(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.[7]
G06F 3/14

(11) 공개번호 10-2004-0089600
(43) 공개일자 2004년10월21일

(21) 출원번호 10-2004-7012142
(22) 출원일자 2004년08월05일
번역문 제출일자 2004년08월05일
(86) 국제출원번호 PCT/US2003/003647
(87) 국제공개번호 WO 2003/067568
(86) 국제출원출원일자 2003년02월06일
(87) 국제공개일자 2003년08월14일

(30) 우선권주장 10/068,461 2002년02월06일 미국(US)
10/288,821 2002년11월06일 미국(US)

(71) 출원인 사이트릭스 시스템스, 인크.
미국 플로리다 33309, 포트 라우더데일, 851 더블류. 사이프레스 크릭 로드

(72) 발명자 블룸필드,마크
미국 플로리다 33064, 라이트하우스 포인트, 47 스트리트, 2610 엔.이.

뮤이에,제프
호주 뉴사우스 웨일즈, 4207, 로건 빌리지, 로트 17 매누카 로드

파나슈크,아나토릴리
호주 엔에스 더블유 2120, 페넌트 힐스, 52 에이 알비온 스트리트

비흐라키스,엘리아스,씨.
미국 메사츠세츠 01826, 드라컷, 20 콘코드 로드

(74) 대리인 신영무

심사청구 : 없음

(54) 웹페이지에 디스플레이된 소프트웨어 어플리케이션과의상호작용

요약

본 발명은 웹브라우저 윈도우(320)내에 임베딩된 어플리케이션-출력 윈도우(326)내에 어플리케이션-출력 데이터의 디스플레이를 가능케 한다. 어플리케이션-출력 윈도우(326)는 어플리케이션-출력 데이터의 소스를 제공하는 어플리케이션 프로그램이 웹 실행되지 않더라도(non-web enabled)(예, 레거시 어플리케이션), 동적으로 이동 및 크기조절되며 웹브라우저 윈도우(320)내에서 조작될 수 있다. 본 발명은 제1 가상 채널(660)을 통해 어플리케이션 출력 윈도우(326)와 연관된 윈도우 속성 정보를 수신하고 제2 가상 채널(670)을 통해 수신된 어플리케이션-출력 데이터를 어플리케이션-출력 윈도우(326)내에 디스플레이하며, 어플리케이션-출력 윈도우는 윈도우 속성 정보를 이용하여 형성 및/또는 변경될 수 있다.

대표도

도 3

색인어

어플리케이션-출력(application-output) 윈도우, 웹브라우저, 윈도우 속성 정보

명세서

기술분야

본 발명은 통신망에서 정보를 디스플레이하는 것에 관한 것으로서, 구체적으로는 웹페이지에 실행 어플리케이션 프로그램의 출력을 디스플레이하는 것에 관한 것이다.

배경기술

오늘날 빠른 속도로 진행되는 글로벌 시장환경에서의 비즈니스는 그들의 직원들에게 작업장내의 고정된 위치에서뿐만 아니라 모바일(mobile) 위치에서도 비즈니스에 중요한 어플리케이션 및 데이터에 대한 액세스를 제공함으로써 그들의 생산성과 수익성을 개선시키려고 애쓴다. 웹기반 컴퓨팅의 대중성은, 모바일 사용자를 위한 정보 액세스를 촉진시킬 필요성과 결합되어, 기업 포털(enterprise portal)의 채택에 박차를 가하였다. 기업 포털은 어플리케이션, 데이터 및 콘텐트를 모으고, 맞춤화하여 사용자에게 서비스하고, 정보를 보다 효율적으로 조직화하고 사용하기 위한 관리툴을 제공하는 회사 웹사이트이다. 일부 회사들의 경우에, 포털은 전통적인 데스크탑 소프트웨어를, 사용하기 용이하고, 편리하고, 유비쿼터스(ubiquitous)한 가상 작업장에 대한 브라우저-기반 액세스로 대체하였다. 또한, 포털을 구현하는 회사들은 그들의 정보 기술 인프라구조에서 작업자 생산성 증가와 개선된 효율성으로 인하여 투자에 대한 신속한 수익결과로부터 이득을 얻고 있다.

진정한 데스크탑 소프트웨어 대체는 포털이 전체 정보 자원을 제공할 것을 필요로 한다. 비즈니스 어플리케이션은 아마도 작업자들이 액세스할 필요가 있는 가장 중요한 정보 자원일 것이다. 또한, 비즈니스 어플리케이션은 전형적으로 주요 투자대상을 대표하고, 종종, 웹-기반 시스템으로 이동시에 손실되지 말아야 할 경쟁적인 장점을 제공한다. 생산성 기준에서, 사용자가 브라우저와 그의 작업을 수행하는 데스크탑간에 왔다갔다하지 않도록 하기 위해서는 포털을 통한 어플리케이션 액세스가 필요하다. 분할된 브라우저/데스크탑 시스템으로는 다양한 소스(source)로부터 자료를 찾고 통합시키는 것이 더 어렵다. 또한, 데스크탑 장치는 브라우저를 통해 액세스될 수 없는 중요한 어플리케이션을 제공하기 때문에 사용자들은 통상적으로 데스크탑 장치에 구속된다.

비즈니스 조종자들은 포털 구현에 기존의 어플리케이션 및 이후의 출현 어플리케이션을 포함하고자 하는 강력한 동기를 제공한다. 그러나, 웹-기반 출하용으로 개발된 어플리케이션을 거의 없으며, 그러한 어플리케이션이 있다고 하더라도 그들의 상응하는 데스크탑 어플리케이션에 비해 축소된 기능을 제공한다. HTML, 스크립팅(scripting), 자바(Java) 및 기타 특정 수단을 이용하여 기존의 어플리케이션을 웹 공표용으로 재설계함으로써 포털내에서 그들의 사용을 가능하게 할 수도 있지만, 이러한 방식은 시간 및 비용이 많이 들고, 포컬 구현을 지연시킬 수 있다. 마찬가지로, 이러한 구현은 감소된 기능을 실현하거나 어플리케이션 실행의 '다운로드 및 실행'모델이 너무 자원-집중적이기 때문에 실행가능하지 않을 수 있다.

따라서, 효율성이 근소하게 개선되어 포털 투자에 대한 수익을 희석시킬 수 있는 고가의 개발 노력을 들이지 않고 레거시 어플리케이션을 기업 포털에 효율적으로 전개(deployment)할 수 있는 방법 및 시스템이 필요하다.

발명의 상세한 설명

본 발명은 웹-시행(web-enablement) 기술을 레거시 어플리케이션에 적용하여 이들 어플리케이션들이 그들의 사용자들이 이미 익숙한 동일한 사용자 인터페이스 및 완전한 기능들을 구비하여 그들의 원래 형태로 유지될 수 있도록 해주고, 뷰어(viewer)들이 웹 페이지내에 디스플레이되는 어플리케이션-출력 윈도우를 통해 어플리케이션과 상호작용할 수 있도록 해줌으로써, 전술한 단점들을 극복한다.

일 실시예에서, 본 발명은 웹브라우저 윈도우내에 위치한 하나 이상의 어플리케이션-출력 윈도우내에 어플리케이션-출력 데이터를 디스플레이하는 방법을 제공한다. 어플리케이션-출력 데이터는 웹-시행 어플리케이션 프로그램 및/또는 수정되지 않은 비-웹 시행 어플리케이션(예, 소위 '레거시 어플리케이션')에 의해 생성된 데이터일 수 있다. 어플리케이션-출력 데이터의 소스를 제공하는 어플리케이션 프로그램은 상이한 어플리케이션 서버들상에 존재할 수 있으며 이들 상이한 어플리케이션 서버들로부터의 어플리케이션-출력 데이터는 동일한 웹브라우저내의 하나 이상의 어플리케이션-출력 윈도우내에 디스플레이된다. 일양상에서, 어플리케이션-출력 윈도우는 웹브라우저 윈도우의 자식 윈도우이다. 일 실시예에서, 어플리케이션-출력 윈도우는 웹브라우저 윈도우의 경계선을 벗어나 재배치될 수 있다

. 또다른 양상에서, 어플리케이션-출력 윈도우와 연관된 윈도우 속성 정보가 제1 가 상 채널을 통해 수신되고 어플리케이션-출력 데이터(예, 그래픽 데이터)는 제2 가상 채널을 통해 수신된다. 또다른 양상에서, 웹브라우저 윈도우내에 디스플레이되는 어플리케이션-출력 윈도우의 윈도우 속성 정보는 웹브라우저 윈도우를 형성한 웹브라우저와 독립적으로 수정될 수 있다. 본 발명은 윈도우 속성 정보에 따라 어플리케이션-출력 윈도우에 어플리케이션-출력 데이터를 디스플레이한다.

일실시예에서, 본 발명은 어플리케이션-출력 윈도우와 연관된 이벤트를 모니터링하고 이에 응답하는 클라이언트 에이전트를 제공하는데, 예를 들어, 웹브라우저 윈도우의 뷰어에 의해 입력되는 크기조절 이벤트를 검출하고 이에 응답하여 영향받는 어플리케이션-출력 윈도우의 크기를 조절한다. 클라이언트 에이전트는 이러한 기능을 웹브라우저와 무관하게 수행한다. 일실시예에서, 클라이언트 에이전트는 제2 가상 채널을 통해 수신된 어플리케이션-출력 윈도우의 어플리케이션 속성 정보를 이용하여 어플리케이션 객체의 적어도 하나의 특성을 수정하고, 수정된 특성은 관련 어플리케이션-출력 윈도우에 상응하는 변화를 개시한다. 또다른 양상에서, 클라이언트 에이전트는 웹브라우저와 무관하게 제1 및 제2 가상 채널을 개설한다.

