



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0071053
(43) 공개일자 2020년06월18일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 9/54 (2018.01) H04L 29/06 (2006.01)
H04L 29/08 (2006.01)</p> <p>(52) CPC특허분류
G06F 9/541 (2013.01)
H04L 67/02 (2013.01)</p> <p>(21) 출원번호 10-2020-0067645(분할)</p> <p>(22) 출원일자 2020년06월04일
심사청구일자 2020년06월04일</p> <p>(62) 원출원 특허 10-2013-0112794
원출원일자 2013년09월23일
심사청구일자 2018년09월18일</p> | <p>(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)</p> <p>(72) 발명자
류영선
경기도 수원시 영통구 삼성로 129</p> <p>(74) 대리인
이건주, 김정훈</p> |
|---|---|

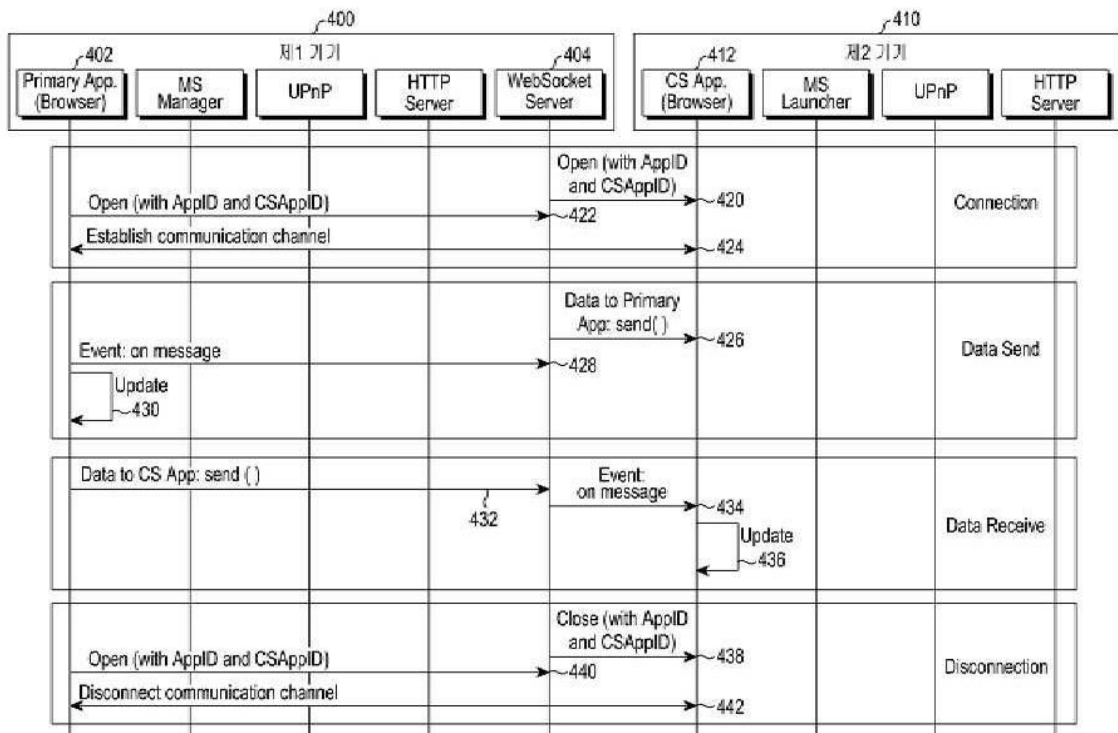
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 기기 별 응용 프로그램간 통신을 위한 장치 및 방법

(57) 요약

본 개시의 실시예들에 따르면, 제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법은 상기 제 1 디바이스에 의해, 상기 제 2 디바이스 내의 웹소켓 서버를 통해 상기 제 1 어플리케이션을 상기 제 2 어플리케이션에 접속시키는 단계, 및 상기 제 1 디바이스에 의해, 상기 웹소켓 서버를 (뒷면에 계속)

대표도



통해 상기 제 1 어플리케이션으로부터 상기 제 2 어플리케이션으로 데이터를 전송하거나, 또는 상기 제 1 디바이스에 의해, 상기 웹소켓 서버를 통해 상기 제 2 어플리케이션으로부터 상기 제 1 어플리케이션에 대한 데이터를 수신하는 단계를 포함하고, 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버가 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속 요청을 수신하는 것에 기초하여 제공되고, 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버의 포트, 및 웹소켓 프로토콜을 지원하는 특정 정보에 기초한다.

(52) CPC특허분류

H04L 67/146 (2013.01)

H04L 69/162 (2013.01)

명세서

청구범위

청구항 1

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법으로

서,

상기 제 1 디바이스에 의해, 상기 제 2 디바이스 내의 웹소켓 서버를 통해 상기 제 1 어플리케이션을 상기 제 2 어플리케이션에 접속시키는 단계; 및

상기 제 1 디바이스에 의해, 상기 웹소켓 서버를 통해 상기 제 1 어플리케이션으로부터 상기 제 2 어플리케이션으로 데이터를 전송하거나, 또는 상기 제 1 디바이스에 의해, 상기 웹소켓 서버를 통해 상기 제 2 어플리케이션으로부터 상기 제 1 어플리케이션에 대한 데이터를 수신하는 단계를 포함하고,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버가 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속 요청을 수신하는 것에 기초하여 제공되고,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버의 포트, 및 웹소켓 프로토콜을 지원하는 특정 정보에 기초하는,

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법.

청구항 2

제 1 항에 있어서,

상기 웹소켓 서버와 상기 제 1 어플리케이션 사이의 제 1 접속은 상기 웹소켓 서버의 프로토콜을 이용하여 제공되고, 그리고

상기 웹소켓 서버와 상기 제 2 어플리케이션 사이의 제 2 접속은 상기 웹소켓 서버의 상기 프로토콜을 이용하여 제공되는,

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법.

청구항 3

제 2 항에 있어서,

상기 제 1 접속 및 상기 제 2 접속은 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 통신 채널을 설정하기 위해 상기 웹소켓 서버에 의해 중계되는(relayed),

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법.

