



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0001706  
(43) 공개일자 2008년01월03일

(51) Int. Cl.

G06F 9/06 (2006.01)

(21) 출원번호 10-2007-7018464

(22) 출원일자 2007년08월10일

심사청구일자 없음

번역문제출일자 2007년08월10일

(86) 국제출원번호 PCT/US2005/030312

국제출원일자 2005년08월25일

(87) 국제공개번호 WO 2006/107324

국제공개일자 2006년10월12일

(30) 우선권주장

11/094, 140 2005년03월31일 미국(US)

(71) 출원인

마이크로소프트 코포레이션

미국 위성턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이

(72) 별명자

스탭, 찰스, 더블유.

미국 98052-6399 위성턴주 레드몬드 원 마이크로  
소프트 웨이

린드세이, 도날드, 제이.

미국 98052-6399 위성턴주 레드몬드 원 마이크로  
소프트 웨이

리가메리, 마크, 알.

미국 98052-6399 위성턴주 레드몬드 원 마이크로  
소프트 웨이

(74) 대리인

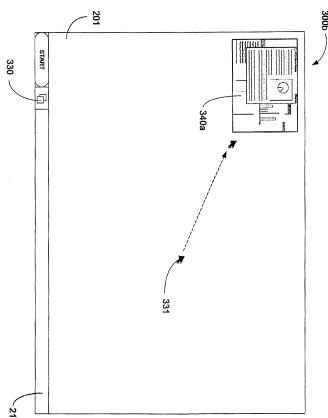
양영준, 백만기

전체 청구항 수 : 총 20 항

## (54) 애플리케이션 윈도우 그룹화 및 관리를 위한 방법 및 장치

**(57) 요약**

표시 스크린 상에 복수의 윈도우를 표시하기 위한 방법이 설명된다. 이 방법은 표시 스크린 상에 제1 애플리케이션 윈도우 및 제2 애플리케이션 윈도우를 표시하는 단계; 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 선언하는 단계 - 상기 제2 애플리케이션 윈도우는 상기 제1 애플리케이션 윈도우에 독립적이고 외부적임 -; 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 그룹화하는 단계; 및 상기 그룹을 상기 표시 스크린 상에 하나의 표현으로서 표시하는 단계를 포함한다. 새로운 애플리케이션 윈도우들이 열릴 때, 이들은 상기 그룹에 추가될 수 있고, 그리고/또는 애플리케이션 윈도우들의 제2 그룹의 멤버가 될 수 있다. 사용자는 다양한 그룹 사이에서 스위칭할 수 있다.

**대표도** - 도3B

## 특허청구의 범위

### 청구항 1

복수의 애플리케이션 윈도우를 그룹화하기 위한 방법으로서,

표시 스크린 상에 제1 애플리케이션 윈도우 및 제2 애플리케이션 윈도우를 표시하는 단계;

상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 선언하는 단계 - 상기 제2 애플리케이션 윈도우는 상기 제1 애플리케이션 윈도우에 독립적이고 외부적임 -;

상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 그룹화하는 단계; 및

상기 그룹을 상기 표시 스크린 상에 하나의 표현으로서 표시하는 단계

를 포함하는 방법.

### 청구항 2

제1항에 있어서, 상기 그룹 상에서 동작을 수행하는 단계를 더 포함하는 방법.

### 청구항 3

제2항에 있어서, 상기 수행하는 단계는 입력 디바이스로부터의 커맨드 수신에 기초하는 방법.

### 청구항 4

제2항에 있어서, 상기 동작을 수행하는 단계는 상기 그룹의 애플리케이션 윈도우들의 서브세트 상에서 수행되는 방법.

### 청구항 5

제1항에 있어서, 상기 제1 애플리케이션 윈도우는 제1 애플리케이션 프로그램에 대응하고, 상기 제2 애플리케이션 윈도우는 제2 애플리케이션 프로그램에 대응하는 방법.

### 청구항 6

제5항에 있어서, 상기 제1 애플리케이션 프로그램 및 상기 제2 애플리케이션 프로그램은 상이한 애플리케이션 프로그램인 방법.

### 청구항 7

제1항에 있어서,

상기 그룹으로부터 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 검색하기 위한 입력을 수신하는 단계; 및

상기 표시 스크린 상에서 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 오리지널 상태로 복원하는 단계

를 더 포함하는 방법.

### 청구항 8

제1항에 있어서, 상기 선언하는 단계는 입력 디바이스로부터 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 그룹화하기 위한 입력을 수신하는 단계를 포함하는 방법.

### 청구항 9

제1항에 있어서, 상기 표현은 상기 표시 스크린의 데스크톱 공간에 표시되는 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우의 축소 버전의 섬네일 이미지인 방법.

### 청구항 10

제1항에 있어서, 상기 표현은 상기 표시 스크린 상의 태스크바 영역 내의 태스크바 버튼인 방법.

### 청구항 11

제1항에 있어서,

적어도 하나의 제3 애플리케이션 윈도우를 표시하는 단계;

상기 적어도 하나의 제3 애플리케이션 윈도우를 하나의 그룹으로 선언하는 단계;

상기 적어도 하나의 제3 애플리케이션 윈도우를 제2 그룹으로 그룹화하는 단계; 및

상기 제2 그룹을 상기 표시 스크린 상에 제2 표현으로서 표시하는 단계

를 더 포함하는 방법.

### 청구항 12

제11항에 있어서,

상기 그룹으로부터 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 검색하기 위한 입력을 수신하는 단계; 및

상기 표시 스크린 상에서 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 오리지널 상태로 복원하는 단계

를 더 포함하는 방법.

### 청구항 13

제11항에 있어서,

상기 제2 그룹으로부터 상기 적어도 하나의 제3 애플리케이션 윈도우를 검색하기 위한 입력을 수신하는 단계; 및

상기 표시 스크린 상에서 상기 적어도 하나의 제3 애플리케이션 윈도우를 오리지널 상태로 복원하는 단계

를 더 포함하는 방법.

### 청구항 14

제1항에 있어서,

적어도 하나의 제3 애플리케이션 윈도우를 표시하는 단계;

상기 그룹으로부터 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 검색하기 위한 입력을 수신하는 단계;

상기 표시 스크린 상에서 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 오리지널 상태로 복원하는 단계;

상기 입력의 수신에 응답하여 상기 적어도 하나의 제3 애플리케이션 윈도우를 제2 그룹으로 그룹화하는 단계; 및

상기 제2 그룹을 상기 표시 스크린 상에 제2 표현으로서 표시하는 단계

를 더 포함하는 방법.

### 청구항 15

복수의 애플리케이션 윈도우를 그룹화하기 위한 일련의 컴퓨터 실행 가능 명령들을 포함하는 컴퓨터 판독 가능 매체로서,

상기 명령들은,

표시 스크린 상에 제1 애플리케이션 윈도우 및 제2 애플리케이션 윈도우를 표시하는 단계;

상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 선언하는 단계 - 상기 제2 애플리케이션 윈도우는 상기 제1 애플리케이션 윈도우에 독립적이고 외부적임 -;  
 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 그룹화하는 단계; 및  
 상기 그룹을 상기 표시 스크린 상에 하나의 표현으로서 표시하는 단계  
 를 수행하는 컴퓨터 판독 가능 매체.

#### 청구항 16

제15항에 있어서,

상기 명령들은,

상기 그룹으로부터 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 검색하기 위한 입력을 수신하는 단계; 및  
 상기 표시 스크린 상에서 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 오리지널 상태로 복원하는 단계  
 를 더 수행하는 컴퓨터 판독 가능 매체.

#### 청구항 17

제15항에 있어서,

상기 명령들은,

적어도 하나의 제3 애플리케이션 윈도우를 표시하는 단계;  
 상기 적어도 하나의 제3 애플리케이션 윈도우를 하나의 그룹으로 선언하는 단계;  
 상기 적어도 하나의 제3 애플리케이션 윈도우를 제2 그룹으로 그룹화하는 단계; 및  
 상기 제2 그룹을 상기 표시 스크린 상에 제2 표현으로서 표시하는 단계  
 를 더 수행하는 컴퓨터 판독 가능 매체.

#### 청구항 18

제15항에 있어서,

상기 명령들은,

적어도 하나의 제3 애플리케이션 윈도우를 표시하는 단계;  
 상기 그룹으로부터 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 검색하기 위한 입력을 수신하는 단계;  
 상기 표시 스크린 상에서 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 오리지널 상태로 복원하는 단계;  
 상기 입력의 수신에 응답하여 상기 적어도 하나의 제3 애플리케이션 윈도우를 제2 그룹으로 그룹화하는 단계;  
 및  
 상기 제2 그룹을 상기 표시 스크린 상에 제2 표현으로서 표시하는 단계  
 를 더 수행하는 컴퓨터 판독 가능 매체.

#### 청구항 19

그룹화된 윈도우들을 표현하는 데이터를 처리하기 위한 소프트웨어 아키텍처로서,

제1 애플리케이션 윈도우 및 제2 애플리케이션 윈도우를 하나의 그룹으로 그룹화하도록 구성된 적어도 하나의 컴포넌트 - 상기 그룹은 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우의 섬네일 이미지를 표

현함 -; 및

상기 컴포넌트에 액세스하기 위한 적어도 하나의 애플리케이션 프로그램 인터페이스  
를 포함하는 소프트웨어 아키텍처.

## 청구항 20

제19항에 있어서, 로그아웃 세션 동안 상기 그룹의 저장을 프로그램 방식으로 제어하도록 구성된 적어도 하나의  
컴포넌트를 더 포함하는 소프트웨어 아키텍처.

## 명세서

### 기술분야

<1> 본 발명의 양태들은 일반적으로 운영 체제에서의 윈도우 배열에 관한 것이다. 구체적으로, 본 발명의 양태들은 운영 체제에서 다수의 윈도우를 그룹화하고 소정의 지능적 거동이 윈도우들 사이에 존재할 수 있게 하는 방법 및 시스템에 관한 것이다.

### 배경기술

<2> 작업 및 개인 생활 양자에서의 컴퓨터의 사용이 증가함에 따라, 컴퓨터의 쉬운 사용이 요구되고 있다. 오늘 날, 많은 운영 체제는 애플리케이션 프로그램의 윈도우 기반 구성을 사용한다. 정보는 표시 스크린 상에서 여러 장의 종이처럼 보이는 것에 표시된다. 사용자는 윈도우와 인터페이스함으로써 한 장의 종이를 잡는 것처럼 임의의 윈도우에 액세스할 수 있다. 윈도우 기반 구성을 사용자가 표시 스크린 상에 둘 이상의 윈도우를 동시에 여는 것을 가능하게 한다.

<3> 애플리케이션 윈도우는 모든 그래픽 사용자 인터페이스(GUI) 시스템의 사용자 인터페이스 수단이다. 애플리케이션 윈도우들은 시스템들에서 외양은 다를 수 있지만, 크기가 조절되고 재배치될 수 있으며 상이한 애플리케이션들과 연관된 다른 애플리케이션 윈도우들 사이에 존재할 수 있는 능력과 같은 많은 공통 속성을 공유한다. 더불어, 다수의 애플리케이션 윈도우가 스크린 상에 동시에 나타날 수 있으며, 서로의 위에 계층화되고, 일반적으로 각각의 윈도우가 사용자에 의해 최종 액세스된 순서로 표현될 수 있다.

<4> 최근의 GUI들에 공통인 사용자 상호작용 시나리오는 공통 스크린 영역을 공유하는 다수의 동시 열림 애플리케이션 윈도우를 수반한다. 다수의 동시 열림 애플리케이션 윈도우에 대한 지원은 최근의 GUI의 매력 및 능력의 일부이지만, 이는 애플리케이션 윈도우들이 서로 중첩되어 가립으로써 사용자가 특정 애플리케이션 윈도우를 찾아내비게이트하는 것을 어렵게 하는 결과를 초래한다. 이러한 타입의 시나리오 및 관련 솔루션들은 윈도우 관리 문제 및 솔루션으로서 공통으로 지칭된다.

<5> 사용자들은 일반적으로 애플리케이션 윈도우를 이메일과 같은 태스크와 연관시키며, 또한 다수의 애플리케이션 윈도우와 단일 태스크 또는 프로젝트 사이에 정신적 연관성을 생성할 수 있다. 예를 들어, 슬라이드 쇼 프리젠테이션을 생성하기 위하여, 사용자는 가능하면 이메일, 그래픽 애플리케이션 및 스프레드시트 애플리케이션과 같은 하나 이상의 애플리케이션으로부터, 3개 또는 4개의 상이한 애플리케이션 윈도우로부터 콘텐츠를 요구할 수 있다. 다수의 애플리케이션 윈도우가 스크린 상에 보일 수 있으며, 각각의 애플리케이션 윈도우는 하나 이상의 태스크와 연관된다. 사용자는 콘텐츠에 액세스하거나 교환하여 태스크를 완수하기 위해서는 이를 상이한 애플리케이션 윈도우 사이에서 찾아내어 네비게이트해야 한다.

