



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2016년02월23일

(11) 등록번호 10-1597082

(24) 등록일자 2016년02월17일

(51) 국제특허분류(Int. Cl.)

G06F 17/21 (2006.01) G06F 17/24 (2006.01)

(52) CPC특허분류

G06F 17/21 (2013.01)

G06F 17/241 (2013.01)

(21) 출원번호 10-2015-7002346(분할)

(22) 출원일자(국제) 2008년10월28일

심사청구일자 2015년01월28일

(85) 번역문제출일자 2015년01월28일

(65) 공개번호 10-2015-0018896

(43) 공개일자 2015년02월24일

(62) 원출원 특허 10-2010-7009797

원출원일자(국제) 2008년10월28일

심사청구일자 2013년10월07일

(86) 국제출원번호 PCT/US2008/081456

(87) 국제공개번호 WO 2009/061638

국제공개일자 2009년05월14일

(30) 우선권주장

11/938,082 2007년11월09일 미국(US)

(56) 선행기술조사문헌

US07249314 B2*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

베일리, 조나단 벅켓

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 코포레이션 국제 특
허부 내

변스테인, 예단 조셉

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 코포레이션 국제 특
허부 내

(뒷면에 계속)

(74) 대리인

제일특허법인

전체 청구항 수 : 총 20 항

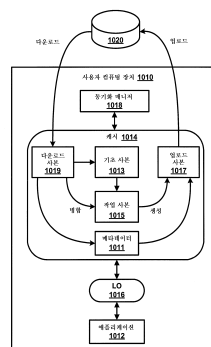
심사관 : 이복현

(54) 발명의 명칭 공동 저작

(57) 요약

공동 저작 애플리케이션은 2명 이상의 사용자들이 한 문서를 동시에 편집할 수 있는 저작 환경을 제공한다. 각 사용자는 상기 문서의 사본을 편집하고, 상기 문서의 마스터 사본에 업데이트들을 송신하고, 상기 문서의 마스터 사본으로부터 업데이트들을 수신한다. 상기 저작 환경은 일반적으로 상기 사용자들이 상충되는 편집 명령을 상기 문서의 마스터 사본에 제공하는 것을 금한다. 예를 들면, 각 사용자는 상기 문서 내의 하나 이상의 데이터 단위에 관한 콘텐츠 잠금을 생성할 수 있다. 상기 저작 환경은 상기 사용자의 요청에 의해서만 콘텐츠 잠금을 자동으로 동기화시키고 콘텐츠를 동기화시킬 수 있다.

대표도 - 도12



(72) 발명자

나이트, 마크 롤란드

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

안토스, 크리스토퍼 제임스

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

시몬즈, 앤드류 리차드

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

존스, 브라이언 마이클

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

클라크, 사이몬 피터

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

선더랜드, 에드가 마크

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

로빈스, 데이비드 벤자민

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

보세, 미코 아나브 사크야 싱하

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

명세서

청구범위

청구항 1

제1 컴퓨팅 장치에서 문서를 저작하기 위한 방법으로서,

상기 제1 컴퓨팅 장치에서, 상기 문서의 제1 데이터 단위가 제1 사용자에 의해 편집되도록 허용하는 단계와,

상기 제1 컴퓨팅 장치에서, 상기 제1 데이터 단위가 편집될 경우 상기 문서가 상기 제1 사용자에 의해 편집되고 있다는 것을 나타내는 제1 주석(annotation)을 상기 문서상에 제공하는 단계와,

상기 제1 컴퓨팅 장치에서, 제1 메타데이터 업데이트를 제2 컴퓨팅 장치로 송신하는 단계 - 상기 제1 메타데이터 업데이트는 상기 문서의 상기 제1 데이터 단위가 편집되고 있다는 것을 나타냄 - 와,

상기 제1 컴퓨팅 장치에서, 제2 메타데이터 업데이트를 상기 제2 컴퓨팅 장치로부터 수신하는 단계 - 상기 제2 메타데이터 업데이트는 상기 문서의 제2 데이터 단위를 편집하고 있는 제2 사용자의 이름을 나타냄 - 와,

상기 제1 컴퓨팅 장치에서, 상기 제2 데이터 단위에 대한 제1 잠금을 설정하는 단계 - 상기 제2 데이터 단위에 대한 상기 제1 잠금은 상기 제2 데이터 단위가 상기 제1 컴퓨팅 장치에서 편집되는 것을 금지함 - 와,

상기 제1 컴퓨팅 장치에서, 상기 문서상에 제2 주석을 제공하는 단계 - 상기 제2 주석은 상기 제2 데이터 단위가 상기 제2 사용자에 의해 편집되고 있다는 것을 나타냄 - 와,

상기 제1 컴퓨팅 장치에서, 상기 문서에 관한 콘텐츠 업데이트들을 상기 제2 컴퓨팅 장치로부터 수신하는 단계와,

상기 제1 컴퓨팅 장치에서, 상기 콘텐츠 업데이트들을 수신하는 단계에 응답하여 상기 문서 내로 상기 콘텐츠 업데이트들을 인스턴스화하기 위한 명령을 상기 제1 사용자로부터 수신하는 단계와,

상기 제1 컴퓨팅 장치에서, 상기 제1 사용자로부터의 상기 명령에 응답하여 상기 콘텐츠 업데이트들을 통합하도록 상기 문서를 업데이트하는 단계

를 포함하는 저작 방법.

청구항 2

제1항에 있어서,

상기 제1 주석은 상기 제1 사용자가 커서를 상기 제1 데이터 단위로 이동시킬 경우 상기 문서상에 디스플레이되는

저작 방법.

청구항 3

제1항에 있어서,

상기 제1 주석은 네임태그를 포함하고, 상기 네임태그는 상기 제1 사용자의 이름을 나타내는

저작 방법.

청구항 4

제1항에 있어서,

상기 제1 데이터 단위가 편집될 경우 상기 제1 데이터 단위에 대한 제2 잠금을 설정하는 단계 - 상기 제2 잠금은 상기 제1 사용자가 아닌 다른 사용자들에 의해 상기 제1 데이터 단위가 편집되는 것을 금지함 -
를 더 포함하는 저작 방법.

청구항 5

제4항에 있어서,
상기 제1 메타데이터 업데이트는 상기 제1 데이터 단위에 대해 상기 제2 잠금이 설정되었다는 것을 나타내는 저작 방법.

청구항 6

제4항에 있어서,
상기 제1 사용자는 상기 제2 잠금의 소유자인 저작 방법.

청구항 7

제1항에 있어서,
상기 제1 메타데이터 업데이트는 상기 제1 사용자의 이름을 포함하는 저작 방법.

청구항 8

제1항에 있어서,
상기 제2 주석은 네임태그를 포함하고, 상기 네임태그는 상기 제2 사용자의 이름을 나타내는 저작 방법.

청구항 9

제8항에 있어서,
상기 네임태그는 상기 제2 사용자를 색상에 의해 식별하는 저작 방법.

청구항 10

제1항에 있어서,
상기 제1 컴퓨팅 장치에서, 상기 문서에 관하여 업데이트들이 이용 가능할 경우 경고를 디스플레이하는 단계를 더 포함하는 저작 방법.

청구항 11

제10항에 있어서,
상기 경고는 시각적 경고인
저작 방법.

청구항 12

제11항에 있어서,
상기 경고는 상기 제1 컴퓨팅 장치의 사용자 인터페이스상의 텍스트 박스를 포함하는
저작 방법.

청구항 13

제1항에 있어서,
상기 콘텐츠 업데이트들을 통합하도록 상기 문서를 업데이트하는 단계는, 상기 제1 컴퓨팅 장치에서 상기 제1 사용자에게 의한 액션에 의해 상기 업데이트하는 단계를 개시하는 단계를 더 포함하는
저작 방법.

청구항 14

제1 컴퓨팅 장치에서 문서를 저작하기 위한 방법으로서,
상기 제1 컴퓨팅 장치에서, 상기 문서의 제1 데이터 단위를 편집하는 단계 - 상기 제1 데이터 단위는 제1 사용자에게 의해 편집되고 있는 것임 - 와,
상기 제1 컴퓨팅 장치에서, 상기 제1 데이터 단위가 편집될 경우 상기 문서가 상기 제1 사용자에게 의해 편집되고 있다는 것을 나타내는 제1 주석(annotation)을 상기 문서상에 제공하는 단계 - 상기 제1 주석은 상기 제1 사용자가 커서를 상기 제1 데이터 단위로 이동시킬 경우 상기 문서상에 디스플레이되고, 상기 제1 주석은 상기 제1 사용자의 이름을 나타내는 네임태그를 포함함 - 와,
상기 제1 컴퓨팅 장치에서, 상기 제1 데이터 단위가 편집될 경우 상기 제1 데이터 단위에 대한 제1 잠금을 설정하는 단계 - 상기 제1 잠금은 상기 제1 사용자가 아닌 다른 사용자에게 의해 상기 제1 데이터 단위가 편집되는 것을 금지함 - 와,
상기 제1 컴퓨팅 장치에서, 제1 메타데이터 업데이트를 제2 컴퓨팅 장치로 송신하는 단계 - 상기 제1 메타데이터 업데이트는 상기 문서의 상기 제1 데이터 단위가 편집되고 있다는 것을 나타내고, 상기 제1 메타데이터 업데이트는 또한 상기 제1 사용자에게 의해 상기 제1 데이터 단위가 잠겼다는 것을 나타냄 - 와,
상기 제1 컴퓨팅 장치에서, 제2 메타데이터 업데이트를 상기 제2 컴퓨팅 장치로부터 수신하는 단계 - 상기 제2 메타데이터 업데이트는 제2 데이터 단위를 편집하고 있는 제2 사용자의 이름을 나타냄 - 와,
상기 제1 컴퓨팅 장치에서, 상기 제2 데이터 단위에 대한 제2 잠금을 설정하는 단계 - 상기 제2 데이터 단위에 대한 상기 제2 잠금은, 상기 제2 데이터 단위가 상기 제2 사용자에게 의해 편집되고 있으며 상기 제2 데이터 단위를 편집할 경우 충돌이 발생할 수 있다는 경고를 상기 제1 사용자에게 제공하되, 상기 제2 데이터 단위가 상기 제1 컴퓨팅 장치에서 편집되는 것을 금지하지는 않음 - 와,
상기 제1 컴퓨팅 장치에서, 상기 문서상에 제2 주석을 제공하는 단계 - 상기 제2 주석은 상기 제2 데이터 단위가 상기 제2 사용자에게 의해 편집되고 있다는 것을 나타내고, 상기 제2 주석은 상기 제2 사용자의 이름을 나타내는 네임태그를 포함함 -

를 포함하는 저작 방법.

청구항 15

제14항에 있어서,

상기 제1 컴퓨팅 장치에서, 상기 문서에 관한 콘텐츠 업데이트들을 상기 제2 컴퓨팅 장치로부터 수신하는 단계를 더 포함하는 저작 방법.

청구항 16

제15항에 있어서,

상기 콘텐츠 업데이트들을 상기 제2 컴퓨팅 장치로부터 수신할 경우 상기 제1 컴퓨팅 장치에서의 상기 문서상에 시각적 경고를 디스플레이하는 단계

를 더 포함하는 저작 방법.

청구항 17

제16항에 있어서,

상기 시각적 경고는 상기 제1 컴퓨팅 장치에서의 상기 문서상의 텍스트 박스를 포함하는 저작 방법.

청구항 18

제16항에 있어서,

상기 시각적 경고는 색상이 변하는 상기 제2 주석 내의 네임태그를 포함하는 저작 방법.

청구항 19

제15항에 있어서,

상기 제1 컴퓨팅 장치에서, 상기 콘텐츠 업데이트들 내에서 수신된 상기 제2 데이터 단위에 대한 변경사항들을 통합하도록 상기 제2 데이터 단위를 업데이트하는 단계

를 더 포함하는 저작 방법.

청구항 20

명령어를 포함하는 컴퓨터-판독가능 저장 매체로서, 상기 명령어는 제1 컴퓨팅 장치에 의해 실행될 경우 상기 제1 컴퓨팅 장치로 하여금,

문서의 제1 데이터 단위를 편집하고 - 상기 제1 데이터 단위는 제1 사용자에 의해 편집되는 것임 -,

상기 문서가 상기 제1 사용자에 의해 편집되고 있다는 것을 나타내는 제1 주석을 상기 문서상에 제공하고,

제1 메타데이터 업데이트를 제2 컴퓨팅 장치로 송신하고 - 상기 제1 메타데이터 업데이트는 상기 문서의 상기 제1 데이터 단위가 편집되고 있다는 것을 나타냄 -,

제2 메타데이터 업데이트를 상기 제2 컴퓨팅 장치로부터 수신하고 - 상기 제2 메타데이터 업데이트는 제2 데이터 단위를 편집하고 있는 제2 사용자의 이름을 나타냄 -,

상기 제2 데이터 단위에 대한 제1 잠금을 설정하고 - 상기 제2 데이터 단위에 대한 상기 제1 잠금은 상기 제2 데이터 단위가 상기 제1 컴퓨팅 장치에서 편집되는 것을 금지함 -,

상기 문서상에 제2 주석을 제공하고 - 상기 제2 주석은 상기 제2 데이터 단위가 상기 제2 사용자에 의해 편집되고 있다는 것을 나타냄 -,

상기 문서에 관한 콘텐츠 업데이트들을 상기 제2 컴퓨팅 장치로부터 수신하고,

상기 콘텐츠 업데이트들을 수신하는 것에 응답하여 상기 문서 내로 상기 콘텐츠 업데이트들을 인스턴스화하기 위한 명령을 상기 제1 사용자로부터 수신하고,

상기 제1 사용자로부터 상기 명령을 수신한 후, 수신된 상기 콘텐츠 업데이트들을 통합하도록 상기 문서를 업데이트하도록 하는

컴퓨터-판독가능 저장 매체.

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

청구항 47

삭제

청구항 48

삭제

청구항 49

삭제

청구항 50

삭제

청구항 51

삭제

청구항 52

삭제

청구항 53

삭제

청구항 54

삭제

청구항 55

삭제

청구항 56

삭제

청구항 57

삭제

청구항 58

삭제

청구항 59

삭제

청구항 60

삭제

청구항 61

삭제

청구항 62

삭제

청구항 63

삭제

청구항 64

삭제

청구항 65

삭제

청구항 66

삭제

발명의 설명

배 경 기 술

- [0001] 전통적인 공동 편집은 순차적으로 수행되는 경향이 있다. 사용자들은 교대로 문서에 액세스하고, 문서를 편집하고, 그들의 편집을 저장한다. 액세스하는 사용자는 액세스하는 사용자가 문서를 편집하고 있을 때 다른 사용자들이 문서를 편집하는 것을 금하기 위해 파일에 잠금(lock)을 걸 수 있다. 반복적인 편집 프로세스는 각 사용자가 문서를 편집하는 데 차례를 기다릴 수 있으므로 지연을 일으킬 수 있다. 또한, 반복적인 편집 프로세스는 관리하기 어려울 수 있다. 예를 들면, 각 사용자는 누가 문서의 어느 부분들을 편집하고 있고, 문서의 어느 버전이 가장 최근의 것이고, 그 사용자가 언제 차례가 될지를 계속 추적할 필요가 있을 수 있다.
- [0002] 전통적인 공동 편집의 다른 유형들에서, 각 사용자는 문서의 상이한 사본을 편집할 수 있다. 그 후, 편집된 사본들 모두가 단일 문서로 병합될 수 있다. 이 대규모의 병합은 또한 지연을 일으켜, 다수의 편집 충돌들을 초래하고, 및/또는 관리하기 어려울 수 있다. 예를 들면, 문서들을 병합하는 것에 대한 책임이 있는 사용자는 문서들 사이의 관계를 추적할 필요가 있을 수 있다. 그 사용자는 또한 편집된 사본들 중 2개 이상의 사본들 사이의 충돌들을 해결하는 것에 대한 책임이 있을 수 있다.
- [0003] 이러한 및 다른 고려 사항들에 관련하여 본 명세가 만들어졌다.

발명의 내용

- [0004] 이 개요는 아래 상세한 설명에서 더 설명되는 선택된 개념들을 단순화된 형태로 소개하기 위해 제공된다. 이 개요는 청구된 내용의 중요한 특징들 또는 본질적인 특징들을 확인하기 위해 의도된 것이 아니고, 청구된 내용의 범위를 결정하는 데 도움이 되는 것으로서 의도된 것도 아니다.
- [0005] 공동 저작 애플리케이션은 2명 이상의 사용자들이 한 문서를 동시에 편집할 수 있는 저작 환경을 제공한다. 각 사용자는 상기 문서의 사본을 편집하고, 주기적으로 상기 문서의 마스터 사본에 업데이트들을 송신하고, 주기적으로 상기 문서의 마스터 사본으로부터 업데이트들을 수신한다. 상기 저작 환경은 일반적으로 상기 사용자들이 상충되는 편집 명령을 상기 문서의 마스터 사본에 제공하는 것을 금한다. 다른 실시예들에서, 상기 저작 환경은 상기 사용자들이 상이한 시간에 상기 문서를 편집한다면 편집 충돌들을 금할 수 있다.
- [0006] 본 명세의 양태들에 따르면, 저작되는 각 문서는 하나 이상의 데이터 단위들로 분할될 수 있다. 각 사용자는 상기 데이터 단위들 중 하나 이상의 데이터 단위에 관한 콘텐츠 잠금(content lock)을 생성할 수 있다. 데이터 단위에 관한 콘텐츠 잠금을 생성하는 것은 다른 사용자들이 잠긴 데이터 단위를 편집하는 것을 금한다. 일 실시예에서, 콘텐츠 잠금들은 상기 사용자가 상기 문서를 편집할 때 자동으로 확장 및/또는 축소될 수 있다.
- [0007] 본 명세의 다른 양태들에 따르면, 상기 저작 환경은 상기 문서의 사용자 사본들 및 마스터 사본 사이에 콘텐츠 및 메타데이터 양쪽 모두를 동기화시킨다. 일반적으로, 상기 저작 환경은 상기 사용자들 중 한 명의 요청에 의해서만 메타데이터를 자동으로 동기화시키고 콘텐츠를 동기화시킨다. 일 실시예에서, 상기 메타데이터는 콘텐츠 잠금들을 포함한다.
- [0008] 이러한 및 다른 특징들 및 이점들은 다음의 상세한 설명을 읽고 관련 도면들을 검토하는 것으로부터 명백해질 것이다. 전술한 일반적인 설명 및 다음의 상세한 설명은 양쪽 모두 단지 설명적인 것이고 청구된 양태들을 제한하는 것이 아니라는 것을 이해해야 할 것이다.

