(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년06월04일
(11) 등록번호 10-1153137
(24) 등록일자 2012년05월29일

(51) 국제특허분류(Int. Cl.)
H04L 12/28 (2006.01) *H04L 12/20* (2006.01)
H04L 12/24 (2006.01)
(21) 출원번호 10-2005-0114102
(22) 출원일자 2005년11월28일
심사청구일자 2010년11월19일
(65) 공개번호 10-2006-0083117
(43) 공개일자 2006년07월20일
(30) 우선권주장
11/035,617 2005년01월14일 미국(US)
(56) 선행기술조사문헌
JP2004171083 A
WO03085525 A2
KR1020030014513 A

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
클라크, 시몬 피.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
라스무센, 데이비드
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
코프맨, 이고르
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
(74) 대리인
제일특허법인

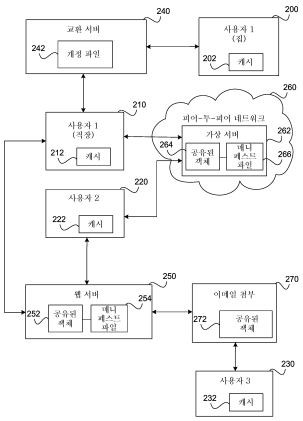
전체 청구항 수 : 총 17 항 심사관 : 김창범

(54) 발명의 명칭 **동기적 통신 모드와 비동기적 통신 모드 사이를전이하기위한 방법 및 시스템**

(57) 요 약

수많은 다른 사용자들은 여러 다른 트랜스포트를 통해 동일한 공유된 객체를 동시에 액세스하고, 개정하고, 갱신할 수 있다. 클라이언트들은 물리적인 서버를 통해 공유된 객체를 액세스함으로써 비동기적으로 통신할 수 있다. 클라이언트들은 또한 피어-투-피어 네트워크를 통해 공유된 객체를 액세스함으로써 동기적으로 통신할 수 있다. 공유된 객체에 연관된 매니페스트 파일(manifest file)이 공유된 객체의 다른 버전들 및 인스턴스들이 저장되어 있는 위치들을 식별한다. 공유된 객체 및 연관된 매니페스트 파일에 의해 클라이언트는 로컬 액세스, 동기적 통신 및 비동기적 통신 사이를 자연스럽게 전이할 수 있어서, 사용자는 통신 모드가 변경되었다는 것을 인식하지 못하게 된다. 사용자는 위치를 변경할 수 있으며, 임의 이용가능한 데이터 공유 트랜스포트는 자동으로 식별된다. 그러므로, 사용자는 공유된 객체를 액세스할 수 있고, 다른 메커니즘을 통해 다른 인가된 사용자와 협력할 수 있다.

대 표 도 - 도2



특허청구의 범위

청구항 1

동기적 통신 모드와 비동기적 통신 모드 사이에서 전이하여 복수의 사용자 개정들을 공유된 객체에 동기화시키기 위한 컴퓨터 구현된 방법으로서,

공유된 객체를 액세스하는 단계;

상기 공유된 객체와 연관된 매니페스트 파일을 가져오는 단계 - 상기 매니페스트 파일은 상기 공유된 객체의 고유한 위치 식별자에 의해 식별된 위치로부터 가져 와지고, 상기 매니페스트 파일은 상기 공유된 객체의 다른 인스턴스가 저장되어 있는 위치를 식별함 - ;

상기 매니페스트 파일에서 식별된 상기 공유된 객체의 인스턴스를 또한 액세스하고 있는 적어도 하나의 클라이언트와 동기적으로 직접 제1 클라이언트에 의해 통신하는 단계 - 상기 동기적으로 직접 통신하는 단계는 피어-투-피어 네트워크를 통해 상기 공유된 객체를 액세스하는 단계를 포함함 - ; 및

제1 서버를 통해 상기 공유된 객체를 또한 액세스하고 있는 적어도 하나의 클라이언트와 비동기적으로 상기 제1 클라이언트에 의해 통신하는 단계 - 상기 비동기적으로 통신하는 단계는 상기 제1 서버를 통해 상기 공유된 객체를 액세스하는 단계를 포함함 -

를 포함하고,

상기 피어-투-피어 네트워크가 이용가능하지 않을 때에 상기 공유된 객체는 상기 제1 서버를 통해 액세스가능하고,

상기 동기적으로 직접 통신하는 단계는 상기 제1 클라이언트가 상기 제1 서버로부터 연결이 끊어져 상기 적어도 하나의 클라이언트와 동기적으로 직접 계속해서 통신하는 것을 허용하는 방법.

청구항 2

제1항에 있어서, 상기 동기적으로 직접 통신하는 단계는 상기 제1 서버를 통해 상기 공유된 객체를 액세스하도록 인가된 클라이언트들을 식별하는 단계를 더 포함하는 방법.

청구항 3

제1항에 있어서, 상기 공유된 객체를 액세스하는 단계는 제2 서버를 통해 공유된 객체를 액세스하는 단계를 더 포함하는 방법.

청구항 4

제1항에 있어서, 상기 동기적으로 직접 통신하는 단계는 상기 매니페스트 파일에서 식별된 상기 공유된 객체의 인스턴스를 또한 액세스하고 있는 상기 적어도 하나의 클라이언트와 상기 피어-투-피어 네트워크를 구축하는 단계를 더 포함하고, 상기 피어-투-피어 네트워크는 클라이언트들 간의 직접 통신을 허용하는 방법.

청구항 5

제1항에 있어서, 상기 비동기적으로 통신하는 단계는 상기 제1 서버와 연결되는 단계를 더 포함하는 방법.

청구항 6

동기적 통신 모드와 비동기적 통신 모드 사이에서 전이하여 복수의 사용자 개정들을 공유된 객체에 동기화시키기 위한 시스템으로서,

제1 클라이언트;

상기 제1 클라이언트에 의해 액세스되도록 구성된 공유된 객체 - 매니페스트 파일을 가져오기 위한 위치는 상기 공유된 객체의 고유한 위치 식별자에 의해 식별됨 - ; 및

상기 공유된 객체와 연관되어 상기 공유된 객체의 다른 인스턴스가 저장되어 있는 위치를 식별하는 매니페스트 파일

을 포함하고,

상기 제1 클라이언트는, 피어-투-피어 네트워크를 통해 상기 공유된 객체를 액세스함으로써, 상기 매니페스트 파일에서 식별된 상기 공유된 객체의 인스턴스를 또한 액세스하고 있는 적어도 하나의 클라이언트와 동기적으로 직접 통신하고,

상기 제1 클라이언트는, 제1 서버를 통해 상기 공유된 객체를 액세스함으로써, 상기 제1 서버를 통해 상기 공유된 객체를 또한 액세스하고 있는 적어도 하나의 클라이언트와 비동기적으로 통신하고,

상기 피어-투-피어 네트워크가 이용가능하지 않을 때에 상기 공유된 객체는 상기 제1 서버를 통해 액세스가능하며,

상기 동기적으로 직접 통신하는 것은 상기 제1 클라이언트가 상기 제1 서버로부터 연결이 끊어져 상기 적어도 하나의 클라이언트와 동기적으로 직접 계속해서 통신하는 것을 허용하는 시스템.

청구항 7

제6항에 있어서, 상기 제1 클라이언트는 상기 공유된 객체의 고유한 위치 식별자에 의해 식별된 위치로부터 상기 매니페스트 파일을 가져오는 시스템.

청구항 8

제6항에 있어서, 상기 제1 클라이언트가 상기 적어도 하나의 클라이언트와 동기적으로 직접 통신할 때에, 상기 제1 서버를 통해 상기 공유된 객체를 액세스하도록 인가된 다른 클라이언트가 식별되는 시스템.

청구항 9

제6항에 있어서, 상기 제1 클라이언트는 제2 서버를 통해 상기 공유된 객체를 액세스하는 시스템.

청구항 10

제9항에 있어서, 상기 제1 클라이언트는 상기 매니페스트 파일에서 식별된 상기 공유된 객체의 버전을 또한 액세스하고 있는 적어도 하나의 클라이언트와 상기 피어-투-피어 네트워크를 구축함으로써 동기적으로 통신하고, 상기 피어-투-피어 네트워크는 클라이언트들 간의 직접 통신을 허용하는 시스템.

청구항 11

제10항에 있어서, 제1 피어-투-피어 네트워크가 구축된 이후에 상기 제1 클라이언트는 상기 제2 서버로부터 연결이 끊어지는 시스템.

청구항 12

제6항에 있어서,

상기 제1 클라이언트는 상기 제1 서버에 연결되어 비동기적으로 통신하는 시스템.

청구항 13

제12항에 있어서,

상기 제1 클라이언트가 상기 제1 서버로 연결된 이후에, 상기 제1 클라이언트는 상기 피어-투-피어 네트워크로부터 연결이 끊어지는 시스템.

청구항 14

동기적 통신 모드와 비동기적 통신 모드 사이에서 전이하여 복수의 사용자 개정들을 공유된 객체에 동기화시키기 위한 방법을 수행하는 컴퓨터 실행가능 명령어들이 저장된 컴퓨터 판독가능 기록 매체로서,

상기 방법은,

공유된 객체를 액세스하는 단계;

상기 공유된 객체와 연관된 매니페스트 파일을 가져오는 단계 - 상기 매니페스트 파일은 상기 공유된 객체의

고유한 위치 식별자에 의해 식별된 위치로부터 가져 와지고, 상기 매니페스트 파일은 상기 공유된 객체의 다른 인스턴스가 저장되어 있는 위치를 식별함 - ;

상기 매니페스트 파일에서 식별된 상기 공유된 객체의 인스턴스를 또한 액세스하고 있는 적어도 하나의 클라이언트와 동기적으로 직접 제1 클라이언트에 의해 통신하는 단계 - 상기 동기적으로 직접 통신하는 단계는 피어-투-피어 네트워크를 통해 상기 공유된 객체를 액세스하는 단계를 포함함 - ; 및

제1 서버를 통해 상기 공유된 객체를 또한 액세스하고 있는 적어도 하나의 클라이언트와 비동기적으로 상기 제1 클라이언트에 의해 통신하는 단계 - 상기 비동기적으로 통신하는 단계는 상기 제1 서버를 통해 상기 공유된 객체를 액세스하는 단계를 포함함 -

를 포함하고,

상기 피어-투-피어 네트워크가 이용가능하지 않을 때에 상기 공유된 객체는 상기 제1 서버를 통해 액세스가능하고,

상기 동기적으로 직접 통신하는 단계는 상기 제1 클라이언트가 상기 제1 서버로부터 연결이 끊어져 상기 적어도 하나의 클라이언트와 동기적으로 직접 계속해서 통신하는 것을 허용하는 컴퓨터 판독가능 기록 매체.

청구항 15

제14항에 있어서, 상기 매니페스트 파일을 가져오는 단계는 상기 공유된 객체의 고유한 위치 식별자에 의해 식별된 위치로부터 매니페스트 파일을 가져오는 단계를 더 포함하는 컴퓨터 판독가능 기록 매체.

청구항 16

제14항에 있어서, 상기 동기적으로 직접 통신하는 단계는 제2 서버를 통해 상기 공유된 객체를 액세스하도록 인가된 클라이언트들을 식별하는 단계를 더 포함하는 컴퓨터 판독가능 기록 매체.

청구항 17

제14항에 있어서,

상기 방법은, 제2 서버 및 상기 피어-투-피어 네트워크 중 하나로부터 연결을 끊는 단계를 더 포함하는 컴퓨터 판독가능 기록 매체.

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0019] 파일 공유 애플리케이션은 여러 다른 사용자들이 공동으로 정보를 공유하게 해준다. 여러 클라이언트들이 서버상에서 공유된 파일을 비동기적으로 액세스할 수 있다. 한 클라이언트가 또 다른 클라이언트와 피어-투-피어 네트워크를 구축하여, 양쪽 클라이언트가 동기적으로 직접(즉, 실시간으로) 공유된 파일을 액세스할 수 있다. 클라이언트는 피어-투-피어 네트워크 또는 연결이 상실되는 경우 서버 상의 공유 파일을 액세스할 수 있을 것이다. 서버로의 연결은 서버가 충돌하는(crashes) 경우, 또는 클라이언트가 서버로부터 연결이 끊어

진 경우 상실될 수 있다. 피어-투-피어 네트워크로의 연결은 두 클라이언트가 더 이상 같은 부근에 있지 않을 경우, 또는 클라이언트들 중 한쪽이 피어-투-피어 네트워크로부터 연결이 끊어진 경우 상실될 수 있다. 이 중 어느 한 경우에, 클라이언트는 공유된 파일로의 액세스를 다시 획득하기 위해, 수동으로 피어-투-피어 네트워크 또는 서버에 재연결되어야 한다.