청구항 4

제 3 항에 있어서,

상기 제 1 접속 및 상기 제 2 접속은 동일한 채널 식별자로서 할당되는,

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법.

청구항 5

제 1 항에 있어서,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 접속은 상기 웹소켓 서버에 의해 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속해제(disconnection) 요청을 수신하는 것에 의해 응답하여 접속해제되는,

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법.

청구항 6

제 1 항에 있어서,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 사용자 요청을 위하여 디스플레이된 요청에 응답하여 상기 제 1 디바이스의 사용자 또는 상기 제 2 디바이스의 사용자 중 적어도 하나에 의한 사용자 입력 이후에 제공되는,

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법.

청구항 7

제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 상기 제 1 디바이스로서,

통신 인터페이스; 및

상기 통신 인터페이스에 연결된 제어기를 포함하고,

상기 제어기는:

상기 제 2 디바이스 내의 웹소켓 서버를 통해 상기 제 1 어플리케이션을 상기 제 2 어플리케이션에 접속시키고; 그리고

상기 웹소켓 서버를 통해 상기 제 1 어플리케이션으로부터 상기 제 2 어플리케이션으로 데이터를 전송하거나, 또는 상기 웹소켓 서버를 통해 상기 제 2 어플리케이션으로부터 상기 제 1 어플리케이션에 대한 데이터를 수신하도록 구성되고,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버가 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속 요청을 수신하는 것에 기초하여 제공되고,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버의 포트, 및 웹소켓 프로토콜을 지원하는 특정 정보에 기초하는,

제 1 디바이스.

청구항 8

제 7 항에 있어서,

상기 웹소켓 서버와 상기 제 1 어플리케이션 사이의 제 1 접속은 상기 웹소켓 서버의 프로토콜을 이용하여 제공되고, 그리고

상기 웹소켓 서버와 상기 제 2 어플리케이션 사이의 제 2 접속은 상기 웹소켓 서버의 상기 프로토콜을 이용하여 제공되는,

제 1 디바이스.

청구항 9

제 8 항에 있어서,

상기 제 1 접속 및 상기 제 2 접속은 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 통신 채널을 설정하기 위해 상기 웹소켓 서버에 의해 중계되는(relayed),

제 1 디바이스.

청구항 10

제 9 항에 있어서,

상기 제 1 접속 및 상기 제 2 접속은 동일한 채널 식별자로서 할당되는,

제 1 디바이스.

청구항 11

제 7 항에 있어서,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 접속은 상기 웹소켓 서버에 의해 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속해제(disconnection) 요청을 수신하는 것에 의해 응답하여 접속해제되는,

제 1 디바이스.

청구항 12

제 7 항에 있어서,

상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 사용자 요청을 위하여 디스플레이된 요청에 응답하여 상기 제 1 디바이스의 사용자 또는 상기 제 2 디바이스의 사용자 중 적어도 하나에 의한 사용자 입력 이후에 제공되는,

제 1 디바이스.

발명의 설명

기술 분야

[0001] 본 발명은 서로 다른 기기를 통해서 구동되는 응용프로그램들 간의 통신 방법 및 장치에 관한 것이다.

배경 기술

[0003] 무선 통신의 증대에 따라 실시간 양방향 데이터의 통신 및 다수의 동시 접속자를 수용 가능한 웹 소켓(websocket) 통신이 도입되었다. 이러한 웹소켓 통신은 서버와 클라이언트간의 통신을 기반으로 이루어진다.

[0004] 도 1은 일반적인 웹소켓 통신의 동작 방식의 일 예를 도시한 도면이다.

[0005] 도 1을 참조하면, 서버(102)는 웹 소켓 통신을 지원하는 서버이며, 클라이언트들 1 내지 4(104~110)들 중 하나

의 클라이언트에서는 웹소켓 프로토콜(protocol)을 사용하여 원하는 웹소켓 서버와 연결하여 데이터를 주고 받는 경우를 가정하자.

[0006] 이러한 웹소켓 통신은 기존의 통신 방법 중 하나인 HTTP(Hyper Text Transfer Protocol)가 가지고 있는 단점 (stateless, 단방향 통신)을 극복하여 서버와 클라이언트 사이에 끊임없는 양방향 통신을 제공해준다. 또한, 서버 측에서 복잡한 프로그래밍 없이 웹(web)을 통해서 일반적인 TCP(Transmission Control Protocol) 소켓과 같이 실시간 연결지향 양방향 통신을 가능하게 하는 HTML5와의 연동이 쉽게 됨에 따라 기본적인 양방향 통신이 필요한 환경에서 다양하게 활용될 수 있을 것으로 기대된다.

[0007] 따라서 이러한 웹소켓 통신의 장점을 기반으로, 서로 다른 기기에서 구동되는 응용프로그램들간의 통신에 적용하는 방안을 고려해볼 수 있다. 그러므로, 서버와 클라이언트 사이의 양방향 통신을 위해 설계된 웹소켓 통신을 응용프로그램들간의 통신에 활용하기 위해서 추가적인 절차들이 요구된다.

발명의 내용

해결하려는 과제

[0009] 본 발명은 서로 다른 기기에서 동작하는 서로 다른 응용프로그램간 통신을 제공할 수 있는 장치 및 방법을 제안한다.

[0010] 본 발명은, 제1기기에서 하나의 응용프로그램이 구동되고, 제2기기에서 상기 응용프로그램과 연동된 보조 응용프로그램이 구동되는 경우, 웹소켓 프로토콜을 사용하여 두 응용프로그램간의 통신을 제공하기 위한 장치 및 방법을 제안한다.

과제의 해결 수단

[0011] 본 개시의 일 실시예에 따른 제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 방법은, 상기 제 1 디바이스에 의해, 상기 제 2 디바이스 내의 웹소켓 서버를 통해 상기 제 1 어플리케이션을 상기 제 2 어플리케이션에 접속시키는 단계, 및 상기 제 1 디바이스에 의해, 상기 웹소켓 서버를 통해 상기 제 1 어플리케이션으로부터 상기 제 2 어플리케이션으로 데이터를 전송하거나, 또는 상기 제 1 디바이스에 의해, 상기 웹소켓 서버를 통해 상기 제 2 어플리케이션으로부터 상기 제 1 어플리케이션에 대한 데이터를 수신하는 단계를 포함하고, 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버가 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속 요청을 수신하는 것에 기초하여 제공되고, 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버의 포트, 및 웹소켓 프로토콜을 지원하는 특정 정보에 기초한다.

