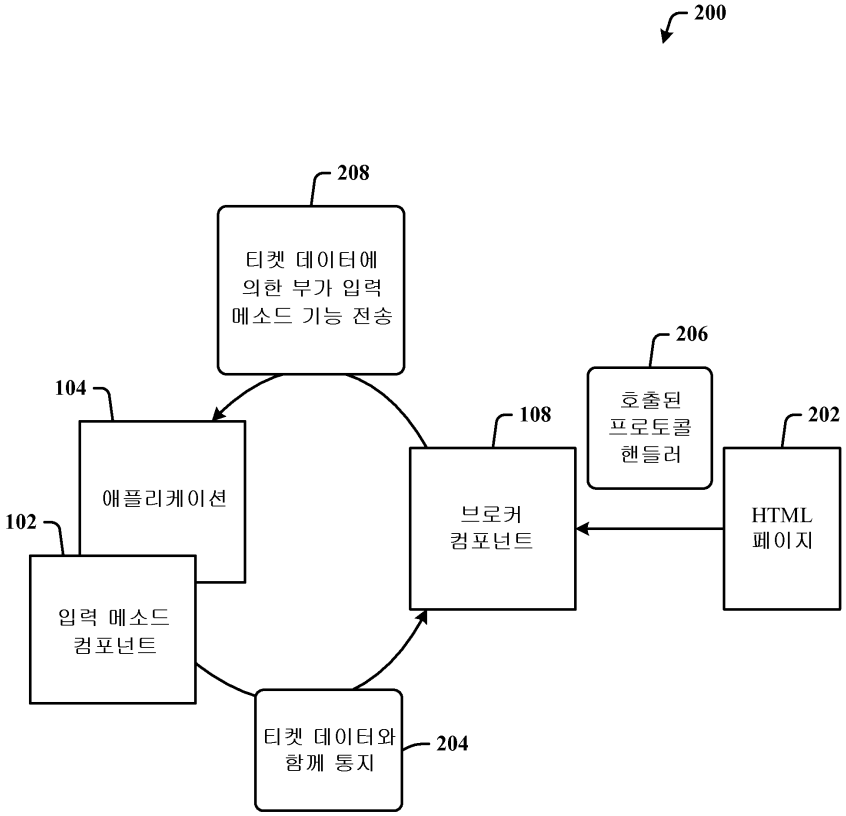
[0063] 개시된 아키텍처의 일례들에 대해 이상에서 설명하였다. 물론, 컴포넌트 및/또는 방법론들의 생각 가능한 모든 조합을 다 기술하는 것은 불가능하지만, 당업자라면 많은 추가의 조합 및 치환이 가능함을 알 수 있을 것이다. 따라서, 이 새로운 아키텍처는, 첨부된 특허청구범위의 사상 및 범위 내에 속하는 이러한 모든 변경, 수정 및 변형을 포괄하기 위한 것이다. 게다가, "포함한다"라는 용어가 상세한 설명 또는 특허청구범위에서 사용되는 한, 이러한 용어는, "포함하는"이라는 용어가 청구항에서 전이구(transitional word)로써 이용될 때 해석되는 것과 유사한 방식으로 포함적(inclusive)인 것으로 보아야 할 것이다.

