



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년04월24일
(11) 등록번호 10-0893305
(24) 등록일자 2009년04월07일

(51) Int. Cl.
G06F 17/00 (2006.01) G06F 15/16 (2006.01)
(21) 출원번호 10-2007-0000004
(22) 출원일자 2007년01월01일
심사청구일자 2008년03월06일
(65) 공개번호 10-2008-0080424
(43) 공개일자 2008년09월04일
(56) 선행기술조사문헌
논문1: 한국정보처리학회*
KR1020060126729 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
주식회사 로직플랜트
서울시 서초구 방배동 474-15 화산빌딩
(137-060)
(72) 발명자
심재범
서울 서초구 방배동 988-1 신동아아파트 1-1311
(74) 대리인
김익수

전체 청구항 수 : 총 11 항

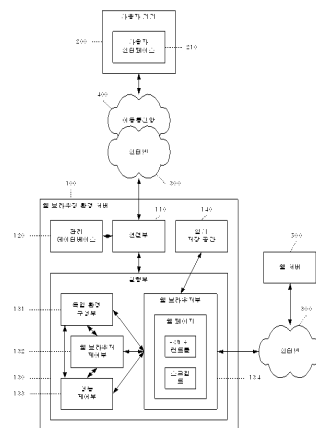
심사관 : 박성우

(54) 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징환경 제공 시스템과 그 제어 방법

(57) 요약

본 발명은 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템과 그 제어 방법에 관한 것이다. 본 발명에 따른 다중 사용자용 웹 브라우징 환경 제공 시스템은 웹 페이지의 열람과 웹 브라우징의 제어를 위한 사용자 인터페이스가 표시되는 사용자 기기, 그리고 다중 사용자들을 위해 각 사용자 별로 독립적인 웹 브라우징 환경을 효율적으로 생성, 관리하고 그 상태를 사용자 기기에 통보하며, 웹 페이지를 렌더링하고 웹 페이지 내의 동적인 콘텐츠를 실행하며, 렌더링된 웹 페이지 화면을 사용자 기기에 전송하고, 사용자 기기에서 요청한 동작이 웹 브라우징 환경에서 수행되도록 제어하며, 웹 브라우징 동작을 통제함으로써 성능 관련 자원을 절약하여 다수의 사용자를 지원하는 웹 브라우징 환경 서버를 구비한다. 이에 의해, 하나의 웹 브라우징 환경 서버로 다수의 사용자가 웹 브라우징의 주요 도구인 PC보다 사양이 낮거나 PC와의 호환성이 떨어지는 사용자 기기에서도 PC와 같은 수준의 웹 브라우징 작업을 수행할 수 있다.

대표도 - 도1



특허청구의 범위

청구항 1

웹 페이지의 열람과 웹 브라우징의 제어를 위한 사용자 인터페이스가 표시되는 사용자 기기; 및

상기 사용자 기기들과 네트워크를 통해 연결되어 상기 사용자 기기별로 독립적인 웹 브라우징 환경을 생성, 관리하고, 상기 사용자 기기에서 요청한 동작이 상기 웹 브라우징 환경에서 수행되도록 제어하여, 다수의 사용자 기기에 대해 웹 브라우징 환경을 제공하는 웹 브라우징 환경 서버를 포함하며,

상기 웹 브라우징 환경 서버는, 상기 사용자 기기의 사용자 인증을 위한 정보를 저장 및 관리하는 관리 데이터 베이스;

상기 관리 데이터베이스의 동작을 제어하는 연결부; 및

웹 페이지를 렌더링하고 상기 웹 페이지 내의 동적인 콘텐츠를 실행하는 웹 브라우저부; 상기 사용자 기기로부터 명령을 전달받아 상기 웹 브라우저부를 제어하고, 상기 웹 브라우저부의 상태 변경을 감지한 결과와 상기 웹 브라우저부가 렌더링하는 웹 페이지 화면을 상기 사용자 기기에 전달하는 웹 브라우저 제어부; 상기 사용자 기기별로 독립적인 웹 브라우징 환경을 구성하는 독립 환경 구성부; 및 웹 페이지 렌더링 빈도의 감소, 웹 페이지의 동적 요소를 웹 페이지 사용 여부에 따라 선별적으로 정지, 웹 브라우징 관련 동작의 동시 수행 개수를 제어 중 적어도 하나를 이용하여 웹 브라우징 동작을 통제함으로써 성능 관련 자원을 절약하도록 하는 성능 제어부를 구비하는 실행부;를 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템.

청구항 2

삭제

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 사용자 기기에 표시되는 상기 사용자 인터페이스는,

상기 웹 브라우징 환경 서버가 제공하는 웹 페이지 화면, 웹 브라우징을 제어하기 위한 메뉴, 및 웹 브라우징 환경 전반을 제어하기 위한 공통 메뉴 중 적어도 하나를 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템.

청구항 5

네트워크에 접속되어 상호 통신이 가능한 사용자 기기, 및 웹 페이지를 렌더링하고 제어하며, 웹 브라우저부를 구비하고 윈도우즈 호환 운영체제를 사용하는 웹 브라우징 환경 서버를 포함하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법에 있어서,

(a) 상기 사용자 기기별로 독립적으로 상기 웹 브라우징 환경 서버에 로그인해서 다른 사용자 기기와 격리된 독립적인 윈도우즈 데스크탑에서 웹 브라우징 기능을 수행하는 프로세스를 실행하고, 상기 사용자 기기별 프로세스가 처리하는 웹 페이지와 상기 웹 페이지 내부에서 실행되는 ActiveX 컨트롤을 포함하는 동적 요소가 상기 사용자 기기별로 화면에 표시되고 상기 사용자 기기별로 직접 제어될 수 있도록 상기 사용자 기기별 프로세스 단위로 소정 윈도우즈 API를 후킹하여 처리하는 과정을 이용하여, 상기 웹 브라우징 환경 서버에 상기 사용자 기기별로 독립적인 웹 브라우징 환경을 생성하는 단계;

(b) 상기 사용자 기기에 상기 웹 브라우징 환경을 위한 사용자 인터페이스를 표시하는 단계;

(c) 상기 사용자 기기에서 상기 사용자 인터페이스를 조회하여 실행할 기능을 선택하고, 상기 선택한 기능을 상기 네트워크를 통해 상기 웹 브라우징 환경 서버에 전송하는 단계; 및

(d) 상기 웹 브라우징 환경 서버에서 상기 선택된 기능에 대응되는 동작이 수행되도록 상기 웹 브라우저부를 제어하여, 상기 웹 브라우저부가 상기 선택된 기능에 대응하는 동작을 수행하고, 상기 사용자 기기에서 새로운 웹 페이지를 생성하거나 웹 페이지 전환을 하면서 기존 웹 페이지를 사용하지 않게 되는 경우, 대응하는 기존 웹 페이지의 동적 요소만을 선별적으로 정지시키는 단계;를 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 6

제5항에 있어서,

상기 (a) 단계 전에, 상기 사용자 기기가 상기 웹 브라우징 환경 서버에 인증을 요청하여 인증을 받는 단계;를 더 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 7

제5항에 있어서,

상기 사용자 인터페이스에는, 상기 웹 브라우징 환경 서버가 제공하는 웹 페이지 화면, 웹 브라우징을 제어하기 위한 메뉴, 및 웹 브라우징 환경 전반을 제어하기 위한 공통 메뉴 중 적어도 하나를 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 8

제5항에 있어서,

상기 (a) 단계에서 상기 웹 브라우징 환경의 생성시, 모든 사용자 기기는 단일한 운영체제 사용자 계정을 사용하며, 상기 사용자 계정은 서비스 시작 시 이미 로그인한 상태로, 모든 사용자 기기에 대해 공통의 윈도우즈 데스크탑 화면을 제공하도록 한 상태에서, 상기 윈도우즈 데스크탑 화면상에 각각 별도의 프로세스로 실행하여, 하나의 화면, 키보드, 및 마우스를 공유하도록 제어하고,

