



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0016347
(43) 공개일자 2017년02월13일

- (51) 국제특허분류(Int. Cl.)
G06F 9/44 (2006.01)
- (52) CPC특허분류
G06F 8/68 (2013.01)
- (21) 출원번호 10-2016-7033796
- (22) 출원일자(국제) 2015년06월02일
심사청구일자 없음
- (85) 번역문제출일자 2016년12월01일
- (86) 국제출원번호 PCT/US2015/033634
- (87) 국제공개번호 WO 2015/187598
국제공개일자 2015년12월10일
- (30) 우선권주장
14/295,478 2014년06월04일 미국(US)

- (71) 출원인
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
- (72) 발명자
아가왈 아비섹
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠
(8/1172) 마이크로소프트 테크놀로지 라이선싱,
엘엘씨
크루거 앤터니 디
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠
(8/1172) 마이크로소프트 테크놀로지 라이선싱,
엘엘씨
(뒷면에 계속)
- (74) 대리인
제일특허법인

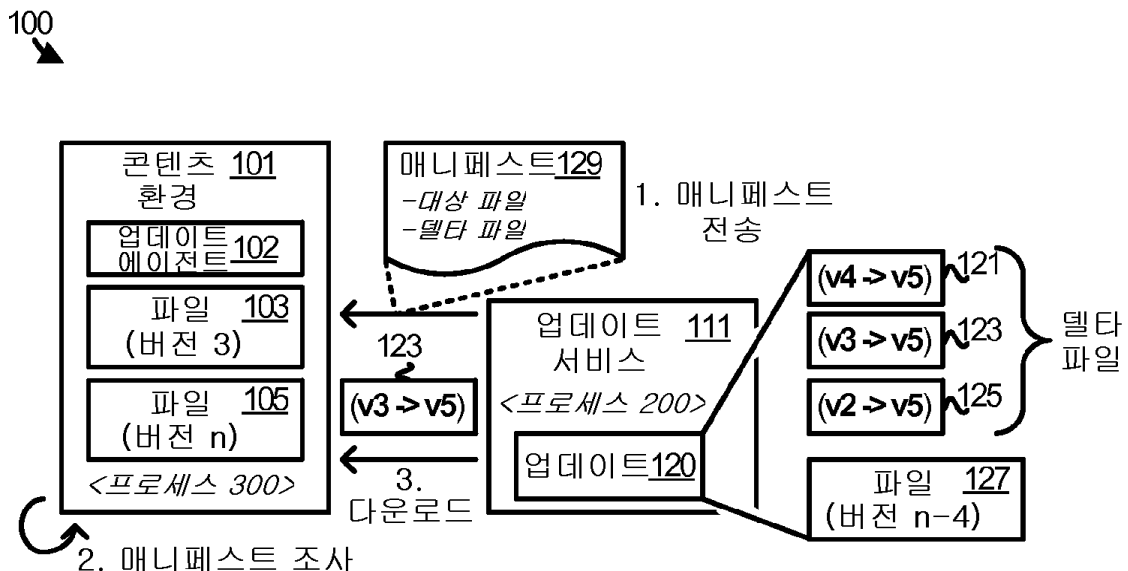
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 디지털 콘텐츠를 위한 업데이트 개선 기법

(57) 요약

본 명세서에 디지털 콘텐츠의 업데이트를 개선하기 위한 시스템, 방법 및 소프트웨어가 개시된다. 적어도 하나의 구현예에서, 업데이트 에이전트는 파일 세트로부터, 적어도 파일의 현재 버전에서 파일의 새 버전으로 업데이트 되도록 스케줄링된 파일을, 파일 세트의 업데이트의 일부로서 식별한다. 상기 업데이트는 파일을 파일의 이전 버전들에서 새 버전으로 업데이트하기 위한 델타 파일 세트 및 파일을 새 버전으로 업데이트하기 위한 완전한 파일을 포함할 수 있다.