<6> 많은 GUI에는 공통적인 윈도우 관리 솔루션이 존재하며, 이를 통해 애플리케이션 윈도우는 최소화되고, 주요 작업 스크린 공간 내의 뷰로부터 제거될 수 있다. 최소화된 애플리케이션 윈도우는 일반적으로 훨씬 더 작은 UI 요소로 표현된다. 워싱턴 레드먼드의 마이크로소프트사의 윈도우 XP에서, 최소화된 애플리케이션 윈도우들은 태스크 바 제어와 같은 제어 바 상에 버튼으로 표현된다. 캘리포니아 쿠퍼티노의 애플 컴퓨터사의 맥 OS에서, 최소화된 애플리케이션 윈도우들은 도크 내에 미니어처 섬네일 버튼으로 표현된다. 애플리케이션 윈도우들의 최소화는 사용자가 불필요한 애플리케이션 윈도우들을 스크린 공간으로부터 제거하고 보다 작은 세트의 애플리케이션 윈도우들에 집중하는 것을 가능하게 한다. 그러나, 애플리케이션 윈도우들의 최소화는 개별적으로만 수행될 수 있다. 마찬가지로, 애플리케이션 윈도우들은 최소화된 상태에서 검색되고 가시적 애플리케이션 윈도우들의 세트로 다시 나타날 수 있지만, 이는 개별 애플리케이션 윈도우 기반으로만 행해질 수 있다.

<7> 현재, 독립적이고 외부적인 윈도우들은 함께 그룹화될 수 없으며, 따라서 이종 윈도우들은 함께 관리될 수

없다. 운영 체제들은 현재 다수의 윈도우가 서로 함께 그룹화되는 것을 허용하지 않으며, 따라서 하나의 윈도우 상에서 수행되는 동작이 제2의 독립적이고 외부적인 애플리케이션 윈도우 상에서 자동으로 수행될 수는 없다.

### 발명의 상세한 설명

- <8> 임의의 둘 이상의 애플리케이션 윈도우를 함께 그룹화하여 이종 윈도우들을 함께 관리할 수 있는 능력이 필요하다. 표시 스크린 상에 제1 애플리케이션 윈도우 및 제2 애플리케이션 윈도우를 표시하고, 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 선언하고 - 상기 제2 애플리케이션 윈도우는 상기 제1 애플리케이션 윈도우에 독립적이고 외부적임 -, 상기 제1 애플리케이션 윈도우 및 상기 제2 애플리케이션 윈도우를 하나의 그룹으로 그룹화하고, 상기 그룹을 상기 표시 스크린 상에 하나의 표현으로서 표시하기 위한 방법이 필요하다.
- <9> 본 발명의 양태들은 다수의 애플리케이션 윈도우를 하나의 그룹으로서 관리하는 것을 지원하기 위한 윈도우 관리 시스템을 제공한다. 본 발명의 양태들은 애플리케이션 윈도우들의 그룹을 생성하고, 이 그룹을 최소화하고, 이전에 식별된 그룹들이 검색되는 것을 가능하게 하는 새로운 방법 및 장치를 제공한다. 이 방법은 사용자들이 그들의 애플리케이션 윈도우-테스크 연관성을 정형화하고, 예를 들어 그룹을 생성하고, 이를 연관성을 관리하기 위한 수단, 예를 들어 애플리케이션 윈도우 그룹 아이콘을 제공하는 것을 가능하게 한다.

### 실시예

- <22> 아래의 다양한 실시예의 설명에서는, 그 일부를 구성하며 본 발명이 실시될 수 있는 다양한 실시예가 예시적으로 도시된 첨부 도면들을 참조한다. 다른 실시예들이 이용될 수 있으며, 본 발명의 범위로부터 벗어나지 않고 구조적, 기능적 변경이 이루어질 수 있음을 이해해야 한다.
- <23> 도 1은 본 발명이 구현될 수 있는 적절한 컴퓨팅 시스템 환경(100)의 일례를 나타낸다. 컴퓨팅 시스템 환경(100)은 적절한 컴퓨팅 환경의 일례일 뿐이며, 본 발명의 이용 또는 기능의 범위에 관한 임의의 제한을 제시하는 것을 의도하지 않는다. 컴퓨팅 시스템 환경(100)은 예시적인 컴퓨팅 시스템 환경(100)에 예시되는 컴퓨팅 모듈들 중 임의의 하나 또는 조합에 관한 어떠한 종속성도 요구도 갖지 않는 것으로 해석되어야 한다.
- <24> 본 발명은 다양한 다른 범용 또는 특수 컴퓨팅 시스템 환경 또는 구성에서 이용할 수 있다. 본 발명을 이용하는 데 적합할 수 있는 공지된 컴퓨팅 시스템, 환경 및/또는 구성의 예는 개인용 컴퓨터, 서버 컴퓨터, 핸드헬드 또는 랙톱 디바이스, 멀티프로세서 시스템, 마이크로프로세서 기반 시스템, 셋톱 박스, 프로그래머블 소비자 전자 장치, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 임의의 상기 시스템 또는 디바이스를 포함하는 분산형 컴퓨팅 환경 등을 포함하지만 이에 한정되지 않는다.
- <25> 본 발명은 일반적으로 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행 가능 명령들과 관련하여 설명될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 타입을 구현하는 루틴, 프로그램, 오브젝트, 컴퓨팅 모듈, 데이터 구조 등을 포함한다. 본 발명은 또한 통신 네트워크를 통해 링크되는 원격 처리 디바이스들에 의해 태스크들이 수행되는 분산형 컴퓨팅 환경에서 실시될 수 있다. 분산형 컴퓨팅 환경에서, 프로그램 모듈들은 메모리 저장 디바이스들을 포함하는 로컬 및 원격 컴퓨터 저장 매체 양자에 위치할 수 있다.
- <26> 도 1A를 참조하면, 본 발명을 구현하기 위한 예시적인 시스템은 컴퓨터(110)의 형태인 범용 컴퓨팅 디바이스를 포함한다. 컴퓨터(110)의 컴퓨팅 모듈들은 처리 유닛(120), 시스템 메모리(130), 및 시스템 메모리를 포함하는 다양한 시스템 컴퓨팅 모듈들을 처리 유닛(120)에 결합시키는 시스템 버스(121)를 포함할 수 있지만, 이에 한정되지 않는다. 시스템 버스(121)는 메모리 버스 또는 메모리 제어기, 주변 버스, 및 다양한 버스 아키텍처 중 임의의 아키텍처를 사용하는 로컬 버스를 포함하는 여러 타입의 버스 구조 중 임의의 구조일 수 있다. 예를 들어, 이러한 아키텍처는 ISA 버스, MCA 버스, EISA 버스, VESA 로컬 버스, 및 메자닌 버스로도 알려진 PCI 버스를 포함하지만 이에 한정되지 않는다.
- <27> 컴퓨터(110)는 일반적으로 다양한 컴퓨터 판독 가능 매체를 포함한다. 컴퓨터 판독 가능 매체는 컴퓨터(110)에 의해 액세스될 수 있는 임의의 가용 매체일 수 있으며, 휴대용 및 불휘발성 매체, 분리식 및 비분리식 매체를 포함한다. 예를 들어, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있지만 이에 한정되지 않는다. 컴퓨터 저장 매체는 컴퓨터 판독 가능 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 휴대용 및 불휘발성, 분리식 및 비분리식 매체를 포함

함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD 또는 다른 광학 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨터(110)에 의해 액세스될 수 있는 임의의 다른 매체를 포함하지만 이에 한정되지 않는다. 통신 매체는 일반적으로 컴퓨터 판독 가능 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터를 반송파 또는 다른 전송 메카니즘과 같은 변조된 데이터 신호 내에 합체시키며, 임의의 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는, 신호내의 정보가 암호화되도록 그 신호의 하나 이상의 특성을 설정 또는 변경시킨 신호를 의미한다. 예를 들어, 통신 매체는 유선 네트워크 또는 직접 유선 접속과 같은 유선 매체, 및 음향, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 포함하지만 이에 한정되지 않는다. 상기한 것들 중 임의의 것들의 조합들도 컴퓨터 판독 가능 매체의 범위 내에 포함되어야 한다.

<28> 시스템 메모리(130)는 ROM(131) 및 RAM(132)과 같은 휘발성 및/또는 불휘발성 메모리의 형태인 컴퓨터 저장 매체를 포함한다. 예를 들어 시동 동안 컴퓨터(110) 내의 요소들 사이에 정보를 전송하는 것을 돋는 기본 루틴을 포함하는 기본 입출력 시스템(133)(BIOS)은 일반적으로 ROM(131)에 저장된다. RAM(132)은 일반적으로 처리 유닛(120)에 의해 즉시 액세스되고 그리고/또는 현재 동작하고 있는 데이터 및/또는 프로그램 모듈을 포함한다. 예를 들어, 도 1A는 운영 체제(134), 애플리케이션 프로그램(135), 다른 프로그램 모듈(136) 및 프로그램 데이터(137)를 도시하지만 이에 한정되지 않는다.

<29> 컴퓨터(110)는 또한 다른 분리식/비분리식, 휘발성/불휘발성 컴퓨터 저장 매체를 포함할 수 있다. 단지 예로서, 도 1A는 비분리식, 불휘발성 자기 매체로부터 판독하고 그에 기입하는 하드 디스크 드라이브(141), 분리식, 불휘발성 자기 디스크(152)로부터 판독하고 그에 기입하는 자기 디스크 드라이브(151), 및 CD ROM 또는 다른 광학 매체와 같은 분리식, 불휘발성 광 디스크(156)로부터 판독하고 그에 기입하는 광 디스크 드라이브(155)를 도시하고 있다. 예시적인 동작 환경에서 사용될 수 있는 다른 분리식/비분리식, 휘발성/불휘발성 컴퓨터 저장 매체는 자기 테이프 카세트, 플래시 메모리 카드, DVD, 디지털 비디오 테이프, 반도체 RAM, 반도체 ROM 등을 포함하지만 이에 한정되지 않는다. 하드 디스크 드라이브(141)는 일반적으로 인터페이스(140)와 같은 비분리식 메모리 인터페이스를 통해 시스템 버스(121)에 접속되며, 자기 디스크 드라이브(151) 및 광 디스크 드라이브(155)는 일반적으로 인터페이스(150)와 같은 분리식 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

<30> 위에서 설명되었고 도 1A에 도시된 드라이브들 및 이들과 연관된 컴퓨터 저장 매체는 컴퓨터(110)를 위한 컴퓨터 판독 가능 명령, 데이터 구조, 프로그램 모듈 및 다른 데이터의 저장을 제공한다. 도 1A에서, 예를 들어, 하드 디스크 드라이브(141)는 운영 체제(144), 애플리케이션 프로그램(145), 다른 프로그램 모듈(146), 및 프로그램 데이터(147)를 저장하는 것으로 예시된다. 이들 컴포넌트는 운영 체제(134), 애플리케이션 프로그램(135), 다른 프로그램 모듈(136) 및 프로그램 데이터(137)와 동일하거나 다를 수 있다는 점에 유의한다. 운영 체제(144), 애플리케이션 프로그램(145), 다른 프로그램 모듈(146) 및 프로그램 데이터(147)는 본 명세서에서 최소한 이들이 상이한 사본들임을 나타내기 위해 상이한 번호를 갖는다. 사용자는 디지털 카메라(163), 키보드(162), 및 마우스 트랙볼 또는 터치 패드를 공통 지칭하는 포인팅 디바이스(161)와 같은 입력 디바이스를 통해 컴퓨터(110)에 커맨드 및 정보를 입력할 수 있다. 다른 입력 디바이스(도시되지 않음)는 웹, 스타일러스 및 태블릿, 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너 등을 포함할 수 있다. 이들 및 다른 입력 디바이스는 종종 시스템 버스(121)에 결합되는 사용자 입력 인터페이스(160)를 통해 처리 유닛(120)에 접속되지만, 병렬 포트, 게임 포트 또는 USB와 같은 다른 인터페이스 및 버스 구조에 의해 접속될 수도 있다. 모니터(191) 또는 다른 타입의 표시 디바이스도 비디오 인터페이스(190)와 같은 인터페이스를 통해 시스템 버스(121)에 접속된다. 모니터 외에도, 컴퓨터는 출력 주변 인터페이스(195)를 통해 접속될 수 있는 스피커(197) 및 프린터(196)와 같은 다른 주변 출력 디바이스를 더 포함할 수 있다.

<31> 컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터에 대한 논리 접속을 이용하여 네트워크 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 개인용 컴퓨터, 서버, 라우터, 네트워크 PC, 피어 디바이스 또는 다른 공통 네트워크 노드일 수 있으며, 도 1A에는 메모리 저장 디바이스(181)만이 예시되어 있지만, 일반적으로 컴퓨터(110)와 관련하여 전술한 요소들의 대부분 또는 모두를 포함한다. 도 1A에 도시된 논리 접속은 LAN(171) 및 WAN(173)을 포함하지만, 다른 네트워크를 포함할 수도 있다. 이러한 네트워킹 환경은 사무실, 기업 광역 컴퓨터 네트워크, 인트라넷 및 인터넷에서 일반적이다.