이러한 방식으로, 본 발명은 고가의 개발 및 사용자 교육 비용을 회피하고, 웹 액세스를 갖는 모바일 위치에서 원래 레거시 어플리케이션의 완전한 기능에 대한 액세스를 제공한다.

도면의 간단한 설명

본 발명은 첨부된 청구항에 구체적으로 지적되어 있다. 본 발명의 전술한 장점 및 더나아간 장점들은 이하의 첨부된 도면과 함께 이하의 설명을 참조하면 가장 잘 이해될 것이다.

도 1은 데이터 통신망을 통해 연결된 클라이언트 장치, 웹서버 및 서버팜(server farm)을 개략적으로 예시하며, 여기에서 클라이언트 장치상의 클라이언트 에이전트와 서버팜내의 하나 이상의 서버 에이전트는 본 발명의 실시예에 따라 작동한다.

도 2는 본 발명의 실시예에 따라 클라이언트 에이전트와 웹서버에 의해 수행되는 단계들을 설명한 상위 수준 흐름도를 제공한다.

도 3은 클라이언트 데스크탑의 예시적인 스크린 표시로서, 클라이언트 데스크탑의 웹브라우저 윈도우내에 디스플레이되는 어플리케이션-출력 윈도우가 본 발명의 실시예에 따라 형성된다.

도 4는 도 3의 어플리케이션-출력 윈도우의 스크린 표시로서, 본 발명의 실시예에 따라 어플리케이션-출력 윈도우가 웹브라우저 윈도우내에서 이동 및/또는 크기조절될 수 있음을 도시한 것이다.

도 5는 도 3의 어플리케이션-출력 윈도우의 스크린 표시로서, 어플리케이션-출력 윈도우가 본 발명의 실시예에 따라 웹브라우저 윈도우의 경계선 내부에 및/또는 경계선을 벗어나 디스플레이될 수 있으며, z-순서를 변화시킬 수 있음을 보여준다.

도 6은 본 발명의 실시예에 따라 클라이언트 에이전트와 서버 에이전트간에 전송되는 데이터 유형과 클라이언트 에이전트의 프로세스를 개략적으로 도시한다.

실시예

원격 사용자들에게 어플리케이션-출력 데이터를 디스플레이할 수 있는 포털 및 기타 웹-기반 구현은 바람직하게는 서버-기반 컴퓨팅 모델을 이용하여 구현된다. 서버-기반 컴퓨팅은 기업 포털과 유사하며, 이는 서버-기반 컴퓨팅이 단일-지점 어플리케이션 관리를 통해 포털의 전체 효율성을 개선한다는 점에서, 정보 자원(예, 어플리케이션 프로그램들)에 대한 단일-지점 액세스를 통한 사용자의 생산성을 향상시킨다.

서버-기반 컴퓨팅 모델에서, 어플리케이션 처리, 관리, 지원 및 전개는 전형적으로 하나 이상의 중앙 서버에 기반을 두는데, 이들 서버는 사용자의 디스플레이 장치로부터 지리적으로 떨어져 있다. 원격 사용자들은, 키스트로크, 마우스 이동 및 다른 입/출력 동작들을 데이터 통신망(예, LAN, MAN, WAN, 인터넷, 인트라넷)을 통해 어플리케이션에 전송함으로써 어플리케이션 서버상에 호스팅된 특정 어플리케이션들과 상호작용하고, 그들로부터 스크린/윈도우 갱신, 파일 및 기타 다른 데이터를 수신한다. 전부는 아니더라도 대부분의 어플리케이션 처리가 서버상에서 발생하기 때문에, 사용자의 디스플레이 장치는 딘(thin) 클라이언트로서 기본적으로 작동하므로 컴퓨터-집약 어플리케이션이 될 수 있는 어플리케이션-출력 데이터를 디스플레이하기 위해 적은 자원(예, 처리능력, 비휘발성 메모치, 휘발성 메모리 등

)을 필요로 하게 된다. 또한, 네트워크를 통해 경유하는 데이터의 전체 양을 감소시킴으로써 어플리케이션 성능 및 보안에 상당한 개선이 실현될 수 있다.

이러한 서버-기반 컴퓨팅 모델과 어플리케이션 프로그램(예, 포털)의 웹-기 반 구현의 결합은 사용자가 웹에 연결된 임의의 장치(예, 홈 컴퓨터, 랩탑 컴퓨터, 컴퓨터 워크스테이션, 무선 휴대용 통신 장치 및 다양한 범위의 플랫폼상에서 작동되는 정보 어플라이언스(appliance)를 포함함)상에서 실질적으로 비즈니스에 중요한 어플리케이션을 액세스하는 것을 가능케 해준다. 장치 및 플랫폼 유동성은 모바일 작업자들이 하나의 장치에서 다른 장치로 끊김없이(seamlessly) 이동하고 비즈니스에 중요한 레거시 어플리케이션의 전체 기능에 대한 액세스를 포함하는 일관되고 맞춤화된 정보 집합을 수신할 수 있도록 해준다.

간략히 살펴보면, 웹에 연결된 장치의 사용자는 웹서버에 있는 하나 이상의 어플리케이션 프로그램에 대한 액세스를 요청한다. 사용자 증명서(credential)를 인증한 후에, 웹 서버는 웹서버에 결합된 메모리로부터 사용자에 특정되고 어플리케이션에 특정되는 파라미터를 액세스한다. 다음에, 웹 서버는 이들 파라미터를 상기 요청된 어플리케이션 프로그램을 호스팅하는 하나 이상의 어플리케이션 실행 서버에 전달하고, 어플리케이션 실행 서버상에서 작동하는 소프트웨어 프로세스는 상기 전달된 파라미터를 이용하여 상기 요청된 어플리케이션 프로그램을 실행시키고 초기화한다. 이러한 방식으로, 어플리케이션 프로그램의 각 인스턴스가 특정의 요청 사용자에에 대하여 맞춤화된다. 이렇게 맞춤화된 어플리케이션 프로그램을 호스팅하는 어플리케이션 실행 서버의 구체적인 네트워크 어드레스가 사용자 장치에 전달되고, 사용자 장치는 어플리케이션 실행 서버와 통신 링크 및 클라이언트-서버 세션을 개설한다.

실행 어플리케이션 프로그램과 연관된 명령, 이벤트, 그래픽 데이터 및 윈도 우 속성 정보가 클라이언트-서버 세션동안 사용자 장치와 어플리케이션 실행 서버간에 통신되어 어플리케이션-출력 데이터가 사용자 장치의 데스크탑상에 끊김없이 디스플레이되는 것을 보장한다. 어플리케이션-출력 데이터의 끊김없는 디스플레이는 로컬 실행 어플리케이션이 사용자 장치의 로컬 데스크탑에서 제공되고 처리되는 방식과 일치하는 방식으로 사용자 데스크탑상에 데이터를 표시하는 것을 가리킨다. 즉, 사용자는 원격 어플리케이션 프로그램이 로컬에서 실행되는 것처럼 어플리케이션 프로그램에 의해 생성된 어플리케이션-출력 데이터를 보고 상호작용한다.

일실시예에서, 어플리케이션 프로그램의 출력은 사용자 장치의 웹브라우저에 의해 디스플레이되는 웹페이지내에 위치한 하나 이상의 어플리케이션-출력 윈도우내에 디스플레이된다. 다른 실시예에서, 어플리케이션-출력 윈도우의 속성은 어프리케이션-출력 윈도우가 웹페이지의 경계 내에서 이동가능하고 크기조절이 가능하도록 수정될 수 있다. 또 다른 실시예에서, 어플리케이션 출력 윈도우가 처음에는 웹페이지의 경계선 내에 보이다가 이후에 웹페이지의 경계선 바깥에 위치하도록 이동할 수 있으며, 이에 따라 어플리케이션-출력 윈도우가 원격-실행 어플리케이션보다는 로컬-실행 어플리케이션에 대응하는 인상을 준다. 또한, 또다른 실시예에서, 어플리케이션-출력 윈도우가 처음에는 웹페이지의 경계선 외부에 보이며, 따라서, 로컬-실행 어플리케이션에 대응하는 것처럼 보인다. 일 실시예에서, 어플리케이션-출력 윈도우내에 디스플레이되는 어플리케이션 출력과 어플리케이션-출력 윈도우의 속성 그 자체는, 어플리케이션 프로그램에 대한 액세스를 초기에 제공하는 웹서버 또는 웹브라우저의 간섭없이, 사용자 장치 및 어플리케이션 실행 서버상 의 소프트웨어 프로세스에 의해 전달되고 처리된다.

좀더 구체적으로 도1을 참조하면, 서버-기반 컴퓨팅 구조(100)는 원격 사용자에게 웹 및 레거시 어플리케이션(예, 웹 기반 출하용으로 설계되지 않은 변경되지 않은(unmodified) 어플리케이션 프로그램)의 전체 기능에 대한 웹 액세스를 제공할 수 있으며, 클라이언트 장치(110, 예를 들어, 임의의 디지털 데이터 처리 장치), 웹 서버(112), 단독이거나 서버 팜(116)내에 클러스터링되고 방화벽(118)에 의해 바람직하게 보호되는 하나 이상의 어플리케이션 실행 서버(114)와, 이들 요소들 각각이 서로 통신하는 것을 가능하게 하기 위해 필요한 연결성을 제공하는 데이터 통신망(120), 예를 들어, 인터넷, 인트라넷 등)을 포함한다.