도면의 간단한 설명

[0009]

- 도 1은 본 명세의 발명의 양태들의 예들인 특징들을 갖는 예시의 저작 시스템을 예시하는 개략 블록도이다.
- 도 2는 본 명세의 원리들에 따른, 제1 컴퓨팅 장치 상에 저장된 문서가 콘텐츠 및 메타데이터를 포함할 수 있는 도 1의 저작 시스템을 예시하는 개략 블록도이다.
- 도 3은 본 명세의 원리들에 따른, 저작 시스템이 사용자 컴퓨팅 장치 상에 저장된 문서의 사본을 그 문서의 마스터 사본과 동기화시킬 수 있는 예시적인 동기화 프로세스를 예시하는 흐름도이다.
- 도 4는 본 명세의 원리들에 따른, 5개의 데이터 단위들을 갖고, 그 중 제2 데이터 단위가 잠겨 있는 문서의 개략 블록도이다.
- 도 5는 본 명세의 원리들에 따른, 제3 데이터 단위에 콘텐츠 잠금이 추가된 도 4의 문서의 개략 블록도이다.
- 도 6은 본 명세의 원리들에 따른, 제2 및 제3 데이터 단위들에 대한 콘텐츠 잠금들이 해제되고 제5 데이터 단위에 새로운 콘텐츠 잠금이 추가된 도 4의 문서의 개략 블록도이다.
- 도 7은 본 명세의 원리들에 따른 잠금 테이블의 개략 블록도이다.
- 도 8은 본 명세의 원리들에 따른, 저작 애플리케이션의 사용자에게 의해 행해진 업데이트들을 다른 사용자들과 공유하기 위해 저작 애플리케이션에 의해 구현되는 예시의 동기화 프로세스를 예시하는 흐름도이다.
- 도 9는 본 명세의 원리들에 따른, 저작 애플리케이션에 의해 편집되고 있는 문서 내에 다른 사용자들에 의해 행해진 업데이트들을 인스턴스화하기 위해 저작 애플리케이션에 의해 구현되는 다른 예시의 동기화 프로세스를 예시하는 흐름도이다.
- 도 10은 본 명세의 원리들에 따른, 저작될 문서의 마스터 사본이 저장될 제1 컴퓨팅 장치를 포함하는 저작 환경의 개략 블록도이다.
- 도 11은 본 명세의 원리들에 따른 저작 환경을 구현하도록 구성된 사용자 컴퓨팅 시스템의 개략 블록도이다.
- 도 12는 본 명세의 원리들에 따른 사용자 컴퓨팅 장치 상의 저작 애플리케이션에 의해 구현되는 동기화 사이클을 예시하는 저작 시스템의 개략 블록도이다.
- 도 13은 본 명세의 원리들에 따른, 사용자 컴퓨팅 시스템이 문서를 편집하는 동안에 업데이트들을 송신 및 수신할 수 있는 다른 예시적인 동기화 프로세스에 대한 동작 흐름을 예시하는 흐름도이다.
- 도 14-26은 본 명세의 원리들에 따른, 제1 및 제2 사용자 컴퓨팅 장치들의 제1 사용자 및 제2 사용자가 한 문서를 동시에 편집하는 공동 저작 세션 동안에 상이한 시점들에서 제1 컴퓨팅 장치, 제1 사용자 컴퓨팅 장치, 및 제2 사용자 컴퓨팅 장치 상에 저장된 캐시들의 개략 블록도들이다.
- 도 27-31은 본 명세의 원리들에 따른, 제1 사용자 및 제2 사용자가 한 문서를 공동으로 저작할 때 제1 저작 애플리케이션의 예시적인 사용자 인터페이스의 변경들을 예시한다.

발명을 실시하기 위한 구체적인 내용

[0010]

다음의 상세한 설명에서는, 본 명세서의 일부를 형성하고 예시로서 특정한 실시예들 또는 예들이 도시되어 있는 첨부 도면들이 참조된다. 본 명세는 일반적으로 컴퓨터 시스템 상의 운영 체제에서 실행하는 애플리케이션 프로그램과 관련하여 실행하는 프로그램 모듈들에 관련하여 설명될 것이지만, 숙련된 당업자들은 본 명세는 다른 프로그램 모듈들과 관련하여 구현될 수도 있다는 것을 인지할 것이다. 여기에 설명된 실시예들은 조합될 수 있고 본 명세의 정신 또는 범위에서 벗어남이 없이 다른 실시예들이 이용될 수 있다. 그러므로 다음의 상세한 설명은 제한적인 점에서 해석되지 않아야 하고, 본 발명의 범위는 첨부된 청구항들 및 그들의 균등물들에 의해 정의된다.

[0011]

본 명세의 실시예들은 다수의 사용자들이 한 문서를 공동으로 저작할 수 있는 환경을 제공한다. 도 1은 본 명세의 발명의 양태들의 예들인 특징들을 갖는 예시의 저작 시스템(100)을 예시한다. 이 저작 시스템(100)은 문서(150)의 마스터 사본을 저장하는 저장 장치(120)를 포함한다. 일 실시예에서, 저장 장치(120)는 컴퓨팅 장치를 포함할 수 있다. 다른 실시예에서, 저장 장치(120)는 하나 이상의 저장 장치들(예를 들면, 컴퓨팅 장치들의

네트워크)을 포함할 수 있다.

- [0012] 저작 시스템(100)은 또한 저장 장치(120)에 통신 연결되어 있는 적어도 하나의 사용자 컴퓨팅 장치(110)를 포함한다. 사용자 컴퓨팅 장치들(110) 각각은 문서(150)의 사용자 사본(155)을 생성하고 그 사용자 사본(155)을 편집하는 것에 의해 문서(150)를 편집할 수 있다. 문서(150)의 사용자 사본들(155)은 사용자 컴퓨팅 장치들(110)이 다른 사용자 컴퓨팅 장치들과 공유될 업데이트들을 저장 장치(120)에 주기적으로 송신하고 다른 사용자 컴퓨팅 장치들로부터의 업데이트들을 저장 장치(120)로부터 주기적으로 획득할 때 동기화된다.
- [0013] 이 용어가 여기에서 사용될 때, 사용자 컴퓨팅 장치(110)는 저작될 문서의 사용자 사본을 그 문서의 마스터 사본으로부터 획득하는 임의의 컴퓨팅 장치를 포함한다. 사용자 컴퓨팅 장치(110)는 저장 장치(120)와 다를 수 있고 또는 저장 장치(120) 상에 구현된 상이한 사용자 계정을 포함할 수 있다. 일 실시예에서, 하나의 문서에 대한 저장 장치(120)로서 기능하는 컴퓨팅 장치는 상이한 문서에 대한 사용자 컴퓨팅 장치(110)로서 기능할 수 있고 그 반대가 될 수도 있다.
- [0014] 도시된 예에서는, 4개의 사용자 컴퓨팅 장치들(110A, 110B, 110C, 및 110D)이 저장 장치(120)에 통신 연결되어 있다. 그러나, 다른 실시예들에서는, 임의의 수의 컴퓨팅 장치들(110)이 저장 장치(120)에 연결될 수 있다. 도시된 예에서는, 각 사용자 컴퓨팅 장치(110A, 110B, 110C, 110D)는 그 사용자 컴퓨팅 장치의 사용자에게 의해 생성된 업데이트들을 저장 장치(120)에 송신할 수 있고 다른 사용자 컴퓨팅 장치들의 사용자에게 의해 생성된 업데이트들을 저장 장치(120)로부터 요청할 수 있다. 일 실시예에서, 저장 장치(120)는 서버 컴퓨팅 장치일 수 있고 사용자 컴퓨팅 장치들(110A, 110B, 110C, 110D)은 클라이언트 컴퓨팅 장치들일 수 있다.
- [0015] 도 2에 도시된 바와 같이, 저장 장치(120) 상에 저장된 문서(150)는 콘텐츠(152) 및 메타데이터(154)를 포함할 수 있다. 사용자 컴퓨팅 장치들(110) 상의 저작 애플리케이션들(130)은 문서(150)의 사용자 사본들(155)의 콘텐츠 및 메타데이터를 처리하고 조작한다. 일부 실시예들에서, 메타데이터(154)는 콘텐츠(152)와 별도로 저장될 수 있다. 예를 들면, 콘텐츠(152)는 문서(150)에 저장될 수 있고 메타데이터(154)는 문서(150)와 별도의 테이블(도 7 참조)에 저장될 수 있다. 그러나, 다른 실시예들에서, 메타데이터(154)는 문서(150) 내에 저장될 수 있다.
- [0016] 일반적으로, 사용자 컴퓨팅 장치들(110)은 메타데이터(154)에의 업데이트들과 별도로 콘텐츠(152)에의 업데이트들을 동기화시킬 수 있다. 일반적으로, 메타데이터 업데이트들(154)은 저장 장치(120) 및 사용자 컴퓨팅 장치들(110) 사이에 자동으로 동기화되는 반면, 각 사용자 컴퓨팅 장치(110)로부터의 콘텐츠 업데이트들(152)은 각각의 사용자의 요청에 의하여 동기화된다.
- [0017] 도 3은 저작 시스템(100)이 사용자 컴퓨팅 장치(110) 상에 저장된 문서의 사본(155)을 저장 장치(120) 상에 저장된 그 문서의 마스터 사본(150)과 동기화시킬 수 있는 예시적인 동기화 프로세스(300)를 예시하는 흐름도이다. 동기화 프로세스(300)는 시작 모듈(302)에서 초기화하고 시작하여 제1 업데이트 동작(304)으로 진행한다.
- [0018] 제1 업데이트 동작(304)은 사용자와의 어떠한 상호 작용도 요구하지 않고 사본(155)의 메타데이터를 문서(150)의 메타데이터와 동기화시킨다. 예를 들면, 제1 업데이트 동작(304)은 주기적인 시간 간격으로 저장 장치(120)와 사용자 컴퓨팅 장치(110) 사이의 메타데이터의 교환을 제공할 수 있다. 일 실시예에서, 제1 업데이트 동작(304)은 몇 초마다 메타데이터의 교환을 제공한다. 그러나, 다른 실시예에서, 제1 업데이트 동작(304)은 보다 짧은 또는 보다 긴 시간 간격으로 발생하는 메타데이터의 교환을 제공할 수 있다.
- [0019] 요청 모듈(306)은 콘텐츠를 동기화시키는 요청이 수신되었는지를 결정한다. 예를 들면, 요청 모듈(306)은 사용자 컴퓨팅 장치들(110) 중 하나의 컴퓨팅 장치의 사용자가 다른 사용자들과 콘텐츠 변경들을 공유하기 위해 요청하였는지를 결정할 수 있다. 다른 실시예에서, 요청 모듈(306)은 사용자 컴퓨팅 장치들(110) 중 하나의 컴퓨팅 장치의 사용자가 다른 사용자들에 의해 행해진 콘텐츠 변경들을 보기 위해 요청하였는지를 결정할 수 있다.
- [0020] 만일 요청 모듈(306)이 콘텐츠를 동기화시키는 요청이 행해지지 않았다고 결정하면, 동기화 프로세스(300)는 제1 업데이트 동작(304)으로 회귀하고 다시 시작한다. 그러나, 만일 요청 모듈(306)이 콘텐츠를 동기화시키는 요청이 수신되었다고 결정하면, 동기화 프로세스(300)는, 저장 장치(120)와 사용자 컴퓨팅 장치(110) 사이의 콘텐츠의 교환을 제공하는, 제2 업데이트 동작(308)으로 진행한다. 동기화 프로세스(300)는 정지 모듈(310)에서 완료하고 종료한다.
- [0021] 용어가 여기에서 사용될 때, 콘텐츠 업데이트들(152)은 문서의 실제적인 콘텐츠에 행해진 임의의 추가, 삭제, 및/또는 수정을 지칭한다. 예를 들면, 워드 프로세싱 문서에 대한 콘텐츠 업데이트들은 추가된 단락들(즉, 또

는 그의 부분들), 삭제된 단락들(즉, 또는 그의 부분), 및/또는 수정된 단락들(즉, 또는 그의 부분들)을 포함할 수 있다. 다른 실시예에서, 프리젠테이션 문서에 대한 콘텐츠 업데이트들은 추가된, 삭제된, 및/또는 수정된 그림들, 텍스트, 애니메이션들, 사운드들, 및 기타 그러한 데이터 개체들을 포함할 수 있다.

[0022] 용어가 여기에서 사용될 때, 메타데이터 업데이트들(154)은 문서의 메타데이터에 행해진 임의의 추가, 삭제, 및/또는 수정을 지칭한다. 문서 메타데이터의 비제한적인 예들은 콘텐츠 잠금들, 프레즌스 정보(presence information), 및 기타 그러한 데이터를 포함한다. 여기에서 논의되는 바와 같이, 콘텐츠 잠금들은 그 잠금을 소유하지 않는 사용자에게 의한 그 잠금 내의 콘텐츠의 편집을 금한다. 프레즌스 정보는 여기에서 더 상세히 설명되는 바와 같이 어느 사용자들이 문서를 편집하려는 의도를 나타내었는지를 표시한다.

[0023] 도 4-9를 참조하면, 콘텐츠 잠금들은 문서의 어느 부분들이 다른 사용자에게 의해 청구(claim)되었는지를 표시하는 것에 의해 편집 충돌들을 금한다. 일부 실시예들에서, 콘텐츠 잠금들은 사용자가 다른 사용자에게 의해 청구된 문서의 부분을 편집하는 것을 방해할 수 있다(즉, 금할 수 있다). 그러나, 다른 실시예들에서, 사용자는 콘텐츠 잠금을 깨고 문서의 그 부분을 편집하기로 결정할 수 있다. 그러한 경우에, 잠금은 잠긴 부분을 편집할 때 충돌이 일어날 수 있다고 사용자에게 경고할 수 있다.

[0024] 예를 들면, 도 4는 5개의 데이터 단위들(410, 420, 430, 440, 450)을 갖는 문서(400)의 개략 블록도이다. 일 실시예에서, 문서(400)는 워드 프로세싱 문서이고 데이터 단위들(410-450)은 텍스트의 단락들이다. 다른 실시예에서, 문서(400)는 프리젠테이션 문서이고 제1 데이터 단위(410)는 표제(title) 또는 주제 제목(subject heading)이고, 제2 데이터 단위(420)는 그림 또는 다른 데이터 개체이고, 나머지 데이터 단위들(430, 440, 450)은 텍스트의 블록들이다.

[0025] 제1 사용자는 문서(400)의 데이터 단위들(410-450) 중 하나 이상의 데이터 단위에 관한 잠금을 생성할 수 있다. 도 4에 도시된 예에서, 제2 데이터 단위(420)는 사선(cross-hatching)에 의해 나타내어진 바와 같이, 제1 사용자에게 의해 잠겼다. 콘텐츠 잠금을 생성함으로써, 제1 사용자는 제1 사용자가 그 데이터 단위(420)를 편집하려고 한다는 것을 나타낼 수 있다. 예를 들면, 사용자는 그 사용자가 그 데이터 단위(420)를 활발히 편집하고 있을 때 잠금을 생성할 수 있다. 다른 실시예에서, 사용자는 잠금을 생성하고 그 후 나중에 데이터 단위(420)를 편집할 수 있다. 전술한 바와 같이, 문서(400)를 관리하는 저작 애플리케이션은 제1 사용자 이외의 사용자가 잠긴 데이터 단위(420)를 편집하는 것을 방해할 수 있다.

[0026] 일반적으로, 제1 사용자에게 의해 생성된 콘텐츠 잠금은 그 잠금의 수명 동안에 문서에 액세스하는 모든 다른 사용자에게 디스플레이된다. 일 실시예에서, 제1 사용자도 콘텐츠 잠금을 볼 수 있다. 잠금들은 상이한 표시 유형들을 이용하여 디스플레이될 수 있다. 예를 들면, 일 실시예에서, 잠긴 데이터 단위의 배경이 채색되거나, 음영이 들어가거나(shade), 패턴될 수 있다(예를 들면, 도 4의 데이터 단위(420) 참조). 다른 실시예에서, 잠긴 단위의 콘텐츠(예를 들면, 텍스트, 그림, 모양, 또는 다른 데이터 개체)가 채색되거나, 그늘지거나, 패턴될 수 있다. 또 다른 실시예에서는, 잠금을 표시하기 위해 잠긴 데이터 단위에 인접하여 박스, 각괄호(bracket), 또는 심벌이 디스플레이될 수 있다.

[0027] 콘텐츠 잠금은 문서 내의 하나 이상의 데이터 단위들 주위에 생성될 수 있다. 일부 실시예들에서, 콘텐츠 잠금은 사용자로부터의 명백한 명령 없이 암시적으로 생성될 수 있다. 예를 들면, 저작 애플리케이션은 사용자가 데이터 단위 내에 커서를 배치하거나, 데이터 단위를 강조(highlight)하거나, 또는 다른 방법으로 데이터 단위를 선택할 때 그 데이터 단위에 관한 콘텐츠 잠금을 생성할 수 있다. 다른 실시예에서, 저작 애플리케이션은 사용자가 데이터 단위를 편집하기 시작할 때 그 데이터 단위에 관한 콘텐츠 잠금을 생성할 수 있다. 다른 실시예에서, 저작 애플리케이션은 사용자가 데이터 단위에 행해진 편집들을 저장할 때 그 데이터 단위에 관한 콘텐츠 잠금을 생성할 수 있다. 일부 실시예들에서, 저작 애플리케이션은 데이터 단위들 중 하나의 데이터 단위만이 사용자에게 의해 청구되는 경우에도 문서 내의 다수의 상호 의존적인 데이터 단위들 주위에 콘텐츠 잠금들을 생성할 수 있다. 다른 실시예들에서, 사용자는 하나 이상의 데이터 단위에 관한 콘텐츠 잠금을 명시적으로 정의할 수 있다.

[0028] 일부 실시예들에서, 콘텐츠 잠금은 문서 내의 추가적인 데이터 단위들을 포함하도록 확장되고 및/또는 문서 내의 하나 이상의 데이터 단위들을 해제하도록 축소될 수 있다. 일 실시예에서, 콘텐츠 잠금은 인접한(contiguous) 데이터 단위들만을 포함하도록 확장될 수 있다. 예를 들면, 도 5에 도시된 바와 같이, 제2 데이터 단위(420) 주위의 콘텐츠 잠금(예를 들면, 사선 참조)은 만일 사용자가 제3 데이터 단위(430)를 선택하거나 편집하면 제3 데이터 단위(430)를 포함하도록 확장할 수 있다. 콘텐츠 잠금 또는 그의 부분들은 일반적으로 콘텐츠 잠금을 생성한 사용자의 재량으로 해제될 수 있다. 일부 실시예들에서, 콘텐츠 잠금을 생성한 사용자는

다른 데이터 단위를 선택하거나 편집하는 것에 의해 콘텐츠 잠금을 해제한다.

