(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년06월09일
(11) 등록번호 10-2262220
(24) 등록일자 2021년06월02일

(51) 국제특허분류(Int. Cl.)
G06F 3/048 (2021.01) *G06F 3/0484* (2013.01)
G06F 3/14 (2006.01) *G09B 5/02* (2006.01)
G09G 5/14 (2006.01)
(52) CPC특허분류
G06F 3/048 (2021.01)
G06F 3/0484 (2013.01)
(21) 출원번호 10-2020-7011214(분할)
(22) 출원일자(국제) 2013년11월26일
심사청구일자 2020년04월17일
(85) 번역문제출일자 2020년04월17일
(65) 공개번호 10-2020-0044144
(43) 공개일자 2020년04월28일
(62) 원출원 특허 10-2015-7017412
원출원일자(국제) 2013년11월26일
심사청구일자 2018년10월15일
(86) 국제출원번호 PCT/US2013/072089
(87) 국제공개번호 WO 2014/085502
국제공개일자 2014년06월05일
(30) 우선권주장
61/731,399 2012년11월29일 미국(US)
(56) 선행기술조사문헌
국제공개특허 제2012-087847호(2012.06.28.) 1부.*
한국공개특허 제10-2012-0092598호(2012.08.21.) 1부.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
파세토, 인크.
미국 85251 애리조나주 스코츠데일 노스 스코츠데일 로드 4110 스위트 315
(72) 발명자
크리스마스, 코이
미국 54880 위스콘신주 수피리어 커밍 애비뉴1020
말패스, 루크
영국 에스티6 4이엔 스태퍼드셔 스톡-온-트렌트 서턴 애비뉴 26
루츠, 파넬
미국 55803 미네소타주 덜루스 클레이모어 스트리트 824
(74) 대리인
양영준, 장덕순

전체 청구항 수 : 총 13 항 심사관 : 김종기

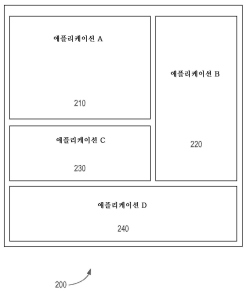
(54) 발명의 명칭 **여러 애플리케이션들을 디스플레이하기 위한 시스템 및 방법**

(57) 요 약

복수의 애플리케이션을 그래픽 사용자 인터페이스("GUI")상에 디스플레이하는 시스템들, 방법들, 및 컴퓨터 프로그램 제품들이 제공된다. 애플리케이션 제어 시스템("ACS")은 어느 애플리케이션들이 디스플레이되어야 하는지 결정할 수 있고, ACS는 GUI상의 애플리케이션에 대한 최적의 레이아웃을 결정할 수 있다. ACS는 애플리케이션들

(뒷면에 계속)

대 표 도 - 도2



의 이동 및/또는 크기 조정을 행할 수 있고 애플리케이션들이 GUI의 일부를 실질적으로 채우도록 최적의 레이아웃을 자동으로 결정할 수 있다. 부가적으로, ACS는 애플리케이션 프로그래밍 인터페이스("API")를 이용하여 애플리케이션들 간의 통신을 용이하게 할 수 있다. 애플리케이션은 API를 통해 요청을 전송함으로써 GUI상에서 임의의 다른 애플리케이션으로부터 정보를 얻을 수 있다.

(52) CPC특허분류
G06F 3/14 (2020.08)
G09B 5/02 (2013.01)
G09G 5/14 (2013.01)
G09G 2354/00 (2013.01)

명 세 서

청구범위

청구항 1

컴퓨터 구현 방법으로서,

컴퓨터 기반 시스템에 의해, 그래픽 사용자 인터페이스("GUI")에 디스플레이될 애플리케이션의 디폴트 개수를 결정하는 단계;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 애플리케이션의 디폴트 개수에 기초하여, 상기 GUI에 디스플레이될 복수의 애플리케이션을 선택하는 단계;

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션 중 제1 애플리케이션을 활성화하는 단계;

상기 컴퓨터 기반 시스템에 의해, 상기 제1 애플리케이션의 크기를 조정하는 단계;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 제1 애플리케이션을 활성화하는 것에 응답하여, 상기 복수의 애플리케이션 중 제2 애플리케이션을 활성화하는 단계;

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션이 상기 GUI를 채우도록 상기 제2 애플리케이션의 크기를 조정하는 단계;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 제1 애플리케이션을 통해, 제1 태스크를 포함하는 입력을 수신하는 단계;

상기 컴퓨터 기반 시스템에 의해, 상기 입력을 키워드들로 파싱하는 단계;

상기 컴퓨터 기반 시스템에 의해, 상기 제2 애플리케이션이 상기 제1 태스크를 완수하기 위해 획득될 수 있는 데이터를 포함한다고 결정하는 단계;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 키워드들에 기초하여, 상기 제2 애플리케이션에 요청을 전송하는 단계; 및

상기 컴퓨터 기반 시스템에 의해, 상기 데이터를 상기 제2 애플리케이션으로부터 애플리케이션 프로그래밍 인터페이스를 통해 상기 제1 애플리케이션에 전송하는 단계

를 포함하는, 컴퓨터 구현 방법.

청구항 2

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션에 후속 애플리케이션을 부가하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 3

제2항에 있어서,

상기 컴퓨터 기반 시스템에 의해, 상기 후속 애플리케이션을 상기 GUI에 통합하는 단계를 더 포함하고,

상기 후속 애플리케이션을 상기 GUI에 통합하는 단계는 상기 복수의 애플리케이션과 상기 후속 애플리케이션이 상기 GUI를 채우도록 상기 복수의 애플리케이션의 크기를 조정하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 4

제1항에 있어서,

상기 복수의 애플리케이션 각각에 대한 상대적 크기는 상기 복수의 애플리케이션의 사용의 빈도수, 및 상기 복수의 애플리케이션의 사용자 순위 중 적어도 하나에 기초하는, 컴퓨터 구현 방법.

청구항 5

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해, 사용자 입력에 기초하여 상기 복수의 애플리케이션 중 상기 제1 애플리케이션의 크기를 조정하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 6

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해 그리고 제1 사용자 입력에 기초하여, 상기 복수의 애플리케이션 중 상기 제1 애플리케이션의 크기를 로크(locking)하는 단계;

상기 컴퓨터 기반 시스템에 의해 그리고 제2 사용자 입력에 기초하여, 상기 복수의 애플리케이션 중 상기 제2 애플리케이션의 크기를 조정하는 단계; 및

상기 컴퓨터 기반 시스템에 의해 그리고 상기 제2 사용자 입력에 응답하여, 상기 복수의 애플리케이션이 상기 GUI를 채우도록 상기 복수의 애플리케이션 중 나머지 애플리케이션들의 크기를 조정하는 단계

를 더 포함하고,

상기 제1 애플리케이션의 크기는 일정하게 남아 있는, 컴퓨터 구현 방법.

청구항 7

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해 그리고 상기 제1 애플리케이션을 활성화하는 것에 응답하여, 상기 제1 애플리케이션의 크기를 증가시키는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 8

제1항에 있어서,

상기 복수의 애플리케이션은 활성 애플리케이션에 기초하여 선택되는, 컴퓨터 구현 방법.

청구항 9

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해 그리고 상기 복수의 애플리케이션 중 상기 제1 애플리케이션을 통해 요청을 수신하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 10

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션 중 상기 제1 애플리케이션에서 제1 데이터를 수신하는 단계, 및 상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션 중 상기 제2 애플리케이션에서 제2 데이터를 로딩하는 단계

를 더 포함하고,

상기 제2 데이터는 상기 제1 데이터와 관련된, 컴퓨터 구현 방법.

청구항 11

제1항에 있어서,

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션 중 학생 리스트 애플리케이션에서 학생의 선택을 수신하는 단계;

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션 중 등급(grades) 애플리케이션과 상기 복수의 애플

리케이션 중 통신 애플리케이션에서 상기 학생과 관련된 콘텐츠를 로딩하는 단계; 및

상기 컴퓨터 기반 시스템에 의해 그리고 상기 등급 애플리케이션에서의 정보에 기초하여, 상기 통신 애플리케이션을 통해 상기 학생에게 메시지를 전송하는 단계

를 더 포함하는, 컴퓨터 구현 방법.

청구항 12

시스템으로서,

애플리케이션들을 그래픽 사용자 인터페이스("GUI")에 디스플레이하는 프로세서; 및

상기 프로세서와 통신하도록 구성된 실체의(tangible), 비일시적 메모리를 포함하고,

상기 실체의, 비일시적 메모리에는 명령어들이 저장되어 있고, 상기 명령어들은 상기 프로세서에 의한 실행에 응답하여, 상기 프로세서로 하여금 동작들을 수행하도록 하고, 상기 동작들은:

상기 프로세서에 의해, 상기 GUI에 디스플레이될 애플리케이션의 디폴트 개수를 결정하는 동작;

상기 프로세서에 의해 그리고 상기 애플리케이션의 디폴트 개수에 기초하여, 상기 GUI에 디스플레이될 복수의 애플리케이션을 선택하는 동작;

상기 프로세서에 의해, 상기 복수의 애플리케이션 중 제1 애플리케이션을 활성화하는 동작;

상기 프로세서에 의해, 상기 제1 애플리케이션의 크기를 조정하는 동작;

상기 프로세서에 의해 그리고 상기 제1 애플리케이션을 활성화하는 것에 응답하여, 상기 복수의 애플리케이션 중 제2 애플리케이션을 활성화하는 동작;

상기 프로세서에 의해, 상기 복수의 애플리케이션이 상기 GUI를 채우도록 상기 제2 애플리케이션의 크기를 조정하는 동작;

상기 프로세서에 의해 그리고 상기 제1 애플리케이션을 통해, 제1 태스크를 포함하는 입력을 수신하는 동작;

상기 프로세서에 의해, 상기 입력을 키워드들로 파싱하는 동작;

상기 프로세서에 의해, 상기 제2 애플리케이션이 상기 제1 태스크를 완수하기 위해 획득될 수 있는 데이터를 포함한다고 결정하는 동작;

상기 프로세서에 의해 그리고 상기 키워드들에 기초하여, 상기 제2 애플리케이션에 요청을 전송하는 동작; 및

상기 프로세서에 의해, 상기 데이터를 상기 제2 애플리케이션으로부터 애플리케이션 프로그래밍 인터페이스를 통해 상기 제1 애플리케이션에 전송하는 동작

을 포함하는, 시스템.

청구항 13

컴퓨터 실행 가능 명령어들이 저장되어 있는 컴퓨터 판독 가능 저장 매체로서, 상기 컴퓨터 실행 가능 명령어들은 애플리케이션들을 그래픽 사용자 인터페이스("GUI")에 디스플레이하는 컴퓨터 기반 시스템에 의한 실행에 응답하여, 상기 컴퓨터 기반 시스템으로 하여금 동작들을 수행하도록 하고, 상기 동작들은:

상기 컴퓨터 기반 시스템에 의해, 상기 GUI에 디스플레이될 애플리케이션의 디폴트 개수를 결정하는 동작;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 애플리케이션의 디폴트 개수에 기초하여, 상기 GUI에 디스플레이될 복수의 애플리케이션을 선택하는 동작;

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션 중 제1 애플리케이션을 활성화하는 동작;

상기 컴퓨터 기반 시스템에 의해, 상기 제1 애플리케이션의 크기를 조정하는 동작;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 제1 애플리케이션을 활성화하는 것에 응답하여, 상기 복수의 애플리케이션 중 제2 애플리케이션을 활성화하는 동작;

상기 컴퓨터 기반 시스템에 의해, 상기 복수의 애플리케이션이 상기 GUI를 채우도록 상기 제2 애플리케이션의

크기를 조정하는 동작;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 제1 애플리케이션을 통해, 제1 태스크를 포함하는 입력을 수신하는 동작;

상기 컴퓨터 기반 시스템에 의해, 상기 입력을 키워드들로 파싱하는 동작;

상기 컴퓨터 기반 시스템에 의해, 상기 제2 애플리케이션이 상기 제1 태스크를 완수하기 위해 획득될 수 있는 데이터를 포함한다고 결정하는 동작;

상기 컴퓨터 기반 시스템에 의해 그리고 상기 키워드들에 기초하여, 상기 제2 애플리케이션에 요청을 전송하는 동작; 및

상기 컴퓨터 기반 시스템에 의해, 상기 데이터를 상기 제2 애플리케이션으로부터 애플리케이션 프로그래밍 인터페이스를 통해 상기 제1 애플리케이션에 전송하는 동작

을 포함하는, 컴퓨터 판독 가능 저장 매체.

발명의 설명

기 술 분 야

[0001] 본 개시는 일반적으로 그래픽 사용자 인터페이스들에 관련되고, 더 구체적으로는 대화형 애플리케이션들을 그래픽 사용자 인터페이스상에 디스플레이하는 시스템들 및 방법들에 관련된다.

배 경 기 술

[0002] 그래픽 사용자 인터페이스들("GUI")은 컴퓨팅 리소스들과 상호작용하기 위한 비주얼 디스플레이를 사용자에게 제공한다. 소프트웨어 애플리케이션들의 복잡성과 속력이 계속 증가함에 따라, 사용자들이 여러 프로그램들과 상호작용하는 것을 허용하는 사용자 친화적 디스플레이들을 GUI들이 제공하는 것이 더 중요하게 된다. 기존 GUI들은 사용자들이 여러 애플리케이션들을 동시에 열게 허용한다. 그러나, 사용자는 원하는 레이아웃을 만들기 위해 GUI상의 각각의 애플리케이션을 위치시켜야 한다. 사용자는 또한 각각의 애플리케이션의 크기를 개별적으로 조정해야 한다. 사용자가 디스플레이에 추가의 애플리케이션을 부가하고 싶으면, 사용자는 모든 애플리케이션들이 동시에 보이도록 하기 위해 애플리케이션들의 크기를 다시 조정해야 한다.

[0003] 또한, 기존 시스템들에서, 컴퓨터 시스템상에서 실행되는 애플리케이션들은 서로 통신하기 위한 제한된 수단을 가질 수 있다. 각각의 애플리케이션은 특정 목적을 가질 수 있고, 하나의 애플리케이션이 다른 애플리케이션에 유용할 데이터를 포함할 수 있더라도, 데이터를 공유하기 위해 애플리케이션들에 대해 추가의 소프트웨어 또는 애플리케이션들의 프로그래밍의 변경들을 필요로 할 수 있거나, 또는 사용자가 수동으로 하나의 애플리케이션으로부터 데이터를 얻고 그 데이터를 다른 애플리케이션에 입력하도록 요구될 수 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0004] 본 개시는 여러 애플리케이션들을 그래픽 사용자 인터페이스("GUI")상에 디스플레이하는 시스템, 방법, 및 컴퓨터 프로그램 제품을 포함한다. 다양한 실시예들에서, 애플리케이션 제어 시스템("ACS")은 GUI상에 디스플레이할 애플리케이션의 개수를 결정할 수 있다. ACS는 각각의 애플리케이션에 대한 상대적 크기를 결정할 수 있다. ACS는 애플리케이션들이 실질적으로 GUI의 일부를 채우도록, 각각의 애플리케이션에 대한 최적의 로케이션을 결정할 수 있다. GUI는 최적의 레이아웃에서 애플리케이션들을 디스플레이하기 위해 웹 클라이언트에 데이터를 전송할 수 있다.

[0005] 다양한 실시예들에서, ACS는 제1 애플리케이션으로부터 데이터를 수신할 수 있다. ACS는 제2 애플리케이션이, 제1 애플리케이션으로부터 수신된 데이터와 관련된 부가 데이터를 가질 가능성이 높다고 결정할 수 있다. ACS는 제2 애플리케이션에 그 부가 데이터를 요청할 수 있다. ACS는 부가 데이터를 수신할 수 있고 부가 데이터를

제1 애플리케이션에 전송할 수 있다. ACS는 부가 데이터를 이용하여 제1 애플리케이션으로부터의 요청을 완수할 수 있다.

도면의 간단한 설명

[0006] 도면들과 관련하여 고려될 때, 상세한 설명과 청구항들을 참조함으로써 더 충분한 이해가 얻어질 수 있으며, 여기서, 도면들에 걸쳐 동일 참조 번호들은 유사한 구성요소들을 지칭한다.

도 1은 본 개시의 다양한 실시예들에 따라 GUI상에 여러 애플리케이션들을 디스플레이하는 시스템의 블록도를 도시한다.

도 2는 다양한 실시예들에 따라 4개의 애플리케이션을 디스플레이하는 GUI의 예시적 스크린을 도시한다.

도 3은 다양한 실시예들에 따라 5개의 애플리케이션을 디스플레이하는 GUI의 예시적 스크린을 도시한다.

도 4는 다양한 실시예들에 따라 크기가 조정된 애플리케이션을 갖는 GUI의 예시적 스크린을 도시한다.

도 5는 다양한 실시예들에 따라 최적의 레이아웃을 결정하기 위한 프로세스의 흐름도를 도시한다.

도 6은 다양한 실시예들에 따라 애플리케이션들 간에 데이터를 전송하기 위한 프로세스의 흐름도를 도시한다.

도 7은 다양한 실시예들에 따라 교육 애플리케이션들을 갖는 GUI의 예시적 스크린을 도시한다.

도 8은 다양한 실시예들에 따라 복수의 애플리케이션을 디스플레이하기 위한 프로세스의 흐름도를 도시한다.

도 9는 다양한 실시예들에 따라 디스플레이로부터 애플리케이션을 제거하기 위한 프로세스의 흐름도를 도시한다.

도 10은 다양한 실시예들에 따라 디스플레이의 크기를 조정하기 위한 프로세스의 흐름도를 도시한다.

도 11은 다양한 실시예들에 따라 애플리케이션을 불능화하기 위한 프로세스의 흐름도를 도시한다.

도 12는 다양한 실시예들에 따라 애플리케이션의 크기를 증가시키기 위한 프로세스의 흐름도를 도시한다.