동작에 있어서, 도 2를 참조하면, 클라이언트 장치(110)의 사용자는 클라이언트 장치(110)상에 실행되는 브라우저(122)에게 웹서버(112)를 통해 액세스될 수 있는 특정 웹페이지 콘텐트(125)에 대한 액세스 요청을 제출할 것을 지시한다(단계 210). 일실시예에서, 사용자는 브라우저(122)에 URL 어드레스를 입력한다. URL은 웹서버(112)에 의해 호스팅되는 웹페이지 콘텐트(124)와 연관되어 있고 브라우저(122)는 적절한 URL 어드레스에 대한 액세스 요청을 전송함으로써 응답한다. 웹서버(112)는 통상적으로 사용자 증명 정보(예, 사용자 ID, 패스워드, 그룹/프로젝트 멤버쉽 식별자 등)를 포함하는 액세스 요청을 수신하고, 웹페이지 콘텐트(124)의 일부를 제공하는 서버팜(116) 또는 개별 서버(114)에 대해 사용자를 인증한다(단계 212).

웹서버(112)는 기지정된(previously-assigned) 증명정보와 사용자에 의해 입 력된 증명정보를 비교하는 인증 프로세스를 액세스함으로써 사용자를 인증한다. 일 실시예에서, 인증 프로세스 및 기지정된 증명정보의 데이터베이스는 웹서버(112)에 저장되고 유지된다. 다른 실시예에서, 기-지정된 증명정보는 서버팜(116)내에, 개별 어플리케이션 실행 서버(114), 및/또는 인터넷 또는 다른 데이터 통신망을 통해 웹서버(112)에 연결된 관리 서버(미도시)상에 저장될 수 있다.

웹페이지 콘텐트(124)가 어플리케이션 집합(126, 예, 포털 관리자에 의해 사용자를 위해 맞춤화된 어플리케이션 프로그램 집합)에 대한 액세스를 제공하는 기업 포털에 대응하는 시나리오의 경우에, 웹서버(112)는 웹서버측 스트립트를 호출하는 하나 이상의 어플리케이션 객체(128, 예, COM-컴플라이언트(compliant) 자바 객체, 액티브 X 객체(ActiveX objects), HTML 태그 등)를 액세스하여 사용자를 인증하고/인증하거나(단계 212), 서버팜(116)으로부터 포털과 사용자에 연관된 어플리케이션 집합(126) 정보를 획득한다(단계 214). 또한, 어플리케이션 객체(128)는 사용자 및/또는 포털을 통해 제공되는 어플리케이션 집합(126)내의 특정 어플리케이션(130)과 연관된 특성을 포함한다. 사용자 특성은, 예를 들어, 사용자가 그룹/프로젝트의 다른 구성원들과 협력하여 작업할 수 있도록 하기 위해 사용자가 액세스할 필요가 있는 특정 어플리케이션(130) 및 데이터를 식별하는 그룹/프로젝트 정보를 포함한다. 어플리케이션 특성은, 예를 들어, 어플리케이션 집합(126)의 각 어플리케이션(130)에 대한 사용자 프레퍼런스(preference)를 포함한다.

어플리케이션 객체(128)에 의해 호출된 스크립트는, 예를 들어, 서버 팜(116)내의 각 서버(114)를 모니터링하고 제어하는 중앙 관리 프로세스(미도시)를 통해, 웹서버(112)와 서버팜(116)간의 네트워크 세션을 개설한다. 관리 프로세스는 예를 들어, 서버 및/또는 네트워크 성능 기반에 근거하여, 어플리케이션 객체(128)에 의해 지정된 어플리케이션 집합(126)의 어플리케이션 프로그램(130)을 호스팅하는 하나 이상의 서버를 선택한다. 원하는 어플리케이션 집합(126)은 서버(114)상에 호스팅되는 다수의 어플리케이션들(130, 132)로부터 어플리케이션 집합(126)의 각 어플리케이션(130)을 선택/할당함으로써 단일 서버(114)에 의해 완전히 제공될 수 있다. 대안적으로, 어플리케이션 집합(126')은 다수의 서버(114)에 의해 제공될 수 있으며, 다수의 서버(114)의 각각은 어플리케이션 집합(126')의 어플리케이션 프로그램중 적어도 하나를 호스팅한다. 서버 팜 및 그들의 관리/동작에 대한 좀더 상세한 설명은 국제특허출원 PCT US01/14314에서 찾아볼 수 있으며, 상기 출원은 전체적으로 본명세서에서 참조된다.

관리 프로세스는 어플리케이션 객체(128)에 의해 호출된 스크립트에 응답하여 선택/할당된 서버(114)상에 하나 이상의 서버 에이전트(134)를 개시한다(launch). 서버 에이전트(134)는 어플리케이션 객체(128)에 의해 지정된 특성에 따라 어플리케이션 집합(126)의 어플리케이션 프로그램(130) 각각을 실행시키고, 초기화시키고, 이들과 상호작용하는 소프트웨어 프로세스이다. 일 실시예에서, 어플리케이션 집합(126)의 각 어플리케이션 프로그램(130)에 대한 서버 에이전트(134)가 있다. 다른 실시예에서, 모든 어플리케이션 프로그램(130)이 동일한 서버(114)상에 호스팅되도록, 어플리케이션 집합(130)에 대한 단일 서버 에이전트(134)가 있다. 다른 실시예에서, 각 서버(114)에 대한 단일 서버 에이전트(134)가 있 다. 서버 에이전트(134)는 어플리케이션 집합(126)의 어플리케이션 프로그램의 출력결과뿐만 아니라, 어플리케이션 집합(126)에 관련된 기타 다른 정보를 웹서버(112)에 제공하고, 웹서버(112)는 어플리케이션 집합 정보를 웹페이지 콘텐트(124)로 포맷팅한다(단계 216). 웹페이지 콘텐트(124)는 어플리케이션 집합(130)의 하나 이상의 어플리케이션에 대응하는 어플리케이션 아이콘과 상기 하나 이상의 어플리케이션 프로그램(130)으로부터의 어플리케이션-출력 데이터를 포함할 수 있다. 일실시예에서, 어플리케이션 프로그램(130)에 의해 제공되는 어플리케이션-출력 데이터는 어플리케이션 객체(128)의 특성에 의해 초기에 지정된 속성(예, 웹페이지상의 윈도우 위치, 크기, 스타일, z-순서 등)을 나타내는 윈도우에 맞도록 포맷되는 그래픽 데이터에 대응한다.

하나의 예시적인 실시예에 있어서, 도 3을 참조하면, 브라우저(122)는 웹페이지 콘텐트(124)를 수신하여 브라우저 윈도우(320)내에 디스플레이하는데, 브라우저 윈도우(320)는 클라이언트 장치(110)에 연결된 디스플레이 장치상에 디스플레이된 클라이언트 데스크탑(310)을 형성하는 다수의 가능한 그래픽 사용자 인터페이스(GUI) 요소들(예, 메뉴(312), 로컬 윈도우(314) 등)을 포함한다(단계 218). 이러한 특정 실시예에서, 웹페이지 콘텐트(124)는 브라우저(320)에 의해 디스플레이되는 웹페이지(322)내에 디스플레이되고, 어플리케이션 집합(126)과 연관된 하나 이상의 어플리케이션 아이콘(324) 및/또는 하나 이상의 어플리케이션-출력 윈도우(326)를 포함한다. 일 실시예에서, 하나 이상의 어플리케이션 객체(128)는 웹페이지(322)의 웹페이지 콘텐트(124)의 일부를 또한 형성하고 따라서 어플리케이션-출력 윈도우(326)의 초기 속성(크기, z-순서, 위치)을 설정할 수 있다. 후술되는 바와 같이, 웹페이지(322)상에 디스플레이되는 어플리케이-출력 윈도우(326) 각각의 초기 방위, 크기, 위치 및 z-순서가 수정되어, 어플리케이션-출력 윈도우(326)가 웹페이지(322)에 대하여 및/또는 클라이언트 데스크탑(310)에 대하여 상이한 방위, 크기, 위치 및 z-순서를 보여주도록 할 수 있다.

어플리케이션 객체(128)는 웹페이지 콘텐트(124)를 디스플레이하는 어플리케이션-출력 윈도우(326)가 웹페이지(322)의 특정 위치에 디플레이되어야 함을 브라우저(122)에 지시하는 임의의 데이터 구조물일 수 있다. 어플리케이션 객체(128)는, 높이, 폭, 경계선 스타일, 어플리케이션-출력 윈도우(326)의 배경색상 또는 패턴뿐만 아니라 윈도우(326)내에 디스플레이될 수 있는 어플리케이션(130)의 표시자, 출력 디스플레이가 갱신되는 횟수, 또는 어플리케이션 출력 디스플레이를 증강시키는데 유용한 부가적인 정보를 포함할 수 있다.

예시적인 일실시예에서, 어플리케이션 객체(128)는 HTML 파일에 임베딩된 윈도우 태그이며, 이러한 태그의 예가 다음에 기술된다.

액티브 X 태그

<object claaid='clsid'238f6f83-b8b4-11cf-8771-00a024541ee3'

data = '/ica/direct.ica'CODEBASE='/cab/wfica.cab'

width = 436 height=295>

<param name='Start' value='Auto'>

<param name='Boarder' value='On'>

</object>

넷스케이프 플러그인 태그

<embed src='http://www.citrix.com/ica/direct.ica'

pluginpage='http://www.citrix.com/plugin.html'

height=295 width=436 Start=Auto Border=On>

<embed>

자바 태그

<applet code=JICA.class width = 436 height=295>

<param name=Address value='128.4.1.64'>

<param name=InitialProgram value=Microsoft Word7.0>

<param name=Start value=Auto>

<param name=Border value=On>

</applet>

전술한 각 경우에서, 태그는 295개의 픽셀 높이 및 436개의 픽셀 폭을 갖는 어플리케이션-출력 윈도우(326)가 어플리케이션 프로그램(130)으로부터 출력 데이터를 수신하도록 그려져야 됨을 표시한다. 또한, 각 태그는 어플리케이션 프로그램(130)이 실행을 자동 시작하고 어플리케이션 출력이 디스플레이되는 어플리케이션-출력 윈도우(326)가 경계선으로 그려져야 함을 지정한다. 액티브X 및 넷스케이프 플러그인 태그는 디렉토리 '/ica'에 위치한 'direct.ica' 파일에 지정된 원격 어플리케이션(130)의 속성을 갖는다. 자바 태그는 원격 어플리케이션(130)의 속성 을 직접 지정한다. 전술한 예에서, 실행될 어플리케이션 프로그램(130)의 명칭뿐만 아니라, 어플리케이션 프로그램(130)을 호스팅하는 서버(114)의 어드레스 또한 지정된다.