도면

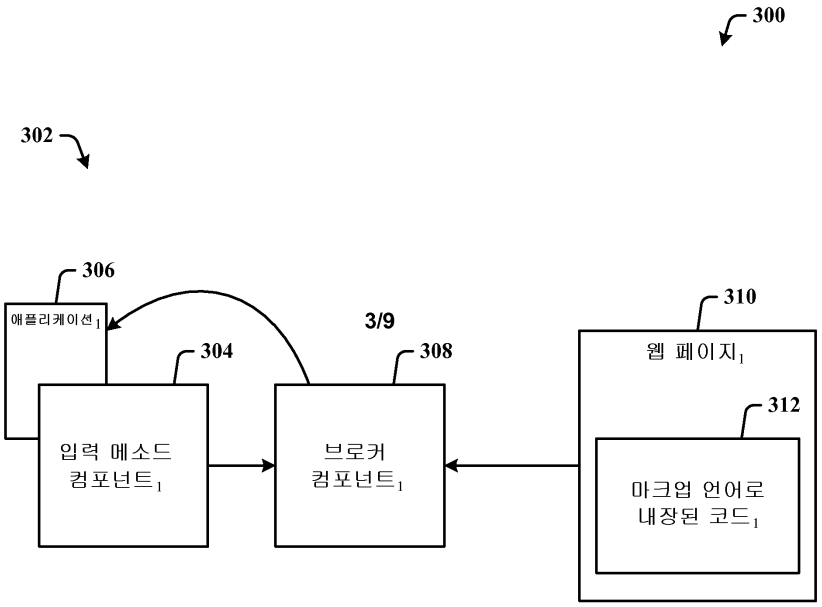
도면1


100
104
애플리케이션
106
기본 입력
메소드 기능
114
부가 기능
102
입력 메소드
컴포넌트
118
브라우저
110
데이터
112
108
웹 문서
116
브로커 컴포넌트
내장 코드

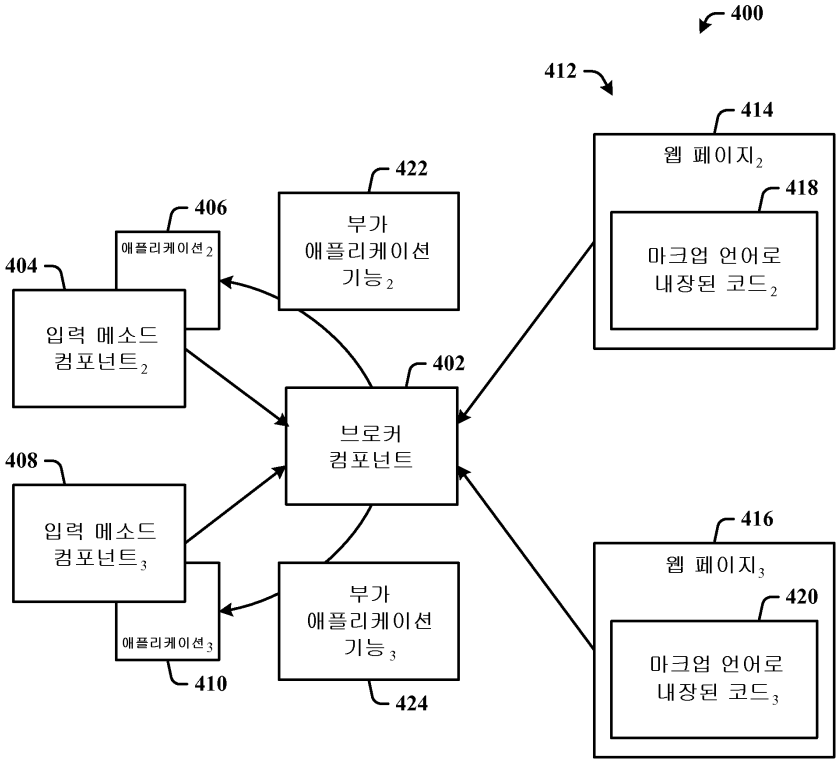
도면2


200
208
티켓 데이터에 의한 부가 입력 메소드 기능 전송
206
호출된 프로토콜 핸들러
104
108
202
애플리케이션
브로커 컴포넌트
HTML 페이지
102
입력 메소드 컴포넌트
티켓 데이터와 함께 통지
204