도 13은 다양한 실시예들에 따라 애플리케이션을 이동시키기 위한 프로세스의 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0007] 본 명세서의 예시적 실시예들의 상세한 설명은 다양한 실시예들을 예시적으로 도시하는 첨부 도면 및 도해를 참조한다. 이러한 다양한 실시예들이 통상의 기술자가 본 발명을 실시하는 것을 가능하게 하기에 충분할 정도로 상세히 기술되지만, 다른 실시예들이 실현될 수 있다는 점, 및 본 개시의 사상 및 범위로부터 벗어나지 않고 논리적 및 기계적 변형들이 이루어질 수 있다는 점을 이해해야 한다. 따라서, 본 개시의 상세한 설명은 한정이 아닌 예시를 위해서만 제시된다. 예를 들어, 방법 또는 프로세스에 관한 설명들 중 임의의 것에서 언급된 단계들은 임의의 순서로 실행될 수 있고, 제시된 순서로 한정되지 않는다. 더욱이, 어떤 기능들 또는 단계들은 아웃소싱될 수 있거나 또는 하나 이상의 제3자에 의해 수행될 수 있다. 또한, 어느 단수의 언급은 복수의 실시예를 포함하고, 어느 하나 초과의 컴포넌트의 언급은 단수의 실시예를 포함할 수 있다.

[0008] 시스템들, 방법들, 및 컴퓨터 프로그램 제품들이 제공된다. 본 명세서의 상세한 설명에서, "다양한 실시예들", "일 실시예", "실시예", "예시적 실시예" 등의 언급은 설명된 실시예가 특별한 특징, 구조, 또는 특성을 포함할 수 있지만, 모든 실시예가 반드시 그러한 특별한 특징, 구조, 또는 특성을 포함하는 것은 아닐 수 있음을 지시한다. 더욱이, 이러한 문구들이 반드시 동일 실시예를 언급하는 것은 아니다. 또한, 특별한 특징, 구조, 또는 특성이 실시예와 관련하여 기술될 때, 이러한 특징, 구조 또는 특성을, 명시적으로 기술되어 있든 그렇지 않든 간에, 다른 실시예들과 관련하여 실시하는 것은 통상의 기술자의 지식의 범위 내에 있다고 하기로 한다. 전술한 설명을 읽은 후에, 본 개시를 대안의 실시예들에서 구현하는 방법이 관련 기술(들)의 통상의 기술자에게 자명할 것이다.

[0009] 그래픽 사용자 인터페이스("GUI")상에 복수의 애플리케이션을 디스플레이하는 시스템들, 방법들, 및 제품들이 개시된다. 다양한 실시예들에서, 복수의 애플리케이션이 GUI상에 디스플레이될 수 있다. 다양한 실시예들에서, 애플리케이션들은 또한 타일들로서 지칭될 수 있다. 복수의 애플리케이션 각각이 GUI의 일부에 위치될 수 있고, 복수의 애플리케이션이 실질적으로 GUI를 커버할 수 있도록, 복수의 애플리케이션이 GUI상에 배치될 수 있다. 애플리케이션 제어 시스템("ACS")은 각각의 애플리케이션의 상대적 크기와 위치를 제어할 수

있다. ACS는, 상대적으로 높은 레벨의 중요성을 갖는 것으로 고려되는 애플리케이션들이 GUI 상에 더 크게 나타나고, 상대적으로 낮은 레벨의 중요성을 갖는 것으로 고려되는 애플리케이션들이 GUI 상에 더 작게 나타나도록, 각각의 애플리케이션에 대한 디폴트 설정들을 포함할 수 있다. ACS는 다양한 애플리케이션들의 사용자의 사용을 추적하는 것에 응답하여 디폴트 설정들을 조정할 수 있다. ACS는 추가의 애플리케이션들을 GUI에 통합하거나 또는 GUI로부터 애플리케이션들을 제거하는 것에 응답하여 애플리케이션들의 크기 및 위치를 조정할 수 있다. 또한, 사용자는 설정들을 사용자 선호들로 조정할 수 있다.

[0010] 다양한 실시예들에서, 다양한 애플리케이션들은 서로 통신할 수 있다. 다양한 애플리케이션들은 일련의 호출들을 통해 서로 통신할 수 있다. 다양한 실시예들에서, 제1 애플리케이션은 제2 애플리케이션으로부터의 정보가 태스크를 완수하기 위해 필요할 수 있다고 결정할 수 있다. 제1 애플리케이션은 애플리케이션 프로그래밍 인터페이스("API")를 통해 요망되는 정보에 대한 요청을 제2 애플리케이션에 전송할 수 있다. 제2 애플리케이션은 요망되는 정보를 제1 애플리케이션에 전송할 수 있고, 제1 애플리케이션은 태스크를 완수하기 위해 정보를 이용할 수 있다.

[0011] 다양한 실시예들에서, API는 클라우드 기반의 시스템의 일부일 수 있다. 클라우드 컴퓨팅은 본 명세서에서 더 상세히 논의된다. 각각의 애플리케이션은, 애플리케이션이 특정 기능들을 수행하고 다양한 제3자 서버들 및 데이터베이스들로부터 데이터를 얻는 것을 허가하는 다양한 허가들을 가질 수 있다.

[0012] 도 1을 참조하면, 복수의 애플리케이션을 디스플레이하기 위한 시스템(100)이 다양한 실시예들에 따라 도시된다. 시스템(100)은 애플리케이션 제어 시스템("ACS")(110), 웹 클라이언트(130), 하나 이상의 서버(140), 및 하나 이상의 데이터베이스(150)를 포함할 수 있다. ACS(110)는 API(160)를 포함할 수 있다. ACS(110)는 서버들과 데이터베이스들의 임의의 조합을 더 포함할 수 있다. 다양한 시스템 컴포넌트들은 네트워크(120)를 통해 통신할 수 있다. 웹 클라이언트(130)는 그래픽 사용자 인터페이스("GUI")(170)를 포함할 수 있다. 다양한 실시예들에서, ACS(110)는 GUI(170)상에 애플리케이션들을 디스플레이하는 것과 관련하여 프로세스의 전부 또는 일부를 수행할 수 있거나 또는 수행하도록 구성될 수 있다. ACS(110)는 서버들, 데이터베이스들, 방화벽들, 컴퓨터들 등과 같은, 하드웨어와 소프트웨어의 임의의 조합을 포함할 수 있다. 네트워크(120)는 하드웨어 및/또는 소프트웨어 컴포넌트들을 통합한 임의의 클라우드, 클라우드 컴퓨팅 시스템, 또는 전자 통신 시스템 또는 방법을 포함할 수 있다. 개체들 간의 통신은 임의의 적절한 통신 채널들, 예를 들어, 전화 네트워크, 엑스트라넷, 인트라넷, 인터넷, 상호작용 디바이스의 포인트(판매점 디바이스, 개인용 정보 단말기(예를 들어, iPhone®, Palm Pilot®, Blackberry®, 셀룰러 폰, 키오스크, 기타 등등), 온라인 통신, 위성 통신, 오프라인 통신, 무선 통신, 트랜스폰더 통신, 근거리 네트워크(LAN), 광역 네트워크(WAN), 가상 사설 네트워크(VPN), 네트워크된 또는 링크된 디바이스들, 키보드, 마우스, 및/또는 임의의 적절한 통신 또는 데이터 입력 모달리티 등을 통해 달성될 수 있다. 또한, 시스템은 본 명세서에서 TCP/IP 통신 프로토콜들로 구현되는 것으로 종종 기술되지만, 시스템은 또한 IPX, Appletalk, IP-6, NetBIOS, OSI, 임의의 터널링 프로토콜(예를 들어, IPsec, SSH), 또는 임의의 개수의 기존의 또는 미래의 프로토콜들을 사용하여 구현될 수 있다. 네트워크가 인터넷과 같은 공중 네트워크의 성질을 갖는 경우에, 네트워크가 보안되지 않고 도청자들에게 노출된 것으로 추정하는 것이 유리할 수 있다. 일반적으로 인터넷과 관련하여 활용되는 프로토콜들, 표준들, 및 애플리케이션 소프트웨어와 관련된 특정 정보는 일반적으로 통상의 기술자에게 알려져 있고, 따라서 본 명세서에서 상세히 기술할 필요가 없다. 예를 들어, DILIP NAIK, INTERNET STANDARDS AND PROTOCOLS (1998); JAVA 2 COMPLETE, various authors, (Sybex 1999); DEBORAH RAY AND ERIC RAY, MASTERING HTML 4.0 (1997); and LOSHIN, TCP/IP CLEARLY EXPLAINED (1997) and DAVID GOURLEY AND BRIAN TOTTY, HTTP, THE DEFINITIVE GUIDE (2002)를 참고하고, 그 내용들은 본 명세서에 참조되어 포함된다. 다양한 시스템 컴포넌트들은 독립적으로, 개별적으로, 또는 데이터 링크들을 통해 집합적으로 네트워크에 적절히 결합될 수 있으며, 이러한 네트워크는 예를 들어, 표준 모뎀 통신, 케이블 모뎀, 디시(dish) 네트워크, ISDN, DSL(Digital Subscriber Line), 또는 다양한 무선 통신 방법들과 관련하여 전형적으로 사용되는 로컬 루프(local loop)를 통해 ISP(Internet Service Provider)에 대한 접속을 포함하고, 예를 들어, GILBERT HELD, UNDERSTANDING DATA COMMUNICATIONS (1996)를 참고하고, 그 내용은 본 명세서에 참조되어 포함된다. 네트워크는 ITV(Interactive Television) 네트워크와 같은 다른 타입들의 네트워크들로서 구현될 수 있다는 것을 유의한다. 또한, 시스템은 본 명세서에 기술된 유사한 기능을 갖는 임의의 네트워크를 통한 임의의 상품, 서비스, 또는 정보의 사용, 판매, 또는 배포를 고려한다.

[0013] 웹 클라이언트(130)는 네트워크(120)와 상호작용할 수 있는 임의의 디바이스를 포함할 수 있다. 다양한 실시예들에서, 웹 클라이언트(130)는 퍼스널 컴퓨터를 포함할 수 있다. 그러나, 다양한 실시예들에서, 웹 클라이언트(130)는 PDA, 랩톱, 셀룰러 폰, GPS 디바이스, 자동차 내비게이션 시스템, 또는 임의의 다른 디바이스를 포함할

수 있다. 다양한 타입들의 웹 클라이언트들이 본 명세서에서 더 상세히 기술된다.

[0014] 다양한 실시예들에서, ACS(110)는 복수의 애플리케이션을 GUI(170)상에 디스플레이할 수 있다. 복수의 애플리케이션은 GUI(170)상에 보여질 수 있는 임의의 애플리케이션들일 수 있다. 예를 들어, 다양한 실시예들에서 애플리케이션들은 인터넷 브라우저, 캘린더 애플리케이션, 웹 채팅 애플리케이션, 이메일 애플리케이션, 문서 뷰어 애플리케이션, 미디어 플레이어 애플리케이션, 기타 등등을 포함할 수 있다. 통상의 기술자는 본 개시와 일관되게 이용될 수 있는 무한한 개수의 애플리케이션들이 있다는 것을 알 것이다.

[0015] 도 2를 참조하면, 본 개시의 다양한 실시예들에 따라 4개의 애플리케이션을 가진 GUI(200)가 도시된다. GUI(200)는 애플리케이션 A(210), 애플리케이션 B(220), 애플리케이션 C(230), 및 애플리케이션 D(240)를 포함한다. ACS(110)는 애플리케이션들이 실질적으로 GUI(200)의 일부를 채우도록 레이아웃을 포맷팅한다. 예시된 실시예에서, GUI(200)의 일부는 실질적으로 전체 GUI(200)이다. 그러나, GUI(200)의 일부는 GUI(200)의 임의의 일부일 수 있다. ACS(110)는 디폴트 설정들, 사용자 선호들, 각각의 애플리케이션의 순위, 각각의 애플리케이션에서의 데이터의 타입과 양, 및 각각의 애플리케이션의 사용 빈도수를 포함하지만 이들로 한정되지 않는 다양한 팩터들에 근거하여 각각의 애플리케이션의 상대적 크기를 정할 수 있다.

[0016] 다양한 실시예들에서, ACS(110)는 GUI(170)상에 디스플레이될 애플리케이션의 디폴트 개수를 선택할 수 있다. 예를 들어, 다양한 실시예들에서, ACS(110)는 처음에 5개의 애플리케이션을 GUI(170)상에 동시에 디스플레이할 수 있다. 다양한 실시예들에서, ACS(110)는 사용자에 의해 가장 자주 사용되는 애플리케이션들을 추적할 수 있고, 사용자에 대한 애플리케이션의 디폴트 개수를 변경할 수 있다. 예를 들어, 사용자가 정기적으로 4개의 애플리케이션을 사용한다면, ACS(110)는 애플리케이션의 디폴트 개수를 4개로 변경할 수 있다. 사용자가 정기적으로 7개의 애플리케이션을 사용한다면, ACS(110)는 사용자에 대한 애플리케이션의 디폴트 개수를 7개로 변경할 수 있다. 다양한 실시예들에서, 사용자는 GUI(170)상에 디스플레이될 애플리케이션의 상이한 개수를 선택할 수 있다. 다양한 실시예들에서, 사용자는 디스플레이될 애플리케이션의 개수에 대한 복수의 설정을 세이브할 수 있다. 예를 들어, 사용자는 3개, 6개, 또는 10개의 애플리케이션을 디스플레이하기 위한 설정을 가질 수 있다. 사용자는 원하는 개수의 애플리케이션을 디스플레이하기 위해 세이브된 설정들 간에 전환할 수 있다.

[0017] 도 3을 참조하면, 다양한 실시예들에 따라 5개의 애플리케이션을 가진 GUI(300)가 도시된다. GUI(300)는 GUI(200)에 디스플레이된 4개의 애플리케이션을 디스플레이하지만, 추가의 애플리케이션 E(350)가 또한 디스플레이된다. 사용자가 GUI(300)에 애플리케이션 E(350)를 추가했을 수 있거나, 또는 ACS(110)가 애플리케이션 E(350)도 디스플레이되어야 한다고 결정했을 수 있다. ACS(110)는 애플리케이션들이 실질적으로 GUI(300)의 일부를 채우도록 5개의 애플리케이션에 대한 최적의 레이아웃을 결정한다. ACS(110)는 최적의 레이아웃을 제공하기 위해 이전의 애플리케이션을 이동시키거나 그 크기를 조정할 수 있다.

[0018] 다양한 실시예들에서, ACS(110)는 애플리케이션들이 실질적으로 GUI(170)의 일부를 채우도록 애플리케이션들을 GUI(170)상에 디스플레이할 수 있다. 그 일부는 GUI(170)의 임의의 일부, 예컨대 GUI(170)의 실질적으로 전부를 포함할 수 있다. ACS(110)가 애플리케이션들을 추가하거나 제거하는 것에 응답하여, 몇 개의 애플리케이션이 디스플레이되는지에 상관없이 애플리케이션들이 실질적으로 GUI(170)의 일부를 채우도록, ACS(110)는 애플리케이션들의 상대적 크기와 위치를 변경할 수 있다. 다양한 실시예들이 각각 실질적으로 직사각형으로서 디스플레이될 수 있으나, 애플리케이션들은 임의의 형상을 포함할 수 있다. ACS(110)는 각각의 애플리케이션에 대해 최적의 크기와 위치를 결정하기 위한 알고리즘을 사용할 수 있다. 알고리즘은 디스플레이되는 애플리케이션의 개수, 애플리케이션의 사용 빈도수, 애플리케이션에서 디스플레이되는 정보의 양 또는 타입, 가장 최근에 사용된 애플리케이션들, 애플리케이션의 사용자 순위 중 적어도 하나에 기초하여 다양한 애플리케이션들에 대한 최적의 크기와 위치를 결정할 수 있다. ACS(110)가 그리드에서 좌표들에 기초하여 최적의 레이아웃을 결정하기 위한 알고리즘을 사용하도록, ACS(110)는 GUI(170)를 그리드로 분할할 수 있다.

[0019] 다양한 실시예들에서, 사용자는 애플리케이션들의 크기와 위치를 커스터마이즈할 수 있다. 사용자는 애플리케이션들을 순위매김할 수 있고, ACS(110)는 애플리케이션들에 대한 최적의 레이아웃을 결정할 때 사용자 순위들을 고려할 수 있다. 사용자는 애플리케이션의 크기를 조정할 수 있고, ACS(110)는 크기 조정된 애플리케이션을 통합하는 레이아웃을 결정할 수 있다. ACS(110)는 사용자가 애플리케이션의 크기를 조정하는 것에 응답하여 다양한 애플리케이션들의 크기 및/또는 위치들을 변경할 수 있다. 다양한 실시예들에서, 사용자는 본 기술 분야에 알려진 임의의 다른 방법에 의해 애플리케이션을 드래깅하거나 크기를 변경함으로써 애플리케이션의 위치를 변경할 수 있고, ACS(110)는 애플리케이션의 새로운 위치를 통합하는 새로운 레이아웃을 결정할 수 있다. 다양한 실시예들에서, ACS(110)가 애플리케이션들의 레이아웃을 다시 포맷팅할 때, 로크된 애플리케이션의 크기 및/

또는 위치가 변하지 않도록 하기 위해, 사용자는 애플리케이션의 크기 또는 위치를 로크할 수 있다.

[0020] 다양한 실시예들에서, ACS(110)는 하나 이상의 애플리케이션이 활성 애플리케이션이라고 결정할 수 있다. ACS(110)는 사용자가 소정의 기간 내에 애플리케이션과 상호작용한다고 결정함으로써 애플리케이션이 활성 애플리케이션이라고 결정할 수 있다. 다양한 실시예들에서, ACS(110)는 복수의 애플리케이션이 활성 애플리케이션이라고 결정할 수 있다. 그러나, 다양한 실시예들에서, 임의의 주어진 시간에 하나의 애플리케이션만 활성일 수 있다. 다양한 실시예들에서, 사용자는 애플리케이션을 클릭하거나 탭핑함으로써, 또는 본 기술 분야에 알려진 임의의 다른 방법에 의해 애플리케이션을 선택함으로써 애플리케이션을 활성 애플리케이션으로 만들 수 있다.