대표도 - 도1



(72) 발명자

뉘옌 후이 큐

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨

카이 피터

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨

크리들 제퍼슨 비

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨

명세서

청구범위

청구항 1

디지털 콘텐츠를 업데이트하기 위한 프로그램 명령어가 저장된 하나 이상의 컴퓨터 판독형 저장 매체로서,
상기 프로그램 명령어는 처리 시스템에 의해 실행될 때 상기 처리 시스템으로 하여금 적어도:

파일 세트로부터, 적어도 파일의 현재 버전에서 상기 파일의 새 버전으로 업데이트되도록 스케줄링된 상기 파일을, 상기 파일의 이전 버전들에서 상기 새 버전으로 상기 파일을 업데이트하기 위한 델타 파일(delta file) 세트 및 상기 파일을 상기 새 버전으로 업데이트하기 위한 완전한 파일(complete file)을 포함하는 상기 파일 세트의 업데이트의 일부로서, 식별하게 하고,

상기 델타 파일 세트 중 어느 델타 파일이든 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는 데 사용될 수 있는지 여부를 판정하게 하며,

상기 델타 파일 세트가 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는 데 이용할 델타 파일을 적어도 포함하는 경우, 상기 델타 파일을 다운로드하게 하고,

상기 델타 파일 세트 중 어느 델타 파일도 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는 데 사용될 수 없는 경우, 상기 파일을 상기 새 버전으로 업데이트하는 데 사용되는 완전한 파일을 다운로드하게 하는,

컴퓨터 판독형 저장 매체.

청구항 2

제1항에 있어서,

적어도 상기 파일을 식별하기 위해, 상기 프로그램 명령어는 상기 처리 시스템으로 하여금, 상기 파일 세트 중 어느 파일 서브세트가 업데이트의 대상이 되는지를 식별하며 업데이트 서비스에 의해 전달된 업데이트 매니페스트(manifest)를 조사하게 하는

컴퓨터 판독형 저장 매체.

청구항 3

제2항에 있어서,

상기 업데이트 매니페스트는 상기 업데이트에 포함되는 델타 파일 세트들을 더 식별하고, 상기 델타 파일 세트들 각각은 상기 업데이트의 대상인 파일 서브세트의 서로 다른 파일에 대응하며, 상기 델타 파일 세트들 중 하나의 세트는 상기 파일을 상기 이전 버전들에서 상기 새 버전으로 업데이트하기 위한 델타 파일 세트를 포함하는,

컴퓨터 판독형 저장 매체.

청구항 4

제3항에 있어서,

상기 매니페스트는 상기 델타 파일 세트의 각각의 델타 파일이 상기 파일을 이전 버전들 중 어느 이전 버전으로부터 상기 새 버전으로 업데이트하는 데 사용될 수 있는지를 더 식별하는,

컴퓨터 판독형 저장 매체.

청구항 5

제4항에 있어서,

상기 프로그램 명령어는 상기 처리 시스템으로 하여금, 상기 업데이트 서비스에 상기 매니페스트에 대한 요청을 개시하게 하는,

컴퓨터 판독형 저장 매체.

청구항 6

제1항에 있어서,

상기 파일 세트의 업데이트는 제1 애플리케이션 및 제2 애플리케이션을 포함하는 애플리케이션 군의 업데이트를 포함하고, 상기 파일 세트는 상기 제1 애플리케이션과 연관된 제1 애플리케이션 파일 및 상기 제2 애플리케이션과 연관된 제2 애플리케이션 파일을 포함하는,

컴퓨터 판독형 저장 매체.

청구항 7

제1항에 있어서,

상기 파일 세트는 애플리케이션 파일, 미디어 파일, 및 운영 체제 파일 중 적어도 하나를 포함하는

컴퓨터 판독형 저장 매체.

청구항 8

디지털 콘텐츠를 업데이트하기 위한 방법으로,

파일 세트로부터, 적어도 파일의 현재 버전에서 상기 파일의 새 버전으로 업데이트되도록 스케줄링된 상기 파일을, 상기 파일의 이전 버전들에서 상기 새 버전으로 상기 파일을 업데이트하기 위한 델타 파일(delta file) 세트 및 상기 파일을 상기 새 버전으로 업데이트하기 위한 완전한 파일(complete file)을 포함하는 상기 파일 세트의 업데이트의 일부로서, 식별하는 단계,