[0013] 본 개시의 다른 실시예에 따른 제 1 디바이스 상의 제 1 어플리케이션과 제 2 디바이스 상의 제 2 어플리케이션 사이의 통신을 위한 상기 제 1 디바이스는, 통신 인터페이스, 및 상기 통신 인터페이스에 연결된 제어기를 포함하고, 상기 제어기는, 상기 제 2 디바이스 내의 웹소켓 서버를 통해 상기 제 1 어플리케이션을 상기 제 2 어플리케이션에 접속시키고, 그리고 상기 웹소켓 서버를 통해 상기 제 1 어플리케이션으로부터 상기 제 2 어플리케이션으로 데이터를 전송하거나, 또는 상기 웹소켓 서버를 통해 상기 제 2 어플리케이션으로부터 상기 제 1 어플리케이션에 대한 데이터를 수신하도록 구성되고, 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버가 상기 제 1 어플리케이션 및 상기 제 2 어플리케이션의 각각으로부터 접속 요청을 수신하는 것에 기초하여 제공되고, 상기 제 1 어플리케이션과 상기 제 2 어플리케이션 사이의 상기 웹소켓 서버를 통한 접속은 상기 웹소켓 서버의 포트, 및 웹소켓 프로토콜을 지원하는 특정 정보에 기초한다.

발명의 효과

[0015] 본 발명은, 서로 다른 기기에서 구동되는 응용프로그램들간의 통신 수단을 제공함으로써, 예를 들어, 제1기기의 응용프로그램과 연동된 제2기기의 응용프로그램을 작성할 수 있다. 이를 통해, 제2기기에서 제1기기를 제어할

수 있는 리모트 콘트롤 응용프로그램을 작성한다거나, 문자 입력이 필요한 경우 제1기기의 불편한 입력장치 대신 사용할 수 있는 제2기기의 입력 응용프로그램의 작성 등의 사용자 경험 등을 제공할 수 있는 효과가 있다.

도면의 간단한 설명

- [0017] 도 1은 일반적인 웹소켓의 동작 방식의 일 예를 도시한 도면,
- 도 2는 본 발명의 실시 예에 따른 응용 프로그램간 통신을 설명하기 위한 기본적인 구성도,
- 도 3은 본 발명의 실시 예에 따라 제1기기에서 웹소켓 서버를 구동 및 정지하는 동작 흐름도,
- 도 4는 본 발명의 실시 예에 따라 제2기기의 응용프로그램이 제1기기의 응용프로그램과 통신하는 동작 흐름도,
- 도 5는 본 발명의 실시 예에 따른 제1기기의 동작 흐름도,
- 도 6은 본 발명의 실시 예에 따른 제2기기의 동작 흐름도,
- 도 7은 본 발명이 실제 적용된 실시 예를 도시한 도면.

발명을 실시하기 위한 구체적인 내용

[0018] 이하 첨부된 도면을 참조하여 본 발명의 바람직한 실시 예에 대한 동작 원리를 상세히 설명한다. 도면상에 표시된 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 참조번호로 나타내었으며, 다음에서 본 발명을 설명함에 있어 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0020] 본 발명은 서로 다른 기기에서 동작하는 서로 다른 응용프로그램간 통신을 제공할 수 있는 장치 및 방법을 제안한다. 구체적으로, 본 발명은 제1기기에서 하나의 응용프로그램이 구동되고, 제2기기에서 상기 응용프로그램과 연동된 보조 응용프로그램이 구동되는 경우, 웹소켓 프로토콜을 사용하여 두 응용프로그램간의 통신을 제공하기 위한 장치 및 방법을 제안한다.

[0022] 도 2는 본 발명의 실시 예에 따른 응용 프로그램간 통신을 설명하기 위한 기본적인 구성도의 일 예이다. 설명의 편의상 도 2의 기기 구성들은 본 발명의 실시 예에 따른 구성들만 포함하는 개략적인 구성도이다. 따라서, 사업자의 의도나 상황에 따라 해당 구성들은 하나의 유닛(unit)으로 합쳐지거나, 기능에 따라 서브(sub) 유닛들로 분할되어 구성될 수도 있다.

[0023] 도 2를 참조하면, 제1기기(200)는 일 예로, 웹브라우저(204)와 웹소켓 서버(206)를 포함한다. 여기서, 상기 웹브라우저(204)는 웹 응용프로그램을 구동할 수 있는 구성요소에 해당하며, 상기 웹소켓 서버(206)는 본 발명의 실시 예에 따라 다른 기기의 응용프로그램과, 상기 제1기기(200)의 응용프로그램간 통신을 위해서 추가된 구성요소이다.

[0024] 상기한 바와 같은 제1기기(200)의 구성과 비교하여, 제2기기(210)는 일 예로, 웹브라우저(212)를 포함하도록 구성한다. 상기 제1기기(200)와 마찬가지로 상기 웹브라우저(212)는 웹 응용프로그램을 구동할 수 있는 구성요소이다. 이하, 명세서에서는 웹 응용프로그램에 대해 예로 들어, 응용프로그램간 통신으로 설명하고 있다. 그러나, 본 발명에서 설명하는 절차들과 API(Application Program Interface)를 지원하는 응용프로그램일 경우, 어떠한 응용프로그램도 적용될 수 있다. 예를 들어, 안드로이드 OS(Operation System)나 아이폰의 OS(iOS)에서 동작하는 네이티브 응용프로그램 등이 해당한다.