[0021] 도 4를 참조하면, 크기가 조정된 애플리케이션을 가진 GUI(400)가 도시된다. GUI(200)와 유사하게, GUI(400)는 애플리케이션 A(210), 애플리케이션 B(220), 애플리케이션 C(230), 및 애플리케이션 D(240)를 포함한다. 그러나, GUI(400)에서, ACS(110)는 애플리케이션 A(210)이 더 커야 한다고 결정했다. 사용자는 수동으로 애플리케이션 A(210)의 크기를 조정했을 수 있다. 다양한 실시예들에서, ACS(110)는 애플리케이션 A(210)이 활성 애플리케이션이고 더 커야 한다고 결정했을 수 있다. 애플리케이션 A(210)이 활성 애플리케이션이라고 결정하는 것에 응답하여, ACS(110)는 애플리케이션 A(210)의 크기를 증가시켰다. ACS(110)는 애플리케이션들이 여전히 실질적으로 GUI(400)를 채우도록 다른 애플리케이션들의 크기를 감소시킴으로써 레이아웃을 다시 포맷팅한다.

[0022] 다양한 실시예들에서, ACS(110)는 제1 애플리케이션이 활성 애플리케이션이라고 결정할 수 있고, ACS(110)는 제1 애플리케이션이 활성인 것에 응답하여 제2 애플리케이션을 활성화할 수 있다. 예를 들어, ACS(110)는 사용자가 제1 애플리케이션을 이용한 후 단기간 내에 일반적으로 제2 애플리케이션을 이용한다고 결정할 수 있다. 제1 애플리케이션이 활성인 것에 응답하여, ACS(110)는 사용자가 제2 애플리케이션을 이용할 것이라고 예측할 수 있고, ACS(110)는 사용자가 더 쉽게 제2 애플리케이션을 이용하게 해주기 위해 제2 애플리케이션을 활성 애플리케이션으로 만들 것이다. ACS(110)는 사용자의 애플리케이션들의 이전의 사용에 기초하여 임의의 개수의 애플리케이션이 활성화되도록 결정하기 위한 복잡한 알고리즘들을 사용할 수 있다.

[0023] 애플리케이션이 활성 애플리케이션인 것에 응답하여, ACS(110)는 애플리케이션의 디스플레이를 변경할 수 있다. 예를 들어, ACS(110)는 활성 애플리케이션의 크기를 증가시킬 수 있다. 활성 애플리케이션의 크기를 증가시키는 것에 응답하여, ACS(110)는 비활성 애플리케이션들의 레이아웃을 다시 포맷팅할 수 있다. ACS(110)는 비활성 애플리케이션들의 크기를 감소시킬 수 있고/있거나 비활성 애플리케이션을 레이아웃으로부터 제거할 수 있다. 다양한 실시예들에서, ACS(110)는 활성 애플리케이션의 밝기, 불투과성, 또는 다른 시각적 특성들을 변경할 수 있다.

[0024] 다양한 실시예들에서, ACS(110)는 활성 애플리케이션에 기초하여 GUI 상에 디스플레이될 애플리케이션들을 선택할 수 있다. ACS(110)는 활성 애플리케이션과 함께 가장 많이 공통으로 사용되는 애플리케이션들을 결정할 수 있고, 공통으로 사용되는 애플리케이션들을 GUI상에 디스플레이할 수 있다. 다양한 실시예들에서, 사용자가 활성 애플리케이션이 될 애플리케이션을 선택할 때마다, ACS(110)는 디스플레이될 상이한 애플리케이션들을 선택한다. 사용자는 어느 애플리케이션들이 활성인지에 기초하여 어느 애플리케이션들이 디스플레이되어야 하는지를 커스터마니즈할 수 있다.

[0025] 다양한 실시예들에서, ACS(110)는 특정 웹 클라이언트에 대한 최적의 레이아웃을 결정할 수 있다. ACS(110)는 퍼스널 컴퓨터에 대한 모니터의 픽셀 크기를 결정하고, 모니터에 최선으로 적합한 애플리케이션들의 개수와 상대적 크기를 조정할 수 있다. 또한, ACS(110)는 웹 클라이언트가 스마트 폰 또는 태블릿 등과 같은 휴대용 컨슈머 디바이스라고 결정할 수 있고, ACS(110)는 사용자가 더 작은 디스플레이와 더 효율적으로 상호작용하게 해주기 위해 더 적은 개수의 애플리케이션들을 GUI(170)상에 디스플레이할 수 있다. 사용자는 다양한 웹 클라이언트들에 대해 제한된 디스플레이 또는 완전한 디스플레이 간에 선택하기 위한 옵션을 가질 수 있다.

[0026] 도 5를 참조하면, 다양한 실시예들에 따라 최적의 레이아웃을 결정하기 위한 프로세스(500)가 도시된다. ACS(110)는 디스플레이할 애플리케이션의 개수를 결정할 수 있다(단계 510). ACS(110)는 본 명세서에서 기술된 인자들에 기초하여 각각의 애플리케이션의 상대적 크기를 결정할 수 있다(단계 520). ACS(110)는 GUI의 그리드 크기를 결정할 수 있고, 각각의 애플리케이션에 대한 최적의 위치를 결정할 수 있다(단계 530). ACS(110)는 애플리케이션들을 최적의 레이아웃으로 디스플레이하기 위해 웹 클라이언트(130)에 데이터를 전송할 수 있다(단계 540).

[0027] 다양한 실시예들에서, 애플리케이션들은 서로 통신할 수 있다. 각각의 애플리케이션은 네트워크(120)를 통해

데이터 소스에 접속될 수 있다. 제1 애플리케이션은 제2 애플리케이션이 제1 애플리케이션에 의해 요망되는 데이터에 대한 액세스를 갖는다고 결정할 수 있다. 제1 애플리케이션은 요망되는 정보에 대한 요청을 제2 애플리케이션에 전송할 수 있고, 제2 애플리케이션은 제1 애플리케이션에 그 요망되는 정보를 전송할 수 있다.

[0028] 도 6을 참조하면, 다양한 실시예들에 따라 애플리케이션들 간에 정보를 통신하기 위한 프로세스(600)가 도시된다. 사용자는 제1 애플리케이션에 데이터를 입력할 수 있고, ACS(110)는 데이터를 수신할 수 있다(단계 610). ACS(110)는 데이터와 연관된 요청을 완수하기 위해 부가 정보가 필요하다고 결정할 수 있다. ACS(110)는 제2 애플리케이션이 그 부가 정보를 가질 가능성이 높다고 결정할 수 있다(단계 620). ACS(110)는 부가 정보에 대한 요청을 제2 애플리케이션에 전송할 수 있다(단계 630). 제2 애플리케이션은 가용의 부가 정보를 갖고 있을 수 있다. 다양한 실시예들에서, 제2 애플리케이션은 네트워크(120)를 통해 부가 정보를 획득할 수 있다. 제2 애플리케이션은 부가 정보를 ACS(110)로 전송할 수 있고, ACS(110)는 부가 정보를 제1 애플리케이션에 전송할 수 있다(단계 640). 제1 애플리케이션은 요청을 완수하기 위해 부가 정보를 이용할 수 있다(단계 650).

[0029] 다양한 실시예들에서, ACS(110)는 제2 애플리케이션이 제1 애플리케이션에서 시작된 요청을 완수하는 것이 더 효율적이라고 결정할 수 있다. 제1 애플리케이션은 바라는 요청을 ACS(110)에 전송할 수 있다. ACS(110)는 요청이 제2 애플리케이션에 의해 더 효율적으로 완수될 수 있다고 결정할 수 있다. ACS(110)는 요청을 제2 애플리케이션에 전송할 수 있고, 제2 애플리케이션은 요청을 완수할 수 있다.

[0030] 도 7을 참조하여, 다양한 실시예들에 따라 통신을 보내기 위한 프로세스가 예시의 스크린 샷을 참조하여 기술된다. 도 7은 교육 설정에서 관리자에 의해 이용될 수 있는 GUI상의 레이아웃(700)을 도시한다. 레이아웃(700)은 학생 리스트 애플리케이션(710), 등급 애플리케이션(720), 학생 정보 애플리케이션(730), 및 통신 애플리케이션(740)을 포함할 수 있다. 요청의 예로서, 관리자 사용자는 특정 학생의 부모에게 이메일을 보내고 싶을 수 있다. 관리자는 학생 리스트 애플리케이션(710)으로부터 학생을 선택할 수 있다. 다양한 실시예들에서, 등급 애플리케이션(720), 학생 정보 애플리케이션(730), 및 통신 애플리케이션(740) 중 적어도 하나는 관리자가 학생을 선택하는 것에 응답하여, 그 학생에 관련된 콘텐츠를 자동으로 로딩할 수 있다. 학생 리스트 애플리케이션(710)은, 사용자가 학생을 선택했음을 지시하는 데이터를 API(160)에 전송할 수 있고, API(160)는 그 학생에 관련된 정보를 각각의 애플리케이션에 전송할 수 있다. 등급 애플리케이션(720)은 학생의 현재 등급들을 로딩할 수 있고, 학생 정보 애플리케이션(730)은 학생과 학생의 부모의 연락처 정보를 로딩할 수 있고, 통신 애플리케이션(740)은 학생의 부모에게 이메일 또는 전화 통화와 같은 통신을 시작할 수 있다. 다양한 실시예들에서, 통신 애플리케이션(740)은 API(130)를 통해 학생 정보 애플리케이션(730)에 연락처 정보를 요청할 수 있다. 관리자는 통신 애플리케이션(740)에 메시지를 입력할 수 있고 그 메시지를 학생의 부모에게 전송할 수 있다.

[0031] 다양한 실시예들에서, 관리자는 학생 리스트 애플리케이션(710)에서 관리자가 학생의 부모에게 메시지를 보내기를 원한다고 지시할 수 있다. 관리자는 학생 리스트 애플리케이션(710)에서 메시지를 타이핑할 수 있다. 그러나, 학생 리스트 애플리케이션(710)은 부모들의 연락처 정보에 액세스할 수 없거나 또는 이메일들을 보내기 위한 기능을 갖지 않을 수 있다. 학생 리스트 애플리케이션(710)은 요청을 학생 정보 애플리케이션(730)에 전송할 수 있고, 학생 정보 애플리케이션(730)은 부모의 연락처 정보를 요청과 연관시킬 수 있고, 부모들의 연락처 정보뿐 아니라 요청을 API(130)를 통해 통신 애플리케이션(740)에 보낼 수 있다. 통신 애플리케이션(740)은 그 메시지를 부모에게 전송할 수 있다.

[0032] 교육 설정에서 메시지를 보내는 것을 참조하여 설명되었지만, ACS(110)는 여러 애플리케이션들을 수반하는 일반 컴퓨팅뿐만 아니라, 의학 분야들, 사업 분야들을 포함하는, 임의의 분야에서 호환적일 수 있다. 또한, 임의의 개수의 애플리케이션이 태스크를 완수하기 위해 서로 통신할 수 있고, 메시지를 보내는 것은 단순히 통상의 기술자에 의해 이해될 ACS(110)의 여러 기능들 중 단지 하나의 예이다.

[0033] 다양한 실시예들에서, 하나 이상의 애플리케이션들은 활성 애플리케이션의 콘텐츠에 기초하여 콘텐츠를 디스플레이할 수 있다. 사용자는 제1 애플리케이션에서 정보를 입력 또는 수신할 수 있고, 제2 애플리케이션은 제1 애플리케이션의 정보와 관련된 콘텐츠를 검색 및/또는 디스플레이할 수 있다. 제2 애플리케이션은 제1 애플리케이션에서 사용된 키워드들에 기초하여 정보를 검색할 수 있다. 예를 들어, 사용자는 통신 애플리케이션을 통해 대화에 참여하고 있을 수 있다. 사용자는 대화의 일부로서 단어 "베토벤"을 타이핑할 수 있고, 베토벤에 관한 비디오를 디스플레이하는 비디오 애플리케이션, 및 베토벤과 관련된 웹사이트들에 대한 링크들을 디스플레이하는 인터넷 검색 애플리케이션 등과 같은 하나 이상의 애플리케이션이 베토벤과 관련된 콘텐츠를 디스플레이할 수 있다.

[0034] 다양한 실시예들에서, GUI상에 디스플레이되는 애플리케이션은 마스터 애플리케이션일 수 있다. 사용자는 GUI

상에 디스플레이된 애플리케이션들을 제어하기 위해 마스터 애플리케이션과 상호작용할 수 있다. 다양한 실시예들에 있어서, 사용자는 마스터 애플리케이션에서 정보를 입력할 수 있고, 일부 또는 모든 애플리케이션들이 그 정보와 연관된 콘텐츠를 디스플레이할 수 있다. 다양한 실시예들에서, 사용자는 마스터 애플리케이션에 데이터를 입력함으로써 태스크를 시작할 수 있다. 예를 들어, 사용자는 "수학 과목에서 C 등급을 가진 모든 학생들에게 공부 기술 웹사이트의 링크를 전송"이라는 문구를 입력할 수 있다. 마스터 애플리케이션은 그 문구를 ACS(110)에 전송할 수 있고, ACS(110)는 그 문구를 키워드들로 파싱할 수 있고, 언어 규칙들을 이용하여 그 문구의 의미를 결정한다. ACS(110)는 태스크를 완수하기 위해 요청들을 다양한 애플리케이션들에 전송할 수 있다. 예를 들어, ACS(110)는 수학 과목의 학생들의 리스트가 학생 리스트 애플리케이션(710)으로부터 획득될 수 있다고 결정할 수 있고, 리스트에 대한 요청을 전송할 수 있다. 리스트를 이용하여, ACS(110)는 수학 과목에서 C 등급을 가진 모든 학생들의 리스트를 식별하기 위해 등급 애플리케이션에 요청을 전송할 수 있다. ACS(110)는 공부 기술 웹사이트들에 대한 웹 검색을 수행하기 위한 요청을 인터넷 검색 애플리케이션에 전송할 수 있다. ACS(110)는 수학 과목에서 C 등급을 가진 학생들에게 공부 기술 웹사이트에 대한 링크를 보내기 위해 학생 정보 애플리케이션(730) 및 통신 애플리케이션(740)에 요청들을 전송할 수 있다.

[0035] 애플리케이션들은 실질적으로 실시간으로 네트워크(120)와 통신할 수 있다. 애플리케이션들은 현재 정보를 다운로드할 수 있다. 애플리케이션들은 최신 상태를 유지하기 위해 현재 정보를 서로 공유할 수 있다. 본 명세서에서 애플리케이션들로서 기술되지만, 다양한 실시예들에서 애플리케이션들은 웹 클라이언트(130)와는 독립적으로 동작하는 클라우드 기반의 시스템의 일부일 수 있고, 웹 클라이언트(130)는 사용자와 클라우드 기반 시스템 간의 인터페이스로서 작용한다.

[0036] 도 8을 참조하면, 다양한 실시예들에 따라 복수의 애플리케이션을 디스플레이하기 위한 프로세스가 도시된다. 사용자는 서버에 가용의 뷰들을 요청한다. 서버는 사용자에 대해 저장된 뷰들을 검색한다. ACS(110)는 레이아웃 엔진에 뷰의 상세들을 로딩한다. ACS(110)는 GUI의 타입(예를 들어, 모바일 또는 데스크톱)을 결정한다. GUI가 모바일 GUI인 것에 응답하여, ACS(110)는 대역폭을 감소시킬 수 있고 레이아웃을 모바일 친화적으로 구성할 수 있다. GUI가 데스크톱인 것에 응답하여, ACS(110)는 대형 디스플레이들 및 고해상도 그래픽들에 적합하도록 레이아웃을 스케일링할 수 있다. 각각의 애플리케이션은 그의 연관된 데이터를 로딩할 수 있다.

[0037] 도 9를 참조하면, 다양한 실시예들에 따라 레이아웃으로부터 애플리케이션을 제거하기 위한 프로세스가 도시된다. 사용자는 애플리케이션이 GUI로부터 제거되도록 요청할 수 있다. ACS(110)는 GUI로부터 그 애플리케이션을 제거할 수 있다. ACS(110)는 실질적으로 GUI를 채우도록 남은 애플리케이션들의 위치를 조정할 수 있다.

[0038] 도 10을 참조하면, 다양한 실시예들에 따라 레이아웃의 크기를 조정하기 위한 프로세스가 도시된다. 사용자는 뷰의 크기를 조정할 수 있다. ACS(110)는 배향의 변화 또는 크기의 중요한 변화가 있는지 결정할 수 있다. 배향의 변화 또는 크기의 중요한 변화에 응답하여, ACS(110)는 실질적으로 뷰를 채우기 위해 애플리케이션들을 스케일링할 수 있다. 사용자가 현저하게 뷰의 크기를 변경했다거나 또는 배향을 변경했다고 결정하는 것에 응답하여, ACS(110)는 본 명세서에서 더 기술되는 바와 같이 최적의 레이아웃을 재계산할 수 있다. 도 11을 참조하면, 다양한 실시예들에 따라 애플리케이션을 비활성화하기 위한 프로세스가 도시된다. 제1 애플리케이션은 제2 애플리케이션을 비활성화하기 위한 요청을 ACS(110)에 전송할 수 있다. ACS(110)는 제1 애플리케이션이 제2 애플리케이션을 비활성화하기 위한 적절한 허가를 갖는지 결정할 수 있다. 제1 애플리케이션이 적절한 허가를 갖지 않는다고 결정하는 것에 응답하여, ACS(110)는 요청이 거부된다는 통지를 GUI에 전송할 수 있다. 제1 애플리케이션이 제2 애플리케이션을 비활성화하기 위한 적절한 허가를 갖는다는 결정에 응답하여, ACS(110)는 제2 애플리케이션을 비활성화할 수 있고, GUI상의 디스플레이를 갱신할 수 있다.

[0039] 도 12를 참조하면, 다양한 실시예들에 따라 애플리케이션의 크기를 증가시키기 위한 프로세스가 도시된다. 사용자는 애플리케이션이 더 큰 모드에서 표시되도록 요청할 수 있다. ACS(110)는 애플리케이션을 더 큰 모드에서 디스플레이할 수 있고, ACS(110)는 실질적으로 GUI의 나머지 부분을 채우도록 나머지 애플리케이션들의 위치를 조정할 수 있다.