일실시예에서, 어플리케이션 프로그램(130)은 웹페이지(322)의 디스플레이와 거의 동일한 시점에 실행한다. 또다른 실시예에서, 어플리케이션 프로그램(130)은, 웹페이지 콘텐트(124)를 웹서버(112)에 제공하는 부분으로서, 서버 에이전트(114)에 의해 실행할 것을 지시받으면 실행한다. 또다른 예에서, 어플리케이션 프로그램은 사용자-지정 입력(예, 웹페이지(322)상에서 어플리케이션 아이콘(324)의 선택)과 같은 신호에 응답하여 실행한다. 어플리케이션 프로그램(130)의 실행이 개시되면, 브라우저(122)는 클라이언트 장치(110)상에 클라이언트 에이전트(136)를 인스턴스화한다(instantiate)(단계 220). 대안적으로, 클라이언트 에이전트(136)는 웹페이지(322)의 디플레이와 거의 동일한 시점에 또는 사용자-지정 입력에 응답하여 인스턴스화된다.

클라이언트 에이전트(136)는 하나 이상의 소프트웨어 프로세스를 포함하고, 이는 클라이언트 장치(110)상에서 실행되며 서버 에이전트(134), 브라우저(122), 어플리케이션-출력 윈도우(326) 및/또는 웹서버(112)와 상호작용하도록 구성된다. 일실시예에서, 클라이언트 에이전트(136)는 브라우저(122)의 자식 프로세스로서 스펀(spawn)된다. 다른 실시예에서, 클라이언트 에이전트(136)는 브라우저(122)와 피어(peer) 프로세스이거나, 브라우저(122)와 연관된 동적 연결 라이브러리(dynamically linked library)이다. 일 실시예에서, 웹페이지(322)에 디스플레이 되는 각 어플리케이션-출력 윈도우(326)에 대하여 클라이언트 에이전트(136)가 인스턴스화된다. 또다른 실시예에서, 어플리케이션 집합(126)의 특정 어플리케이션 프로그램(130)과 연관된 하나 이상의 어플리케이션-출력 윈도우(326)에 대해 단일 클라이언트 에이전트(136)가 인스턴스화된다. 또다른 실시예에서, 웹페이지 콘텐트(124)를 제공하는 각각의 서버 에

이전트(134)에 대하여 단일 클라이언트 에이전트(136)가 인스턴스화된다. 또다른 실시예에서, 전체 어플리케이션 집합(126)에 대하여 단일 클라이언트 에이전트(136)가 인스턴스화된다.

브라우저(122)는 어플리케이션 집합(126)내의 특정 어플리케이션 프로그램들(130)에 관한 어플리케이션 객체(128)의 속성들을 동일한 어플리케이션 프로그램(126)들에 연관된 클라이언트 에이전트(136)에게 전달한다. 또한, 브라우저(122)는 클라이언트 에이전트(136)에 어플리케이션-출력 윈도우(326)의 핸들(handle)을 전달하거나, 클라이언트 에이전트(136)는 브라우저(122)에 질의함으로써 어플리케이션-출력 윈도우(326)의 핸들을 검색할 수 있다. 브라우저(122) 또는 어플리케이션 객체(128)에 의해 지정되지 않은 어플리케이션 특성들은 디폴트값으로 설정될 수 있다. 또한, 클라이언트 에이전트(136)는 특정의 특성 디폴트값이 하드-코드화되거나, 클라이언트 에이전트(136)는 특성 디폴트값을 포함하는 파일을 액세스할 수 있다. 클라이언트 에이전트(136)는 어플리케이션 객체(128) 특성의 일부로서 제공되는 어플리케이션 프로그램(130) 명칭과 어플리케이션 실행 서버(114) 주소를 이용하여, 통신 링크를 개설하고 서버(114) 및 어플리케이션 프로그램(130)과 연관된 서버 에이전트(134)와 클라이언트-서버 세션을 개시한다. (단계 222). 클라이언 트 에이전트(136)는 필요한 디폴트값과 함께 서버 에이전트(134)에 어플리케이션 객체(128) 특성의 일부 또는 전부를 전달한다. 대안적으로, 서버 에이전트(134)는 이후에 디스플레이되는 웹페이지 콘텐트(124)를 제공하기 이전에 웹서버(112)로부터 어플리케이션 객체(128) 속성의 일부 또는 전부를 이미 수신하였을 수 있다. 특정의 특성이 서버 에이전트(134)에 전달되지 않는다면, 서버 에이전트(134)는 필요한 특성(예, 사용자 ID)의 디폴트값이 없는 경우에, 클라이언트 에이전트(136)에 요청하거나, 서버 에이전트(134)가 임의의 특성(예, 실행 우선순위)에 대하여 자신의 디폴트값을 제공할 수 있다.

서버 에이전트(134)는 클라이언트 에이전트(136)로부터 수신된 특성을 이용하여 클라이언트 에이전트(136)를 인증하고, 소정의 어플리케이션 프로그램(130)을 (그것이 아직 개시되지 않았다면) 실행시킨다. 어플리케이션 프로그램(130)이 실행되고 클라이언트 에이전트(136)가 일단 인증되었다면, 어플리케이션 프로그램(130)은, 브라우저(122) 또는 웹서버(112)의 개입없이, 클라이언트 에이전트(136)와 서버 에이전트(134)를 통해 직접 통신한다. 클라이언트 에이전트(!36)는 어플리케이션 프로그램(130)으로부터 출력 데이터를 수신하여 웹페이지(322)의 적절한 어플리케이션 출력 윈도우(326)에 그 출력 데이터를 디스플레이한다. 또한, 클라이언트 에이전트(136)는 어플리케이션-출력 윈도우(130)와 연관된 마우스 클릭 및 키보드 입력과 같은 입력 이벤트들을 검출하고 이러한 입력 이벤트들을 서버 에이전트(134)를 통해 어플리케이션 프로그램(130)에 전달한다. 이러한 유형의 클라이언트-서버 세션은 사용자에 의해 선택된 어플리케이션 집합(126)의 각 어플리케이 션(130)에 대해 반복되며, 따라서, 사용자가 어플리케이션 집합(126)의 모든 자원들과 상호작용하는 것을 가능케 한다 (단계 224).

클라이언트-서버 세션동안 클라이언트 에이전트(136)와 서버 에이전트(134)간에 교환되는 데이터는 어플리케이션 프로그램(130)의 입력 이벤트 및 그래픽 출력 데이터를 포함할 뿐만 아니라 윈도우 속성 정보(예, 윈도우 위치, z-순서, 크기, 스타일, 색상 등)를 포함한다. 어플리케이션-출력 윈도우(326)의 윈도우 속성 정보는 웹페이지(322)에 임베딩된 어플리케이션 객체(128)에 의해 초기에 지정된다. 예를 들어, 어플리케이션 객체(128)는 액티브X 제어를 포함할 수 있으며, 이는 클라이언트-서버 세션중에 어플리케이션-출력 윈도우(326)의 윈도우 속성을 지정하고 제어한다. 일실시예에서, 어플리케이션-출력 윈도우(326)는 대응하는 액티브X 제어와 동일한 치수를 표시한다.

클라이언트 에이전트(136)는 클라이언트 데스크탑(310)에 관련된 정보(310, 예, 크기, 해상도 등)와 함께 서버 에이전트(134)에 로컬 어플리케이션-출력 윈도우(130)의 초기 윈도우 속성값을 전달한다. 서버 에이전트(134)는 자신의 서버 데스크탑 크기를 클라이언트 데스크탑(310)에 맞추고 로컬 서버 윈도우의 윈도우 속성들을 클라이언트 데스크탑(310)상의 어플리케이션-출력 윈도우(326)의 속성들에 맞춤으로써 응답한다. 클라이언트 데스크탑(310)상의 어플리케이션-출력 윈도우(326)와 서버 데스크탑상의 서버 윈도우는 동일한 윈도우 속성을 표시하고 어플리케이션(130)에 의해 생성되는 그래픽 출력 데이터와 동일한 데이터를 디스플레이한다. 서버 데스크탑은 서버의 비디오 메모리내에 포함된 오프스크린 표면 또는 서 버(114)에 연결되는 디스플레이 장치상에 디스플레이되는 온스크린 표면에 대응한다.

클라이언트 장치(110)의 사용자는, 클라이언트 에이전트(136)에 의해 검출되고 서버 에이전트(134)에 전달되는 입력 이벤트를 입력함으로써, 클라이언트-서버 세션동안에 어플리케이션-출력 윈도우(326)를 이동시키거나 크기조절하고/하거나, z-순서 또는 기타 초기 윈도우 속성을 변경할 수 있다. 서버 에이전트(134)는 그의 데스크탑 및/또는 윈도우를 입력 이벤트에 일치하도록 맞춘 다음에, 입력 이벤트에 대응하여, 갱신된 그래픽 출력 데이터를 어플리케이션-출력 윈도우(326)들을 갱신하라는 명령과 함께 클라이언트 에이전트(136)에 전송함으로써, 윈도우(326)들이 서버(114)의 윈도우와 매치되도록 한다.