상기 사용자 기기별 프로세스 단위로 주요 윈도우즈 API를 후킹해서 내부적으로 처리하고, 특정 윈도우 메시지를 발생시키고, 상기 윈도우 메시지 디스패치 시 메시지의 내용에 따라 내부 정보를 변경함으로써, 각 프로세스 단위로 독립적인 마우스와 키보드 상태를 가지는 것처럼 인지되도록 제어하며,

운영체제의 사용자 폴더 관련 환경 변수를 각 사용자 기기별로 별도로 지정하거나, 파일 읽기 및 저장 관련 윈도우즈 API를 후킹해서 내부적으로 처리하여 각 사용자 기기별로 별도로 쿠키와 임시 인터넷 파일을 관리하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 9

제8항에 있어서,

상기 윈도우즈 데스크탑 화면을 제공하도록 한 상태에서 상기 사용자 기기별 프로세스의 웹 브라우저 윈도우를 상기 윈도우즈 데스크탑 화면에 위치시키되, 상기 윈도우즈 데스크탑 화면 위를 빈 윈도우로 완전히 가려서, 상기 웹 브라우저부와 상기 ActiveX 컨트롤들이 필요하지 않을 때에도 자동으로 윈도우 내용을 다시 렌더링하는 것을 방지하고,

상기 사용자 기기에 화면 내용을 전송해야 할 시점에만 상기 웹 브라우저부와 상기 ActiveX 컨트롤들이 상기 윈도우 내용을 렌더링하는 단계;를 더 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 10

제5항에 있어서,

상기 (c) 단계에서, 상기 사용자 기기에서 일정 시간 동안 상기 실행할 기능이 선택되지 않거나, 상기 사용자 기기의 사용자 인터페이스 구성상 기존 웹 페이지 화면을 사용하지 않게 되는 경우, 대응하는 웹 페이지의 동적 요소들을 정지시키고,

상기 사용자 기기에서 다시 상기 실행할 기능이 선택되거나 상기 사용하지 않게된 웹 페이지 화면을 다시 사용하게 되면 상기 정지된 웹 페이지의 동적 요소들이 다시 실행되도록 하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 11

삭제

청구항 12

제10항에 있어서,

상기 동적 요소들의 정지는,

SetTimer와 KillTimer 윈도우즈 API를 후킹해서 웹 페이지의 동적 요소들이 등록하는 타이머들을 선별적으로 중지 및 재생하는 방법, 및

웹 브라우저나 ActiveX 컨트롤, 스크립트 실행기, 및 스크립트 자체에서 동작을 중지 및 재생하는 인터페이스를 제공하는 경우, 상기 인터페이스를 사용하는 방법 중 적어도 하나의 방법을 사용하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 13

제12항에 있어서,

상기 타이머들의 선별적 중지 및 재생은, 웹 페이지를 각각 별도의 관리하는 스레드 단위로 타이머를 구분하는 방법, 웹 브라우저와 웹 페이지 내에서 동작하는 자체 구현된 스크립트 실행기가 타이머 요청을 할 때 웹 페이지별로 타이머를 별도 관리하는 방법, 및 웹 브라우저부가 ITimer와 이와 관련된 COM 인터페이스를 제공하여, 웹 페이지 내에서 대응하는 인터페이스를 사용해서 타이머를 요청하는 동적 요소들을 웹 페이지별로 구분하는 방법 중 적어도 하나의 방법을 사용하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

청구항 14

제5항에 있어서,

상기 (d) 단계에서, 상기 웹 브라우저가 상기 선택된 기능에 대응하는 동작을 수행하는 경우나 상기 선별적으로 정지된 동적 요소들이 타이머를 포함하는 이벤트에 의해 자체적으로 동작을 수행하는 경우, 상기 웹 브라우징 환경 서버에서 동작하는 프로세스들이 전역적인 운영체제상의 동기화 객체를 획득한 상태에서만 동작하도록 하여, 소정 시간 동안에 일정 개수의 프로세스만이 실행되도록 제어하는 단계;를 더 포함하는 것을 특징으로 하는 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<9> 본 발명은 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템과 그 제어 방법에 관한 것으로, 더욱 상세하게는 하나의 웹 브라우징 환경 서버로 다수의 사용자가 웹 브라우징의 주요 도구인 PC보다 사양이 낮거나 PC와의 호환성이 떨어지는 사용자 기기에서도 PC와 같은 수준의 웹 브라우징 작업을 수행할 수 있는 웹 브라우징 환경 제공 시스템과 그 제어 방법에 관한 것이다.

<10> 이하에서는 현재 널리 사용되는 사용자 기기 중 PC보다 사양이 낮고 PC와의 호환성이 떨어지는 대표적인 기기인 휴대폰을 예로 들어 현재의 문제점을 서술한다. 무선 인터넷이 활성화되면서, 휴대폰은 임의의 장소에서 사용할 수 있는 정보 단말기로서의 위치를 굳히게 되었다. 그에 따라, 인터넷의 주요 사용처인 웹 브라우징에도 휴대폰을 사용하고자 하는 수요가 커지고 있다. 이 수요를 만족시키기 위해 이동통신 업계에서는 휴대폰에서도 기

존 웹 브라우저의 주요 도구인 PC와 같은 수준의 웹 브라우저를 수행한다는 의미의 ‘풀 브라우저’이라는 용어를 만들고, 그 환경을 구현하고자 노력하고 있다.

- <11> 그러나 현재 휴대폰에서 ‘풀 브라우저’를 실현하는 데에는 여러 가지 문제점들이 존재하는데, 그 중 본 특허와 관련된 문제들은 다음과 같다.
- <12> 첫째, 현재 웹 페이지들은 PC 수준의 성능을 가지는 기기에서 표시되며 동작하는 것을 전제로 해서 만들어져 있으므로, PC에 비해 성능이 낮은 휴대폰에서는 제대로 표현되기 어렵다. 웹 페이지들은 사용자의 주의를 끌기 위해 플래시나 기타 동적인 화면 구성을 통해 화려한 화면을 보여주는데, 이와 같이 빠른 속도로 변경되는 화려한 화면을 적절히 표시하기에는 휴대폰의 성능이 부족하며, 휴대폰에서 웹 페이지 화면을 렌더링하는 구성의 경우 이 문제에 대한 별다른 대책이 없다.
- <13> 둘째로, 현재 웹 페이지들에는 플래시 콘텐츠나 자바스크립트 등, 실제 웹 페이지가 동작하는 시스템의 운영체제나 CPU 등의 종류에 영향을 받지 않는 요소들 외에도, ActiveX 컨트롤과 같이 특정 운영체제와 CPU에서만 동작이 가능한 요소(ActiveX 컨트롤의 경우 운영체제는 마이크로소프트 윈도우즈이고, CPU는 인텔 x86 계열 및 그 호환 기종)가 존재해서, 운영체제와 CPU가 PC와 호환되지 않는 휴대폰에서는 해당 요소를 표시하거나 사용할 수 없다는 문제가 있다. 대한민국의 경우 주요 포털 사이트와 전자상거래 사이트가 상기 요소들을 빈번히 사용하므로, 상기 요소들을 지원하지 않는 경우 정상적인 웹 브라우징 작업이 어렵다. 이 문제를 해결하기 위해 PC 환경을 휴대폰에 모사해서 구현하는 방법을 고려해볼 수 있으나 휴대폰의 사양이 낮아 실용성이 없고, 중간 단계의 서버에서 상기 요소들을 일부 실행하는 환경을 구축할 수 있으나 휴대폰에서 렌더링하는 나머지 콘텐츠와의 연동이 어려울 수 있다. 현재는 이런 요소들의 지원을 단순히 포기하는 추세이다.
- <14> 상기 문제점들을 모두 해결한 ‘풀 브라우저’는 현재 존재하지 않으며, 상기 문제점들 중 일부분만을 해결한 제품들만이 출시되고 있어서, 사용자들은 휴대폰에서 만족할 만한 웹 브라우징 작업을 하고 있지 못한 실정이다.
- <15> 위에서는 휴대폰을 예로 들었지만, 그 외의 사용자 기기들도 상기 문제점들 중 하나 이상을 가진 경우가 많다.
- <16> 상기 문제점들을 해결하기 위해서는 웹 브라우징 작업을 PC와 사양이 같거나 더 높으며 PC와 완전한 호환성을 가지는 서버에서 수행하고, 사용자 기기에서는 웹 페이지의 열람과 웹 브라우징의 제어를 위한 사용자 인터페이스를 제공하는 정도로 호환성과 무관하며 높은 성능을 요구하지 않는 기능을 담당하도록 하는 구성을 가진 시스템이 필요하다. 또한 상기 구성을 위해서는 하나의 서버에서 여러 사용자의 웹 브라우징을 지원하는 것이 바람직하므로, 그것을 위한 웹 브라우징 효율화 방안을 모색해야 한다.

