(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년12월28일
(11) 등록번호 10-1577841
(24) 등록일자 2015년12월09일

(51) 국제특허분류(Int. Cl.)
*G06F 15/16* (2006.01) *G06F 9/44* (2006.01)
(21) 출원번호 10-2010-7014564
(22) 출원일자(국제) 2008년12월22일
심사청구일자 2013년11월27일
(85) 번역문제출일자 2010년06월30일
(65) 공개번호 10-2010-0111673
(43) 공개일자 2010년10월15일
(86) 국제출원번호 PCT/US2008/087925
(87) 국제공개번호 WO 2009/088727
국제공개일자 2009년07월16일
(30) 우선권주장
11/971,206 2008년01월08일 미국(US)
(56) 선행기술조사문헌
US20030182463 A1
US20060218159 A1

(73) 특허권자
**마이크로소프트 테크놀로지 라이센싱, 엘엘씨**
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
**쿠네오, 앤드류 알.**
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 엘씨에이 국제 특허부 내
**워린, 벤**
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 엘씨에이 국제 특허부 내
**젠즈, 에릭 엠.**
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 엘씨에이 국제 특허부 내
(74) 대리인
**제일특허법인**

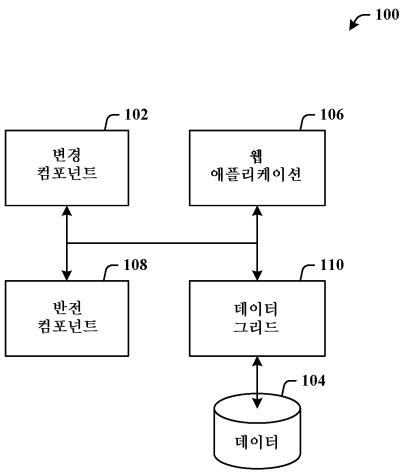
전체 청구항 수 : 총 16 항 심사관 : 권현수

(54) 발명의 명칭 **자바스크립트 그리드에서의 비동기식 멀티-레벨 실행 취소 지원**

## (57) 요 약

본 발명은 그리드 기반 애플리케이션들에서의 클라이언트 상의 멀티-레벨 실행 취소를 위한 아키텍처에 관한 것이다. 이 아키텍처는 변경 추적이 비동기식(및 동기식) 시나리오들에서 심리스하게 작동하는 제어 구동의 캐스케이딩 변경 시스템이다. 클라이언트 애플리케이션이 그리드 개체와 관련되고, 그리드 개체를 인스턴스화하고 구성한다. 애플리케이션은 그리드 내의 데이터에 대한 변경을 시작할 수 있고 및/또는 사용자는 그리드 내의 데이터를 직접 편집할 수 있다. 변경의 결과는 애플리케이션으로의 통지이고, 그 통지는 순서 키를 포함한다. 애플리케이션은 그 통지를 소비하고 그 후 순서 키를 이용하여 업데이트 함수를 호출함으로써 동기식 또는 비동기식 계산들에 기초하여 새로운 변경들을 추가할 수 있다. 애플리케이션은 실행 취소/재실행을 위해 적당히 함께 모아지는 추가적인 업데이트들을 첨부하기 위해 상기 키를 이용한다.

*대 표 도* - 도1



# *명 세 서*

## *청구범위*

### 청구항 1

컴퓨터 구현된 실행 취소(undo) 시스템으로서,

웹 애플리케이션을 통해 데이터에 대한 비동기식 데이터 변경들을 추적하기 위한 변경 컴포넌트 - 상기 비동기식 데이터 변경들은 비동기식 유효성 검사 및 확대들(validations or augmentations)을 필요로 하는 변경이며 데이터에 직접 영향을 미치는 명시적 변경들 및 데이터에 영향을 미치지 않는 암시적 변경을 포함함 - ,

상기 비동기식 데이터 변경들에 대해 멀티-레벨 실행 취소/재실행 반전 동작(multi-level undo/redo reverse operation)들을 수행하기 위한 반전 컴포넌트 - 상기 반전 컴포넌트는 명시적 변경들 및 암시적 변경들을 저장하는 실행 취소 스택(undo stack)과 상기 명시적 변경들에 대한 엔트리들을 저장하는 변경 추적 데이터 구조를 포함하고, 실행 취소 명령은 명시적 변경들에 대해 엔트리를 상기 변경 추적 데이터 구조로부터 제거하고, 명시적 변경이 복귀될 때까지 상기 실행 취소 스택으로부터 변경들을 삭제 및 복귀시킴 - ,

상기 웹 애플리케이션에 저장된 데이터에 대한 테이블 표현(tabular representation)을 제공하기 위한 데이터 그리드 - 상기 테이블 표현은 상기 비동기식 데이터 변경들에 대한 편집 및 시각화를 지원하기 위하여 상기 비동기식 데이터 변경들을 기록하기 위한 셀을 포함하고, 상기 데이터 그리드는 상기 비동기식 데이터 변경들과 연관된 상기 웹 애플리케이션에 통지를 보내고, 상기 통지는 다른 데이터 변경들에 대해 상기 그리드 내에서 상기 비동기식 데이터 변경들을 순서화 가능하게 하는 순서 키(order key)를 포함함 - 및

적어도 상기 변경 컴포넌트 및 상기 반전 컴포넌트와 연관된 컴퓨터 실행가능 명령어들을 실행하는 프로세서

를 포함하는 시스템.

### 청구항 2

제1항에 있어서, 상기 데이터 변경들은 웹페이지와 관련되는 시스템.

### 청구항 3

제1항에 있어서, 상기 웹 애플리케이션은 브라우저 애플리케이션인 시스템.

### 청구항 4

제1항에 있어서, 상기 데이터 그리드는 클라이언트 기반 데이터 그리드(client-based data grid)이고, 상기 비동기식 데이터 변경들은 셀 내의 데이터의 이전 버전 및 이후 버전과, 상기 셀 데이터가 변경되도록 하는 동작을 포함하는 시스템.

### 청구항 5

제4항에 있어서, 상기 비동기식 데이터 변경들은 상기 그리드를 통해 수동으로 또는 프로그램적으로 행해지는 시스템.

### 청구항 6

제1항에 있어서, 상기 반전 동작들은 상기 데이터를 적어도 2개의 이전의 변경들에 따라 존재했던 상태로 복구하는 시스템.

### 청구항 7

제1항에 있어서, 상기 프로세서는 상기 웹 애플리케이션을 통해 상기 통지를 상기 데이터 그리드에 송신하는 시스템.

### 청구항 8

제1항에 있어서, 상기 실행 취소 스택과 상기 변경 추적 데이터 구조는 클라이언트 기반인 시스템.

**청구항 9**

데이터에서의 반전 동작들을 제공하는 컴퓨터 구현된 방법으로서,

클라이언트 기반 그리드를 통해 서버의 웹 기반 애플리케이션의 서버 문서에서 비동기식 데이터 변경들을 검출하는 단계 - 상기 클라이언트 기반 그리드는 상기 비동기식 데이터 변경들을 편집하고 시각화하는 것을 지원하기 위해 상기 비동기식 데이터 변경들을 기록하기 위한 셀을 테이블 표현 내에 포함하며, 상기 비동기식 데이터 변경들은 데이터에 직접 영향을 미치는 명시적 변경과 데이터에 영향을 미치지 않는 암시적 변경을 포함함 - 와,

변경 통지에서 상기 비동기식 데이터 변경들에 순서 키들을 할당하는 단계 - 상기 순서 키들은 상기 비동기식 데이터 변경들의 상대적 순서화가 상기 데이터 그리드에서 이루어지도록 함 - 와,

비동기식 유효성 검사를 위해 상기 변경 통지를 상기 데이터 그리드로부터 상기 웹 기반 애플리케이션으로 송신한 다음 상기 서버로 송신하는 단계와,

상기 변경 통지에 기초하여 상기 서버로부터 비동기식 유효성 검사 정보를 수신하는 단계와,

상기 순서 키들에 따라서 상기 데이터 그리드에서 상기 유효성 검사 정보를 순서화(order)하는 단계와,

상기 순서 키들에 기초하여 상기 서버 문서에서 실행 취소/재실행(undo/redo) 동작들을 관리하는 단계 - 상기 실행 취소 스택은 명시적 변경들 및 암시적 변경들을 저장하고, 변경 추적 데이터 구조는 상기 명시적 변경들에 대한 엔트리들을 저장하고, 실행 취소 명령은 명시적 변경들에 대해 엔트리를 상기 변경 추적 데이터 구조로부터 제거하고, 명시적 변경이 복귀될 때까지 상기 실행 취소 스택으로부터 변경들을 삭제 및 복귀시킴 - 와,

상기 검출하는 단계, 할당하는 단계, 송신하는 단계, 수신하는 단계, 순서화하는 단계 및 관리하는 단계를 적어도 포함하는 동작들을 실행하기 위해 메모리에 저장되어 있는 명령어들을 실행하는 프로세서를 사용하는 단계

를 포함하는 방법.

**청구항 10**

제9항에 있어서, 상기 데이터 그리드 내의 변경들에 기초하여 실행 취소 스택 및 변경 추척 트래커를 통하여 상기 변경들을 캐스케이딩(cascading)하는 단계를 더 포함하는 방법.