발명이 이루고자 하는 기술적 과제

[0020] 본 발명은 동기적 통신 모드와 비동기적 통신 모드 사이를 전이하기위한 방법 및 시스템에 관한 것이다. 수많은 다른 사용자들이 여러 다른 트랜스포트를 통해 동일한 공유된 객체를 동시에 액세스하고, 개정하고, 갱신할 수 있다. 객체는 파일 등의 공유될 수 있는 임의의 개체일 수 있다. 클라이언트들은 물리적인 서버를 통해 그 공유된 객체를 액세스함으로써 비동기적으로 통신할 수 있다. 서버 한계(limitations)로 인해 공유된 객체의 동기화는 개정들(revisions)이 사용자에 의해 실행된 시기보다 지연되게 된다. 클라이언트들은 또한 피어-투-피어 네트워크를 통해 공유된 객체를 액세스함으로써 동기적으로 통신할 수 있다. 피어-투-피어 네트워크에 의해 공유된 객체로의 개정들이 서버를 통하지 않고 클라이언트들 간에 직접 전송되게 된다.

[0021] 각각의 공유된 객체는 매니페스트 파일(manifest file)에 연관된다. 매니페스트 파일은 공유된 객체의 다른 버전들 및 인스턴스들이 시스템 내에 저장되어 있는 위치들(locations)을 식별한다. 클라이언트는 공유된 객체를 액세스하고, 그 연관된 매니페스트 파일을 해당하는 공유된 객체의 고유한 위치 식별자에 의해 식별되는 위치로부터 가져온다. 공유된 객체 및 매니페스트 파일에 의해 클라이언트는 로컬 액세스, 동기적 통신, 및 비동기적 통신 사이를 자연스럽게 전이할 수 있어서, 사용자는 통신 모드가 변경되었음을 인식하지 못한다. 사용자는 위치를 변경할 수 있으며, 임의 이용가능한 데이터 공유 트랜스포트들은 자동으로 식별된다. 그러므로, 사용자는 공유된 객체를 액세스하고, 여러 가지 메커니즘을 통해 다른 인가된 사용자들과 협력할 수 있다.

[0022] 클라이언트는 서버상에서 공유된 객체를 액세스할 수 있다. 그 클라이언트는 그 공유된 객체를 또한 액세스하고 있는 다른 클라이언트들에 자동으로 연결된다. 공유된 객체에 액세스하도록 인가된 다른 클라이언트들은 피어 그룹 내에 포함된다. 클라이언트는 공유된 객체에 연관된 매니페스트 파일을 가져온다. 매니페스트 파일은 공유된 객체의 다른 버전들 및 인스턴스들의 위치들을 식별한다. 그러므로, 클라이언트는, 피어 그룹 내의 임의의 클라이언트가 매니페스트 파일 내에서 식별된 공유된 객체의 버전 또는 인스턴스를 액세스할 경우, 피어 그룹 내의 임의의 다른 클라이언트와 피어-투-피어 네트워크를 구축할 수 있다. 그러면, 클라이언트는 서버로부터의 연결이 끊어질 수 있으나, 피어-투-피어 네트워크상에서 공유된 객체를 계속해서 액세스할 수 있다.

[0023] 클라이언트는 피어-투-피어 네트워크로부터 공유된 객체를 액세스할 수 있다. 클라이언트는 공유된 객체에 연관된 매니페스트 파일을 가져온다. 매니페스트 파일은 공유된 객체가 존재하는 다른 위치들을 식별한다. 클라이언트는 서버상에서 공유된 객체를 액세스할 수 있고, 어떤 인가된 클라이언트들이 또한 서버에 연결되어 있는지를 판정할 수 있다. 연결된 클라이언트들은 피어-투-피어 네트워크가 이용 불가능할 경우 서버를 통해 액세스될 수 있다.

[0024] 본 발명의 한 양상으로, 공유된 객체를 액세스한다. 공유된 객체에 연관된 매니페스트 파일을 가져온다. 매니페스트 파일은 공유된 객체의 또 다른 인스턴스가 저장되어 있는 위치를 식별한다. 직접적인 동기적 통신(direct synchronous communication)은 매니페스트 파일 내에서 식별된 공유된 객체의 인스턴스를 또한 액세스하고 있는 적어도 한 클라이언트와 확립된다. 비동기적 통신은 서버상에서 공유된 객체를 또한 액세스하고 있는 적어도 한 클라이언트와 확립된다.

발명의 구성 및 작용

[0025] 본 발명은 동기적 통신 모드와 비동기적 통신 모드 사이를 전이하기위한 방법 및 시스템에 관한 것이다. 수많은 다른 사용자들은 여러 다른 트랜스포트를 통해 동일한 공유된 객체를 동시에 액세스하고, 개정하고, 갱신할 수 있다. 클라이언트들은 물리적인 서버를 통해 공유된 객체를 액세스함으로써 비동기적으로 통신할 수 있다. 클라이언트는 또한 피어-투-피어 네트워크를 통해 공유된 객체를 액세스함으로써 동기적으로 통신할 수 있다.

[0026] 각각의 공유된 객체는 매니페스트 파일에 연관된다. 매니페스트 파일은 공유된 객체의 다른 버전들 및 인스턴스들이 시스템 내에 저장되어 있는 위치들을 식별한다. 클라이언트는 공유된 객체를 액세스하고, 그 연관된 매니페스트 파일을해당하는 공유된 개체의 고유한 위치 식별자에 의해 식별되는 위치로부터 가져온다. 공

유된 객체 및 매니페스트 파일에 의해 클라이언트는 로컬 액세스, 동기적 통신, 및 비동기적 통신 사이를 자연스럽게 전이할 수 있어서, 사용자는 통신 모드가 변경되었음을 인식하지 못한다. 사용자는 위치를 변경할 수 있으며, 임의 이용가능한 데이터 공유 트랜스포트들이 자동으로 식별된다. 그러므로, 사용자는 공유된 객체를 액세스하고, 여러 가지 메커니즘을 통해 다른 인가된 사용자들과 협력할 수 있다.

[0027] 예시적인 운영 환경

[0028] 도 1을 참조하면, 본 발명을 구현하기위한 한 예시적인 시스템이 컴퓨팅 장치(100) 등의 컴퓨팅 장치를 포함한다. 컴퓨팅 장치(100)는 클라이언트, 서버, 모바일 장치, 또는 네트워크 기반의 협력 시스템에서 데이터를 교환하는 임의 기타 컴퓨팅 장치로서 구성될 수 있다. 가장 기본적인 구성으로, 컴퓨팅 장치(100)는 전형적으로 적어도 하나의 프로세싱 유닛(102)과 시스템 메모리(104)를 포함한다. 컴퓨팅 장치의 그 구성과 타입에 따라, 시스템 메모리(104)는 휘발성(RAM 등), 비휘발성(ROM, 플래시 메모리 등) 또는 이 둘의 일부 결합이 될 수 있다. 시스템 메모리(104)는 전형적으로 운영 체제(105), 및 하나 이상의 애플리케이션(106)을 포함하며, 프로그램 데이터(107)를 포함할 수 있다. 하기에 상세히 기술된 통신 모드 전이 모듈(108)은 애플리케이션(106) 내에 구현된다.

[0029] 컴퓨팅 장치(100)는 부가적인 특징 또는 기능을 가질 수 있다. 예를 들면, 컴퓨팅 장치(100)는 예컨대, 자기 디스크, 광 디스크, 또는 테이프 등의 부가적인 데이터 저장 장치들(분리형 및/또는 비분리형)도 포함할 수도 있다. 이런 부가적인 저장장치는 분리형 저장장치(109) 및 비분리형 저장장치(110)에 의해 도 1에서 예시된다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 기타 데이터 등의 정보를 저장하기 위한 임의 방법 또는 기술로 구현되는 휘발성 혹은 비휘발성, 분리형 혹은 비분리형 매체를 포함할 수 있다. 시스템 메모리(104), 분리형 저장장치(109), 및 비분리형 저장장치(110)는 컴퓨터 저장 매체의 모든 예들이다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리, 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disks), 또는 기타 광 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장장치 또는 기타 자기 저장 장치들, 또는 컴퓨팅 장치(100)에 의해 액세스될 수 있고, 원하는 정보를 저장하는데 사용될 수 있는 임의 기타 매체를 포함하나, 이에만 한정되지 않는다. 컴퓨팅 장치(100)는 또한 키보드, 마우스, 펜, 음성 입력 장치, 터치 입력 장치 등의 입력 장치(들)(112)을 가질 수 있다. 디스플레이, 스피커, 프린터 등의 출력 장치(들)(114)이 또한 포함될 수 있다.

[0030] 컴퓨팅 장치(100)는 네트워크 등을 통해, 그 장치가 기타 컴퓨팅 장치(118)와 통신하게 하는 통신 연결(116)도 포함한다. 네트워크는 LAN 및 WAN을 포함할 뿐 아니라, 인트라넷 및 익스트라넷을 비롯하여 기타 대규모의 네트워크를 포함하나 이에만 한정되지 않는다. 통신 연결(116)은 통신 매체의 한 예이다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어들, 데이터 구조들, 프로그램 모듈들, 또는 기타 데이터를 반송파 또는 기타 전송 메커니즘 등의 변조된 데이터 신호에 구현한 것이며, 임의의 정보 전달 매체를 포함한다. "변조된 데이터 신호"란 용어는 신호의 하나 이상의 특성을 그 신호에 실린 정보를 부호화하는 방식으로 설정 또는 변경시킨 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 유선에 의한 직접 연결과 같은 유선 매체와, 음향, RF, 적외선, 및 기타 무선 매체 등의 무선 매체를 포함하나, 이에만 한정되지는 않는다. 본원에 사용된 바와 같은 컴퓨터 판독가능 매체는 저장 매체 및 통신 매체 모두를 포함한다.

[0031] 동기적 및 비동기적 통신 모드 사이를 전이하기

[0032] 도 2는 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 시스템의 블록도를 예시한다. 객체는 파일 등의 공유될 수 있는 임의의 객체일 수 있다. 시스템은 클라이언트(200, 210, 220, 및 230), 교환 서버(240), 웹 서버(250), 피어-투-피어 네트워크(260) 등의 파일 저장 능력을 가진 이메일 서버, 및 이메일 첨부(270)를 포함한다. 클라이언트(200 및 210)는 교환 서버(240)에 연결된다. 클라이언트(210 및 220)는 웹 서버(250)에 연결된다. 클라이언트(210 및 220)는 또한 피어-투-피어 네트워크(260)를 통해 함께 연결된다. 이메일 첨부(270)는 웹 서버(250)에 의해 클라이언트(230)로/로부터 전송되도록 배치된다. 클라이언트(200 및 210)는 모두 동일한 사용자(사용자 1)에 연관된다. 예를 들면, 클라이언트(200)는 사용자 1에 의해 집에서 액세스되고, 클라이언트(210)는 사용자 1에 의해 직장에서 액세스된다. 클라이언트(220 및 230)는 다른 사용자(각각, 사용자 2 및 사용자 3)에 연관된다. 클라이언트(200, 210, 220, 및 230)는 공유된 객체를 지역적으로 저장하는 캐시(cache)(202, 212, 222, 및 232)를 각각 포함한다. 피어-투-피어 네트워크(260)는 클라이언트들(210 및 220) 간에 공유된 객체를 전송하기위한 가상 서버(262)를 포함한다. 개정 파일(232) 및 공유된 객체들(252, 264, 및 272)은 교환 서버(240), 웹 서버(250), 가상 서버(262), 및 이메일 첨부(270)에 각각 저장되어 있다. 개정 파일(242) 및 공유된 객체들(252, 264, 및 272)은 피어 그룹 식별자에 연관될 수 있다. 피어 그룹 식별자는 특정 공유된 객체를 액세스하고 개정하도록 인가된 사용자들(즉, 피어 그룹)을 식별한다.

일 실시예에서, 피어 그룹 식별자는 임의의 웹 클라이언트로 결정될 수 있는 피어 그룹으로의 URL(uniform resource locator)일 수 있다. 공유된 객체(252 및 264)는 매니페스트 파일(254 및 266)에 각각 연관된다.

[0033] 수많은 다른 사용자들은 동일한 공유된 객체를 여러 다른 트랜스포트를 통해 동시에 액세스하고, 편집하고, 갱신할 수 있다. 예를 들면, 클라이언트(210)의 사용자 1, 클라이언트(220)의 사용자 2는 공유된 객체(252)를 웹 서버(250)로부터 액세스할 수 있다. 공유된 객체는 대응하는 캐시(212, 및 222)에 지역적으로 저장되어 있다. 사용자 1과 사용자 2 모두는 공유된 객체(252)를 개정할 수 있다. 개정들은 웹 서버(250) 상에서 공유된 객체(252)와 동기화되어, 사용자 1은 사용자 2에 의해 행해진 개정들을 알 수 있고, 사용자 2는 사용자 1에 의해 행해진 개정들을 알 수 있다.