발명이 이루고자 하는 기술적 과제

- <17> 따라서, 본 발명의 목적은, 하나의 웹 브라우징 환경 서버로 다수의 사용자가 웹 브라우징의 주요 도구인 PC보다 사양이 낮거나 PC와의 호환성이 떨어지는 사용자 기기에서도 PC와 같은 수준의 웹 브라우징 작업을 수행할 수 있는 웹 브라우징 환경 제공 시스템과 그 제어 방법을 제공함에 있다.

발명의 구성 및 작용

- <18> 상기 목적을 달성하기 위한 본 발명에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템은, 웹 페이지의 열람과 웹 브라우징의 제어를 위한 사용자 인터페이스가 표시되는 사용자 기기, 그리고 다중 사용자들을 위해 각 사용자 별로 독립적인 웹 브라우징 환경을 효율적으로 생성, 관리하고 그 상태를 사용자 기기에 통보하며, 웹 페이지를 렌더링하고 웹 페이지 내의 동적인 콘텐츠를 실행하며, 렌더링된 웹 페이지 화면을 사용자 기기에 전송하고, 사용자 기기에서 요청한 동작이 웹 브라우징 환경에서 수행되도록 제어하며, 웹 브라우징 동작을 통제함으로써 성능 관련 자원을 절약하여 다수의 사용자를 지원하는 웹 브라우징 환경 서버를 포함한다.
- <19> 상기 웹 브라우징 환경 서버는, 사용자 기기가 네트워크를 통해 접속해 올 때 해당 접속을 받아들이는 연결부, 사용자 인증을 위한 정보를 저장하는 관리 데이터베이스, 실제 웹 브라우징 기능을 수행하는 실행부, 웹 브라우징을 수행할 때 발생하는 쿠키나 인터넷 임시 파일을 저장하기 위한 임시 저장 공간을 포함하는 것이 가능하다. 상기 실행부는 사용자 기기로부터 명령을 전달받아 웹 브라우저부를 제어하고 웹 브라우저부의 상태 변경을 감지하며 해당 감지 결과와 웹 브라우저부가 렌더링하는 웹 페이지 화면을 사용자 기기에 전달하는 웹 브라우저 제어부, 사용자별로 독립적인 웹 브라우징 환경을 구성하는 독립 환경 구성부, 웹 브라우징 동작을 통제함

으로써 성능 관련 자원을 절약하도록 하는 성능 제어부, 웹 페이지를 렌더링하고 제어하는 웹 브라우저부를 포함하는 것이 가능하다.

- <20> 상기 사용자 기기에 표시되는 상기 사용자 인터페이스에는, 상기 웹 브라우징 환경 서버의 웹 브라우저 제어부가 제공하는 웹 페이지 화면, 웹 브라우징을 제어하기 위한 메뉴, 웹 브라우징 환경 전반을 제어하기 위한 공통 메뉴 중 적어도 하나를 포함하는 것이 가능하다.
- <21> 한편, 본 발명에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어 방법은, 네트워크에 접속되어 상호 통신이 가능한 사용자 기기, 및 웹 브라우징 환경 서버를 구비하는 시스템에서, 상기 사용자 기기에 상기 웹 브라우징 환경 서버가 제공하는 웹 페이지 화면과 그 제어를 위한 사용자 인터페이스를 표시하는 단계, 상기 사용자 기기에서 상기 사용자 인터페이스를 사용하여 특정 기능을 선택하는 단계, 선택된 기능을 상기 사용자 기기에서 네트워크를 통해 상기 웹 브라우징 환경 서버에 전송하는 단계, 선택된 기능에 대응되는 동작이 수행되도록 상기 웹 브라우징 환경 서버가 웹 브라우징을 제어하는 단계, 상기 웹 브라우징 환경 서버가 웹 브라우징 상태 변화를 감지하는 단계, 및 상기 웹 브라우징 환경 서버가 감지된 상태 변화 정보를 네트워크를 통해 상기 사용자 기기에 통보하는 단계를 포함한다. 바람직하게는, 상기 사용자 기기가 상기 웹 브라우징 환경 서버에 인증을 요청하고, 이에 대한 인증을 받는 단계를 더 포함한다.
- <22> 또한, 본 발명에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 제어 방법은, 상기 사용자 기기로부터의 기능 선택이 없이도, 웹 브라우징 환경 서버가 웹 브라우징의 자동적인 상태 변화를 감지하는 단계, 및 웹 브라우징 환경 서버가 상기 감지 결과를 네트워크를 통해 상기 사용자 기기로 전달하는 단계를 더 포함한다.
- <23> 상기 사용자 인터페이스에는, 상기 웹 브라우징 환경 서버의 웹 브라우저 제어부가 제공하는 웹 페이지 화면, 웹 브라우징을 제어하기 위한 메뉴, 웹 브라우징 환경 전반을 제어하기 위한 공통 메뉴 중 적어도 하나의 메뉴 항목이 표시되는 것이 가능하다.
- <24> 본 발명에서는, 웹 브라우징 작업을 PC와 사양이 같거나 더 높으며 PC와 완전한 호환성을 가지는 웹 브라우징 환경 서버에서 수행하도록 하며, 다수의 사용자가 하나의 웹 브라우징 환경 서버에서 웹 브라우징을 하는 상황에서 개개의 사용자가 다른 사용자와 격리된 독립적인 웹 브라우징 환경을 가지도록 하는 방법, 하나의 웹 브라우징 환경 서버로 보다 많은 사용자가 웹 브라우징을 할 수 있도록 하기 위한 웹 브라우징 환경 서버 내의 웹 브라우징 효율화 방법을 제공한다.
- <25> 이하에서는 도면을 참조하여 본 발명을 보다 상세하게 설명한다.
- <26> 도 1은 본 발명의 일실시에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 구성을 설명하기 위한 도면이다.
- <27> 도 1을 참조하면, 본 발명 서비스를 제공하는 업체에 웹 브라우징 환경 서버(100)가 설치된다. 웹 브라우징 환경 서버(100)는 인터넷(300) 및 이동통신망(400)을 통해 사용자 기기(200)와 통신이 가능하다.
- <28> 웹 브라우징 기능은 웹 브라우징 환경 서버(100)가 담당하며, 사용자 기기(200)는 웹 브라우징 환경 서버(100)가 전달하는 웹 페이지 화면을 표시하고 사용자로부터 제어 입력을 받는 역할을 한다. 상기 역할 분담 방식에 의해 사용자 기기(200)는 성능과 호환성의 문제로부터 자유로워지며, 그에 따라 PC보다 사양이 낮거나 PC와의 호환성이 떨어지는 기기에서도 PC와 같은 수준의 웹 브라우징이 가능해진다.
- <29> 웹 브라우징 환경 서버(100)는 연결부(110), 관리 데이터베이스(120), 실행부(130), 임시 저장 공간(140)을 포함한다. 연결부(110)는 사용자 기기(200)가 네트워크를 통해 접속해 왔을 때 관리 데이터베이스(120)를 이용하여 사용자 인증을 수행하며, 각 사용자 별로 실행부(130)가 실행되도록 관리한다. 관리 데이터베이스(120)에는 사용자 인증을 위한 정보가 데이터베이스화 되어 저장된다. 실행부(130)는 각 사용자 별로 운영체제 상의 별도의 프로세스로 실행되어 사용자 기기(200)와 네트워크를 통해 상호 작용한다. 실행부(130)는 독립 환경 구성부(131), 웹 브라우저 제어부(132), 성능 제어부(133), 웹 브라우저부(134)를 포함한다. 실행부(130)가 사용자 기기(200)와 네트워크를 통해 연결되면, 독립 환경 구성부(131)는 다른 사용자들이 연결된 실행부(130)들에 대해 독립적인 웹 브라우징 환경을 구축해서, 사용자가 자신만의 웹 브라우징 환경을 사용할 수 있도록 한다. 웹 브라우저 제어부(132)는 사용자 기기(200)로부터의 명령을 전달받아 웹 브라우저부(134)를 제어하여 웹 브라우징을 수행하고, 웹 브라우저부(134)의 상태 변화를 감지, 그 결과를 사용자 기기(200)로 전달하며, 또한 웹 브라우저부(134)가 렌더링하는 웹 페이지 화면을 사용자 기기(200)에 전달하여 사용자 인터페이스(210) 화면에 표시하도록 한다. 성능 제어부(133)는 웹 브라우저부(134)의 동작을 통제하여 웹 브라우징 환경 서버(100)의 성