[0040] 도 13을 참조하면, 다양한 실시예들에 따라 애플리케이션을 이동시키기 위한 프로세스가 도시된다. 사용자는 GUI상의 애플리케이션을 이동시키기 위해 애플리케이션을 드래깅할 수 있거나 또는 달리 요청을 지시할 수 있다. ACS(110)는 사용자가 애플리케이션을 이동시킴에 따라 실시간으로 최적의 레이아웃을 다시 포맷팅할 수 있다. 사용자가 무효 위치에 애플리케이션을 배치하는 것에 응답하여, ACS(110)는 디스플레이를 이전 레이아웃으로 되돌릴 수 있다. 사용자가 유효 위치에 애플리케이션을 배치하는 것에 응답하여, ACS(110)는 사용자의 바라는 위치에서 애플리케이션을 가진 최적의 레이아웃을 재계산할 수 있다.

[0041] 본 명세서에서 논의되는 임의의 통신, 전송, 및/또는 채널은 콘텐츠(예를 들어, 데이터, 정보, 메타데이터, 기타 등등)를 전달하기 위한 그리고/또는 콘텐츠 자체를 위한 임의의 시스템 또는 방법을 포함할 수 있다. 콘텐츠는 임의의 형태 또는 매체에 제시될 수 있고, 다양한 실시예들에서, 콘텐츠는 전자적으로 전달될 수 있고/있거나 전자적으로 제시될 수 있다. 예를 들어, 채널은 웹사이트, 유니폼 리소스 로케이터(Uniform Resource Locator: "URL"), 문서(예를 들어, Microsoft Word 문서, Microsoft Excel 문서, Adobe.pdf 문서, 기타 등등), "전자책", "전자잡지", 애플리케이션 또는 마이크로애플리케이션(후술됨), SMS 또는 다른 타입의 텍스트 메시지, 이메일, 페이스북, 트위터, MMS 및/또는 다른 타입의 통신 기술을 포함할 수 있다. 다양한 실시예들에서, 채널은 데이터 파트너에 의해 호스팅될 수 있거나 또는 제공될 수 있다. 다양한 실시예들에서, 배포 채널은 머천트 웹사이트, 소셜 미디어 웹사이트, 제휴 또는 파트너 웹사이트, 외부 벤더, 모바일 디바이스 통신, 소셜 미디어 네트워크 및/또는 위치 기반 서비스 중 적어도 하나를 포함할 수 있다. 배포 채널은 머천트 웹사이트, 소셜 미디어 사이트, 제휴 또는 파트너 웹사이트, 외부 벤더, 및 모바일 디바이스 통신 중 적어도 하나를 포함할 수 있다. 소셜 미디어 사이트들의 예들은 Facebook®, foursquare®, Twitter®, My Space®, LinkedIn®, 및 기타 등등을 포함한다. 또한, 모바일 디바이스 통신의 예들은 스마트폰들에 대한 텍스팅, 이메일, 및 모바일 애플리케이션들을 포함한다.

[0042] "소비자 프로파일" 또는 "소비자 프로파일 데이터"는 소비자와 연관된 속성(예를 들어, 선호, 관심, 인구 통계 정보, 개인 식별 정보, 및 기타 등등)을 기술하는 소비자에 관한 임의의 정보 또는 데이터를 포함할 수 있다.

[0043] 다양한 실시예들에서, 본 명세서에서 기술되는 방법들은 본 명세서에서 기술되는 다양한 특정 머신들을 이용하여 구현된다. 본 명세서에서 기술되는 방법들은 통상의 기술자에 의해 즉시 이해될 바와 같이, 하기의 특정 머신들 및 이 후 개발되는 것들을 이용하여 임의의 적절한 조합으로 구현될 수 있다. 또한, 본 개시로부터 명백한 바와 같이, 본 명세서에서 기술되는 방법들은 특정 아티클들의 다양한 변환들을 야기할 수 있다.

[0044] 또한, 간결함을 위해, 시스템들(그리고 시스템들의 개별 동작 컴포넌트들 중의 컴포넌트들)의 종래의 데이터 네트워킹, 애플리케이션 개발, 및 다른 기능 양태들은 본 명세서에 상세히 기술되지 않을 수 있다. 또한, 본 명세서에 포함된 다양한 도면들에 도시된 연결선들은 다양한 소자들 간의 예시적인 기능적 관계들 및/또는 물리적 연결들을 나타내고자 한다. 많은 대안적인 또는 추가적인 기능적 관계들 또는 물리적 연결들이 실제 시스템에 존재할 수 있음을 유의한다.

[0045] 본 명세서에서 논의되는 다양한 시스템 컴포넌트들은 다음 중 하나 이상을 포함할 수 있다: 디지털 데이터를 처리하는 프로세서를 포함하는 호스트 서버 또는 다른 컴퓨팅 시스템들; 디지털 데이터를 저장하는 프로세서에 연결된 메모리; 디지털 데이터를 입력하는 프로세서에 연결된 입력 디지타이저; 메모리에 저장되고 프로세서에 의한 디지털 데이터의 처리를 지령하는 프로세서에 의해 액세스 가능한 애플리케이션 프로그램; 프로세서에 의해 처리된 디지털 데이터로부터 유도된 정보를 디스플레이하기 위한 프로세서 및 메모리에 연결된 디스플레이 디바이스; 및 복수의 데이터베이스. 본 명세서에 사용되는 다양한 데이터베이스들은 다음을 포함할 수 있다: 클라이언트 데이터; 머천트 데이터; 금융 기관 데이터; 및/또는 시스템의 운영에 유용한 유사한 데이터. 통상의 기술자가 이해할 바와 같이, 사용자 컴퓨터는 운영 시스템(예를 들어, Windows NT, Windows 95/98/2000, Windows XP, Windows Vista, Windows 7, OS2, UNIX, Linux, Solaris, MacOS, 등)뿐만 아니라 다양한 종래의 지원 소프트웨어 및 컴퓨터들과 전형적으로 연관된 드라이버들을 포함할 수 있다.

[0046] 본 시스템 또는 그것의 임의의 일부(들) 또는 기능(들)은 하드웨어, 소프트웨어, 또는 이들의 조합을 사용하여 구현될 수도 있고, 하나 이상의 컴퓨터 시스템 또는 다른 프로세싱 시스템들에서 구현될 수도 있다. 그러나, 실시예들에 의해 수행되는 조작들은 사람 운영자에 의해 수행되는 정신적 작용들과 통상 연관된 매칭 또는 선택 등의 용어들로 종종 지칭되었다. 본 명세서에 기술된 동작들 중 임의의 동작들에 있어서 대부분의 경우에, 이러한 사람 운영자의 기능이 반드시 필요한 것은 아니며, 바람직하지 않을 수 있다. 오히려, 동작들은 머신 동작들일 수 있다. 다양한 실시예들을 수행하기 위한 유용한 머신들은 범용 디지털 컴퓨터들 또는 유사한 디바이스들을 포함한다.

[0047] 사실, 다양한 실시예들에서, 실시예들은 본 명세서에 설명된 기능을 수행할 수 있는 하나 이상의 컴퓨터 시스템에 관한 것이다. 컴퓨터 시스템은 하나 이상의 프로세서를 포함한다. 프로세서는 통신 인프라스트럭처(예를 들어, 통신 버스, 크로스 오버 바아, 또는 네트워크)에 연결된다. 다양한 소프트웨어 실시예들은 이 예시적 컴퓨터 시스템에 관해 설명된다. 이러한 설명을 읽은 후에, 다양한 실시예들을 다른 컴퓨터 시스템들 및/또는 아키텍처들을 사용하여 구현하는 방법은 관련 기술 분야(들)의 통상의 기술자에게 명백해질 것이다. 컴퓨터 시스템은 디스플레이 유닛 상에서의 표시를 위해 통신 인프라스트럭처로부터(또는 도시되지 않은 프레임

버퍼로부터) 그래픽, 텍스트, 및 기타 데이터를 전달하는 디스플레이 인터페이스를 포함할 수 있다.

[0048] 컴퓨터 시스템은 또한 메인 메모리, 예컨대 랜덤 액세스 메모리(RAM)를 포함하고, 보조 메모리도 포함할 수 있다. 보조 메모리는, 예를 들어 하드 디스크 드라이브, 및/또는 플로피 디스크 드라이브, 자기 테이프 드라이브, 광학 디스크 드라이브, 등을 나타내는 이동식 저장소 드라이브를 포함할 수 있다. 이동식 저장소 드라이브는 공지된 방식으로 이동식 저장소 유닛으로부터 판독하고/하거나 그것에 기입한다. 이동식 저장소 유닛은 이동식 저장소 드라이브에 의해 판독되고 기입되는 플로피 디스크, 자기 테이프, 광학 디스크 등을 나타낸다. 이해하게 될 것인 바와 같이, 이동식 저장소 유닛은 컴퓨터 소프트웨어 및/또는 데이터가 저장되어 있는 컴퓨터 이용 가능 저장 매체를 포함한다.

[0049] 다양한 실시예들에서, 보조 메모리는 컴퓨터 프로그램 또는 기타 명령들이 컴퓨터 시스템으로 로딩되도록 해주는 다른 유사 디바이스들을 포함할 수 있다. 이러한 디바이스들은 예를 들어, 이동식 저장소 유닛과 인터페이스를 포함할 수 있다. 그러한 것의 예들로는 프로그램 카트리지 및 카트리지 인터페이스(예컨대 비디오 게임 디바이스들에서 발견되는 것 등), 이동식 메모리 칩(예를 들어, 소거 가능한 프로그래머블 판독 전용 메모리(EPROM), 또는 프로그래머블 판독 전용 메모리(PROM)) 및 부속 소켓, 및 기타 이동식 저장소 유닛들 및 인터페이스들을 포함할 수 있고, 이들은 소프트웨어 및 데이터가 이동식 저장소 유닛으로부터 컴퓨터 시스템으로 전송되도록 해준다.

[0050] 컴퓨터 시스템은 또한 통신 인터페이스를 포함할 수 있다. 통신 인터페이스는 소프트웨어와 데이터가 컴퓨터 시스템과 외부 디바이스들 사이에 전송되도록 해준다. 통신 인터페이스의 예들은 모뎀, 네트워크 인터페이스(예컨대 이더넷 카드), 통신 포트, PCMCIA(Personal Computer Memory Card International Association) 슬롯 및 카드, 기타 등등을 포함할 수 있다. 통신 인터페이스를 통해 전달되는 소프트웨어 및 데이터는 전자 신호, 전자기 신호, 광학 신호, 또는 통신 인터페이스에 의해 수신될 수 있는 다른 신호들일 수 있는 신호들의 형태이다. 이러한 신호들은 통신 경로(예를 들어, 채널)를 통해 통신 인터페이스에 제공된다. 이러한 채널은 신호들을 전달하고, 와이어, 케이블, 광 섬유, 전화선, 셀룰러 링크, 라디오 주파수(RF) 링크, 무선 및 다른 통신 채널들을 사용하여 구현될 수 있다.

[0051] 용어들 "컴퓨터 프로그램 매체"와 "컴퓨터 이용 가능 매체"는 일반적으로 이동식 저장소 드라이브 등의 매체 및 하드 디스크 드라이브에 인스톨된 하드 디스크를 지칭하기 위해 사용된다. 이러한 컴퓨터 프로그램 제품들은 소프트웨어를 컴퓨터 시스템에 제공한다.

[0052] 컴퓨터 프로그램들(컴퓨터 제어 로직이라고도 지칭됨)은 메인 메모리 및/또는 보조 메모리에 저장된다. 컴퓨터 프로그램들은 또한 통신 인터페이스를 통해 수신될 수 있다. 그러한 컴퓨터 프로그램들은, 실행될 때, 본 명세서에서 논의된 바와 같은 특징들을 컴퓨터 시스템이 수행할 수 있게 한다. 특히, 컴퓨터 프로그램들은, 실행될 때, 프로세서가 다양한 실시예들의 특징들을 수행할 수 있게 한다. 따라서, 그러한 컴퓨터 프로그램들은 컴퓨터 시스템의 제어기들을 나타낸다.

[0053] 다양한 실시예들에서, 소프트웨어는 이동식 저장소 드라이브, 하드 디스크 드라이브, 또는 통신 인터페이스를 이용하여 컴퓨터 프로그램 제품에 저장될 수 있고 컴퓨터 시스템에 로딩될 수 있다. 제어 로직(소프트웨어)은, 프로세서에 의해 실행될 때, 프로세서가 본 명세서에 기재된 다양한 실시예들의 기능들을 수행하게 한다. 다양한 실시예들에서, 주문형 집적 회로들(ASICs)과 같은 하드웨어 컴포넌트들. 관련 기술 분야(들)의 통상의 기술자에게는 본 명세서에 기술된 기능들을 수행하기 위한 하드웨어 상태 머신(state machine)의 구현이 자명할 것이다.

[0054] 다양한 실시예들에서, 서버는 애플리케이션 서버들(예를 들어, WEB SPHERE, WEB LOGIC, JBOSS)를 포함할 수 있다. 다양한 실시예들에서, 서버는 웹 서버들(예를 들어, APACHE, IIS, GWS, SUN JAVA SYSTEM WEB SERVER)을 포함할 수 있다.

[0055] 웹 클라이언트는 예를 들어, 본 명세서에서 논의된 것들과 같은 임의의 네트워크를 통해 통신하는 임의의 디바이스(예를 들어, 퍼스널 컴퓨터)를 포함한다. 그러한 브라우저 애플리케이션들은 온라인 트랜잭션들 및/또는 통신을 수행하기 위해 컴퓨터 유닛 또는 시스템 내에 설치된 인터넷 브라우징 소프트웨어를 포함한다. 이러한 컴퓨팅 유닛들 또는 시스템들은 컴퓨터 또는 컴퓨터들의 세트의 형태를 취할 수 있지만, 랩톱들, 노트북들, 태블릿들, 핸드헬드 컴퓨터들, 개인용 정보 단말기들, 셋탑 박스들, 워크스테이션들, 컴퓨터-서버들, 메인 프레임 컴퓨터들, 미니-컴퓨터들, PC 서버들, 편재형 컴퓨터들, 컴퓨터들의 네트워크 세트들, 퍼스널 컴퓨터들, 예컨대 iPad들, iMAC들, 및 MacBook들, 키오스크들, 단말기들, 판매점(POS) 디바이스들 및/또는 단말기들, 텔레비전들,

또는 네트워크를 통해 데이터를 수신할 수 있는 임의의 다른 디바이스를 포함한, 다른 타입의 컴퓨팅 유닛들 또는 시스템들이 이용될 수 있다. 웹-클라이언트는 Microsoft Internet Explorer(마이크로소프트 인터넷 익스플로러), Mozilla Firefox(모질라 파이어폭스), Google Chrome(구글 크롬), Apple Safari(애플 사파리), 또는 인터넷 브라우징에 이용 가능한 무수한 소프트웨어 패키지들 중 임의의 다른 것을 실행할 수 있다.

[0056] 전문가들은 웹 클라이언트가 애플리케이션 서버와 직접 연락할 수 있거나 그렇지 않을 수 있다는 것을 알 것이다. 예를 들어, 웹 클라이언트는 인터넷 서버에 직접 또는 간접 접속을 가질 수 있는 다른 서버 및/또는 하드웨어 컴포넌트를 통해 애플리케이션 서버의 서비스들에 액세스할 수 있다. 예를 들어, 웹 클라이언트는 부하 조정기를 통해 애플리케이션 서버와 통신할 수 있다. 다양한 실시예들에서, 액세스는 상업적으로 가용인 웹-브라우저 소프트웨어 패키지를 통해 네트워크 또는 인터넷을 통한다.

[0057] 통상의 기술자가 이해할 것인 바와 같이, 웹 클라이언트는 운영 시스템(예를 들어, Windows NT, 95/98/2000/CE/모바일, OS2, UNIX, Linux, Solaris, MacOS, PalmOS, 기타 등등)뿐만 아니라 컴퓨터들과 전형적으로 연관된 다양한 종래의 지원 소프트웨어 및 드라이버들을 포함할 수 있지만, 이에 한정되지 않는다. 웹 클라이언트는 임의의 적절한 퍼스널 컴퓨터, 네트워크 컴퓨터, 워크스테이션, 개인용 정보 단말기, 셀룰러 폰, 스마트 폰, 미니 컴퓨터, 메인프레임, 또는 그와 유사한 것 등을 포함할 수 있지만 이에 한정되지 않는다. 웹 클라이언트는 네트워크에의 액세스를 갖는 홈 또는 비즈니스 환경에 있을 수 있다. 다양한 실시예들에서, 액세스는 상업적으로 가용인 웹-브라우저 소프트웨어 패키지를 통해 네트워크 또는 인터넷을 통한다. 웹 클라이언트는 보안 소켓 계층(Secure Sockets Layer: SSL) 및 전송 계층 보안(TLS) 등과 같은 보안 프로토콜들을 구현할 수 있다. 웹 클라이언트는 http, https, ftp, 및 sftp를 포함한 여러 애플리케이션 계층 프로토콜들을 구현할 수 있다.

[0058] 다양한 실시예들에서, 시스템(100)의 컴포넌트들, 모듈들, 및/또는 엔진들은 마이크로-애플리케이션들 또는 마이크로-앱들로서 구현될 수 있다. 마이크로-앱들은 전형적으로 예를 들어, 팜(Palm) 모바일 운영 시스템, 윈도우즈(Windows) 모바일 운영 시스템, 안드로이드(Android) 운영 시스템, 애플(Apple) iOS, 블랙베리(Blackberry) 운영 시스템, 및 기타 등등을 포함한, 모바일 운영 시스템의 정황에서 전형적으로 배치된다. 마이크로-앱은 다양한 운영 시스템들과 하드웨어 리소스들의 동작들을 통제하는 미리 정해진 규칙들의 세트를 통해 더 큰 운영 시스템 및 연관된 하드웨어의 리소스들에 영향을 주기 위해 구성될 수 있다. 예를 들어, 마이크로-앱이 모바일 디바이스 또는 모바일 운영 시스템 외의 디바이스 또는 네트워크와 통신하기를 원하는 경우에, 마이크로 앱은 모바일 운영 시스템의 소정의 규칙들하에서 운영 시스템 및 연관된 디바이스 하드웨어의 통신 프로토콜에 영향을 줄 수 있다. 또한, 마이크로-앱이 사용자로부터의 입력을 원하는 경우에, 마이크로-앱은 다양한 하드웨어 컴포넌트들을 모니터링하고 그 후 하드웨어로부터 검출된 입력을 마이크로-앱으로 통신하는 운영 시스템으로부터의 응답을 요청하도록 구성될 수 있다.