[0034] 다른 예에서, 사용자 3은 공유된 객체(272)에 대한 액세스를 사용자 2와 함께 이메일 첨부(270)를 통해 공유할 수 있다. 사용자 2는 지역적으로 저장된 공유된 객체를 개정할 수 있고, 전체 공유된 객체 또는 단지 첨부된 공유된 객체로의 개정들만을 가진 이메일 메시지를 사용자 3에게 전송할 수 있다. 사용자 2에 의해 행해진 개정들은 웹 서버(250) 상에서 공유된 객체(252)와 동기화된다. 그 이메일이 클라이언트(230)에 수신될 때, 사용자 2에 의해 행해진 개정들은 캐시(232)에 저장된 지역적인 공유된 객체와 자동으로 동기화된다. 그러면 사용자 3은 공유된 객체(272)에 대한 세부 개정들을 행할 수 있고, 전체 공유된 객체 또는 단지 이메일 첨부(270)로서 포함된 공유된 객체로의 개정들만을 가진 사용자 2에게 응답할 수 있다. 사용자 3에 의해 행해진 개정들은 웹 서버(250)상에서 공유된 객체(252)와 동기화된다. 클라이언트(220)의 공유된 객체는 또한 사용자 3에 의해 행해진 개정들을 포함하도록 갱신된다.

[0035] 다른 실시예에서, 사용자 1은 클라이언트(200)의 집 또는 클라이언트(210)의 직장 중 어느 한 곳에서 교환 서버(240)를 통해 공유된 객체를 액세스할 수 있다. 교환 서버는 외부 서버로의 액세스가 허용되지 않거나 이용가능하지 않을 경우 종종 이용된다. 개정 파일(242)은 공유된 객체로의 개정들을 포함한다. 개정 파일(242)은 USB(a universal serial bus) 드라이브, 이메일 애플리케이션, 또는 개정들이 오갈 수 있게 전송되도록 하는 일부 기타 메커니즘을 통해 클라이언트들(200 및 210) 간에 전송될 수 있다. 개정들을 클라이언트들(200, 및 210)에 적용하여서, 캐시(202, 및 212)에 저장된 지역적인 공유된 객체를 갱신할 수 있다.

[0036] 교환 서버(240)는 서버가 처리할 수 있는 파일들의 크기에 관한 제약(예컨대, 2 Mbytes)을 가질 수 있다. 사용자 1은 클라이언트(200)의 공유된 객체에 대한 임의 개정들을 포함하는 개정 파일(242)을 교환 서버(240)에 업로드할 수 있다. 개정 파일(242)이 교환 서버(240)의 크기 제약을 초과하는 경우, 개정 파일(242)은 부분 섹션(subsections)들로써 클라이언트(200)에서부터 클라이언트(210)로 전송될 수 있다. 파일 프로토콜은 부분섹션들을 전송하기 위한 요청/충족(request/fill) 프로세스를 허용한다. 일 실시예에서, 교환 서버(240)는 이메일 애플리케이션에 연관된다. 또 다른 사용자(사용자 2)에 의해 행해진 개정들은 클라이언트(220)에서부터 클라이언트(210)로 웹 서버(250) 또는 피어-투-피어 네트워크를 통해 전송된 다음에, 사용자 1에게 발행된 이메일 계정을 통해 클라이언트(200)에게 전송될 수 있다. 다른 실시예에서, 클라이언트(200)는 현재의 개정 파일을 위해 교환 서버(240)를 주기적으로 폴링한다(poll).

[0037] 다른 예에서, 피어-투-피어 네트워크(260)는 클라이언트들(210 및 220)에서부터 웹 서버(250)로의 연결이 상실될 경우, 또는 사용자 1과 사용자 2가 실시간으로 동기적으로 직접 통신하는 것을 선호할 경우, 클라이언트들(210 및 220) 간에 구축될 수 있다. 사용자 1과 사용자 2는 피어-투-피어 네트워크(260)를 통해 통신하길 선호할 수 있는데, 왜냐하면 웹 서버(250)를 통한 객체의 공유가 개정들이 클라이언트(210)에서 행해지는 시점과 그 개정들이 클라이언트(220)에서 이용가능하게 되는 시점 간에 지연(lag)을 초래할 수 있기 때문이다. 이 지연은 과도한 서버 트래픽의 결과일 수 있다. 피어-투-피어 네트워크(260)는 공유된 객체로의 개정들이 웹 서버를 통하지 않고 클라이언트들(210 및 220) 간에 직접 전송되도록 해준다. 일 실시예에서, 피어-투-피어 네트워크(260)는 TCP/IP(a transmission control protocol/internet protocol) 다이렉트 네트워크이다. TCP/IP 다이렉트 네트워크는 개정들을 신속히 저장하거나 가져올 수 있게 한다.

[0038] 클라이언트들(210 및 220)은 웹 서버(250) 연결이 방해받을 경우, 캐시(212 및 222)에 지역적으로 저장된 공유된 객체(252)의 복제본을 각각 가질 수 있다. 공유된 객체(252)에 연관된 피어 그룹 식별자는 사용자 1과 사용자 2 모두가 공유된 객체를 동시에 액세스하고 있다는 것을 나타낸다. 그 사용자들은 그들 모두가 구축된 피어-투-피어 네트워크를 액세스하는 경우(예컨대, 양 사용자가 같은 비행기 내의 랩톱 컴퓨터들 상에서 작업하는 경우) 서로에 대해 잘 알게 된다. 피어-투-피어 네트워크(260)에 의해 사용자 1과 사용자 2는 가상 서버(262)상에서 공유된 객체(264)로의 개정들을 동시에 액세스하게 되고 세부 개정들을 구현하게 된다. 그 개정들은 즉시 클라이언트들(210 및 220)에 복제되어서, 사용자 1과 사용자 2는 공유된 객체(264)에 관해 활동적으로 협력할 수가 있다. 피어-투-피어 네트워크(260)는 사용자 1과 사용자 2가 더 이상 동일한 부근에

존재하지 않을 경우(예컨대, 각각의 사용자가 그들 각자의 사무실로 돌아갈 경우), 또는 사용자 1과 사용자 2가 더 이상 실시간으로 통신하길 원하지 않는 경우, 디스에이블될 수 있다. 그러면 공유된 객체(252)는 웹 서버로부터 액세스될 수 있다. 피어-투-피어 네트워크(260)와 웹 서버(250) 상에서의 공유된 객체 개정들의 액세스 간의 전이는 자동적이며 자연스럽게 행해진다.

[0039] 클라이언트들(210 및 220)은 웹 서버(250)와 피어-투-피어 네트워크(260) 모두로부터 현재의 개정들을 수신할 수 있다. 공유된 객체에 행해진 각각의 개정은 GUID(a global unique identifier)와 타임 스탬프(time stamp)에 연관된다. 타임 스탬프는 개정이 행해졌던 시각을 식별한다. 클라이언트(210)는 웹 서버(250)상에서 공유된 객체(252)를 수정할 수 있다. 클라이언트(220)는 캐싱된 객체에 연관된 GUID와 타임 스탬프를 웹 서버(250)상에서 공유된 객체(252)에 연관된 GUID와 타임 스탬프와 비교함으로써 캐시(222)에 있는 그 공유된 객체의 지역적인 버전이 현재의 것인지를 판정한다. 현재의 버전이 지역적으로 저장되어 있지 않으면, 캐싱된 객체에 구현되지 않았던 최근 개정들이 웹 서버(250)에서부터 클라이언트(220)로 로딩되고, 지역적인 파일(local file)과 동기화된다. 그러므로, 전체 공유된 객체는 공유된 객체의 각각의 지역적인 버전을 갱신할 때마다 클라이언트(220)로 로딩될 필요는 없다.

[0040] 일 실시예에서, 클라이언트(220)는 동일한 수정들이 피어-투-피어 네트워크(260)로부터 이용가능하다는 것을 개정에 연관된 GUID와 타임 스탬프로부터 판정할 수 있다. 그러나, 클라이언트(220)가 이미 수정사항을 실행하였기 때문에, 어떤 동작도 초래되지는 않는다. 다른 실시예에서, 동일한 수정들이 피어-투-피어 네트워크(260)로부터 이용가능하지 않다는 것을 개정에 연관된 GUID와 타임 스탬프로부터 판정할 수 있다. 그러므로, 클라이언트(220)는 개정들을 피어-투-피어 네트워크(260)에 제출하여서, 그 피어-투-피어 네트워크(260)에 연결된 다른 사용자들이 공유된 객체의 현재의 버전과 동기화될 수 있도록 한다.

[0041] 클라이언트(220)는 피어-투-피어 네트워크(260)로부터 또 다른 개정들을 수신할 수 있다. 캐시(222)에 있는 공유된 객체를 갱신한다. 클라이언트(220)는 공유된 객체의 현재의 상태가 웹 서버(250)상에서 이용가능한지 여부를 개정에 연관된 GUID와 타임 스탬프를 사용하여 판정한다. 웹 서버(250)상에서 공유된 객체(252)가 공유된 문서의 현재의 상태와 동기화되지 않은 경우, 클라이언트(220)는 최근 개정을 웹 서버(250)에 제출하여서, 공유된 객체(252)가 갱신될 수 있도록 한다.

[0042] 비동기적 통신은 클라이언트가 물리적인 서버를 통해 시스템에 연결되어 있으면서 공유된 객체를 개정하는 경우, 발생할 수 있다. 서버 한계로 인해 공유된 객체 동기화는 개정들이 사용자에 의해 구현되는 시간보다 지연되게 된다. 일 실시예에서, 클라이언트는 시스템에 연결되지 않으면서 공유된 객체의 지역적으로 캐싱된 버전을 개정할 수가 있다. 클라이언트에 의해 행해진 임의 개정들은 클라이언트가 서버를 통해 시스템에 재연결될 경우 공유된 객체와 동기화될 수 있다. 클라이언트는 로컬 액세스, 동기적 통신, 및 비동기적 통신 사이를 자연스럽게 전이할 수 있어서, 사용자는 통신 모드가 변경되었음을 인식하지 못한다. 사용자는 위치를 변경할 수 있으며, 임의 이용가능한 데이터 공유 트랜스포트들(예컨대, 피어-투-피어 네트워크, 서버 등)은 식별된다. 그러므로, 사용자는 공유된 객체를 액세스하고, 다른 메커니즘을 통해 다른 인가된 사용자들과 협력할 수 있다.

[0043] 각각의 공유된 객체는 매니페스트 파일에 연관된다. 매니페스트 파일은 공유된 객체의 다른 버전들 및 인스턴스들이 시스템 내에 저장되어 있는 위치들을 식별한다. 일 실시예에서, 매니페스트 파일은 XML(extensible markup language) 파일이다. 다른 실시예에서, 매니페스트 파일은 복수의 공유된 객체들을 식별한다. 다른 실시예에서, 매니페스트 파일은 클라이언트 간에 공유될 수 있는 임의 객체에 연관될 수 있다. 예를 들면, 매니페스트 파일은 전체 공유된 객체 또는 공유된 객체의 일부분(예컨대, 콘텐츠 컨테이너, 섹션, 페이지, 아웃라인 등)에 연관될 수 있다.

[0044] 매니페스트 파일은 시스템 내의 어느 곳이라도 저장될 수 있다. 도면에 도시된, 매니페스트 파일(254)은 공유된 객체(252)에 연관된다. 공유된 객체(252) 및 매니페스트 파일(254)는 웹 서버(250)상에 저장된다. 다른 실시예에서, 매니페스트 파일은 공유된 객체에 저장된다. 다른 실시예에서, 매니페스트 파일은 활성 디렉토리에 저장된다. 다른 실시예에서, 매니페스트 파일은 시스템 내의 복수의 위치들에 저장된다. 매니페스트 파일은 고유한 위치 식별자에 의해 식별된 위치에 저장된다. 고유한 위치 식별자는 파일 서버, 서버의 공유된 영역, 웹 서버, 또는 피어 그룹을 식별할 수 있다.

[0045] 공유된 객체는 서버 또는 피어-투-피어 네트워크를 통해 지역적으로 캐시로부터 액세스될 수 있다. 클라이언트는 그 매니페스트 파일을 해당하는 공유된 객체의 고유한 위치 식별자에 의해 식별된 위치로부터 가져온다. 일 실시예에서, 클라이언트는 나중 참조를 위해 매니페스트 파일을 지역적으로 저장할 수 있다. 매니페스트

파일은 공유된 객체의 임의 다른 버전들 및 인스턴스들의 위치들을 시스템(예컨대, 부분저장소 또는 피어 그룹) 내에서 식별한다. 공유된 객체의 다른 버전/인스턴스가 피어 그룹 내에 저장되는 경우, 매니페스트 파일은 해당하는 피어 그룹 식별자를 포함할 수 있다.