능 관련 자원을 절약함으로써 하나의 웹 브라우징 환경 서버(100)에 보다 많은 사용자 기기(200)가 연결되어 웹 브라우징을 수행할 수 있도록 한다. 웹 브라우저부(134)는 웹 브라우저 제어부(132)의 제어를 받아 실제로 웹 브라우징을 수행하여, 웹 페이지를 렌더링하고 그 안에 포함된 ActiveX 컨트롤과 스크립트를 실행한다.

- <30> 사용자 기기(200)는 웹 브라우징 환경 서버(100)와 네트워크를 통해 연결되어 웹 브라우징 서비스를 받으며, 사용자 인터페이스(210)에 웹 브라우징 환경 서버(100)가 전달하는 웹 페이지 화면, 웹 브라우징을 제어하기 위한 메뉴, 웹 브라우징 환경 전반을 제어하기 위한 공통 메뉴 중 적어도 하나를 표시하고, 사용자의 입력을 받아 웹 브라우징 환경 서버(100)에 전달한다.
- <31> 이하에서는 실행부(130)의 독립 환경 구성부(131)가 각 사용자 별로 독립적인 웹 브라우징 환경을 구축하는 방법과, 성능 제어부(133)가 웹 브라우저부(134)의 동작을 통제하여 성능 관련 자원을 절약하는 방법을 상세히 설명한다.
- <32> 상기 구현 방법은 웹 브라우저부(134)의 구현 방법에 따라 약간 달라진다. 웹 브라우저부(134)의 구현 방법에는 다음 두 가지가 존재한다. 첫째는 마이크로소프트 윈도우즈 운영체제가 자체적으로 제공하는 웹 브라우저 컨트롤을 이용하는 방법이다. 이 방법에서는 실제 웹 브라우징 기능은 상기 웹 브라우저 컨트롤이 제공하고, 웹 브라우저부(134)는 상기 웹 브라우저 컨트롤의 인터페이스 함수의 호출이나, 독립 환경 구성부(131)를 통한 윈도우 메시지 전송 및 키보드와 마우스 시뮬레이션 등에 의해 명령을 전달하고 상태 변화를 전달받는다. 이 방법은 세계적으로 가장 많이 사용되는 인터넷 탐색기의 실제 웹 브라우징 기능을 담당하는 웹 브라우저 컨트롤을 사용하므로 웹 페이지 처리에 있어 가장 호환성이 높으며, 복잡한 소프트웨어인 웹 브라우저를 별도로 구현할 필요가 없다는 장점이 있다. 반면에, 웹 브라우저 컨트롤이 외부로 노출하는 인터페이스의 한계 상 성능 제어부(133)에서 성능을 완전히 제어하기가 어렵다는 단점이 있다. 둘째는 스크립트 실행기를 포함한 웹 브라우저 기능을 자체적으로 구현하는 방법이다. 이 방법은 상기 첫째 방법과 반대의 장단점이 있다.
- <33> 독립 환경 구성부(131)가 구축하는 각 사용자 별로 독립적인 웹 브라우징 환경이란, 사용자가 독립적으로 웹 브라우징 환경 서버(100)에 로그인해서 다른 사용자와 격리된 독립적인 윈도우즈 데스크탑에서 웹 브라우저를 실행하고, 웹 브라우저가 처리하는 웹 페이지와 그 내부에서 실행되는 ActiveX 컨트롤 등의 동적 요소가 실제 화면에 표시되고 사용자에게 의해 직접 제어되고 있다고 인지하도록 구성된 환경이다.
- <34> 이런 환경을 단순하게 제공하는 방법에는 다음과 같은 것들이 있다. 첫째는 웹 브라우징 환경 서버(100)에 사용자 수만큼의 실제 계정을 생성하고 사용자가 사용자 기기(200)를 사용해서 해당 계정으로 실제로 원격 로그인해서 사용하는 방법이다. 둘째는 웹 브라우징 환경 서버(100)를 사용자 기기마다 하나씩 대응시켜, 사용자가 실제로 하나의 PC를 원격으로 사용하는 방법이다. 셋째는 둘째 방법과 거의 같으나, 실제 서버를 대응시키지 않고, 웹 브라우징 환경 서버(100) 내에 가상 머신들을 구축해서, 마치 사용자가 둘째 방법과 마찬가지로의 환경에서 사용하는 것처럼 인지하게 하는 방법이다. 첫째 방법과 셋째 방법은 어떠한 성능 제어도 없이 각각의 사용자가 하나의 서버를 공유해서 사용하게 되므로 성능 관련 자원이 낭비되어 많은 사용자가 접속할 수 없으며, 계정 수나 운영체제 수에 의해 웹 브라우징 환경 서버(100)의 운영체제 라이선스 비용이 높아질 수 있다는 단점이 있다. 또한, 첫째 방법은 사용자의 로그인마다 생성되는 별도의 윈도우즈 세션 및 데스크탑에 의한 추가 부하가 존재하며, 운영체제가 정의한 원격 로그인 프로토콜을 사용자 기기에서 지원해야 하므로 구현 방법이 복잡해지며 사용자 기기에 성능 부담이 가해질 수 있다. 둘째 방법은 한 서버를 한 사용자가 사용하므로, 접속한 사용자 수만큼 서버가 필요하게 되어, 다사용자 환경을 구축하기가 사실상 불가능하다. 셋째 방법은 다사용자 환경을 구성할 수는 있으나, 자체적으로 운영 체제를 탑재한 실제 서버와 같은 형태의 가상 머신을 서버 내에서 구동하는 데 서버의 성능 관련 자원 중 상당 부분이 할애되어, 웹 브라우징 환경 서버(100)의 성능에 비해 만족할 만한 수의 사용자를 동시에 수용할 수 없다는 단점이 있으며, 가상 머신을 기존의 소프트웨어를 구매하지 않고 자체적으로 구현하는 경우 그 난이도 때문에 구현 비용이 대단히 높다는 단점이 있다. 이와 같이, 상기 세 가지 방법은 각각의 큰 단점 때문에 실용성이 없다.
- <35> 상기 방법들과 같은 문제가 없는 실용적인 환경을 구성하기 위해 웹 브라우징 환경 서버(100)는 다음과 같이 환경을 구성한다. 첫째, 윈도우즈 XP나 그와 호환되는 운영체제를 웹 브라우징 환경 서버(100)의 운영체제로 사용한다. 둘째, 모든 사용자는 단일한 운영체제 사용자 계정을 사용하며, 해당 사용자 계정은 서비스 시작 시 이미 로그인한 상태로, 모든 사용자에게 대해 공통의 윈도우즈 데스크탑 화면을 제공한다. 셋째, 실행부(130)는 상기 단일한 계정의 공통의 윈도우즈 데스크탑 화면 상에 각 사용자 별로 각각 별도의 프로세스로 실행되며, 그에 따라 하나의 화면을 공유하고, 키보드와 마우스 역시 공유한다. 넷째, 독립 환경 구성부(131)는 각 실행부(130) 프로세스 단위로 주요 윈도우즈 API를 후킹해서 내부적으로 처리하고, 특정 윈도우 메시지를 발생시키고,