- [0029] 일 실시예에서, 사용자는 저작 애플리케이션이 잠금을 해제하기 전에 사용자가 데이터 단위의 편집을 완료한 것을 표시해야 한다. 예를 들면, 저작 애플리케이션은 사용자가 문서를 저장하는 명령을 제공한 후에만 잠금을 해제할 수 있다. 다른 실시예에서, 저작 애플리케이션은 사용자가 도 6의 제5 데이터 단위(450)와 같은 원격 데이터 단위를 선택하거나 편집하기 시작한 후에 잠금을 해제할 수 있다. 다른 실시예에서, 저작 애플리케이션은 인접하지 않은(non-contiguous) 데이터 단위로의 이동 및 완료화 단계(finalization act)(예를 들면, 문서를 저장하는 명령) 후에만 잠금을 해제한다. 다른 실시예들에서, 콘텐츠 잠금은 관리자 또는 다른 사용자에게 의해 명시적으로 해제될 수 있다.
- [0030] 도 7을 참조하면, 잠금 메타데이터는 각종의 상이한 포맷으로 저장될 수 있다. 예를 들면, 도 7의 잠금 메타데이터는 테이블 포맷(500)으로 저장된다. 도 7의 잠금 테이블(500)은 사용자들의 목록을 포함하고, 그들 각각은 그 사용자에게 고유하게 할당되는 사용자 식별자(예를 들면, ID 번호(identification number))로 식별된다. 잠길 데이터 단위들은 문서 내의 각 데이터 단위에 고유하게 할당되는 단위 식별자들(예를 들면, ID 번호들)로 식별된다. 잠금 테이블(500)은 잠길 하나 이상의 데이터 단위의 단위 식별자들을 그 잠금들을 소유하는 사용자들의 사용자 식별자들과 관련시킨다.
- [0031] 예를 들면, 잠금 테이블(500)에서, 데이터 단위들(512 및 514)은 제1 사용자(510)와 관련된다. 따라서, 다른 사용자들은 데이터 단위들(512 및 514)을 편집하는 것이 금지된다. 데이터 단위(522)는 사용자(520)와 관련된다. 따라서, 제1 사용자(510)를 포함하는 다른 사용자들은 데이터 단위(522)를 편집하는 것이 금지된다. 제4 사용자(540)는 문서의 어떤 부분도 잠그지 않았고 따라서 어떤 단위 식별자들과도 관련되지 않는다. 그러나, 다른 실시예들에서, 잠금 메타데이터는 상이한 포맷으로 또는 문서 내에 저장될 수 있다. 예를 들면, 잠금 테이블(500)은 사용자 식별자에 의하는 대신에 단위 식별자에 의해 배열될 수 있다.
- [0032] 프레즌스 메타데이터도 각종의 포맷으로 저장될 수 있다. 예를 들면, 프레즌스 메타데이터는 도 7의 잠금 테이블(500)에 저장될 수 있다. 그러나, 다른 실시예에서, 프레즌스 메타데이터는 별도의 테이블에 또는 상이한 포맷으로 저장될 수 있다. 프레즌스 메타데이터는 현재 문서에 액세스하고 있는 또는 문서의 데이터 단위에 대한 권리를 주장한(예를 들면, 콘텐츠 잠금을 생성한) 각 사용자의 사용자 식별자를 포함한다. 예를 들면, 잠금 테이블(500)과 같은 메타데이터 테이블은 문서의 적어도 하나의 데이터 단위에 대한 권리를 갖는 각 사용자의 사용자 식별자를 저장할 수 있다. 잠금 메타데이터와 같이, 프레즌스 메타데이터는 자동으로 동기화될 수 있다.
- [0033] 도 8 및 9는 콘텐츠 및 잠금들이 다수의 컴퓨팅 장치들 사이에 동기화될 수 있는 예시의 동기화 프로세스들(600, 700)을 예시하는 흐름도들이다. 제1 동기화 프로세스(600)는 저작 애플리케이션을 이용해 제1 사용자에게 의해 행해진 업데이트들을 다른 사용자들과 공유하기 위해 저작 애플리케이션에 의해 구현된다. 제2 동기화 프로세스(700)는 저작 애플리케이션을 이용해 편집되고 있는 문서 내에 다른 사용자들에 의해 행해진 업데이트들을 인스턴스화하기 위해 저작 애플리케이션에 의해 구현된다.
- [0034] 도 8의 제1 동기화 프로세스(600)는 시작 모듈(602)에서 초기화하고 시작하여 수신 동작(604)으로 진행한다. 수신 동작(604)은 문서 내의 하나 이상의 데이터 단위에 대한 편집 동작을 수행하기 위한 명령을 제1 사용자로부터 수신한다. 편집 동작들의 비제한적인 예들은 데이터 단위들(예를 들면, 텍스트, 그림, 사진, 모양, 제목 등) 또는 그의 부분들을 추가하고, 편집하고, 및/또는 삭제하는 것, 데이터 단위들의 포맷팅을 추가하고 및/또는 삭제하는 것, 데이터 단위들의 시퀀스를 재배열하는 것, 및 다른 그러한 동작들을 포함한다.
- [0035] 제1 결정 모듈(606)은 편집 동작이 수행될 데이터 단위가 잠겨 있는지를 확인한다. 예를 들면, 제1 결정 모듈(606)은 도 7의 잠금 테이블(500)과 같은 잠금 테이블에 액세스하여 그 데이터 단위에 할당된 고유 식별자가 임의의 사용자와 관련되어 있는지를 결정할 수 있다. 다른 실시예에서, 제1 결정 모듈(606)은 다른 방법으로 그 데이터 단위가 잠겨 있는지를 결정할 수 있다.
- [0036] 만일 제1 결정 모듈(606)이 그 데이터 단위가 잠겨 있지 않다고 결정하면, 생성 동작(608)은 그 데이터 단위에 대한 콘텐츠 잠금을 생성한다. 예를 들면, 일 실시예에서, 생성 동작(608)은 그 데이터 단위의 단위 식별자와 제1 사용자의 사용자 식별자를 관련시키는 항목을 도 7의 잠금 테이블(500)에 추가할 수 있다. 송신 동작(610)은 새로이 생성된 잠금의 존재를 나타내는 잠금 메타데이터를 그 문서의 마스터 사본을 저장하는 장치에 송신한다. 예를 들면, 송신 동작(610)은 그 잠금 메타데이터를 도 1의 컴퓨팅 장치(110A)와 같은 사용자 컴퓨팅 장치로부터 저장 장치(120)와 같은 저장 장치로 송신할 수 있다. 일 실시예에서, 잠금 메타데이터는 그 잠금 메타데이터에의 증분 편집들(incremental edits)(예를 들면, 델타 상태들)을 기술하는 명령을 포함한다. 구현 동

작(614)은 그 데이터 단위에 대한 편집 동작을 수행한다.

- [0037] 대안적으로, 만일 제1 결정 모듈(606)이 그 데이터 단위가 잠겨 있다고 결정하면, 제2 결정 모듈(612)은 그 잠금에 대한 소유권 정보를 획득한다. 일 실시예에서, 만일 제2 결정 모듈(612)이 그 잠금이 다른 사용자에게 의해 소유된다고 결정하면, 동기화 프로세스(600)는 편집 동작을 수행하지 않고 정지 모듈(620)에서 완료하고 종료한다. (도시되지 않은) 다른 실시예에서, 동기화 프로세스(600)는 제1 사용자가 잠금을 깨고 생성 동작(608)으로 진행하기로 결정하는 것을 가능하게 할 수 있다. 그러나, 만일 제2 결정 모듈(612)이 그 잠금이 제1 사용자에게 의해 소유된다고 결정하면, 동기화 프로세스(600)는 전술한 구현 동작(614)으로 진행한다. (도시되지 않은) 다른 실시예에서, 동기화 프로세스(600)는 임의의 데이터 단위들에 대한 잠금이 해제되어야 하는지를 결정할 수 있다.
- [0038] 제3 결정 동작(616)은 제1 사용자가 편집 동작을 구현하는 것으로부터 생긴 데이터 단위들의 콘텐츠 변경들을 공유하기 위한 명령을 (명시적으로 또는 암시적으로) 제공하였는지를 결정한다. 예를 들면, 제3 결정 동작(616)은 제1 사용자가 문서를 저장하는 명령을 제공하였는지를 결정할 수 있다. 다른 실시예에서, 제3 결정 동작(616)은 제1 사용자가 "변경 공유(share changes)" 버튼 또는 다른 그러한 표시를 선택하였는지를 결정할 수 있다. 다른 실시예에서, 제3 결정 동작(616)은 애플리케이션에 의해 자동 저장(auto-save) 특징이 구현된 것을 결정할 수 있다. 만일 제3 결정 동작(616)이 공유하는 명령이 제공되지 않았다고 결정하면, 동기화 프로세스(600)는 수신 동작(604)으로 회귀하고 다시 시작한다.
- [0039] 그러나, 만일 결정 동작(616)이 공유하는 명령이 제공되었다고 결정하면, 동기화 프로세스(600)는 문서의 마스터 사본을 저장하는 장치에 콘텐츠 업데이트들을 송신하는 송신 동작(618)으로 진행한다. 콘텐츠 업데이트들은 편집 동작에 의해 문서의 콘텐츠에 행해진 변경들을 나타낸다. 예를 들면, 송신 동작(618)은 콘텐츠 업데이트들을 도 1의 사용자 컴퓨팅 장치(110A)로부터 저장 장치(120)로 송신할 수 있다. 일 실시예에서, 콘텐츠 업데이트들은 문서의 새로이 생성된 버전을 포함한다. 다른 실시예에서, 콘텐츠 업데이트들은 문서의 현재의 사용자 사본과 문서의 마스터 사본 사이의 콘텐츠에의 증분 편집들(예를 들면, 델타 상태들)을 나타낸다. 동기화 프로세스(600)는 전술한 바와 같이 정지 모듈(620)에서 완료하고 종료한다.
- [0040] 도 9의 제2 동기화 프로세스(700)는 시작 모듈(702)에서 초기화하고 시작하여 수신 동작(704)으로 진행한다. 수신 동작(704)은 다른 사용자들에 의해 문서에 행해진 변경들에 기초한 업데이트를 수신한다. 그 업데이트는 문서의 콘텐츠 및/또는 메타데이터에의 변경들을 포함할 수 있다. 일 실시예에서, 수신 동작(704)은 문서의 마스터 사본을 저장하는 장치를 폴링(poll)하여 업데이트들을 요청할 수 있다. 다른 실시예에서, 마스터 사본을 저장하는 장치는 저작 애플리케이션에 변경들을 푸시(push)할 수 있다.
- [0041] 제1 결정 모듈(706)은 업데이트들이 문서의 메타데이터에의 임의의 변경들을 포함하는지를 결정한다. 예를 들면, 제1 결정 모듈(706)은 임의의 다른 사용자가 새로운 잠금을 설정하였는지 데이터 단위에 대한 잠금을 해제하였는지를 결정할 수 있다. 다른 실시예에서, 제1 결정 모듈(706)은 다른 사용자가 문서에 액세스하였는지를 결정할 수 있다.
- [0042] 만일 제1 결정 모듈(706)이 메타데이터 업데이트가 수신되었다고 결정하면, 구현 동작(708)은 수신된 메타데이터 업데이트들을 자동으로 인스턴스화한다. 예를 들면, 구현 동작(708)은 저작 애플리케이션으로 하여금 메타데이터 업데이트에 기초하여 도 7의 잠금 테이블(500)과 같은 잠금 테이블을 업데이트하게 할 수 있다. 그러나, 만일 제1 결정 모듈(706)이 메타데이터 업데이트가 수신되지 않았다고 결정하면, 제2 동기화 프로세스(700)는 제2 결정 모듈(710)로 진행한다.
- [0043] 제2 결정 모듈(710)은 업데이트가 문서의 콘텐츠에의 임의의 변경들을 포함하는지를 결정한다. 예를 들면, 제2 결정 모듈(710)은 임의의 데이터 단위들이 추가, 삭제, 수정, 또는 이동되었는지를 결정할 수 있다. 만일 제2 결정 모듈(710)이 콘텐츠 업데이트가 수신되지 않았다고 결정하면, 제2 동기화 프로세스(700)는 수신 동작(704)으로 회귀하여 다시 시작한다. 그러나, 만일 제2 결정 모듈(710)이 콘텐츠 업데이트가 수신되었다고 결정하면, 제2 동기화 프로세스(700)는 표시 동작(712)으로 진행한다.
- [0044] 표시 동작(712)은 콘텐츠 업데이트가 보기(viewing)를 위해 이용 가능하다는 것을 제1 사용자에게 경고한다. 예를 들면, 표시 동작(712)은 콘텐츠 업데이트의 이용 가능성을 표시하는 그래픽, 텍스트, 박스, 수치 표시, 또는 다른 메시지를 디스플레이할 수 있다. 일 실시예에서, 표시 동작(712)은 몇 개의 업데이트들이 이용 가능한지를 나타내는 정보를 사용자에게 제공한다. 업데이트들에 관한 다른 정보(예를 들면, 업데이트를 만든 사용자, 타임스탬프 등)가 또한 표시될 수 있다.

- [0045] 제3 결정 모듈(714)은 저작 애플리케이션이 콘텐츠 업데이트들을 보기 위한 및/또는 인스턴스화하기 위한 명령을 제1 사용자로부터 (명시적으로 또는 암시적으로) 수신하였는지를 결정한다. 예를 들면, 제3 결정 모듈(714)은 사용자가 저작 애플리케이션의 뷰 옵션(view option)을 (예를 들면, 버튼, 메뉴, 또는 다른 인터페이스 도구를 통해) 선택하였는지를 결정할 수 있다. 다른 실시예에서, 제3 결정 모듈(714)은 사용자가 자동-인스턴스화 옵션(auto-instantiate option)을 선택한 것을 결정할 수 있다. 만일 제3 결정 모듈(714)이 저작 애플리케이션이 제1 사용자로부터 그러한 명령을 수신하지 않았다고 결정하면, 제2 동기화 프로세스(700)는 수신 동작(704)으로 회귀하여 다시 시작한다.
- [0046] 만일 제3 결정 모듈(714)이 저작 애플리케이션이 제1 사용자로부터 명령을 수신하였다고 결정하면, 인스턴스화 동작(716)은 콘텐츠 업데이트들을 제1 사용자에게 디스플레이한다. 예를 들면, 일 실시예에서, 인스턴스화 동작(716)은 이용 가능한 콘텐츠 업데이트들을 문서의 제1 사용자의 사본에 병합할 수 있다. 다른 실시예에서, 인스턴스화 동작(716)은 어느 콘텐츠가 변화하였는지를 나타내기 위해 문서의 제1 사용자의 사본에 주석을 달 수 있다. 추가적인 주석들은 어느 사용자들이 어느 변경들을 행하였는지를 나타낼 수 있다. 일 실시예에서, 인스턴스화 동작(716)은 사용자 사본과 업데이트들 사이의 충돌 해결을 수행한다. 제2 동기화 프로세스(700)는 정지 모듈(718)에서 완료하고 종료한다.
- [0047] 도 10-12는 문서의 사용자 사본과 마스터 사본 사이의 동기화가 사용자 컴퓨팅 장치에 의해 어떻게 구현되는지를 보다 상세히 제공한다. 도 10은 저작될 문서의 마스터 사본이 저장될 저장 장치(820)를 포함하는 저작 시스템(800)의 개략 블록도이다. 저작 시스템(800)은 또한 저장 장치(820)에 통신 연결된 적어도 하나의 사용자 컴퓨팅 장치(810)를 포함한다.
- [0048] 사용자 컴퓨팅 장치(810)는 사용자가 저작될 문서를 생성하고 및/또는 조작할 수 있는 저작 환경을 제공하도록 구성된 저작 애플리케이션(812)을 포함한다. 사용자 컴퓨팅 장치(810)는 또한 캐시(814), 층 개체(layer object)("LO")(816), 및 동기화 매니저("sync manager")(818)를 포함한다. 캐시(814)는 저작될 문서의 사용자 사본을 저장한다. 캐시(814)는 또한 문서와 관련된, 잠금 및 프레즌스 메타데이터를 포함하는, 메타데이터를 저장한다. 문서의 콘텐츠 및 메타데이터에의 업데이트들도 캐시(814)에 저장될 수 있다.
- [0049] 층 개체(816)는 저작 애플리케이션(812)과 캐시(814) 사이의 인터페이스를 제공한다. 층 개체(816)는 또한 저작 애플리케이션(812)과 동기화 매니저(818) 사이의 인터페이스를 제공한다. 동기화 매니저(818)는 저장 장치(820)와 통신하고 저장 장치(820)와 캐시(814) 사이의 인터페이스를 제공한다. 예를 들면, 동기화 매니저(818)는 저장 장치(820) 및 캐시(814)에 업데이트들을 송신하고 그것들로부터 업데이트들을 획득할 수 있다.
- [0050] 일반적으로, 본 명세의 원리들에 따른 발명의 양태들의 예들인 특징들을 갖는 저작 환경은 사용자 컴퓨팅 장치(예를 들면, 퍼스널 컴퓨터, 서버 컴퓨터, 노트북 컴퓨터, PDA, 스마트폰, 또는 임의의 다른 그러한 컴퓨팅 장치) 상에 구현될 수 있다. 저작 환경을 구현하도록 구성된 사용자 컴퓨팅 시스템(900)의 비제한적인 실시예가 도 11에 관련하여 여기에 설명된다.
- [0051] 도 11에서, 본 명세의 원리들을 구현하기 위한 예시적인 컴퓨팅 시스템(900)은 사용자 컴퓨팅 장치(910)와 같은 사용자 컴퓨팅 장치를 포함한다. 기본 구성에서, 사용자 컴퓨팅 장치(910)는 전형적으로 시스템 메모리(920)에 저장된 애플리케이션들 및 프로그램들을 실행하기 위한 적어도 하나의 처리 장치(915)를 포함한다. 컴퓨팅 장치(910)의 정확한 구성 및 유형에 따라서, 시스템 메모리(920)는 RAM, ROM, EEPROM, 플래시 메모리, CD-ROM, DVD(digital versatile disks) 또는 다른 광학 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 다른 메모리 기술을 포함할 수 있지만, 이에 제한되는 것은 아니다.
- [0052] 시스템 메모리(920)는 전형적으로 컴퓨팅 장치(910)의 동작을 제어하기에 적합한, 워싱턴주, 레드먼드의 마이크로소프트사로부터의 WINDOWS® 운영 체제와 같은 운영 체제(922)를 저장한다. 시스템 메모리(920)는 또한 문서의 사용자 사본(927)이 저장될 수 있는 문서 캐시(926)를 포함할 수 있다. 문서의 메타데이터(929)도 사용자 캐시(926)에 저장될 수 있다.
- [0053] 시스템 메모리(920)는 또한 문서들을 생성 및 편집하기 위한 저작 애플리케이션(924)과 같은 하나 이상의 소프트웨어 애플리케이션을 저장할 수 있다. 본 명세의 원리들에 따른 문서들을 저작하기에 적합한 저작 애플리케이션(924)의 하나의 비제한적인 예는 워싱턴주, 레드먼드의 마이크로소프트사로부터의 MICROSOFT® OFFICE WORD 저작 소프트웨어이다. 저작 애플리케이션들의 다른 비제한적인 예들은 POWERPOINT® 프리젠테이션 소프트웨어 및 VISIO® 드로잉 및 다이어그램 소프트웨어를 포함하고, 이것들도 워싱턴주, 레드먼드의 마이크로소프트사로부터 출시된 것들이다.