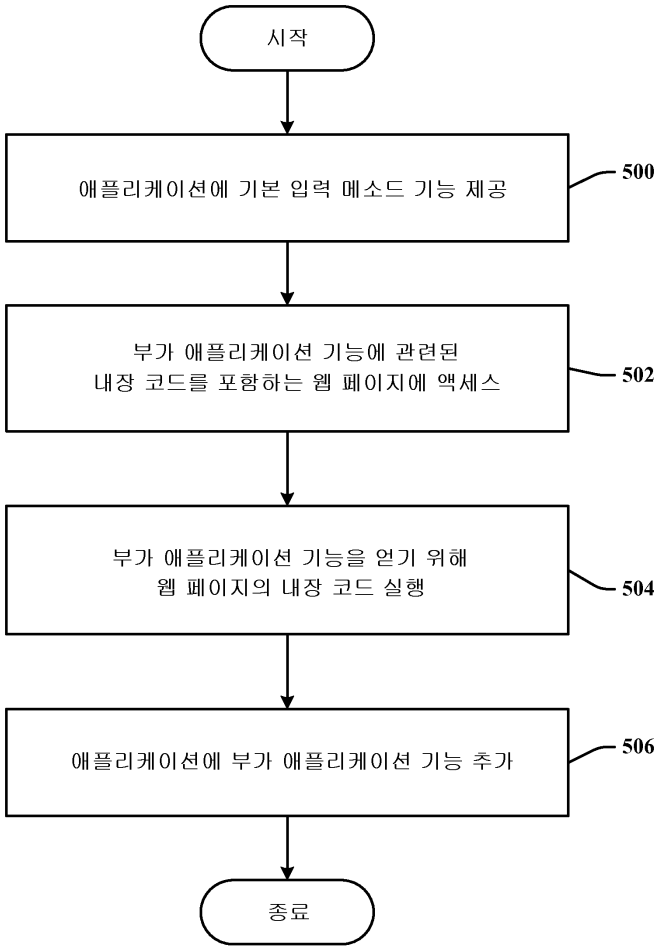
도면3


300
302
306
애플리케이션1
3/9
304
308
310
웹 페이지1
312
입력 메소드 컴포넌트1
브로커 컴포넌트1
마크업 언어로 내장된 코드1

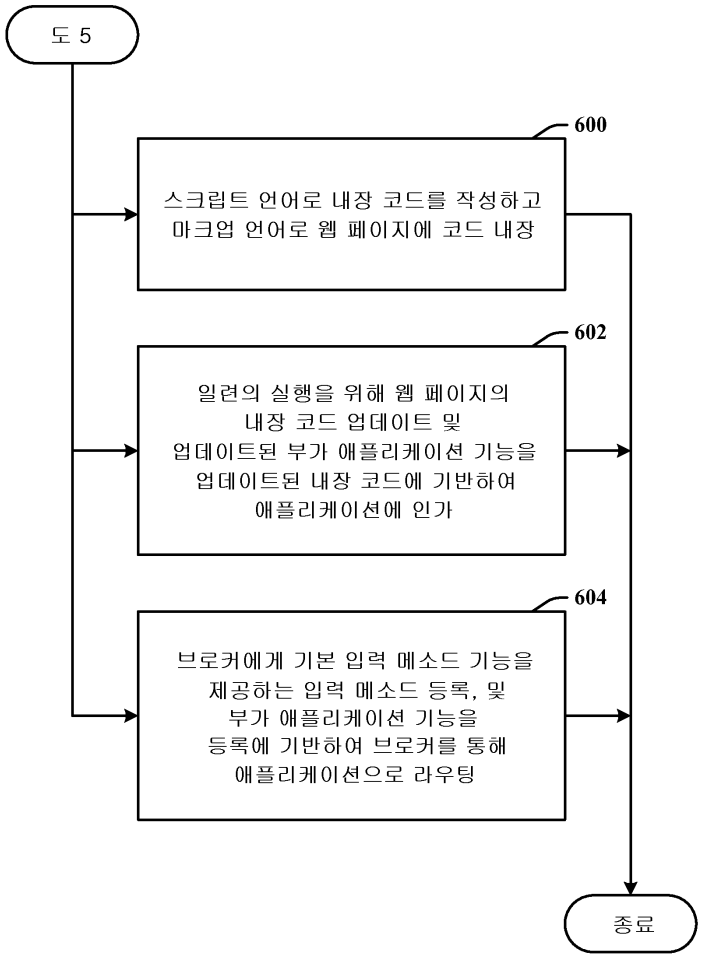
도면4


400
412
414
웹 페이지2
418
마크업 언어로 내장된 코드2
422
부가 애플리케이션 기능2
406
애플리케이션2
404
입력 메소드 컴포넌트2
402
브로커 컴포넌트
408
입력 메소드 컴포넌트3
애플리케이션3
410
부가 애플리케이션 기능3
424
416
웹 페이지3
420
마크업 언어로 내장된 코드3