윈도우 메시지 디스패치 시 메시지의 내용에 따라 내부 정보를 적절히 변경함으로써, 각 실행부(130) 프로세스 단위로 독립적인 마우스와 키보드 상태를 가지는 것처럼 인지되도록 한다. 다섯째, 웹 브라우저 제어부는 웹 브라우저부의 화면을 캡처할 때 기존 화면 캡처 소프트웨어들처럼 실제 윈도우즈 데스크탑 화면을 캡처하지 않고, 개별 웹 브라우저와 관련 윈도우에 대해 특정 윈도우즈 API를 사용해서 화면을 내부 버퍼에 렌더링하도록 함으로써, 실제 화면에 나타나지 않고도 화면을 캡처할 수 있도록 한다. 이렇게 함으로써 다수의 실행부(130)가 소유한 웹 브라우저 윈도우나 관련 윈도우들이 화면에 겹쳐 있어도 각각의 실행부(130)가 정상적으로 자신의 화면을 캡처할 수 있게 된다. 여섯째, 마이크로소프트 윈도우즈 운영체제가 자체적으로 제공하는 웹 브라우저 컨트롤을 사용해서 웹 브라우저부(134)를 구현하는 경우, 운영체제의 사용자 폴더 관련 환경 변수를 각 실행부(130) 별로 별도로 지정하거나, 독립 환경 구성부(131)에서 파일 읽기 및 저장 관련 윈도우즈 API를 후킹해서 내부적으로 처리함으로써, 각 실행부(130) 별로 독립적으로 쿠키와 임시 인터넷 파일을 관리할 수 있도록 한다. 이와 같이 환경을 구성함으로써 각 사용자는 웹 브라우징 환경 서버(100)의 성능 관련 자원을 최소한으로 사용하면서 다른 사용자와 격리된 독립적인 웹 브라우징 환경을 제공받게 된다.

<36> 이하에서는 독립 환경 구성부(131)가 각 실행부(130) 프로세스 단위로 독립적인 마우스와 키보드 상태를 가지는 것처럼 인지되도록 하는 방법을 상세히 설명한다.

<37> 먼저, 윈도우즈 API를 후킹하고, 특정 윈도우 메시지를 발생시키고, 윈도우 메시지 디스패치 시 윈도우 메시지의 내용에 따라 내부 정보를 적절히 변경해야 하는 이유는 다음과 같다. 첫째, 마이크로소프트 윈도우즈 운영체제가 자체적으로 제공하는 웹 브라우저 컨트롤을 사용해서 웹 브라우저부(134)를 구현하는 경우, 웹 브라우저 컨트롤이 마우스와 키보드 상태를 윈도우즈 API와 디스패치 되는 윈도우 메시지를 통해 인지하기 때문이다. 예를 들어, 화면을 렌더링하는 경우 윈도우 정보 관련 윈도우즈 API를 사용해서 현재 마우스 커서 위치와 해당 마우스 커서 위치에서 Z-오더 상으로 가장 위에 있는 윈도우를 알아내서 그 윈도우가 자신의 것이면 현재 마우스 커서가 위치한 곳에 대해 하이퍼링크를 하이라이트 하는 등의 특수 처리를 하며, 사용자가 마우스 커서를 움직이는 경우 그것을 마우스 커서 움직임 메시지를 통해 인지한다. 둘째, 웹 브라우저를 자체적으로 구현해서 내부적인 데이터 교환에 의해 상기 웹 브라우저의 행동을 처리하는 경우에도, 웹 페이지에 포함된 ActiveX 컨트롤들은 타사가 작성한 윈도우즈 응용 프로그램이므로, 여전히 상기와 같이 윈도우즈 API나 디스패치 되는 윈도우 메시지를 통해서 동작을 수행하기 때문이다. 따라서, 웹 브라우저나 ActiveX 컨트롤들이 마우스와 키보드 상태에 대한 정보를 운영체제에서 그대로 전달받는 경우, 하나의 윈도우즈 데스크탑에서 동시에 실행 중인 여러 실행부들은 공통의 정보를 받을 것이며, 한 실행부가 변경하는 상태는 다른 실행부들에 그대로 적용되어서 서로 독립적인 동작을 할 수 없게 되므로, 윈도우즈 API 후킹과 특정 윈도우 메시지 발생, 윈도우 메시지 디스패치 시 내부 정보 변경을 통해 독립적인 상태를 유지해야 한다.

<38> 윈도우즈 API 후킹을 위해서는 기존에 공개된 DLL 임포트 테이블 패치 방법이나 DLL 함수 시작점 명령어 변경 방법 등을 사용하면 된다. 이 방법들은 한 프로세스가 자기 프로세스만을 한정해서 윈도우즈 API 후킹을 수행할 수 있도록 하는데, 이 특징은 실행부(130) 프로세스 단위로 내부 정보를 별도로 관리해야 하는 독립 환경 구성부(131)의 상황에 있어 필수적이다. 이 방법들을 이용해서 DLL에 존재하는 윈도우즈 API 함수들을 내부 함수들로 대체해서, 내부적으로 관리하는 정보를 변경하고 전달한다.

<39> 마우스와 키보드 정보를 위해 후킹하는 윈도우즈 API 중 대표적인 것은 다음과 같다.

<40> GetCursorPos: 현재 마우스 커서 위치를 돌려준다. 내부 함수에서는 각 실행부에서 독립적으로 관리하는 가상적인 마우스 위치를 돌려준다.

<41> SetCursorPos: 현재 마우스 커서 위치를 지정한다. 내부 함수에서는 실제 마우스 커서 위치를 지정하지 않고, 각 실행부에서 독립적으로 관리하는 가상적인 마우스 위치를 지정한다.