**청구항 11**

삭제

**청구항 12**

삭제

**청구항 13**

삭제

**청구항 14**

데이터에서의 반전 동작들을 제공하는 컴퓨터 구현된 방법으로서,

클라이언트 웹 애플리케이션을 통해 서버의 웹 문서에 대한 데이터 변경들을 시작하는 단계 - 상기 데이터 변경들은 데이터에 직접 영향을 미치는 명시적 변경과 데이터에 영향을 미치지 않는 암시적 변경을 포함함 - 와,

상기 데이터 변경들을 상기 데이터 변경들의 순서화된 히스토리(ordered history)로서 클라이언트 실행 취소 스택에 저장하는 단계 - 명시적 변경들에 대해 변경 엔트리들을 별도의 클라이언트 변경 추적 데이터 구조에 저장하고, 상기 실행 취소 스택은 명시적 변경들 및 암시적 변경들을 포함함 - 와,

상기 서버에서 상기 데이터 변경들의 유효성을 검사하는 단계와,

상기 서버로부터 상기 웹 애플리케이션으로 비동기식으로 유효성 검사 정보를 수신하는 단계와,

상기 데이터 변경들의 상기 순서화된 히스토리에 기초하여, 상기 웹 문서에서 실행 취소/재실행 동작들 및 상기 변경 추적 데이터 구조 내의 상기 변경 엔트리들을 관리하는 단계 - 실행 취소 명령은 명시적 변경들에 대해 엔트리를 상기 변경 추적 데이터 구조로부터 제거하고, 명시적 변경이 복귀될 때까지 상기 실행 취소 스택으로부터 변경들을 삭제 및 복귀시킴 - 와,

상기 시작하는 단계, 저장하는 단계, 검사하는 단계, 수신하는 단계, 및 관리하는 단계를 적어도 포함하는 동작들을 실행하기 위해 메모리에 저장되어 있는 명령어들을 실행하는 프로세서를 사용하는 단계

를 포함하는 방법.

**청구항 15**

제14항에 있어서, 클라이언트 데이터 그리드를 통해 상기 데이터 변경들 및 상기 데이터 변경들의 시각화를 익스포즈(expose)하는 단계를 더 포함하는 방법.

**청구항 16**

제14항에 있어서, 상기 웹 애플리케이션에서 수행된 저장 동작 전에 일어난 상기 웹 문서에서의 변경들을 실행취소하는 단계를 더 포함하는 방법.

**청구항 17**

제14항에 있어서, 저장 동작에 응답하여, 상기 변경 추적 데이터 구조로부터 변경들을 판독하고, 상기 변경들을 데이터 소스에 커밋(commit)하고, 상기 변경 추적 데이터 구조를 클리어하는 단계를 더 포함하는 방법.

**청구항 18**

제14항에 있어서, 상기 데이터 변경들의 상기 순서화된 히스토리는 각 변경에 할당된 변경 키에 기초하는 방법.

**청구항 19**

삭제

**청구항 20**

제14항에 있어서, 변경들을 셀-레벨 변경들의 컬렉션으로서 익스포트(export)하는 단계를 더 포함하는 방법.

## *발명의 설명*

### *배 경 기 술*

[0001] 데스크톱 클라이언트 애플리케이션들과 "씬(thin)" 웹 애플리케이션들을 주로 구별짓는 것들 중 하나는 편집 경험의 풍부함이다. 전통적으로, 웹 애플리케이션들은, 사용자가 애플리케이션을 탐색할 때 서버에 데이터를 커밋하는, 개별 포스트-백들(post-backs)에 따라 서버와 데이터를 트랜잭트(transact)한다. 일단 사용자가 웹페이지로부터 떠나 탐색하면 사용자 동작을 실행 취소하는 것은 종종 불가능하다. 반대로, 사용자는 클라이언트 애플리케이션과 더 유동적으로 상호 작용할 수 있고, 사용자가 데이터를 저장할 준비가 되어 있을 때만 그렇게 한다. 또한, 만일 사용자가 클라이언트 애플리케이션에서 작업하는 동안 편집 실수를 한다면, 사용자는 저장된 파일에 어떠한 영향도 없이 그 변경들을 복귀시키기(revert) 위해 한 번 이상 "실행 취소(undo)"를 선택할 수 있다.

[0002] 다수의 소프트웨어 기술들은 현재 고용인들이 웹-유형 애플리케이션들을 통하여 기업 서버들 상의 데이터와 상호 작용할 것을 요구한다. 웹페이지를 통해, 예를 들면, 프로젝트 서버들에서 일반적으로 발견되는 구조적 태스크 데이터(structured task data)의 편집(예를 들면, 태스크들의 추가/삭제, 리소스들의 할당, 스케줄링된 데이터의 변경 등)을 생각해보자. 멀티-레벨 실행 취소 능력이 없다면, 이 경험은 사용자들이 동작들을 실행 취소하도록 허용되지 않기 때문에 위험할 수 있다. 사용자는 빈번한 저장을 수행할 수 있지만, 그러한 데이터세트들을 서버에 저장하는 것은 느린 프로세스이다. 따라서, 효과적인 편집 성능은 복잡한 데이터를 웹페이지에 입력하는 사용자의 편안함과 비례하여 감소한다.

[0003] [발명의 개요]

[0004] 다음은 여기에 설명되는 일부 새로운 실시예들에 대한 기초적인 이해를 제공하기 위하여 간략화한 개요를 제시한다. 이 개요는 광범위한 개관이 아니며, 중요한/결정적인 엘리먼트들을 식별하거나 그것의 범위를 서술하기 위해 의도된 것이 아니다. 그것의 유일한 목적은 뒤에 제시되는 보다 상세한 설명의 서문으로서 일부 개념들을 간략화한 형태로 제시하는 것이다.

[0005] 개시된 멀티-레벨 실행 취소 아키텍처는 변경 추적이 비동기식(및 동기식) 시나리오들에서 심리스하게 작동하는 제어 구동의 캐스케이딩 변경 시스템(control driven cascading changes system)이다. 또한, 실행 취소는 저장 동작들을 초월(transcend)하고, 암시적 및 명시적 변경들 사이의 차이를 이해하고, 그에 따라서 이들 변경들을 처리하여 실행 취소를 위한 컨텍스트를 제공한다.

[0006] 클라이언트 애플리케이션은 애플리케이션의 데이터를 표의 표현(tabular representation)으로서 제시하는 그리드 개체(grid object)와 관련되고, 그에 의해 편집 및 시각화를 지원한다. 애플리케이션은, 부분적으로, 그리드 개체를 인스턴스화하고 구성한다. 애플리케이션은 그리드 내의 데이터에 대한 변경을 프로그램적으로 시작할 수 있고 및/또는 사용자는 그리드 내의 데이터를 직접 편집할 수 있다. 변경의 결과는 애플리케이션으로의 통지이고, 그 통지는 순서 키(order key)를 포함한다. 애플리케이션은 그 통지를 소비하고 그 후 상기 순서 키를 이용하여 업데이트 함수를 호출함으로써 동기식 또는 비동기식 계산들(예를 들면, 스케줄링)에 기초하여 새로운 변경들을 추가할 수 있다. 애플리케이션은 장래의 임의의 시점에서 상기 키를 이용하여 추가적인 업데이트들을 상기 변경에 첨부할 수 있다. 업데이트들은 실행 취소/재실행을 위해 적당히 함께 모아진다.

[0007] 전술한 및 관련 목적들을 달성하기 위하여, 다음의 설명 및 첨부 도면들과 관련하여 특정한 예시적인 양태들이 여기에 설명된다. 그러나, 이들 양태들은 여기에 개시된 원리들이 채용될 수 있는 다양한 방식들 중 소수만을 나타내고 모든 그러한 양태들 및 등가물들을 포함하도록 의도되어 있다. 다른 이점들 및 새로운 특징들은 도면들과 함께 고려될 때 다음의 상세한 설명으로부터 명백해질 것이다.

## *도면의 간단한 설명*

[0008] 도 1은 컴퓨터 구현된 실행 취소 시스템을 예시한다.

도 2는 멀티-레벨 데이터 변경 반전 동작들을 제공하는 클라이언트-서버 시스템의 구현을 예시한다.

도 3은 데이터에 대한 비동기식 실행 취소/재실행 동작들의 예를 예시한다.

도 4는 비동기식 비순차적 변경 처리(asynchronous out-of-order change processing)를 다루기 위한 변경 키들의 이용에 기초한 최종 날짜 업데이트들을 예시한다.

도 5는 변경 컴포넌트의 일부로서 변경 트래커(change tracker)로서, 반전 컴포넌트의 일부로서 실행 취소 스택(undo stack)을 예시한다.

도 6은 그리드 내의 데이터의 최초 그리드 상태, 실행 취소 스택의 스택 상태, 및 변경 트래커의 트래커 상태를 예시한다.

도 7은 암시적 변경, 및 스택 상태 및 트래커 상태에 대한 영향들을 예시한다.

도 8은 그리드 데이터의 태스크 이름의 변경, 및 실행 취소 스택 및 변경 트래커에 대한 영향들을 예시한다.

도 9는 실행 취소 동작의 실행을 예시한다.

도 10은 데이터 내의 반전 동작들을 제공하는 컴퓨터 구현된 방법을 예시한다.

도 11은 명시적/암시적 변경들을 처리하는 방법을 예시한다.

도 12는 데이터 내의 반전 동작들을 제공하는 컴퓨터 구현된 방법을 예시한다.

도 13은 개시된 아키텍처에 따라 멀티-레벨 실행 취소를 실행하도록 동작 가능한 컴퓨팅 시스템의 블록도를 예시한다.

도 14는 멀티-레벨 실행 취소 처리를 위한 예시적인 클라이언트-서버 컴퓨터 환경의 개략적인 블록도를 예시한다.

## *발명을 실시하기 위한 구체적인 내용*

[0009] 개시된 아키텍처는 그리드 기반 애플리케이션들에서 클라이언트 상에서 멀티-레벨 실행 취소를 지원함으로써 "씬" 웹 애플리케이션들의 편집 경험에서의 주요 결함을 메운다. 이 특징은 다수의 온라인 편집 경험들에도 유용하다. 예를 들면, 멀티-레벨 실행 취소는 사용자가 한 번에 더 많은 데이터를 편집하는 것에 관하여 더 기분 좋게 느끼게 하고, 애플리케이션의 긍정적인 성능에 대한 사용자 인식을 증가시킨다. 또한, 애플리케이션들의 온라인 버전들도 사용자가 편집에 관한 완전한 제어를 갖도록 레코드 내의 다수의 변경들이 실행 취소되고 재실행될 수 있도록 이익을 얻을 수 있다.

[0010] 이제 도면들이 참조되고, 도면들에서 같은 참조 번호들은 전체에 걸쳐서 같은 엘리먼트들을 나타내기 위해 이용된다. 다음의 설명에서는, 설명을 목적으로, 그것의 철저한 이해를 제공하기 위하여 다수의 특정한 상세들이 제시된다. 그러나, 새로운 실시예들은 이들 특정한 상세들 없이도 실시될 수 있다는 것은 명백할 것이다. 다른 경우들에서, 잘 알려진 구조들 및 장치들은 그것의 설명을 용이하게 하기 위하여 블록도 형태로 도시된다.