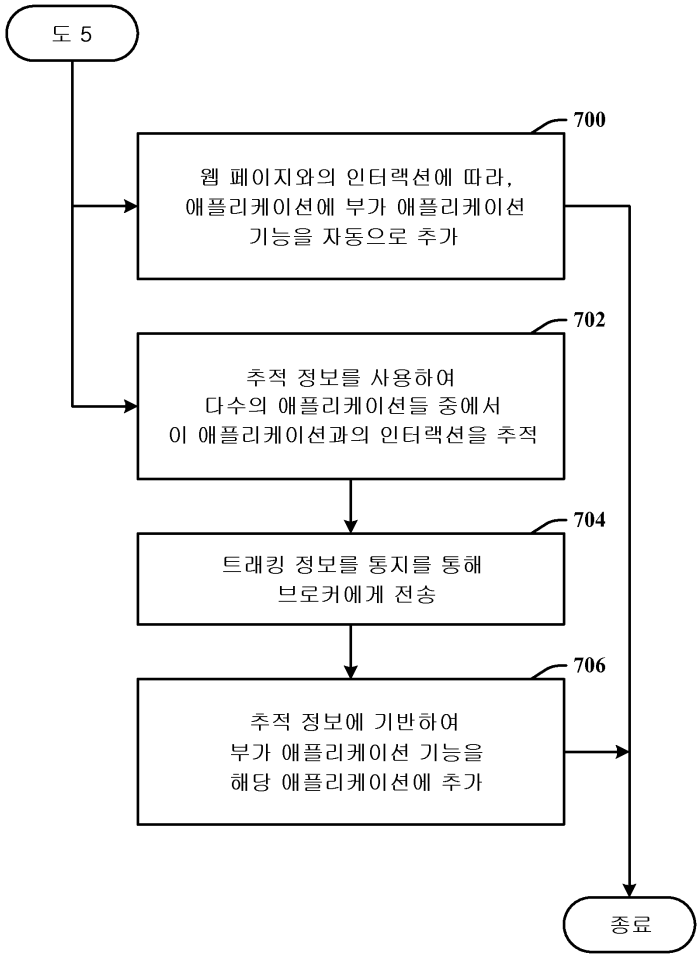
도면5


시작
애플리케이션에 기본 입력 메소드 기능 제공
500
부가 애플리케이션 기능에 관련된 내장 코드를 포함하는 웹 페이지에 액세스
502
부가 애플리케이션 기능을 얻기 위해 웹 페이지의 내장 코드 실행
504
애플리케이션에 부가 애플리케이션 기능 추가
506
종료