[0026] 여기서, 상기 제1기기(200)와 제2기기(210)의 응용프로그램(웹 응용프로그램)은 서로간 직접 통신을 할 수 없기 때문에, 상기 제1기기(200)에 존재하는 웹 소켓 서버(206)를 프락시(Proxy)처럼 사용하여 통신한다. 이때, 상기 제1기기(200)의 웹소켓 서버(206)와 상기 웹브라우저(204) 및 상기 제2기기(210)의 웹브라우저(212 또는 네이티브

브 응용프로그램, 이하 생략) 사이에는 WS(WebServer) 프로토콜을 사용하여 통신한다.

[0028] 이하, 설명의 편의상, 본 발명의 실시 예들에 따른 응용프로그램들 간의 통신을 지원하는 장치 및 동작들은 도 2의 구성 예를 기반으로 설명하기로 한다.

[0030] 도 3은 본 발명의 실시 예에 따라 제1기기에서 웹소켓 서버를 구동 및 정지하는 동작 흐름도이다. 설명의 편의상, 상기 제1기기는 DTV(Digital Television)이고, 제2 기기가 모바일(Mobile) 기기인 경우를 가정하여 설명하기로 한다. 그러나, 본 발명의 실시 예가 적용되는 제 1기기와 제 2기기는 해당 기술을 지원하는 어떠한 디바이스(device)도 될 수 있다.

[0031] 먼저, 제1기기(310)에서 웹소켓 서버(304)의 구동 동작을 설명하기로 한다. 도 3을 참조하면, 322단계에 제1기기(310)의 웹 응용프로그램(302)은 API를 통해 상기 제1기기(310)의 웹소켓 서버(304)를 구동한다. 여기서, API는 하기 [표1]에 나타낸 바와 같이, start API를 사용할 수 있다.

표 1

API	설명	인자	리턴값
startAPI(uint AppID)	기기의 웹소켓 서버를 구동한다.	uint AppID: 웹소켓 서버와 연동할 응용프로그램의 식별자	True: 구동 성공 False: 구동 실패
shutdownWS(uint AppID)	기기의 웹소켓 서버를 정지한다.	uint AppID: 웹소켓 서버와 연동할 응용프로그램의 식별자	True: 정지 성공 False: 정지 실패

[0034] 이때, 상기 제1기기(300)의 응용프로그램(302)과 연동을 위해 상기 응용프로그램의 ID(Identifier)를 함께 제공한다. 여기서, 응용프로그램의 ID는, 실제 구현 시 해당 응용프로그램을 식별할 수 있는 ID나 URL(Uniform resource Locator) 등이 될 수 있다. 그러면, 웹소켓 서버(304)는 상기 응용프로그램의 ID에 대응하는 특정 응용프로그램과 통신하기 위해서 자신을 구동시키고, 이에 대한 결과를 상기 제1기기(300)의 응용 프로그램(302)에게 전송한다.

[0036] 다음으로, 상기 제1기기(300)에서 웹소켓 서버(304)를 정지하는 동작을 설명하기로 한다.

[0037] 326단계에서 제1기기(300)의 웹 응용프로그램(302)은 API를 통해 상기 제1기기(310)의 웹소켓 서버(304)를 정지한다. 이때, 상기 웹 응용프로그램(302)에서 호출하는 API는 상기 [표1]과 같이 shutdown WS를 사용할 수 있다. 마찬가지로, 상기 제1기기(300)의 상기 응용프로그램의 ID를 함께 제공한다. 그러면, 웹소켓 서버(310)는 상기 응용 프로그램의 ID에 대응하는 응용 프로그램과의 통신을 중단하고, 이에 대한 결과를 상기 제1기기(300)의 응용 프로그램(302)에게 전송한다.

[0038] 도 3의 실시 예에서는 특정 응용프로그램을 위한 웹소켓 서버를 제1기기가 직접 구동 및 정지시키는 방법에 대해 설명하였다. 그러나, 다른 실시 예에서는 공용으로 사용할 수 있는 웹소켓 서버를 제2기기가 먼저 구동시킨 후, 상기 켜 놓고, 다음의 절차에서 설명할 API들을 이용하여 제2기기의 응용프로그램이 상기 제1기기의 웹소켓 서버와 연동될 수 있다.

[0040] 도 4는 본 발명의 실시 예에 따라 제2기기의 응용프로그램이 제1기기의 응용프로그램과 통신하는 동작 흐름도이다. 참고로, 하기 <표 2>는 응용 프로그램들 간의 통신을 위한 API들의 예를 나타낸 표이다.

표 2

API	설명	인자	리턴값
startWS(uint AppID)	웹소켓 서버를 구동시킨다.	uint AppID: 웹소켓 서버를 구동시킬 응용프로그램	True: 구동 성공 False: 구동 실패
ShutdownWS(uint AppID)	웹소켓 서버를 정지시킨다.	uint AppID: 웹소켓 서버를 정지시킬 응용프로그램	True: 정지 성공 False: 정지 실패
check_channel(AppID, CSAppID)	제1기기의 응용프로그램(AppID)와 제2기기의 응용프로그램(CSAppID)간 통신채널이 생성되었는지 확인한다.	AppID: 제1기기의 응용프로그램 ID IDCSAppID: 제2기기의 응용프로그램 ID	True: 특정채널이 생성되어 있음 False: 특정채널이 생성되어 있지 않음
getHbbTVAppID()	제1기기의 응용프로그램 ID를 구한다.	없음	uint HbbTVID: 제1기기의 응용프로그램ID
getCSAppID()	제2기기의 응용프로그램 ID를 구한다.	없음	uint CSAppID: 제2기기의 응용프로그램 ID

[0042] 도 4를 참조하면, 420단계에서 제2기기(410)의 응용프로그램(412)은 웹소켓 객체를 생성하고 open() API를 호출하여 제1기기(400)의 웹소켓 서버(404)에게 통신 채널의 생성을 요청한다. 이때, open() API는 상기 제1기기(400)의 응용프로그램 ID(AppID)와, 제2기기(410)의 응용 프로그램의 ID(CSAppID)를 전달한다. 그러면, 422단계에서, 상기 제1기기(400)의 응용프로그램(402)은상기 AppID와 상기 CSAppID를 이용하여 웹소켓 객체를 생성하고, open() API를 호출하여 상기 제1기기(400)의 웹소켓(404)에게 통신 채널의 생성을 요청한다. 마찬가지로, 상기 응용프로그램(402)의 open() API역시 상기 AppID와 상기 CSAppID를 전달한다.