[0011] 도 1은 컴퓨터 구현된 실행 취소 시스템(100)을 예시한다. 시스템(100)은 웹 애플리케이션(106)을 통해 데이터(104)에 대한 비동기식 변경들을 추적하기 위한 변경 컴포넌트(102)를 포함한다. 시스템(100)은 데이터(104)에 대한 데이터 변경들에 대한 이전의 상태들로의 반전 동작들(예를 들면, 실행 취소, 재실행)을 수행하기 위한 반전 컴포넌트(108)를 더 포함한다. 데이터(104)에 대한 변경들은 데이터 그리드(110)를 통해 행해질 수 있다. 데이터(104)에 대한 변경들은 그리드(110)를 통해 수동으로 또는 프로그램적으로 행해질 수 있다.

[0012] 애플리케이션(106)은 그리드(110)를 인스턴스화하고 구성한다. 그리드(110)는 편집 및 시각화를 지원하는, 애플리케이션(106)에 저장되어 있는 데이터의 표의 표현이다. 변경은 셀의 데이터의 이전 및 이후 버전들에, 셀 값이 변하게 하기 위해 요구된 동작을 더한 것이다.

[0013] 일 실시예에서, 웹 애플리케이션(106)은 그를 통해 서버에 의해 호스팅되는 웹 페이지에 대한 변경들이 행해지는 브라우저 애플리케이션이다. 사용자는 웹페이지 문서에 행해진 변경들에 대한 멀티-레벨 실행 취소/재실행 반전 동작들을 수행할 수 있다. 다른 구현에서, 웹 애플리케이션(106)은 사용자가 네트워크 데이터보다는 로컬 데이터와 상호 작용하고, 멀티-레벨 실행 취소/재실행과 같은 반전 동작들을 수행하게 하는 브라우저이다.

[0014] 도 2는 멀티-레벨 데이터 변경 반전 동작들을 제공하는 클라이언트-서버 시스템(200)의 구현을 예시한다. 시스템(200)은 도 1의 시스템(100), 즉, 웹 애플리케이션(106)을 통해 데이터(104)에 대한 비동기식 변경들을 추적하기 위한 변경 컴포넌트(102), 및 데이터 변경들에 대한 이전의 상태들로의 반전 동작들(예를 들면, 실행 취소, 재실행)을 수행하기 위한 반전 컴포넌트(108)를 나타낸다. 데이터 변경들은 데이터 그리드(110)를 통해 수동으로 및/또는 프로그램적으로 행해진다. 여기서, 데이터 변경들은 웹 서버(202)의 웹 문서(예를 들면, 웹 페이지)(200)에 적용된다. 편집들이 그리드(110)에 전달될 때, 비동기식 유효성 검사(asynchronous validation)를 위해 애플리케이션(106)을 통해 서버(202)에 변경 통지들이 보내진다. 일단 유효성 검사되면, 변경들은 데이터(104)에 대한 업데이트들로서 애플리케이션(106)을 통해 그리드(110)에 다시 보내진다.

[0015] 도 3은 데이터에 대한 비동기식 실행 취소/재실행 동작들의 예(300)를 예시한다. 비동기식 변경은 임의의 수의 비동기식 유효성 검사들 또는 확대들(augmentations)을 필요로 하는 변경이다. 엔티티가 그리드 데이터에 대한 변경들을 행할 때, 이 변경의 결과는 애플리케이션(106)에의 통지이다. 이 통지 안에는 순서(또는 변경) 키가 포함된다. 애플리케이션(106)은 통지를 소비하고 그 후 그 순서 키를 이용하여 업데이트 함수를 호출함으로써 일부 동기식 또는 비동기식 계산들(예를 들면, 스케줄링)에 기초하여 새로운 변경들을 추가할 수 있다. 애플리케이션(106)은 장래의 임의의 시점에 순서 키를 자유로이 이용하여 추가적인 업데이트들을 이 변경에 첨부할 수 있다. 이들 업데이트들은 실행 취소/재실행을 위해 적당히 함께 모아진다.

[0016] 여기서, 비동기식 유효성 검사를 필요로 하는, A 및 B로 표시된, 2개의 변경들이 그리드(110)에 입력된다. 편집들이 행해질 때, 변경들은 그리드(110)에 의해 순서대로 캡처된다. 각 변경에는 순서 키가 부가(tag)된다. 예를 들면, 변경 A에는 (변경 키 A로도 표시된) 순서 키 A가 부가되고 (변경 A보다 시간적으로 나중인) 제2 변경 B는 변경 키 B로 표시된다. 데이터 변경이 행해질 때, 그리드(110)는 이것을 검출하고 애플리케이션(106)(예를 들면, 브라우저)에 통지를 보내고, 애플리케이션(106)은 그 후 통지를 유효성 검사를 위해 서버(202)에 보낸다. 서버(202)에서의 유효성 검사 프로세스는 비순차적으로(out-of-order) 일어날 수 있고, 또는 일단 유효성 검사가 완료되면, 서버는 이전에 정돈된 유효성 검사(previously ordered validation)를 비순차적으로 보낼 수 있다. 따라서, 순서 키들은 그리드(110)에서 변경들을 정돈하는 것을 용이하게 한다.

[0017] 이 예에서는, 제2 행(row)에서 지속 기간(Duration)이라는 라벨이 붙은 필드를 갖는 열(column)에 대해 행해진 데이터 변경 A, 및 5일로 설정된 지속 기간의 새로운 값에 대하여 설명하는 제1 변경 통지 요청(302)이 그리드

(110)로부터 애플리케이션(106)으로 보내진다. 후속의 데이터 편집에서, 그리드(110)는 제3 행에서 지속 기간이라는 라벨이 붙은 필드를 갖는 열에 대해 행해진 데이터 변경 B, 및 7일로 설정된 지속 기간의 새로운 값에 대하여 설명하는 제2 변경 통지 요청(304)을 애플리케이션(106)으로 보낸다. 그 변경 통지 요청들(302 및 304)은 순서대로(예를 들면, 변경 B 전에 변경 A) 애플리케이션(106)으로 보내진다. 애플리케이션(106)은 그 후 그 통지 요청들(302 및 304)을 서버(202)에 전송하고, 서버(202)는 그 변경 요청들(302 및 304)을 비동기식으로 유효성 검사하고, 그 유효성 검사들을 다시 애플리케이션(106)에 반환한다.

[0018] 여기서, 서버(202)는 제1 변경 통지 요청(302)에 대한 제1 유효성 검사 프로세스(306)를 시작한다. 다음으로, 서버(202)는 제2 변경 통지 요청(304)을 수신하고 그에 대한 제2 유효성 검사 프로세스(308)를 시작한다. 서버(202)는 제1 유효성 검사 프로세스(306) 전에 제2 유효성 검사 프로세스(308)를 완료한다. 따라서, 관련 데이터를 업데이트하기 위해 제2 업데이트 응답(310)이 서버(202)로부터 애플리케이션(106)을 통하여 그리드(110)에 보내진다. 제2 업데이트 응답(310)은 그 데이터 변경이 제1 데이터 변경 A에 관하여 행해진 순서를 나타내는 순서 키 B를 포함한다. 제2 업데이트 응답(310)은 또한 제3 행에서, 종료 날짜(End Date)라는 라벨이 붙은 필드에서 변경이 행해졌고, 종료 날짜에 대한 새로운 값은 5/27인 것을 포함한다. 이것은 제2 변경 통지 요청(304)에서 7일의 새로운 값에 대응한다.

[0019] 서버(202)는 그 후 제1 유효성 검사 프로세스(306)를 완료하고 관련 데이터를 업데이트하기 위해 제1 업데이트 응답(312)을 애플리케이션(106)을 통하여 그리드(110)로 보낸다. 제1 업데이트 응답(312)은 그 데이터 변경이 제2 데이터 변경 B에 관하여 행해진 순서를 나타내는 순서 키 A를 포함한다. 제1 업데이트 응답(312)은 또한 제2 행에서, 종료 날짜(End Date)라는 라벨이 붙은 필드에서 변경이 행해졌고, 종료 날짜에 대한 새로운 값은 5/25인 것을 포함한다. 이것은 제1 변경 통지 요청(302)에서 5일의 새로운 값에 대응한다. 따라서, 그 변경들은 비순차적으로(변경 A 전에 변경 B) 유효성 검사로부터 되돌아온다.

[0020] 도 4는 비동기식 비순차적 변경 처리(asynchronous out-of-order change processing)를 다루기 위한 변경 키들의 이용에 기초한 최종 날짜 업데이트들을 예시한다. 최종 저장 동작 이후에 일어난 데이터 변경들에 대한 요청들의 결과들(400)이 도시되어 있다. 서버(202)로부터 반환될 때, 결과들은 그 이벤트들이 실제로 일어난 순서(변경 B 후에 변경 A)보다는, 변경(또는 순서) 키들이 지시하는 논리적 순서(변경 A 후에 변경 B)로 배치되고 저장된다. 따라서, 변경 A에 관련된 변경(402)은 함께 저장되고 변경 B 전에 일어나는 것으로 표시된다. 유사하게, 변경 B에 관련된 변경(404)은 함께 저장되고 변경 A 후에 일어나는 것으로 표시된다.

[0021] 실행 취소 및 재실행 반전 동작들도 이런 식으로 처리된다. 따라서, 사용자가 실행 취소를 선택한다면, 변경 B와 관련된 양쪽 변경들(지속 기간 및 종료 날짜)은, 비록 변경 A에 대한 업데이트가 그리드가 마주친 최종 이벤트였다 할지라도, 변경 키 B에 의해 나타내어진 바와 같이, 그것이 사용자가 행한 최종 변경이므로, 실행 취소될 것이다.

[0022] 엔티티(예를 들면, 사용자 또는 시스템)는 명시적으로 또는 암시적으로 그리드 내의 데이터에 대한 변경을 행하기 위해 그리드와 상호 작용한다. 명시적 변경은 데이터에 직접 영향을 미치는(예를 들면, 태스크에 대한 시작 날짜를 변경하는) 그리드에 대해 행해진 변경이다. 암시적 변경은 데이터에 영향을 마치지 않는(예를 들면, 열의 크기를 조정하는) 그리드에 행해지는 변경이다.