[0059] "클라우드" 또는 "클라우드 컴퓨팅"은 최소의 관리 노력 또는 서비스 제공자 상호작용에 의해 빨리 제공될 수 있고 해제될 수 있는 구성가능 컴퓨팅 리소스들(예를 들어, 네트워크들, 서버들, 스토리지, 애플리케이션들, 및 서비스들)의 공유 풀에 대한 편리한 주문형 네트워크 액세스를 가능하게 하기 위한 모델을 포함한다. 클라우드 컴퓨팅은 로케이션-의존적 컴퓨팅을 포함할 수 있음으로써, 공유되는 서버들은 주문시에 컴퓨터들 및 다른 디바이스들에 리소스들, 소프트웨어, 및 데이터를 제공한다. 클라우드 컴퓨팅에 관한 추가의 정보는, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc (2011년 2월 4일 최후 방문함)에서 NIST (National Institute of Standards and Technology: 미국 국립 표준 기술 연구소)의 클라우드 컴퓨팅의 정의를 참조하고, 그 전체 내용이 본 명세서에 참조되어 포함된다.

[0060] 본 명세서에 사용된, "전송"은 하나의 시스템 컴포넌트로부터 네트워크 접속을 경유하여 또 다른 것으로 전자 데이터를 보내는 것을 포함할 수 있다. 또한, 본 명세서에 사용된, "데이터"는 명령들, 쿼리들, 파일들, 스토리지의 데이터, 및 디지털 또는 임의의 다른 형태의 유사한 것 등과 같은 포괄적 정보를 포함할 수 있다.

[0061] 시스템은 웹 서비스들, 유틸리티 컴퓨팅, 편재형 및 개별화된 컴퓨팅, 보안 및 아이덴티티 솔루션, 자동 컴퓨팅, 클라우드 컴퓨팅, 범용 컴퓨팅, 이동 및 무선 솔루션, 오픈 소스, 바이오메트릭, 그리드 컴퓨팅 및/또는 메시 컴퓨팅과 연관되는 사용들을 고려한다.

[0062] 본 명세서에서 논의되는 임의의 데이터베이스들은 관계적인, 계층적인, 그래픽적인, 또는 객체 지향적인 구조 및/또는 임의의 다른 데이터베이스 구성들을 포함할 수 있다. 데이터베이스들을 구현하는 데 사용될 수 있는 공통적인 데이터베이스 제품들은 IBM(Armonk, NY)의 DB2, Oracle Corporation(Redwood Shores, CA)로부터의 가용의 다양한 데이터베이스 제품들, Microsoft Corporation(Redmond, Washington)의 Microsoft Access 또는 Microsoft SQL Server, MySQL AB(Uppsala, Sweden)의 MySQL, 또는 임의의 다른 적합한 데이터베이스 제품을 포

함한다. 또한, 데이터베이스들은 예를 들어, 데이터 테이블들 또는 룩업 테이블들로서 임의의 적절한 방식으로 조직될 수 있다. 각각의 레코드는 단일 파일, 일련의 파일들, 링크된 일련의 데이터 필드들, 또는 임의의 다른 데이터 구조일 수 있다. 특정 데이터의 연관(association)은 본 기술 분야에서 알려진 또는 실시되는 것들 등의 임의의 원하는 데이터 연관 기법을 통해 달성될 수 있다. 예를 들어, 연관은 수동으로 또는 자동으로 달성될 수 있다. 자동 연관은 예를 들어, 검색들, 모든 테이블들과 파일들을 통한 순차 검색들의 속력을 높이기 위해 테이블들에서 키 필드를 이용하고, 검색을 간단화하기 위한 공지된 순서에 따라 파일 내의 레코드들을 소팅하는, 그리고/또는 기타 등등을 행하는 데이터베이스 검색, 데이터베이스 병합, GREP, AGREP, SQL을 포함할 수 있다. 연관 단계는 예를 들어, 미리 선택된 데이터베이스들 또는 데이터 섹터들 내의 "키 필드(key field)"를 사용하여 데이터베이스 병합 기능에 의해 달성될 수 있다. 다양한 데이터베이스 튜닝 단계들이 데이터베이스 성능을 최적화하기 위해 고려된다. 예를 들어, 인덱스들 등과 같은 빈번히 사용되는 파일들이 입력/출력("I/O") 병목 상태를 감소시키기 위해 별도의 파일 시스템들에 배치될 수 있다.

[0063] 더 구체적으로, "키 필드"는 키 필드에 의해 정의된 객체들의 고-레벨 클래스에 따라 데이터베이스를 분할한다. 예를 들어, 특정의 타입들의 데이터가 복수의 관련된 데이터 테이블들 내의 키 필드로서 지정될 수 있고, 그 후 데이터 테이블들은 키 필드 내의 데이터의 타입에 기초하여 링크될 수 있다. 각각의 링크된 데이터 테이블들 내의 키 필드에 대응하는 데이터는 바람직하게는 동일하거나 또는 동일한 타입이다. 그러나, 키 필드들 내의, 동일하지는 않지만, 유사한 데이터를 갖는 데이터 테이블들은 또한 예를 들어, AGREP을 사용하여 링크될 수 있다. 다양한 실시예들에 따르면, 임의의 적절한 데이터 저장 기법이 표준 포맷 없이 데이터를 저장하기 위해 이용될 수 있다. 데이터 세트들은, 예를 들어, ISO/IEC 7816-4 파일 구조를 이용하여 개별적인 파일들을 저장하기, 도메인을 구현함으로써, 하나 이상의 데이터 세트를 포함하는 하나 이상의 기본적인 파일을 노출하는 전용 파일이 선택되도록 하기; 계층적인 파일링 시스템을 이용하여 개별 파일들 내에 저장된 데이터 세트들을 이용하기; 데이터 세트들은 단일 파일 내의 레코드들로서 저장됨(압축, SQL 액세스 가능, 제1 튜플(tuple)에 의한 알파벳, 숫자, 하나 초과의 키들을 통해 해쉬됨 등을 포함함); 바이너리 대형 오브젝트(BLOB); ISO/IEC 7816-6 데이터 요소들을 이용하여 인코딩된 그룹핑되지 않은 데이터 요소들로서 저장됨; ISO/IEC 8824 및 8825에서와 같이 ISO/IEC Abstract Syntax Notation(ASN.1)을 이용하여 인코딩된 그룹핑되지 않은 데이터 요소들로서 저장됨; 그리고/또는 프랙탈 압축 방법들, 이미지 압축 방법들 등을 포함할 수 있는 기타 독점 기법들 등을 포함하는 임의의 적절한 기법을 이용하여 저장될 수 있다.

[0064] 다양한 실시예들에서, 광범위하게 다양한 정보를 상이한 포맷들로 저장하는 능력은 정보를 BLOB로서 저장함으로써 용이하게 된다. 그러므로, 임의의 바이너리 정보는 데이터 세트와 연관된 저장소 공간에 저장될 수 있다. 전술한 바와 같이, 바이너리 정보는 금융 트랜잭션 도구에 저장될 수 있거나 또는 금융 트랜잭션 도구의 외부에 그와 제휴되어 저장될 수 있다. BLOB 방법은 고정된 저장소 할당을 이용하는 고정된 메모리 오프셋, 순환 큐 기법들, 또는 메모리 관리에 대한 최선의 관행(예를 들어, 페이징된 메모리, 최근에 가장 적게 사용된 것, 등)을 통해 데이터 세트들을 바이너리의 블록으로서 포맷팅된, 그룹핑되지 않은 데이터 요소들로서 저장할 수 있다. BLOB 방법들을 사용함으로써, 상이한 포맷들을 갖는 다양한 데이터 세트들을 저장하기 위한 능력은 데이터 세트들의 복수의 무관한 소유자들에 의한 금융 트랜잭션 도구와 연관된 데이터의 저장을 용이하게 한다. 예를 들어, 저장될 수 있는 제1 데이터 세트는 제1자에 의해 제공될 수 있고, 저장될 수 있는 제2 데이터 세트는 무관한 제2자에 의해 제공될 수 있으며, 저장될 수 있는 제3 데이터 세트는 또한 제1자 및 제2자와 무관한 제3자에 의해 제공될 수 있다. 이러한 3개의 예시적 데이터 세트들 각각은 상이한 데이터 저장소 포맷들 및/또는 기법들을 이용하여 저장되는 상이한 정보를 포함할 수 있다. 또한, 각각의 데이터 세트는 또한 다른 서브세트들과 구분될 수 있는 데이터의 서브세트들을 포함할 수 있다.

[0065] 전술한 바와 같이, 다양한 실시예들에서, 데이터는 공통 포맷과 관계없이 저장될 수 있다. 그러나, 다양한 실시예들에서, 데이터 세트(예를 들어, BLOB)는 금융 트랜잭션 도구 상의 데이터를 조작하기 위해 제공될 때 표준 방식으로 주석이 달릴 수 있다. 주석은 짧은 헤더, 트레일러(trailer), 또는 다양한 데이터 세트들을 관리하는 데 유용한 정보를 전달하도록 구성되는 각각의 데이터 세트와 관련된 다른 적절한 지시자를 포함할 수 있다. 예를 들어, 주석은 본 명세서에서 "조건 헤더", "헤더", "트레일러", 또는 "상태"라고 불릴 수 있고, 데이터 세트의 상태의 지시를 포함할 수 있거나, 또는 데이터의 특정 발행자 또는 소유자와 상관된 식별자를 포함할 수 있다. 일 예에서, 각각의 데이터 세트 BLOB의 처음 3개의 바이트는 그 특정 데이터 세트의 상태; 예를 들어, LOADED(로딩됨), INITIALIZED(초기화됨), READY(준비), BLOCKED(차단됨), REMOVABLE(제거가능), 또는 DELETED(삭제됨)를 지시하도록 구성될 수 있거나 구성 가능할 수 있다. 데이터의 연속적 바이트들이 예를 들어, 발행자, 사용자, 트랜잭션/멤버십 계정 식별자 또는 그와 유사한 것의 아이덴티티를 지시하기 위해 사용될 수 있다.

이러한 조건 주석들 각각은 본 명세서에서 더 논의된다.

[0066] 데이터 세트 주석은 또한 다른 타입들의 상태 정보뿐만 아니라 다양한 다른 목적을 위해서도 사용될 수 있다. 예를 들어, 데이터 세트 주석은 액세스 레벨들을 확립하는 보안 정보를 포함할 수 있다. 액세스 레벨들은 예를 들어, 특정 개인들, 피고용자들의 레벨들, 회사들, 또는 다른 주체들만이 특정 데이터 세트들에 액세스하게 허용하도록, 또는 트랜잭션, 머천트, 발행자, 또는 사용자 등에 기초하여 특정 데이터 세트들에 대한 액세스를 허용하도록 구성될 수 있다. 또한, 보안 정보는 특정 동작들 예컨대 데이터 세트들에 대한 액세싱, 변경, 및/또는 삭제 등의 특정 동작들만을 규제/허용할 수 있다. 일 예에서, 데이터 세트 주석은, 데이터 세트 소유자 또는 사용자만이 데이터 세트를 삭제하도록 허용되고, 다양한 식별된 사용자들이 판독을 위해 데이터 세트에 액세스하도록 허용될 수 있고, 그 외의 자들은 모두 데이터 세트에 액세스하는 것이 배제된다는 것을 지시한다. 그러나, 다른 액세스 제한 파라미터들이 또한 다양한 주체들이 다양한 허용 레벨들을 가지고 적절히 데이터 세트에 액세스하도록 해주기 위해 사용될 수 있다.

[0067] 헤더 또는 트레일러를 포함하는 데이터는 헤더 또는 트레일러에 따라 데이터를 추가, 삭제, 변경, 또는 증대하도록 구성된 독립형 상호작용 디바이스에 의해 수신될 수 있다. 이에 따라 다양한 실시예들에서, 헤더 또는 트레일러는 연관된 발행자-소유의 데이터와 함께 트랜잭션 디바이스에 저장되지 않고, 그 대신에 독립형 디바이스에서 트랜잭션 도구 사용자에게 제공함으로써 적절한 동작이 취해질 수 있고, 그 동작에 대한 적절한 옵션이 취해질 수 있다. 시스템은, 데이터의 헤더 또는 트레일러, 또는 헤더 또는 트레일러 이력이 적절한 데이터와 관련하여 트랜잭션 도구에 저장되는 데이터 저장소 구성을 고려할 수 있다.

[0068] 통상의 기술자는 또한 보안 사유로 인해, 임의의 데이터베이스, 시스템, 디바이스, 서버, 또는 시스템의 다른 컴포넌트들이 하나의 위치 또는 복수의 위치에서 그들의 임의의 조합으로 구성될 수 있다는 점을 이해할 것이며, 각각의 데이터베이스 또는 시스템은 방화벽, 액세스 코드들, 암호화, 암호해제, 압축, 압축해제, 및/또는 기타 등등과 같은 다양한 적절한 보안 특징들 중 임의의 것을 포함한다.

[0069] 암호화는 본 기술 분야에서 지금 가용인 또는 가용으로 될 수 있는 임의의 기법들 - 예를 들어, Twofish, RSA, El Gamal, Schorr 시그니처, DSA, PGP, PKI, GPG(GnuPG), 및 대칭 및 비대칭 암호체계 - 에 의해 수행될 수 있다.

[0070] 웹 클라이언트의 컴퓨팅 유닛은 본 기술 분야에 알려진 표준 다이얼업, 케이블, DSL, 또는 임의의 다른 인터넷 프로토콜을 이용하여 인터넷 또는 인트라넷에 접속된 인터넷 브라우저를 더 구비할 수 있다. 웹 클라이언트에서 발생하는 트랜잭션들은 다른 네트워크들의 사용자들로부터의 허가받지 않은 액세스를 방지하기 위해 방화벽을 통과할 수 있다. 또한, 부가적 방화벽들이 ACS의 가변 컴포넌트들 사이에 보안성을 더 강화하기 위해 배치될 수 있다.

[0071] 방화벽은 ACS 컴포넌트들 및/또는 기업 컴퓨팅 리소스들을 다른 네트워크들의 사용자들로부터 보호하기 위해 적절히 구성된 임의의 하드웨어 및/또는 소프트웨어를 포함할 수 있다. 또한, 방화벽은 웹 서버를 통해 접속하는 웹 클라이언트들에 대해 방화벽 뒤의 다양한 시스템들 및 컴포넌트들에의 액세스를 제한하거나 제약하도록 구성될 수 있다. 방화벽은 무엇보다도 상태 기반 검사, 프록시 기반, 액세스 제어 리스트들, 및 패킷 필터링을 포함하는 가변 구성들에 상주할 수 있다. 방화벽은 웹 서버 또는 임의의 다른 ACS 컴포넌트들 내에 통합될 수 있거나, 또는 개별 엔티티로서 추가로 상주할 수 있다. 방화벽은 네트워크 어드레스 변환(network address translation: "NAT") 및/또는 네트워크 어드레스 포트 변환("NAPT")을 구현할 수 있다. 방화벽은 가상 사설 네트워킹에 사용되는 것들 등과 같은 보안 통신들을 용이하게 하기 위해 다양한 터널링 프로토콜들을 수용할 수 있다. 방화벽은 인터넷 등과 같은 공용 네트워크와의 통신들을 용이하게 하기 위해 비무장 존("DMZ")을 구현할 수 있다. 방화벽은 인터넷 서버 내에 소프트웨어로서, 임의의 다른 애플리케이션 서버 컴포넌트들로서 통합될 수 있거나, 또는 다른 컴퓨팅 디바이스 내에 상주할 수 있거나, 또는 독립형 하드웨어 컴포넌트의 형태를 취할 수 있다.

[0072] 본 명세서에서 논의되는 컴퓨터들은 사용자들에 의해 액세스 가능한 적절한 웹사이트 또는 다른 인터넷 기반 그래픽 사용자 인터페이스를 제공할 수 있다. 다양한 실시예들에서, Microsoft Internet Information Server(IIS), Microsoft Transaction Server(MTS), 및 Microsoft SQL Server가 Microsoft 운영 시스템, Microsoft NT 웹 서버 소프트웨어, Microsoft SQL Server 데이터베이스 시스템, 및 Microsoft Commerce Server와 함께 사용된다. 또한, Access 또는 Microsoft SQL Server, Oracle, Sybase, Informix MySQL, Interbase 등과 같은 컴포넌트들이 ADO(Active Data Object) 준수 데이터베이스 관리 시스템을 제공하기 위해 사용될 수 있다. 다양한 실시예들에서, Apache 웹 서버는 Linux 운영 시스템, MySQL 데이터베이스, 및 Perl, PHP, 및/또는

Python 프로그래밍 언어들과 함께 사용된다.

[0073] 본 명세서에서 논의되는 통신들, 입력들, 저장소, 데이터베이스들 또는 디스플레이들 중 임의의 것은 웹 페이지들을 갖는 웹사이트를 통해 용이해질 수 있다. 용어 "웹 페이지"는 본 명세서에 사용되는 바와 같이, 사용자와 상호작용하는 데 사용할 수 있는 문서들 및 애플리케이션들의 타입을 제한하고자 의도되지 않는다. 예를 들어, 전형적인 웹사이트는 표준 HTML 문서들 외에도 다양한 형태들, Java 애플릿(applet), JavaScript, ASP(active server page), CGI(common gateway interface) 스크립트, XML(extensible markup language: 확장 가능 마크업 랭귀지), 동적 HTML, CSS(cascading style sheet), AJAX(Asynchronous Javascript And XML), 헬퍼 애플리케이션, 및 플러그인(plug-in) 등을 포함할 수 있다. 서버는 URL(http://yahoo.com/stockquotes/ge) 및 IP 어드레스(123.56.789.234)를 포함하는 요청을 웹 서버로부터 수신하는 웹 서비스를 포함할 수 있다. 웹 서버는 적절한 웹 페이지들을 검색해내고, 웹 페이지들에 대한 데이터 또는 애플리케이션들을 IP 어드레스에 보낸다. 웹 서비스들은 인터넷과 같은 통신 수단을 통해 다른 애플리케이션들과 상호작용할 수 있는 애플리케이션들이다. 웹 서비스들은 전형적으로 XML, SOAP, AJAX, WSDL, 및 UDDI와 같은 표준 또는 프로토콜에 기초한다. 웹 서비스 방법들은 본 기술 분야에 잘 알려져 있고, 많은 표준 문헌들에서 다루어진다. 예를 들어, ALEX NGHIEM, IT WEB SERVICES: A ROADMAP FOR THE ENTERPRISE (2003)를 참조하고, 그 내용은 본 명세서에 참조되어 포함된다.