[0046] 일 실시예에서, 클라이언트(220)는 웹 서버상에서 공유된 객체(252)를 액세스한다. 클라이언트(220)는 공유된 객체(252)를 또한 액세스하고 있는 다른 클라이언트들(예컨대, 피어 그룹)에 자동으로 연결된다. 클라이언트(220)는 공유된 객체(252)에 연관된 매니페스트 파일(254)을 가져온다. 매니페스트 파일(254)은 공유된 객체의 다른 버전들 및 인스턴스들의 위치들을 식별한다. 그러므로, 피어 그룹 내의 임의 클라이언트가 매니페스트 파일(254)에 의해 식별된 공유된 객체(252)의 버전/인스턴스를 액세스할 경우, 클라이언트(220)는 피어 그룹 내의 임의 다른 클라이언트와 피어-투-피어 네트워크를 구축할 수 있다. 그러면 클라이언트(220)는 웹 서버(250)로부터 연결이 끊어질 수 있으나, 피어-투-피어 네트워크상에서 공유된 객체(252)를 계속해서 액세스할 수 있다.

[0047] 다른 실시예에서, 클라이언트(210)는 공유된 객체(264)를 피어-투-피어 네트워크로부터 액세스할 수 있다. 클라이언트(210)는 공유된 객체(264)에 연관된 매니페스트 파일(266)을 가져온다. 클라이언트(210)는 서버에 접속될 수 있고, 또한 어떤 클라이언트가 또한 그 서버에 연결되어 있는지를 판정할 수 있다. 접속된 클라이언트들은 피어-투-피어 네트워크(260)가 이용불가능할 경우 서버를 통해 액세스될 수 있다. 공유된 객체(264)(또는 252) 및 연관된 매니페스트 파일(264)(또는 254)에 의해 클라이언트(210)(또는 220)는 비동기적 통신 모드와 동기적 통신 모드 간의 자연스럽게 자동으로 이동할 수가 있다.

[0048] 사용자들은 다른 사용자가 공유된 객체를 액세스하는 경우 그 공유된 객체를 액세스하고 개정하지 못하도록 저지당하지 않는다. 임의 인가된 사용자들이 공유된 객체를 동시에 개정할 수 있다. 일 실시예에서, 간단한 블록킹 인스턴스가 개정 트랜잭션의 무결정을 보장하기 위해 생성될 수 있다. 예를 들어, 사용자는 서버로부터 연결이 끊긴 상태에서 공유된 문서를 광범위하게 개정할 수 있다. 사용자가 서버에 재연결될 경우, 그 사용자의 개정 전부가 공유된 객체에 구현될 때까지 다른 클라이언트들은 공유된 객체를 액세스하지 못하도록 저지당할 수 있다.

[0049] 도 3은 공유된 객체의 다른 일부분을 나타내는 링크된 노느들의 계층구조의 그래프를 예시한다. 일 실시예에서, 공유된 객체는 다수의 사용자 간에 공유된 노트북이다. 노트북 노드(300)는 전체 공유된 객체를 부호로 나타낸다. 폴더 노드(310)는 노트북 노드(300) 내에 포함된다. 섹션 노드(320)는 폴더 노드(310) 내에 포함된다. 페이지 노드(330 및 335)는 섹션 노드(310) 내에 포함된다. 테이블 노드(340), 잉크 노드(342), 아웃라인 노드(344), 및 이미지 노드(346)는 페이지 노드(330) 내에 포함된다. 아웃라인 구성요소 노드(350)는 아웃라인 노드(344) 내에 포함된다. 텍스트 노드(360)는 아웃라인 구성요소 노드(350) 내에 포함된다. 다른 노드들은 콘텐츠 컨테이너에서 함께 그룹지어질 수 있다. 예를 들면, 아웃라인 노드(344), 아웃라인 구성요소 노드(350), 및 텍스트 노드(360)는 콘텐츠 컨테이너 R0로서 함께 그룹지어질 수 있다. 콘텐츠 컨테이너 R0에는 GUID(예컨대, GUID-0)가 지정된다. GUID는 고유하게 콘텐츠 컨테이너 R0을 식별한다.

[0050] 콘텐츠 컨테이너는 공유된 객체 콘텐츠(예컨대, 단어, 문장, 문단, 테이블, 그림, 필적(handwriting), URL, 또는 공유된 객체 내에 포함된 데이터의 임의 조합)를 포함한다. 콘텐츠 컨테이너는 함께 그룹지어진 객체 콘텐츠에 차원(dimension)을 제공한다. 예를 들면, 콘텐츠 컨테이너는 라인, 문단, 페이지, 또는 특정한 페이지 구성요소(예컨대, 단지 특정 페이지의 테이블만)에 해당할 수 있다.

[0051] 공유된 객체는 그래프의 초기 버전을 저장한다. 그러면 특정 연산들이 각각의 콘텐츠 컨테이너에 대해 수행될 수 있다. 예를 들면, 사용자는 콘텐츠 컨테이너의 데이터를 개정할 수 있다. 공유된 객체로의 개정이 그 콘텐츠 컨테이너의 상태로서 식별될 수 있다. 공유된 객체는 그래프의 개정된 콘텐츠 컨테이너들을 저장한다. 콘텐츠 컨테이너의 현재의 상태는 GUID와 타임 스탬프를 사용하여 이전 상태와 비교되어서, 그 콘텐츠 컨테이너가 개정되었는지에 관한 판정이 행해질 수 있다.

[0052] 예를 들면, 2명의 다른 사용자가 공유된 문서를 각각 액세스하고, 콘텐츠 컨테이너 R0을 수정할 수 있다. 한 사용자는 텍스트 노드(360)를 삭제함으로써 콘텐츠 컨테이너 R0을 수정할 수 있다(개정 R1에 도시됨). 개정 R1은 공유된 객체에 저장되어 있다. 개정 R1에는 개정된 컨테이너를 고유하게 식별하기 위한 GUID(예컨대, GUID-1), 및 공유된 객체에 개정 R1을 기록한 일시(the time and date)를 식별하는 타임 스탬프가 지정된다. 다른 사용자는 텍스트 노드(380)를 아웃라인 구성요소 노드(350)에 첨가함으로써 콘텐츠 컨테이너 R0를 개정할 수 있다(개정 R2에 도시됨). 개정 R2는 공유된 객체에 저장되어 있다. 개정 R2에는 타임 스탬프와 개정된 콘텐츠 컨테이너를 고유하게 식별하기 위한 GUID가 지정된다.

[0053] 다른 사용자들은 공유된 객체의 복수의 버전들이 공존할 수 있도록 다른 시간에 공유된 객체를 개정할 수 있다. 그러나, 공유된 객체의 최근 버전(latest version)은 단 하나 존재한다. 일 실시예에서, 공유된 객체의 최근 버전은, 공유된 객체와 동기화되며 다른 인가된 사용자들에게 이용가능하게 된 가장 최근 개정들을 포함하는 버전이다.

[0054] 예를 들면, 사용자는 아웃라인 구성요소 노드(370)를 아웃라인 노드(344)에 첨가함으로써 개정 R1으로서 식별되는 공유된 객체의 콘텐츠 컨테이너를 개정할 수 있다(개정 R3에 도시됨). 개정 R3는 공유된 객체에 저장되어 있다. 개정 R3에는 또한 타임 스탬프와 개정된 콘텐츠 컨테이너를 고유하게 식별하기 위한 GUID(예컨대, GUID-3)가 지정된다. 개정 R3는 개정 R1의 확장이다. 그러므로, 개정 R1은 사용자가 인식하고 있었던 공유된 객체의 최근 버전(예컨대, 지역적으로 저장된 버전)이다. 공유된 객체의 최근 버전이 여전히 개정 R1인지를 판정하기 위해 공유된 객체를 조사한다. 일 실시예에서, 공유된 객체의 최근 버전은 다른 콘텐츠 컨테이너들의 GUID와 타임 스탬프를 비교함으로써 판정될 수 있다. 공유된 객체의 최근 버전이 개정 R1보다 더욱 최근의 타임 스탬프에 연관되어 있을 경우, 다른 사용자(예컨대, 개정 R2를 생성하였던 사용자)가 동일한 콘텐츠 컨테이너를 계속해서 수정했었다는 것이다.

[0055] 개정 R1이 공유된 객체와 동기화된 이후로, 다른 사용자가 동일한 콘텐츠 컨테이너를 수정했던 경우, 개정 R1의 확장인 임의의 개정들(예컨대, 개정 R3)은 임의의 연속된 개정들이 공유된 객체와 동기화되고 임의 모순되는 개정들이 조정되고 병합될 때까지, 그 공유된 객체와 동기화될 수 없다. 예를 들면, 개정 R2가 개정 R1 이후에 공유된 객체와 동기화된다. 그러므로, 공유된 객체의 최근 버전은 개정 R2를 포함한다. 개정 R3가 공유된 객체와 동기화되기 이전에, 개정 R3는 임의 개정들이 모순되는지 여부를 판정하기 위해, 개정 R2와 비교된다. 그 비교는 개정 R3가 개정 R1의 확장이며, R1은 더 이상 공유된 객체의 최근 버전에 연관되지 않기 때문에 필수적이다. 개정 R3는 개정 R2와 모순되지 않도록 결정되는데, 아웃라인 구성요소 노드(370)가 개정 R2를 붕괴시키지 않으면서 아웃라인 노드(344)에 첨가될 수 있기 때문이다.

[0056] 일 실시예에서, 콘텐츠 컨테이너를 공유된 객체 내의 한 위치에서 다른 위치로 이동시킴으로써 공유된 객체를 개정한다. 예를 들면, 테이블 노드(340)를 페이지 노드(330)에서부터 페이지 노드(335)로 이동시킬 수 있다. 테이블 노드(340)가 이동되었으나 새로운 위치를 결정할 수 없다는 판정이 행해진다. 프록시 노드가 테이블 노드(340)의 원래의 위치에 생성된다. 테이블 노드의 새로운 위치가 결정되는 경우, 프록시 노드가 테이블 노드(340)의 새로운 위치에 구현된다. 새로운 위치가 결정되기 전에 테이블 노드(340)는 삭제되는 경우, 프록시 노드는 버려진다.

[0057] 여러 사용자들이 공유된 객체를 동시에 편집할 수 있다. 일반적으로, 사용자들은 공유된 객체의 다른 콘텐츠 컨테이너들을 개정하고 있다. 그러므로, 각각의 사용자의 개정들이 세부 처리없이 공유된 객체와 동기화될 수 있다. 두 명의 사용자가 공유된 객체의 동일한 콘텐츠 컨테이너(예컨대, 동일한 테이블 값, 동일한 문장)를 편집할 경우 모순이 발생할 수 있다. 다른 사용자 개정들 간의 모순은 비동기적으로 초래될 수 있다. 예를 들면, 사용자는 서버에 연결되지 않은 채 지역적으로 캐싱된 공유된 객체의 버전을 개정할 수 있다. 그 개정들은 사용자가 서버에 재연결될 경우 동기화된다. 그러나, 그 개정들은 이미 공유된 객체와 동기화되었던 다른 개정들과 모순될 수 있다.

[0058] 예를 들면, 개정 R4는 개정 R3의 확장이다. 개정 R4는 아웃라인 구성요소 노드(350)를 아웃라인 노드(344)로부터 삭제한다. 공유된 객체의 최근 버전은 개정 R2를 포함하도록 결정된다. 아웃라인 구성요소 노드(350)가 개정 R2에 존재하지만 개정 R4에서는 삭제되었기 때문에, 개정 R2와 개정 R4 간의 비교에 의해 모순을 식별하게 된다.

[0059] 3단계 병합(three-way merge)이 모순을 조정하기 위해 콘텐츠 컨테이너의 마스터 버전과 콘텐츠 컨테이너의 2개의 일치하지 않는 버전들 간에 수행된다. 예를 들면, 콘텐츠 컨테이너 R0(즉, 마스터 버전), 개정 R2, 및 개정 R4는 공유된 객체의 현재의 버전을 설정하기 위해 병합된다. 콘텐츠 컨테이너의 마스터 버전은 서버상에서 공유된 객체와 마지막으로 동기화되었던 버전일 수 있다. 마스터 버전은 모순되지 않는(non-conflicting) 개정들을 포함한다.

[0060] 모순되는 콘텐츠 컨테이너들을 조정하여 병합 알고리즘에 의해 설정되는 일련의 규칙에 따라 공유된 객체로 병합한다. 병합 알고리즘은 어떤 개정들이 공유된 객체와 동기화되는지를 판정한다. 예를 들면, 여러 사용자들은 한 사용자의 개정들(즉, 일차적인 편집들)이 다른 모든 사용자보다 우선하도록 우선순위에 따라 정렬될 수 있다. 낮은 우선순위 사용자가 높은 우선순위 사용자에 의해 이미 개정되었던 공유된 객체의 콘텐츠 컨테이너를 개정하려고 시도하는 경우, 낮은 우선순위 사용자는 개정들(즉, 이차적인 편집들)이 공유된 객체

와 동기화되지 않을 것이란 사실을 알게 된다. 그러므로, 일차적인 편집들이 공유된 객체의 마스터 페이지상에서 디스플레이되고, 이차적인 편집들은 공유된 객체와 동기화되지 않을 것으로 플래그된다(flagged).