[0023] 도 5는 변경 컴포넌트(102)의 일부로서 변경 트래커(change tracker)(500) 및 반전 컴포넌트(108)의 일부로서 실행 취소 스택(undo stack)(502)을 예시한다. 엔티티가 그리드 데이터에 대한 변경들을 행할 때, 각 변경의 유형이 지시되고, 실행 및 실행 취소하는 함수들이 실행 취소 스택(502) 상에 배치된다. 실행 취소 스택(502)은 동작들의 최종 세트의 정돈된 히스토리를 저장하는 데이터 구조이다. 실행 취소 이벤트가 일어날 때, 실행 취소 스택(502)으로부터 동작들이 제거되고 최초 명시적 변경과 마주칠 때까지 복귀된다. 최초 명시적 변경은 복귀된 최종 동작이다. 이 시점에서, 실행 취소 동작은 종료한다. 후속의 실행 취소 동작들은 동일한 방식으로 처리되고; 실행 취소 명령은 다음 명시적 동작 전에 모든 암시적 동작들을 복귀시킬 것이고, 그 후 명시적 동작을 복귀시킬 것이다.

[0024] 이 시스템의 효과는 실행 취소 동작 후에, 그리드의 시각적 상태는 실행 취소 동작 전으로 복구된다는 것이다. 모든 명시적 변경들은 실행 취소 스택(502)으로부터 별개의 구조(변경 트래커(500))에 저장된다. 이 변경 트래커(500)는 셀-레벨 변경들의 컬렉션으로서 익스포트(export)될 수 있다.

[0025] 실행 취소 스택(502) 및 변경 트래커(500)는 독립된 구조들이다. 변경이 일어날 때, 그 변경은 실행 취소 스택(502) 상에 푸시(push)되고, 그 변경이 명시적이라면 그 변경에 대한 항목이 변경 트래커(500)에 만들어진다.

만일 변경이 암시적이라면, 그 변경은 변경 트래커(500)에 로그(log)되지 않는다.

[0026] 실행 취소 이벤트가 일어날 때, 변경은 실행 취소 스택(502) 및 변경 트래커(500)로부터 제거되고, 새로운 값이 변경 트래커(500)에 기록된다. 만일 변경 트래커(500)에 변경이 존재한다면, 그 변경은 제거된다. 만일 변경 트래커(500)에 변경이 존재하지 않는다면, 변경을 복귀시키는 동작(그 자체가 변경임)이 변경 로그에 추가된다. 저장 동작이 일어날 때 변경들은 변경 트래커(500)로부터 판독되고, 데이터소스에 커밋되고, 변경 트래커(500)는 클리어된다.

[0027] 이 시스템의 효과는 변경을 실행 취소하기 위해 필요한 정보가 실행 취소 스택(502)에 저장되기 때문에 엔티티가 저장 이벤트 전에 일어난 동작들을 실행 취소할 수 있다는 것이다. 유사하게, 변경 트래커(500)는 실행 취소 스택(502)의 부재 시에 기능할 수 있다.

[0028] 도 6-9는 그리드 데이터의 상태, 실행 취소 스택, 및 변경 트래커에서 일어나는 변경들을 예시하는 일련의 다이어그램들을 예시한다. 도 6은 그리드 내의 데이터의 최초 그리드 상태(600), 실행 취소 스택의 스택 상태(602), 및 변경 트래커의 트래커 상태(604)를 나타낸다. 변경들은 그리드 데이터의 행 2 및 3에서 일어날 것이다.

[0029] 엔티티는 울타리를 페인트칠하는 것(painting the fence)이 1일 대신에 2일이 걸릴 것이라는 것을 결정함으로써 명시적 변경들을 행한다. 변경 그리드 상태(606)는 지속 기간(Duration) 열 내의 제2 행에서 2일로의 변경을 포함한다. 예를 들면, 프로젝트 관리 애플리케이션의 일부로서 프로젝트의 경우에, 지속 기간의 변경은 완성(Finish)이라 불리는 종료 날짜의 변경을 초래한다(캐스케이딩 변경). 양쪽 변경들을 포함하는 캐스케이딩 트랜잭션이 생성된다. 따라서, 완성 열 내의 제2 행의 필드 정보는, 변경된 그리드 상태(606)에서 나타내어진 바와 같이, 월요일(Mon) 5/21/xx로부터 화요일(Tue) 5/22/xx로 변화한다.

[0030] 이들 지속 기간 및 완성 변경들은 스택 상태(608)에서 나타내어진 바와 같이 실행 취소 스택 상에 푸시된다. 또한, 트래커 상태(610)는 그 변경들을 반영한다. 도 7은 암시적 변경, 및 스택 상태 및 트래커 상태에 대한 영향들을 예시한다. 여기서, 엔티티는 완성 열을 감추고 그에 의해 그리드 상태(612)로 변경하고, 이는 변경을 스택 상태(614)로 변화하는 실행 취소 스택 상에 푸시하지만, 변경 트래커의 트래커 상태(610)는 변경하지 않는다. 그리드의 데이터는 이제 서버 데이터와 일관되기 때문에, 엔티티는 프로젝트를 저장하고, 이는 변경 트래커 내의 트래커 상태(610)가 스택 상태(616)로 클리어되게 한다.

[0031] 도 8은 그리드 데이터의 태스크 이름의 변경, 및 실행 취소 스택 및 변경 트래커에 대한 영향들을 예시한다. 엔티티는 행 3 내의 태스크 이름의 'Clean-up tools'로부터 'Clean-up project'로의 변경에 기초하여 그리드 상태(618)를 변경한다. 스택 상태(620)는 스택 상에 푸시된 변경을 반영하고, 트래커는 트래커 상태(622)를 반영한다.

[0032] 도 9는 실행 취소 동작의 실행을 예시한다. 실행 취소는 스택 상태(622) 및 트래커 상태(626)(전으로부터 어떠한 변경도 없음)에 나타내어진 바와 같이, 실행 취소 스택 및 변경 트래커로부터 변경을 팝(pop)하는 것에 의해 최종 명시적 변경을 복귀시킨다. 그 후 엔티티는 실행 취소를 한 번 더 실행하고, 이는 암시적 변경 및 남아 있는 명시적 변경 양쪽 모두를 복귀시킨다. 이 캐스케이딩 효과는 실행 취소 스택으로부터 변경을 제거하여 스택 상태(628)로 만들고, 변경 트래커에 반전 변경을 추가하여 트래커 상태(630)로 만들고, 그리드 상태를 도 6의 그리드 상태(600)와 동일한 그리드 상태(632)로 다시 변경한다.

[0033] 다음은 개시된 아키텍처의 새로운 양태들을 수행하기 위한 예시적인 방법들을 나타내는 일련의 순서도들이다. 설명의 간소화를 위하여, 예를 들면, 순서도 또는 흐름도의 형태로, 여기에 나타내어진 하나 이상의 방법들은 일련의 단계들로서 도시되고 설명되지만, 그 방법들은, 어떤 단계들은, 그에 따라서, 여기에 도시되고 설명된 것과는 다른 순서로 및/또는 다른 단계들과 동시에 일어날 수 있으므로, 단계들의 순서에 의해 제한되지 않는다는 것을 이해하고 인식해야 한다. 예를 들면, 숙련된 당업자들은 방법이 대안적으로, 상태도에서와 같이, 일련의 상호 관련된 상태들 또는 이벤트들로서 표현될 수 있다는 것을 이해하고 인식할 것이다. 또한, 방법에서 예시된 단계들이 새로운 구현을 위해 모두 다 필요하지 않을 수도 있다.

[0034] 도 10은 데이터 내의 반전 동작들을 제공하는 컴퓨터 구현된 방법을 예시한다. 1000에서는, 서버의 서버 문서 내의 데이터 변경들이 클라이언트 기반 그리드를 통해 검출된다. 1002에서는, 변경 통지에서 변경들에 순서 키들이 할당된다. 1004에서는, 변경 통지들이 유효성 검사를 위해 서버에 보내진다. 1006에서는, 통지들에 기초하여 서버로부터 비동기식 유효성 검사 정보가 수신된다. 1008에서는, 유효성 검사 정보가 순서 키들에 따라서 그리드 내에서 정돈된다. 1010에서는, 서버 문서 내의 실행 취소/재실행 동작들이 순서 키들에 기초하여 관리

된다.

[0035] 도 11은 명시적/암시적 변경들을 처리하는 방법을 예시한다. 1100에서는, 데이터의 변경이 그리드를 통해 수신된다. 1102에서는, 변경의 유형 및 변경을 실행 취소/재실행하는 함수가 실행 취소 스택에 저장된다. 1104에서는, 반전 동작(예를 들면, 재실행, 실행 취소)이 수신된다. 1106에서는, 실행 취소 동작들이 스택으로부터 제거되고 (복귀되는 최종 동작인) 최초 명시적 변경과 마주칠 때까지 복귀된다.

[0036] 도 12는 데이터 내의 반전 동작들을 제공하는 컴퓨터 구현된 방법을 예시한다. 1200에서는, 클라이언트 웹 애플리케이션을 통해 서버의 웹 문서에 대해 데이터 변경들이 시작된다. 1202에서는, 그 변경들이 클라이언트 실행 취소 스택 내에 변경들의 정돈된 히스토리로서, 클라이언트 변경 트래커 내에 변경 항목들로서 저장된다. 1204에서는, 변경들이 서버에서 유효성 검사된다. 1206에서는, 웹 애플리케이션에서 서버로부터 비동기식으로 유효성 검사 정보가 수신된다. 1208에서는, 웹 문서 내의 실행 취소/재실행 동작들이 실행 취소 스택 내의 변경들의 정돈된 히스토리 및 변경 트래커 내의 변경 항목들에 기초하여 클라이언트에서 관리된다.