- [0054] 컴퓨팅 장치(910)는 또한 데이터를 입력하고 조작하기 위한, 키보드, 마우스, 펜, 음성 입력 장치, 터치 입력 장치 등의, 입력 장치(들)(930)를 가질 수 있다. 디스플레이 화면, 스피커, 프린터 등의, 출력 장치(들)(935)도 포함될 수 있다. 이들 출력 장치들(935)은 당업계에 잘 알려져 있어 여기서 장황하게 설명될 필요가 없다.
- [0055] 컴퓨팅 장치(910)는 또한 그 장치(910)가 분산 컴퓨팅 환경에서의 네트워크(예를 들면, 인트라넷 또는 인터넷)를 통하여 다른 컴퓨팅 장치들, 예를 들면, 도 10의 저장 장치(820)와 통신할 수 있게 하는 통신 접속들(940)을 포함할 수 있다. 예로서, 통신 장치 매체들(940)은 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체들, 및 음향, RF, 적외선 및 기타 무선 매체들과 같은 무선 매체들을 포함하지만, 이에 제한되는 것은 아니다.
- [0056] 도 12는 사용자 컴퓨팅 장치 상의 저작 애플리케이션에 의해 구현되는 동기화 사이클을 예시하는 저작 시스템(1000)의 개략 블록도이다. 저작 시스템(1000)은 저작될 문서가 저장될 저장 장치(1020) 및 사용자 컴퓨팅 장치(1010)를 포함한다. 사용자 컴퓨팅 장치(1010)는 저작 애플리케이션(1012), 캐시(1014), 층 개체(1016), 및 동기화 매니저(1018)를 포함한다. 사용자 컴퓨팅 장치(1010)의 캐시(1014)는 저작될 문서의 기초 사본(base copy)(1013), 작업 사본(working copy)(1015), 업로드 사본(1017), 및 다운로드 사본(1019)을 저장하도록 구성된다. 캐시(1014)는 또한 문서의 메타데이터(1011)를 저장하도록 구성된다.
- [0057] 도 13은 사용자 컴퓨팅 장치(1010)와 같은 사용자 컴퓨팅 시스템이 문서를 저작하는 동안에 업데이트들을 송신 및 수신할 수 있는 예시적인 동기화 프로세스(1100)에 대한 동작 흐름을 예시하는 흐름도이다. 동기화 프로세스(1100)는 시작 모듈(1102)에서 초기화하고 시작하여 획득 동작(1104)으로 진행한다. 획득 동작(1104)은 사용자 컴퓨팅 장치(1010)의 동기화 매니저(1018)로 하여금 저장 장치(1020)로부터 저작될 문서의 다운로드 사본(1017)을 획득하게 한다. 문서의 다운로드 사본(1017)은 저장 장치(1020)로부터 문서가 획득되는 때의 문서의 마스터 사본의 상태를 반영한다. 예를 들면, 획득 동작(1104)은 동기화 매니저(1018)로 하여금 저장 장치(1020)로부터 문서의 마스터 사본의 가장 최근 버전의 사본을 획득하게 할 수 있다.
- [0058] 생성 동작(1106)은 다운로드 사본(1017)에 기초하여 문서의 기초 사본(1013) 및 작업 사본(1015)을 생성한다. 문서의 작업 사본(1015)은 사용자가 문서를 편집할 수 있도록 저작 애플리케이션(1012)에 의해 조작될 수 있다. 따라서, 작업 사본(1015)은 저작 애플리케이션(1012)의 사용자에게 의해 편집된 문서의 현재 상태를 반영한다. 기초 사본(1013)은 사용자가 문서를 편집하기 시작하기 바로 전 시점의 문서의 상태를 반영한다. 일 실시예에서, 층 개체(1016)는 문서의 기초 사본(1013) 및 작업 사본(1015)을 생성하고 작업 사본(1015)을 저작 애플리케이션(1012)에 제공한다.
- [0059] 다른 사용자들에 의해 생성된 콘텐츠 업데이트들을 획득하기 위해, 폴링 동작(poll operation)(1108)은 동기화 매니저(1018)로 하여금 주기적으로 저장 장치(1020)와 접촉하여 저작되고 있는 문서의 업데이트된 다운로드 사본(1017)을 획득하게 한다. 다운로드 사본(1017)은 다운로드 시점의 문서의 마스터 사본의 상태를 반영한다. 마스터 사본은 저장 장치(1020) 상에 저장되고 기초 사본(1013)이 생성된 이후 다른 사용자들에 의해 행해진 변경들을 포함할 수 있다. 동기화 매니저(1018)는 문서의 임의의 업데이트된 다운로드 사본(1017)을 캐시(1014)에 저장한다.
- [0060] 편집 동작(1110)은 문서에 대한 편집 동작을 수행하기 위해 사용자에게 의해 제공되는 임의의 명령을 저작 애플리케이션(1012)에 수신한다. 애플리케이션(1012)은 문서(1015)의 작업 사본에 편집들을 행한다. 저작 애플리케이션(1012)은 또한 편집 동작을 구현할 때 적절한 메타데이터 변경들을 행한다.
- [0061] 제1 결정 모듈(1112)에서, 층 개체(1016)는 캐시(1014)를 확인하여 폴링 동작(1108) 동안에 임의의 업데이트들이 수신되었는지(즉, 캐시(1014)에서 업데이트된 다운로드 사본(1017)이 이용 가능한지)를 결정한다. 일 실시예에서, 제1 결정 모듈(1112)은 사용자가 문서의 작업 사본을 편집하였는지 여부에 관계없이 주기적인 간격으로 수행된다. 상이한 실시예들에서, 제1 결정 모듈(1112)은, 예를 들면, 몇 초, 밀리초, 분마다 수행될 수 있다.
- [0062] 만일 층 개체(1016)가 새로운 다운로드 사본(1017)이 이용 가능하지 않다고 결정하면, 동기화 프로세스(1100)는 아래에 더 상세히 논의되는 제2 결정 모듈(1116)로 진행한다. 그러나, 만일 층 개체(1016)가 새로운 다운로드 사본(1017)이 이용 가능하다고 결정하면, 동기화 프로세스(1100)는 동기화 동작(1114)으로 진행한다. 일 실시예에서, 동기화 동작(1114)은 저작 애플리케이션(1012)의 사용자와의 어떤 상호 작용도 요구하지 않고 다운로드 사본(1017)에서 발견되는 임의의 메타데이터 업데이트들을 자동으로 인스턴스화한다. 예를 들면, 층 개체(1016)는 다운로드 사본(1017)에 반영된 임의의 잠금 메타데이터를 문서의 작업 사본(1015)에 자동으로 인스턴스화할 수 있다.

- [0063] 일부 실시예들에서, 동기화 동작(1114)은 다운로드 사본(1017)으로부터의 임의의 콘텐츠 업데이트들을 작업 사본(1015)에 자동으로 인스턴스화하지 않는다. 그보다는, 동기화 동작(1114)은 저작 애플리케이션(1012)의 사용자가 콘텐츠 업데이트들을 통합하는 명령을 제공할 때에만 콘텐츠 업데이트들을 인스턴스화한다. 그러나, 다른 실시예들에서, 동기화 동작(1114)은 콘텐츠 업데이트들을 자동으로 인스턴스화한다. 일 실시예에서, 동기화 동작(1114)은 층 개체(1016)로 하여금 통합하는 명령이 제공될 때 업데이트된 다운로드 사본(1017)을 작업 사본(1015)과 병합하게 한다. 일 실시예에서, 동기화 동작(1114)은 도 9에 도시된 제2 동기화 프로세스(700)와 동일하다. 그러나, 다른 실시예들에서는, 업데이트들을 문서의 작업 사본(1015)과 통합하기 위해 다른 동기화 프로세스들이 이용될 수 있다. 동기화 프로세스(1100)는 동기화 동작(1114)으로부터 제2 결정 모듈(1116)로 진행된다.
- [0064] 제2 결정 모듈(1116)은 사용자가 편집 동작(1110) 동안에 문서의 작업 사본(1015)에 또는 문서의 메타데이터(1011)에 임의의 변경들을 행하였는지를 결정한다. 만일 제2 결정 모듈(1116)이 어떠한 편집도(즉, 콘텐츠도 메타데이터도) 행해지지 않았다고 결정하면, 동기화 프로세스(1100)는 폴링 동작(1108)으로 회귀하여 다시 시작한다. 그러나, 만일 제2 결정 모듈(1116)이 사용자가 콘텐츠 편집들(예를 들면, 문서의 데이터 단위를 수정하는 것) 및/또는 메타데이터 편집들(예를 들면, 문서의 데이터 단위를 잠그는 것)을 행하였다고 결정하면, 공유 동작(1118)은 그 편집들을 저장 장치(1020)와 동기화시킨다.
- [0065] 공유 동작(1118)은 사용자 컴퓨팅 장치(1010)의 동기화 매니저(1018)로 하여금 저작 애플리케이션(1012)의 사용자로부터의 어떠한 상호 작용도 요구하지 않고 문서의 메타데이터에 임의의 변경들을 저장 장치(1020)에 자동으로 송신하게 한다. 일 실시예에서, 공유 동작(1118)은 동기화 매니저(1018)로 하여금 콘텐츠 업데이트들을 저장 장치(1020)에 자동으로 송신하게 하지 않는다. 그보다는, 동기화 매니저(1018)는 층 개체(1016)에 의해 그렇게 하도록 지시를 받는 경우에만 콘텐츠 업데이트들을 저장 장치(1020)에 송신한다. 메타데이터만을 자동으로 푸시하는 것은 업데이트 효율을 증가시킴으로써 사용자 경험을 향상시키고, 네트워크 대역폭 사용을 감소시키고, 콘텐츠의 자동 동기화(auto-syncing)를 요구하지 않고 충돌 금지를 제공할 수 있다. 그러나, 다른 실시예들에서, 공유 동작(1118)은 메타데이터뿐만 아니라 콘텐츠도 자동으로 푸시한다. 일 실시예에서, 공유 동작(1118)은 도 8의 제1 동기화 프로세스(600)와 동일하다. 그러나, 다른 실시예들에서는 다른 동기화 프로세스들이 이용될 수 있다.
- [0066] 예를 들면, 사용자가 사용자에게 의해 행해진 콘텐츠 수정들을 공유하기를 원할 경우, 사용자는 문서의 마스터 사본과의 통합을 위해 저장 장치(1020)에 그 수정들을 제출하도록 저작 애플리케이션(1012)에 지시할 것이다. 일 실시예에서, 저작 애플리케이션(1012)은 문서의 업로드 사본(1019)을 생성하도록 층 개체(1016)에 지시할 것이다. 일반적으로, 층 개체(1016)는 작업 사본(1015)에 기초하여 업로드 사본(1019)을 생성한다. 동기화 매니저(1018)는 업로드 사본(1019)을 저장 장치(1020)에 푸시하려고 시도한다. 동기화 프로세스(1100)는 정지 모듈(1120)에서 완료하고 종료한다.
- [0067] 일부 실시예들에서, 공유 동작(1118)은 저장 장치(1020)로부터 수신된 모든 이전의 콘텐츠 업데이트들이 문서의 작업 사본(1015)에 인스턴스화될 때까지 콘텐츠 업데이트들을 저장 장치(1020)에 송신할 수 없다. 그러한 실시예들에서, 업로드 사본(1019)은 문서의 작업 사본(1015)을 가장 최근에 획득된 다운로드 사본(1017)과 병합하는 것에 의해 생성될 수 있다. 일 실시예에서, 동기화 매니저(1018)는 층 개체(1016)가 업로드 사본(1019)을 생성하려고 시도할 때 업데이트된 다운로드 사본(1017)을 위해 저장 장치(1020)를 폴링한다.
- [0068] 만일 동기화 매니저(1018)가 업로드 사본(1019)을 저장 장치(1020)에 푸시하는 데 실패하면(예를 들면, 업로드 사본(1019)이 마스터 사본에 인스턴스화된 가장 최근의 변경들을 반영하지 않는다면), 동기화 매니저(1018)는 수정된 업데이트 문서(1019)로 재시도한다. 예를 들면, 동기화 매니저(1018)는 저장 장치(1020)로부터 새로운 다운로드 사본(1017)을 획득할 수 있고 층 개체(1016)는 작업 사본 및 새로운 다운로드 사본(1017)으로부터 새로운 업로드 사본을 생성할 수 있다. 일 실시예에서, 동기화 매니저(1018)는 성공할 때까지 계속해서 업로드 사본(1019)을 수정하고 저장 장치(1020)에 푸시할 것이다.
- [0069] 도 14-31을 참조하여, 몇몇 예시의 애플리케이션들을 통해 나아감으로써 본 명세의 원리들이 더 잘 이해될 수 있다. 도 14-26은 제1 사용자 및 제2 사용자가 한 문서를 동시에 편집하는 공동 저작 세션 동안에 저장 장치의 캐시, 제1 사용자 컴퓨팅 장치의 캐시(1214), 및 제2 사용자 컴퓨팅 장치의 캐시(1234)에 행해지는 변경들을 나타낸다. 그러나, 다른 실시예들에서, 제1 및 제2 사용자들은 동일한 프로세스들을 이용하여 상이한 시간들에서 문서를 편집할 수 있다.
- [0070] 도 14에서, 제1 사용자는 문서를 생성하고 그 문서의 초안을 마스터 사본(1250)으로서 저장 장치에 저장하였다.

대안적으로, 제1 사용자는 기존의 문서의 마스터 사본(1250)에 액세스하였을 수 있다. 제2 사용자는 아직 그 문서에 액세스하지 않았다.

[0071] 문서의 마스터 사본(1250)은 대응하는 마스터 잠금 테이블을 갖는다. 편의상, 도 14-26에 도시된 예에서, 마스터 잠금 테이블은 마스터 사본(1250)의 일부로서 디스플레이된다. 그러나, 다른 실시예들에서, 마스터 잠금 테이블은 문서의 마스터 사본(1250)과 별도로 저장될 수 있다. 유사하게, 이해의 용이함을 위해, 사용자 캐시(1214, 1234) 상에 저장된 문서의 각 사본(예를 들면, 기초 사본, 작업 사본, 다운로드 사본, 및 업로드 사본)과 관련된 잠금들은 그 사본의 일부로서 도 14-26에 도시되어 있다. 그러나, 다른 실시예들에서, 메타데이터는 각 문서 사본에 대하여 또는 각 사용자에게 대하여 별도로 저장될 수 있다.

[0072] 문서의 마스터 사본(1250)은 텍스트 "Hello World"를 포함하는 제1 데이터 단위 및 그 제1 데이터 단위에 대한 잠금 E1을 포함한다. 잠금 E1은 제1 사용자에게 의해 소유된다(즉, 제1 사용자에게 할당된다). 마스터 사본(1250)의 콘텐츠 및 메타데이터는 제1 사용자의 캐시(1214)에 저장된 기초 사본 및 다운로드 사본에 반영된다. 제1 사용자는 문서의 초안을 저장 장치에 송신하였기 때문에, 캐시(1214)에 저장된 업로드 사본도 마스터 사본(1250)의 콘텐츠 및 메타데이터를 반영한다.

[0073] 초안을 저장 장치에 송신한 이후, 제1 사용자는 텍스트 "Testing"을 포함하는 제2 데이터 단위를 포함하도록 문서의 작업 사본을 편집하였다. 그 제2 데이터 단위에 관하여 제2 잠금 E2가 추가되었다. 일 실시예에서, 제2 잠금 E2는 사용자에게 할당된 새로운 잠금이다. 다른 실시예에서, 제2 잠금 E2는 제1 잠금 E1의 확장이다. 예를 들면, 제1 사용자의 캐시(1214)에 저장된 메타데이터 테이블에서 제1 사용자에게 새로운 단위 식별자가 할당될 수 있다. 일 실시예에서, 제1 사용자는 새로운 데이터 단위(예를 들면, 단락)를 시작하기 위해 커서를 이동시키고 "Testing"을 타이핑하는 것에 의해 제2 잠금 E2를 생성한다. 제1 사용자가 (예를 들면, 커서를 이동시킨 후에 저장함으로써) 제1 데이터 단위에 대한 편집이 최종이라는 것을 아직 나타내지 않았기 때문에 제1 잠금 E1은 아직 해제되지 않는다.

[0074] 도 15에서, 제2 사용자는 저장 장치 상에 저장된 문서의 마스터 사본(1250)에 액세스한다. 제2 사용자는 문서의 다운로드 사본을 획득하고 그 다운로드 사본을 제2 사용자의 캐시(1234)에 저장한다. 다운로드 사본은 마스터 사본(1250)의 콘텐츠 및 메타데이터를 포함한다. 다운로드 사본은 제1 사용자의 작업 사본에 포함된 가장 최근의 콘텐츠 및 메타데이터를 포함하지 않는다. 제2 사용자의 총 개체는 다운로드 사본에 기초하여 문서의 기초 사본 및 작업 사본을 생성한다. 제2 사용자에게 의해 이용되는 저작 애플리케이션은 작업 사본의 콘텐츠 및 임의의 잠금들을 제2 사용자에게 디스플레이한다.

[0075] 일부 실시예들에서, 제2 사용자는 공동 저작 세션(coauthoring session)이 완전히 확립되기 전(즉, 제1 및 제2 사용자 컴퓨팅 장치들이 서로의 존재를 알기 전)에도 문서를 편집할 수 있다. 예를 들면, 제2 사용자는 제1 및 제2 사용자 컴퓨팅 장치들이 인사(즉, 악수)를 교환하는 동안에 문서의 제2 작업 사본을 편집하도록 허용된다. 전형적으로, 메타데이터는 제1 및 제2 컴퓨팅 장치들이 서로의 존재를 알기 전에 자동으로 교환되지 않는다.

[0076] 도 16에서, 제1 및 제2 컴퓨팅 장치들은 각각 서로의 존재를 알아챘다. 따라서, 제1 및 제2 컴퓨팅 장치들 각각은 서로에게 자동으로 메타데이터 업데이트들을 송신하기 시작하였다. 도 16에 도시된 바와 같이, 제1 저작 애플리케이션은 제1 사용자의 잠금 테이블로부터의 업데이트된 잠금 메타데이터(예를 들면, 제2 잠금)를 (예를 들면, 총 개체 및 동기화 매니저를 통해) 저장 장치(1220)에 제공한다. 그러나, 그 저작 애플리케이션은, 콘텐츠 업데이트들을 공유하는 어떠한 명령도 제1 사용자로부터 수신되지 않았기 때문에, 업데이트된 콘텐츠 정보를 저장 장치(1220)에 제공하지 않는다.