상기 델타 파일 세트 중 어느 델타 파일이든 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는데 사용될 수 있는지 여부를 판정하는 단계,

상기 델타 파일 세트가 상기 파일을 현재 버전에서 상기 새 버전으로 업데이트하는데 이용할 델타 파일을 적어도 포함하는 경우, 상기 델타 파일을 이용해 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는 단계, 및

상기 델타 파일 세트 중 어느 델타 파일도 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는데 사용될 수 없는 경우, 상기 파일에 대응하는 상기 완전한 파일을 이용해 상기 파일을 상기 현재 버전에서 상기 새 버전으로 업데이트하는 단계를 포함하는,

디지털 콘텐츠를 업데이트하기 위한 방법.

청구항 9

제8항에 있어서,

상기 파일을 식별하는 단계는 상기 파일 세트 중 어느 파일 서브세트가 상기 업데이트의 대상이 되는지를 식별하며 업데이트 서비스에 의해 전달된 업데이트 매니페스트(manifest)를 조사하는 단계를 포함하는

디지털 콘텐츠를 업데이트하기 위한 방법.

청구항 10

제9항에 있어서,

상기 업데이트 매니페스트는 상기 업데이트에 포함되는 델타 파일 세트들을 더 식별하고, 상기 델타 파일 세트들 각각은 상기 업데이트의 대상인 파일 서브세트의 서로 다른 파일에 대응하며, 상기 델타 파일 세트들 중 하나의 세트가 상기 파일을 상기 이전 버전들에서 상기 새 버전으로 업데이트하기 위한 델타 파일 세트를 포함하는,

디지털 콘텐츠를 업데이트하기 위한 방법.

청구항 11

제10항에 있어서,

상기 매니페스트는 상기 델타 파일 세트의 각각의 델타 파일이 상기 파일을 이전 버전들 중 어느 이전 버전으로부터 상기 새 버전으로 업데이트하는 데 사용될 수 있는지를 더 식별하는,

디지털 콘텐츠를 업데이트하기 위한 방법.

청구항 12

제8항에 있어서,

상기 파일 세트의 업데이트는 제1 애플리케이션 및 제2 애플리케이션을 포함하는 애플리케이션 군의 업데이트를 포함하고, 상기 파일 세트는 상기 제1 애플리케이션과 연관된 제1 애플리케이션 파일 및 상기 제2 애플리케이션과 연관된 제2 애플리케이션 파일을 포함하는,

디지털 콘텐츠를 업데이트하기 위한 방법.

청구항 13

제8항에 있어서,

상기 파일 세트는 애플리케이션 파일, 미디어 파일, 및 운영 체제 파일 중 적어도 하나를 포함하는,

디지털 콘텐츠를 업데이트하기 위한 방법.

청구항 14

업데이트 서비스로부터 원격인 애플리케이션 환경에 로컬로 설치된 소프트웨어 패키지를 업데이트하기 위한 방법으로서,

상기 업데이트 서비스에서, 각각의 소프트웨어 패키지의 업데이트 상태를 추적하는 단계 - 상기 각각의 소프트웨어 패키지는 파일 세트를 포함함 - 와,

상기 업데이트 서비스에서, 상기 각각의 소프트웨어 패키지에 대한 상기 업데이트 상태에 적어도 부분적으로 기초로 하여, 적어도 상기 파일 세트 중 어느 파일 서브세트가 업데이트의 대상인지를 식별하며, 상기 파일 서브세트의 각각의 파일을 이전 버전들에서 새 버전으로 업데이트하기 위한 델타 파일 세트를 식별하고, 상기 파일 서브세트의 각각의 파일을 상기 새 버전으로 업데이트하기 위한 완전한 파일을 식별하는 상기 각각의 소프트웨어 패키지에 대한 매니페스트를 생성하는 단계와,

상기 업데이트 서비스에서, 각각의 소프트웨어 패키지에 대한 상기 매니페스트를 상기 애플리케이션 환경에서 실행 중인 대응하는 업데이트 에이전트로 통신하는 단계를 포함하는

소프트웨어 패키지를 업데이트하기 위한 방법.