[0037] 이 출원에서 이용될 때, "컴포넌트" 및 "시스템"이라는 용어들은 하드웨어이든, 하드웨어와 소프트웨어의 조합이든, 소프트웨어이든, 또는 실행 중인 소프트웨어이든 간에, 컴퓨터 관련 엔티티를 나타내도록 의도되어 있다. 예를 들면, 컴포넌트는 프로세스 상에서 실행하는 프로세스, 프로세서, 하드 디스크 드라이브, (광 및/또는 자기 저장 매체의) 다중 저장 드라이브들, 개체, 실행 파일, 실행의 스레드, 프로그램, 및/또는 컴퓨터일 수 있지만, 이들에 제한되는 것은 아니다. 예시로서, 서버 상에서 실행하는 애플리케이션 및 서버는 양쪽 모두 컴포넌트일 수 있다. 하나 이상의 컴포넌트들이 프로세스 및/또는 실행의 스레드 내에 존재할 수 있고, 컴포넌트는 하나의 컴퓨터 상에 국지화될 수 있고 및/또는 2개 이상의 컴퓨터들 사이에 분산될 수 있다.

[0038] 이제 도 13을 참조하면, 개시된 아키텍처에 따라 멀티-레벨 실행 취소를 실행하도록 동작 가능한 컴퓨팅 시스템(1300)의 블록도가 예시되어 있다. 그것의 다양한 양태들에 대한 추가적인 컨텍스트를 제공하기 위하여, 도 13 및 다음의 논의는 다양한 양태들이 구현될 수 있는 적합한 컴퓨팅 시스템(1300)에 대한 간단한 일반적인 설명을 제공하도록 의도되어 있다. 상기 설명은 하나 이상의 컴퓨터들에서 실행할 수 있는 컴퓨터 실행가능 명령어들의 일반적인 컨텍스트에 있지만, 숙련된 당업자들은 다른 프로그램 모듈들과 협력하여 및/또는 하드웨어와 소프트웨어의 조합으로서 새로운 실시예가 또한 구현될 수 있다는 것을 인지할 것이다.

[0039] 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조 등을 포함한다. 또한, 숙련된 당업자들은 본 발명의 방법들은, 각각이 하나 이상의 관련 장치들에 동작 가능하게 연결될 수 있는, 단일 프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론, 퍼스널 컴퓨터, 핸드헬드 컴퓨팅 장치, 마이크로프로세서 기반 또는 프로그램가능한 가전 제품 등을 포함하는 다른 컴퓨터 시스템 구성들과 함께 실시될 수 있다는 것을 알 것이다.

[0040] 예시된 양태들은 또한 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 특정한 태스크가 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘다에 위치할 수 있다.

[0041] 컴퓨터는 통상적으로 각종 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 휘발성 및 비휘발성 매체, 이동식 및 비이동식 매체를 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함하지만 이에 제한되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 이용될 수 있는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다.

[0042] 다시 도 13을 참조하면, 다양한 양태들을 구현하기 위한 예시적인 컴퓨팅 시스템(1300)은 처리 장치(1304), 시스템 메모리(1306) 및 시스템 버스(1308)를 갖는 컴퓨터(1302)를 포함한다. 시스템 버스(1308)는 시스템 메모리(1306)를 포함하지만, 이에 제한되지 않는, 시스템 컴포넌트들을 처리 장치(1304)에 연결하기 위한 인터페이스를 제공한다. 처리 장치(1304)는 각종 상업적으로 이용 가능한 프로세서들 중 임의의 것일 수 있다. 듀얼 마이크로프로세서 또는 기타 멀티-프로세서 아키텍처가 처리 장치(1304)로서 채용될 수도 있다.

[0043] 시스템 버스(1308)는 메모리 버스(메모리 컨트롤러와 함께 또는 메모리 컨트롤러 없이), 주변 장치 버스 및 각

종 상업적으로 이용 가능한 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스에 더 상호 접속할 수 있는 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 시스템 메모리(1306)는 비휘발성 메모리(NON-VOL)(1310) 및/또는 휘발성 메모리(1312)(예를 들면, RAM(random access memory))를 포함할 수 있다. 비휘발성 메모리(1310)(예를 들면, ROM, EPROM, EEPROM 등)에는 기본 입력/출력 시스템(BIOS)이 저장될 수 있고, BIOS는 시동 중과 같은 때에, 컴퓨터(1302) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴들을 저장한다. 휘발성 메모리(1312)는 또한 데이터를 캐싱하기 위한 스태틱 RAM과 같은 고속 RAM을 포함할 수 있다.

[0044] 컴퓨터(1302)는 내부 하드 디스크 드라이브(HDD)(1314)(예를 들면, EIDE, SATA) - 내부 HDD(1314)는 또한 적합한 새시(chassis)에서 외부 사용을 위해 구성될 수도 있음 -, (예를 들면, 이동식 디스켓(1318)에 기록을 하거나 그로부터 판독을 하는) 자기 플로피 디스크 드라이브(FDD)(1316) 및 (예를 들면, CD-ROM 디스크(1322)를 판독하거나, 또는 DVD와 같은 다른 고용량 광 매체에 기록을 하거나 그로부터 판독을 하는) 광 디스크 드라이브(1320)를 더 포함한다. HDD(1314), FDD(1316) 및 광 디스크 드라이브(1320)는 각각 HDD 인터페이스(1324), FDD 인터페이스(1326) 및 광 드라이브 인터페이스(1328)에 의해 시스템 버스(1308)에 접속될 수 있다. 외부 드라이브 구현을 위한 HDD 인터페이스(1324)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술들 중 적어도 하나 또는 양쪽 모두를 포함할 수 있다.

[0045] 드라이브들 및 이들과 관련된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어 등의 비휘발성 저장을 제공한다. 컴퓨터(1302)를 위하여, 드라이브들 및 매체는 적합한 디지털 포맷의 임의의 데이터의 저장을 수용한다. 상기 컴퓨터 판독가능 매체의 설명은 HDD, 이동식 자기 디스켓(예를 들면, FDD), 및 CD 또는 DVD와 같은 이동식 광 매체를 언급하고 있지만, 숙련된 당업자들은 집 드라이브, 자기 카세트, 플래시 메모리 카드, 카트리지 등과 같은, 컴퓨터에 의해 판독가능한 다른 유형의 매체가 예시적인 운영 환경에서 이용될 수도 있고, 또한, 임의의 그러한 매체는 개시된 아키텍처의 새로운 방법들을 수행하기 위한 컴퓨터 실행가능 명령어들을 포함할 수 있다는 것을 인식할 것이다.

[0046] 운영 체제(1330), 하나 이상의 애플리케이션 프로그램(1332), 기타 프로그램 모듈(1334), 및 프로그램 데이터(1336)를 포함하는, 다수의 프로그램 모듈들이 드라이브들 및 휘발성 메모리(1312)에 저장될 수 있다. 그 하나 이상의 애플리케이션 프로그램(1332), 기타 프로그램 모듈(1334), 및 프로그램 데이터(1336)는, 예를 들면, 변경 컴포넌트(102), 데이터(104), 웹 애플리케이션(106), 반전 컴포넌트(108), 데이터 그리드(110), 변경 통지(302 및 304), 업데이트 응답(310 및 312), 변경 트래커(500), 실행 취소 스택(502), 그리드 상태(600, 606, 612, 618, 및 632), 스택 상태(602, 608, 614, 620, 624 및 628), 및 트래커 상태(604, 610, 616, 622, 626 및 630)를 포함할 수 있다.

[0047] 운영 체제, 애플리케이션, 모듈, 및/또는 데이터의 전부 또는 일부는 또한 휘발성 메모리(1312)에 캐싱될 수 있다. 개시된 아키텍처는 다양한 상업적으로 이용 가능한 운영 체제들 또는 운영 체제들의 조합들과 함께 구현될 수 있다는 것을 인식해야 한다.

[0048] 사용자는 하나 이상의 유선/무선 입력 장치들, 예를 들면, 키보드(1338) 및 마우스(1340)와 같은 포인팅 장치를 통하여 컴퓨터(1302)에 명령 및 정보를 입력할 수 있다. (도시되지 않은) 다른 입력 장치들은 마이크, IR 리모컨, 조이스틱, 게임 패드, 스타일러스 펜, 터치 화면 등을 포함할 수 있다. 이들 및 기타 입력 장치들은 종종 시스템 버스(1308)에 연결되는 입력 장치 인터페이스(1342)를 통하여 처리 장치(1304)에 접속되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스 등과 같은 다른 인터페이스들에 의해 접속될 수 있다.

[0049] 모니터(1344) 또는 다른 유형의 디스플레이 장치도 비디오 어댑터(1346) 등의 인터페이스를 통해 시스템 버스(1308)에 접속될 수 있다. 모니터(1344) 외에도, 컴퓨터는 전형적으로 스피커, 프린터 등의 기타 주변 출력 장치(도시되지 않음)를 포함한다.

[0050] 컴퓨터(1302)는 원격 컴퓨터(들)(1348)와 같은 하나 이상의 원격 컴퓨터로의 유선 및/또는 무선 통신을 통한 논리적 접속을 이용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1348)는 워크스테이션, 서버 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서 기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 전형적으로 컴퓨터(1302)와 관련하여 설명된 구성요소들의 다수 또는 전부를 포함하지만, 간결함을 위하여, 메모리/저장 장치(1350)만 예시되었다. 도시된 논리적 접속들은 LAN(local area network)(1352) 및/또는 보다 큰 네트워크, 예를 들면 WAN(wide area network)(1354)으로의 유선/무선 접속을 포함한다. 그러한 LAN 및 WAN 네트워킹 환경은 사무실 및 회사들에서 일반적이고, 인트라넷들과 같은 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 하고, 그러한 네트워크들 모두는 글로벌 통신

네트워크, 예를 들면, 인터넷에 접속될 수 있다.

[0051] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1302)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(1356)를 통해 LAN(1352)에 접속된다. 어댑터(1356)는 LAN(1352)으로의 유선 및/또는 무선 통신을 용이하게 할 수 있고, LAN(1352)은 또한 어댑터(1356)의 무선 기능과 통신하기 위해 그 위에 배치된 무선 액세스 포인트를 포함할 수 있다.