[0043] 그러면, 424단계에서 상기 제1기기(400)의 웹소켓 서버(404)는 AppID에 상응하는 상기 제1기기(400)의 응용프로그램과, 상기 CSAppID에 상응하는 상기 제2기기(410)의 응용프로그램(412) 간에 통신 채널을 생성한다. 상기한 바와 같은 과정으로 통신 채널이 생성되면, 제2기기(410)의 응용프로그램은 상기 통신 채널을 통해 상기 제1기기(400)의 응용프로그램과 통신할 수 있다. 420단계 내지 424단계를 통해서 제1기기(400)와 제2기기(410)의 응용프로그램에서 웹소켓 서버에 연결하는 방법은 하기 <표3>과 같이 나타내어질 수 있다.

표 3

웹소켓 서버의 주소가 ws://192.168.1.52일 경우 웹소켓 연결 방법 (제1기기와 제2기기) ws://192.168.1.52/AppID/CSAppID

[0046] 이후, 상기 제2기기(410)의 응용프로그램(412)에서 상기 제1기기(400)의 응용프로그램(402)으로 전송할 데이터가 발생한 경우, 426단계에서 상기 제2기기(410)의 응용프로그램(412)은 send() API를 호출해 상기 발생한 데이터를 상기 웹소켓 서버(404)에게 전송한다. 그러면, 428단계에서 상기 웹소켓 서버(404)는 Event(on message) API를 호출하여 상기 제1기기(400)의 응용프로그램(402)에게 전달함에 따라 제1기기(400)의 응용프로그램(402)에 수신된 데이터가 있음을 알린다. 그러면, 430단계에서 상기 제1기기(400)의 응용프로그램(402)은 상기 수신된 데이터를 응용프로그램에 반영한다. 일 예로, 상기 응용프로그램(402)이 웹 응용프로그램인 경우, 430단계의 과정은 DOM(Document Object Model) Update를 통해 이루어진다. 한편, 상기 1기기(400)의 응용프로그램(402)에서 변경된 사항이 있는 경우, 432단계에서 send() API를 호출하여, 변경된 사항에 대한 정보를 상기 웹 소켓 서버(404)로 전달한다. 그러면, 434단계에서 상기 1기기(400)의 웹소켓 서버(404)는, 상기 제2기기와 앞서 설명한 과정을 통해서 생성된 상기 통신채널을 통해 상기 정보를 상기 제2기기의 응용프로그램(412)에게 전송한다. 이때, 상기 정보는 Event(on message) API를 통해서 전달된다. 436단계에서 상기 제2기기(410)의 응용프로그램(412)은 상기 정보를 수신하여, 430단계와 유사하게 자신의 응용프로그램에 반영한다.

[0047] 이후, 모든 통신이 완료된 경우, 438단계에서 상기 제2기기(410)의 응용프로그램(412)은 close() API를 호출하여 제1기기의 응용프로그램과의 통신채널의 종료 요청을 상기 웹소켓 서버(404)에게 전송한다. 마찬가지로, 440

단계에서 상기 제1기기(400)의 응용프로그램(410)역시 close() API를 호출하여 제1기기의 응용프로그램과의 통신채널의 종료 요청을 상기 웹소켓 서버(404)에게 전송한다. 그러면, 442단계에서 상기 웹소켓 서버(404)는 상기 제1기기(400)의 응용프로그램(402)과 상기 제2기기(410)의 응용프로그램(412)간의 연결을 종료한다.