청구항 15

제14항에 있어서,

각각의 소프트웨어 패키지는 제1 애플리케이션 및 제2 애플리케이션을 포함하는 애플리케이션 군을 포함하고, 상기 파일 세트는 상기 제1 애플리케이션과 연관된 제1 애플리케이션 파일 및 상기 제2 애플리케이션과 연관된 제2 애플리케이션 파일을 포함하고, 상기 파일 세트는 애플리케이션 파일, 미디어 파일, 및 운영 체제 파일 중 적어도 하나를 포함하는

소프트웨어 패키지를 업데이트하기 위한 방법.

발명의 설명

기술 분야

[0001] 본 발명의 측면들은 컴퓨팅 하드웨어 및 소프트웨어 기술과 관련되며, 더 구체적으로 디지털 콘텐츠를 업데이트 하는 것과 관련된다.

배경 기술

[0002] 모든 유형의 디지털 콘텐츠의 업데이트가 높은 빈도로 클라우드를 통해 전달된다. 일부 소프트웨어 패키지가 1년에 한 번 또는 더 적은 빈도로 업데이트되는 것에 비해, 이제는 소프트웨어 개발자가 매월 또는 이보다 더 자주 업데이트를 제공하는 것이 흔한 일이다. 그 밖의 다른 유형의 디지털 콘텐츠, 가령, 지도 데이터베이스 및 미디어 라이브러리가 또한 높은 빈도로 업데이트된다.

[0003] 과거에는, 애플리케이션의 업데이트가 새로운 설치 패키지를 다운로드하고 설치함으로써 이뤄졌을 수 있다. 이 기법의 단점은 설치 패키지 내 새 파일이 파일의 구 버전을 대체할 때 애플리케이션 프로그램의 사용자 커스텀이 소실될 수 있다는 것이다. 또 다른 단점은 일부 설치 패키지의 큰 크기이며, 이는 시간과 대역폭의 상당한 소비를 초래할 수 있다.

[0004] 많은 경우에, 애플리케이션의 한 버전과 애플리케이션의 또 다른 버전 간 파일 또는 바이너리의 차이가 그리 크지 않을 것이며, 이는 전체-설치 프로세스를 덜 효율적이게 할 수 있다. 한 버전에서 다음 버전으로 변하는 바이너리 내에서도, 변경은 사소할 수 있다. 이들 및 그 밖의 다른 단점이 패칭(patching)의 개발에 의해 완화되었다. 패칭을 이용해, 애플리케이션 패키지 내 제한된 파일 세트가 업데이트되도록 타깅됨으로써, 다운로드될 필요가 있는 파일 수가 감소할 수 있다.

[0005] 차등 압축 기법, 예컨대, 바이너리 델타 압축(binary delta compression)이 활용되어, 패치 파일을 포함하든 완전한 파일을 포함하든 무관하게, 업데이트 패키지의 전체 크기가 축소될 수 있다. 바이너리 델타 압축에 의해, 업데이트 프로세스 동안 구 파일과 신 파일 간 차이만 다운로드될 수 있다. 따라서 업데이트 패키지는, 파일의 완전한 버전을 다운로드할 필요 없이, 파일의 구 버전을 상기 파일의 신 버전에 대해 최신이 되도록 하는 델타 파일을 포함할 수 있다.

[0006] 패칭 및 차등 압축 기법이 특정 업데이트의 성능을 개선할 수 있지만, 개발자는 디지털 콘텐츠 패키지에 대해 업데이트가 신속하게 생성되는 것으로 인한 새로운 상황과 직면한다. 업데이트가 더 자주 생성되기 때문에, 가능한 업데이트 상태의 범위가 확장되었다. 현재 업데이트 패러다임을 계속하는 것은 모든 가능한 업데이트 상태를 지원하기 위해 더 많은 델타 파일이 유지관리될 필요가 있을 것을 의미한다.

발명의 내용

과제의 해결 수단

[0007] 개요

[0008] 디지털 콘텐츠를 업데이트하는 프로세스를 개선하는 시스템, 방법, 및 소프트웨어가 제공된다. 적어도 하나의 구현예에서, 업데이트 에이전트는 파일 세트로부터, 적어도 파일의 현재 버전에서 파일의 새 버전으로 업데이트 되도록 스케줄링된 파일을, 파일 세트의 업데이트의 일부로서 식별한다. 업데이트는 파일을 파일의 이전 버전들에서 새 버전으로 업데이트하기 위한 델타 파일 세트와 상기 파일을 새 버전으로 업데이트하기 위한 완전한 파일을 포함할 수 있다.