[0074] 미들웨어는 개별 컴퓨팅 시스템들 사이에 통신 및/또는 프로세스 트랜잭션들을 용이하게 하기 위해 적절히 구성된 임의의 하드웨어 및/또는 소프트웨어를 포함할 수 있다. 미들웨어 컴포넌트들은 상업적으로 가용이고, 본 기술 분야에 알려져 있다. 미들웨어는 상업적으로 가용인 하드웨어 및/또는 소프트웨어를 통해, 주문형 하드웨어 및/또는 소프트웨어 컴포넌트들을 통해, 또는 그들의 조합을 통해 구현될 수 있다. 미들웨어는 다양한 구성들에 상주할 수 있고, 독립형 시스템으로서 존재할 수 있거나, 또는 인터넷 서버상에 상주하는 소프트웨어 컴포넌트일 수 있다. 미들웨어는 본 명세서에 개시된 목적들 중 임의의 것을 위해 애플리케이션 서버의 다양한 컴포넌트들과 임의의 개수의 내부 또는 외부 시스템들 간의 트랜잭션들을 처리하도록 구성될 수 있다. IBM Inc.(Armonk, NY)에 의한 WebSphere MQTM(이전에는 MQSeries)은 상업적으로 가용인 미들웨어 제품의 예이다. 엔터프라이즈 서비스 버스(Enterprise Service Bus: "ESB") 애플리케이션은 미들웨어의 다른 예이다.

[0075] 전문가들은 또한 브라우저 기반 문서 내에 데이터를 디스플레이하기 위한 다수의 방법들이 있다는 것을 알 것이다. 데이터는 표준 텍스트로서, 또는 고정된 리스트, 스크롤 가능 리스트, 드롭-다운 리스트, 편집 가능 텍스트 필드, 고정 텍스트 필드, 팝업식 윈도우, 및 그와 유사한 것 등 내에 표현될 수 있다. 마찬가지로, 예를 들어 키보드를 이용하여 자유로운 텍스트 입력, 메뉴 항목들, 체크 박스들, 옵션 박스들의 선택, 및 그와 유사한 것 등과 같이 웹 페이지에서 데이터를 수정하기 위해 가용인 다수의 방법들이 있다.

[0076] 시스템 및 방법은 기능 블록 컴포넌트들, 스크린 샷들, 옵션적 선택들, 및 다양한 프로세싱 단계들에 대해 본 명세서에서 기술될 수 있다. 그러한 기능 블록들이 특정 기능들을 수행하기 위해 구성된 임의의 개수의 하드웨어 및/또는 소프트웨어 컴포넌트들에 의해 실현될 수 있다는 것을 이해할 것이다. 예를 들어, 시스템은 하나 이상의 마이크로프로세서 또는 다른 제어 디바이스의 제어 하에 다양한 기능들을 수행할 수 있는 다양한 집적 회로 컴포넌트들 예컨대, 메모리 소자들, 프로세싱 소자들, 로직 소자들, 및 검색 테이블들, 및 등등을 채택할 수 있다. 유사하게, 시스템의 소프트웨어 요소들은 C, C++, C#, Java, JavaScript, VBScript, Macromedia Cold Fusion, COBOL, Microsoft Active Server Pages, 어셈블리, PERL, PHP, 오크(awk), Python, Visual Basic, SQL Stored Procedures, PL/SQL, 임의의 Unix 쉘 스크립트, 및 XML(확장 가능 마크업 랭귀지) 등과 같은 임의의 프로그래밍 또는 스크립팅 랭귀지로 구현될 수 있고, 다양한 알고리즘들은 데이터 구조들, 오브젝트들, 프로세스들, 루틴들, 또는 다른 프로그래밍 요소들의 임의의 조합으로 구현된다. 또한, 시스템이 데이터 전송, 시그널링, 데이터 처리, 네트워크 제어, 및 등등을 위한 임의의 개수의 종래의 기법을 채택할 수 있다는 것에 유의해야 한다. 또한 심지어, 시스템은 JavaScript, VBScript 또는 기타 등등과 같은 클라이언트측 스크립팅 랭귀지에 의한 보안 문제를 검출 또는 예방하기 위해 사용될 수 있다. 암호법 및 네트워크 보안의 기초적 소개를 위해, 다음의 참조 문헌들 중 임의의 것을 참조하고: (1) "Applied Cryptography: Protocols, Algorithms, And Source Code In C," by Bruce Schneier, published by John Wiley & Sons (second edition, 1995); (2) "Java Cryptography" by Jonathan Knudson, published by O'Reilly & Associates (1998); (3) "Cryptography & Network Security: Principles & Practice" by William Stallings, published by Prentice Hall; 이들 모두는 본 명세서에 참조되어 포함된다.

[0077] 통상의 기술자에 의해 이해될 것인 바와 같이, 시스템은 기존의 시스템의 커스터마이제이션, 애드온(add-on) 제품, 업그레이드된 소프트웨어를 실행하는 프로세싱 장치, 독립형 시스템, 분산형 시스템, 방법, 데이터 프로세싱 시스템, 데이터 프로세싱을 위한 디바이스, 및/또는 컴퓨터 프로그램 제품으로서 구현될 수 있다. 따라서,

시스템 또는 모듈의 임의의 일부는 코드를 실행하는 프로세싱 장치, 인터넷 기반 실시예, 전적으로 하드웨어 실시예, 또는 인터넷, 소프트웨어, 및 하드웨어의 양태들을 조합한 실시예의 형태를 취할 수 있다. 또한, 시스템은 저장 매체 내에 구현된 컴퓨터 판독 가능 프로그램 코드 수단을 갖는 컴퓨터 판독 가능 저장 매체상의 컴퓨터 프로그램 제품의 형태를 취할 수 있다. 하드 디스크, CD-ROM, 광학 저장 디바이스, 자기 저장 디바이스, 및/또는 기타 등등을 포함한 임의의 적합한 컴퓨터 판독 가능 저장 매체가 활용될 수 있다.

[0078] 본 명세서에서 시스템 및 방법은 다양한 실시예들에 따라 방법들, 장치(예를 들어, 시스템들), 및 컴퓨터 프로그램 제품들의 스크린 샷들, 블록도들, 및 흐름도들을 참조하여 기술된다. 블록도들 및 흐름도들의 각각의 기능 블록과, 블록도들 및 흐름도들의 각각의 기능 블록들의 조합들은 각각 컴퓨터 프로그램 명령어들에 의해 구현될 수 있다는 것을 잘 이해할 것이다.

[0079] 이러한 컴퓨터 프로그램 명령들은 범용 컴퓨터, 특수 목적 컴퓨터, 또는 다른 프로그래머블 데이터 프로세싱 장치에 로딩될 수 있어서 머신을 생성할 수 있으며, 이로써 컴퓨터 또는 다른 프로그래머블 데이터 프로세싱 장치상에서 실행되는 명령들은 흐름도 블록 또는 블록들에서 지정된 기능들을 구현하기 위한 수단을 생성한다. 컴퓨터 또는 다른 프로그래머블 데이터 프로세싱 장치에 특정 방식으로 동작하도록 지시할 수 있는 이러한 컴퓨터 프로그램 명령들은 또한 컴퓨터 판독 가능 메모리에 저장될 수 있고, 이로써 컴퓨터 판독 가능 메모리 제품에 저장된 명령어들은 흐름도 블록 또는 블록들에서 지정된 기능을 구현하는 명령어 수단을 포함한 제품을 생성한다. 컴퓨터 프로그램 명령어들은 또한 컴퓨터 또는 다른 프로그래머블 데이터 프로세싱 장치상에 로딩될 수 있어서, 일련의 동작 단계들로 하여금 컴퓨터 또는 다른 프로그래머블 데이터 프로세싱 장치상에서 수행되게 하여 컴퓨터 구현된 프로세스를 생성할 수 있으며, 이로써 컴퓨터 또는 다른 프로그래머블 데이터 프로세싱 장치상에서 실행하는 명령어들은 흐름도 블록 또는 블록들에서 지정된 기능들을 구현하는 단계들을 제공한다.

[0080] 따라서, 블록도들 및 흐름도들의 기능 블록들은 특정 기능들을 수행하는 수단들의 조합들, 특정 기능들을 수행하는 단계들의 조합들, 및 특정 기능들을 수행하는 프로그램 명령어 수단들을 지원한다. 또한 블록도들 및/또는 흐름도들의 각각의 기능 블록, 및 블록도들 및 흐름도들의 기능 블록들의 조합들이 특정 기능들 또는 단계들을 수행하는 특수 목적의 하드웨어-기반 시스템들, 또는 특수 목적의 하드웨어 및 컴퓨터 명령어들의 적절한 조합들에 의해 구현될 수 있다는 것이 이해될 것이다. 또한, 프로세스 흐름도와 그 설명은 사용자 윈도우들, 웹 페이지들, 웹사이트들, 웹 형태들, 프롬프트들, 등등을 참조할 수 있다. 전문가들은 본 명세서에서 기술된 예시의 단계들이 윈도우들, 웹페이지들, 웹 형태들, 팝업 윈도우들, 프롬프트들, 및 기타 등등의 이용을 포함한 임의의 개수의 구성들로 이루어질 수 있다는 것을 이해할 것이다. 또한 예시되고 설명된 여러 단계들은 단일 웹페이지들 및/또는 윈도우들로 결합될 수 있지만 간결성을 위해 확장되었다는 것이 더 이해되어야 한다. 다른 경우들에서, 단일 프로세스 단계들로서 예시되고 기술된 단계들은 복수의 웹페이지 및/또는 윈도우로 분리될 수 있지만, 간결성을 위해 결합되었다.

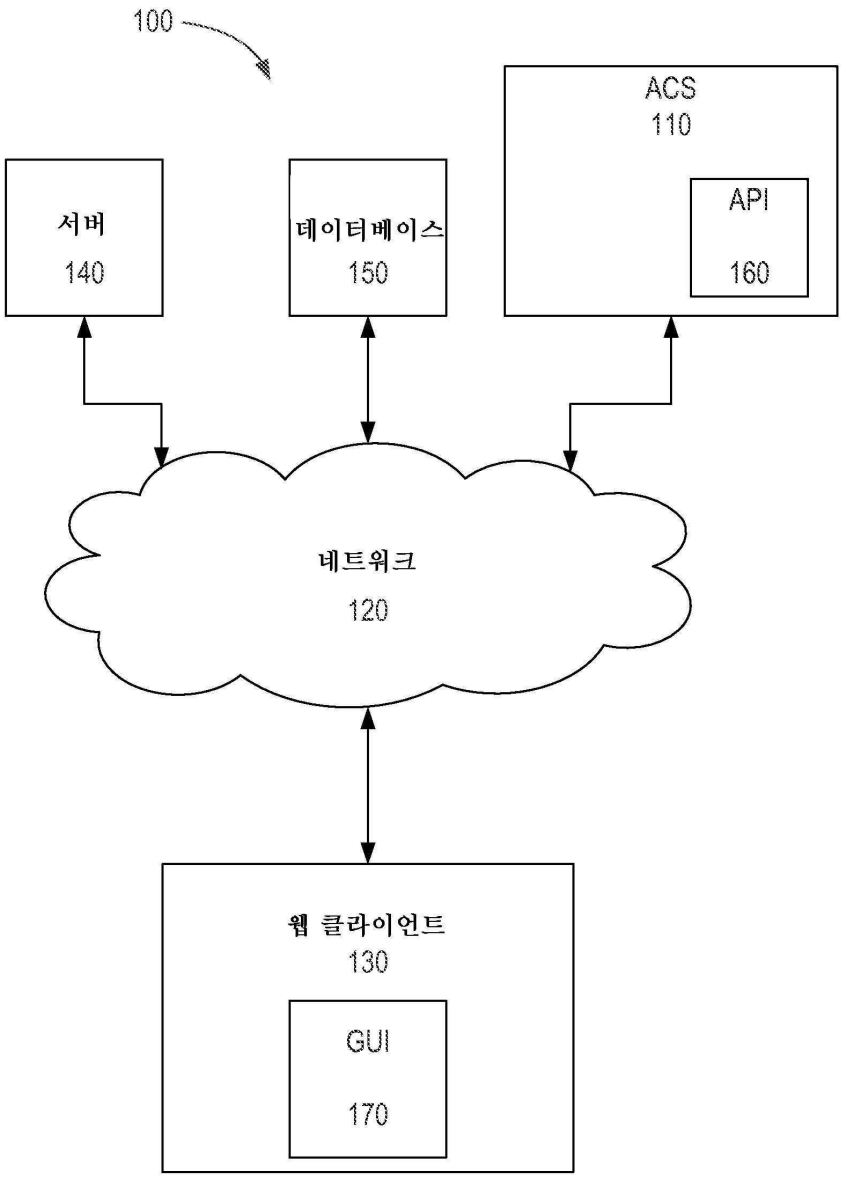
[0081] "비일시적"이란 용어는 청구 범위로부터 일시적 신호들의 전파 자체만을 삭제거하는 것이고, 단지 일시적 신호들의 전파 자체만은 아닌 모든 표준 컴퓨터 판독 가능 매체에 대한 권리들을 포기하는 것은 아니라는 것을 이해해야 한다. 달리 말하면, "비일시적 컴퓨터 판독 가능 매체"와 "비일시적 컴퓨터 판독 가능 저장 매체"라는 용어의 의미는, 35 U.S.C. § 101에 따른 특허가능 주제의 범위 밖에 해당한다고 In Re Nuijten에서 발견되었던 그런 타입들의 일시적 컴퓨터 판독 가능 매체만을 배제하는 것으로 이해되어야 한다.

[0082] 이점들, 다른 장점들, 및 문제들에 대한 해결책들이 특정 실시예들과 관련하여 설명되었다. 그러나, 이점들, 장점들, 및 문제들에 대한 해결책들, 및 임의의 이점, 장점, 및 해결책을 발생시킬 수 있거나 또는 더 현저하게 되도록 할 수 있는 임의의 요소들이 본 개시의 중요한, 필요한, 또는 본질적인 특징들 또는 요소들로서 해석되지 말아야 한다. 단수의 요소에 대한 참조는 명시적으로 그렇게 기술하지 않는 한 "오직 하나"를 의미하는 것을 의도하지 않고, 오히려 "하나 이상"을 의미하도록 의도한다. 또한, 'A, B, 및 C 중 적어도 하나' 또는 'A, B, 또는 C 중 적어도 하나'와 유사한 문구가 청구항들 또는 명세서에 사용될 경우에, 이 문구는 실시예에 A가 단독으로 존재할 수 있거나, 실시예에 B가 단독으로 존재할 수 있거나, 실시예에 C가 단독으로 존재할 수 있거나, 또는 A, B, 및 C의 임의의 조합; 예를 들어, A와 B, A와 C, B와 C, 또는 A 및 B 및 C가 단일 실시예에 존재할 수 있다는 것을 의도한다. 본 개시가 방법을 포함하지만, 예컨대 자기 또는 광학 메모리, 또는 자기 또는 광학 디스크 등과 같은 실체의 컴퓨터 판독 가능 캐리어상의 컴퓨터 프로그램 명령어들로서 구현될 수 있다고 생각된다. 본 기술 분야의 통상의 기술자에게 알려진, 전술한 예시적 실시예들의 구성요소들과의 모든 구조적, 화학적, 및 기능적 등가물들은 본 명세서에 참조되어 명시적으로 포함되고, 본 청구범위에 포함되도록 의도된다. 또한, 디바이스 또는 방법이 본 청구 범위에 포함되기 위해, 그 디바이스 또는 방법이 본 개시에 의해 해결하고자 하는 각각의 그리고 모든 문제를 해결할 필요는 없다. 또한, 본 개시의 어떤 구성요소, 컴포넌

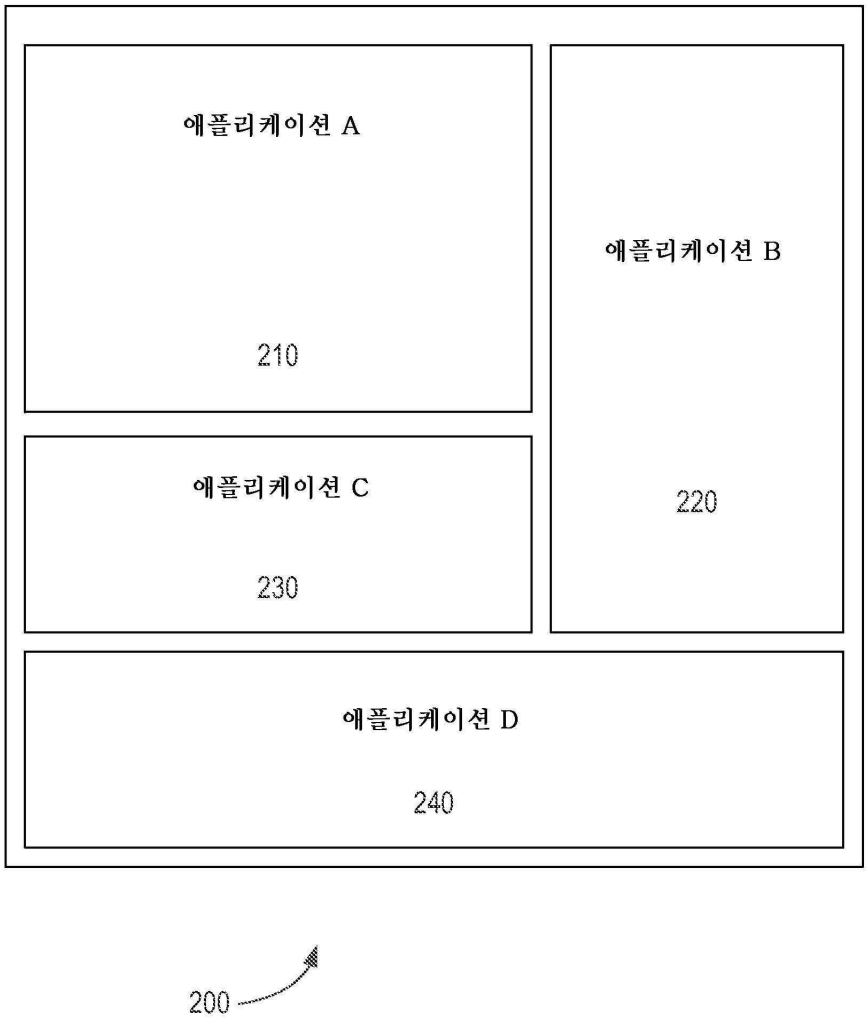
트, 또는 방법 단계도 그 구성요소, 컴포넌트, 또는 방법 단계가 본 청구항들에 명시적으로 기재되는지에 상관없이 공중에 헌정되도록 의도되지 않는다. 본 명세서의 어떤 청구항 구성요소도, 그 구성요소가 "하는 수단"이란 문구를 사용하여 명시적으로 기재되지 않는 한, 35 U.S.C. 112, 제6절의 정의에 따라 해석되어서는 안 된다. 본 명세서에서 사용된 바와 같이, 용어들 "포함하다", "포함하는"이라는 용어들 또는 그들의 임의의 다른 변형들은 비배타적인 포함을 커버하도록 의도되므로, 구성요소들의 리스트를 포함하는 프로세스, 방법, 물품, 또는 장치가 그러한 구성요소들만을 포함하는 것이 아니라, 그러한 프로세스, 방법, 물품, 또는 장치에 고유하거나 또는 명시적으로 열거되지 않은 다른 구성요소들을 포함할 수 있다.