[0061] 다른 예에서, 서버상에서 공유된 객체에 행해진 개정들은 클라이언트상에서 지역적으로 행해진 개정들보다 우선 순위를 가진다. 공유된 객체의 서버 복제본은 많은 다른 사용자들이 서버상에서 공유된 객체를 잠재적으로 액세스하고 개정했기 때문에 마스터 버전이 된다. 오직 한 사용자만 지역적으로 저장된 버전을 액세스하고 개정했다. 공유된 객체와 동기화되지 않는 개정된 콘텐츠 컨테이너들(예컨대, 이차적인 편집들)은 모순되는 것으로서 식별된다. 모순되는 콘텐츠 컨테이너들은 공유된 객체의 해당하는 마스터 페이지에 연관된 모순 페이지들에 저장됨으로써 보존된다.

[0062] 도 4는 공유된 객체의 마스터 페이지 및 연관된 모순 페이지를 예시한다. 마스터 페이지(400)는 콘텐츠 컨테이너들(410 및 420) 등의 모순되지 않는 개정들을 포함한다. 임의의 비병합된 모순되는 개정들은 마스터 페이지(400)상에서 모순 지시자(conflict indicator)에 의해 식별된다. 일 실시예에서, 모순 지시자는 드롭 다운 메뉴(430)이다. 드롭 다운 메뉴(430)의 제1 엔트리는 사용자에 의해 생성되는 가장 최근 모순들일 수 있다. 드롭 다운 메뉴(430)의 엔트리는 사용자의 이름 및 대응하는 시간 스탬프를 포함할 수 있다. 드롭 다운 메뉴(430)의 다른 엔트리는 사용자가 발생시켰으나 조정하지 못했던 다른 모순 페이지들을 포함할 수 있다. 드롭 다운 메뉴(430)의 다른 엔트리들은 다른 사용자에 의해 생성되는 모순에 해당할 수 있다. 드롭 다운 메뉴(430)로부터 엔트리를 선택하면 해당하는 모순 페이지를 공유된 객체의 마스터 버전에서 병합되지 않았던 개정들로 사용자의 주의를 끌기 위해 하이라이팅된 모순되는 개정들로 디스플레이할 수 있다. 그러므로, 사용자는 모순들을 조정하고 마스터 페이지(400)와 함께 병합하거나, 모순들이 관련성이 없다고 결정할 수 있다.

[0063] 다른 실시예에서, 모순 지시자는 탭이다. 마스터 페이지는 탭(440)에 연관될 수 있다. 해당하는 모순 페이지들은 또한 탭(440)과는 구별된 탭들에 연관될 수 있다. 예를 들면, 모순 페이지 탭들은 탭(440)에 관해서 들쑥날쑥하거나(indented) 탭(440) 아래에 축소될 수 있다. 특정 사용자에 연관된 비병합된 모순들을 포함하는 모순 페이지는 사용자의 주의가 해당 사용자에 의해 생성된 모순 페이지들로 집중되도록 다른 탭들과는 구별된 탭에 의해서 식별될 수 있다. 그러면 사용자는 해당하는 모순 페이지를 탐색하기 위해 그 탭을 선택할 수 있다.

[0064] 선택된 모순 페이지는 선택된 객체의 해당하는 마스터 페이지와 나란히 디스플레이될 수 있어서, 사용자는 병합된 개정들의 관점에서 임의의 모순되는 개정들을 조정하고 병합할 수 있게 된다. 모순 페이지(450)는 마스터 페이지(400)에 연관된다. 모순 페이지(450)는 임의 모순되는 콘텐츠 컨테이너들이 그 모순이 조정되지 않았음을 나타내기 위해 하이라이팅되어 있다는 점만 제외하면 마스터 페이지(400)와 유사하다. 예를 들면, 콘텐츠 컨테이너(460)는 사용자의 주의를 비병합된 모순으로 돌리기 위해 하이라이팅된 백그라운드와 함께 제공된다. 콘텐츠 컨테이너(460)는 사용자가 콘텐츠 컨테이너(460)의 데이터를 개정하여 모순을 생성하기 이전에, 삭제되고 공유된 객체와 동기화될 수 있었다. 사용자는 개정을 마스터 페이지(400) 상에서 병합하기 위해 콘텐츠 컨테이너(460)를 선택할 수 있다.

[0065] 일 실시예에서, 모순 페이지는 한 특정 사용자에 연관된다. 그러므로, 하나 이상의 모순 페이지는 하나 이상의 사용자가 병합될 수 없는 개정들을 페이지 상에서 행하는 경우에, 마스터 페이지(400)에 연관될 수 있다. 모든 모순되는 개정들이 공유된 객체를 액세스하도록 인가된 모든 사용자를 위해 저장된다. 추측건대 모순 페이지를 액세스하는 사용자는 그 사용자가 이런 모순들을 조정할 수 있도록 해당 사용자에 의해 생성된 모순들에 가장 관심을 갖고 있다. 예를 들면, 사용자 1은 모순 지시자를 선택할 때 자신의 해당하는 모순 페이지를 제공받는다. 사용자는 또한 다른 사용자에 연관된 모순 페이지를 볼 수 있다. 예를 들면, 사용자 1은 사용자 2에 연관된 모순 페이지를 탐색하기 위해 탭(470)을 선택할 수 있다.

[0066] 공유된 객체의 마스터 페이지에 연관된 수많은 모순 페이지들은 시간이 지나면서 누적될 수 있다. 그 시간 주기 동안에, 사용자는 임의 해당하는 모순 페이지들을 무시하면서, 서버상에서 공유된 객체의 마스터 버전과 여러 개정들을 동기화시킬 수 있다. 그러므로, 사용자가 조정하지 못했던 오래된 모순 페이지들은 아마도 더 이상 관련성 없는 것이 된다. 일 실시예에서, 관련성 없다고 식별된 임의 모순 페이지들은 정해진 시간 주기가 경과되고 사용자가 그 시간 주기 동안에 페이지의 개정들을 동기화시킨 이후에, 삭제될 수 있다. 예를 들면, 마스터 페이지에 연관된 가장 최근의 3개의 모순 페이지들은 보존되지만, 임의의 다른 연관된 모순 페이지들이 생성된지 한달 이후에 제거된다.

[0067] 일 실시예에서, 모순 페이지들은 실시간 통신 동안에 생성되지 않는데, 왜냐하면 모순들이 비동기적 통신 동

안보다 자주 발생할 수 있기 때문이다. 대신에, 사용자들은 동일한 콘텐츠 컨테이너를 협력하여 개정할 수 있다. 모든 사용자들이 신속히 그들의 개정들이 구현되었는지 여부를 판정할 수 있기 때문에, 임의 모순들은 즉시 처리될 수 있다. 대안으로서, 사용자는 다른 사용자가 특정 콘텐츠 컨테이너를 개정하고 있음을 통지받게 된다. 사용자는 기타 사용자의 개정들이 완료될 때까지 다른 콘텐츠 컨테이너를 개정하도록 장려된다.

[0068] 도 5는 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 시스템의 블록도를 예시한다. 이 시스템은 클라이언트들(500, 510, 540, 및 550)과 서버들(520 및 530)을 포함한다. 클라이언트(500)는 서버(520 및 530)에 연결된다. 클라이언트(510)는 서버(520)에 연결된다. 클라이언트들(540 및 550)은 서버(530)에 연결된다. 클라이언트(540)는 저장소(542) 및 자식 저장소(544:child store)를 포함한다. 서버(520)는 저장소(522) 및 자식 저장소(524 및 526)를 포함한다. 서버(530)는 저장소(532)를 포함한다. 저장소(532)는 부분저장소(534 및 536:substore)를 포함한다.

[0069] 저장소(522 및 532), 자식 저장소(524 및 526), 부분저장소(534 및 536)는 공유된 객체에 연관된 개정들을 저장할 수 있다. 저장소(522 및 532), 자식 저장소(524 및 526), 부분저장소(534 및 536)는 계층구조를 가진다. 예를 들면, 저장소(522)는 전체 공유된 노트북 문서에 연관될 수 있다. 자식 저장소(524)는 공유된 노트북 문서의 섹션에 연관될 수 있다. 자식 저장소(526)는 공유된 노트북 문서의 페이지에 연관될 수 있다. 일 실시예에서, 단지 최상위 레벨의 공유된 객체의 가장 최근 버전만이 저장소(522)에 포함된다. 저장소(532)는 전체 최상위 레벨의 공유된 객체를 저장할 수 있다. 부분저장소(534 및 536)는 공유된 객체의 일부분에 연관된다. 예를 들면, 부분저장소(534)는 공유된 객체의 섹션에 연관될 수 있고, 부분저장소(536)는 공유된 객체의 다른 섹션에 연관될 수 있다.

[0070] 애플리케이션이 공유된 객체의 특정 콘텐츠 컨테이너의 현재의 버전을 갖고 있지 않다면 그 공유된 객체를 서버(520) 또는 서버(530)에서부터 클라이언트(500)로 로딩할 수 있다. 예를 들면, 클라이언트(500)는 공유된 객체를 저장소(522)로부터 요청한다. 공유된 객체의 이용가능한 가장 최근 버전은 클라이언트(500)에 제공된다. 공유된 객체의 이용가능한 가장 최근 버전은 공유된 객체의 현재의 버전에 해당하지 않을 수 있는데, 왜냐하면 가장 최근 개정 데이터가 해당하는 자식 저장소(526)에서 이용불가능하기 때문이다. 클라이언트(500)가 공유된 객체를 갱신하기 위해 가장 최근 개정 데이터를 요청한다고 나타내기 위해 요청 태그(request tage)가 자식 저장소(526)에 지정된다. 자식 저장소(526)에는 또한 클라이언트(500)가 자식 저장소(526)로부터 개정 데이터를 요청했던 일시를 식별하는 타임 스탬프가 지정될 수 있다. 자식 저장소에는 또한 데이터를 요청했던 클라이언트(예컨대, 클라이언트(500))를 식별하는 GUID가 지정될 수 있다. 요청 태그, 타임 스탬프, 및 GUID는 클라이언트(500)에게 언제 다른 클라이언트가 자식 저장소(526)를 액세스할지를 알려주기 위해 사용된다. 예를 들면, 클라이언트(510)는 가장 현재의 개정 데이터를 가진 자식 저장소(526)를 액세스할 수 있다. 그러므로, 클라이언트(500)는 공유된 객체의 가장 현재의 개정 데이터가 자식 저장소(526)에서 이용가능하다는 것을 알게 된다.

[0071] 클라이언트(500)는 사용자의 집 컴퓨터일 수 있고, 클라이언트(540)는 사용자의 직장 컴퓨터일 수 있다. 서버(530)는 클라이언트들(500 및 540) 간의 개정 파일을 전송하는 교환 서버일 수 있다. 개정 파일은 클라이언트들(500 및 550)에 저장된 공유된 객체를 갱신하는데 사용될 수 있다. 일 실시예에서, 클라이언트(500)는 예정된 크기(예컨대, 2Mbytes)보다 큰 파일을 처리하는데 제약받는다. 예를 들면, 클라이언트(500)는 수신될 수 있는 이메일 메시지들의 크기를 제한하는 이메일 애플리케이션을 포함할 수 있다. 저장소(542)는 최상위 레벨의 공유된 객체에 연관된 개정들을 포함한다. 자식 저장소(544)는 공유된 객체의 콘텐츠 컨테이너에 연관된 개정들을 포함한다.

[0072] 클라이언트(540)는 다른 클라이언트가 데이터 개정 요청을 제출했는지 여부를 판정하기 위해 서버(530)를 폴링할 수 있다. 클라이언트(540)는 요청된 데이터 개정의 최근 버전이 저장소(542) 또는 자식 저장소(544)에서 이용가능할 경우 요청을 충족시킬 수 있다. 클라이언트(540)는 개정 파일의 크기가 클라이언트(500)에 의해 처리될 수 있는 한계보다 작을 경우 요청된 개정 전체를 클라이언트(500)에 전송할 수 있다. 개정 파일의 크기가 한계보다 클 경우, 그 파일을 보다 작은 파일들로 분할할 수 있다. 대안으로서, 이전 요청들을 삭제함으로써 개정 파일의 크기를 줄일 수 있다. 그러면 작은 파일들이 클라이언트(540)로부터 클라이언트(500)로 서버(530)를 통해 전송된다.