[0009] 업데이트 에이전트는 델타 파일 세트 중 어느 델타 파일이라도 파일을 현재 버전에서 새 버전으로 업데이트하는 데 사용될 수 있는지 여부를 결정한다. 델타 파일 세트가 적어도 파일을 현재 버전에서 새 버전으로 업데이트하는 데 사용될 델타 파일을 포함하는 경우, 상기 델타 파일이 다운로드되고 파일을 업데이트하는 데 사용된다. 델타 파일 세트 중 어느 것도 파일을 현재 버전에서 새 버전으로 업데이트하는 데 사용될 수 없는 경우, 파일을 새 버전으로 업데이트하기 위한 완전한 파일이 다운로드된다.

[0010] 이 개요는 이하의 발명의 설명에서 더 기재된 개념들의 모음을 더 단순화된 형태로 소개하기 위해 제공된다. 이 개요가 청구되는 대상의 핵심 특징 또는 필수 특징을 식별하도록 의도된 것이 아니며 본 발명의 범위를 한정하는 데 사용되도록 의도된 것도 아님이 이해될 수 있다.

도면의 간단한 설명

[0011] 본 발명의 많은 측면들이 다음의 도면을 참조하여 더 잘 이해될 수 있다. 몇 가지 구현예가 이들 도면과 관련되

어 기재되지만, 본 발명은 본 명세서에 개시된 구현예에 한정되지 않는다. 이와 달리, 의도는 모든 대안, 수정 및 균등예를 포함하는 것이다.

도 1은 하나의 구현예에서의 업데이트 아키텍처를 도시한다.

도 2는 하나의 구현예에서의 업데이트 프로세스를 도시한다.

도 3은 하나의 구현예에서의 업데이트 프로세스를 도시한다.

도 4는 하나의 구현예에서의 업데이트 아키텍처를 도시한다.

도 5는 하나의 구현예에서의 동작 시퀀스를 도시한다.

도 6은 하나의 구현예에서의 업데이트 아키텍처를 도시한다.

도 7은 하나의 구현예에서의 업데이트 프로세스를 도시한다.

도 8은 하나의 구현예에서의 업데이트 프로세스의 결과를 도시한다.

도 9는 도 1-9 및 이하의 발명의 설명을 참조하여 본 명세서에 개시된 애플리케이션, 서비스, 프로세스, 아키텍처 및 동작 시나리오 중 임의의 것을 구현하기에 적합한 컴퓨팅 시스템을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 명세서에 개시된 구현예가 디지털 콘텐츠에 대한 개선된 업데이트를 가능하게 함으로써, 제한된 델타 파일 세트가 생성되고 이전 버전에서 더 최신 버전까지 파일을 업데이트하도록 유지관리될 수 있다. 그 후 업데이트 에이전트가 연관된 파일의 업데이트 상태에 따라 델타 파일을 선택적으로 다운로드할 수 있다. 적절한 델타 파일이 이용 가능하지 않을 때, 업데이트 에이전트는 새 파일의 완전한 버전을 다운로드할 수 있다.

[0013] 이러한 구현예는 디지털 콘텐츠의 업데이트가 더 자주 생성되고 이로 인해 가능한 업데이트 상태의 대응하는 범위가 증가되는 경우를 수용하는 데 도움이 된다. 과거에는, 업데이트가 연간 한 번 또는 18개월마다 한 번 생성될 때, 예를 들어, 로컬 설치에 존재할 수 있는 다양한 업데이트 상태가 크게 다양하지 않았다. 예를 들어, 3년 동안 소프트웨어 패키지의 기본 상태로부터 2번의 업데이트가 생성된 경우, 3년의 끝 부분에서 단 3개의 서로 다른 업데이트 상태만 있을 수 있다. 따라서 모든 업데이트 상태를 연결하기에 적합한 델타 파일을 생성하고 유지관리하는 것이 합리적인 노력일 것이다.