도면

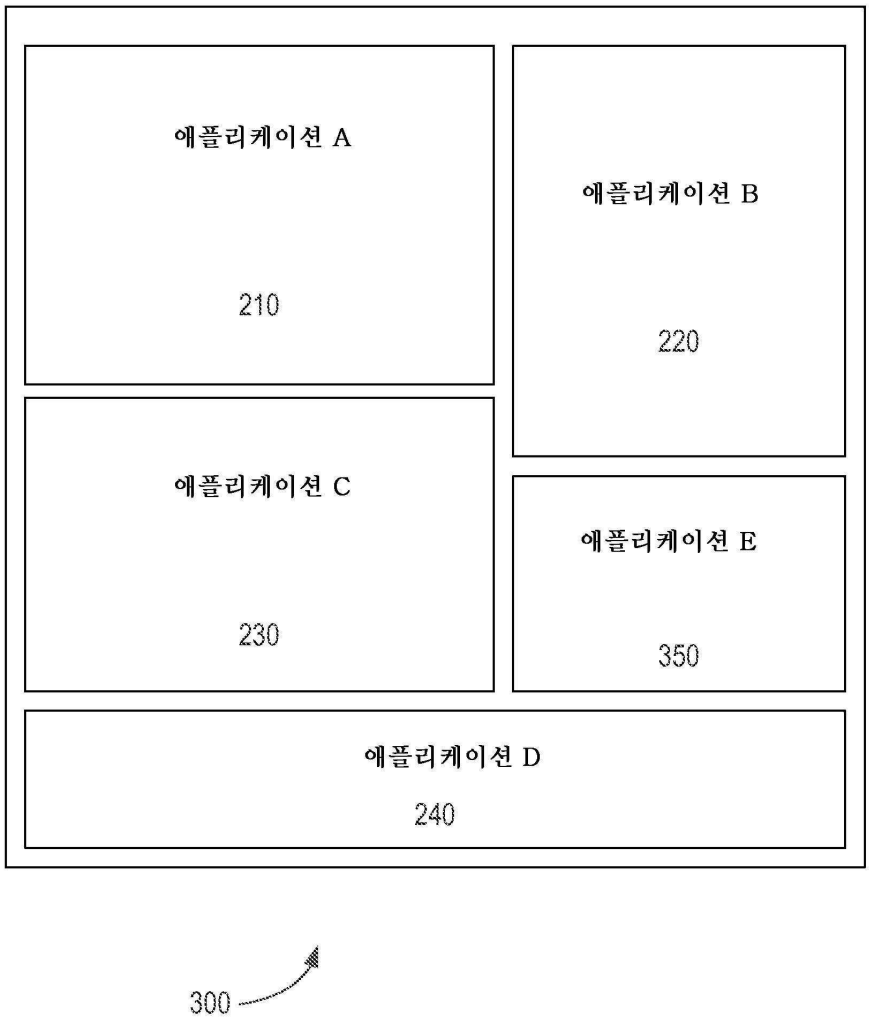
도면1



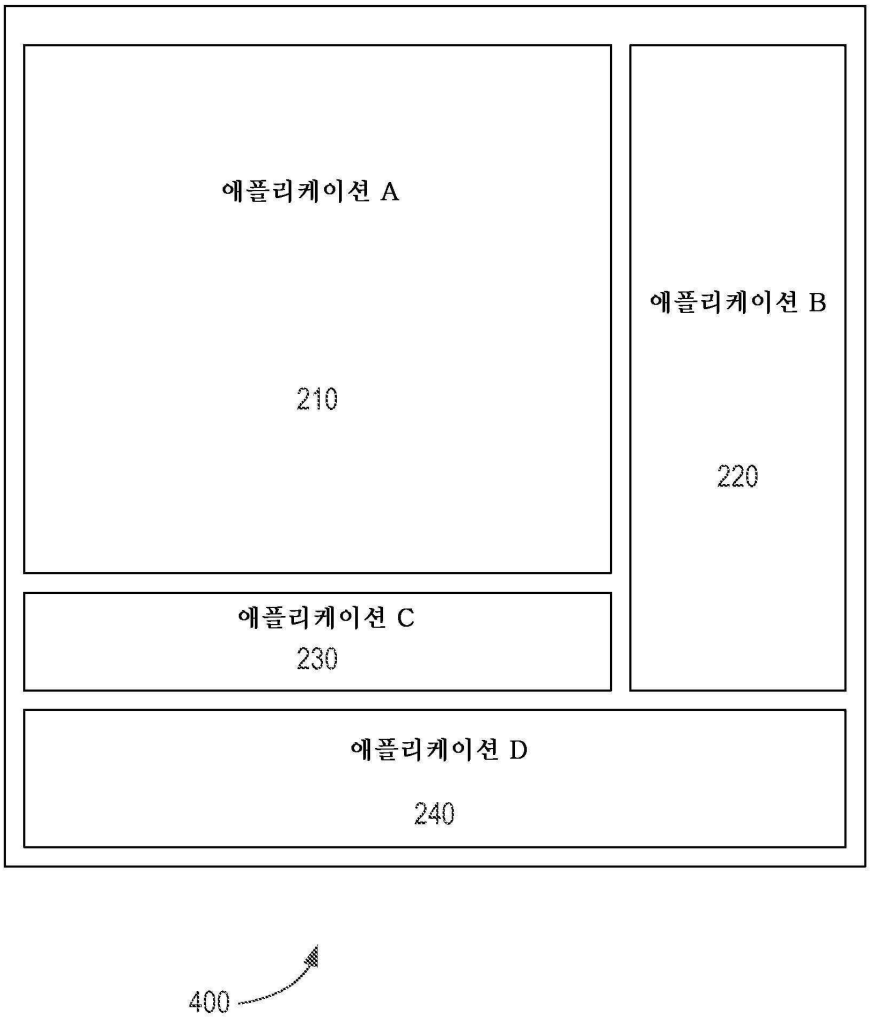
도면2



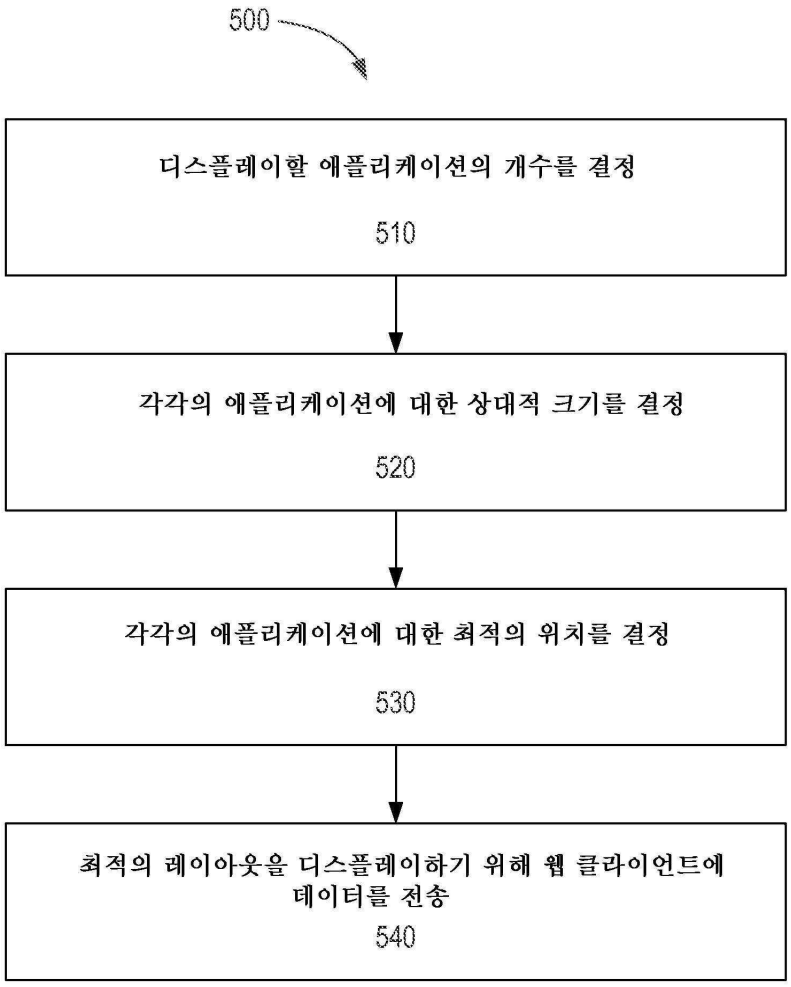
도면3


애플리케이션 A
210
애플리케이션 B
220
애플리케이션 C
230
애플리케이션 E
350
애플리케이션 D
240
300

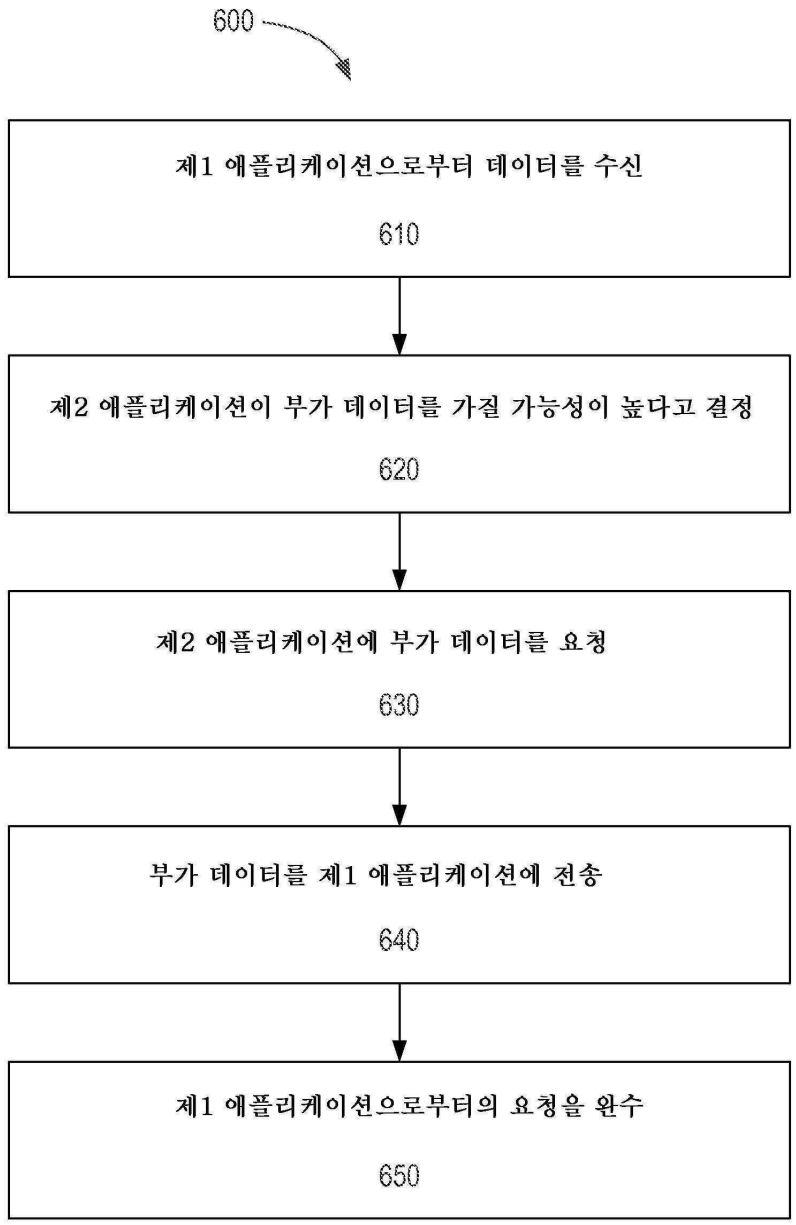
도면4



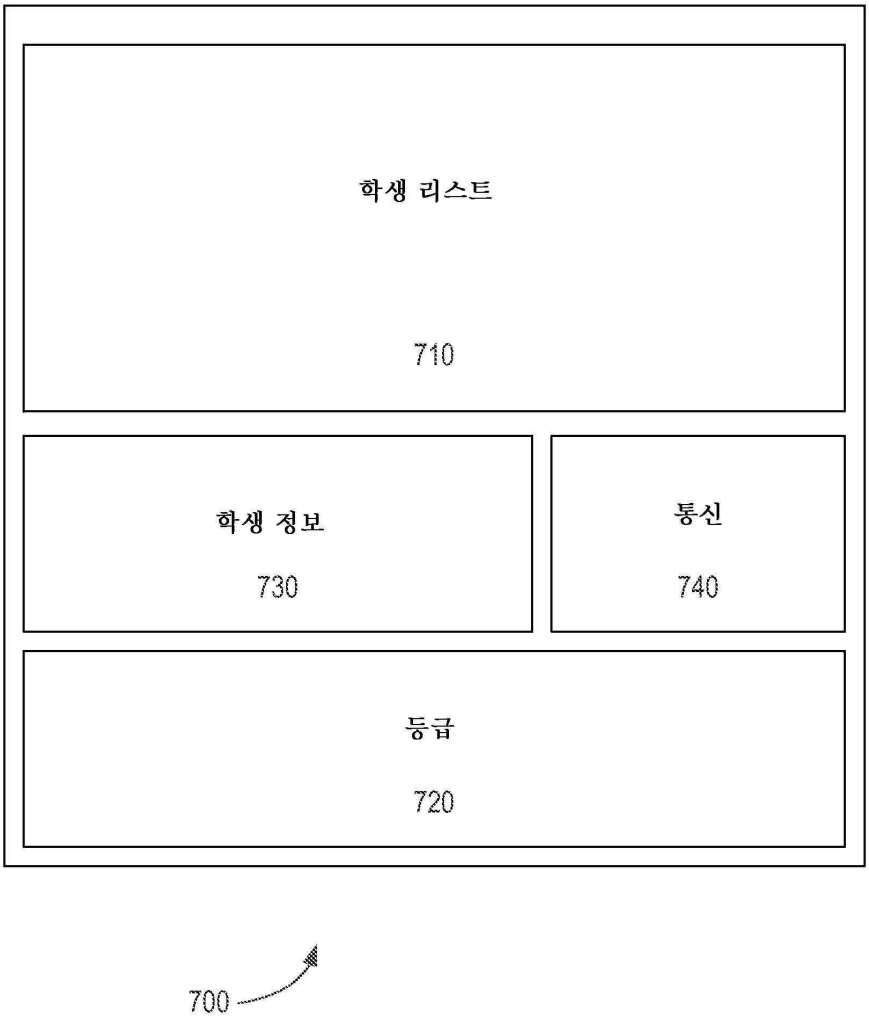
도면5


500
디스플레이할 애플리케이션의 개수를 결정
510
각각의 애플리케이션에 대한 상대적 크기를 결정
520
각각의 애플리케이션에 대한 최적의 위치를 결정
530
최적의 레이아웃을 디스플레이하기 위해 웹 클라이언트에 데이터를 전송
540