[0073] 개정 데이터의 복수의 요청이 서버상에서 대기하고 있을 수 있다. 일 실시예에서, 요청들이 다른 클라이언트들(예컨대, 클라이언트들(500 및 550))로부터 생성될 수 있다. 각각의 요청하는 클라이언트는 여러 가지 파일 크기 한계에 연관되어 있을 수 있다. 예를 들면, 클라이언트(500)는 2Mbyte 보다 작은 파일들로 제한되고, 클라이언트(550)는 20Mbyte까지의 파일들을 처리할 수 있다. 그러므로, 개정 파일이 2Mbyte 보다

클 경우 2가지 요청은 1개의 전송 트랜잭션을 통해 만족될 수 없다. 일 실시예에서, 우선순위 비트가 요청들을 충족시키는 순서를 설정하기 위해 각각의 요청하는 클라이언트에 연관되어 있다.

[0074] 그 요청들은 개정 파일을 클라이언트들(500 및 550)에 동기화시킴으로써 충족된다. 개정 파일을 개정 파일의 크기에 따라 한 트랜잭션으로 혹은 일련의 복수의 트랜잭션을 통해 클라이언트들(500 및 550)에 동기화시킬 수 있다. 각각의 클라이언트(500 및 550)는 개정 파일 전체가 동기화될 때 그 요청이 충족된다고 판정한다. 클라이언트(540)는 요청들이 충족되었기 때문에 요청된 데이터를 제거할 수 있다. 클라이언트(540)는 어떤 부가적인 요청들이 충족되기 위해 대기하고 있는지를 판정하기 위해 나중에 서버(530)를 폴링할 수 있다.

[0075] 도 6은 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 프로세스를 예시하는 동작가능한 흐름도를 예시한다. 프로세스는 시작 블록에서 시작되며, 시작 블록에서 많은 사용자들(즉, 피어 그룹)이 공유된 객체를 동시에 액세스하고 개정하도록 인가받는다. 객체는 파일 등의 공유될 수 있는 임의 개체일 수 있다. 피어 그룹은 피어 그룹 식별자에 의해 식별될 수 있다. 공유된 개체의 다른 버전들은 해당하는 GUID와 타임 스탬프에 의해 식별된다. 타임 스탬프는 공유된 객체가 개정과 마지막으로 동기화되었던 시간을 식별한다.

[0076] 블록(600)으로 이동하여, 사용자는 공유된 객체를 개정한다. 공유된 객체는 서버상에서, 로컬 캐시에서, 또는 피어-투-피어 네트워크에서 개정될 수 있다. 일 실시예에서, 개정은 개정 파일로서 저장된다. 블록(610)으로 진행하면, 개정은 GUID와 타임 스탬프에 연관된다. 타임 스탬프는 사용자가 공유된 객체를 개정하였던 시간을 식별한다.

[0077] 블록(620)으로 진행하여, 공유된 객체의 최근 버전을 위치찾기한다. 공유된 객체의 최근 버전은 공유된 객체와 동기화된 가장 최근 개정들을 포함하고 다른 인가된 사용자들에게 이용가능하게 된 버전이다. 공유된 객체의 최근 버전은 공유된 객체의 다른 버전들에 연관된 타임 스탬프와 GUID로부터 판정될 수 있다.

[0078] 판정 블록(630)으로 이동하여, 어떤 모순되는 개정들이 존재하는지에 관한 판정이 행해진다. 다른 사용자들이 동일한 콘텐츠 컨테이너를 개정하는 경우 개정들이 모순될 수 있다. 모순되는 개정이 존재할 경우 그 개정은 공유된 객체와 동기화될 수 없다. 모순되는 개정들이 존재할 경우, 프로세스는 블록(640)으로 진행하고, 모순되는 개정들이 조정 및 병합된다(도 7을 참조하여 논의됨). 어떤 모순되는 개정도 존재하지 않게 되면, 프로세스는 블록(650)으로 진행하고, 그 개정이 공유된 객체와 동기화되어서 다른 사용자들이 그 개정을 볼 수 있게 된다. 프로세싱은 종료 블록에서 종료된다.

[0079] 도 7은 모순되는, 공유된 객체로의 복수의 사용자 개정들을 조정하고 병합하는 프로세스를 예시하는 동작가능한 흐름도를 예시한다. 프로세스는 시작 블록에서 시작되고, 시작 블록에서 하나 이상의 사용자는 공유된 객체의 동일한 콘텐츠 컨테이너를 개정하였다. 개정된 콘텐츠 컨테이너 중 하나가 공유된 객체와 동기화되는 경우 모순이 발생하여서, 콘텐츠 컨테이너로의 임의 다른 개정들이 동기화될 수 없게 된다.

[0080] 블록(700)으로 이동하여, 모순되는 개정이 모순 페이지상에서 디스플레이된다. 모순 페이지는 모순되는 개정이 하이라이팅되어 동기화된 개정 대신에 디스플레이된다는 점만 제외하면, 대응하는 마스터 페이지와 유사하다.

[0081] 블록(710)으로 진행하여, 모순 지시자가 공유된 객체의 마스터 페이지상에 디스플레이된다. 모순 지시자는 드롭 다운 메뉴, 탭, 또는 사용자에게 모순 페이지가 마스터 페이지용으로 이용가능하다는 것을 알려주기 위한 임의 기타 메커니즘일 수 있다. 특정 사용자에 연관된 모순 페이지용 모순 지시자는 다른 사용자들에 연관된 모순 페이지용 모순 지시자와는 구분될 수 있어서, 현재의 사용자가 자신에 의해 생성된 모순 페이지를 신속히 식별할 수 있게 한다.

[0082] 블록(720)으로 진행하여, 모순 페이지는 모순 지시자를 선택할 때 마스터 페이지와 나란히 디스플레이된다. 사용자는 마스터 페이지의 동기화된 상태와 대응하는 모순 페이지를 제공받는다.

[0083] 블록(730)으로 이동하여, 사용자는 모순되는 개정들을 조정하고 마스터 페이지로 병합한다. 일 실시예에서, 사용자는 콘텐츠 컨테이너가 마스터 페이지와 병합되도록 콘텐츠 컨테이너를 선택할 수 있다. 다른 실시예에서, 사용자는 개정들을 마스터 페이지로 직접 구현할 수 있다. 다른 실시예에서, 사용자는 개정들을 관련성 없는 것으로서 식별할 수 있다.

[0084] 블록(740)으로 진행하여, 관련성 없는 것으로서 식별되는 모순되는 개정들은 제거된다. 일 실시예에서, 모순되는 개정들은 사용자에 의해 관련성 없는 것으로서 식별될 수 있다. 다른 실시예에서, 모순되는 개정들은 관련성 없는 것으로서 자동으로 식별될 수 있다. 예를 들면, 사용자는 임의 해당하는 모순 페이지들을 무시하면서 여러 개정들을 서버상에서 공유된 객체의 마스터 버전과 동기화시킬 수 있었다. 사용자들이 조정하지

못했던 오래된 모순 페이지들을 예정된 시간 주기가 지난 이후에 관련성 없는 것으로서 식별할 수 있다. 그러면 프로세스는 종료 블록에서 종료된다.

[0085] 도 8은 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 프로세스를 예시하는 동작가능한 흐름도를 예시한다. 프로세스는 시작 블록에서 시작되며, 시작 블록에서 공유된 객체의 다른 버전들이 시스템 전역에 걸쳐 다른 위치들에 저장된다. 블록(800)으로 이동하여, 공유된 객체를 저장소로부터 클라이언트로 다운로딩한다.

[0086] 블록(810)으로 진행하여, 공유된 객체가 그 공유된 객체의 현재의 버전인지에 관한 판정이 행해진다. 공유된 객체가 그 공유된 객체의 현재의 버전일 경우, 프로세스는 종료 블록에서 종료된다. 공유된 객체가 그 공유된 객체의 현재의 버전이 아닐 경우, 프로세스는 블록(820)으로 진행한다. 공유된 객체의 콘텐츠 컨테이너로의 가장 최근의 개정이 저장소로부터 이용불가능하기 때문에, 공유된 객체는 현재의 버전일 수 없다.

[0087] 블록(820)으로 진행하여, 클라이언트가 공유된 객체를 갱신하기 위해 가장 최근의 개정 데이터를 요구한다고 나타내기 위해, 요청 태그 및 클라이언트 정보를 저장소에 지정한다. 클라이언트 정보는 요청하는 클라이언트를 식별하는 GUID, 그 클라이언트가 공유된 객체의 현재의 버전을 저장소로부터 요청하였던 시간을 식별하는 타임 스탬프를 포함할 수 있다.

[0088] 블록(830)으로 이동하여, 공유된 객체의 현재의 버전을 저장소에 수신한다. 저장소는 다른 클라이언트가 가장 최근의 개정 데이터를 가진 저장소를 액세스할 경우 공유된 객체의 현재의 버전을 수신할 수 있다. 요청하는 클라이언트는 공유된 객체의 현재의 버전이 저장소에 수신되었다는 것을 알게 된다. 블록(840)으로 진행하면, 공유된 객체의 현재의 버전은 요청하는 클라이언트와 동기화된다. 그러면 프로세스는 종료 블록에서 종료된다.

[0089] 도 9는 비동기적 통신 모드에서부터 동기적 통신 모드로 자연스럽게 전이하기위한 프로세스를 예시하는 동작가능한 흐름도를 예시한다. 프로세스는 시작 블록에서 시작되며, 시작 블록에서 공유된 객체에 액세스하도록 인가된 사용자들을 식별하는 피어 그룹을 설정한다.

[0090] 블록(900)으로 이동하여, 클라이언트는 서버상에서 공유된 객체를 액세스한다. 클라이언트는 공유된 객체를 또한 액세스하고 있는 다른 클라이언트들(즉, 피어 그룹)에 자동으로 연결된다. 공유된 객체는 매니페스트 파일에 연관된다. 공유된 객체는 해당하는 매니페스트 파일이 시스템 내에 저장되어 있는 위치를 식별케하는 고유한 위치 식별자를 포함한다.

[0091] 블록(910)으로 진행하여, 매니페스트 파일을 고유한 위치 식별자에 의해 식별된 위치로부터 가져온다. 매니페스트 파일은 공유된 객체의 다른 버전들 및 인스턴스들이 시스템 내에 저장되어 있는 위치들을 식별한다. 매니페스트 파일은 공유된 객체의 버전이 저장되어 있는 피어 그룹의 경우 피어 그룹 식별자를 포함한다.

[0092] 블록(920)으로 진행하여, 피어 그룹의 임의 다른 클라이언트가 매니페스트 파일에 의해 식별된 공유된 객체의 버전 또는 인스턴스를 액세스할 경우 피어-투-피어 네트워크가 구축된다. 그러므로, 클라이언트는 서버로부터 연결이 끊어져도 피어-투-피어 네트워크 상에서 공유된 파일을 계속해서 액세스할 수 있다. 그러면 프로세싱은 종료 블록에서 종료된다.

[0093] 도 10은 동기적 통신 모드에서 비동기적 통신 모드로 자연스럽게 이동하기위한 프로세스를 예시하는 동작가능한 흐름도를 예시한다. 프로세스는 시작 블록에서 시작되며, 시작 블록에서 피어-투-피어 네트워크가 공유된 파일에 액세스하도록 인가된 적어도 2명의 사용자 간에 구축된다.

[0094] 블록(1000)으로 이동하면, 클라이언트는 피어-투-피어 네트워크상에서 공유된 객체를 액세스한다. 공유된 객체는 매니페스트 파일에 연관된다. 공유된 객체는 해당하는 매니페스트 파일이 시스템 내에 저장되어 있는 위치를 식별케하는 고유한 위치 식별자를 포함한다.

[0095] 블록(1010)으로 이동하여, 공유된 객체에 연관된 매니페스트 파일을 고유한 위치 식별자에 의해 식별된 위치로부터 가져온다. 매니페스트 파일은 공유된 객체의 다른 버전들 및 인스턴스들이 시스템 내에 저장되어 있는 위치들을 식별한다. 블록(1020)으로 진행하여, 클라이언트는 서버에 연결된다. 클라이언트는 어떤 다른 클라이언트들이 또한 서버에 연결되어 있는지를 판정한다. 블록(1030)으로 이동하여, 클라이언트는 피어-투-피어 네트워크로부터 공유된 파일을 액세스하도록 인가된 다른 클라이언트들을 식별한다. 블록(1040)으로 진행하여, 피어-투-피어 네트워크가 이용불가능할 경우 그 클라이언트는 인가된 클라이언트에 연결된다. 그러면 프로세스는 종료 블록에서 종료된다.

[0096] 상기 명세서, 예시 및 데이터는 본 발명의 구성에 대한 사용법 및 제조에 관한 완벽한 설명을 제공한다. 본 발명의 많은 실시예들이 본 발명의 정신 및 범위 내에서 행해질 수 있기 때문에, 본 발명은 이하에 첨부된 청구항에 존재한다.