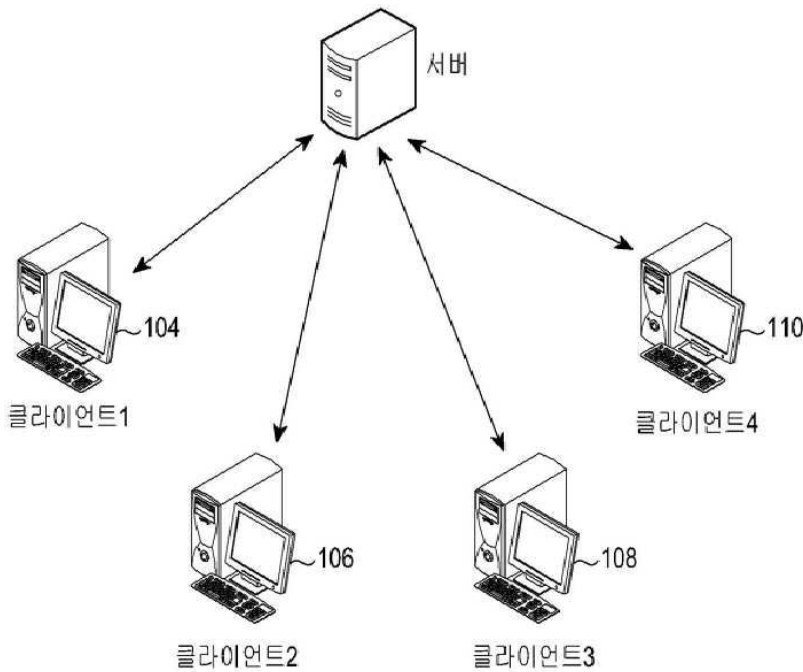
- [0049] 도 5는 본 발명의 실시 예에 따른 제1기기의 동작 흐름도이다.
- [0050] 도 5를 참조하면, 500단계에서 제1기기는 HTML 페이지를 수신한다. 상기 HTML 페이지는 상기 제1기기의 응용프로그램을 지시하는 응용프로그램 ID를 포함한다. 그러면, 502단계에서 상기 제1기기는 상기 HTML 페이지의 API 호출에 따라 상기 응용프로그램 ID를 이용하여 상기 웹소켓 서버를 구동시킨다.
- [0051] 그리고, 504단계에서 상기 제1기기의 응용프로그램은 네트워크에서 가용한 제2기기를 검색한다. 그리고, 506단계에서 상기 제1기기의 응용프로그램은, 상기 검색된 제2기기로 상기 제2기기의 응용프로그램 구동 정보를 송신한 후, 상기 제1기기는, 508단계에서 사용자 입력을 요구하는 HTML 페이지를 자신의 디스플레이 화면에 표시한다. 510단계에서 상기 제1기기는 사용자의 입력을 수신하고, 512단계에서 상기 사용자의 입력에 의해 제2기기의 응용프로그램으로 전송할 데이터의 발생 유무를 확인한다. 상기 확인 결과, 데이터가 발생하지 않은 경우, 516단계로 진행한다.
- [0052] 상기 확인 결과, 상기 데이터가 발생한 경우, 514단계에서 상기 제1기기는 502단계에서 구동시킨 웹소켓 서버로 상기 데이터를 송신한다. 그리고, 516단계로 진행하여, 상기 제1기기는 상기 웹소켓 서버로부터 수신할 데이터가 있는 지 여부를 확인한다. 상기 확인 결과, 수신할 데이터가 존재하지 않는 경우, 508단계로 진행하여, 다른 사용자 입력을 대기할 수 있다.
- [0053] 상기 확인 결과, 수신할 데이터가 발생한 경우, 518단계에서 상기 제1기기는 상기 웹소켓 서버를 통해서 데이터를 수신한다. 그리고, 520단계에서 상기 제1기기는 상기 수신된 데이터로부터 획득한 응용프로그램의 ID가 상기 제1기기의 응용프로그램 ID인지 판단한다. 상기 판단 결과, 상기 데이터로부터 획득한 응용프로그램 ID가 상기 제1기기의 응용프로그램 ID인 경우, 522단계에서 상기 제1기기는 DOM을 업데이트하여 상기 수신된 데이터를 반영한 HTML을 508단계에서 자신의 화면을 통해서 표시한다.
- [0054] 도 6은 본 발명의 실시 예에 따른 제2기기의 동작 흐름도이다.
- [0055] 도 6을 참조하면, 일 예로, 상기 제2기기는 600단계에서 도5의 506단계에 의해서 상기 제1기기가 송신한 정보로서 HTML 페이지를 수신한다. 여기서는 설명의 편의상, 상기 제2기기의 응용프로그램을 웹 응용프로그램으로 가정하여 HTML 페이지를 수신하는 것으로 설명하였다. 그러나, 다른 예로, 상기 제2기기가 600단계에서 상기 제1기기를 통해서 수신된 정보가 제2기기의 네이티브 응용프로그램을 가리키는 경우, 해당 네이티브 응용프로그램을 구동하게 된다.
- [0056] 그리고, 602단계에서 상기 제2기기는 상기 HTML로부터 획득한 제1기기의 응용프로그램 ID와 제2기기의 응용프로그램ID를 이용하여 상기 제1기기의 웹소켓 서버와의 연결을 호출한다.
- [0057] 604단계에서 상기 제2기기는 상기 제1기기의 응용프로그램 ID와 연동할 사용자 입력을 요구하는 HTML 페이지를 자신의 화면에 표시한다. 606단계에서 상기 제2기기는 사용자의 입력의 수신을 감지하면, 608단계에서 상기 사용자의 입력에 따라 제1기기로 전송할 데이터가 존재하는 지 여부를 확인한다. 상기 확인 결과, 전송할 데이터가 존재하지 않는 경우, 상기 제2기기는 612단계로 진행한다.
- [0058] 상기 확인 결과, 상기 사용자의 입력에 따라 상기 제1기기로 전송할 데이터가 존재하는 경우, 610단계에서 상기 제2기기는 602단계에서 연결한 상기 제1기기의 웹소켓 서버를 통해서 데이터를 송신한다.
- [0059] 한편, 612단계에서 상기 제2기기는 상기 사용자 입력에 따라 수신할 데이터가 있는 경우, 614단계에서 해당 데이터를 수신한다. 그리고, 616단계에서 상기 제2기기는 상기 수신된 데이터를 반영하여 DOM을 업데이트한다. 이후, 604단계로 복귀하여 업데이트 결과를 기반으로 HTML을 재구성하여 자신의 화면에 표시한다.
- [0060] 한편, 상기 612단계에서의 확인 결과, 수신할 데이터가 존재하지 않으면, 상기 제2기기는 상기 604단계로 복귀하여 다른 사용자 입력을 대기할 수 있다.
- [0062] 도 7은 본 발명이 실제 적용된 실시 예를 도시한 도면이다.

[0063] 도 7을 참조하면, 일 예로, 사용자가 제1기기(700)로 방송 콘텐츠를 수신하면서, 제2기기(705)를 통해 상기 제1기기를 제어하는 경우를 가정하자. 구체적인 예로, 상기 사용자는 사용자 입력을 위한 조작이 편리한 상기 제2기기(705)를 이용하여 상기 제1기기(700)에게 리모트 콘트롤, 문자 입력 등을 위한 사용자 입력을 수행할 수 있다. 또한, 상기 제1기기(700)의 방송 콘텐츠에 따라 디스플레이되는 화면을 선택하거나 투표 등을 상기 제2기기(705)를 이용하여 실시하는 등의 다양한 실시 예가 가능하게 된다.