<32> LAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 일반적으로 인터넷과 같은 WAN(173)을 통한 통신을 설정하기 위한 모뎀(172) 또는 다른 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(172)은 사용자 입력

인터페이스(160) 또는 다른 적절한 메카니즘을 통해 시스템 버스(121)에 접속될 수 있다. 네트워크 환경에서, 컴퓨터(110)와 관련하여 도시된 프로그램 모듈들 및 그 일부는 원격 메모리 저장 장치에 저장될 수 있다. 예를 들어, 도 1A는 원격 애플리케이션 프로그램들(185)이 메모리 장치(181)에 위치하는 것으로 도시하고 있지만 이에 한정되지 않는다. 도시된 네트워크 접속은 예시적이며, 컴퓨터들 사이에 통신 링크를 설정하는 다른 수단이 사용될 수도 있음을 이해할 것이다.

<33> 도시된 네트워크 접속은 예시적이며, 컴퓨터들 사이에 통신 링크를 설정하는 다른 수단이 사용될 수 있음을 이해할 것이다. TCP/IP, 이더넷, FTP, HTTP 등과 같은 임의의 다양한 공지 프로토콜의 존재를 가정하며, 시스템은 클라이언트-서버 구성에서 동작하여, 사용자가 웹 기반 서버로부터 웹 페이지를 검색하는 것을 허가할 수 있다. 임의의 다양한 통상의 웹 브라우저를 이용하여 웹 페이지 상에 데이터를 표시하고 조작할 수 있다.

<34> 프로그래밍 인터페이스(또는 더 간단하게는 인터페이스)는 코드의 하나 이상의 세그먼트(들)가 코드의 하나 이상의 다른 세그먼트(들)에 의해 제공되는 기능과 통신하고 그에 액세스할 수 있게 하는 임의의 메카니즘, 프로세스, 프로토콜로서 보여질 수 있다. 대안으로, 프로그래밍 인터페이스는 시스템의 다른 컴포넌트(들)의 하나 이상의 메카니즘(들), 메소드(들), 함수 호출(들), 모듈(들) 등에 대해 통신 결합할 수 있는 시스템의 하나의 컴포넌트의 하나 이상의 메카니즘(들), 메소드(들), 함수 호출(들) 모듈(들), 오브젝트(들) 등으로 보여질 수 있다. 앞 문장에서 "코드의 세그먼트"라는 용어는 하나 이상의 명령 또는 코드 라인을 포함하는 것을 의도하며, 적용되는 용어법에 관계없이, 또는 코드 세그먼트들이 개별적으로 컴파일되는지에 관계없이, 또는 코드 세그먼트들이 소스, 중간 또는 오브젝트 코드로서 제공되는지에 관계없이, 코드 세그먼트들이 실행시간 시스템 또는 프로세스에서 사용되는지에 관계없이, 또는 이들이 동일 또는 상이한 머신들 상에 위치하는지 다수의 머신들에 분산되는지에 관계없이, 또는 코드 세그먼트들에 의해 표현되는 기능이 소프트웨어에서 전적으로, 하드웨어에서 전적으로, 또는 하드웨어 및 소프트웨어의 조합으로 구현되는지에 관계없이, 예를 들어 코드 모듈, 오브젝트, 서브루틴, 함수 등을 포함한다.

<35> 개념적으로, 프로그래밍 인터페이스는 일반적으로 도 1B 또는 도 1C에 도시된 바와 같이 보여질 수 있다. 도 1B는 인터페이스 Interface1을, 제1 및 제2 코드 세그먼트가 통신할 수 있는 콘 MERCHANTABILITY으로 도시하고 있다. 도 1C는 시스템의 제1 및 제2 코드 세그먼트가 매체(M)를 통해 통신할 수 있게 하는 인터페이스 오브젝트들(I1, I2)(이는 제1 및 제2 코드 세그먼트의 일부이거나 일부가 아닐 수 있다)를 포함하는 것으로서 인터페이스를 도시하고 있다. 도 1C를 참조할 때, 인터페이스 오브젝트들(I1, I2)을 동일 시스템의 개별 인터페이스들로서 간주할 수 있으며, 또한 오브젝트들(I1, I2) 플러스 매체(M)가 인터페이스를 포함하는 것으로 간주할 수 있다. 도 1B 및 1C는 양방향 흐름 및 흐름의 각 측면 상의 인터페이스를 도시하고 있지만, 소정의 구현들은 한 방향의 정보 흐름만을 가질 수 있거나(또는 후술하는 바와 같이 어떠한 정보 흐름도 갖지 못함), 한 측면 상의 인터페이스 오브젝트만을 가질 수 있다. 예를 들어, API, 엔트리 포인트, 메소드, 함수, 서브루틴, 원격 프로시저 호출 및 컴포넌트 오브젝트 모델(COM) 인터페이스와 같은 용어들은 프로그래밍 인터페이스의 정의 내에 포함되지만 이에 한정되지 않는다.

<36> 이러한 프로그래밍 인터페이스의 양태들은 제1 코드 세그먼트가 제2 코드 세그먼트로 정보(여기서, "정보"는 가장 넓은 의미로 사용되며, 데이터, 커맨드, 요청 등을 포함한다)를 전송하는 메소드, 제2 코드 세그먼트가 정보를 수신하는 메소드, 및 정보의 구조, 시퀀스, 신택스, 조직, 스키마, 타이밍 및 콘텐츠를 포함할 수 있다. 이와 관련하여, 기반 전송 매체 자체는 정보가 인터페이스에 의해 정의되는 방식으로 전송되는 한, 매체가 유선 또는 무선, 또는 이들의 조합인 것에 관계없이 인터페이스의 동작에 중요하지 않을 수 있다. 소정의 상황에서, 하나의 코드 세그먼트가 제2 코드 세그먼트에 의해 제공되는 기능에 간단히 액세스할 때와 같이, 정보 전달이 다른 메카니즘을 통해 이루어지거나(예를 들어, 코드 세그먼트들 간의 정보 흐름으로부터 분리되어 정보가 버퍼, 파일 등에 위치), 존재하지 않을 수 있게 됨에 따라, 정보는 통상적으로 하나 또는 양 방향으로 전달되지 않을 수 있다. 이를 양태의 일부 또는 모두는 소정의 상황에서, 예를 들어 코드 세그먼트들이 느슨하게 결합되는 구성을 또는 단단히 결합되는 구성을 시스템의 일부인지에 따라 중요할 수 있으며, 따라서 이 리스트는 예시적이고 비제한적인 것으로 간주되어야 한다.

<37> 이러한 프로그래밍 인터페이스의 개념은 이 분야의 전문가에게 공지되어 있으며, 전술한 발명의 상세한 설명으로부터 명백하다. 그러나, 프로그래밍 인터페이스를 구현하는 다른 방법들이 존재하며, 명시적으로 배제되지 않는 한, 이들도 본 명세서의 끝에 설명되는 청구범위에 포함되는 것을 의도한다. 그러한 다른 방법들은 도 1B 및 1C에 간단하게 보여지는 것보다 정교하거나 복잡하게 보일 수 있지만, 그럼에도 이들은 동일한 전체적 결과를 달성하도록 동일한 기능을 수행한다. 이제, 프로그래밍 인터페이스의 몇몇 예시적인 대안적 구현을 간단히

설명한다.

&lt;38&gt;

#### A. 팩터링

&lt;39&gt;

하나의 코드 세그먼트에서 다른 코드 세그먼트로의 통신은 통신을 다수의 개별 통신으로 분할함으로써 간접적으로 달성될 수 있다. 이것은 도 1D 및 1E에 개략적으로 도시되어 있다. 도시된 바와 같이, 몇몇 인터페이스는 분할 가능한 기능 세트들로 설명될 수 있다. 따라서, 도 1B 및 1C의 인터페이스 기능은 수학적으로 24, 또는 2x2x3x2를 제공할 수 있는 것과 같이 동일 결과를 달성하도록 팩터링될 수 있다. 따라서, 도 1D에 도시된 바와 같이, 인터페이스 Interface1에 의해 제공되는 기능은 동일 결과를 달성하면서 인터페이스의 통신을 다수의 인터페이스 Interface1A, Interface1B, Interface1C 등으로 변환하도록 세분될 수 있다. 도 1E에 도시된 바와 같이, 인터페이스(I1)에 의해 제공되는 기능은 동일 결과를 달성하면서 다수의 인터페이스(I1a, I1b, I1c 등)로 세분될 수 있다. 마찬가지로, 제1 코드 세그먼트로부터 정보를 수신하는 제2 코드 세그먼트의 인터페이스(I2)는 다수의 인터페이스(I2a, I2b, I2c 등)로 팩터링될 수 있다. 팩터링시, 제1 코드 세그먼트에 포함되는 인터페이스의 수는 제2 코드 세그먼트에 포함되는 인터페이스의 수와 일치할 필요는 없다. 도 1D 및 1E의 경우들 중 어느 하나에서, 인터페이스들 Interface1 및 I1의 기능적 사상은 각각 도 1B 및 1C와 동일하게 유지된다. 인터페이스들의 팩터링은 또한 결합, 교환 및 다른 수학적 특성을 따를 수 있으며, 따라서 팩터링은 인식하기 어려울 수도 있다. 예를 들어, 연산들의 순서는 중요하지 않을 수 있으며, 결과적으로 인터페이스에 의해 수행되는 기능은 다른 코드 또는 인터페이스에 의해 인터페이스에 도달하기 전에 양호하게 수행되거나, 시스템의 개별 컴포넌트에 의해 수행될 수 있다. 더욱이, 프로그래밍 분야의 전문가는 동일한 결과를 달성하는 상이한 함수 호출들을 행하는 다양한 방법이 존재한다는 것을 알 수 있다.

&lt;40&gt;

#### B. 재정의

&lt;41&gt;

몇몇 경우에, 의도한 결과를 계속 달성하면서 프로그래밍 인터페이스의 소정 양태들(예를 들어, 파라미터들)을 무시, 추가 또는 재정의하는 것이 가능할 수 있다. 이것은 도 1F 및 1G에 도시되어 있다. 예를 들어, 도 1B의 인터페이스 Interface1이 3개의 파라미터, 즉 입력, 정밀도 및 출력을 포함하고 제1 코드 세그먼트에서 제2 코드 세그먼트로 발행되는 호출인 함수 호출 제곱(입력, 정밀도, 출력)을 포함하는 것으로 가정한다. 중간 파라미터인 정밀도가 도 1F에 도시된 바와 같이 주어진 시나리오에서 중요하지 않은 경우, 이는 무시되거나 심지어 의미 없는(이 상황에서) 파라미터로 대체되는 편이 낫다. 또한, 중요하지 않은 추가 파라미터를 추가할 수도 있다. 어느 경우에나, 입력이 제2 코드 세그먼트에 의해 제곱된 후에 출력이 반환되는 한, 제곱의 기능이 달성될 수 있다. 정밀도는 컴퓨터 시스템의 소정의 다운스트림 또는 다른 부분에 대해 매우 양호하게 의미 있는 파라미터일 수도 있으나, 정밀도가 제곱을 계산하는 좁은 목적에는 필요하지 않은 것으로 인식되는 경우, 이는 대체 또는 무시될 수 있다. 예를 들어, 유효 정밀도 값을 전달하는 대신에, 결과에 악영향을 미치지 않고 생일과 같은 의미 없는 값이 전달될 수 있다. 마찬가지로, 도 1G에 도시된 바와 같이, 인터페이스(I1)는 파라미터를 무시하거나 인터페이스에 추가하도록 재정의된 인터페이스(I1')로 대체된다. 마찬가지로, 인터페이스(I2)는 불필요한 파라미터, 또는 다른 곳에서 처리될 수 있는 파라미터를 무시하도록 재정의되는 인터페이스(I2')로서 재정의될 수 있다. 여기서, 요점은 몇몇 경우에 프로그래밍 인터페이스가 소정 목적에 필요하지 않은 파라미터들과 같은 양태들을 포함할 수 있으며, 따라서 이들은 무시 또는 재정의되거나, 다른 곳에서 다른 목적으로 처리될 수 있다는 것이다.

&lt;42&gt;

#### C. 인라인 코딩

&lt;43&gt;

2개의 개별 코딩 모듈의 기능의 일부 또는 모두를 병합하여 이들 간의 "인터페이스"가 형태를 변경하게 하는 것이 또한 가능할 수 있다. 예를 들어, 도 1B 및 1C의 기능은 각각 도 1H 및 1I의 기능으로 변환될 수 있다. 도 1H에서, 도 1B의 이전의 제1 및 제2 코드 세그먼트들은 이를 양자를 포함하는 모듈로 합체된다. 이 경우, 코드 세그먼트들은 여전히 서로 통신할 수 있지만, 인터페이스는 단일 모듈에 보다 적절한 형태로 적응될 수 있다. 따라서, 예를 들어, 형식적인 호출 및 반환 명령문은 더 이상 필요하지 않을 수 있지만, 인터페이스 Interface1에 따른 유사한 처리 또는 응답(들)이 여전히 유효할 수 있다. 마찬가지로, 도 1I에 도시된 바와 같이, 도 1C로부터의 인터페이스(I2)의 일부(또는 모두)는 인터페이스(I1)로 인라인 기입되어 인터페이스(I1")를 형성할 수 있다. 도시된 바와 같이, 인터페이스(I2)는 I2a 및 I2b로 분할되며, 인터페이스 부분(I2a)은 인터페이스(I1)과 함께 인라인 코딩되어 인터페이스(I1")를 형성하였다. 구체적인 예로서, 도 1C로부터의 인터페이스(I1)가, 인터페이스(I2)에 의해 수신되는 것으로서, 제2 코드 세그먼트에 의해 입력과 함께(이를 제곱하기 위해) 전달되는 값을 처리한 후에 제곱한 결과를 출력과 함께 역 전달하는 함수 호출 제곱(입력, 출력)을 수행하는 것을 고려한다. 이 경우, 제2 코드 세그먼트에 의해 수행되는 처리(입력의 제곱)는 인터페이스에 대한 호출 없이 제1 코드

세그먼트에 의해 수행될 수 있다.