예를 들어, 클라이언트 장치(110)의 사용자가 어플리케이션-출력 윈도우(326)중 하나를 어플리케이션-출력 객체(128)에 의해 원래 지정된 크기로부터 (예를 들어, 마우스로 클릭하고 어플리케이션-출력 윈도우(326)의 경계선을 소정의 위치/크기로 드래깅함으로써) 조절한다면, 클라이언트 에이전트(136)는 마우스 동작에 의해 생성된 입력 이벤트를 검출하고 그것을 서버 에이전트(134)에 전달하며, 이는 서버(114)의 온/오프 스크린면에 동일한 크기조절 이벤트 결과를 낳는다. 그다음에, 서버 에이전트(134)는 클라이언트 에이전트(136)에 갱신된 그래픽 출력 데이터 및 윈도우 속성 정보와 함께 리페인트(repaint) 및 크기조절 명령 메시지를 전송한다. 응답하여, 클라이언트 에이전트(136)는 크기

조절 이벤트(예, 전술한 액티브X 제어)에 영향받은 적절한 어플리케이션 객체를 변경함으로써 대응 어플리케 이션-출력 윈도우(326)의 크기가 조절되고 갱신된 그래픽 출력 데이터가 어플리케이션-출력 윈도우(326)의 경계선내에 그려지도록 한다.

따라서, 본 발명은 어플리케이션-출력 윈도우(326)의 윈도우 속성이 변경되어 어플리케이션-출력 윈도우(326)가 브라우저 윈도우(320)의 경계선 내에서 이동되거나 크기 조절되는 등이 가능하도록 한다. 예로서, 도 4를 참조하면, 어플리케이션-출력 윈도우 B'(410)가 전술한 방법을 이용하여 크기조절되어 어플리케이션-출력 윈도우B''(420)를 형성할 수 있으며, 이는 어플리케이션-출력 윈도우 F(430)를 오버랩한다 (따라서, 상이한 z-순서 형태를 표시한다). 대안적으로, 어플리케이션-출력 윈도우(326)는 브라우저 윈도우(320)를 벗어나 확장되거나 완전히 외부에 위치하도록 이동되거나 크기조절될 수 있다. 예로서, 도 5를 참조하면, 어플리케이션-출력 윈도우 J(510)는 브라우저 윈도우(320)의 경계선 내부에 놓이고, 어플리케이션-출력 윈도우 K(520)는 브라우저 윈도우(320)의 경계선 밖으로 확장되고, 어플리케이션-출력 윈도우 L(530)는 브라우저 윈도우(320)를 완전히 벗어나 있다. 어플리케이션-출력 윈도우는 클라이언트 데스크탑(310)의 다른 요소들에 관하여 z-순서를 변화시켜 표시될 수 있다. 예를 들어, 로컬 윈도우(540)는 브라우저 윈도우(320)와 어플리케이션-출력 윈도우 L(530) 사이의 z-순서를 나타낼 수 있다. 이러한 실시예에서, 클라이언트 에이전트(136)는, 어플리케이션-출력 윈도우(326)의 윈도우 속성을 초기에 확립하였던 웹페이지(322)내에 임베딩된 어플리케이션 객체(128)의 속성을 먼저 변경할 필요없이, 클라이언트 장치(110)의 운영체제에 서버 에이전트(134)로부터 수신된 명령 메시지에 응답하여 소정의 어플리케이션-출 력 윈도우(326)를 그리도록 지시한다.

일실시예에서, 어플리케이션-출력 윈도우(326)에 영향을 미치는 각각의 입력 이벤트는 서버 에이전트(114)에 전달되어 처리되고, 서버 에이전트(114)는 클라이언트 에이전트(136)에 어플리케이션-출력 윈도우(326)에 해당 변경을 수행할 것을 지시한다. 또다른 실시예에서, 하나 이상의 입력 이벤트 유형(예, 어플리케이션-출력 윈도우(326)를 웹페이지(322)의 또다른 그리드 위치로 이동시키고자 하는 클릭 및 드래그 마우스 동작)이 클라이언트 에이전트(136)에 의해 전체적으로 처리되고 서버 에이전트(134)에 보고되지 않으며, 어플리케이션-출력 윈도우(326)내에 디스플레이된 그래픽 출력 데이터는 변경되지 않은 상태로 남는다.

좀더 상세히 도 6을 참조하면, 클라이언트 에이전트(136)는, 모니터 프로세스(610)와, 명령 프로세스(620)와, 메시지 수신 프로세스(630)와 메시지 전송 프로세스(640)를 포함한다. 일 실시예에서, 각 프로세스(610, 620, 630, 640)는 다른 프로세스들과 독립적으로 작동하는 개별적 기능 코드 세그먼트이다. 예를 들어, 메시지 수신 프로세스(630)와 명령 프로세스(620)는 별개의 스레드들로서 구현될 수 있고, 이들은 명명 파이프(named pipe) 또는 공유 메모리를 통해 서로 통신한다. 공통 데이터 집합을 이용함으로써 메시지 수신 프로세스(630)와 메시지 전송 프로세스(640)의 동기화를 허용한다.

메시지 수신 프로세스(630)는 클라이언트-서버 세션동안 클라이언트 에이전트(136)와 서버 에이전트(134)사이의 연결을 제공하는 통신 링크를 통해 서버 에이전트(134)로부터, 그래픽 데이터, 윈도우 속성 정보 및 명령을 수신한다. 통신 링 크는 바람직하게는 제1 가상 채널(660) 및 제2 가상 채널(670)을 포함한다. 명령, 이벤트 및 윈도우속성 정보는 제1 가상 채널(660)을 통해 클라이언트 에이전트(136)와 서버 에이전트(134)간에 전달되고, 어플리케이션-출력 윈도우(326)의 그래픽 콘텐트에 대응하는 그래픽 데이터는 제2 가상 채널(670)을 통해 전달된다. 메시지 수신 프로세스(630)는 서버 에이전트(134)로부터 수신된 명령, 윈도우 속성 및 그래픽 데이터를 알리고, 명령 프로세스(620)는 이러한 데이터를 더 처리한다.

일실시예에서, 명령 프로세스(620)는 클라이언트 운영 체제(650)에 서버 에이전트(134)에 의해 지정된 윈도우 속성에 따라 영향받는 어플리케이션-출력 윈도우(326)를 형성 및/또는 변경할 것을 지시함으로써 서버 에이전트(134)로부터 수신된 명령을 처리한다. 또한, 명령 프로세스(620)는 적절한 어플리케이션-출력 윈도우(326)에 서버 에이전트(134)에 의해 제공된 그래픽 데이터를 디스플레이할 것을 클라이언트 운영체제(650)에 지시한다. 일실시예에서, 명령 프로세스(620)는 GDI 명령을 송출함으로써 클라이언트 데스크탑(310)내에 어플리케이션-출력 윈도우(326)에 변경을 구현한다. 다른 실시예에서, 명령 프로세스(620)는 연관 그래픽 부시스템에 직접 또는 그래픽 API 어댑터를 통해 명령어를 송출한다.

명령 프로세스(620)는 모니터링 프로세스(610)에게 어플리케이션-출력 윈도우(326)에 영향을 주는 변화를 검출하기 위해 클라이언트 데스크탑(310)을 정기적으로 모니터링하도록 또한 지시한다. 일실시예에서, 모니터 프로세스(610)는 클라이언트 운영체제(650)에게 선정된 폴링(polling) 간격으로 클라이언트 데스크탑(310)에 관한 정보를 리턴할 것을 지시한다. 다른 실시예에서, 모니터 프로세스 (610)는 어플리케이션-출력 윈도우에 영향을 주는 변화를 검출하기 위해 클라이언트 운영체제(650)에 의해 유지되는 메시지 큐를 모니터링한다. 모니터 프로세스(610)는 검출된 변화의 일부 또는 전부를 이후 처리를 위해 명령 프로세스(620)에 전달한다.

일실시예에서, 명령 프로세스(620)는 메시지 전송 프로세스(640)에게 모니터 프로세스(610)에 의해 검출된 변화 모두를 제1 가상 채널을 통해 서버 에이전트(134)에 전송할 것을 지시한다. 다른 실시예에서, 명령 프로세스(620)는 메시지 전송 프로세스(640)에게, 어플리케이션-출력 윈도우(326)의 그래픽 데이터 및/또는 윈도우 속성에만 영향을 주는 변화와 같이, 검출된 변화의 부집합을 전송할 것을 지시한다. 서버 에이전트(134)는 명령 프로세스(620)로부터의

명령과 함께 검출된 변화 및 검출된 변화를 트리거한 클라이언트 장치(110)의 사용자에 의해 만들어진 임의의 입력 이벤트를 수신한다. 서버 에이전트(134)는 검출된 변화에 적합하게 그의 로컬 데스크탑을 수정하고, 연관 명령, 윈도우 속성 및 그래픽 데이터를 클라이언트 메시지 수신 프로세스(630)에 다시 전송한다. 이러한 방식으로, 본 발명은 클라이언트와 서버 데스크탑에 공통인, 어플리케이션-출력 윈도우(326)와 같은 데스크탑 요소가 로크(lock) 단계로 유지되도록 보장한다.

클라이언트 에이전트(136)의 명령 프로세스(620)는 공통 윈도우 리스트를 유지함으로써 클라이언트 및 서버 데스크탑의 유사/공통 요소들이 로크 상태로 유지되도록 보장한다. 공통 윈도우 리스트는 클라이언트 데스크탑(310)의 각 윈도우 및 서버 데스크탑의 각 대응 윈도우에 대한 윈도우 속성 정보를 포함한다. 일 실 시예에서, 다수의 클라이언트 에이전트(136)가 클라이언트 장치(110)상에서 실행되는 경우에, 단일 클라이언트 에이전트(136)의 명령 프로세스(620)는 공통 윈도우 리스트를 유지하기 위한 제1 책임을 갖는다. 단일 클라이언트 에이전트(136)가 종료하고 다른 클라이언트 에이전트들이 동작중에 있다면, 나머지 클라이언트 에이전트들은 공통 윈도우 리스트를 유지하기 위해 또 다른 제1 클라이언트 에이전트를 선택할 것이다.