[0077] 도 17에서, 저장 장치는 업데이트된 잠금 메타데이터(예를 들면, 제2 잠금 E2)를 마스터 사본(1250)의 마스터 메타데이터에 통합하였다. 잠금 메타데이터는 아직 제2 사용자 컴퓨팅 장치에 의해 획득되지 않았다. 도 18에서, 제2 사용자 컴퓨팅 장치는 저장 장치를 폴링하여 마스터 사본(1250)의 새로운 다운로드 사본을 획득하였다. 새로운 다운로드 사본은 또한 업로드 사본에 기초하여 제1 캐시(1214) 상에 (예를 들면, 총 개체를 통해) 생성되었다. 다른 실시예에서, 제1 사용자 컴퓨팅 장치는 또한 저장 장치를 폴링하여 마스터 사본(1250)의 새로운 다운로드 사본을 획득하였다. 양쪽의 새로운 다운로드 사본들은 제1 사용자의 작업 사본으로부터의 업데이트된 잠금 정보(예를 들면, 잠금 E2)를 반영한다.

[0078] 도 19에서, 제2 컴퓨팅 장치는 새로운 다운로드 사본에 기초하여 새로운 작업 사본을 (예를 들면, 총 개체를 통해) 생성한다. 따라서, 제2 컴퓨팅 장치의 새로운 작업 사본은 제2 데이터 단위에 대한 제2 잠금 E2를 포함한다. 새로운 작업 사본은, 그러나, 제2 데이터 단위 자체를 포함하지는 않는다. 일 실시예에서, 제2 사용자 컴

퓨팅 장치의 작업 사본은 공백/빈 데이터 단위를 디스플레이하고 공백/빈 데이터 단위가 잠겨 있는 것을 표시한다. 다른 실시예에서, 작업 사본은 제2 데이터 단위를 전혀 디스플레이하지 않고 잠금 E2는 제2 데이터 단위가 제2 사용자에게 보일 때까지 제2 사용자에게 표시되지 않는다.

[0079] 도 20에서, 제2 사용자는 문서에 제3 데이터 단위를 추가하고 제3 데이터 단위의 주위에(예를 들면, 저작 애플리케이션의 메모리에) 제3 잠금 E3를 추가한다. 제3 데이터 단위는 텍스트 "Goodbye World"를 포함한다. 제2 저작 애플리케이션은 제3 데이터 단위 및 제3 잠금 E3를 포함하는 문서의 작업 사본을(예를 들면, 층 개체를 통해) 생성한다. 제2 저작 애플리케이션은 또한 제3 잠금 E3를 저장 장치에(예를 들면, 동기화 매니저를 통해) 자동으로 제공한다. 그러나, 제2 저작 애플리케이션은 제3 데이터 유닛을 저장 장치에 제공하지 않는다.

[0080] 또한 도 20에서, 제1 사용자는 제1 사용자가 제1 데이터 단위의 편집을 완료한 것을 표시한 후에 제1 잠금 E1을 해제한다. 예를 들면, 제1 사용자는 제1 사용자의 커서가 제2 데이터 단위 내에 배치될 때 문서를 저장하는 명령을 제공할 수 있다. 일 실시예에서, 제2 데이터 단위는 제1 데이터 단위로부터 원격으로 위치할 수 있다. 도시된 예에서, 제1 사용자에 의해 제공되는 완료화 표시는 제1 사용자가 제1 사용자에 의해 행해진 콘텐츠 변경들을 공유하기를 원한다는 것을 표시하지 않는다. 따라서, 제1 저작 애플리케이션은 저장 장치에 배포하기 위한 콘텐츠 변경들을 포함하는 업로드 사본을(예를 들면, 층 개체를 통해) 생성하지 않는다.

[0081] 도 21에서, 제3 잠금 E3는 저장 장치 상의 문서의 마스터 사본(1250)의 마스터 잠금 테이블에 저장되었다. 제1 사용자 컴퓨팅 장치는 그 후 업데이트들에 대하여 확인하기 위해 저장 장치를(예를 들면, 동기화 매니저를 통해) 폴링한다. 제1 사용자 컴퓨팅 장치는 마스터 사본(1250)에 기초하여 저장 장치로부터 문서의 새로운 다운로드 사본을(예를 들면, 동기화 매니저를 통해) 획득하였다. 새로운 다운로드 사본은 제3 잠금 E3를 포함한다. 제1 저작 애플리케이션은 제3 잠금 E3를 제1 저작 애플리케이션의 작업 사본에(예를 들면, 층 개체를 통해) 자동으로 인스턴스화한다. 전술한 바와 같이, 제3 잠금 E3는 옵션으로 제1 저작 애플리케이션의 사용자 인터페이스를 이용하여 제1 사용자에게 디스플레이될 수 있다.

[0082] 도 22에서, 제2 사용자는 제2 사용자가 제3 데이터 단위의 편집을 완료한 것을 표시할 때 제3 잠금 E3를 해제한다. 제3 잠금 E3는 제2 사용자의 메타데이터 테이블로부터 제거된다. 도 22에 도시된 예시의 블록도에서, 제3 잠금 E3는 제2 저작 애플리케이션의 작업 사본에 대응하는 메타데이터 테이블들로부터 제거된다.

[0083] 도 22에 예에서, 제2 사용자에 의해 제공되는 완료화 표시는 제2 사용자가 다른 사용자들과 콘텐츠 변경들을 공유하기를 원한다는 것을 표시한다. 예를 들면, 제2 사용자는 제3 데이터 단위로부터 커서를 제거한 다음 문서를 저장하는 명령을 제공할 수 있다. 따라서, 제2 저작 애플리케이션은 작업 사본에 기초하여 콘텐츠 변경들을 포함하는 업로드 사본을 생성할 수 있다. 일 실시예에서, 업로드 사본은 또한 작업 사본에 아직 인스턴스화되지 않은 다운로드 사본에서 발견되는 임의의 콘텐츠 업데이트들(예를 들면, 이 경우에는 없음)에 기초할 수 있다. 도시된 예에서, 업로드 사본은 구(phrase) "Goodbye World"를 포함하는 제3 데이터 단위를 포함한다.

[0084] 도 23에서, 저장 장치는 제2 사용자 컴퓨팅 장치의 업로드 사본으로부터 제3 데이터 단위를 획득하고 그 제3 데이터 단위를 저장 장치의 마스터 사본(1250)에 병합한다. 마스터 사본(1250)과 관련된 잠금 테이블은 제3 잠금 E3가 해제된 것을 표시한다. 도 24에 도시된 바와 같이, 저장 장치에 성공적으로 푸시되면, 제2 저작 애플리케이션은 업로드 사본을 제2 캐시(1234)의 다운로드 사본 및 기초 사본에(예를 들면, 층 개체를 통해) 복사한다.

[0085] 도 25에서, 제1 사용자 컴퓨팅 장치는 문서의 마스터 사본(1250)의 새로운 다운로드 사본을(예를 들면, 동기화 매니저를 통해) 획득한다. 제1 저작 애플리케이션은 새로운 다운로드 사본이 획득된 것을(예를 들면, 층 개체를 통해) 결정하고 새로운 다운로드 사본이 콘텐츠 변경들을 포함하는 것을 결정한다. 저작 애플리케이션은 임의의 새로운 잠금들(이 경우에는 없음)을 캐시(1214)의 작업 사본에 자동으로 병합한다. 제1 저작 애플리케이션의 사용자 인터페이스는 콘텐츠 업데이트들이 보기를 위해 이용 가능하다는 것을 표시한다.

[0086] 도 26에서, 제1 사용자는 캐시(1214)에서 이용 가능한 콘텐츠 업데이트들을 문서의 작업 사본에 인스턴스화하기로 결정한다. 따라서, 저작 애플리케이션은 캐시(1214)에 저장된 다운로드 사본을 제1 저작 애플리케이션의 작업 사본과 병합한다. 제3 데이터 단위는 제1 저작 애플리케이션의 작업 사본에 추가되고 제1 사용자에게 디스플레이된다. 제3 잠금 E3는 제1 저작 애플리케이션의 작업 사본과 관련된 메타데이터 테이블에서 해제된다. 그러므로, 제1 사용자는 제3 데이터 단위를 편집하도록 제1 저작 애플리케이션에 지시할 수 있다.

[0087] 이제 도 27-31을 참조하여, 공동 저작 세션 동안의 사용자 경험이 아래에 상세히 설명된다. 도 27-31은 제1 사용자 및 제2 사용자가 한 문서(2010)를 공동으로 저작할 때 제1 저작 애플리케이션의 예시적인 사용자 인터페이스(2000)의 변경들을 예시한다. 제1 사용자인 Jane Doe는 제1 저작 애플리케이션을 이용하여 문서(2010)를 편

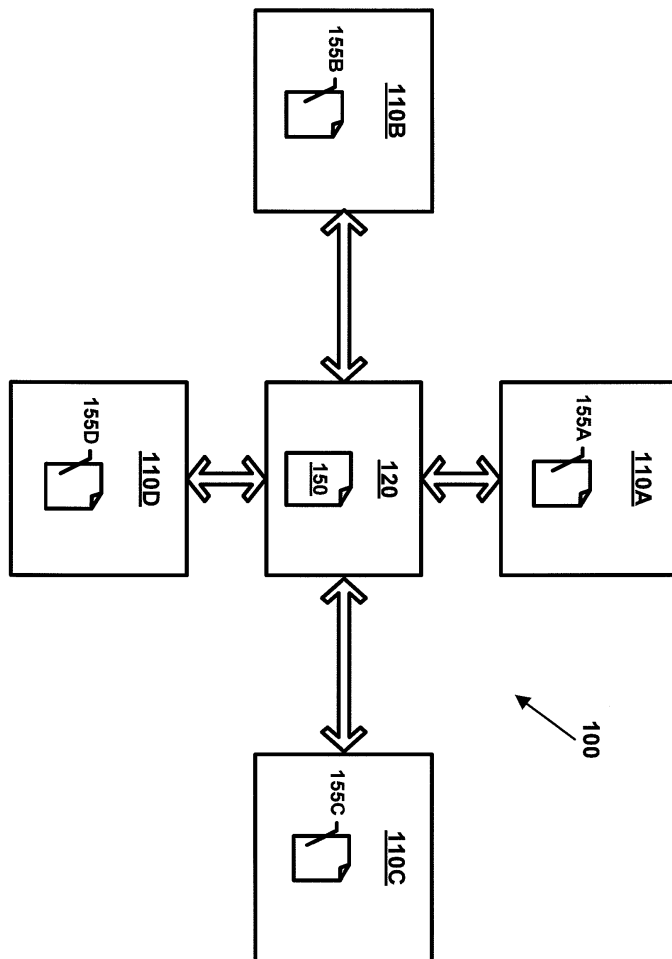
집하고 제2 사용자인 John Doe는 다른 저작 애플리케이션을 이용하여 문서(2010)를 편집한다.

- [0088] 문서(2010)는 제1 데이터 단위(2012), 제2 데이터 단위(2014), 제3 데이터 단위(2016), 및 제4 데이터 단위(2018)를 포함한다. 제1 사용자는 제1 데이터 단위(2012)의 주위에 콘텐츠 잠금(2022)을 건다. 예를 들면, 일 실시예에서, 제1 사용자는 커서(2015)를 제1 데이터 단위(2012)에 이동시킬 수 있다. 다른 실시예에서, 제1 사용자는 제1 데이터 단위(2012)에 대한 하나 이상의 편집 동작들을 수행하는 명령을 제공할 수 있다.
- [0089] 도 28에서, 제1 사용자는 제1 데이터 단위(2012)에 텍스트를 추가하였다. 제1 데이터 단위에 대한 잠금은 제2 사용자가 제1 데이터 단위(2012)를 편집하는 것을 금한다. 일 실시예에서, 제1 잠금(2022)은 제2 사용자가 제1 데이터 단위(2012)를 편집하는 것을 방해한다. 다른 실시예에서, 제1 잠금(2022)은 제1 데이터 단위(2012)를 편집하는 것에 대하여 제2 사용자에게 경고하지만, 제2 사용자에게 의한 편집을 금하지는 않는다. 또 다른 실시예에서, 제1 잠금(2022)은 단지 제1 사용자가 제1 데이터 단위(2012)를 편집하려는 의도를 표현한 것을 표시한다.
- [0090] 도 29에서, 제2 사용자는 제3 데이터 단위(2016)에 관한 콘텐츠 잠금(2024)을 걸었다. 제2 사용자는 그의 변경들을 공유하지 않기로 결정하였기 때문에, 제2 잠금은 제1 저작 애플리케이션의 사용자 인터페이스(2000) 상에서 보일 수 있지만, 제2 사용자에게 의해 제3 데이터 단위(2016)에 행해진 임의의 변경들은 아직 보이지 않는다. 도시된 예에서, 제2 잠금(2024)은 제2 잠금(2024)의 범위를 표시하는 각괄호를 디스플레이하는 것에 의해 사용자 인터페이스(2000)에 의해 표시될 수 있다. 제2 잠금(2024)은 또한 제2 잠금(2024)의 소유자를 표시하는 네임태그(nametag)를 포함할 수 있다. 도 29에서, 네임태그는 제2 사용자를 식별하는 텍스트 및 아이콘을 포함한다. 아이콘은 색, 기호, 또는 다른 그래픽에 의해 사용자를 더욱 식별할 수 있다.
- [0091] 도 30에서, 제1 저작 애플리케이션은 문서(2010)에 대한 콘텐츠 업데이트들을 수신한다. 제2 사용자는 다른 사용자들과 콘텐츠 변경들을 공유하기로 결정하였고 따라서 문서(2010)의 마스터 사본에 콘텐츠 업데이트들을 병합하였다(예를 들면, 위에 설명된 동기화 프로세스들 중 하나를 통해). 제1 저작 애플리케이션의 동기화 매니저는 마스터 사본을 저장하는 장치를 폴링함으로써 문서(2010)의 마스터 사본으로부터 업데이트들을 획득하였다.
- [0092] 제1 저작 애플리케이션의 사용자 인터페이스(2000)는 콘텐츠 업데이트의 이용 가능성을 표시하는 경고(2030)를 제1 사용자에게 디스플레이한다. 도시된 예에서, 텍스트 박스(2030)가 사용자 인터페이스(2000)에 나타난다. 다른 실시예에서, 제2 잠금 또는 제2 잠금 내의 텍스트는 색, 패턴, 또는 포매팅을 변경할 수 있다. 다른 실시예에서는, 각 새로운 업데이트가 이용 가능할 때 업데이트 카운터가 증가한다.
- [0093] 제1 사용자는 제2 사용자에게 의해 행해진 변경들을 보기로/인스턴스화하기로 또는 콘텐츠 업데이트들을 무시하기로 결정할 수 있다. 제1 사용자는 콘텐츠 업데이트들을 무시하면서 문서(2010) 내의 임의의 잠기지 않은 데이터 단위들을 계속해서 편집할 수 있다. 일 실시예에서, 제1 사용자는 그의 업데이트들을 마스터 사본과 병합하기 전에 콘텐츠 업데이트들을 인스턴스화해야 한다. 도 31에서, 제1 사용자는 제1 저작 애플리케이션의 사용자 인터페이스(2000)에 의해 디스플레이된 문서(2010)에 콘텐츠 업데이트들을 인스턴스화하도록 제1 저작 애플리케이션에 지시한다. 제2 사용자에게 의해 행해진 콘텐츠 변경들은 이제 제1 사용자에게 보인다.
- [0094] 본 명세의 실시예들은 컴퓨터 프로세스(방법), 컴퓨팅 시스템으로서, 또는 컴퓨터 프로그램 제품 또는 컴퓨터 판독 가능한 매체와 같은 제조물로서 구현될 수 있다. 프로세스들(프로그램들)은, 이 문서에서 설명된 구조들을 포함하여, 임의의 수의 방법으로 구현될 수 있다. 하나의 그러한 방법은 이 문서에서 설명된 유형의 장치들의 머신 동작들에 의한 것이다. 다른 옵션의 방법은 그 방법들의 개별 동작들 중 하나 이상의 동작이 그 동작들 중 일부를 수행하는 한 명 이상의 인간 조작자들과 관련하여 컴퓨팅 장치 상에서 수행되는 것이다. 이들 인간 조작자들은 서로 같은 곳에 있을 필요가 없고, 각각은 단지 프로그램의 일부를 수행하는 머신과 함께 있을 수 있다.
- [0095] 컴퓨터 프로그램 제품은 컴퓨터 시스템에 의해 판독 가능하고 컴퓨터 프로세스를 실행하기 위한 명령어들의 컴퓨터 프로그램을 인코딩한 컴퓨터 저장 매체일 수 있다. 컴퓨터 프로그램 제품은 또한 컴퓨팅 시스템에 의해 판독 가능하고 컴퓨터 프로세스를 실행하기 위한 명령어들의 컴퓨터 프로그램을 인코딩한 반송파 상의 전파 신호일 수 있다. 여기에 사용된 용어 컴퓨터 판독 가능한 매체는 저장 매체 및 통신 매체 양쪽 모두를 포함한다.
- [0096] 숙련된 당업자들은 본 명세가 핸드헬드 장치, 마이크로프로세서 시스템, 마이크로프로세서 기반 또는 프로그램 가능한 소비자 전자 장치, 미니컴퓨터, 메인프레임 컴퓨터 등을 포함한 기타 컴퓨터 시스템 구성을 이용해 실시될 수 있다는 것을 알 것이다. 본 명세는 또한 통신 네트워크를 통하여 연결되어 있는 원격 처리 장치들에 의

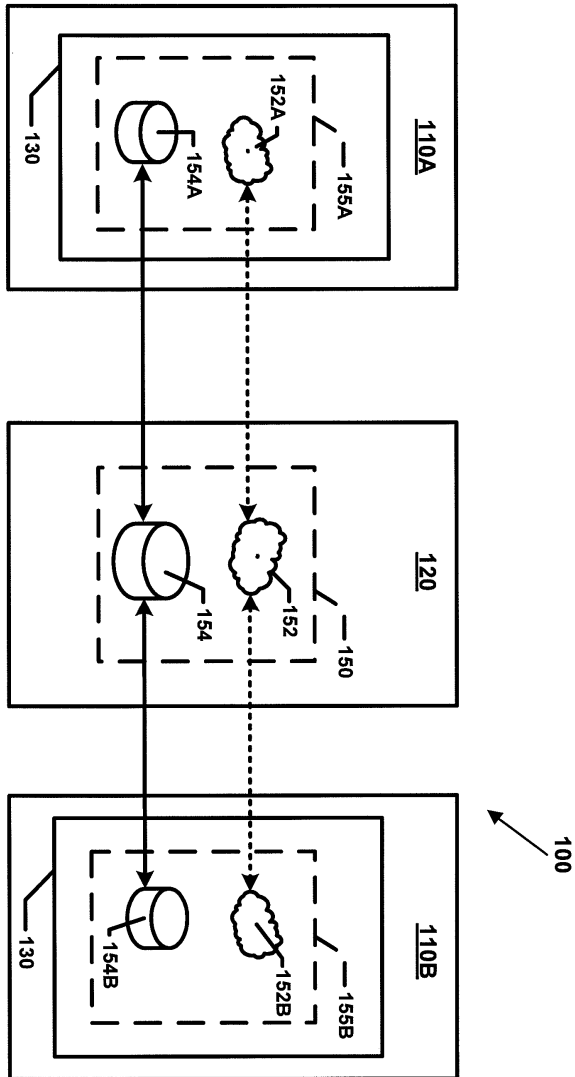
해 태스크들이 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈들은 로컬 및 원격 양쪽 모두의 메모리 저장 장치들에 위치할 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조, 및 기타 유형의 구조를 포함한다.