&lt;44&gt;

#### D. 분리

&lt;45&gt;

하나의 코드 세그먼트에서 다른 코드 세그먼트로의 통신은 통신을 다수의 개별 통신으로 분할함으로써 간접적으로 달성될 수 있다. 이것은 도 1J 및 1K에 개략적으로 도시되어 있다. 도 1J에 도시된 바와 같이, 제1 인터페이스 Interface1 상에서의 통신들을, 이들이 상이한 인터페이스, 이 경우에는 인터페이스들 Interface2A, Interface2B 및 Interface2C를 따르도록 변환하기 위한 하나 이상의 미들웨어(이는 오리지널 인터페이스로부터 기능 및/또는 인터페이스 기능들을 분리하므로 분리 인터페이스(들)이다)가 제공된다. 이것은 예를 들어 Interface1 프로토콜에 따라 운영 체제와 통신하도록 설계된 애플리케이션들의 설치 기반이 존재하지만, 이후 운영 체제가 상이한 인터페이스, 이 경우에는 인터페이스들 Interface2A, Interface2B 및 Interface2C를 이용하도록 변경되는 경우에 행해질 수 있다. 요점은 제2 코드 세그먼트에 의해 사용되는 오리지널 인터페이스가 변경되어 제1 코드 세그먼트에 의해 사용되는 인터페이스와 더 이상 호환되지 않고, 따라서 구형 및 신형 인터페이스들이 호환될 수 있도록 매개체가 사용된다는 점이다. 마찬가지로, 도 1K에 도시된 바와 같이, 제3 코드 세그먼트가 인터페이스(I1)로부터 통신을 수신하기 위한 분리 인터페이스(DI1)과 함께, 그리고 인터페이스 기능을 예를 들어 DI2와 함께 동작하도록 재정의된 인터페이스들(I2a, I2b)로 전송하지만 동일한 기능적 결과를 제공하는 분리 인터페이스(DI2)와 함께 도입될 수 있다. 마찬가지로, DI1 및 DI2는 동일하거나 유사한 기능적 결과를 제공하면서도 1C의 인터페이스들(I1, I2)의 기능을 새로운 운영 체제로 전달하도록 함께 동작할 수 있다.

&lt;46&gt;

#### E. 재작성

&lt;47&gt;

또 다른 가능한 변형은 인터페이스 기능을, 그 밖의 다른, 그러나 동일한 전체적인 결과를 달성하는 무언가로 대체하도록 코드를 동적으로 재작성하는 것이다. 예를 들어, 중간 언어(예를 들어, 마이크로소프트 IL, 자바 바이트코드 등)로 표시되는 코드 세그먼트가 실행 환경(.NET 프레임워크에 의해 제공되는 환경, 자바 실행시간 환경, 또는 다른 유사한 실행 시간 타입 환경 등)에서 JIT(Just-in-Time) 컴파일러 또는 인터프리터로 제공되는 시스템이 존재할 수 있다. JIT 컴파일러는 제1 코드 세그먼트에서 제2 코드 세그먼트로의 통신들을, 이들이 제2 코드 세그먼트(오리지널 또는 상이한 제2 코드 세그먼트)에 의해 요구될 수 있는 바와 같은 상이한 인터페이스를 따르도록 동적으로 변환하도록 작성될 수 있다. 이것은 도 1L 및 1M에 도시되어 있다. 도 1L에 도시된 바와 같이, 이러한 접근법은 전술한 분리 시나리오와 유사하다. 이것은 예를 들어 애플리케이션들의 설치 기반이 Interface1 프로토콜에 따라 운영 체제와 통신하도록 설계되지만, 이후 운영 체제가 상이한 인터페이스를 사용하도록 변경되는 경우에 행해질 수 있다. JIT 컴파일러는 설치 기반 애플리케이션들로부터의 온더플라이 통신들이 운영 체제의 새로운 인터페이스를 따르게 하는 데 사용될 수 있다. 도 1M에 도시된 바와 같이, 인터페이스(들)를 동적으로 재작성하는 이러한 접근법은 인터페이스(들)을 또한 동적으로 팩터링하거나 변경하기 위해 적용될 수 있다.

&lt;48&gt;

또한, 대안적인 실시예들을 통해 동일 또는 유사한 결과를 하나의 인터페이스로서 달성하기 위한 전술한 시나리오들은 다양한 방식으로, 직렬로, 그리고/또는 병렬로, 또는 다른 중간 코드를 이용하여 조합될 수도 있다는 점에 유의한다. 따라서, 전술한 대안적인 실시예들은 서로 배타적인 것이 아니라, 혼합, 매칭 및 조합되어 도 1B 및 1C에 제시된 일반 시나리오와 동일하거나 동등한 시나리오를 생성할 수 있다. 또한, 대부분의 프로그래밍 구조에서와 같이, 본 명세서에서 설명되지 않을 수 있으나 그럼에도 본 발명의 사상 및 범위에 의해 표현되는 인터페이스의 동일 또는 유사한 기능을 달성하는 다른 유사한 방법들이 존재한다는 점에 유의해야 하는데, 즉 이것은 적어도 부분적으로는, 하나의 인터페이스의 가치의 기초가 되는 인터페이스에 의해 표현되는 기능, 및 그에 의해 가능해지는 이로운 결과들이라는 점에 유의해야 한다.

&lt;49&gt;

도 2는 그래픽 사용자 인터페이스 시스템들에 공통인 통상의 윈도우 관리 시나리오(200)를 나타낸다. 도시된 바와 같이, 다수의 열린 애플리케이션 윈도우는 정면에 위치하는 다른 보다 큰 윈도우들에 의해 부분적으로 또는 완전히 가려져 있다. 구체적으로, 도 2는 애플리케이션 윈도우(203d)가 윈도우들(203a, 203b, 203c)보다 Z 순위가 높은 Z 순위 구성에서의 4개의 윈도우(203a-203d)의 시나리오(200)를 나타낸다. 애플리케이션 윈도우들(203a-203d)은 데스크톱 공간(201) 내에 도시되어 있다.

&lt;50&gt;

데스크톱 공간(201)은 애플리케이션 프로그램들에 대응하는 윈도우들의 표시를 허가하는 표시 영역이다. 하부의 태스크바(210)는 어느 애플리케이션 윈도우들이 현재 사용되고 있는지를 나타내는데, 이들은 보여지거나 최소화될 수 있다. 태스크바(210)는 윈도우들을 나열하고, 활성화, 이동, 숨김 및 최소화와 같은 윈도우들의 조작을 가능하게 하는 데 사용되는 온스크린 윈도우 원격 제어의 특정 구현이다. 윈도우(203a)는 태스크바 버튼

(213a)에 의해 표시되고, 윈도우(203b)는 태스크바 버튼(213b)에 의해, 윈도우(203c)는 태스크바 버튼(213c)에 의해, 윈도우(203d)는 태스크바 버튼(213d)에 의해 표시된다. 이 예에서 도시된 바와 같이, 4개의 윈도우(203a-203d)가 열려 있는 것으로 도시되어 있다. 윈도우들(203a-203d)의 콘텐츠의 파일명은 각각의 윈도우의 타이틀 바 영역(223a-223d)을 따라 도시된다. 이를 파일명은 윈도우를 동작시키는 애플리케이션 프로그램에 의해 생성되거나, 사용자에 의해 주문화될 수 있다. 각각의 윈도우(203a-203d)에 대해 동일한 파일명은 대응하는 태스크바 버튼(213a-213d) 내에 도시된다.

<51> 윈도우들(203a-203d)은 애플리케이션 프로그램들에 의해 사용자에게 콘텐츠를 표시하는 데 사용된다. 윈도우(203a)와 연관된 애플리케이션 프로그램은 상이한 애플리케이션 프로그램이거나, 윈도우(203b, 203c 및/또는 203d)에 대응하는 동일 애플리케이션 프로그램일 수 있다. 윈도우들(203a-203d) 각각은 다른 윈도우들(203a-203d)에 독립적이고 외부적이다. 본 명세서에서 사용되는 바와 같이, 제1 윈도우가 제2 윈도우에 포함되지 않고, 제2 윈도우가 제1 윈도우내에 포함되지 않을 때, 제1 윈도우는 제2 윈도우에 외부적이다. 제1 윈도우 및 제2 윈도우가 단순히 중첩되는 경우, 제1 윈도우는 제2 윈도우에 포함되지 않는 것으로 이해해야 한다.

<52> 도 2에 도시된 바와 같이, 사용자는 태스크를 완료하기 위하여 이 애플리케이션 윈도우들(203a-203d)의 세트를 명확히 열었다. 태스크는 열린 애플리케이션 윈도우들(203a-203d) 각각으로부터 정보 또는 콘텐츠를 얻는 것을 포함할 수 있다. 예를 들어, 이 태스크는 인터넷 브라우저 윈도우 및 그래픽 편집 애플리케이션으로부터의 2개의 윈도우를 필요로 할 수 있다. 사용자가 제1 태스크와 관계 없는 추가 애플리케이션 윈도우들을 여는 경우, 이를 기준의 윈도우(203a-203d)는 데스크톱 공간(201)에 변경 없이 유지된다. 기준 윈도우 관리 솔루션을 이용하여, 사용자는 제1 태스크와 연관된 애플리케이션 윈도우들(203a-203d) 각각을 개별적으로 최소화하여 이들을 뷰로부터 제거할 수 있다. 오리지널 태스크로 복귀하기 위하여, 사용자는 각각의 최소화된 애플리케이션 윈도우들(203a-203d)을 개별적으로 검색해야 할 것이다.

<53> 도 3A에 도시된 바와 같이, 본 발명의 양태들에 따르면, 현재 열려 있고 그리고/또는 활성화된 한 세트의 애플리케이션 윈도우들(203a-203d)을 하나의 윈도우 그룹으로 선언하기 위한 사용자 인터페이스 제어(330)가 제공된다. 도 3A는 본 발명의 적어도 일 양태에 따라 그룹화 제어(330)를 포함하는 윈도우 관리 시스템의 일례를 나타낸다. 본 명세서에서 사용되는 용어 "선언"은 2개의 독립적이고 외부적인 애플리케이션 윈도우 사이에 관계를 생성하기 위한 액션을 개시하는 것을 의미하는 것으로 정의된다.

<54> 시나리오(300a)에서, 사용자는 포인터(331)를 애플리케이션 윈도우 그룹화 제어(330)를 향해 이동한다. 태스크바 영역(210)에 도시되어 있지만, 그룹화 제어 요소(330)는 데스크톱 공간(201) 내와 같은 다른 위치에 위치할 수 있다는 것을 이 분야의 전문가들은 이해해야 한다. 또한, 도 3A에는 가동 버튼으로서 도시되어 있지만, 그룹화 제어(330)는 다른 수단에 의해 애플리케이션 윈도우들을 그룹화하도록 개시될 수도 있다. 예를 들어, 일 실시예에서, 사용자는 단일 또는 일련의 키/버튼 스트로크를 수행하여, 애플리케이션 윈도우들(203a-203d)과 같은 둘 이상의 윈도우를 그룹화할 수 있다. 사용자는 키보드 상에서 예정된 일련의 키 스트로크를 누름으로써 애플리케이션 윈도우들을 그룹화할 수 있다. 본 발명의 다른 실시예에서, 사용자는 태스크바 버튼(213a)을 클릭하고 이를 태스크바 버튼(213b)으로 드래그함으로써 윈도우(203a) 및 윈도우(203b)를 그룹화하기 위한 프로세스를 개시할 수 있다. 또 다른 실시예에서, 사용자는 전자 마우스와 같은 입력 디바이스 상의 작동 버튼을 클릭하여 직접적으로, 또는 태스크바 버튼들(213a, 213b)을 작동시킴으로써 간접적으로 애플리케이션 윈도우들(213a, 213b)을 그룹화할 수 있다. 본 발명은 본 명세서에 포함된 예들로 한정되지 않음을 이 분야의 전문가들은 이해해야 한다.