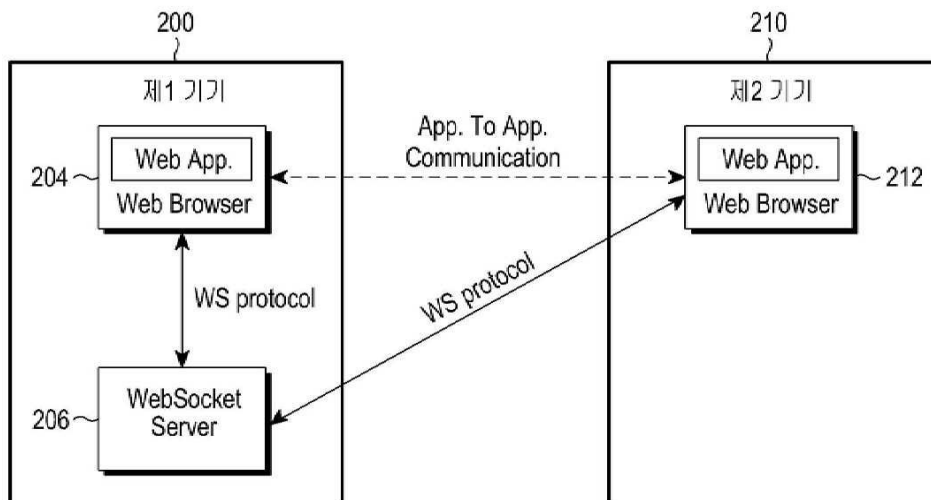
[0064] 한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 안되며 후술하는 특허 청구의 범위뿐만 아니라 이 특허 청구의 범위와 균등한 것들에 의해 정해져야 한다.