[0014] 업데이트 생성의 빈도의 증가와 함께, 동일한 3년 동안 훨씬 더 많은 수의 업데이트 상태가 있을 수 있다. 예를 들어, 소프트웨어 패키지가 매월 업데이트되는 경우, 3년 전체에 걸쳐 수십 개의 서로 다른 업데이트 상태가 존재할 수 있다. 이들의 다양한 업데이트 상태 모두에서 소프트웨어 패키지의 인스턴스를 소프트웨어 패키지의 최신 버전으로 연결하는 데 필요한 모든 델타 파일을 생성하고 유지관리하는 것은 고되고 비효율적인 작업이다.

[0015] 본 발명에 따르면, 지원되는 델타 파일의 수가 제한되어, 존재할 수 있는 다양한 업데이트 상태의 서브세트만 델타 파일을 이용해 디지털 콘텐츠의 신 버전으로 연결될 수 있다. 이 기법은 다른 경우라면 업데이트 파일을 생성하고 유지관리하는 데 관련됐을 수 있는 노력과 자원의 양을 감소시킨다.

[0016] 업데이트의 일부로서 이용 가능한 상기 델타 파일이 업데이트 매니페스트(update manifest)로 열거될 수 있으며, 이 업데이트 매니페스트는 업데이트 서비스에 의해 다양한 콘텐츠 환경에서 실행되는 업데이트 에이전트에게로 통신된다. 특정 업데이트 에이전트가 매니페스트를 조사하여 파일 세트 중 어느 파일(또는 파일들)이 파일의 이전 버전에서 새 버전으로의 업데이트의 타겟인지를 식별할 수 있다. 또한 매니페스트는 파일을 파일의 이전 버전에서 새 버전으로 업데이트하는 데 이용 가능한 델타 파일을 열거할 수 있다. 또한 새 버전으로 파일을 업데이트하기 위한 완전한 파일이 업데이트에 포함될 수도 있다.

[0017] 업데이트 에이전트는 매니페스트로부터 파일을 현재 버전에서 새 버전으로 업데이트하기 위해 적합한 업데이트에서 델타 파일이 이용 가능한지 여부를 결정할 수 있다. 적합한 델타 파일이 이용 가능한 경우, 업데이트 에이전트는 업데이트 서비스로부터 또는 선택사항으로서 그 밖의 다른 일부 소스로부터 델타 파일의 다운로드를 개시할 수 있다. 적합한 델타 파일이 이용 가능하지 않은 경우, 완전한 파일이 다운로드되고 사용되어 파일을 업데이트할 수 있다.

[0018] 업데이트의 타겟인 특정 소프트웨어 패키지 또는 그 밖의 다른 유형의 디지털 콘텐츠가 파일 세트를 포함할 수 있으며, 이들 파일 중 적어도 일부가 업데이트의 대상이 된다. 따라서 업데이트가 델타 파일 세트를 포함할 수

있다. 각각의 개별 델타 파일 세트는 업데이트 타깃 내 파일들의 개별 파일에 대응할 수 있다. 예를 들면, 하나의 소프트웨어 패키지가 복수의 파일을 포함할 수 있다. 따라서 소프트웨어 패키지로의 업데이트가 소프트웨어 파일들 중 하나의 소프트웨어 파일에 대응하는 하나의 델타 파일 세트와 소프트웨어 파일들 중 다른 하나의 소프트웨어 파일에 대응하는 또 다른 델타 파일 세트를 포함할 수 있다.