발명의 효과

[0097] 동기적 통신 모드와 비동기적 통신 모드 사이를 전이하기위한 방법 및 시스템을 제공하는 효과가 있다.

도면의 간단한 설명

[0001] 도 1은 본 발명의 예시적인 실시예에 따라서 사용될 수 있는 컴퓨팅 장치를 예시.

[0002] 도 2는 본 발명에 따르면, 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 시스템의 블록도.

[0003] 도 3은 본 발명에 따르면, 공유된 객체의 다른 부분들을 나타내는 링크된 노드들의 계층구조의 그래프를 예시.

[0004] 도 4는 본 발명에 따르면, 공유된 객체의 마스터(master) 페이지 및 연관된 모순(conflict) 페이지를 예시.

[0005] 도 5는 본 발명에 따르면, 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 시스템의 블록도.

[0006] 도 6은 본 발명에 따르면, 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 프로세스를 예시하는 동작가능한 흐름도.

[0007] 도 7은 본 발명에 따르면, 공유된 객체에 있는 복수의 사용자로부터의 모순되는 개정들을 조정 및 병합하는 프로세스를 예시하는 동작가능한 흐름도

[0008] 도 8은 본 발명에 따르면, 공유된 객체로의 복수의 사용자 개정들을 동기화시키는 프로세스를 예시하는 동작가능한 흐름도.

[0009] 도 9는 본 발명에 따르면, 비동기적 통신 모드에서 동기적 통신 모드로 자연스럽게 전이하기위한 프로세스를 예시하는 동작가능한 흐름도.

[0010] 도 10은 본 발명에 따르면, 동기적 통신 모드에서 비동기적 통신 모드로 자연스럽게 전이하기위한 프로세스를 예시하는 동작가능한 흐름도.

[0011] <도면의 주요 부분에 대한 부호의 설명>

[0012] 240: 교환 서버

[0013] 250: 웹 서버

[0014] 260: 피어-투-피어 네트워크

[0015] 270: 이메일 첨부

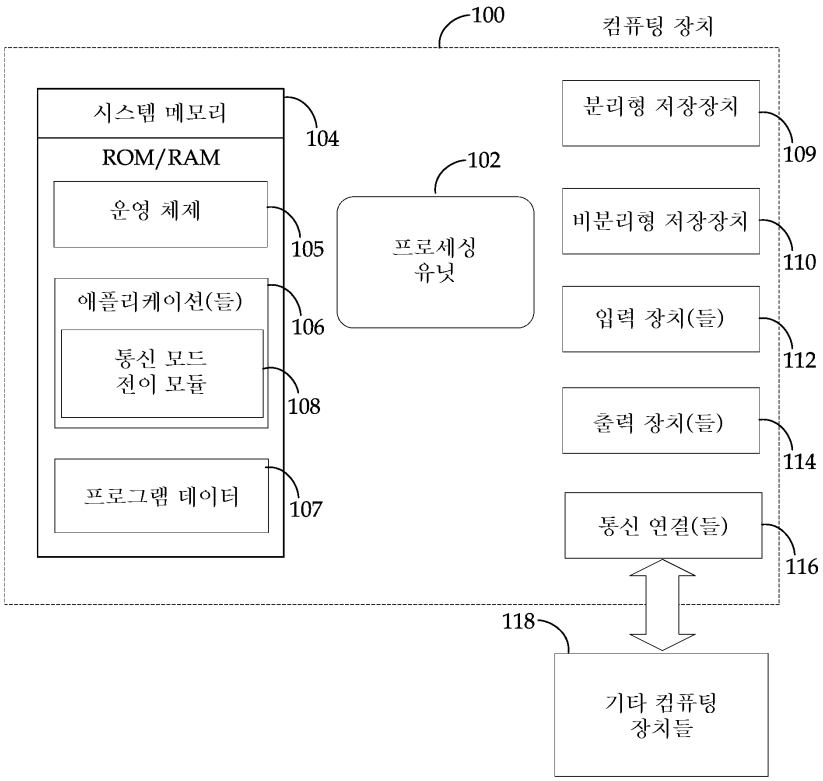
[0016] 242: 개정 파일

[0017] 252, 264: 공유된 객체

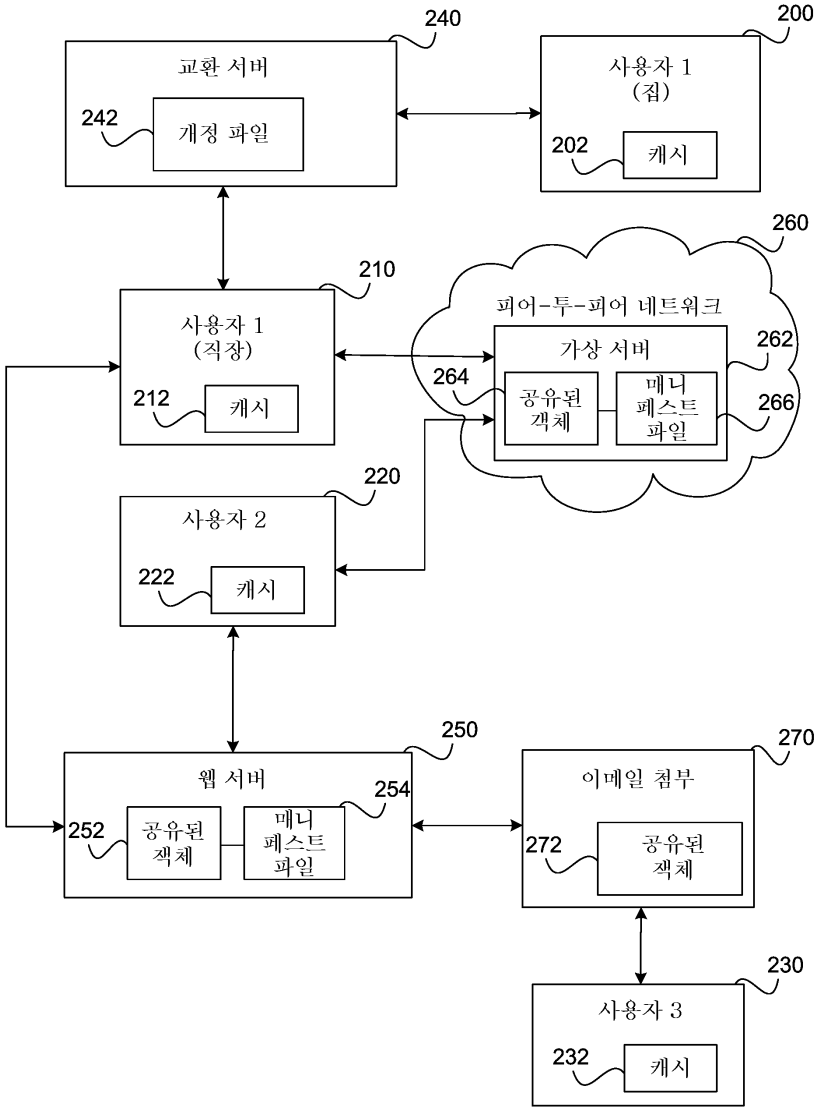
[0018] 254, 266: 매니페스트 파일

도면

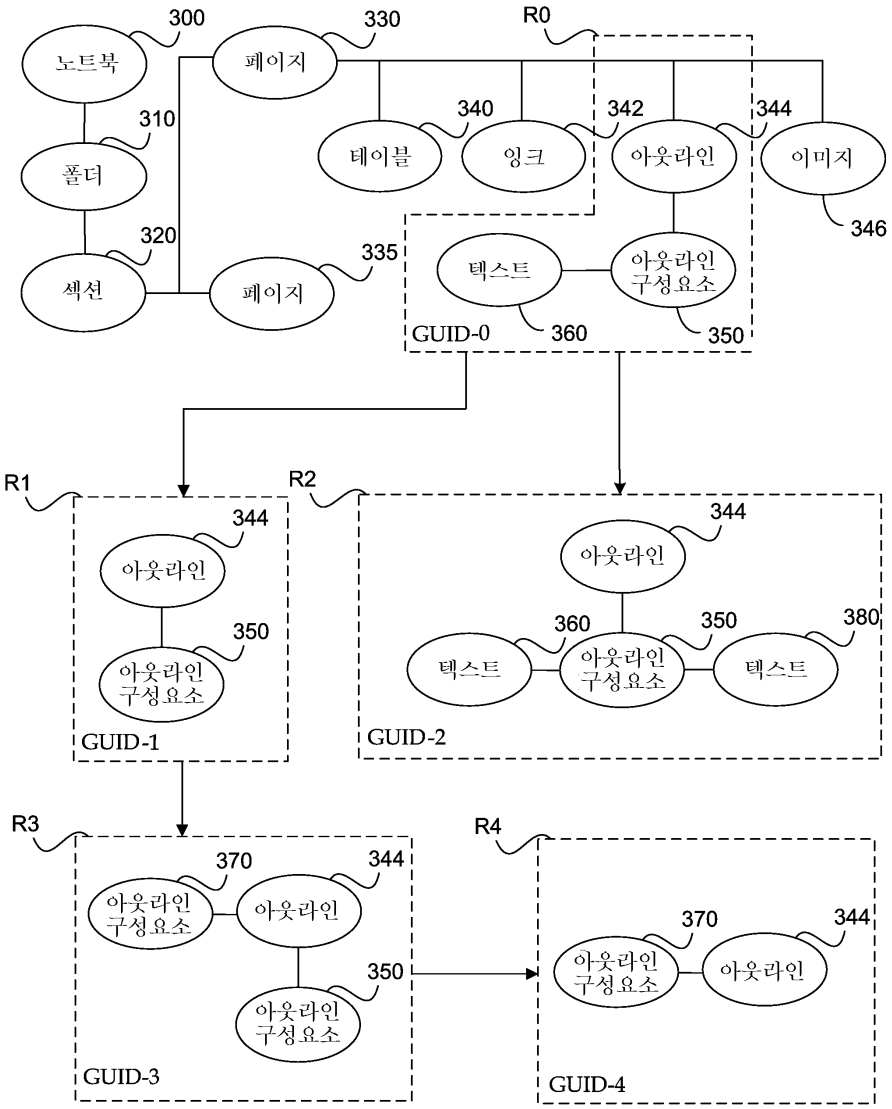
도면1


100
컴퓨팅 장치
시스템 메모리
104
ROM/RAM
운영 체제
105
애플리케이션(들)
106
통신 모드
진이 모듈
108
프로그램 데이터
107
102
프로세싱
유닛
분리형 저장장치
109
비분리형 저장장치
110
입력 장치(들)
112
출력 장치(들)
114
통신 연결(들)
116
118
기타 컴퓨팅
장치들

도면2


240
교환 서버
242
개정 파일
200
사용자 1
(집)
202
캐시
260
피어-투-피어 네트워크
210
사용자 1
(직장)
212
캐시
가상 서버
262
264
공유된
객체
매니
페스트
파일
266
220
사용자 2
222
캐시
250
웹 서버
254
252
공유된
객체
매니
페스트
파일
270
이메일 첨부
272
공유된
객체
230
사용자 3
232
캐시

도면3


노트북
300
페이지
330
R0
폴더
310
테이블
340
잉크
342
아웃라인
344
이미지
346
섹션
320
페이지
335
텍스트
아웃라인
구성요소
GUID-0
360
350
R1
아웃라인
344
아웃라인
구성요소
350
GUID-1
R2
아웃라인
344
텍스트
360
아웃라인
구성요소
350
텍스트
380
GUID-2
R3
아웃라인
구성요소
370
아웃라인
344
아웃라인
구성요소
350
GUID-3
R4
아웃라인
구성요소
370
아웃라인
344
GUID-4