<55> 도 3A를 다시 참조하면, 그룹화 제어(330)를 누를 때, 윈도우들(203a-203d)은 함께 그룹화될 수 있으며, 이후 이 그룹은 도 3B에 도시된 바와 같은 섬네일 이미지(340a)의 크기로 집합적으로 축소될 수 있다. 그룹화 제어(330)의 누름의 일례가 도 3B의 시나리오(300b)에 도시되어 있다. 이 시나리오 예(300b)에서, 윈도우들(203a-203d)의 그룹은 데스크톱 공간(201) 내에 섬네일 이미지(340a)로 축소되어 있다. 또한, 태스크바 버튼들(213a-213d)은 태스크바 영역(210)으로부터 제거되어 있다. 이 예에서는 도시되지 않지만, 대응하는 태스크바 버튼들(213a-213d) 중 하나 이상이 태스크바 영역(210) 내에 남을 수 있다는 것을 이해해야 한다.

<56> 그룹화된 윈도우들은 그룹의 표현으로서 함께, 예를 들어 저장, 캱신 등 관리될 수 있는 한 세트의 윈도우들이다. 윈도우들을 함께 그룹화하는 것은 사용자가 행해야 하는 윈도우 관리의 양을 줄이는 것은 물론, 중첩 윈도우들로 구성되는 작업면의 시각적인 혼란을 줄인다. 또한, 윈도우들의 그룹은 함께 유지될 수 있으며, 따라서 윈도우들의 그룹은 나중에 검색될 수 있다. 임의의 애플리케이션 윈도우가 애플리케이션 윈도우들(203a-203d)의 섬네일 이미지(340a)에 추가되거나 그로부터 나중에 제거될 수 있다.

- <57> 섬네일 이미지(340a)로부터 이전에 최소화된 애플리케이션 윈도우들을 복원하기 위하여, 사용자는 포인터(331)를 이동하여 섬네일 이미지(340a)를 활성화, 예를 들어 클릭할 수 있다. 테스크톱 아이콘(340a) 상의 사용자 클릭에 응답하여, 윈도우들(203a-203d)의 그룹은 그룹화 제어(330)의 실행 이전 상태로 복원된다. 이 예에서, 윈도우들의 외양은 도 3A에 보였던 것처럼 보일 것이다.
- <58> 도 3C는 본 발명의 적어도 일 양태에 따라 그룹화 제어(330)의 실행시 윈도우 관리 시스템의 다른 예를 나타낸다. 도 3C에 도시된 바와 같이, 시나리오(300c)는 윈도우들(203a-203d)의 그룹을 애플리케이션 윈도우들(203a-203d)의 섬네일 이미지(340a) 대신에 소정 타입의 아이콘(340b)으로 도시할 수 있다. 대안으로, 그리고/또는 추가적으로, 태스크바 버튼(340c)이 윈도우들(203a-203d)의 그룹을 표현하기 위해 생성될 수 있다. 섬네일 이미지(340a)에서와 같이, 윈도우들(203a-203d)은 아이콘(340b) 및/또는 태스크바 버튼(340c)의 활성화에 의해 복원될 수 있다. 이 예에서, 윈도우들의 외양은 또한 도 3A에 보여진 바와 같이 보여질 것이다.
- <59> 도 4는 본 발명의 적어도 일 양태에 따라 그룹화 제어(330)의 실행시 윈도우 관리 시스템의 다른 예를 나타낸다. 이 시나리오(400)에서, 4개의 열린 애플리케이션 윈도우들(203a-203d) 중 일부, 예를 들어 3개가 하나의 그룹으로 선언되었고 섬네일 이미지(440)로 축소되었다. 따라서, 본 발명의 일 실시예에서, 사용자는 윈도우(203d)를 테스크톱 공간(201)에 열려 있고 활성 상태로 남겨 두면서 윈도우들(203a-203c)을 그룹화하기로 선택할 수 있다. 테스크톱 공간(201) 내의 모든 열린 윈도우보다 적은 수의 윈도우들의 그룹화를 허가하기 위해 많은 다른 방법이 이용될 수 있으며, 본 발명은 임의의 하나의 방법으로 제한되지 않는다. 태스크바 영역(210)으로 최소화된 열린 윈도우들은 그룹화 제어(330)의 작동에 의해 윈도우 그룹에 포함될 수도 있다는 것을 이 분야의 전문가들은 이해해야 한다. 다른 실시예들에서, 시스템은 다른 열린 윈도우에 의해 중첩되는 경우에도 임의의 최소화된 윈도우를 그룹화하지 않으면서 열린 활성 상태의 윈도우들만을 그룹화하도록 구성될 수 있다.
- <60> 윈도우 그룹이 선언되고 섬네일 이미지(340a)로 최소화된 경우, 사용자는 새로운 애플리케이션 윈도우들의 열림 또는 이전에 최소화된 애플리케이션 윈도우들의 복원을 진행할 수 있다. 윈도우 그룹이 명시적으로 선언된 후에 열린 윈도우들은 제2 윈도우 그룹의 멤버로 간주될 수 있다. 도 5A는 윈도우 그룹(340a)이 선언되고 최소화되었으며 추가 애플리케이션 윈도우들(503a, 503b)이 후속적으로 열린 시나리오(500a)를 나타낸다. 이를 애플리케이션 윈도우는 제2 윈도우 그룹의 멤버이다. 윈도우들(203a-203d)과 마찬가지로, 윈도우들(503a, 503b)은 태스크바 영역(210) 내에 대응하는 태스크바 버튼들(513a, 513b)을 갖는다.
- <61> 윈도우 그룹이 선언되고 최소화된 경우, 사용자는 섬네일 이미지(340a) 상의 더블 클릭, 또는 우측 클릭 및 관련 메뉴로부터의 '그룹 열림'의 선택과 같은 기존의 표준 열림 메카니즘을 통해 그룹을 열어 그룹을 검색할 수 있다. 도 5A에 도시된 바와 같이, 사용자는 포인터(331)를 섬네일 이미지 그룹(340a)으로 이동한다. 그룹의 열림은 섬네일 이미지 그룹(340a)으로부터 애플리케이션 윈도우들(203a-203d)이 풀 스크린으로 애니메이트되는 동시에 애플리케이션 윈도우들(503a, 503b)이 시스템에 의해 애플리케이션 윈도우들의 제2 그룹으로 선언되고 도 5B에 도시된 섬네일 이미지(540)로 애니메이트되는 결과를 가져온다. 제2 윈도우 그룹은 제1 윈도우 그룹의 검색시 자동으로 선언될 수 있다는 것을 이 분야의 전문가들은 이해해야 한다. 대안으로, 시스템은 사용자에게 제2 윈도우 그룹을 능동적으로 선언할 것을 요구하도록 구성될 수 있다.
- <62> 도 5B는 섬네일 이미지(340a)를 통해 제1 윈도우 그룹을 여는 시나리오(500b)를 나타낸다. 제2 그룹은 후속적으로 섬네일 이미지(540)로 최소화되어, 섬네일 이미지(340a)로서의 제1 그룹의 위치를 대체하였다. 윈도우 그룹이 검색된 경우, 정상적인 윈도우 상호작용이 유효하다. 개별 애플리케이션 윈도우들은 이동, 활성화 및 폐쇄될 수 있다. 윈도우들(503a, 503b)은 다른 윈도우 그룹이 이미 존재한다는 사실에 의해 암시적으로 그룹화될 수 있다. 따라서, 사용자는 윈도우들(503a, 503b)을 그룹화하기 위해 그룹화 제어(330)를 다시 작동시킬 필요가 없다.
- <63> 본 발명의 양태들은 도면들에 도시된 윈도우들 및/또는 그룹들의 수로 한정되지 않으며, 임의 수의 상이한 윈도우 그룹들이 생성될 수 있음을 이 분야의 전문가들은 이해해야 한다. 또한, 본 발명의 다양한 양태에 따라 하나 이상의 윈도우가 윈도우 그룹에 추가되거나 제거될 수 있음을 이해해야 한다. 또한, 하나의 윈도우 그룹 내의 애플리케이션 윈도우는 다른 윈도우 그룹 내에 나타날 수도 있다. 더욱이, 본 발명의 양태들은 윈도우 그룹들이 머신 시동 및/또는 로그아웃/로그인 세션들에서 저장 및 복원되는 것을 가능하게 한다.
- <64> 본 발명의 양태들은 윈도우 그룹 상에서의 윈도우인 동작을 지원한다. 예를 들어, 윈도우 그룹 상에서 콘텐츠를 닫거나 저장하거나 갱신하기 위해 동작하는 커맨드는 그 그룹 내의 모든 윈도우가 그의 콘텐츠를 닫거나 저작하거나 갱신하게 할 수 있다. 커맨드는 그룹 전체에 글로벌하거나, 그룹 내의 윈도우들의 서브세트에 고유하

거나, 하나의 애플리케이션 윈도우에 고유할 수 있다. 하나의 애플리케이션 윈도우가 2개의 개별 그룹 내에 있을 때, 하나의 그룹 내의 애플리케이션 윈도우 상에서 수행되는 커맨드는 다른 그룹 내의 동일 애플리케이션 윈도우 상에서 동일한 동작을 수행하거나 수행하지 않도록 구성될 수 있다. 더욱이, 새로운 애플리케이션 윈도우들이 그룹에 추가되고 그리고/또는 제거됨에 따라 커맨드 옵션들이 변경될 수 있다. 예를 들어, 갱신 동작을 각각 수행할 수 있는 2개의 윈도우를 포함하는 그룹이 그 그룹에 추가된 콘텐츠를 갱신할 수 있는 능력 없이 새로운 윈도우를 갖는 경우, 갱신 커맨드는 사용자가 그 윈도우 그룹을 조작할 때 이용할 수 있는 옵션이 더 이상 아닐 수 있다. 하나의 그룹으로서의 윈도우들 상에서 이용하기 위한 커맨드들은 시스템이 어떻게 구성되어 있는가에 따라 동적이거나 동적이 아닐 수 있다.

&lt;65&gt;

도 6은 본 발명의 적어도 일 양태에 따라 복수의 애플리케이션 윈도우를 그룹화하기 위한 방법의 일례의 흐름도이다. 프로세스는 복수의 애플리케이션 윈도우가 열리는 단계 601에서 시작한다. 단계 603에서, 복수의 열린 애플리케이션 윈도우를 하나의 그룹으로 선언하기 위한 입력이 수신된다. 프로세스는 모든 열린 윈도우가 함께 그룹화될 것인지에 대한 결정이 이루어지는 단계 605로 이동한다. 모든 열린 윈도우가 함께 그룹화될 경우, 프로세스는 복수의 윈도우 모두가 하나의 그룹으로서 선언되는 단계 607로 이동하며, 이어서 프로세스는 단계 611로 진행한다. 단계 605의 결정이, 열린 윈도우들 모두가 함께 그룹화되지는 않는 것인 경우, 프로세스는 복수의 애플리케이션 윈도우의 일부가 하나의 그룹으로 선언되는 단계 609로 진행하며, 프로세스는 단계 611로 진행한다. 단계 611에서, 애플리케이션 윈도우들의 그룹의 표현이 표시된다. 단계 611에서 윈도우들의 표현이 표시됨과 함께 프로세스가 종료될 수 있다. 단계 611은 섬네일 이미지(340a)와 같은 섬네일 이미지, 아이콘(340b)과 같은 아이콘, 및/또는 태스크바 버튼(340c)과 같은 태스크바 버튼일 수 있음을 이해해야 한다.

&lt;66&gt;

프로세스는 또한 표현 내의 애플리케이션 윈도우들의 그룹 상에서 동작을 수행하기 위한 입력이 수신되었는지에 대한 결정이 이루어지는 단계 613으로 계속될 수 있다. 한 타입의 동작은 표현 내의 애플리케이션 윈도우들의 콘텐츠를 저장하기 위한 동작을 포함할 수 있다. 입력이 수신되지 않은 경우, 프로세스는 종료된다. 단계 613에서 입력이 수신된 경우, 프로세스는 그룹 내의 애플리케이션 윈도우들 상에서 동작이 수행되는 단계 615로 이동한다. 콘텐츠를 저장하는 경우에, 동작이 글로벌한 경우에는, 애플리케이션 윈도우들 각각의 콘텐츠가 저장된다. 대안으로, 동작이 그룹 내의 애플리케이션 윈도우들의 서브세트 또는 단일 애플리케이션 윈도우에 관한 것인 경우에는, 동작은 관련된 서브세트 또는 하나의 윈도우 상에서만 수행된다.

&lt;67&gt;

단계 615로부터 프로세스는 그룹으로부터 애플리케이션 윈도우들을 검색하기 위한 입력이 수신되었는지에 대한 결정이 이루어지는 단계 617로 이동한다. 예를 들어, 사용자는 애플리케이션 윈도우들을 그들의 오리지널 상태로 복원하기를 원하는 것으로 결정할 수 있다. 입력이 수신되지 않은 경우, 프로세스는 종료된다. 단계 617에서 입력이 수신된 경우, 프로세스는 데스크톱 공간에서 애플리케이션 윈도우들이 그들의 오리지널 상태로 복원되는 단계 619로 이동하고, 프로세스는 종료된다.

&lt;68&gt;

도 7은 본 발명의 적어도 일 양태에 따라 다수의 윈도우 그룹 간의 스위칭을 위한 방법의 일례의 흐름도이다. 프로세스는 복수의 애플리케이션 윈도우가 열리는 단계 701에서 시작한다. 단계 703에서, 복수의 열린 애플리케이션 윈도우를 하나의 그룹으로 선언하기 위한 입력이 수신된다. 단계 705에서, 애플리케이션 윈도우들이 하나의 그룹으로 선언된다. 단계 707로 진행하여, 애플리케이션 윈도우들의 제1 그룹의 표현이 표시된다.