[0052] WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1302)는 모뎀(1358)을 포함할 수 있고, 또는 WAN(1354) 상의 통신 서버에 접속되고, 또는 예를 들면 인터넷을 통하여, WAN(1354) 상에서 통신을 설정하기 위한 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 및/또는 무선 장치일 수 있는 모뎀(1358)은 입력 장치 인터페이스(1342)를 통해 시스템 버스(1308)에 접속된다. 네트워크화된 환경에서, 컴퓨터(1302) 또는 그의 부분들과 관련하여 기술된 프로그램 모듈은 원격 메모리/저장 장치(1350)에 저장될 수 있다. 도시된 네트워크 접속은 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 이해할 것이다.

[0053] 컴퓨터(1302)는, 예를 들면, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, PDA(personal digital assistant), 통신 위성, 무선으로 검출 가능한 태그와 관련된 임의의 장비 또는 위치(예를 들면, 키오스크, 신문 판매점, 화장실), 및 전화기와 무선 통신하도록 동작 가능하게 배치된 무선 장치들과 같은 IEEE 802 계열의 표준들(예를 들면, IEEE 802.11 공중파 변조(over-the-air modulation) 기법들)을 이용하여 유선 및 무선 장치들 또는 엔티티들과 통신하도록 동작 가능하다. 이것은 적어도 Wi-Fi(또는 Wireless Fidelity), WiMax, 및 Bluetooth™ 무선 기술들을 포함한다. 따라서, 통신은 종래의 네트워크에서와 같이 미리 정의된 구조이거나 또는 단순히 적어도 2개의 장치들 사이의 애드 혹 통신일 수 있다. Wi-Fi 네트워크들은 안전하고, 신뢰성 있고, 빠른 무선 접속을 제공하기 위해 IEEE 802.11x(a, b, g 등)로 불리는 라디오 기술들을 이용한다. Wi-Fi 네트워크는 컴퓨터들을 서로, 인터넷에, 또한 (IEEE 802.3 관련 매체 및 기능들을 이용하는) 유선 네트워크에 접속하기 위해 이용될 수 있다.

[0054] 이제 도 14를 참조하면, 멀티-레벨 실행 취소 처리를 위한 예시적인 클라이언트-서버 컴퓨팅 환경(1400)의 개략적인 블록도가 예시되어 있다. 환경(1400)은 하나 이상의 클라이언트(들)(1402)를 포함한다. 클라이언트(들)(1402)는 하드웨어 및/또는 소프트웨어(예를 들면, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 클라이언트(들)(1402)는 예를 들면 쿠키(들) 및/또는 관련된 컨텍스트 정보를 수용할 수 있다.

[0055] 환경(1400)은 또한 하나 이상의 서버(들)(1404)를 포함한다. 서버(들)(1404)도 하드웨어 및/또는 소프트웨어(예를 들면, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 서버들(1404)은 예를 들면 아키텍처를 채용함으로써 변환을 수행하는 스레드들을 수용할 수 있다. 클라이언트(1402)와 서버(1404) 사이의 하나의 가능한 통신은 2개 이상의 컴퓨터 프로세스들 사이에 전송되도록 적응된 데이터 패킷의 형태일 수 있다. 데이터 패킷은 예를 들면 쿠키 및/또는 관련된 컨텍스트 정보를 포함할 수 있다. 환경(1400)은 클라이언트(들)(1402)와 서버(들)(1404) 사이의 통신을 용이하게 하기 위해 채용될 수 있는 통신 프레임워크(1406)(예를 들면, 인터넷과 같은 글로벌 통신 네트워크)를 포함한다.

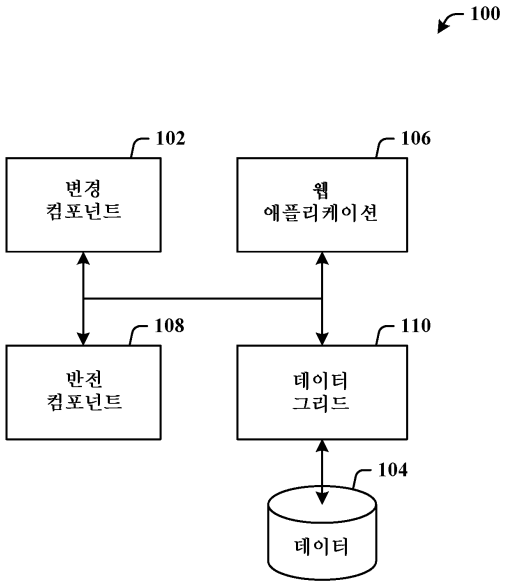
[0056] 통신은 유선(광섬유를 포함함) 및/또는 무선 기술을 통해 용이해질 수 있다. 클라이언트(들)(1402)는 클라이언트(들)(1402)에 로컬인 정보(예를 들면, 쿠키(들) 및/또는 관련된 컨텍스트 정보)를 저장하기 위해 채용될 수 있는 하나 이상의 클라이언트 데이터 저장소(들)(1408)에 동작 가능하게 접속된다. 유사하게, 서버(들)(1404)는 서버들(1404)에 로컬인 정보를 저장하기 위해 채용될 수 있는 하나 이상의 서버 데이터 저장소(들)(1410)에 동작 가능하게 접속된다.

[0057] 클라이언트(들)(1402)는 웹 애플리케이션(106)을 포함할 수 있고 클라이언트 데이터 저장소(들)(1408)는 데이터(104)를 포함할 수 있다. 서버(들)(1404)는 서버(202) 및 비동기식 유효성 검사 프로세스들(306 및 308)을 포함할 수 있고, 서버 데이터 저장소(들)(1410)는 문서(200)를 포함할 수 있다.

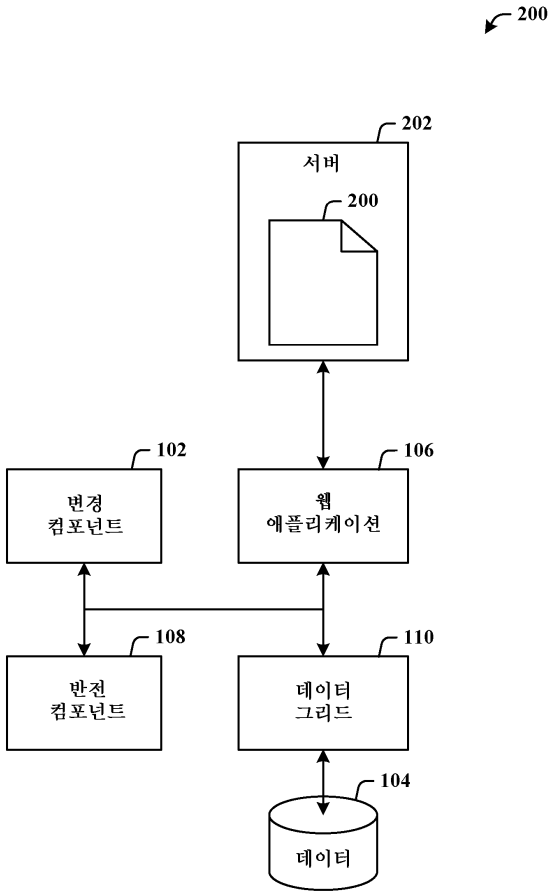
[0058] 위에 설명된 것은 개시된 아키텍처의 예들을 포함한다. 물론, 컴포넌트들 및/또는 방법들의 생각할 수 있는 모든 조합을 설명하는 것은 가능하지 않지만, 통상의 기술을 가진 당업자는 많은 추가적인 조합들 및 치환들이 가능하다는 것을 인지할 수 있다. 따라서, 이 새로운 아키텍처는 부속된 청구항들의 정신 및 범위 안에 있는 모든 그러한 변경들, 수정들 및 변형들을 포함하도록 의도되어 있다. 또한, "포함한다(includes)"라는 용어가 상세한 설명 또는 청구항들에서 이용되는 한에는, 그러한 용어는 "포함하는(comprising)"이라는 용어가 청구항에서 전이구로서 채용될 때 해석되는 것과 같이 "포함하는(comprising)"이라는 용어와 유사한 방식으로 포괄적인 것으로 의도된다.