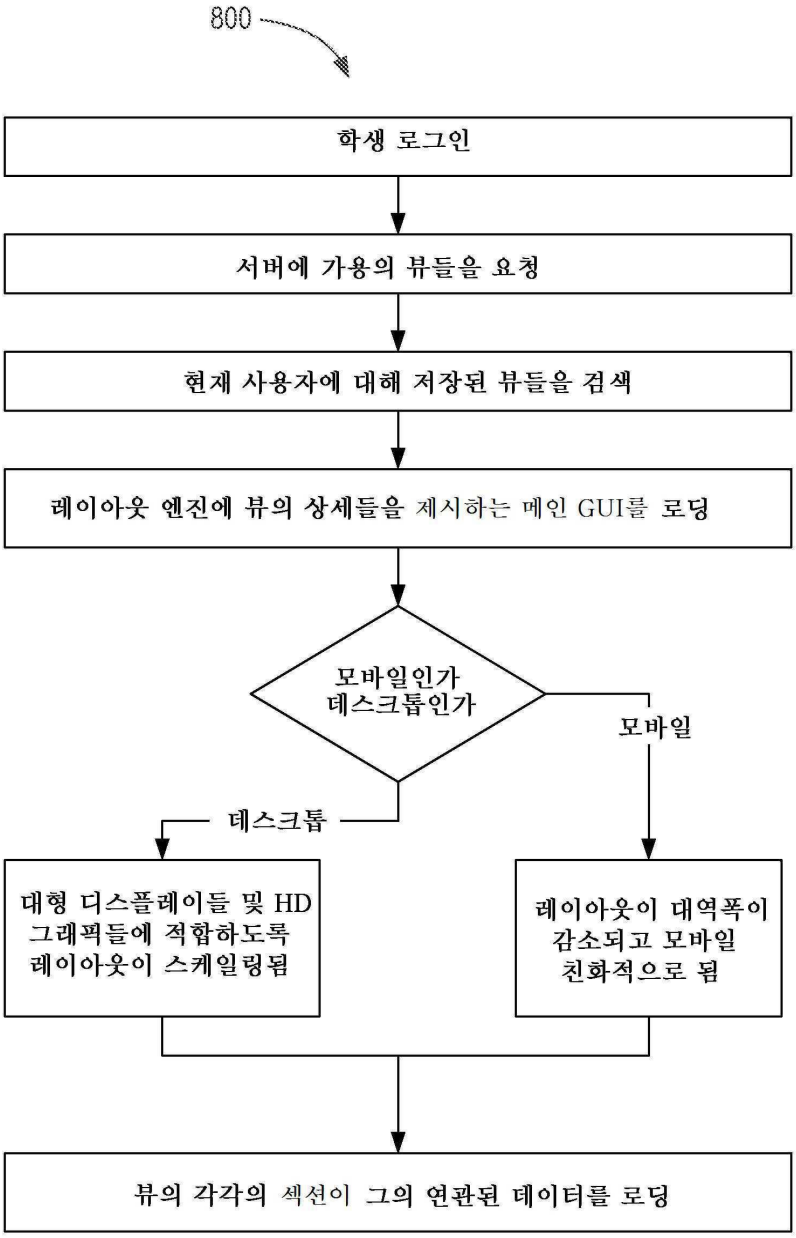
도면6


600
제1 애플리케이션으로부터 데이터를 수신
610
제2 애플리케이션이 부가 데이터를 가질 가능성이 높다고 결정
620
제2 애플리케이션에 부가 데이터를 요청
630
부가 데이터를 제1 애플리케이션에 전송
640
제1 애플리케이션으로부터의 요청을 완수
650

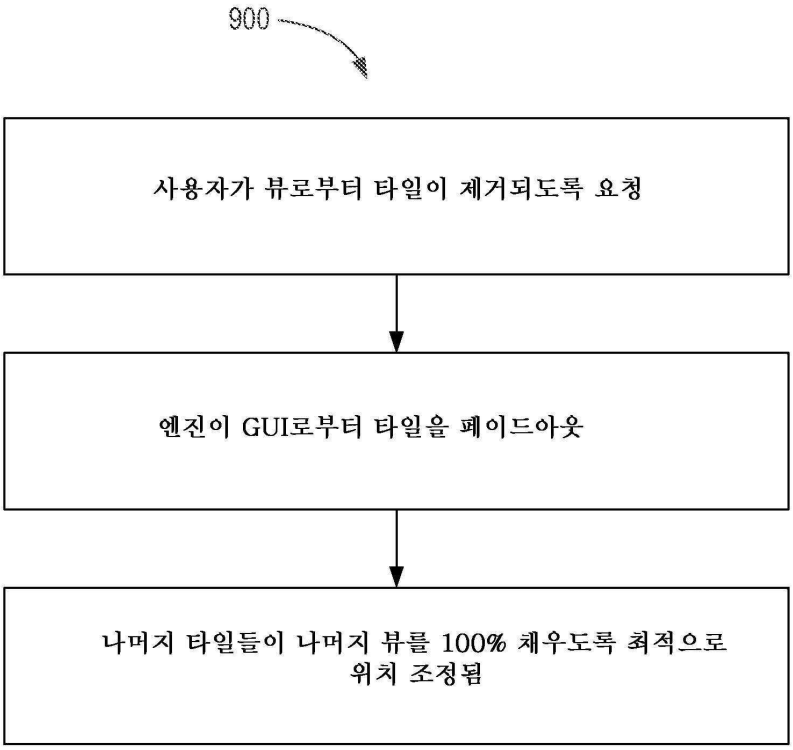
도면7



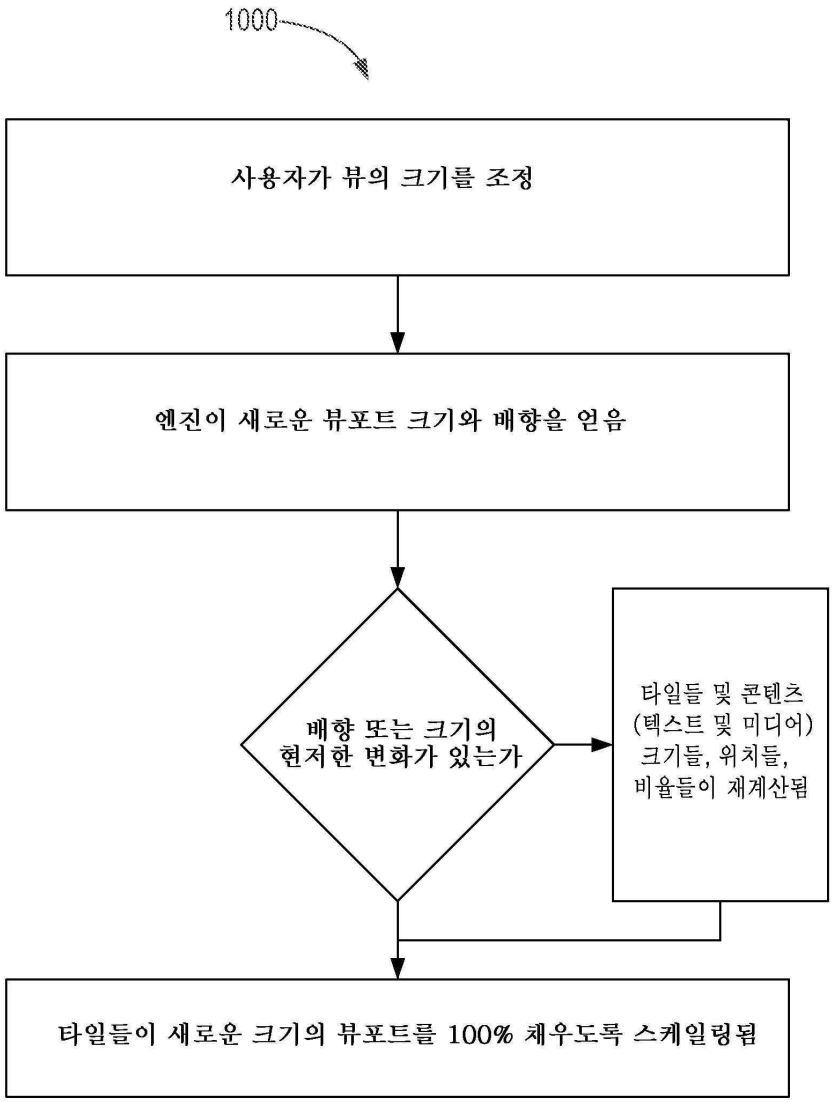
도면8


800
학생 로그인
서버에 가용의 뷰들을 요청
현재 사용자에 대해 저장된 뷰들을 검색
레이아웃 엔진에 뷰의 상세들을 제시하는 메인 GUI를 로딩
모바일인가
데스크톱인가
모바일
데스크톱
대형 디스플레이들 및 HD
그래픽들에 적합하도록
레이아웃이 스케일링됨
레이아웃이 대역폭이
감소되고 모바일
친화적으로 됨
뷰의 각각의 섹션이 그의 연관된 데이터를 로딩

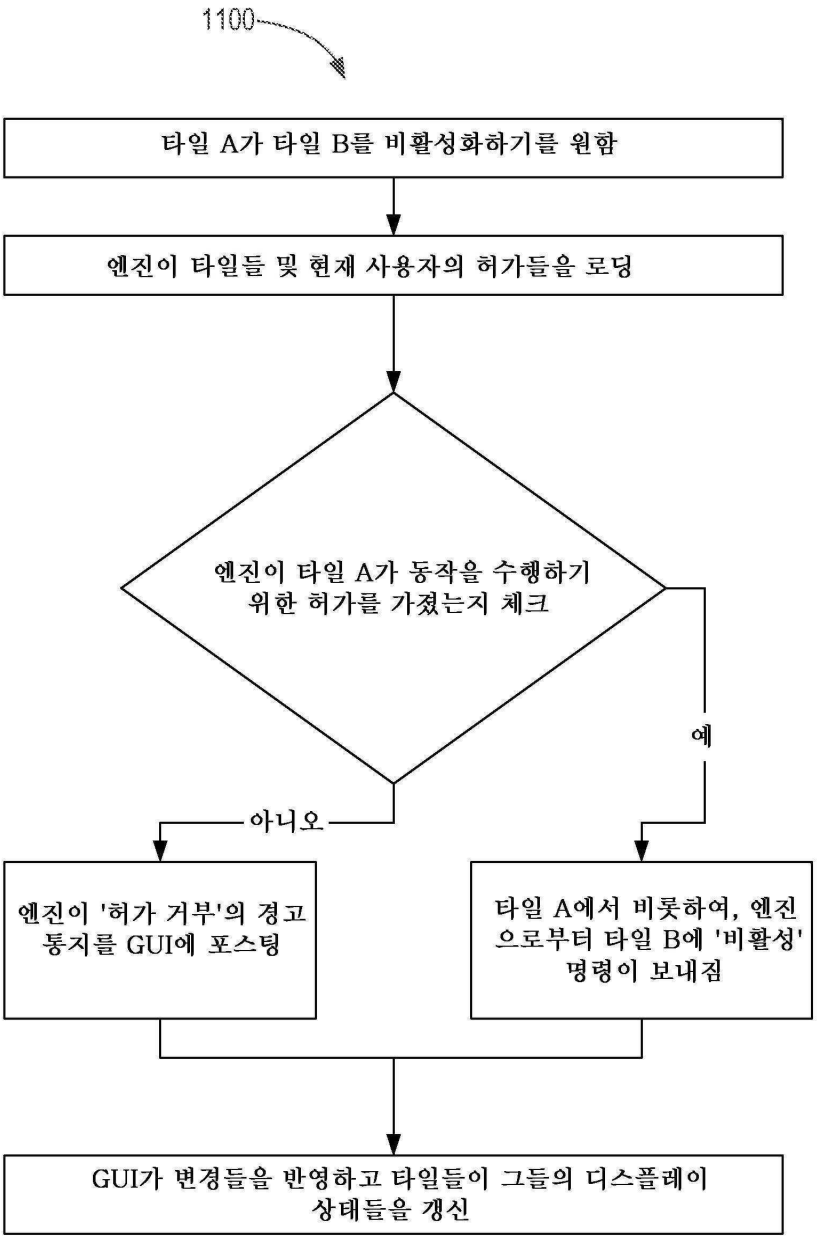
도면9


900
사용자가 뷰로부터 타일이 제거되도록 요청
엔진이 GUI로부터 타일을 페이드아웃
나머지 타일들이 나머지 뷰를 100% 채우도록 최적으로 위치 조정됨

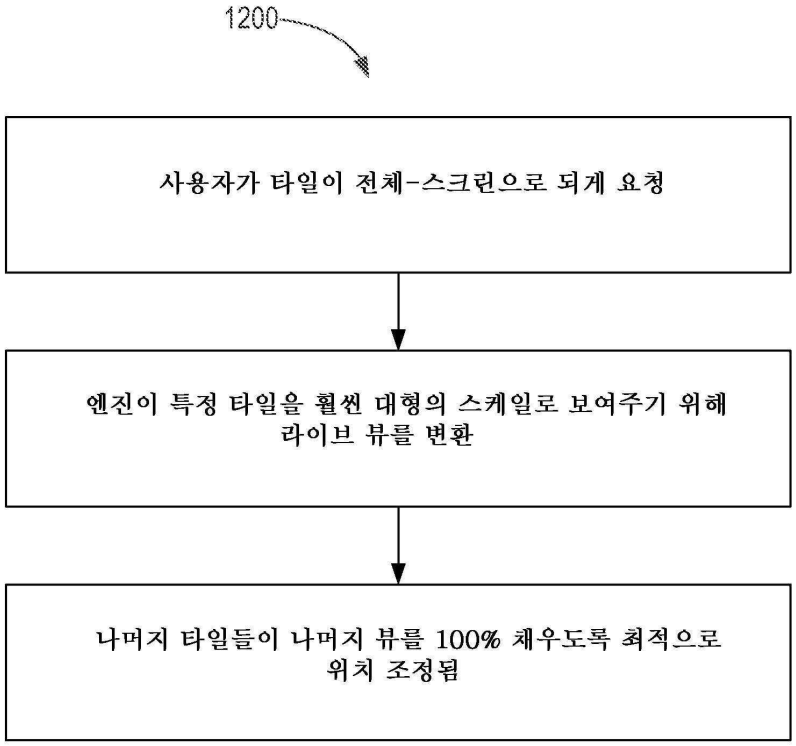
도면10


1000
사용자가 뷰의 크기를 조정
엔진이 새로운 뷰포트 크기와 배향을 얻음
배향 또는 크기의 현저한 변화가 있는가
타일들 및 콘텐츠 (텍스트 및 미디어) 크기들, 위치들, 비율들이 재계산됨
타일들이 새로운 크기의 뷰포트를 100% 채우도록 스케일링됨

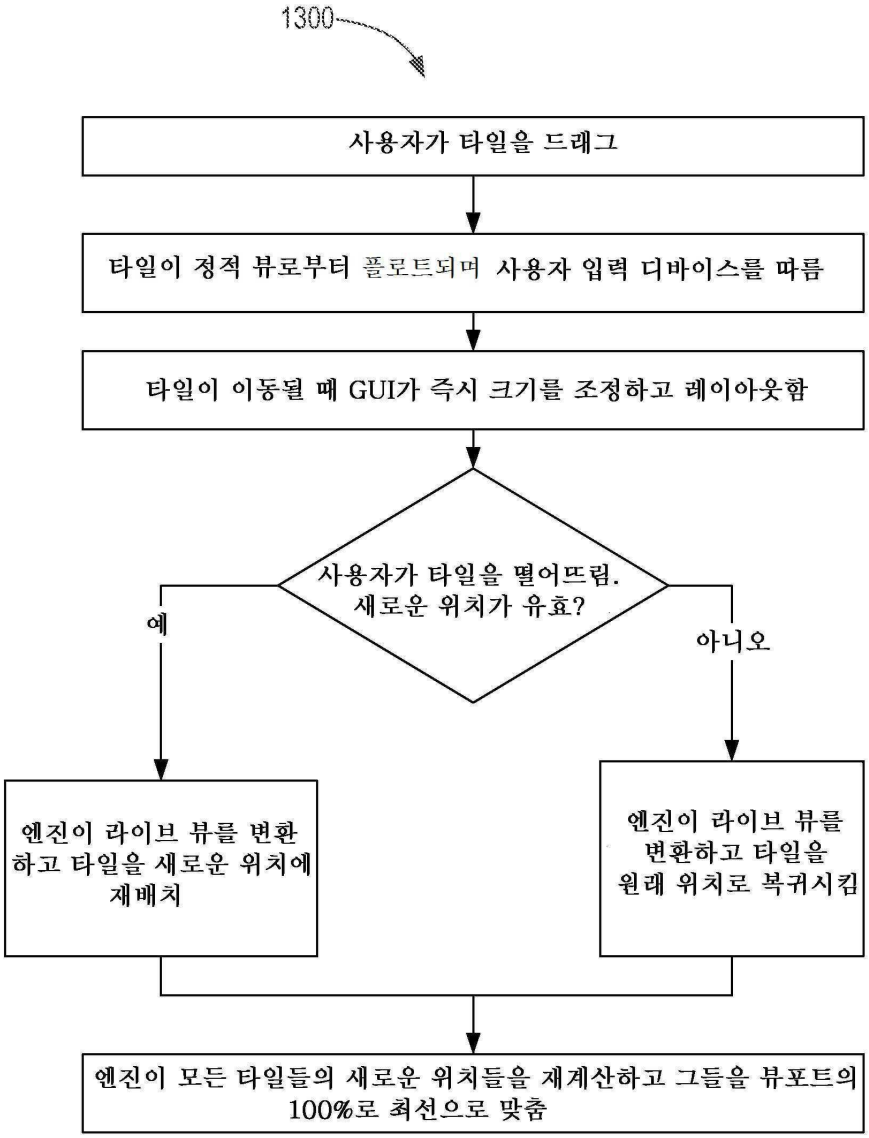
도면11


1100
타일 A가 타일 B를 비활성화하기를 원함
엔진이 타일들 및 현재 사용자의 허가들을 로딩
엔진이 타일 A가 동작을 수행하기 위한 허가를 가졌는지 체크
예
아니오
엔진이 '허가 거부'의 경고 통지를 GUI에 포스팅
타일 A에서 비롯하여, 엔진으로부터 타일 B에 '비활성' 명령이 보내짐
GUI가 변경들을 반영하고 타일들이 그들의 디스플레이 상태들을 갱신

도면12


1200
사용자가 타일이 전체-스크린으로 되게 요청
엔진이 특정 타일을 훨씬 대형의 스케일로 보여주기 위해 라이브 뷰를 변환
나머지 타일들이 나머지 뷰를 100% 채우도록 최적으로 위치 조정됨

도면13


1300
사용자가 타일을 드래그
타일이 정적 뷰로부터 플로트되며 사용자 입력 디바이스를 따름
타일이 이동될 때 GUI가 즉시 크기를 조정하고 레이아웃함
사용자가 타일을 떨어뜨림. 새로운 위치가 유효?
예
아니오
엔진이 라이브 뷰를 변환하고 타일을 새로운 위치에 재배치
엔진이 라이브 뷰를 변환하고 타일을 원래 위치로 복귀시킴
엔진이 모든 타일들의 새로운 위치들을 재계산하고 그들을 뷰포트의 100%로 최선으로 맞춤