<43> 삭제

<44> 삭제

<45> 삭제

- <46> 발생시키는 윈도우 메시지 중 대표적인 것은 다음과 같다.
- <47> WM_MOUSEMOVE: 사용자 기기에서 명령이 전달되거나, 후킹된 SetCursorPos API에서 마우스 커서 위치를 변경할 때, 내부적으로 해당 지점이 포함된 윈도우 중 Z-오더 상 가장 위에 있는 윈도우에 내부적으로 변경된 마우스 커서 위치를 전달한다.
- <48> WM_KEYDOWN, WM_KEYUP, WM_CHAR: 사용자 기기에서 전달된 키보드 입력을 실행하기 위해 발생시킨다.
- <49> 삭제
- <50> 삭제
- <51> 윈도우 메시지 디스패치 시 윈도우 메시지의 내용에 따라 내부 정보를 적절히 변경해야 하는 경우는 다음과 같다.
- <53> WM_KEYDOWN, WM_KEYUP: 윈도우즈 API에서 돌려줄 가상 키 입력 상태를 갱신한다.
- <54> 키 입력 상태 중 하나인 키보드 포커스는 윈도우즈 운영체제 자체가 각 스레드 별로 별도의 포커스 정보를 가지고 있도록 하므로 통상적인 경우 문제가 없으나, 서버에서 관리자가 직접 실제 마우스를 클릭해서 별도의 프로그램에 포커스를 부여하는 경우, 윈도우즈 운영체제의 동작 특성 상 각 실행부(130)의 스레드 별로 소유하고 있던 포커스가 사라지는 문제가 생길 수 있다. 이 경우를 위해, 사용자 기기에서 발생하는 키보드 입력 명령의 경우, 입력 기능을 현재 마우스 커서 위치의 윈도우에 한정시키고, 현재 마우스 위치의 윈도우에 SetFocus API를 통해 포커스를 다시 지정하거나 해당 윈도우에 마우스 클릭 메시지를 전달해서 포커스를 소유하도록 한 다음에 키를 입력하도록 한다. 키 입력을 웹 브라우저나 ActiveX 컨트롤들이 자체적으로 발생시키는 경우는 사실상 없으므로, 이 방법으로 포커스 관련 문제를 해결할 수 있다.
- <55> 독립 환경 구성부(131)는 각 실행부(130) 별로 독립적으로 쿠키와 임시 인터넷 파일을 관리할 수 있도록 하기 위해서 윈도우즈 API를 후킹할 수 있다. 이때 후킹하는 윈도우즈 API 중 대표적인 것으로는, CreateFileA, CreateFileW, DeleteFileA, DeleteFileW 등이 있으며, 이 함수들을 대체하는 내부 함수에서는 입력 파라미터의 파일 경로명을 각 실행부(130) 별 고유한 경로로 변경함으로써 파일을 별도의 경로에서 관리할 수 있도록 한다. 또한 윈도우즈 사용자 고유 경로를 나타내는 환경변수를 각 실행부(130) 별로 고유한 경로로 변경하는 방법이 존재하며, 마이크로소프트 윈도우즈 운영체제가 자체적으로 제공하는 웹 브라우저 컨트롤의 경우 이 방법으로 상기 쿠키와 임시 인터넷 파일을 각 실행부(130) 별 고유한 경로에서 관리할 수 있다.
- <56> 성능 제어부(133)는 웹 브라우저와 ActiveX 컨트롤들의 화면 표시와 이벤트를 통제하고, 전체 실행부들에서 CPU를 소모하는 동작들을 특정 시각에 일정 개수 이상 실행하지 못하도록 통제함으로써 성능 관련 자원을 절약한다.
- <57> 화면 표시의 경우, 웹 브라우저 윈도우를 윈도우즈 데스크탑에 위치시키기는 하되, 그 위를 빈 윈도우로 완전히 가려서, 웹 브라우저와 ActiveX 컨트롤들이 실행부(130)가 필요하지 않을 때에도 자동으로 윈도우 내용을 다시 렌더링하는 것을 막는다. 이것은 응용 프로그램들이 자신의 윈도우가 화면에서 가려진 상태에서는 윈도우 내용을 다시 렌더링하지 않는 특성을 이용한 것이다. 대신, 웹 브라우저 제어부(132)는 웹 페이지 화면을 캡처할 때 실제 윈도우즈 데스크탑 화면을 캡처하지 않고, 개별 웹 브라우저와 관련 윈도우에 대해 윈도우즈 API인 PrintWindow를 사용해서 화면을 내부 버퍼에 렌더링하도록 함으로써, 실제 화면에 나타나지 않고도 화면을 캡처할 수 있도록 한다. 이렇게 함으로써 실행부(130)는 주기적으로 사용자 기기에 화면 내용을 전송해야 할 시점에만 웹 브라우저와 ActiveX 컨트롤들이 윈도우 내용을 렌더링하게 할 수 있게 되어, 결과적으로 윈도우 내용을 렌더링하는 데에 소모되는 성능 관련 자원을 절약할 수 있게 된다. 현재 웹 브라우저에서 빈번하게 사용되는 플래시 ActiveX 컨트롤의 경우, 1초에 10번 이상 윈도우 내용을 다시 렌더링하는 경우가 있으므로, 실행부(130)의 화면 내용 전송 주기에 따라 상당량의 성능 관련 자원을 절약할 수 있게 된다.
- <58> 타이머는 ActiveX 컨트롤이나 스크립트 등 웹 페이지 내의 동적 요소들이 지속적으로 동작하는 데에 필요한 기본 요소이다. 상기 동적 요소들은 그 특성 상 CPU를 무한히 점유하며 실행되지 않고, 타이머를 적정 주기로 등록한 뒤 해당 타이머 이벤트를 받아서 주기적으로 동작하도록 구현된다. 그런데, 플래시 ActiveX 컨트롤이나 상당수의 스크립트의 경우, 수십에서 수백 밀리초 단위의 타이머를 등록하므로 초당 동작 횟수가 상당히

많으며, 웹 페이지가 화면에 보이는가, 따라서 사용자가 해당 웹 페이지를 보고 있는가의 여부와 무관하게 실행된다. 또한, 대표적인 포털 사이트들의 경우, 하나의 웹 페이지에 플래시 ActiveX 컨트롤과 타이머로 동작하는 스크립트들이 다수 존재해서, 결과적으로 1초에 수십회에서 100회 내외까지의 타이머 이벤트가 발생하게 된다. 따라서, 이 타이머 이벤트에 따른 동작들은 하나의 웹 브라우징 환경 서버(100)가 다수의 사용자를 수용하는 데 성능 상 심각한 장애 요소로 작용한다.