&lt;69&gt;

단계 709에서, 적어도 하나의 새로운 애플리케이션 윈도우를 열기 위한 입력이 수신된다. 단계 711에서, 적어도 하나의 새로운 애플리케이션 윈도우가 열린다. 적어도 하나의 새로운 애플리케이션 윈도우는 애플리케이션 윈도우들의 제1 그룹에서와 동일한 애플리케이션 윈도우일 수 있다는 것을 이해해야 한다. 단계 711로부터, 단계들 713-721에 대한 실선으로 정의된 바와 같이, 그리고 단계들 723-729에 대한 점선으로 정의된 바와 같이, 프로세스의 2개의 진행 중 하나가 이루어질 수 있다. 제1 진행에 도시된 바와 같이, 단계 711로부터, 프로세스는 적어도 하나의 새로운 애플리케이션 윈도우를 하나의 그룹으로 선언하기 위한 입력이 수신되는 단계 713으로 진행한다. 단계 715에서, 적어도 하나의 새로운 애플리케이션 윈도우가 제2 그룹으로 선언된다. 단계 717로 진행하여, 적어도 하나의 애플리케이션 윈도우의 제2 그룹의 표현이 표시된다. 단계 719에서, 선택된 제1 또는 제2 그룹으로부터 애플리케이션 윈도우(들)를 검색하기 위한 입력이 수신되었는지에 대한 결정이 이루어진다. 수신되지 않은 경우, 프로세스는 선택이 이루어지고 수신될 때까지 기다린다. 단계 719에서 입력이 수신된 경우, 프로세스는 선택된 그룹의 애플리케이션 윈도우(들)가 데스크톱 공간에서 그들의 오리지널 상태로 복원되는 단계 721로 진행하고, 프로세스는 종료된다.

&lt;70&gt;

단계들 간의 점선으로 정의된 제2 진행에 도시된 바와 같이, 단계 711로부터, 프로세스는 제1 그룹으로부터 애플리케이션 윈도우들을 검색하기 위한 입력이 수신되었는지에 대한 결정이 이루어지는 단계 723으로 진행한다.

수신되지 않은 경우, 프로세스는 입력이 수신될 때까지 기다린다. 단계 723에서 입력이 수신된 경우, 프로세스는 시스템이 데스크톱 공간 내의 적어도 하나의 새로운 애플리케이션 윈도우를 제2 그룹으로 자동 선언하는 단계 725로 진행한다. 사용자는 애플리케이션 윈도우들의 제1 그룹으로 복귀하기로 결정하였으므로, 시스템은 암시적으로 새로운 애플리케이션 윈도우(들)를 새로운 그룹으로 그룹화한다. 프로세스는 애플리케이션 윈도우들의 제2 그룹의 표현이 표시되는 단계 727로 진행한다. 단계 729에서, 제1 그룹의 애플리케이션 윈도우들은 데스크톱 공간에서 그들의 오리지널 상태로 복원되며, 프로세스는 종료된다. 사용자는 전술한 단계들 중 하나 이상을 따름으로써 원하는 대로 애플리케이션 윈도우들의 그룹들 사이에서 스위칭할 수 있다는 것을 이 분야의 전문가들은 이해해야 한다.

<71> 또 하나의 실시예는 윈도우들의 하나의 그룹으로의 직접 선언 및 그룹화를 프로그램 방식으로 제어하는 것이다. 예를 들어, 마이크로소프트 오피스와 같은 하나 이상의 애플리케이션이 여러 문서로 이루어지는 프로젝트의 개념을 공유할 수 있다. 이들 문서는 동시에 하나의 그룹으로 열리거나, 프로젝트 내의 새로운 문서들이 사용자에 의해 열림에 따라 하나의 그룹에 자동으로 추가될 수 있다.

<72> API와 관련하여, 본 발명의 다양한 양태가 API에 의해 수행될 수 있다. 예를 들어, 공개 API들은 운영 체제와 인터페이스하여 운영 체제가 본 발명의 다양한 특징을 제공하는 것을 가능하게 할 수 있다. 일 실시예에서, 그룹화된 윈도우들을 표현하는 데이터를 처리하기 위한 소프트웨어 아키텍처는 제1의 열린 윈도우 및 제2의 열린 윈도우를 하나의 그룹으로 그룹화하도록 구성된 컴포넌트, 및 이 컴포넌트에 액세스하기 위한 애플리케이션 프로그램 인터페이스를 포함할 수 있다. API는 2개의 윈도우를 그룹화하기 위한 요청을 수신하고, 동작을 수행하는 데 필요한 기능(들)에 액세스한 후, 결과들을 운영 체제에 역 전송할 수 있다. 운영 체제는 API로부터 제공되는 데이터를 이용하여 본 발명의 다양한 기능을 수행할 수 있다.

<73> 애플리케이션 프로그램들이 윈도우 그룹들을 이용하게 하는 API들은 윈도우들을 프로그램 방식으로 그룹화 및 비그룹화하는 것은 물론, 콘텐츠의 저장, 인쇄, 간접, 버전들의 비교, 소스 코드의 컴파일링 등과 같은 커맨드가 그룹 상에서 구현될 때 애플리케이션 프로그램들이 윈도우들의 상태가 어떻게 처리될지를 지정하게 하는 것을 포함한다. 이 상태 정보는 그룹이 검색되고 대응하는 애플리케이션 윈도우들이 복원될 때 그룹 내의 애플리케이션 프로그램의 윈도우가 디폴트 상태 대신에 적절한 상태로 복원되는 것을 가능하게 한다. 예를 들어, 하나의 그룹으로 선언된 최소화된 윈도우는 오리지널 상태로 복원될 수 있다.

<74> 본 발명의 다양한 양태를 구현하는 본 명세서에 설명된 바와 같은 예시적인 시스템 및 방법이 도시되지만, 본 발명은 이들 실시예로 한정되지 않음을 이 분야의 전문가들은 이해할 것이다. 특히 전술한 가르침에 비추어 이 분야의 전문가들에 의해 변경이 이루어질 수 있다. 예를 들어, 전술한 실시예들의 요소들 각각은 단독으로 또는 조합하여 또는 다른 실시예들의 요소들과 조합하여 이용될 수 있다. 또한, 본 발명의 진정한 사상 및 범위를 벗어나지 않고 변경이 이루어질 수 있음을 인식하고 이해할 것이다. 따라서, 본 설명은 본 발명에 대한 제한이 아니라 예시적인 것으로 간주되어야 한다.

## 도면의 간단한 설명

<10> 전술한 본 발명의 요약은 물론, 아래의 실시예들의 상세한 설명은, 청구 발명에 관한 한정이 아니라 예시적으로 포함되는 첨부 도면들과 관련하여 읽을 때 보다 잘 이해된다.

도 1A는 본 발명의 소정 양태들이 구현될 수 있는 범용 디지털 컴퓨팅 환경의 개략도이다.

도 1B-1M은 본 발명의 하나 이상의 양태를 지원하는 범용 컴퓨팅 환경을 나타내는 도면이다.

도 2는 통상의 윈도우 관리 시나리오를 나타내는 도면이다.

도 3A는 본 발명의 적어도 하나의 양태에 따라 그룹화 제어를 포함하는 윈도우 관리 시스템의 일례를 나타내는 도면이다.

도 3B는 본 발명의 적어도 하나의 양태에 따라 그룹화 제어의 실행시 윈도우 관리 시스템의 일례를 나타내는 도면이다.

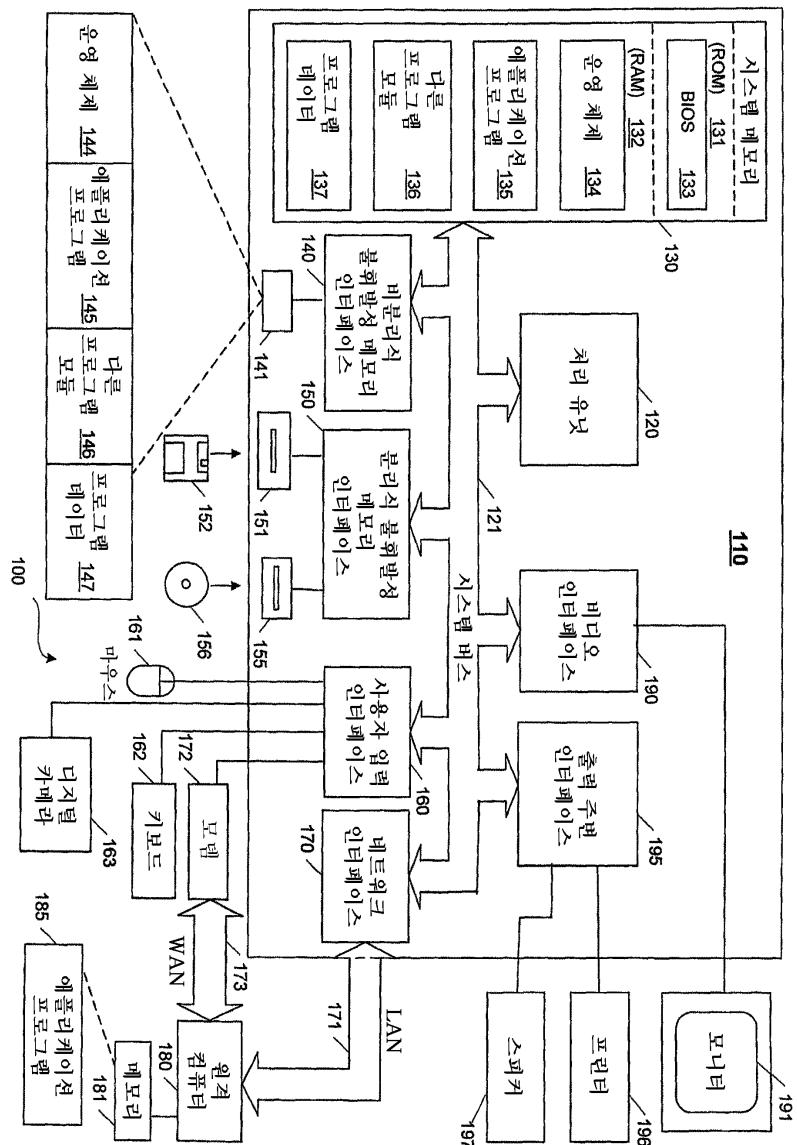
도 3C는 본 발명의 적어도 하나의 양태에 따라 그룹화 제어의 실행시 윈도우 관리 시스템의 다른 예를 나타내는 도면이다.

도 4는 본 발명의 적어도 하나의 양태에 따라 그룹화 제어의 실행시 윈도우 관리 시스템의 또 다른 예를 나타내는 도면이다.

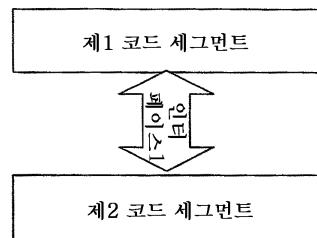
- <18> 도 5A는 본 발명의 적어도 하나의 양태에 따라 다수의 윈도우 그룹이 도시되는 윈도우 관리 시스템의 또 다른 예를 나타내는 도면이다.
- <19> 도 5B는 본 발명의 적어도 하나의 양태에 따라 제1 윈도우 그룹의 재열림시 도 5A의 윈도우 관리 시스템의 일례를 나타내는 도면이다.
- <20> 도 6은 본 발명의 적어도 하나의 양태에 따라 2개의 윈도우를 그룹화하기 위한 방법의 일례의 흐름도이다.
- <21> 도 7은 본 발명의 적어도 하나의 양태에 따라 다수의 윈도우 그룹 간의 스위칭을 위한 방법의 일례의 흐름도이다.