본 발명은, 바람직하게 클라이언트 에이전트와 서버 에이전트간에 전송되는 그래픽 데이터량을 감소시키기 위해, 동일 출원인의 국제특허출원 제 WO01/92973(본 명세서에 전체적으로 참조됨)에 개시된 기술을 이용함으로써 클라이언트-서버 세션의 성능을 또한 증강시킨다. 일실시예에서, 본 발명은 클라이언트 에이전트(136)와 서버 에이전트(134)간에 전송된 프로토콜 스트림내에서 반복되는 비교적 작은 표현들로 데이터를 인코드한다. 이러한 방식으로, 본 발명은 전송되어야 하는 각 이산 데이터 요소의 크기를 최소화하고 프로토콜 스트림내에 데이터의 반복성을 증가시켜 반복적으로 인코딩된 데이터상에 더 효율적으로 작동하는 압축 알고리즘이 더 나은 수준의 압축 효율성을 실현할 수 있도록 한다. 본 발명은 이러한 개선된 압축을 위해 여러 기술들을 이용한다. 이러한 기술에는, 먼저-송출된 명령을 대체하는 나중-송출 명령어들을 위한 명령어 큐를 스캐닝하는 것, 클라이언트-세션중에 이미 처했던 데이터 압축을 개선시키는 디스크-캐싱 기술, 양끝점 및/또는 연관 스트립의 각도(즉, 공통 각도를 표시하는 일련의 연속적인 픽셀들)를 전송하는 것을 피하기 위해 양자화된 구역을 이용하는 상대적 좌표의 전송, 및 클라 이언트-서버 세션동안 성능을 증강시키기 위해 오프-스크린면을 조작하는 것을 포함한다. 또한, 본 발명은 고속 서버와 상대적으로 느린 네트워크 또는 클라이언트 장치간에 성능 부조화가 있는 경우에 처해지는 오버스크롤 문제점의 빈도수를 감소시킨다.

본 발명의 특정 실시예를 기술하였으나, 본 발명의 개념을 포함하는 다른 실시예들도 이용될 수 있음은 관련 기술분야의 당업자들에게 자명하다. 따라서, 본 발명은 특정 실시예에 국한되어서는 않되며, 이하의 청구항의 정신 및 범위에 의해서만 국한될 것이다.

(57) 청구의 범위

청구항 1.

웹브라우저에 어플리케이션-출력 데이터를 디스플레이하는 방법에 있어서, 상기 방법은,

웹브라우저 윈도우를 제공하는 단계와.

상기 웹브라우저내에 위치하는 적어도 하나의 어플리케이션-출력 윈도우를 제공하는 단계와,

상기 어플리케이션-출력 윈도우에 연결된 제1 가상 채널을 통해 상기 어플리케이션-출력 윈도우에 대응하는 윈도우 속성 정보- 상기 윈도우 속성 정보는 웹브라우저와 무관하게 변경될 수 있음-를 수신하는 단계와,

상기 어플리케이션-출력 윈도우에 연결된 제2 가상 채널을 통해 어플리케이션-출력 데이터를 수신하는 단계와,

상기 윈도우 속성 정보에 따라 상기 어플리케이션-출력 윈도우를 디스플레이하는 단계와,

상기 어플리케이션-출력 윈도우에 상기 어플리케이션-출력 데이터를 디스플레이하는 단계

를 포함하는 방법.

청구항 2.

제1항에 있어서, 상기 웹 브라우저와 무관하게, 상기 어플리케이션-출력 윈도우에 연관된 이벤트를 모니터링하고 상기 이벤트에 응답하는 클라이언트 에이전 트를 제공하는 단계를 더 포함하는 방법.

청구항 3.

제2항에 있어서, 상기 클라이언트 에이전트에 의해 수신된 입력에 응답하여 상기 어플리케이션-출력 윈도우의 크기를 상기 클라이언트 에이전트가 조절하는 단계를 더 포함하는 방법.

청구항 4.
제1항에 있어서, 상기 웹 브라우저와 무관하게, 상기 제1 및 제2 가상 채널을 개설하는 단계를 더 포함하는 방법.

청구항 5.
제1항에 있어서, 변경되지 않은 어플리케이션 프로그램에 의해 생성된 어플리케이션-출력 데이터를 수신하는 단계를 더 포함하는 방법.

청구항 6.
제1항에 있어서, 웹브라우저 윈도우에 디스플레이된 웹페이지에 임베딩된 어플리케이션 객체에 응답하여 클라이언트 에이전트를 실행시키는 단계를 더 포함하고, 상기 클라이언트 에이전트는 웹브라우저와 무관하게 제1 및 제2 가상 채널을 개설하는 방법.

청구항 7.
제6항에 있어서, 상기 어플리케이션 객체의 적어도 하나의 특성을 변경하기 위해 상기 수신된 윈도우 속성 정보를 이용하는 단계를 더 포함하고, 상기 어플리케이션 객체의 변경된 특성은 어플리케이션-출력 윈도우내에 대응 변화를 트리거하 는(triggering) 방법.

청구항 8.
제1항에 있어서, 상기 웹브라우저 윈도우내에 디스플레이되는 웹 페이지에 임베딩된 액티브X 제어에 응답하여 클라이언트 에이전트를 실행시키는 단계를 더 포함하고, 상기 클라이언트 에이전트는 웹브라우저와 무관하게 상기 제1 및 제2 가상 채널을 개설하는 방법.

청구항 9.
제1항에 있어서, 상기 웹브라우저 윈도우내에 위치한 다수의 어플리케이션-출력 윈도우를 제공하는 단계를 더 포함하고, 상기 어플리케이션-출력 윈도우 각각은 상이한 어플리케이션 서버로부터 수신된 어플리케이션-출력 데이터를 디스플레이하는 방법.

청구항 10.
웹브라우저내에 어플리케이션-출력 데이터를 디스플레이하기 위한 시스템에 있어서, 상기 시스템은,

웹브라우저 윈도우와,

상기 웹브라우저 윈도우내에 위치하여 상기 어플리케이션-출력 데이터를 디스플레이하는 적어도 하나의 어플리케이션-출력 윈도우와,

상기 어플리케이션-출력 윈도우에 대응하는 윈도우 속성 정보 - 상기 윈도우 속성 정보는 웹브라우저와 무관하게 변경가능함-를 수신하는 제1 가상 채널과,

상기 어플리케이션-출력 데이터 - 상기 어플리케이션-출력 데이터가 상기 윈 도우 속성 정보에 따라 상기 어플리케이션-출력 윈도우내에 디스프레이됨-를 수신하는 제2 가상 채널

을 포함하는 시스템.

청구항 11.
제10항에 있어서, 상기 어플리케이션-출력 윈도우에 연관된 이벤트를 모니터링하고 상기 이벤트에 응답하는 클라이언트 에이전트를 더 포함하는 시스템.

청구항 12.
제11항에 있어서, 상기 클라이언트 에이전트는 상기 클라이언트 에이전트에 의해 수신된 입력에 응답하여 상기 어플리케이션-출력 윈도우의 크기를 조절하는 시스템.

청구항 13.
제11항에 있어서, 상기 클라이언트 에이전트는 상기 웹브라우저와 무관하게 상기 제1 및 제2 가상 채널을 형성하는 시스템.

청구항 14.
제10항에 있어서, 상기 어플리케이션-출력 데이터를 생성하는 변경되지 않은 어플리케이션 프로그램을 더 포함하는 시스템.

청구항 15.
제10항에 있어서,

상기 웹브라우저 윈도우에 디스플레이된 웹페이지에 임베딩된 어플리케이션 객체 - 상기 어플리케이션 객체는 상기 어플리케이션-출력 윈도우에 영향을 미치는 적어도 하나의 특성을 포함함-와,

상기 어플리케이션 객체에 응하여 인스턴스화(instantiate)되고 상기 제2 가상 채널로부터 상기 윈도우 속성 정보를 수신한 후에 상기 어플리케이션 객체의 특성을 변경하는 클라이언트 에이전트

를 더 포함하는 시스템.

청구항 16.
제15항에 있어서, 상기 어플리케이션 객체는 액티브X 제어(ActiveX control)인 시스템.

청구항 17.
제10항에 있어서, 상기 어플리케이션-출력 윈도우는 상기 웹브라우저 윈도우의 자식 윈도우인 시스템.

청구항 18.
제10항에 있어서, 상기 웹브라우저 윈도우내에 위치한 복수의 어플리케이션-출력 윈도우 - 상기 어플리케이션-출력 윈도우의 각각은 상이한 어플리케이션 서버로부터 수신된 어플리케이션-출력 데이터를 디스플레이함-를 더 포함하는 시스템.

청구항 19.
원격 서버상에서 실행되는 어플리케이션 프로그램에 의해 생성된 어플리케이션-출력 데이터를 로컬 데스크탑에 디스플레이하는 방법에 있어서,

웹브라우저를 통해 적어도 하나의 어플리케이션 프로그램을 액세스하고자 하는 요청을 전송하는 단계와,

상기 어플리케이션 프로그램에 의해 생성된 어플리케이션-출력 데이터를 디스플레이하기 위한 적어도 하나의 어플리케이션-출력 윈도우를 제공하는 단계와,

상기 어플리케이션-출력 윈도우에 연결된 제1 채널을 통해 상기 어플리케이션-출력 윈도우에 대응하는 윈도우 속성 정보를 수신하는 단계와,

상기 어플리케이션-출력 윈도우에 연결된 제2 채널을 통해 어플리케이션-출력 데이터를 수신하는 단계와,

상기 제1 가상 채널을 통해 수신된 윈도우 속성 정보에 따라 상기 어플리케이션-출력 윈도우에 제2 채널을 통해 수신된 어플리케이션-출력 데이터를 디스플레이하는 단계

를 포함하는 방법.