## 도면

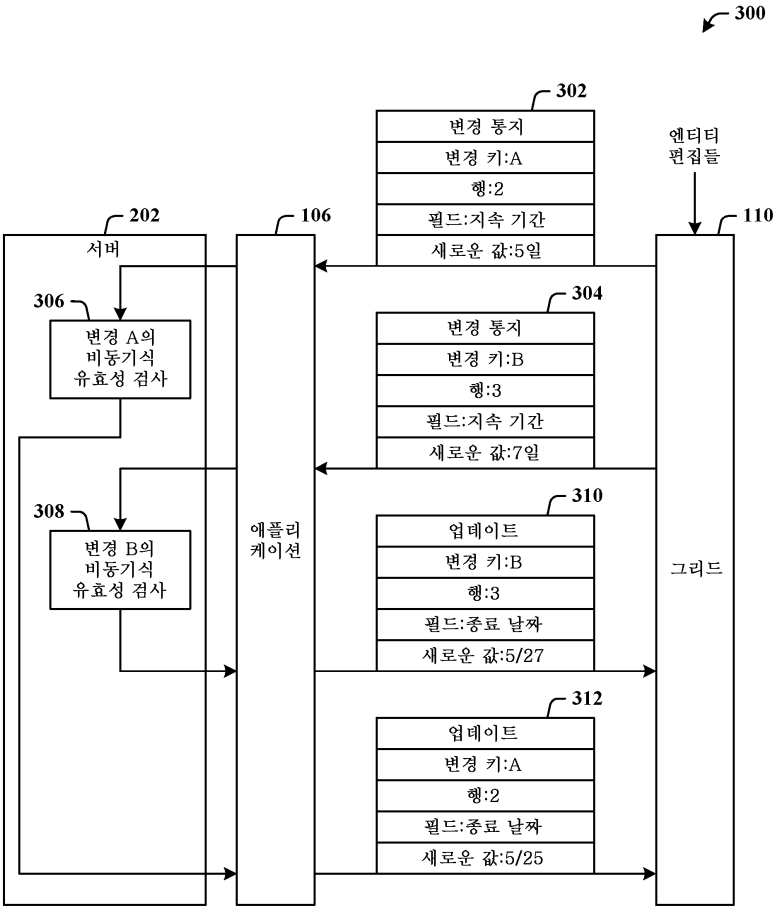
### 도면1



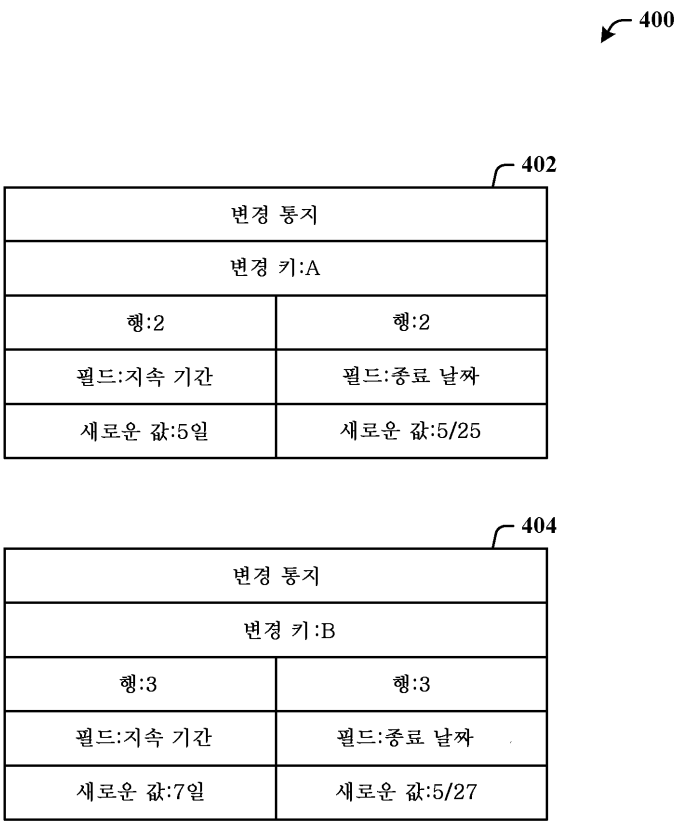
도면2


200
202
서버
200
102
변경
컴포넌트
106
웹
애플리케이션
108
반전
컴포넌트
110
데이터
그리드
104
데이터

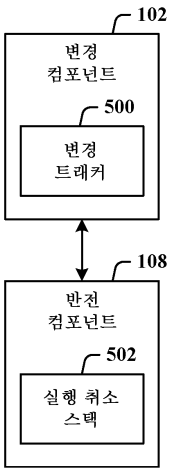
도면3


300
302
변경 통지
변경 키:A
행:2
필드:지속 기간
새로운 값:5일
엔티티
편집들
110
202
서버
106
306
변경 A의
비동기식
유효성 검사
304
변경 통지
변경 키:B
행:3
필드:지속 기간
새로운 값:7일
308
변경 B의
비동기식
유효성 검사
애플리
케이션
310
업데이트
변경 키:B
행:3
필드:종료 날짜
새로운 값:5/27
그리드
312
업데이트
변경 키:A
행:2
필드:종료 날짜
새로운 값:5/25

도면4


400
402
변경 통지
변경 키:A
행:2
행:2
필드:지속 기간
필드:종료 날짜
새로운 값:5일
새로운 값:5/25
404
변경 통지
변경 키:B
행:3
행:3
필드:지속 기간
필드:종료 날짜
새로운 값:7일
새로운 값:5/27

도면5


102
변경
컴포넌트
500
변경
트래커
108
반전
컴포넌트
502
실행 취소
스택

도면6


그리드의 상태
600
TASK NAME
START
FINISH
DURATION
1
SEND THE FENCE
MON 5/21/XX
MON 5/21/XX
1 DAY
2
PAINT THE FENCE
MON 5/21/XX
MON 5/21/XX
1 DAY
3
CLEAN-UP TOOLS
MON 5/21/XX
MON 5/21/XX
1 DAY
실행 취소 스택의 상태
602
변경 트래커의 상태
604
그리드의 상태
606
TASK NAME
START
FINISH
DURATION
1
SEND THE FENCE
MON 5/21/XX
MON 5/21/XX
1 DAY
2
PAINT THE FENCE
MON 5/21/XX
TUE 5/22/XX
2 DAYS
3
CLEAN-UP TOOLS
MON 5/21/XX
MON 5/21/XX
1 DAY
실행 취소 스택의 상태
608
EXECUTE:
UPDATECELL{{2,DURATION},2}
UNDO:
UPDATECELL{{2,DURATION},1}
변경 트래커의 상태
610
EXECUTE:
UPDATECELL{{2,FINISH},5/23}
UNDO:
UPDATECELL{{2,FINISH},5/22}
{2,DURATION}:> 2
{2,FINISH}:->5/23

도면7


그리드의 상태
612
TASK NAME
START
DURATION
1
SEND THE FENCE
MON 5/21/XX
1 DAY
2
PAINT THE FENCE
MON 5/21/XX
2 DAYS
3
CLEAN-UP TOOLS
MON 5/21/XX
1 DAY
실행 취소 스택의 상태
614
EXECUTE:
HIDECOLUMN{FINISH}
UNDO:
SHOWCOLUMN{FINISH}
EXECUTE:
UPDATECELL{{2,DURATION},2}
UNDO:
UPDATECELL{{2,DURATION},1}
EXECUTE:
UPDATECELL{{2,FINISH},5/23}
UNDO:
UPDATECELL{{2,FINISH},5/22}
변경 트래커의 상태
{2,DURATION}-> 2
{2,FINISH}->5/23
610
그리드의 상태
TASK NAME
START
DURATION
1
SEND THE FENCE
MON 5/21/XX
1 DAY
2
PAINT THE FENCE
MON 5/21/XX
2 DAYS
3
CLEAN-UP TOOLS
MON 5/21/XX
1 DAY
실행 취소 스택의 상태
EXECUTE:
HIDECOLUMN{FINISH}
UNDO:
SHOWCOLUMN{FINISH}
EXECUTE:
UPDATECELL{{2,DURATION},2}
UNDO:
UPDATECELL{{2,DURATION},1}
EXECUTE:
UPDATECELL{{2,FINISH},5/23}
UNDO:
UPDATECELL{{2,FINISH},5/22}
변경 트래커의 상태
616

도면8

그리드의 상태 618

| | TASK NAME | START | DURATION |
|---|---|---|---|
| 1 | SEND THE FENCE | MON 5/21/XX | 1 DAY |
| 2 | PAINT THE FENCE | MON 5/21/XX | 2 DAYS |
| 3 | CLEAN-UP PROJECT | MON 5/21/XX | 1 DAY |

실행 취소 스택의 상태 620

EXECUTE:
UPDATECELL{{3,NAME},'...CT}
UNDO:
UPDATECELL{{3,NAME},'...OLS}

EXECUTE:
HIDECOLUMN{FINISH}
UNDO:
SHOWCOLUMN{FINISH}

EXECUTE:
UPDATECELL{{2,DURATION},2}
UNDO:
UPDATECELL{{2,DURATION},1}

EXECUTE:
UPDATECELL{{2,FINISH},5/23}
UNDO:
UPDATECELL{{2,FINISH},5/22}

변경 트래커의 상태 622

{3,NAME}-> 'CLEAN-UP PROJECT'

도면9

그리드의 상태

| | TASK NAME | START | DURATION |
|---|---|---|---|
| 1 | SEND THE FENCE | MON 5/21/XX | 1 DAY |
| 2 | PAINT THE FENCE | MON 5/21/XX | 2 DAYS |
| 3 | CLEAN-UP TOOLS | MON 5/21/XX | 1 DAY |

실행 취소 스택의 상태 624

EXECUTE:
HIDECOLUMN{FINISH}
UNDO:
SHOWCOLUMN{FINISH}

EXECUTE:
UPDATECELL{{2,DURATION},2}
UNDO:
UPDATECELL{{2,DURATION},1}

EXECUTE:
UPDATECELL{{2,FINISH},5/23}
UNDO:
UPDATECELL{{2,FINISH},5/22}

변경 트래커의 상태 626

{2,DURATION}:-> 1
{2,FINISH}:->5/22

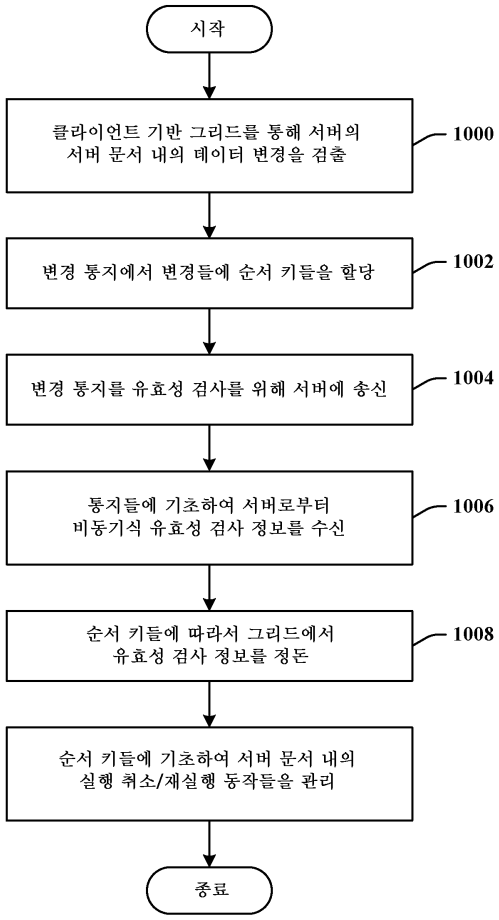
그리드의 상태 632

| | TASK NAME | START | FINISH | DURATION |
|---|---|---|---|---|
| 1 | SEND THE FENCE | MON 5/21/XX | MON 5/21/XX | 1 DAY |
| 2 | PAINT THE FENCE | MON 5/21/XX | MON 5/21/XX | 1 DAY |
| 3 | CLEAN-UP TOOLS | MON 5/21/XX | MON 5/21/XX | 1 DAY |