## 도면

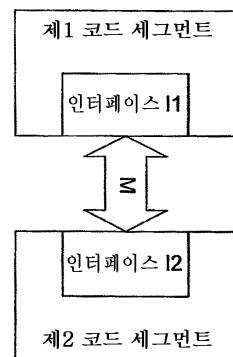
도면1A



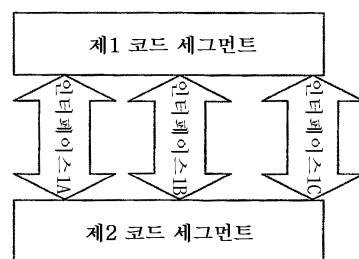
도면1B



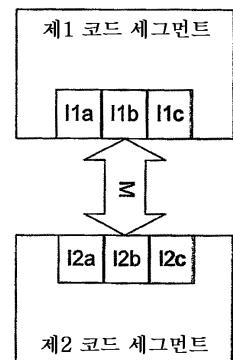
도면1C



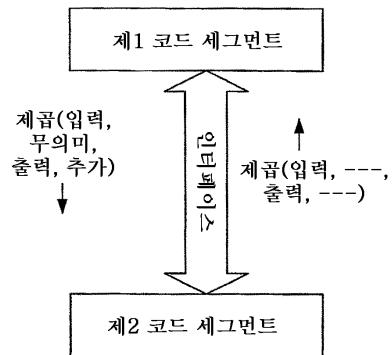
도면1D



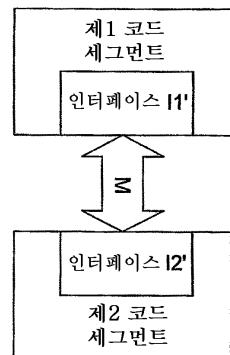
도면1E



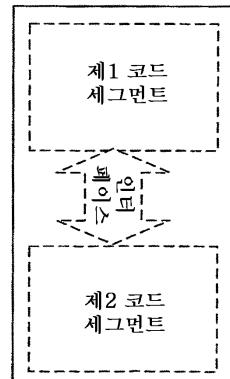
도면1F



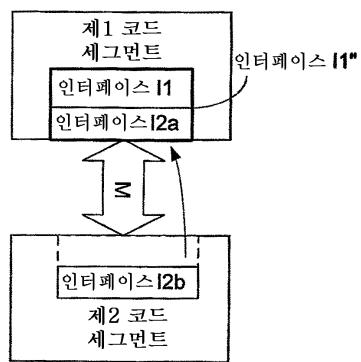
도면1G



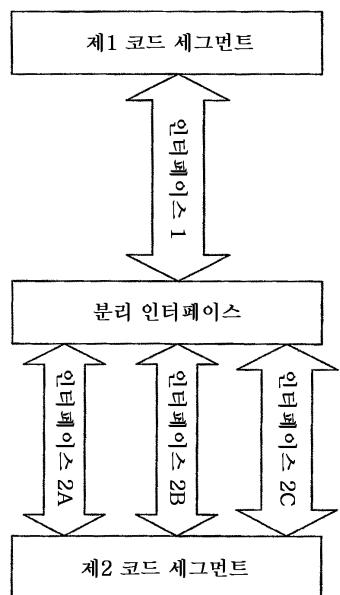
도면1H



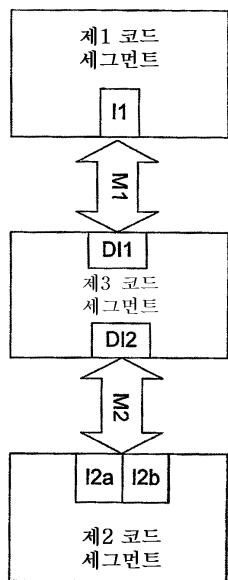
도면1I



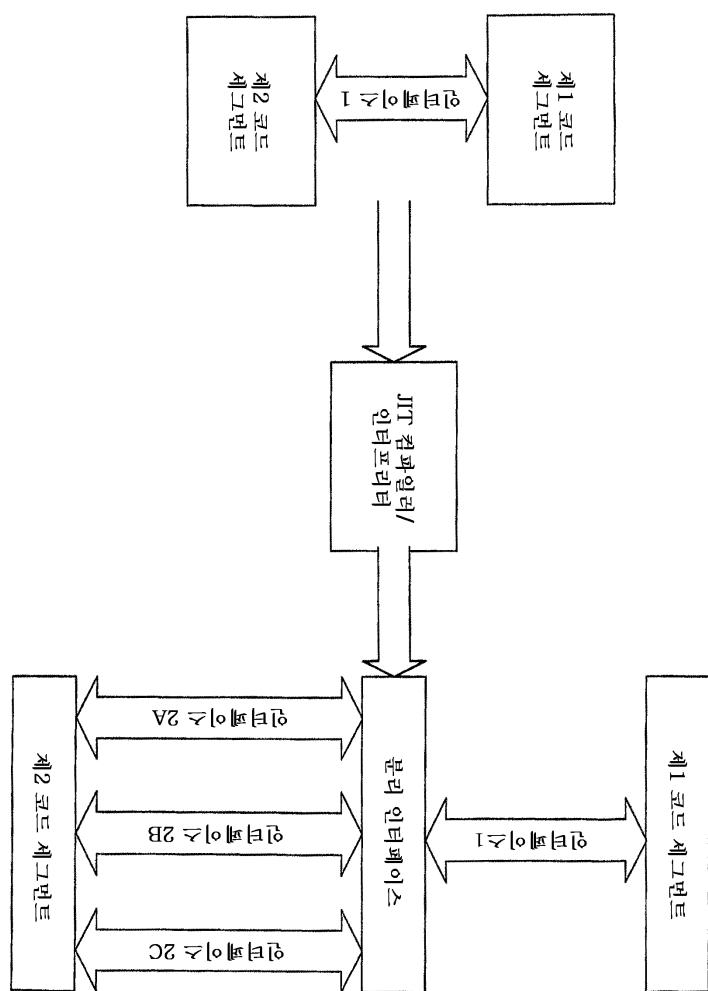