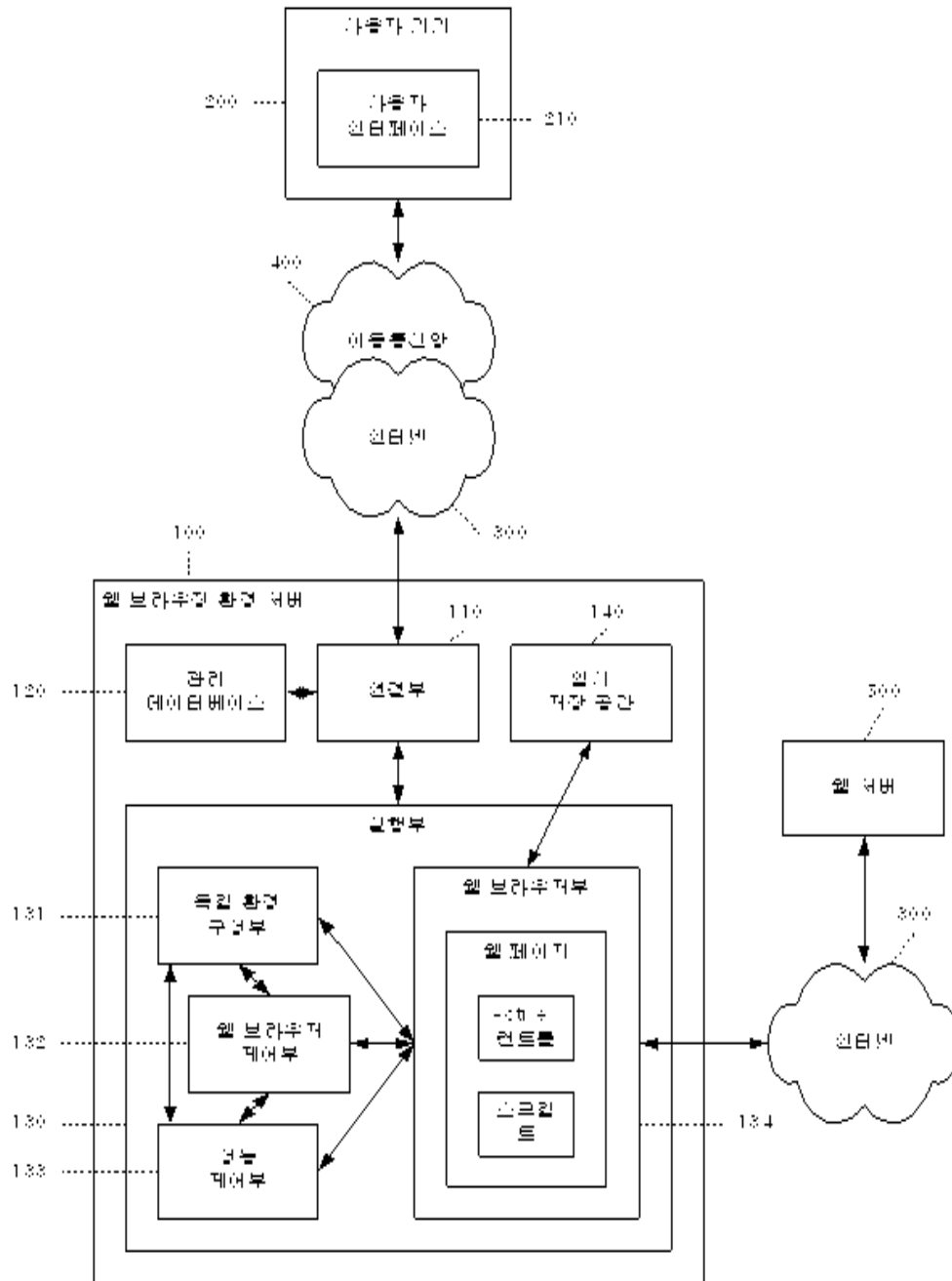
- <59> 이에 대해 성능 제어부(133)는 다음과 같은 방법으로 성능 관련 자원을 제어한다.
- <60> 먼저, SetTimer와 KillTimer 윈도우즈 API를 독립 환경 구성부(131)와 같은 방법으로 후킹해서, 등록되는 타이머들을 관리하면서 타이머를 선별적으로 중지시킨다. 타이머를 중지시키는 경우는 두 가지가 있다. 첫째로, 일정 기간 동안 사용자 기기(200)에서 전혀 입력이 없거나, 사용자 기기(200)의 사용자 인터페이스(210) 구성 상 웹 페이지 화면을 보지 않게 되는 경우가 생기면, 관리하던 타이머들을 내부적으로 모두 중지시킨다. 이렇게 함으로써 웹 페이지의 동적 요소들이 정지되며, 따라서 성능 관련 자원이 대폭 절약된다. 추후 사용자 기기(200)에서 입력이 전달되거나 다시 웹 페이지 화면을 보게 되면 타이머들을 재생시켜 웹 페이지의 동적 요소들이 다시 실행되도록 한다. 둘째, 사용자가 새로운 웹 페이지를 생성하거나 웹 페이지 전환을 하면서 기존 웹 페이지를 보지 않게 되는 경우, 해당 웹 페이지의 동적 요소로부터 생성된 타이머만을 선별적으로 중지시켜서 각 사용자 별로 최소 개수의 웹 페이지의 동적 요소만 동작하도록 한다. 이 선별 중지 방법은 웹 브라우저부(134)의 구현 방법에 따라 적용 방법과 적용 여부가 달라지는데, 그것은 어느 웹 페이지에서 SetTimer와 KillTimer를 요청했는지를 구분할 수 있는가의 여부에 따른 것이다. 웹 브라우저부(134)를 마이크로소프트 윈도우즈 운영체제가 자체적으로 제공하는 웹 브라우저 컨트롤을 사용하게 구현한 경우는 각각의 웹 페이지를 별개의 스레드에서 동작하도록 구성할 때에만 적용 가능한데, 이것은 SetTimer와 KillTimer가 요청될 때 어떤 스레드에서 요청되었는지를 구분할 수 있지만 어떤 웹 브라우저에서 요청되었는지를 구분할 수 없기 때문이다. 반면에 웹 브라우저와 웹 페이지 내에서 동작하는 스크립트 실행기를 자체적으로 구현하는 경우는 스크립트 실행기가 자체적으로 SetTimer와 KillTimer를 요청하게 되므로, 웹 페이지들이 같은 스레드에서 동작하는 경우에도 어떤 웹 페이지에서 요청되었는지 구분할 수 있다. 다만 ActiveX 컨트롤의 경우는 컨트롤이 자체적으로 요청하므로 여전히 구분할 수 없어서, 어떤 경우든 각각의 웹 페이지를 별개의 스레드에서 동작하도록 구성하는 것이 바람직하다.
- <61> 또한 웹 페이지 내의 동적 요소들 중 일부는 자신을 포함하는 컨테이너, 즉 웹 브라우저가 ITimerService 및 이와 관련된 COM 인터페이스를 제공하는 경우 타이머가 필요할 때 해당 인터페이스를 통해 요청하게 되므로, 웹 브라우저부(134)가 해당 인터페이스를 제공해서, 웹 페이지 내의 동적 요소들이 타이머를 요청할 때 생성되는 타이머를 SetTimer와 KillTimer를 후킹한 경우와 같이 관리한다. 이 방법에서는 해당 인터페이스가 요청된 컨테이너를 구분할 수 있어서 결국 어떤 웹 페이지로부터 요청되었는지 구분할 수 있다. 다만, 이 방법은 동적 요소들 중 일부만이 사용해서 적용 범위가 넓지 않다는 문제가 있지만, 타이머 관련 성능에 큰 영향을 미치는 플래시 ActiveX 컨트롤이 이 방법을 사용해서 타이머를 요청하므로, SetTimer와 KillTimer 방법을 적용할 수 없는 경우 대신 적용할 가치는 충분하다.
- <62> 상기와 같이 타이머를 기반으로 구성 요소들을 제어하는 방법 이외에, 웹 브라우저나 ActiveX 컨트롤, 스크립트 실행기, 혹은 스크립트 자체에서 동작을 중지 및 재생하는 인터페이스를 제공하는 경우, 그 인터페이스를 사용해서 동작을 중지 및 재생시킬 수 있다.
- <63> 전체 실행부들에서 CPU를 소모하는 동작들을 특정 시각에 일정 개수 이상 실행하지 못하도록 통제하는 것은 해당 동작들을 웹 브라우징 환경 서버(100)에 전역적인 운영체제 상의 동기화 객체를 사용해서 구현한다. 적절한 동기화 객체로는 세마포어가 있다. 통제할 동작들은 주로 CPU를 많이 소모하는 것들인데, 대표적으로는 웹 브라우저 화면을 캡처한 뒤 색상 수를 줄이거나 사용자 기기(200)의 해상도에 맞추어 변형하고 이전에 전송한 화면과 달라진 부분만을 선별해서 전송하기 위해 이전 화면과 비교하는 동작, 플래시 ActiveX 컨트롤이 타이머 이벤트마다 화면을 다시 렌더링하는 동작, 기타 웹 페이지들의 동적 요소들이 스크립트 실행기에 의해 타이머 이벤트마다 수행하는 동작 등이 있다. 하나의 웹 브라우징 환경 서버(100)에서 다수의 실행부(130) 프로세스가 실행되고, 그에 따라 다수의 웹 페이지가 동작하게 되면, 각 프로세스의 스레드들이 CPU를 많이 소모하는 동작들을 수행하면서 운영체제 상에서 CPU 시간을 할당받기 위해 경쟁하게 되고, 그에 따라 운영체제는 선점형
- <64> 멀티태스킹 방식으로 각 스레드들에게 CPU 사용 시간을 할당해 주게 되는데, 경쟁하는 스레드들이 너무 많은 경우 각 스레드들이 미처 자신의 CPU 소모 작업을 미처 완료하지 못한 상태에서 CPU를 빼앗기게 되어서 다시 CPU를 할당받기 위해 기다리게 되므로, 실제로 스레드가 CPU를 사용하는 기간에 비해 운영체제가 CPU를 각 스레드