도면

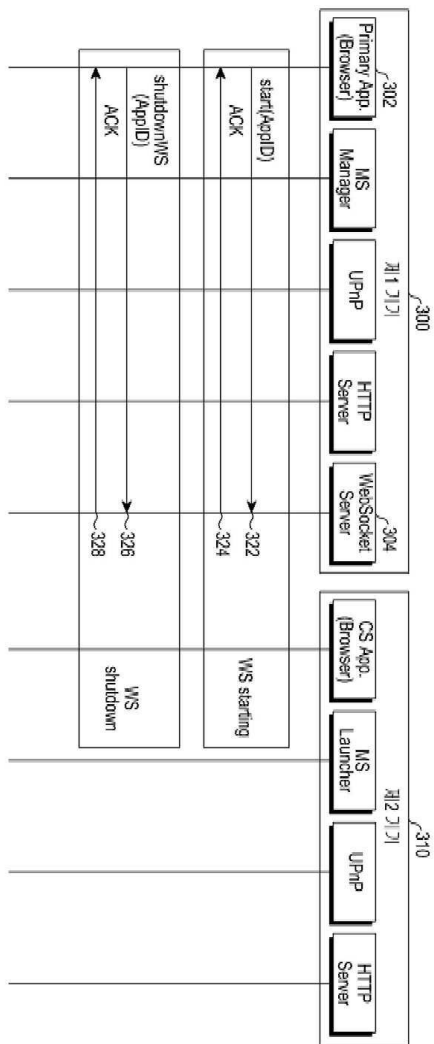
도면1



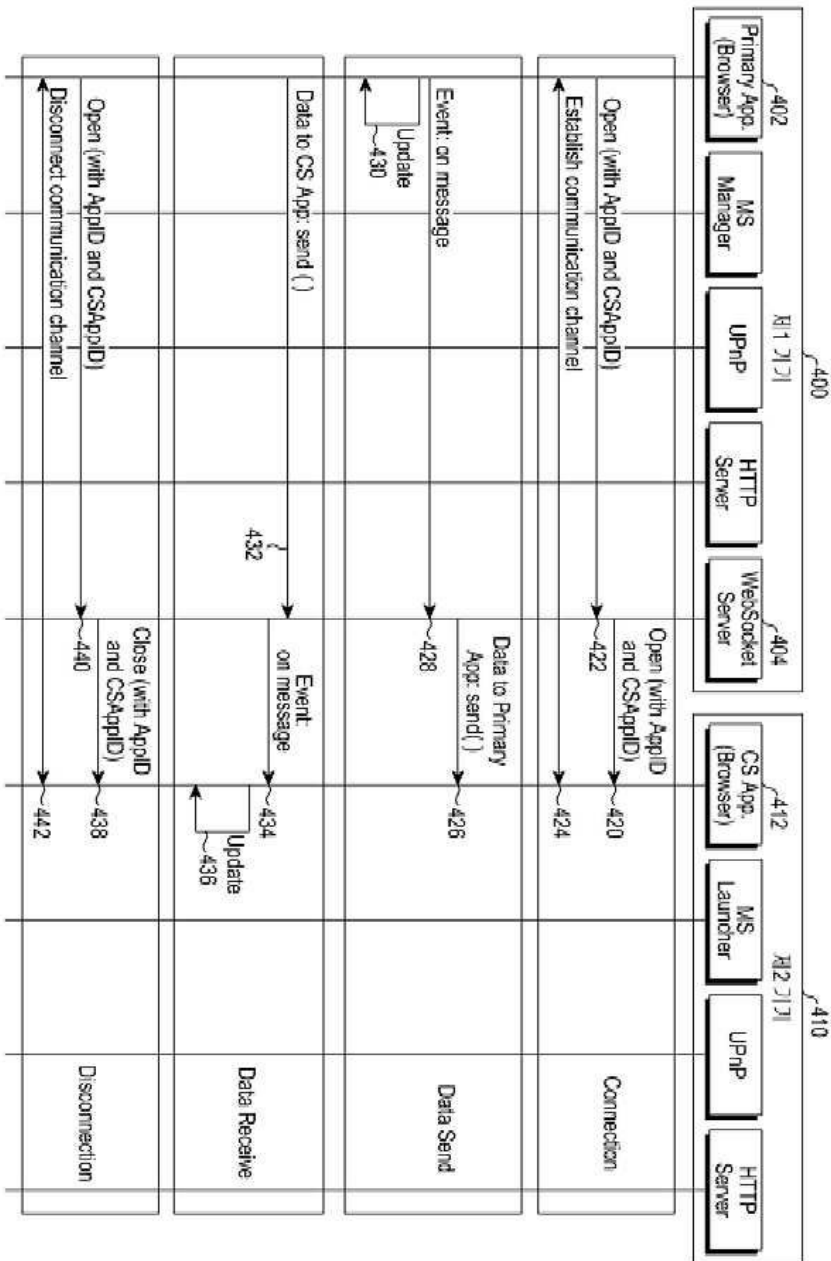
도면2



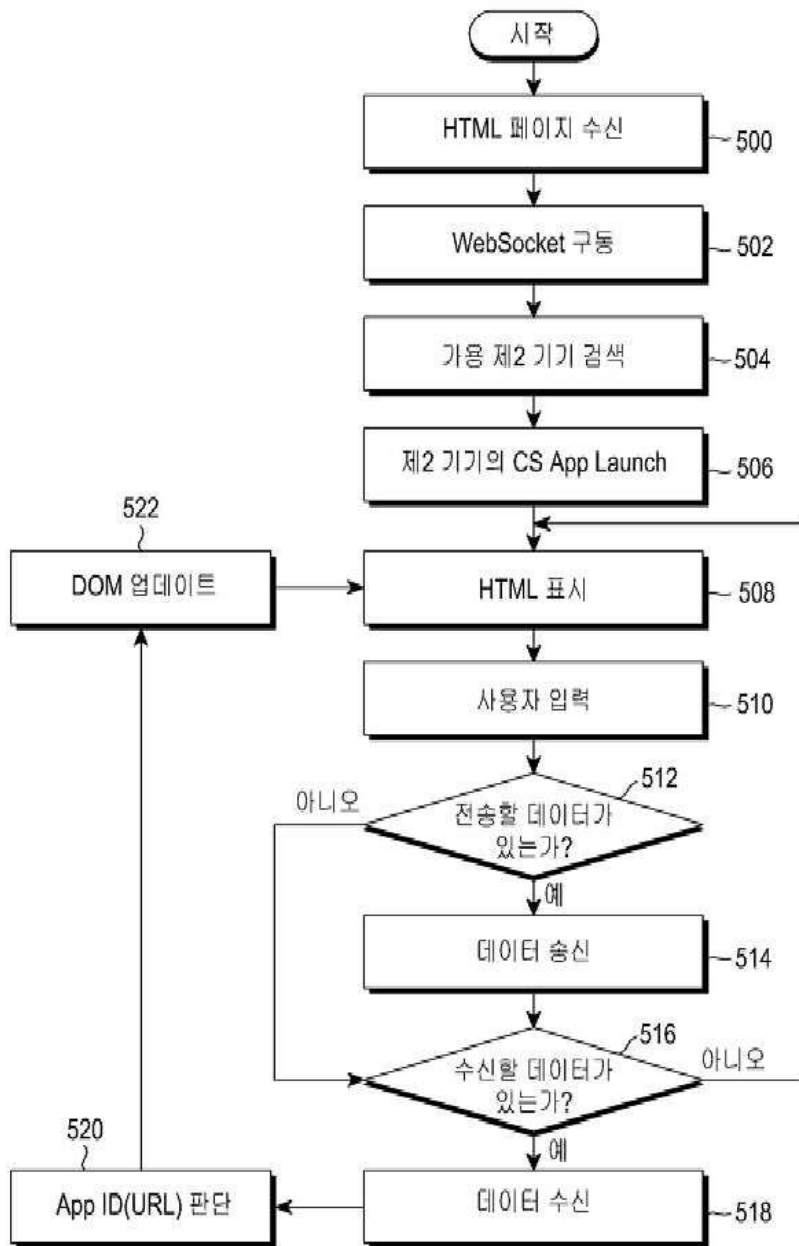
도면3



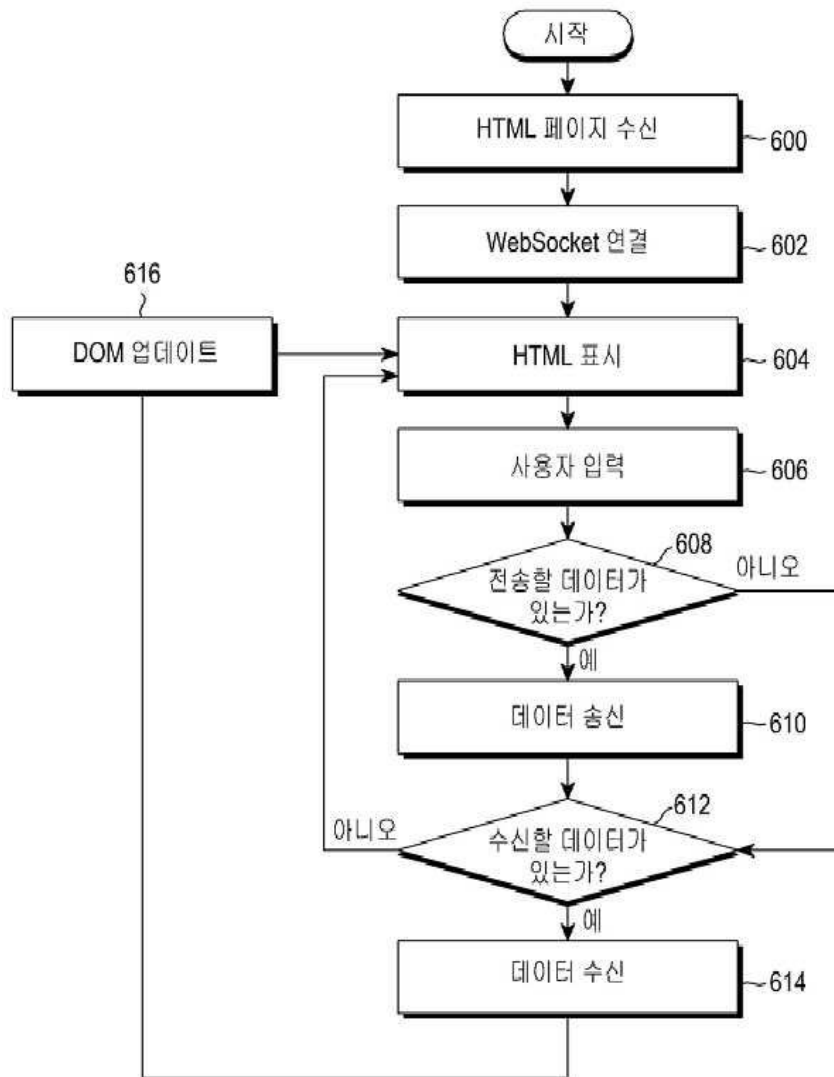
도면4



도면5



도면6



도면7