도면1J



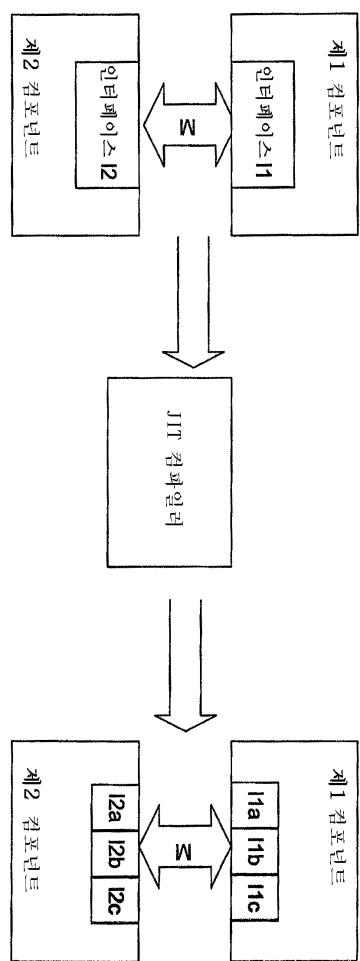
도면1K



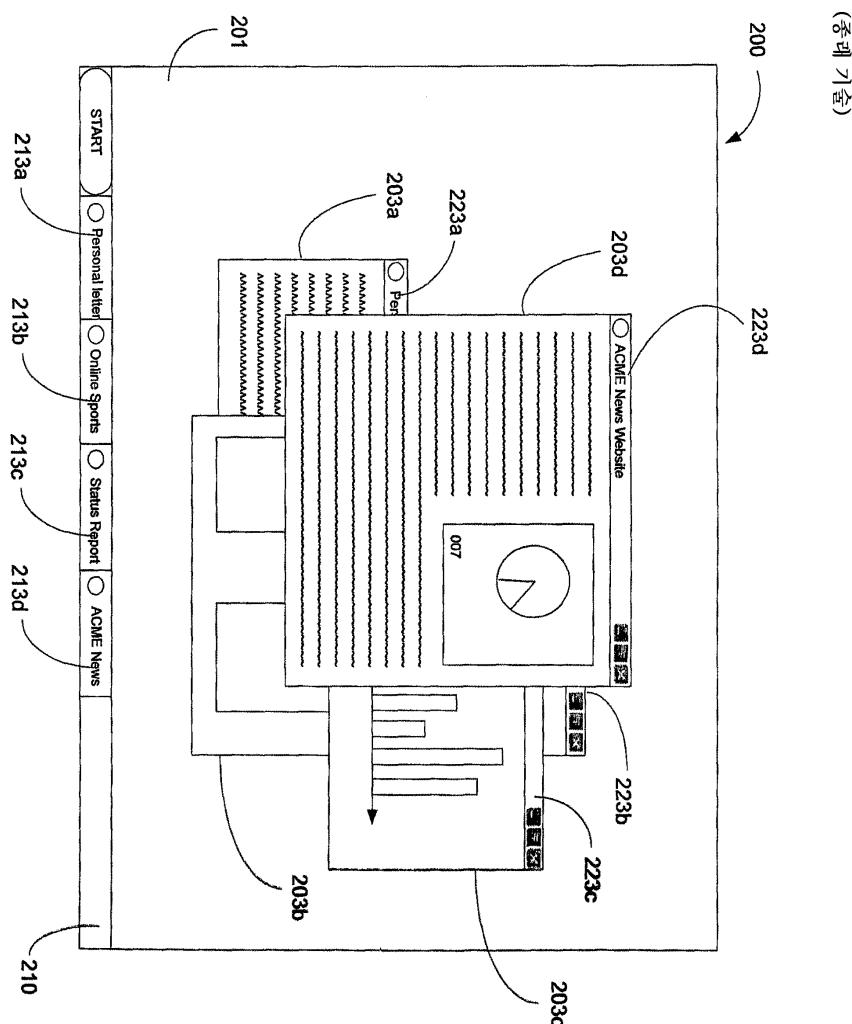
도면1L



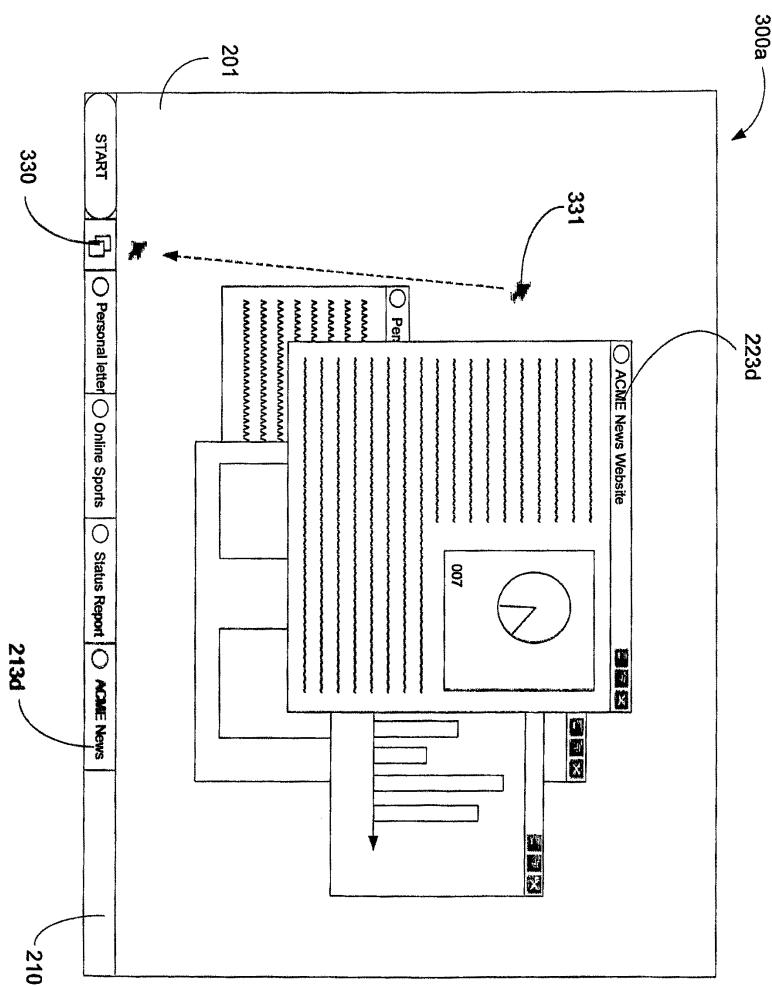
## 도면 1M



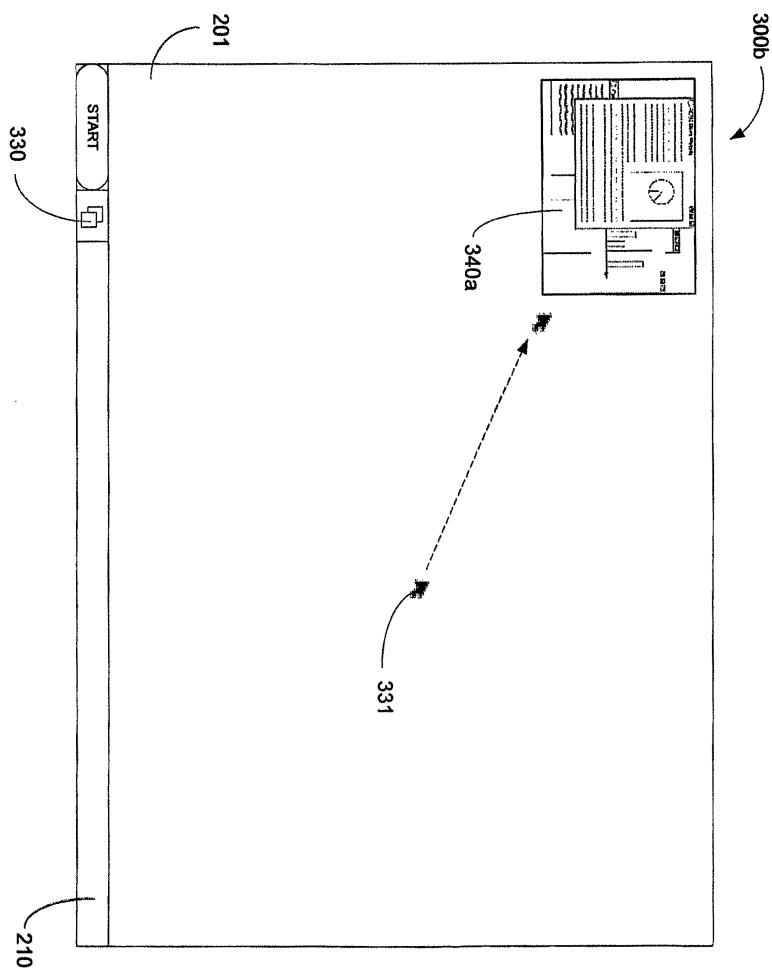
## 도면2



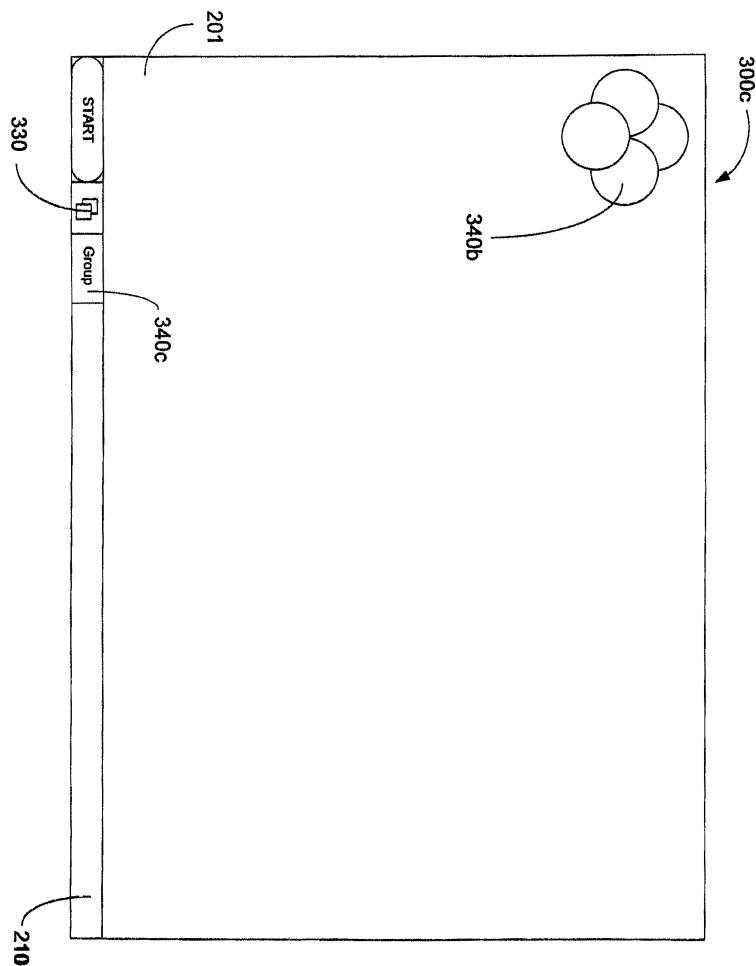
## 도면3A



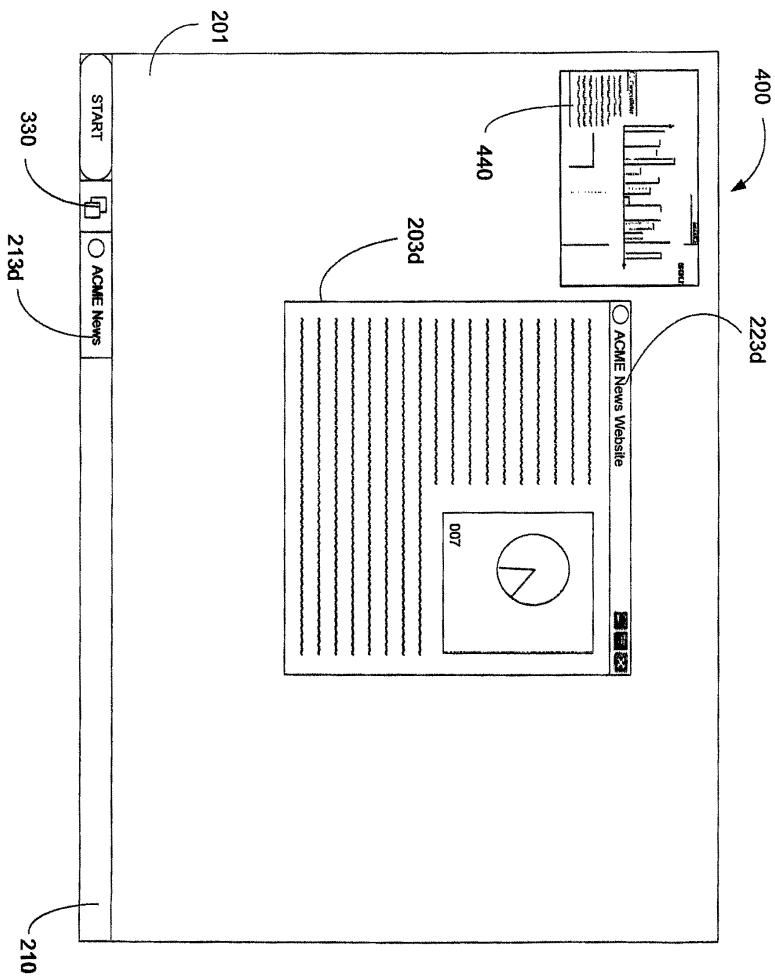
도면3B



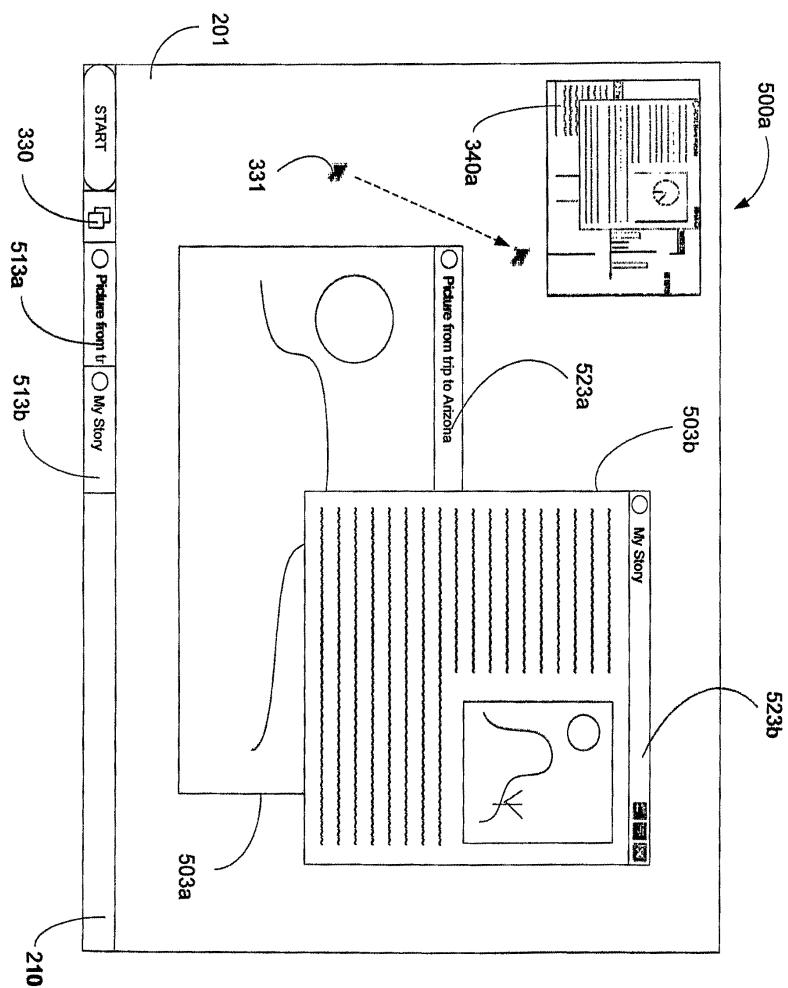
도면3C



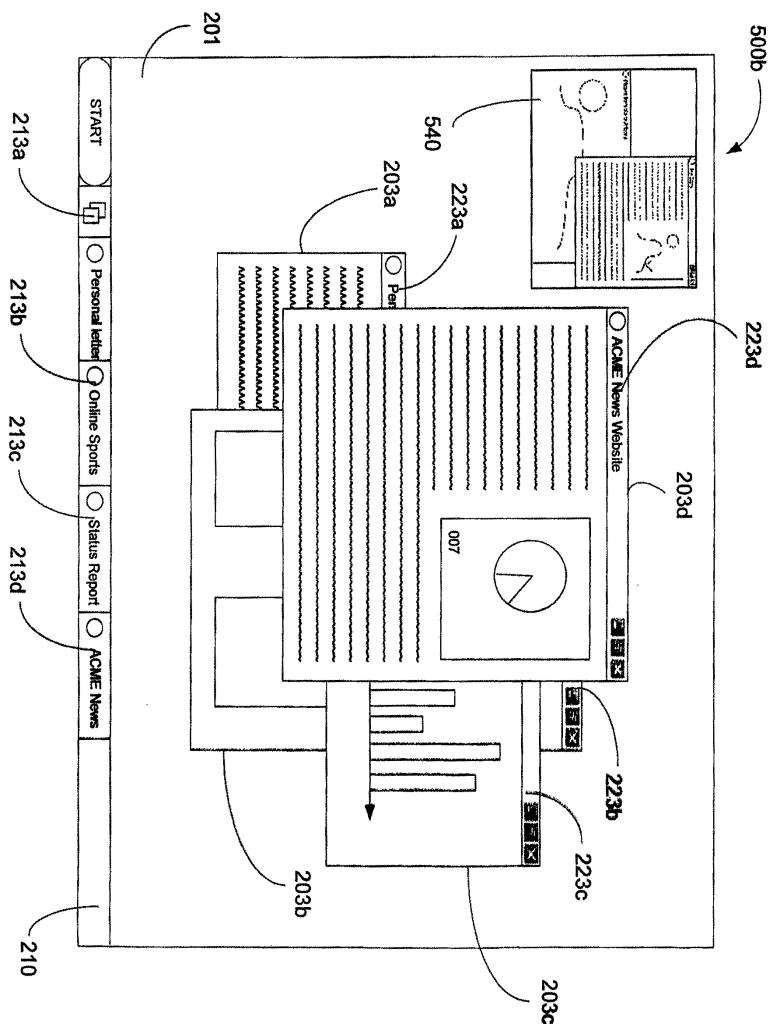
도면4



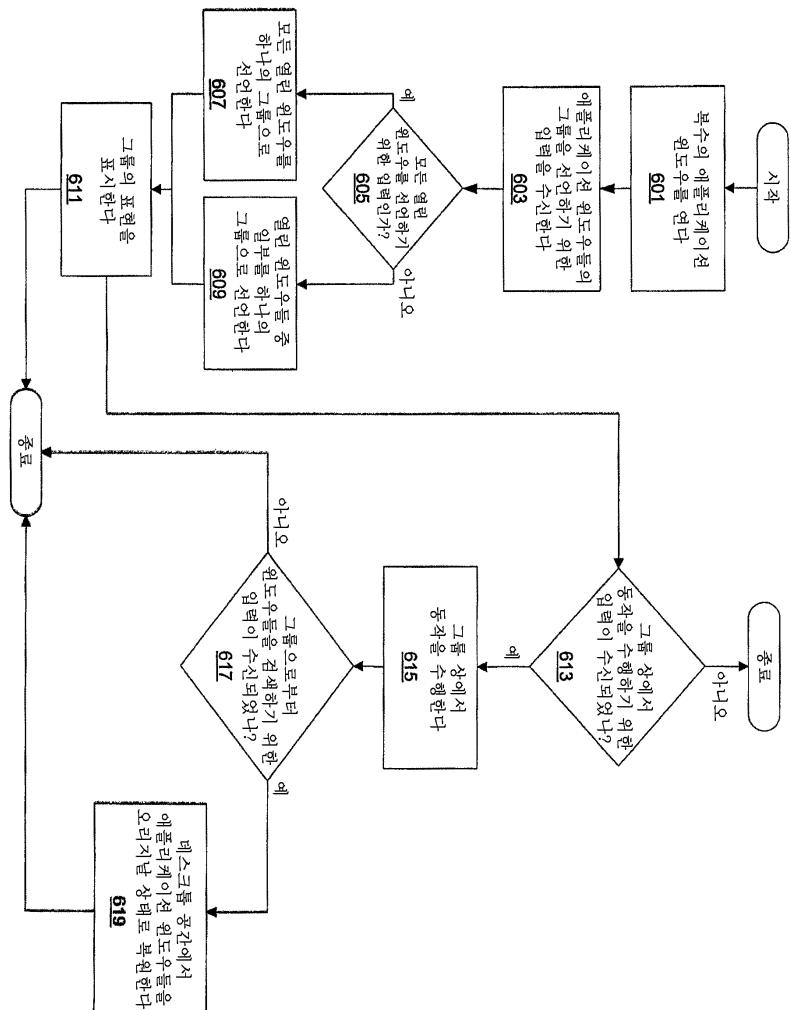
도면5A



## 도면5B



## 도면6



## 도면7