에 할당해주는, 즉 태스크를 전환하는 작업 자체가 많아져서 무시하지 못할 성능 저하 요소로 작용하게 된다. 이때, 해당 동작을 동시에 일정 개수 이상 실행하지 못하도록 통제해서 스레드들이 각자의 작업을 완료할 시간을 확보해 줌으로써, 태스크를 전환하는 작업을 감소시켜 결과적으로 성능 관련 자원을 절약할 수 있게 된다. 여기서, 실행 개수는 웹 브라우징 환경 서버(100)가 보유한 CPU 개수와 같게 하는 것이 적절하다. 또한, 웹 브라우저 화면 캡처 및 전송에 비해 웹 페이지 내의 ActiveX 컨트롤과 스크립트들이 타이머 이벤트에 의해 실행하는 작업의 중요도가 떨어지므로, 두 가지 동작을 별도로 관리함으로써 전자의 동작이 보다 원활하게 실행되도록 할 수 있다. 예를 들어, CPU가 4개인 웹 브라우징 환경 서버(100)에서는 전자의 작업은 동시에 세 개까지 실행될 수 있도록 하고, 후자의 작업은 동시에 한 개만 실행될 수 있도록 구성할 수 있다. 여기서, 타이머 이벤트에 의한 작업들의 통제는 SetTimer API를 사용할 때 타이머 이벤트가 윈도우와 관련되도록 지정했는지, 아니면 윈도우 없이 단순히 콜백만 호출되도록 지정했는지에 따라 다르게 구현해야 한다. 먼저, 윈도우와 관련된 이벤트의 경우, 윈도우 메시지를 디스패치하는 반복 구문에서 WM_TIMER 메시지가 발견되면 동기화 객체를 획득한 상태에서 해당 WM_TIMER 이벤트를 해당 윈도우에 디스패치하거나 해당 콜백 함수를 호출하는 방식으로 구현한다. 윈도우 없는 콜백의 경우, SetTimer를 후킹할 때 원래의 콜백 대신 성능 제어부(133)가 제공하는 콜백을 운영체제에 넘긴 뒤, 운영체제가 성능 제어부(133)가 제공하는 콜백을 호출했을 때 동기화 객체를 획득한 상태에서 원래의 콜백을 호출하는 방식으로 구현한다.

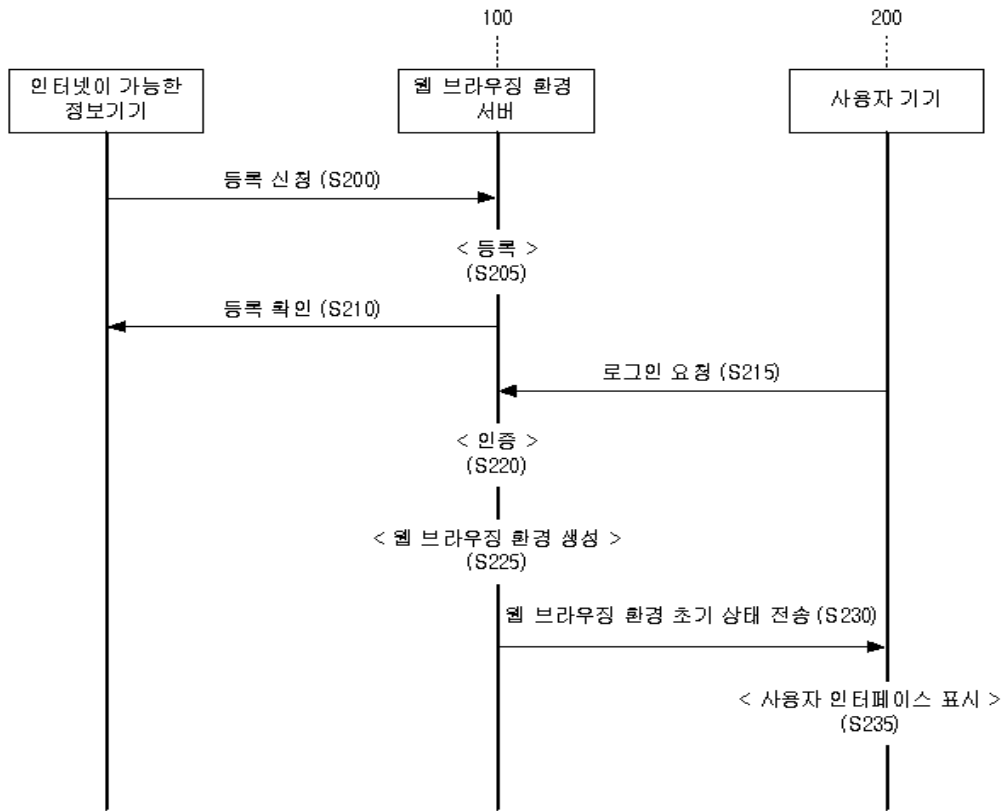
- <65> 이 외에, 웹 브라우저를 자체적으로 구현하는 경우, 하나의 웹 브라우징 환경 서버(100)에서 웹 브라우저부(134)들이 웹 페이지를 다운로드하기 위해 사용할 수 있는 소켓의 개수를 동기화 객체를 통해 제한해서 운영체제 자원을 절약할 수 있다.
- <66> 사용자 기기(200)는 인터넷(300) 및 이동통신망(400)을 통해 웹 브라우징 환경 서버(100)와 통신 가능하다.
- <67> 사용자 기기(200)는 사용자 인터페이스(210)를 사용자에게 보여주고, 원하는 동작이 선택되면 그 동작을 웹 브라우징 환경 서버(100)에 전달, 동작을 수행하도록 한다. 또한 동작 결과에 따라 변경된 상태나 웹 브라우징 환경에서 자동으로 변경된 상태와 웹 페이지 화면을 웹 브라우징 환경 서버(100)로부터 전송받아 사용자 인터페이스(210)에 반영한다.
- <68> 도 2와 도 3은 본 발명의 일실시예에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 동작방법의 설명에 제공되는 메시지 시퀀스 차트이다.
- <69> 먼저, 도 2는 본 발명의 일실시예에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 동작 방법 중 초기화에 관련한 사항들을 설명하기 위한 메시지 시퀀스 차트이다. 도 2를 참조하면, 사용자 기기(200)를 포함하여 인터넷이 가능한 정보기기를 통해 웹 브라우징 환경 서버(100)에 서비스 이용을 위한 등록을 신청하게 되고(S200), 등록이 완료되어(205) 등록 확인을 받으면(210), 사용자는 사용자 기기(200)에 사용자 인터페이스(210)를 제공하는 프로그램이 이미 설치되어 있지 않은 경우 해당 프로그램을 다운로드한다.
- <70> 이후, 사용자 기기(200)에서 웹 브라우징 환경을 사용하기 위해 로그인을 요청하게 되고(S215), 웹 브라우징 환경 서버(100)는 이에 대한 인증절차를 거친 후에(220) 해당 사용자용 웹 브라우징 환경을 생성하고(S225), 웹 브라우징 환경 초기 상태를 전송해 준다(S230). 사용자 기기(200)에서는 전송 받은 웹 브라우징 환경 초기 상태를 메뉴와 함께 사용자 인터페이스에 표시한다(S235).
- <71> 도 3은 본 발명의 일실시예에 따른 원격지 서버 컴퓨터를 이용한 다중 사용자용 웹 브라우징 환경 제공 시스템의 동작 방법 중 사용자 기기(200)의 동작 선택부터 실제 웹 브라우저부(134)의 동작 및 결과 전달에 관련한 사항들을 설명하기 위한 메시지 시퀀스 차트이다. 도 3을 참조하면, 사용자 기기(200)에서는 화면에 표시된 사용자 인터페이스(210)를 조회하여(S300), 필요한 기능을 선택하여 해당 동작을 웹 브라우징 환경 서버(100)에 전달한다(S305).
- <72> 웹 브라우징 환경 서버(100)의 웹 브라우저 제어부(132)는 사용자 기기(200)로부터 요청된 동작을 웹 브라우저부(134)에 전달한다(S310). 이때 동작 전달 방법은 웹 브라우저부(134)를 구성하는 웹 브라우저 컨트롤의 인터페이스 함수의 호출이나, 독립 환경 구성부(131)를 통한 윈도우 메시지 전송 및 키보드와 마우스 시뮬레이션 등이 있다. 웹 브라우저부(134)는 전달받은 동작을 수행한다(S315). 이때 웹 브라우저부(134)는 성능 제어부(133)로부터 성능 통제를 받을 수 있다. 웹 브라우저 제어부(132)는 웹 브라우저부(134)의 상태 변경이 감지되면(S320) 변경된 사항에 관한 정보를 사용자 기기(200)에 전달한다.
- <73> 정보를 전달받은 사용자 기기(200)는 사용자 인터페이스(210)에 정보를 적용해 표시한다(S335).

도면

도면1



도면2



도면3