도면

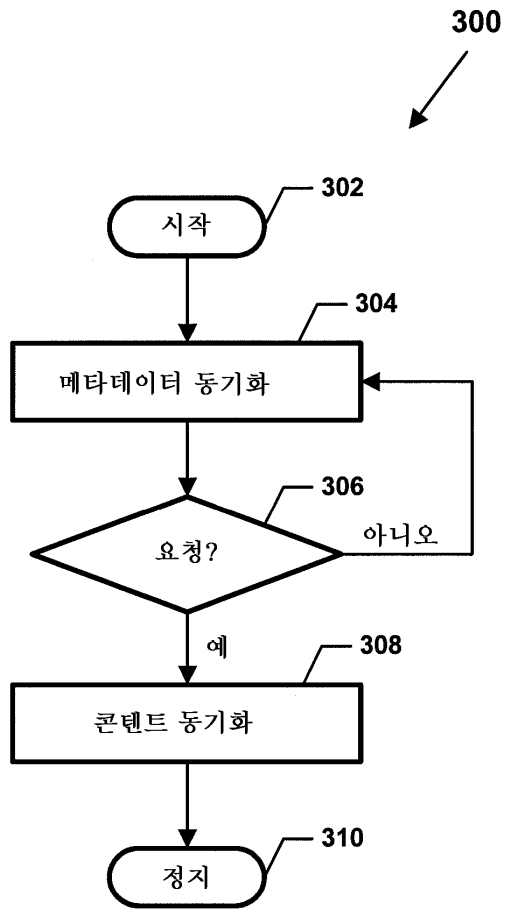
도면1



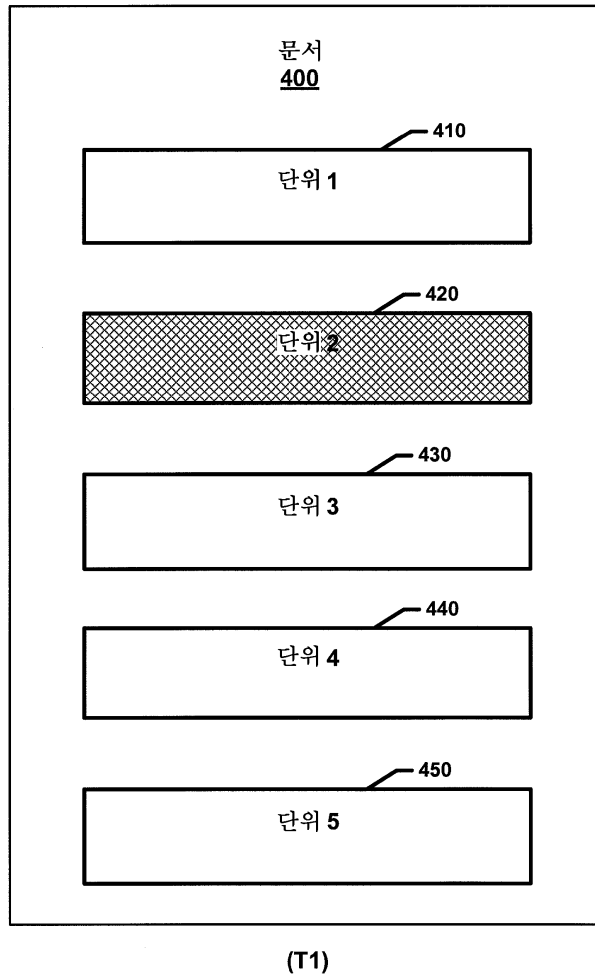
도면2



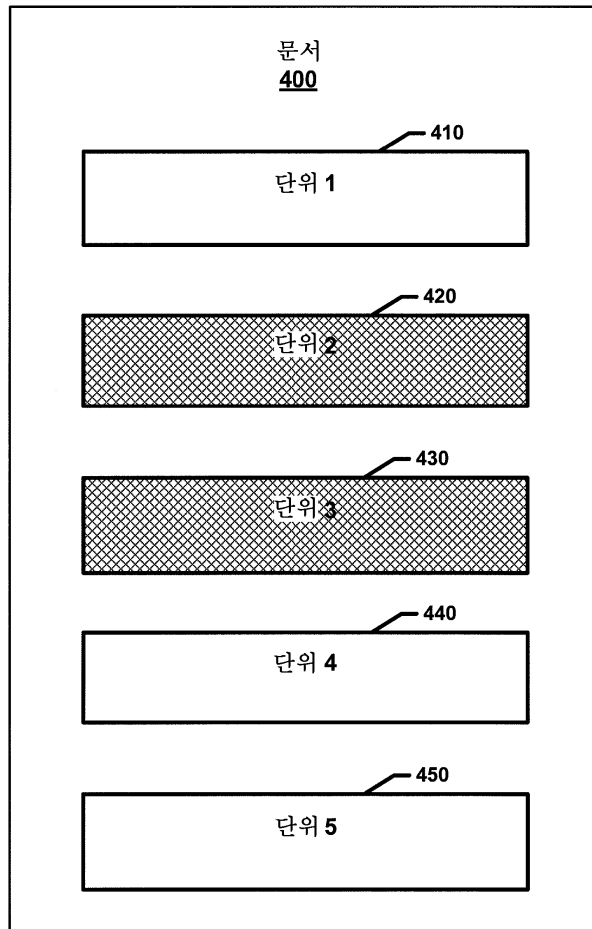
도면3



도면4

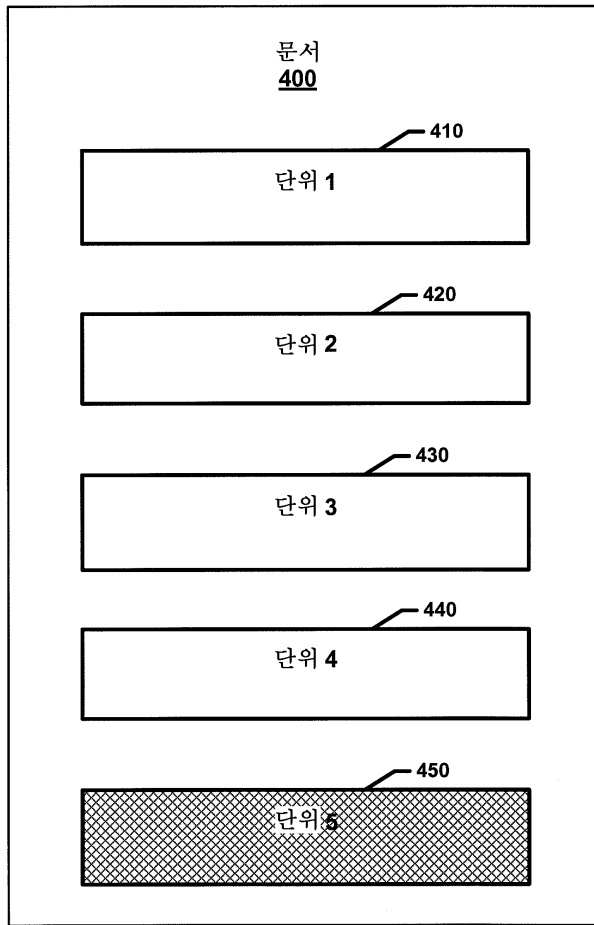


도면5



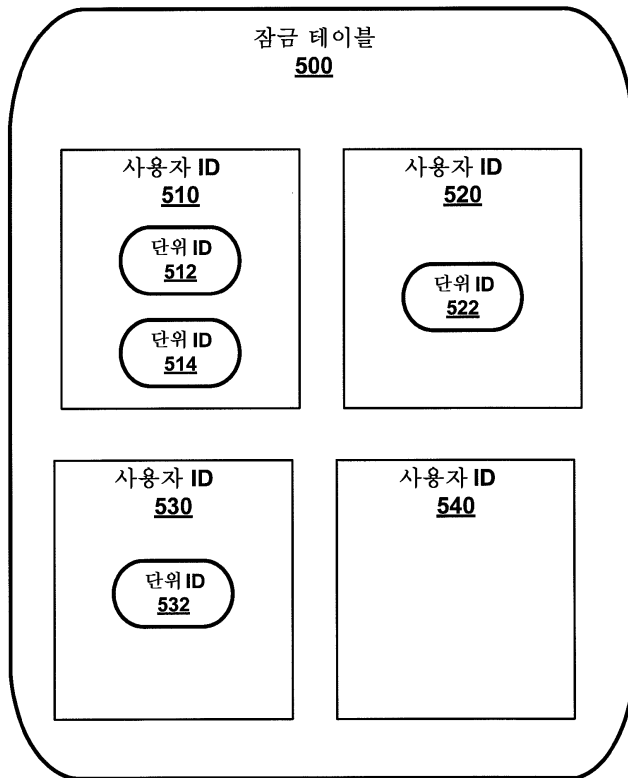
(T2)

도면6

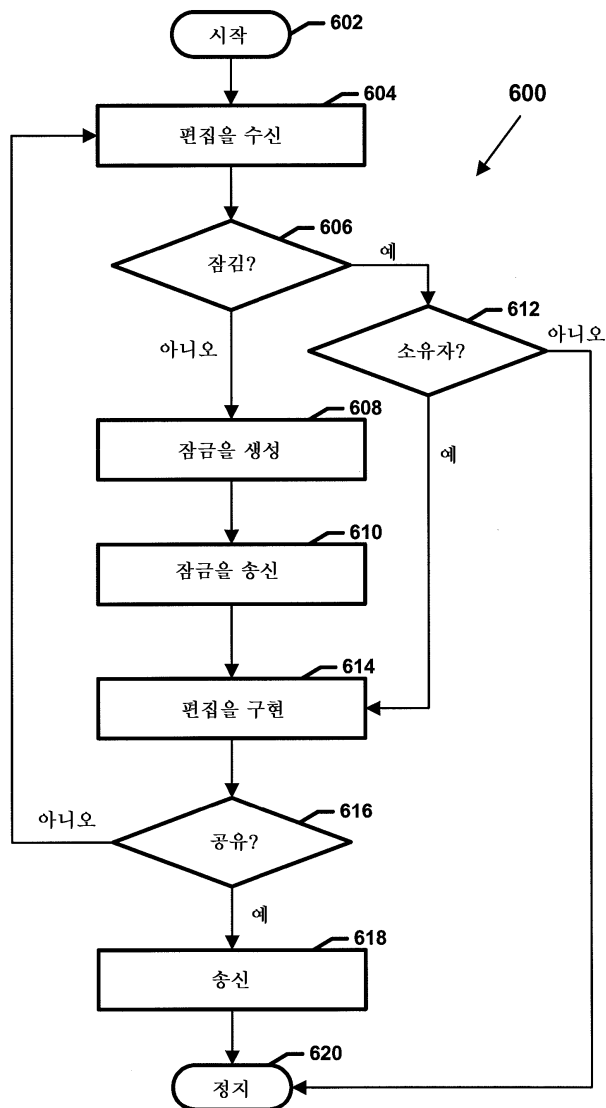


(T3)