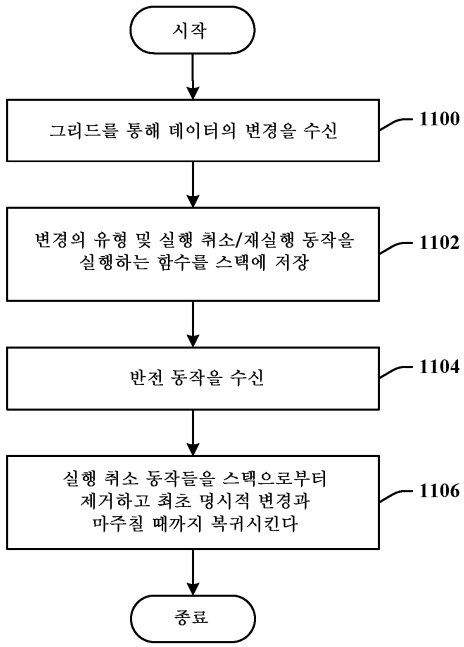
실행 취소 스택의 상태 628

변경 트래커의 상태 630

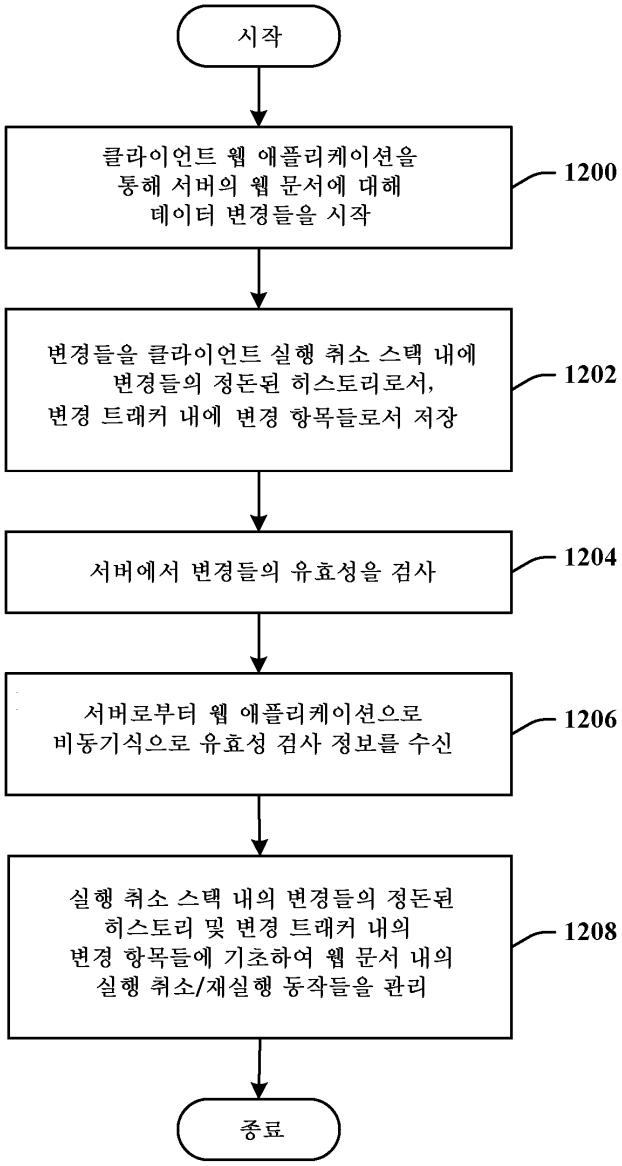
도면10


시작
클라이언트 기반 그리드를 통해 서버의 서버 문서 내의 데이터 변경을 검출
1000
변경 통지에서 변경들에 순서 키들을 할당
1002
변경 통지를 유효성 검사를 위해 서버에 송신
1004
통지들에 기초하여 서버로부터 비동기식 유효성 검사 정보를 수신
1006
순서 키들에 따라서 그리드에서 유효성 검사 정보를 정돈
1008
순서 키들에 기초하여 서버 문서 내의 실행 취소/재실행 동작들을 관리
종료

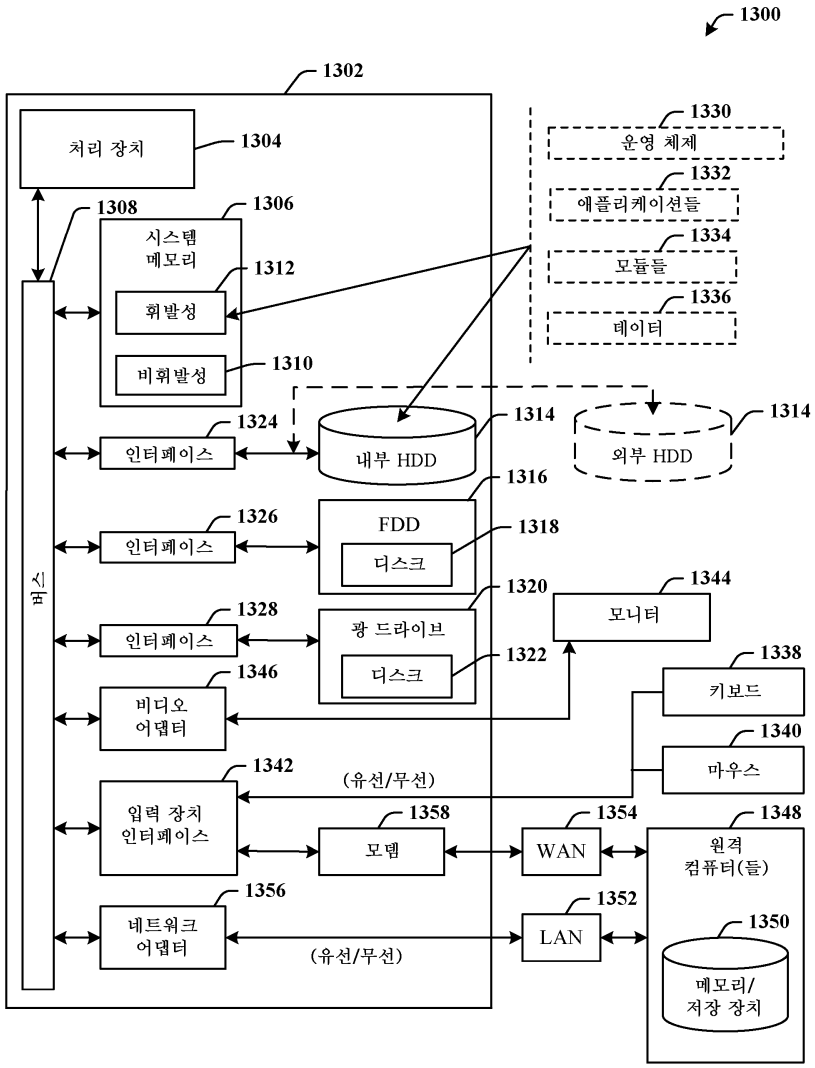
도면11


시작
그리드를 통해 데이터의 변경을 수신
1100
변경의 유형 및 실행 취소/재실행 동작을 실행하는 함수를 스택에 저장
1102
반전 동작을 수신
1104
실행 취소 동작들을 스택으로부터 제거하고 최초 명시적 변경과 마주칠 때까지 복귀시킨다
1106
종료

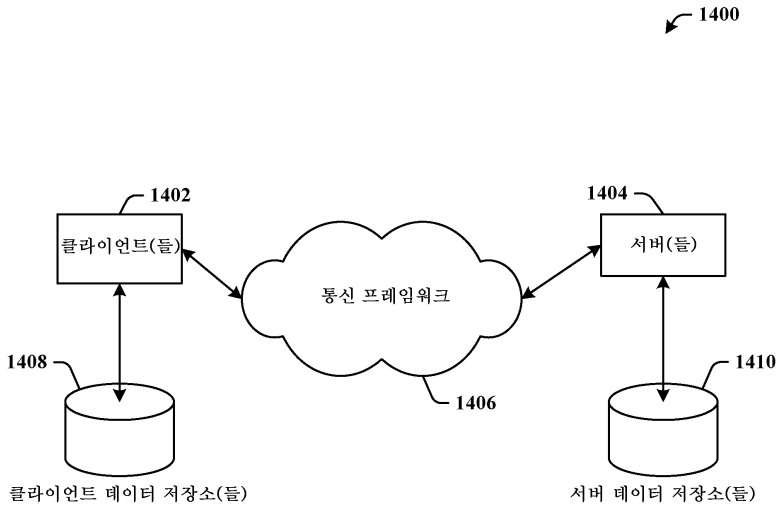
도면12


시작
클라이언트 웹 애플리케이션을 통해 서버의 웹 문서에 대해 데이터 변경들을 시작
1200
변경들을 클라이언트 실행 취소 스택 내에 변경들의 정돈된 히스토리로서, 변경 트래커 내에 변경 항목들로서 저장
1202
서버에서 변경들의 유효성을 검사
1204
서버로부터 웹 애플리케이션으로 비동기식으로 유효성 검사 정보를 수신
1206
실행 취소 스택 내의 변경들의 정돈된 히스토리 및 변경 트래커 내의 변경 항목들에 기초하여 웹 문서 내의 실행 취소/재실행 동작들을 관리
1208
종료

도면13


1300
1302
처리 장치
1304
1308
시스템 메모리
1306
1312
휘발성
1310
비휘발성
버스
1330
운영 체제
1332
애플리케이션들
1334
모듈들
1336
데이터
1324
인터페이스
내부 HDD
1314
외부 HDD
1314
1316
FDD
디스크
1318
1326
인터페이스
1320
광 드라이브
디스크
1322
1328
인터페이스
1344
모니터
1338
키보드
1340
마우스
1346
비디오 어댑터
1342
입력 장치 인터페이스
(유선/무선)
1358
모뎀
1354
WAN
1348
원격 컴퓨터(들)
1356
네트워크 어댑터
(유선/무선)
1352
LAN
1350
메모리/ 저장 장치

도면14


1400
1402
클라이언트(들)
통신 프레임워크
1404
서버(들)
1408
1406
1410
클라이언트 데이터 저장소(들)
서버 데이터 저장소(들)

【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 제9항 11줄

【변경전】

상기 데이트 그리드

【변경후】

상기 데이터 그리드