도면4

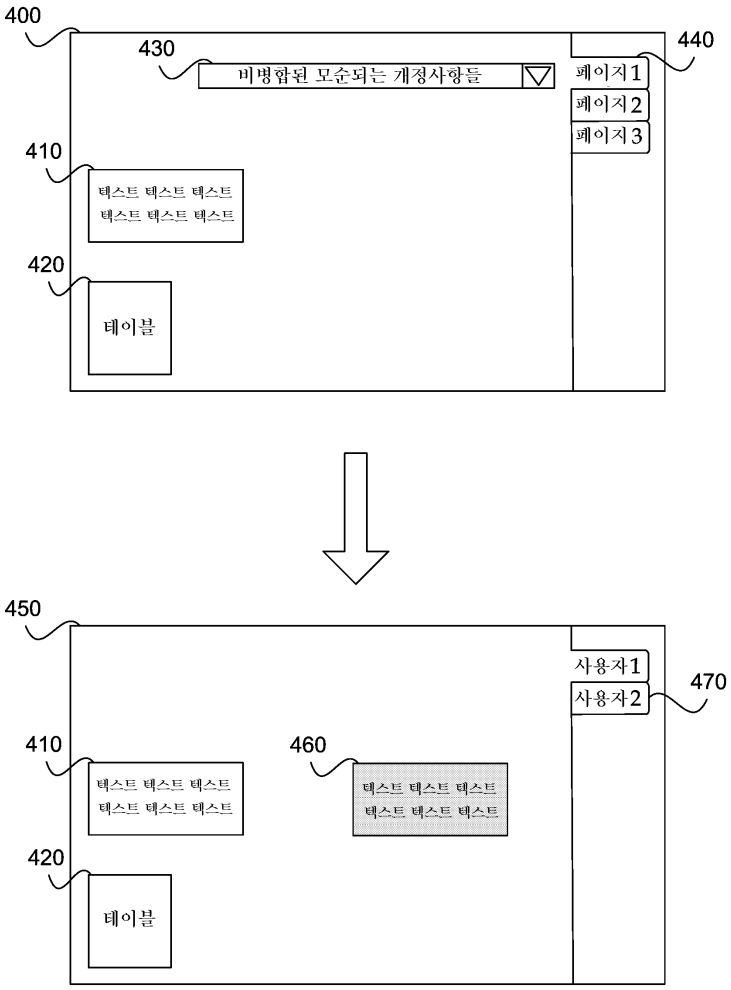
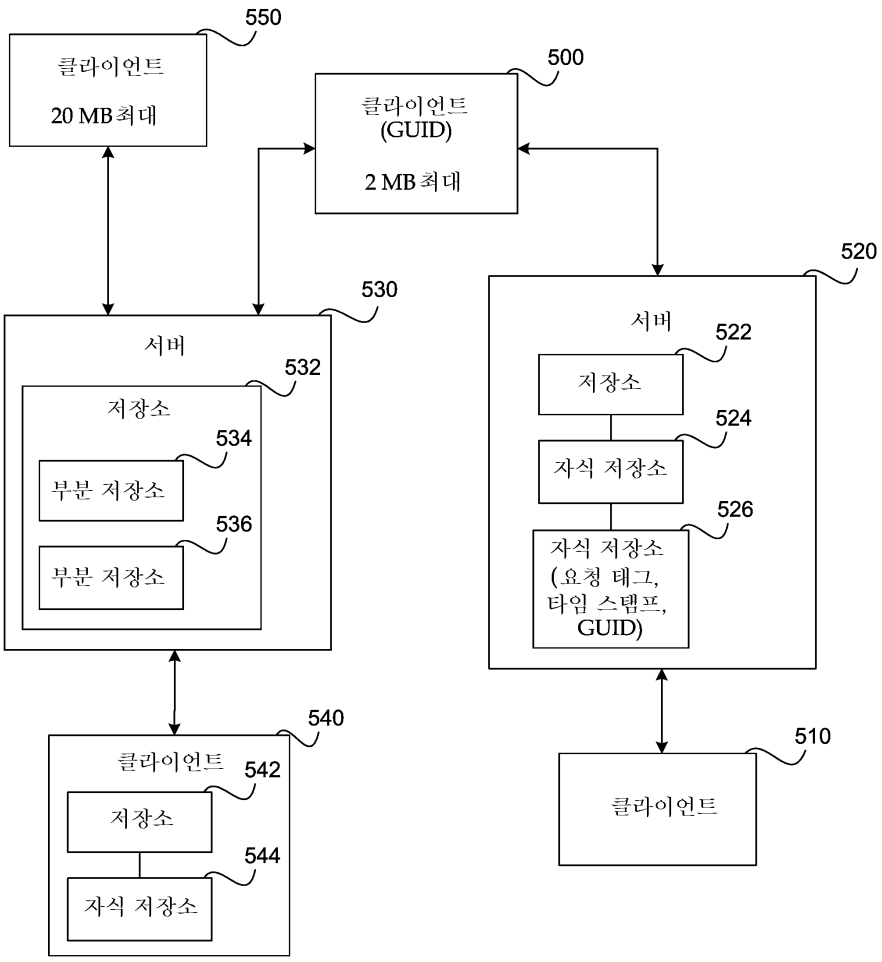
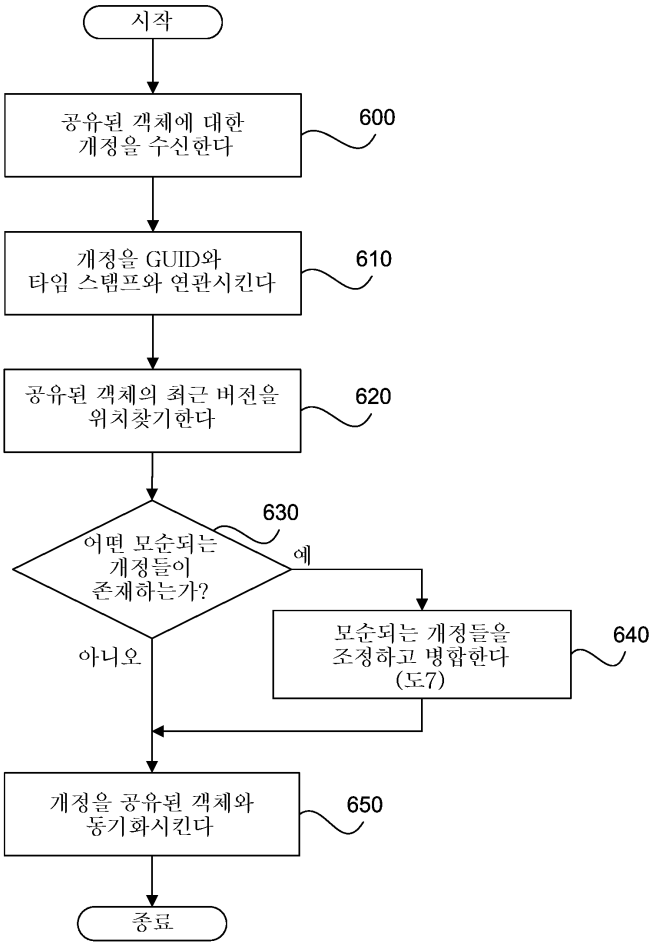

400
430
비병합된 모순되는 개정사항들
440
페이지1
페이지2
페이지3
410
텍스트 텍스트 텍스트
텍스트 텍스트 텍스트
420
테이블
450
사용자1
사용자2
470
410
텍스트 텍스트 텍스트
텍스트 텍스트 텍스트
460
텍스트 텍스트 텍스트
텍스트 텍스트 텍스트
420
테이블

Fig. 4

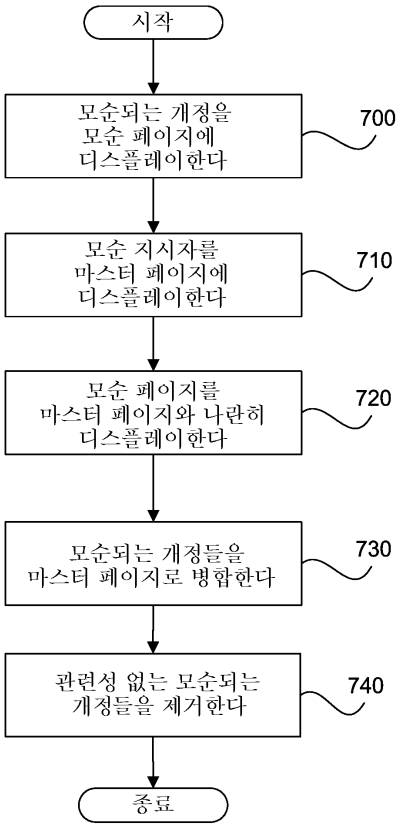
도면5


550
클라이언트
20 MB 최대
500
클라이언트
(GUID)
2 MB 최대
520
530
서버
532
저장소
534
부분 저장소
536
부분 저장소
서버
522
저장소
524
자식 저장소
526
자식 저장소
(요청 태그,
타임 스탬프,
GUID)
540
클라이언트
542
저장소
544
자식 저장소
510
클라이언트

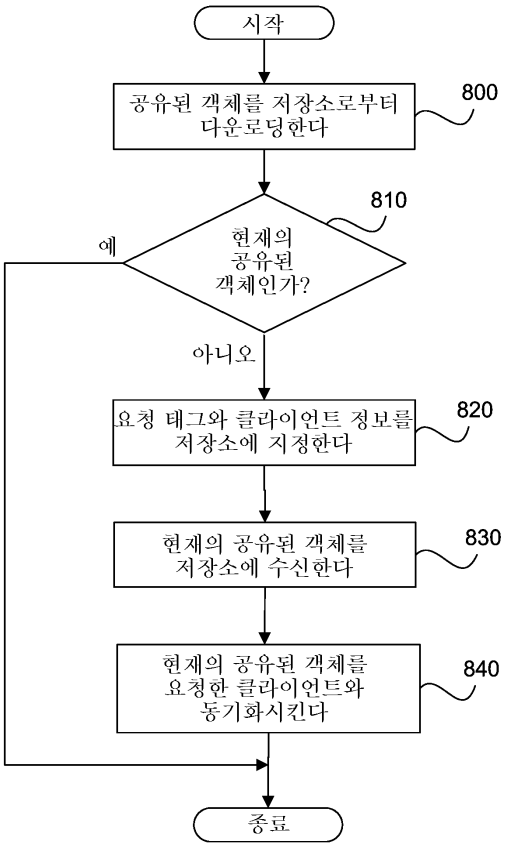
도면6


시작
공유된 객체에 대한 개정을 수신한다
600
개정을 GUID와 타임 스탬프와 연관시킨다
610
공유된 객체의 최근 버전을 위치찾기한다
620
어떤 모순되는 개정들이 존재하는가?
630
예
아니오
모순되는 개정들을 조정하고 병합한다 (도7)
640
개정을 공유된 객체와 동기화시킨다
650
종료

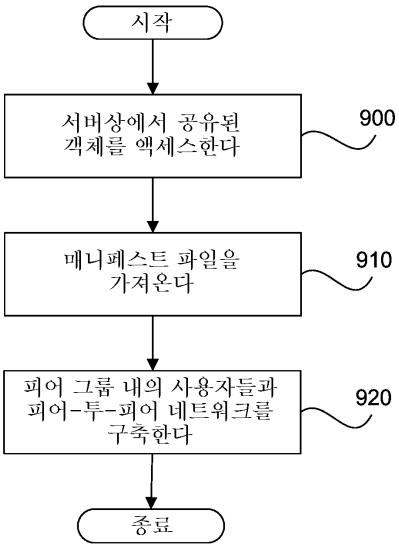
도면7


시작
모순되는 개정을 모순 페이지에 디스플레이한다
700
모순 지시자를 마스터 페이지에 디스플레이한다
710
모순 페이지를 마스터 페이지와 나란히 디스플레이한다
720
모순되는 개정들을 마스터 페이지로 병합한다
730
관련성 없는 모순되는 개정들을 제거한다
740
종료

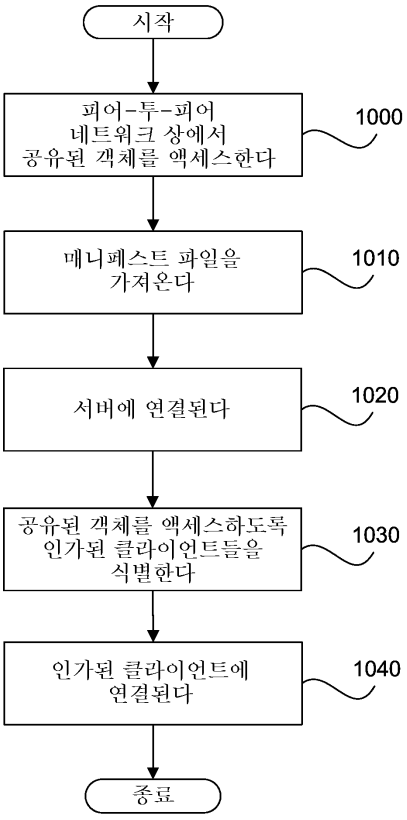
도면8


시작
공유된 객체를 저장소로부터 다운로딩한다
800
현재의 공유된 객체인가?
810
예
아니요
요청 태그와 클라이언트 정보를 저장소에 지정한다
820
현재의 공유된 객체를 저장소에 수신한다
830
현재의 공유된 객체를 요청한 클라이언트와 동기화시킨다
840
종료

도면9


시작
서버상에서 공유된 객체를 액세스한다
900
매니페스트 파일을 가져온다
910
피어 그룹 내의 사용자들과 피어-투-피어 네트워크를 구축한다
920
종료

도면10


시작
피어-투-피어 네트워크 상에서 공유된 객체를 액세스한다
1000
메니페스트 파일을 가져온다
1010
서버에 연결된다
1020
공유된 객체를 액세스하도록 인가된 클라이언트들을 식별한다
1030
인가된 클라이언트에 연결된다
1040
종료