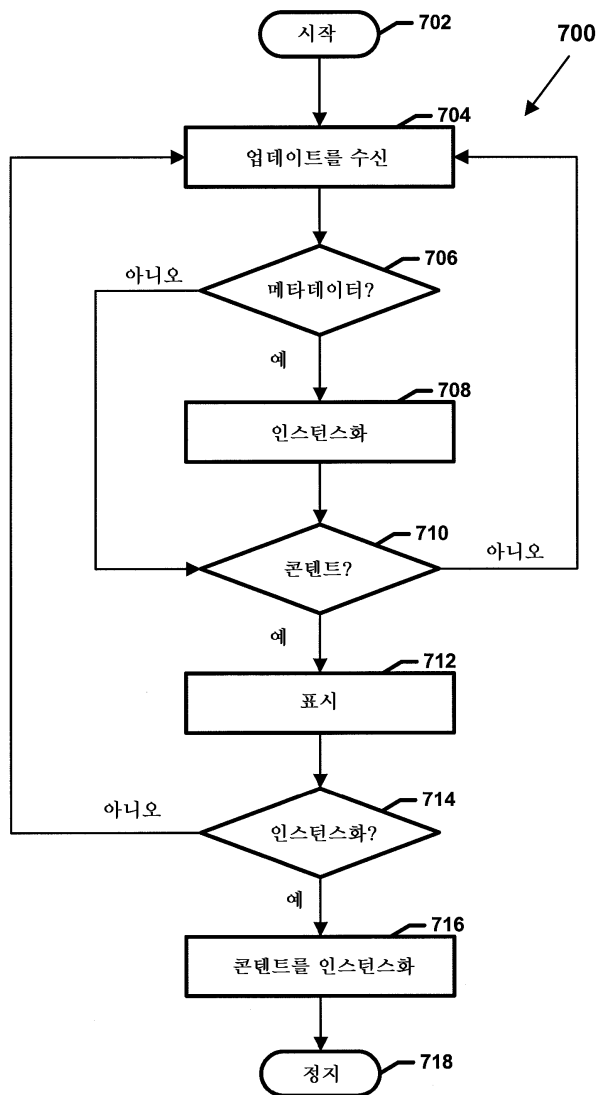
도면7



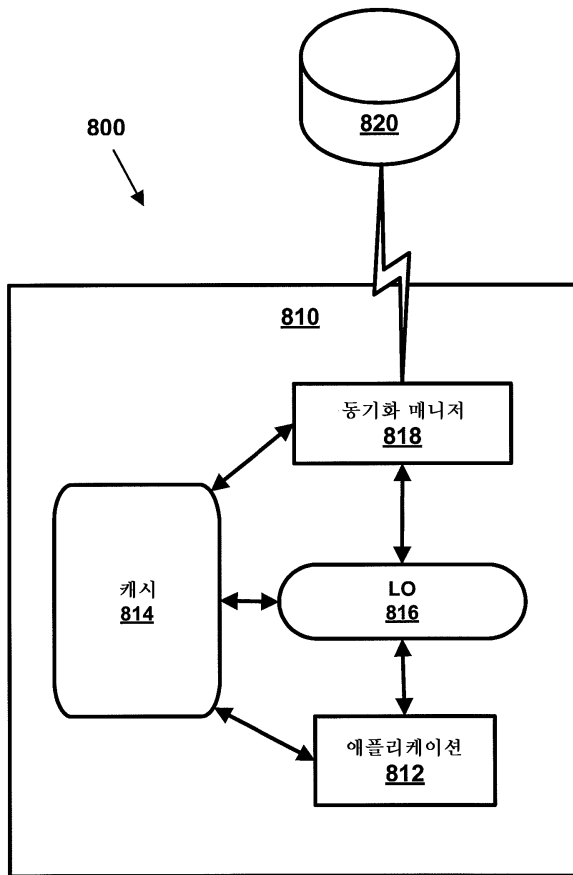
도면8



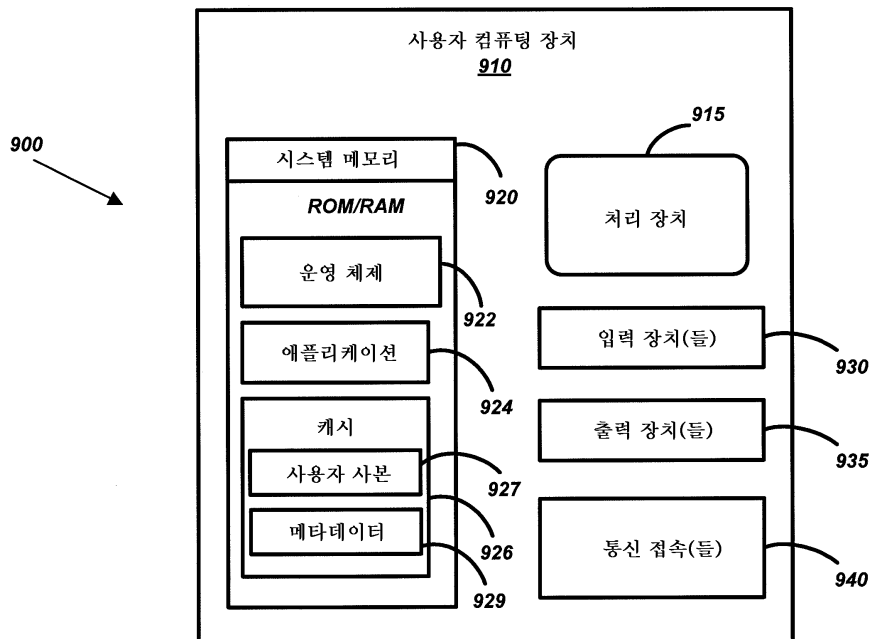
도면9



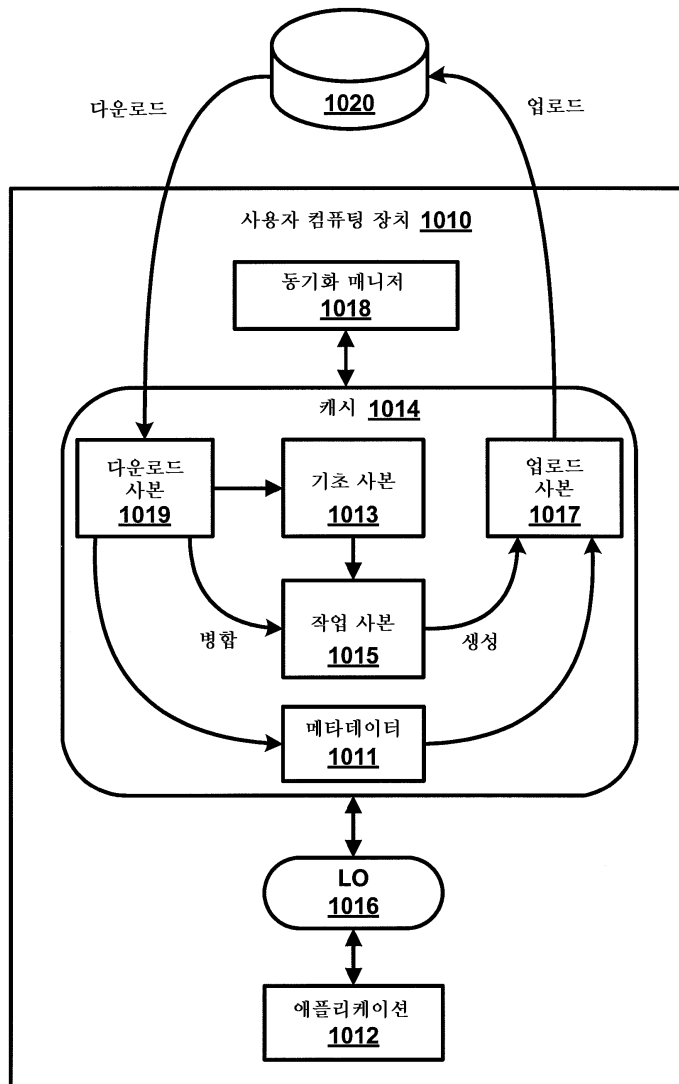
도면10



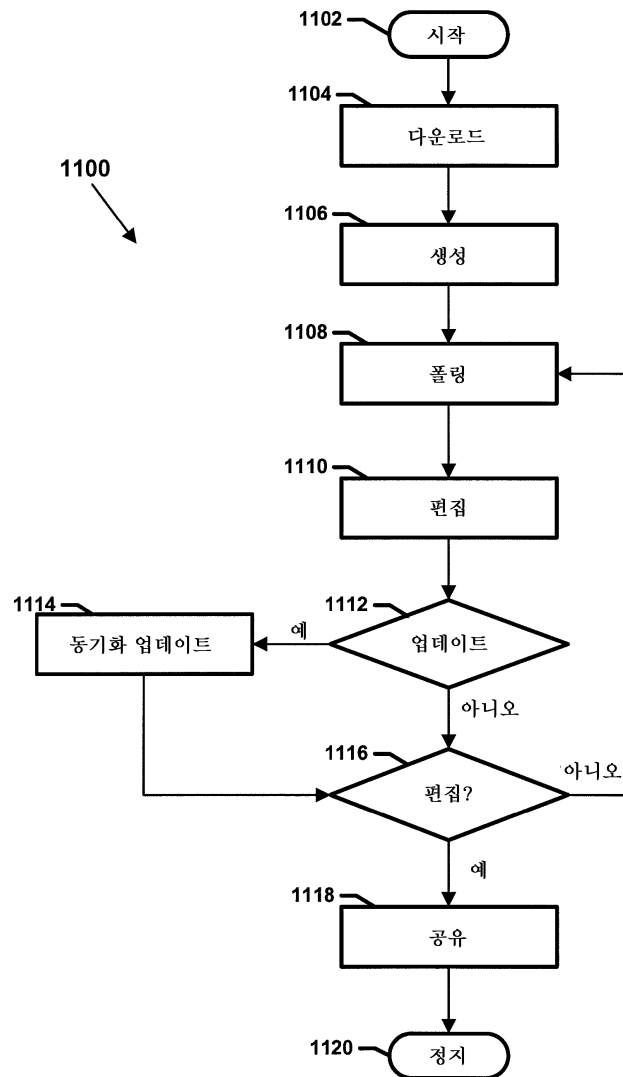
도면11



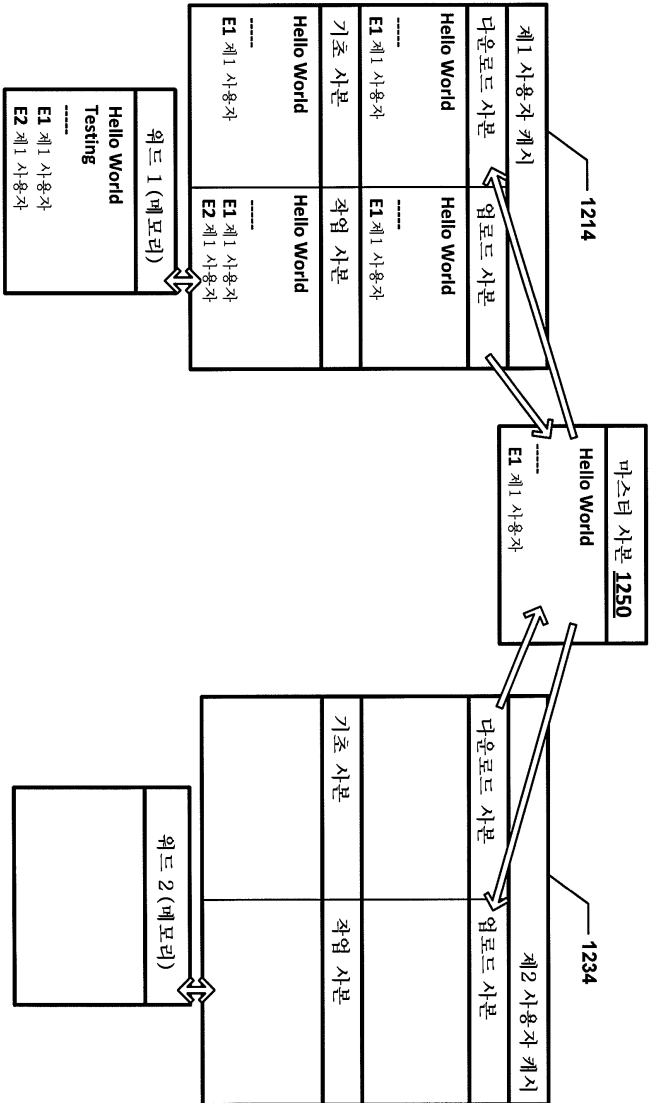
도면12



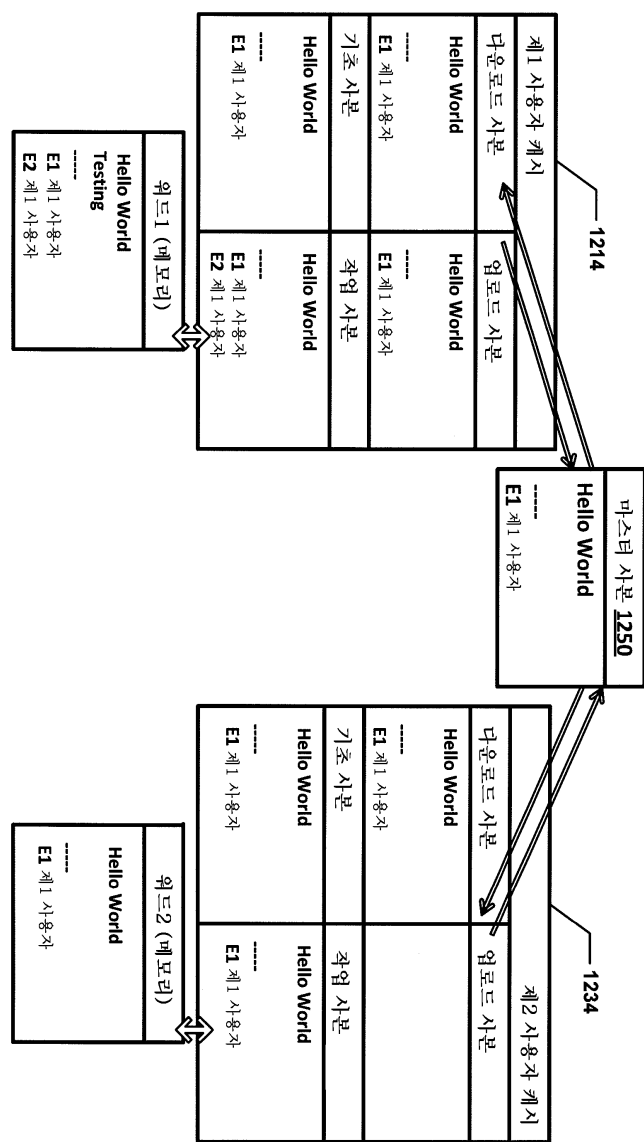
도면13



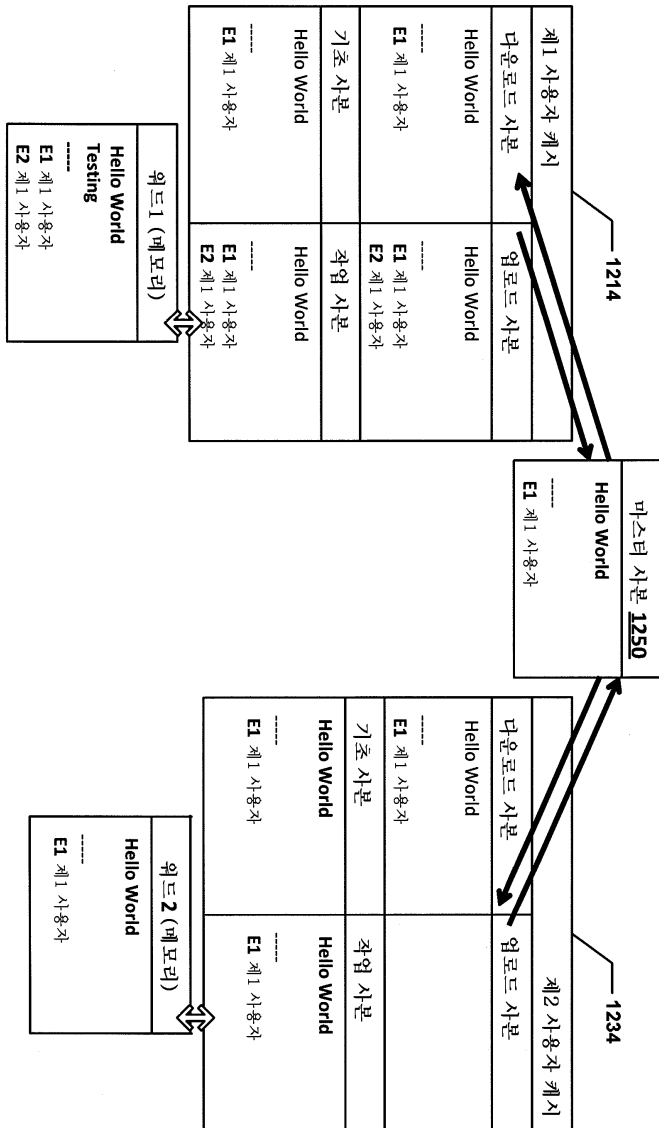
도면14



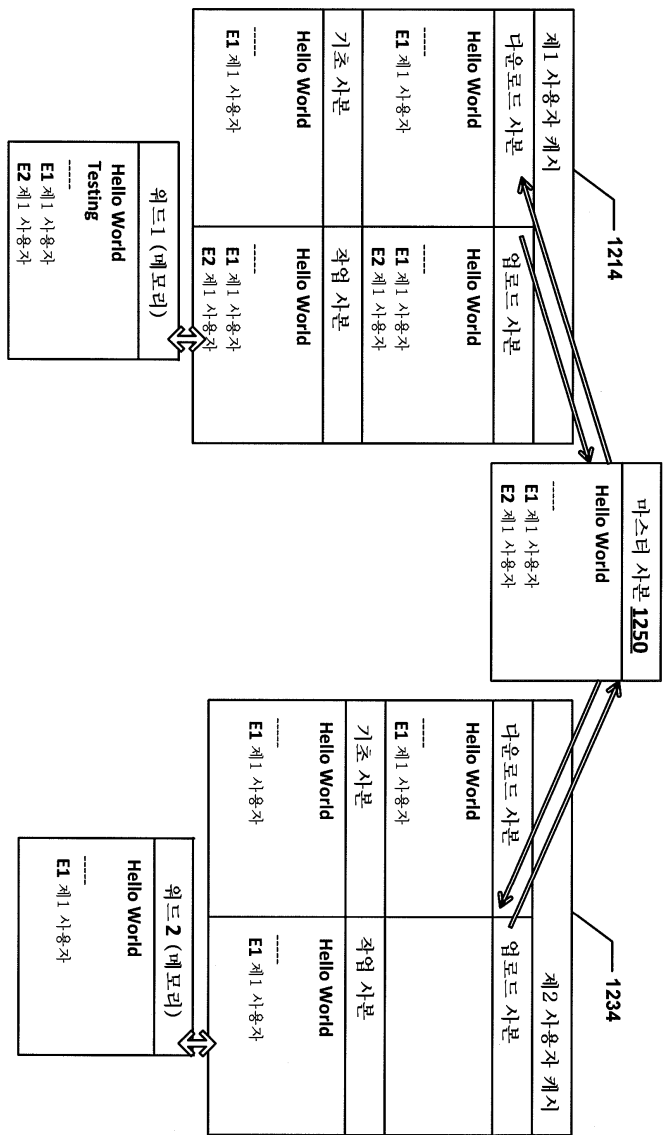
도면15



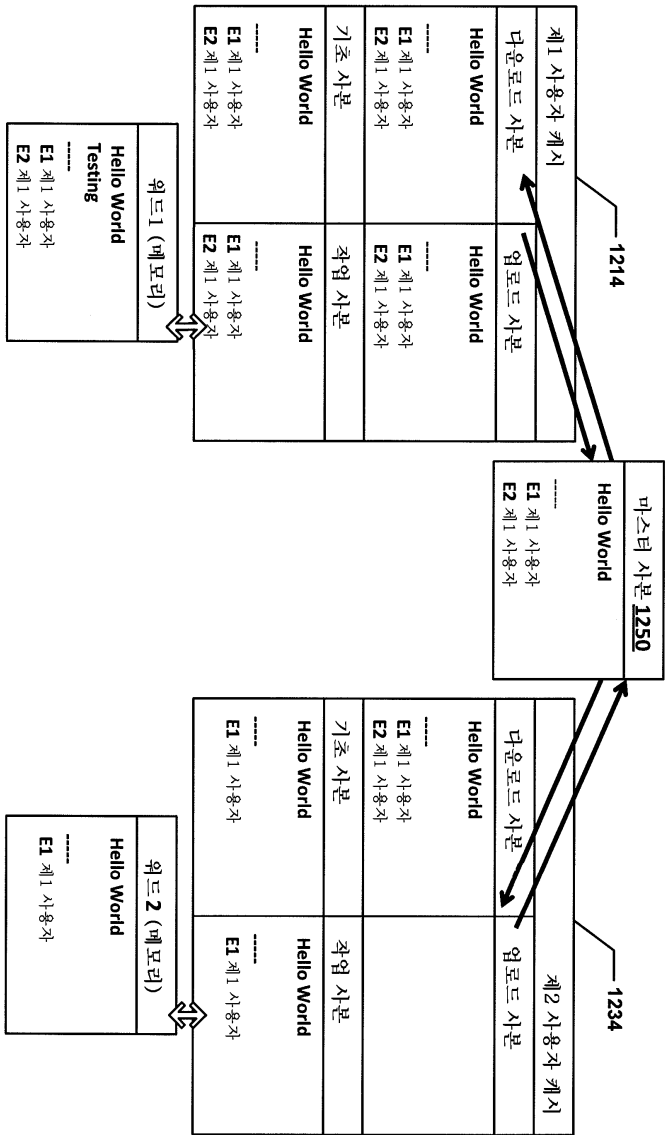
도면16