청구항 20.
제19항에 있어서, 상기 웹 브라우저와 무관하게, 상기 어플리케이션-출력 윈도우에 연관된 이벤트를 모니터링하고 상기 이벤트에 응답하는 클라이언트 에이전트를 제공하는 단계를 더 포함하는 방법.

청구항 21.
제20항에 있어서, 상기 클라이언트 에이전트에 의해 수신된 입력에 응답하여 상기 어플리케이션-출력 윈도우의 크기를 상기 클라이언트 에이전트가 조절하는 단계를 더 포함하는 방법.

청구항 22.
제19항에 있어서, 상기 웹 브라우저와 무관하게, 상기 제1 및 제2 가상 채널 을 개설하는 단계를 더 포함하는 방법.

청구항 23.

제19항에 있어서, 상기 어플리케이션-출력 데이터를 생성하는 어플리케이션 프로그램은 변경되지 않은 어플리케이션 프로그램인 방법.

청구항 24.
제19항에 있어서, 웹브라우저에 디스플레이된 웹페이지에 임베딩된 어플리케이션 객체에 응답하여 클라이언트 에이전트를 실행시키는 단계를 더 포함하고, 상기 클라이언트 에이전트는 웹브라우저와 무관하게 제1 및 제2 가상 채널을 개설하는 방법.

청구항 25.
제24항에 있어서, 상기 어플리케이션 객체의 적어도 하나의 특성을 변경하기 위해 상기 수신된 윈도우 속성 정보를 이용하는 단계를 더 포함하고, 상기 어플리케이션 객체의 변경된 특성은 어플리케이션-출력 윈도우내에 대응 변화를 트리거하는 방법.

청구항 26.
제19항에 있어서, 상기 웹브라우저에 디스플레이되는 웹 페이지에 임베딩된 액티브X 제어에 응답하여 클라이언트 에이전트를 실행시키는 단계를 더 포함하고, 상기 클라이언트 에이전트는 웹브라우저와 무관하게 상기 제1 및 제2 가상 채널을 개설하는 방법.

청구항 27.
제19항에 있어서, 웹브라우저 윈도우내에 위치하는 복수의 어플리케이션-출력 윈도우를 제공하는 단계를 더 포함하고, 상기 어플리케이션-출력 윈도우의 적어도 일부는 상이한 원격 서버들상에 실행하는 어플리케이션 프로그램들에 의해 생성된 어플리케이션-출력 데이터를 디스플레이하는 방법.

청구항 28.
원격 서버상에서 실행되는 어플리케이션 프로그램에 의해 생성된 어플리케이션-출력 데이터를 로컬 데스크탑에 디스플레이하기 위한 시스템에 있어서,

적어도 하나의 어플리케이션 프로그램에 대한 액세스를 요청하는 웹브라우저와,

상기 어플리케이션 프로그램에 연관된 윈도우 속성 정보를 수신하는 제1 채널과,

상기 어플리케이션 프로그램에 의해 생성된 어플리케이션-출력 데이터를 수신하는 제2 채널과,

상기 제1 채널을 통해 수신된 상기 윈도우 속성 정보에 따라 상기 제2 채널을 통해 수신된 상기 어플리케이션-출력 데이터를 디스플레이하는 적어도 하나의 어플리케이션-출력 윈도우

를 포함하는 시스템.

청구항 29.
제28항에 있어서, 상기 어플리케이션-출력 윈도우에 연관된 이벤트를 모니터링하고 상기 이벤트에 응답하는 클라이언트 에이전트를 더 포함하는 시스템.

청구항 30.
제29항에 있어서, 상기 클라이언트 에이전트는 상기 클라이언트 에이전트에 의해 수신된 입력에 응답하여 상기 어플리케이션-출력 윈도우의 크기를 조절하는 시스템.

청구항 31.
제29항에 있어서, 상기 클라이언트 에이전트는 상기 웹브라우저와 무관하게 상기 제1 및 제2 가상 채널을 형성하는 시스템.

청구항 32.
제28항에 있어서, 상기 어플리케이션-출력 데이터를 생성하는 변경되지 않은 어플리케이션 프로그램을 더 포함하는 시스템.

청구항 33.
제28항에 있어서,

상기 웹브라우저 윈도우에 디스플레이된 웹페이지에 임베딩된 어플리케이션 객체 - 상기 어플리케이션 객체는 상기 어플리케이션-출력 윈도우에 영향을 미치는 적어도 하나의 특성을 포함함-와,

상기 어플리케이션 객체에 응하여 인스턴스화(instantiate)되고 상기 제2 가상 채널로부터 상기 윈도우 속성 정보를 수신한 후에 상기 어플리케이션 객체의 특성을 변경하는 클라이언트 에이전트

를 더 포함하는 시스템.

청구항 34.
제33항에 있어서, 상기 어플리케이션 객체는 액티브X 제어(ActiveX control)인 시스템.

청구항 35.
제28항에 있어서, 상기 어플리케이션-출력 윈도우는 상기 웹브라우저 윈도우의 자식 윈도우인 시스템.

청구항 36.
제28항에 있어서, 웹브라우저 윈도우내에 위치한 복수의 어플리케이션-출력 윈도우를 더 포함하고, 상기 어플리케이션-출력 윈도우의 적어도 일부는 상이한 원격 서버들상에서 실행되는 어플리케이션 프로그램들에 의해 생성되는 시스템.

청구항 37.
제19항에 있어서, 상기 제1 및 제2 가상 채널은 통신 링크의 가상 채널에 대응하는 방법.

청구항 38.
제19항에 있어서,

웹브라우저 윈도우를 형성하는 단계와,

상기 웹브라우저 윈도우의 경계선내에 상기 어플리케이션-출력 윈도우를 재배치가능하게 배치하는 단계

를 더 포함하는 방법.

청구항 39.
제19항에 있어서,

웹브라우저 윈도우를 형성하는 단계와,

상기 웹브라우저 윈도우의 경계선내에 상기 어플리케이션-출력 윈도우를 형성하는 단계와,

로컬 데스크탑에서 실행하는 클라이언트 에이전트에 의해 수신된 입력에 응답하여 상기 웹브라우저 윈도우의 경계선을 벗어나도록 상기 어플리케이션-출력 윈도우를 재배치하는 단계

를 더 포함하는 방법.

청구항 40.
제39항에 있어서, 상기 어플리케이션-출력 윈도우는 웹브라우저와 무관하게 재배치되는 방법.

청구항 41.
원격 컴퓨팅 장치상에서 생성된 그래픽 데이터를 로컬 데스크탑에 디스플레이하는 방법에 있어서,

브라우저에 의해 형성된 출력 윈도우 - 상기 출력 윈도우는 로컬 데스크탑의 브라우저 윈도우의 경계선내에 위치함-내에 상기 그래픽 데이터를 디스플레이하는 단계와,

브라우저와 무관하게 브라우저 윈도우의 경계선을 벗어나도록 상기 출력 윈도우를 재배치하는 단계

를 포함하는 방법.

청구항 42.

제28항에 있어서, 상기 제1 및 제2 채널은 통신 링크의 가상 채널에 대응하는 시스템.

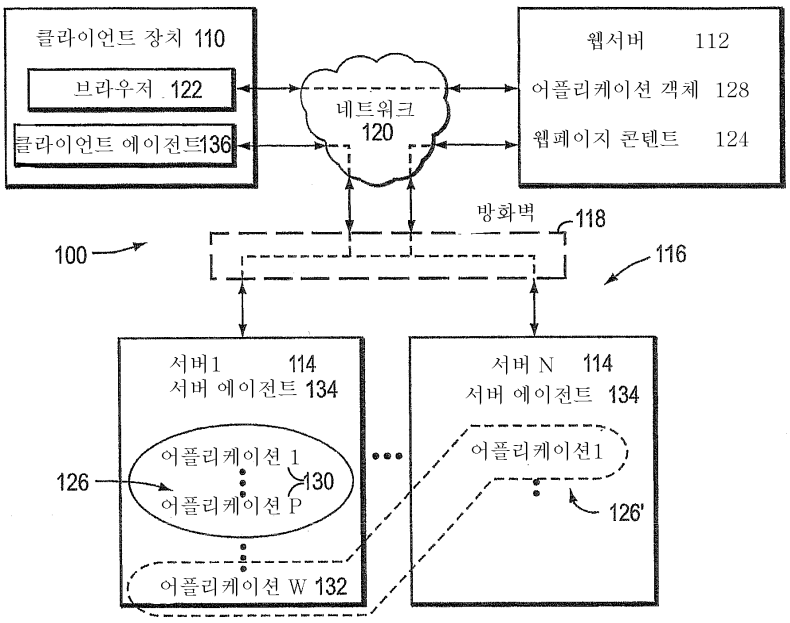
청구항 43.
제28항에 있어서, 상기 웹브라우저에 의해 형성된 웹브라우저 윈도우를 더 포함하고, 상기 어플리케이션-출력 위도우는 상기 웹브라우저 윈도우의 경계선내에 재배치가능하게 위치하는 시스템.

청구항 44.
제43항에 있어서, 상기 웹브라우저에 의해 인스턴스화되는 클라이언트 에이전트를 더 포함하고, 상기 클라이언트 에이전트는 상기 클라이언트 에이전트에 의해 수신된 입력에 응답하여 상기 웹브라우저 윈도우의 경계선을 벗어나도록 상기 어플리케이션-출력 윈도우를 재배치하는 시스템.