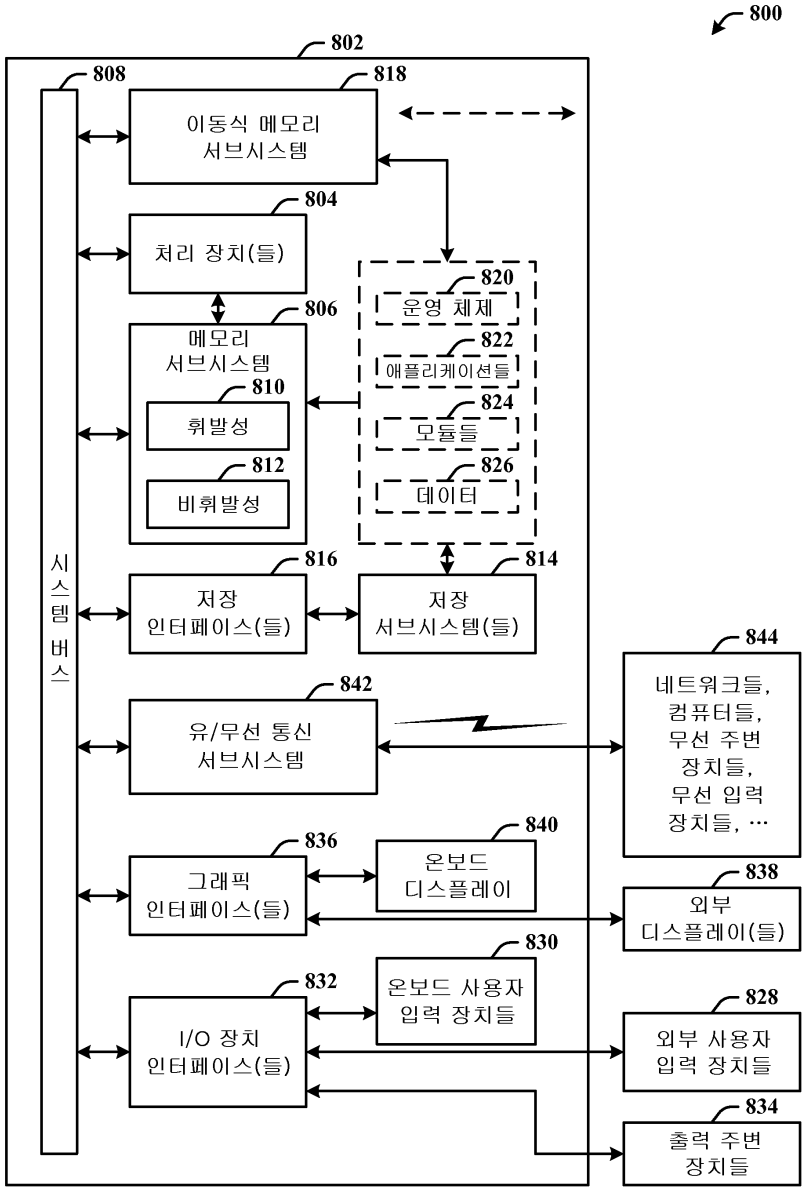
도면6


도 5
600
스크립트 언어로 내장 코드를 작성하고
마크업 언어로 웹 페이지에 코드 내장
602
일련의 실행을 위해 웹 페이지의
내장 코드 업데이트 및
업데이트된 부가 애플리케이션 기능을
업데이트된 내장 코드에 기반하여
애플리케이션에 인가
604
브로커에게 기본 입력 메소드 기능을
제공하는 입력 메소드 등록, 및
부가 애플리케이션 기능을
등록에 기반하여 브로커를 통해
애플리케이션으로 라우팅
종료

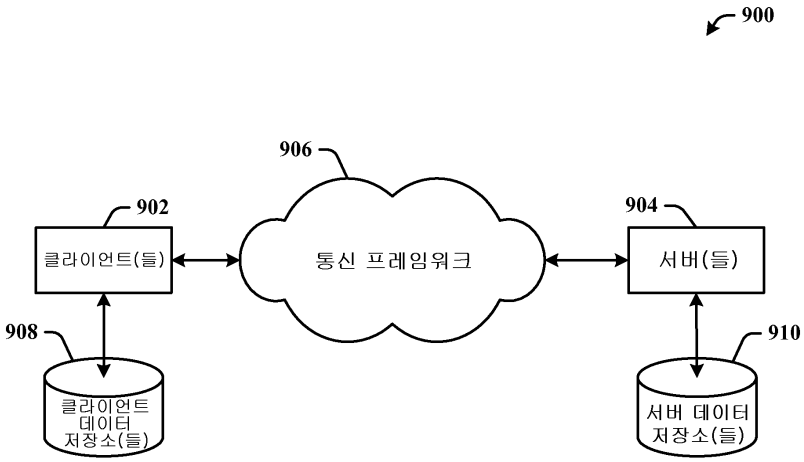
도면7


도 5
700
웹 페이지와의 인터랙션에 따라,
애플리케이션에 부가 애플리케이션
기능을 자동으로 추가
702
추적 정보를 사용하여
다수의 애플리케이션들 중에서
이 애플리케이션과의 인터랙션을 추적
704
트래킹 정보를 통지를 통해
브로커에게 전송
706
추적 정보에 기반하여
부가 애플리케이션 기능을
해당 애플리케이션에 추가
종료

도면8


800
802
808
818
이동식 메모리 서브시스템
804
처리 장치(들)
820
운영 체제
822
애플리케이션들
824
모듈들
826
데이터
806
메모리 서브시스템
810
휘발성
812
비휘발성
816
저장 인터페이스(들)
814
저장 서브시스템(들)
시스템 버스
842
유/무선 통신 서브시스템
844
네트워크들, 컴퓨터들, 무선 주변 장치들, 무선 입력 장치들, …
836
그래픽 인터페이스(들)
840
온보드 디스플레이
838
외부 디스플레이(들)
832
I/O 장치 인터페이스(들)
830
온보드 사용자 입력 장치들
828
외부 사용자 입력 장치들
834
출력 주변 장치들

도면9


900
906
902
클라이언트(들)
통신 프레임워크
904
서버(들)
908
클라이언트 데이터 저장소(들)
910
서버 데이터 저장소(들)