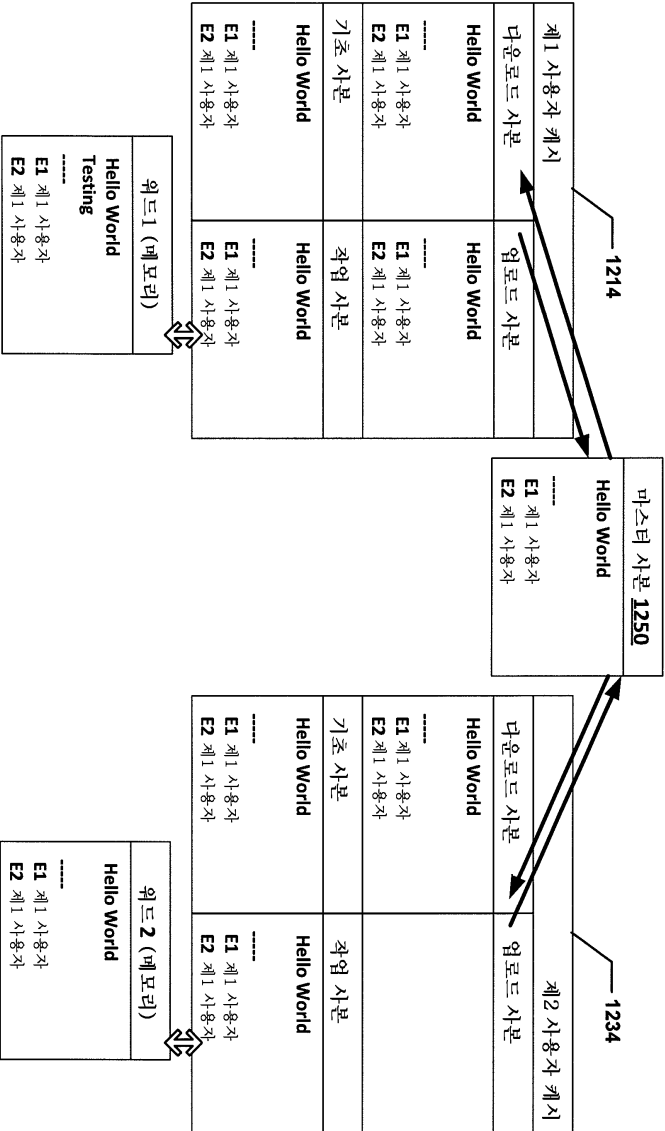
도면17



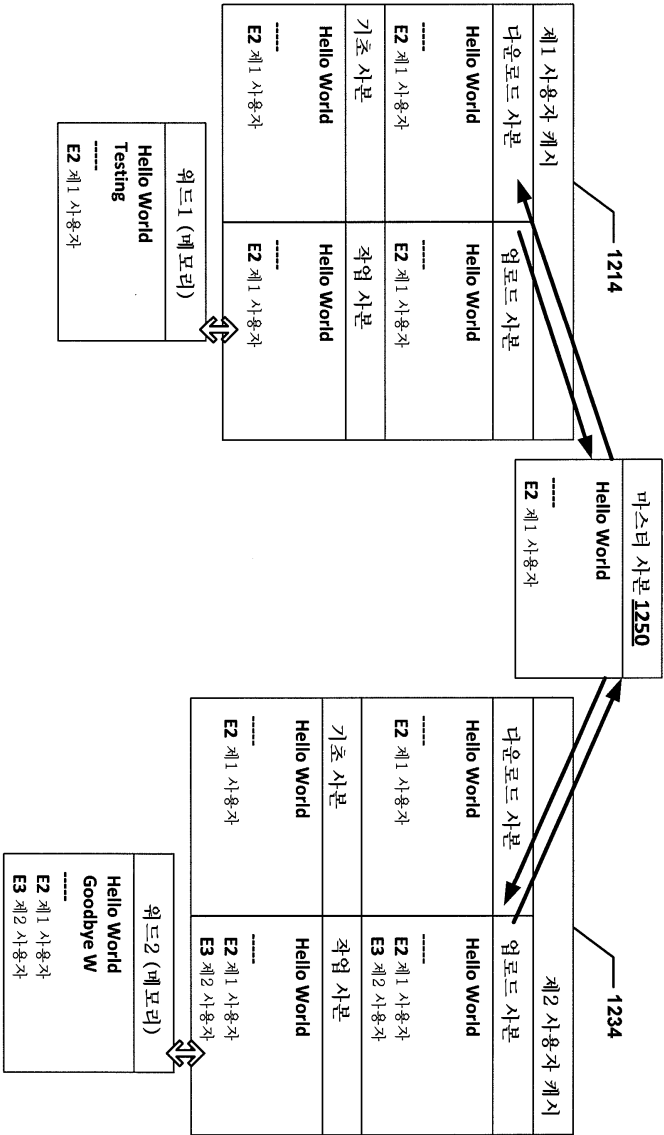
도면18



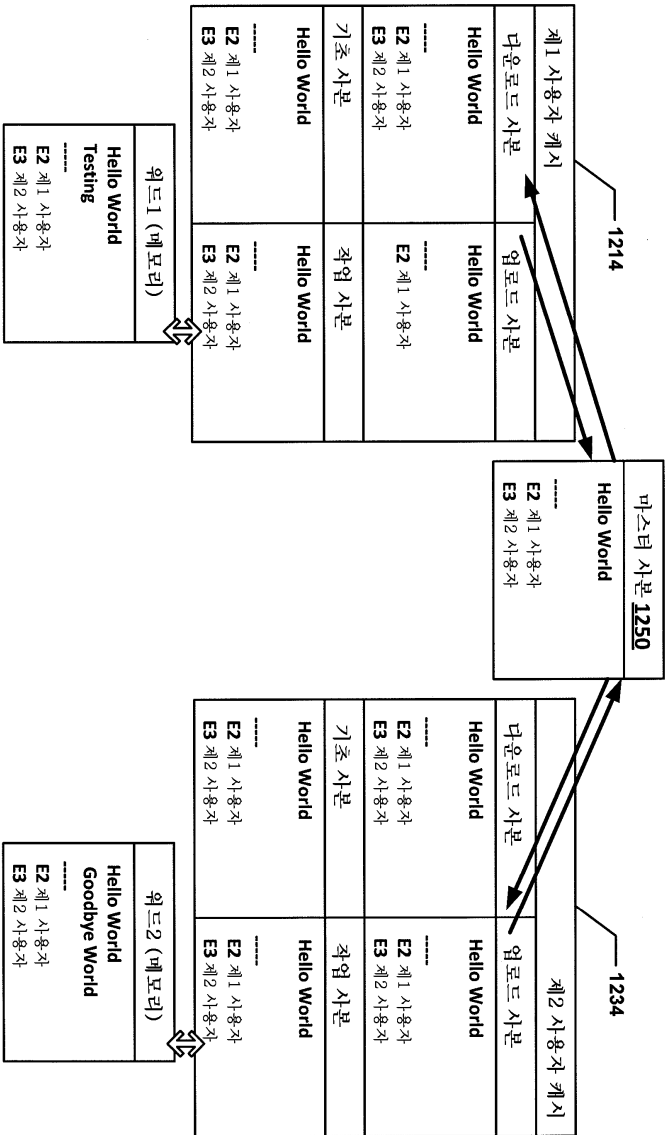
도면19



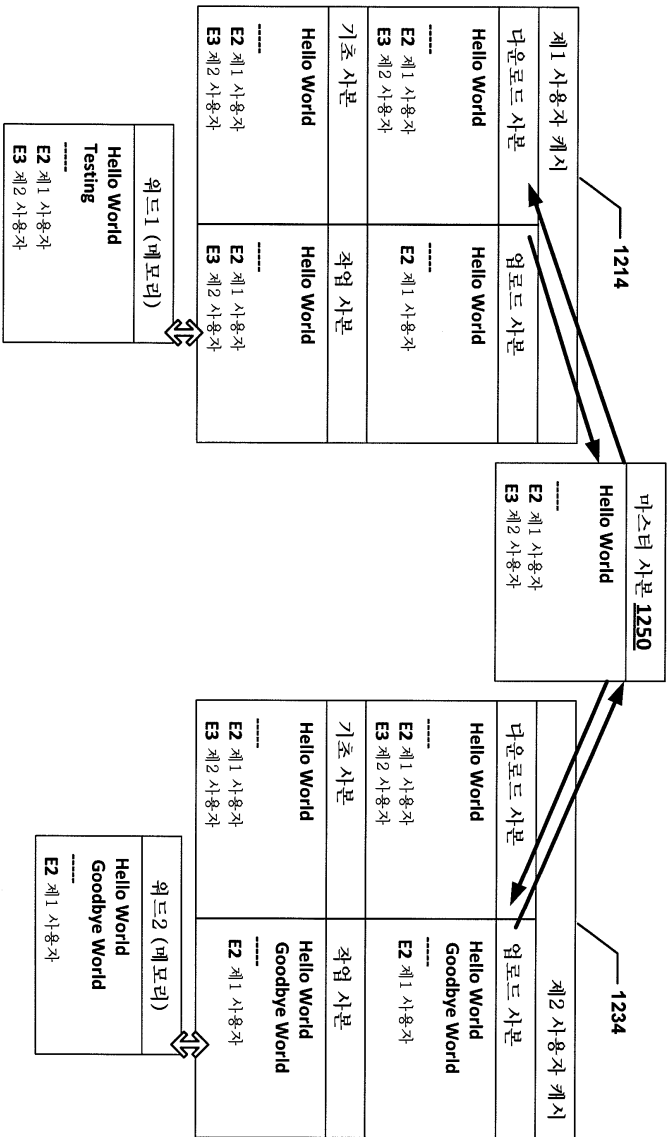
도면20



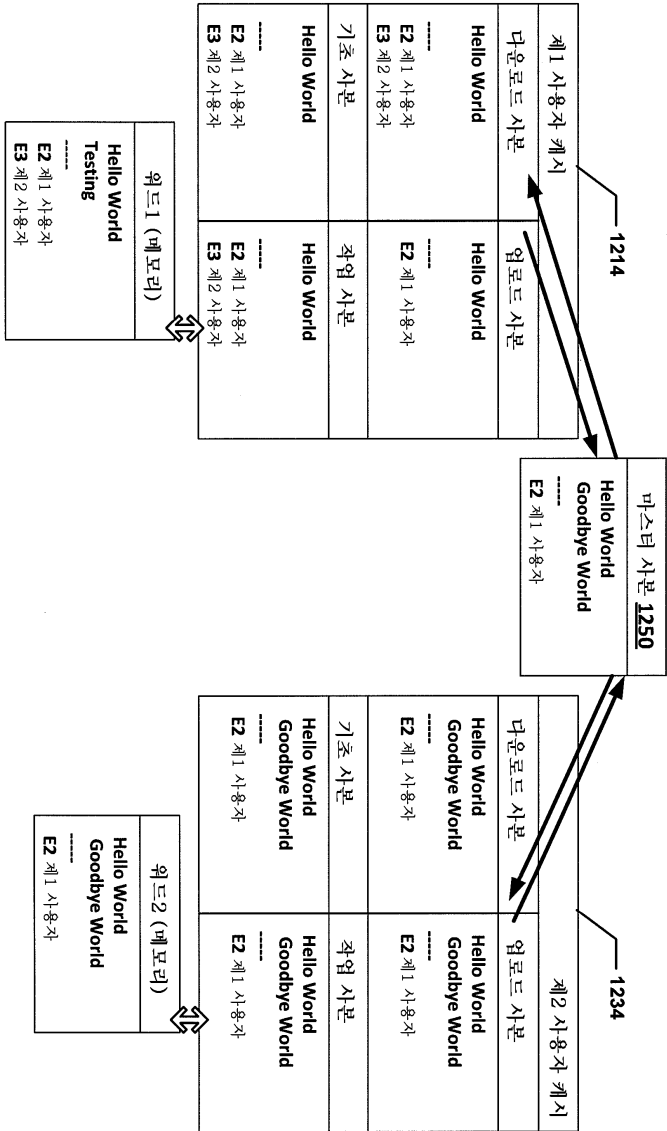
도면21



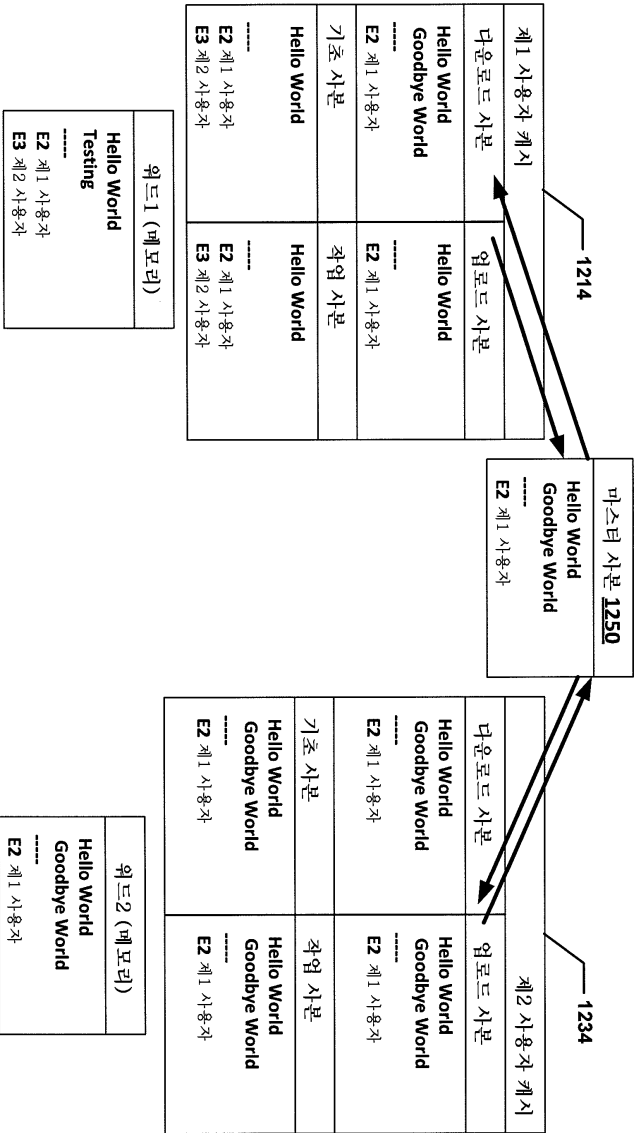
도면22



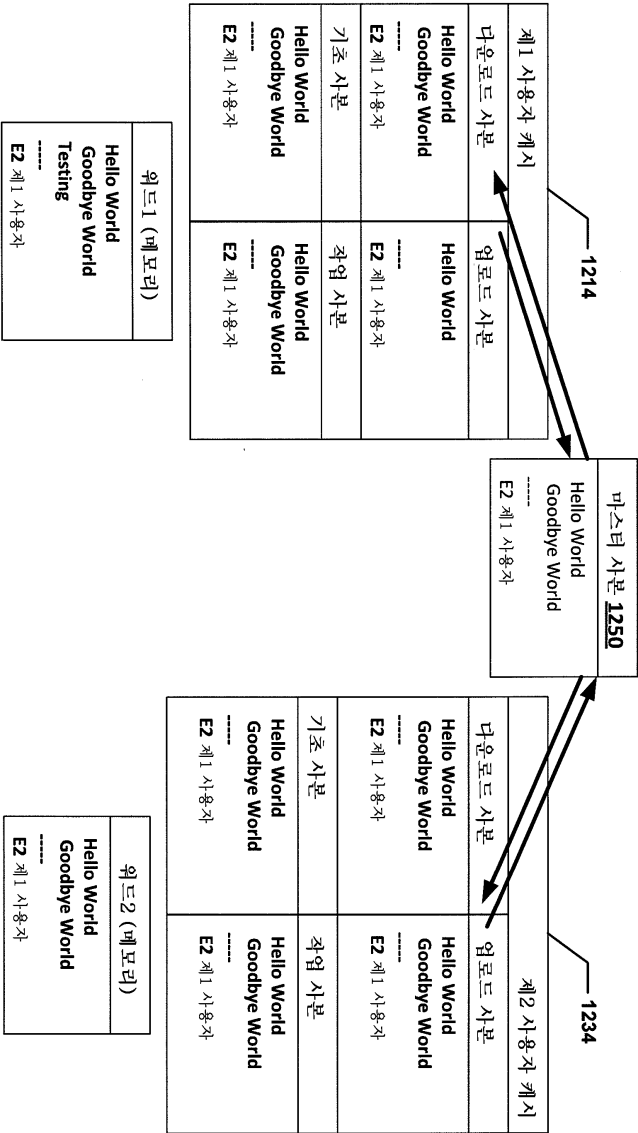
도면24



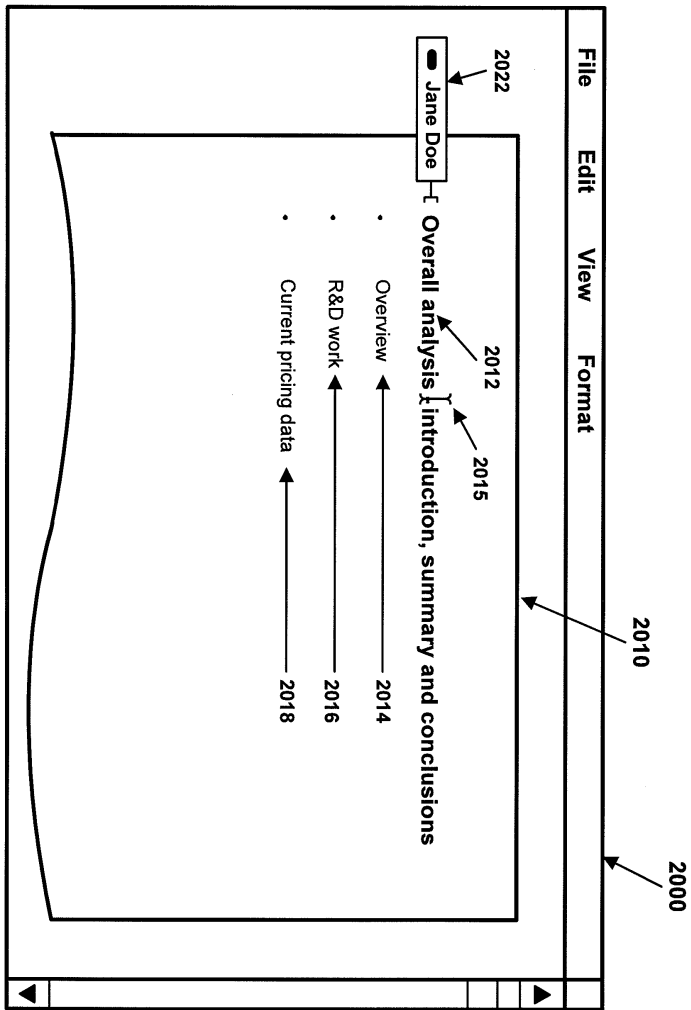
도면25



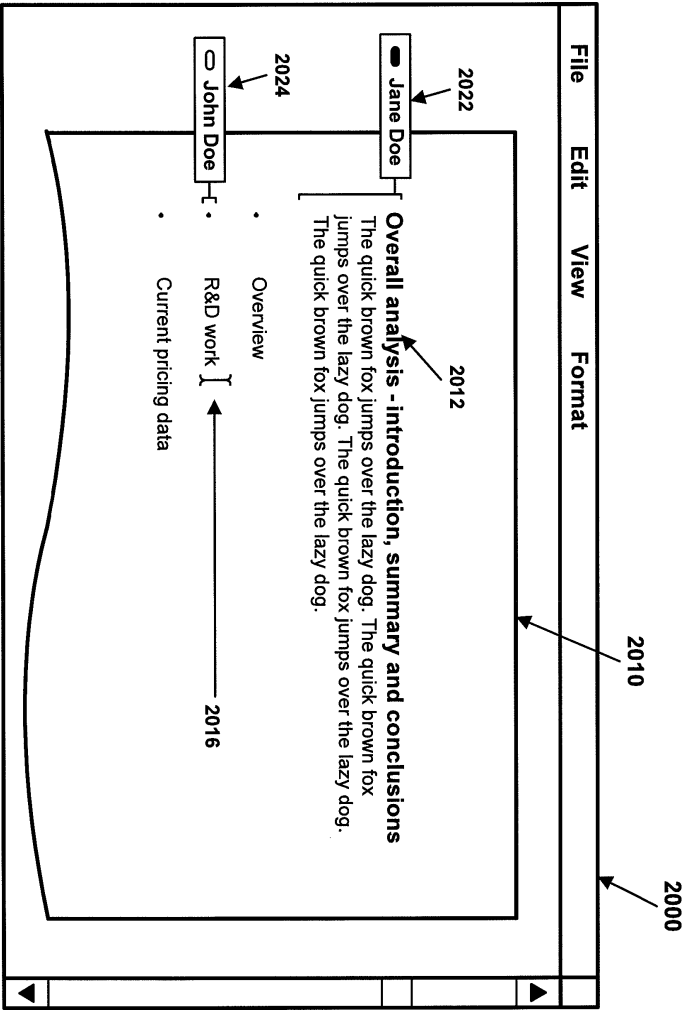
도면26



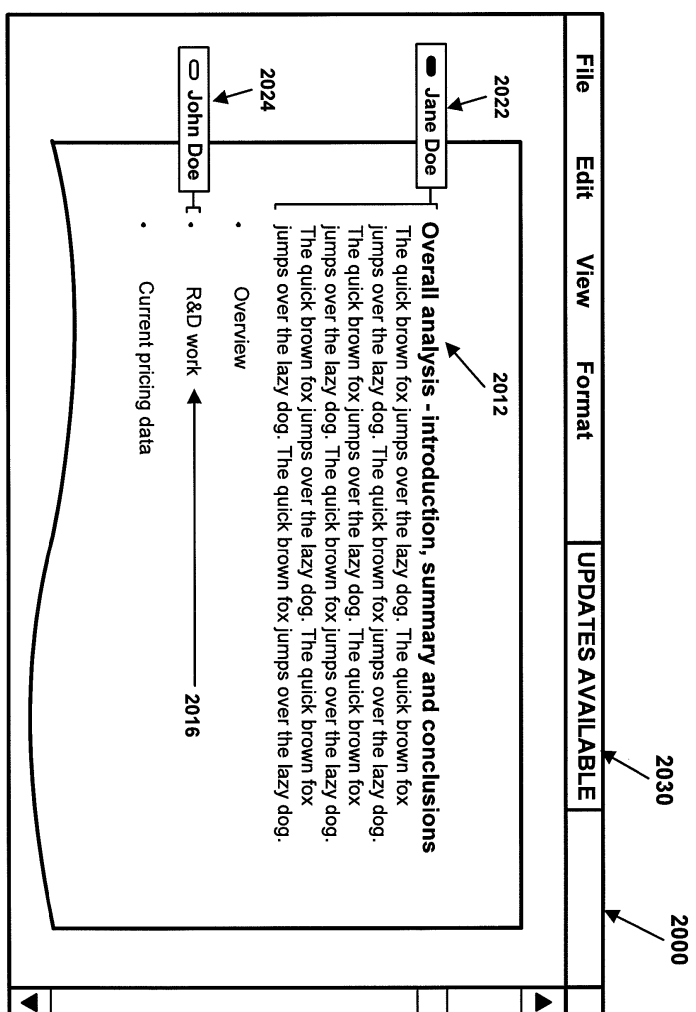
도면27



도면29



도면30



도면31