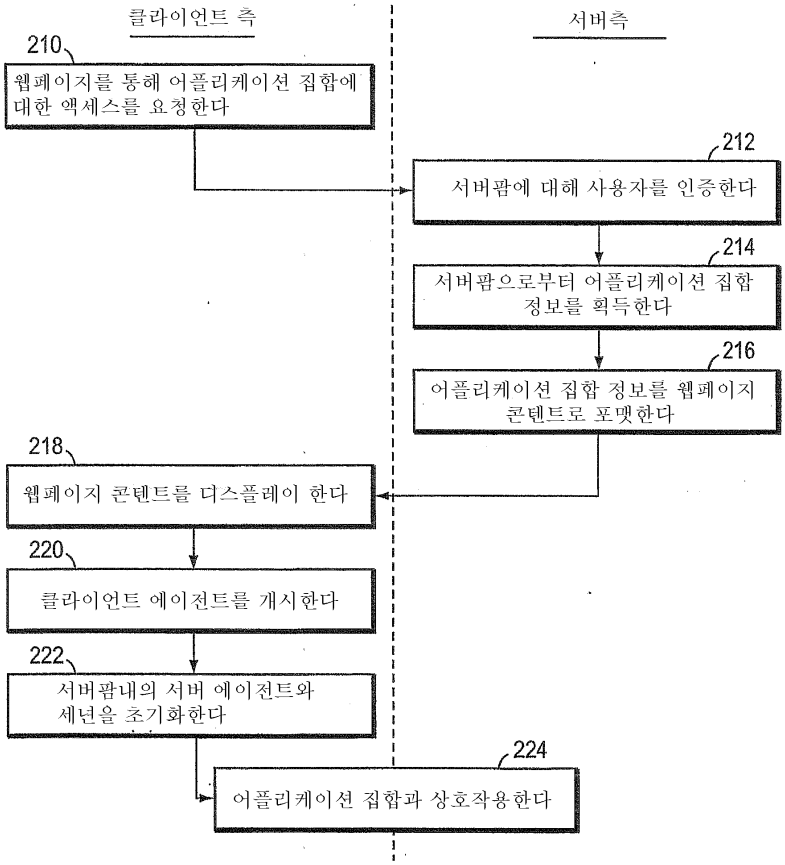
청구항 45.
제44항에 있어서, 상기 어플리케이션-출력 윈도우는 웹브라우저와 무관하게 재배치되는 시스템.

도면

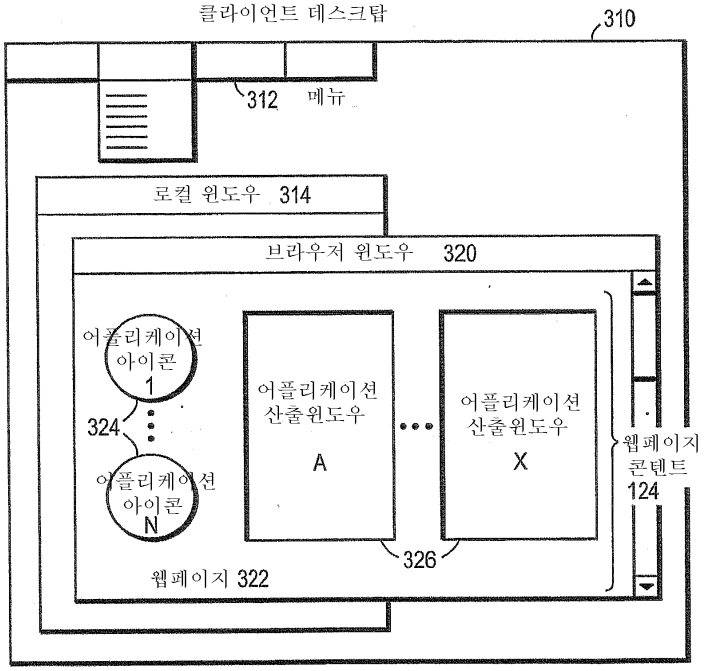
도면1



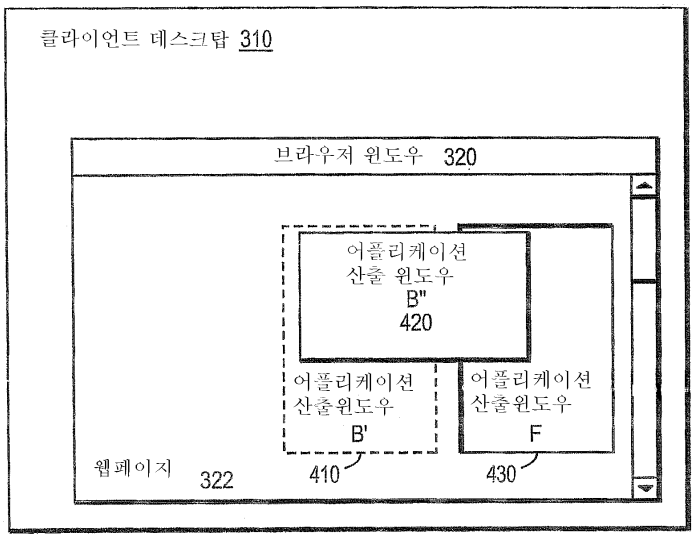
도면2


클라이언트 측
서버측
210
웹페이지를 통해 어플리케이션 집합에 대한 액세스를 요청한다
212
서버팜에 대해 사용자를 인증한다
214
서버팜으로부터 어플리케이션 집합 정보를 획득한다
216
어플리케이션 집합 정보를 웹페이지 콘텐트로 포맷한다
218
웹페이지 콘텐트를 디스플레이 한다
220
클라이언트 에이전트를 개시한다
222
서버팜내의 서버 에이전트와 세션을 초기화한다
224
어플리케이션 집합과 상호작용한다

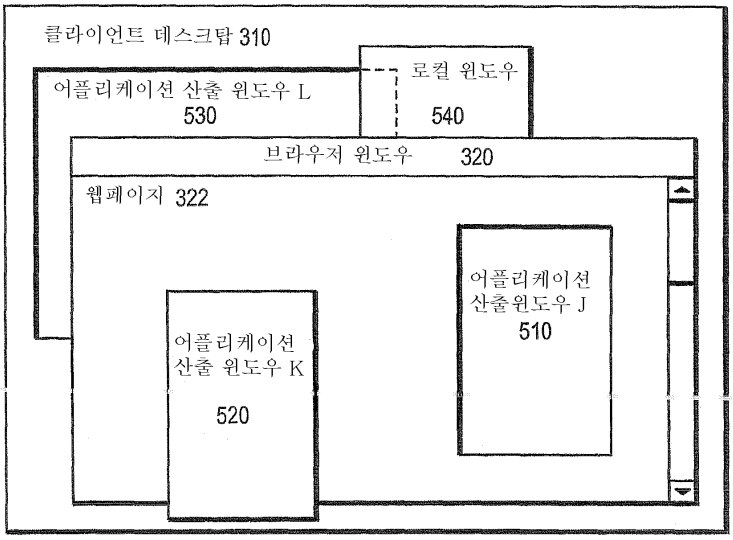
도면3


클라이언트 데스크탑
310
312
메뉴
로컬 윈도우 314
브라우저 윈도우 320
어플리케이션 아이콘 1
324
어플리케이션 아이콘 N
어플리케이션 산출윈도우 A
어플리케이션 산출윈도우 X
326
웹페이지 322
웹페이지 콘텐트 124

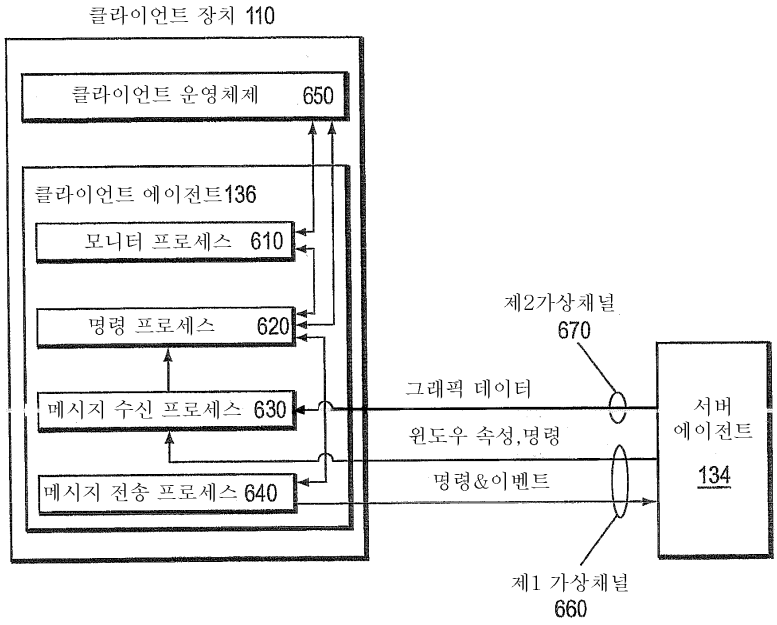
도면4


클라이언트 데스크탑 310
브라우저 윈도우 320
어플리케이션
산출 윈도우
B"
420
어플리케이션
산출윈도우
B'
410
어플리케이션
산출윈도우
F
430
웹페이지
322

도면5


클라이언트 데스크탑 310
어플리케이션 산출 윈도우 L
530
로컬 윈도우
540
브라우저 윈도우 320
웹페이지 322
어플리케이션
산출윈도우 J
510
어플리케이션
산출 윈도우 K
520

도면6


클라이언트 장치 110
클라이언트 운영체제 650
클라이언트 에이전트136
모니터 프로세스 610
명령 프로세스 620
메시지 수신 프로세스 630
메시지 전송 프로세스 640
제2가상채널
670
그래픽 데이터
윈도우 속성,명령
명령&이벤트
제1 가상채널
660
서버
에이전트
134