- [0019] 이들 시나리오에서, 업데이트 에이전트가 매니페스트를 조사하여 소프트웨어 패키지 또는 그 밖의 다른 디지털 콘텐츠 내 어느 파일 또는 파일 서브세트가 업데이트 대상이 되는지를 결정할 수 있다. 이로부터, 업데이트 에이전트는 파일 서브세트의 각각의 파일에 대해, 업데이트가 파일을 이전 버전에서 새 버전으로 업데이트하는 데 적합한 델타 파일을 포함하는지 여부를 결정할 수 있다. 일부 시나리오에서, 매니페스트는 업데이트 내 델타 파일에 대해, 파일의 이전 버전들 중 어느 버전에 각각의 델타 파일이 적용될 수 있는지를 기술할 수 있다.
- [0020] 업데이트 에이전트는 업데이트 서비스로의 요청을 개시하여 업데이트에 대한 매니페스트를 획득할 수 있다. 이러한 요청은 업데이트가 이용 가능하다는 업데이트 서비스에 의해 제공되는 통지에 응답하여 이뤄질 수 있다. 그러나 매니페스트를 전달하기 위한 대안적 기법이 사용될 수 있으며, 가령, 업데이트 서비스가 업데이트 통지와 함께 매니페스트를 통신하는 것이 있다.
- [0021] 디지털 콘텐츠의 예시로는 개별 소프트웨어 애플리케이션, 소프트웨어 패키지 또는 소프트웨어 제품군, 비디오 콘텐츠, 디지털 북 및 정기 간행물, 디지털 음악 및 이따금 업데이트될 수 있는 그 밖의 다른 임의의 유형의 디지털 콘텐츠이다. 업데이트될 수 있는 파일의 예시로는 데이터베이스 파일, 애플리케이션 파일, 운영 체제 파일, 및 미디어 파일(가령, 비디오, 오디오, 또는 디지털 인쇄된 미디어)이 있다.
- [0022] 일부 구현예에서, 업데이트 서비스가 파일 세트를 각각 포함하고 이들의 업데이트 상태를 기초로 소프트웨어 패키지에 대한 매니페스트를 생성하는 소프트웨어 패키지의 업데이트 상태를 추적할 수 있다. 이는, 예를 들어, 소프트웨어의 하나의 버전 또는 가능한 버전의 범위에 대응하는 소프트웨어 패키지의 설치된 인스턴스의 공유부분(share)을 추적하는 것을 포함할 수 있다. 설치된 인스턴스의 특정 버전이 갖는 공유부분이 얼마나 큰지에 따라, 패키지의 설치된 인스턴스를 소프트웨어 패키지의 상기 특정 버전에서 새 버전으로 업데이트하기에 적합한 델타 파일이 패키지의 각각의 파일에 대해 생성될 수 있다.
- [0023] 델타 파일(또는 파일 세트)을 구축할지 여부를 결정할 때 설치된 인스턴스 중 특정 버전이 차지하는 공유부분이 또한 고려될 수 있다. 특정 버전에 대한 공유부분이 시간의 흐름에 따라 감소할 때, 이는 어느 순간 특정 임계값 미만이 되어, 어떠한 델타 파일도 생성되지 않도록 할 것이다. 업데이트에 대해 생성된 매니페스트가 특정 업데이트 타깃을 위해 생성되는 앞서 언급된 델타 파일 및 특정 델타 파일에 의해 파일이 업데이트될 수 있는 특정 버전을 식별할 수 있다.
- [0024] 도 1을 다시 참조하면, 업데이트 아키텍처(100)는 콘텐츠 환경(101) 및 업데이트 서비스(111)를 포함한다. 콘텐츠 환경(101)은 업데이트 에이전트(102), 파일(103), 및 파일(105)을 포함한다. 파일(103 및 105)은 이따금 업데이트 서비스(111)에 의해 제공되는 업데이트의 대상이 될 수 있다. 업데이트(120)는 이러한 하나의 업데이트를 나타내고 델타 파일(121), 델타 파일(123), 델타 파일(125) 및 파일(127)을 나타낸다.
- [0025] 동작 중에, 업데이트(120)가 형성되고 업데이트(120)의 내용을 기술하는 대응하는 매니페스트가 생성된다. 매니페스트(129)는 업데이트(120)와 관련하여 생성될 수 있는 매니페스트를 나타낸다. 매니페스트(129)는 콘텐츠 환경에서 어느 파일이 업데이트되도록 스케줄링될 수 있는지에 대한 정보를 포함한다. 또한 정보는 업데이트에서 이용 가능한 콘텐츠를 나타내며, 가령, 어느 델타 파일이 업데이트 내에 있는지 그리고 어느 완전한 파일이 업데이트에 있는지에 대한 정보이다.
- [0026] 매니페스트 내에 특정된 델타 파일 각각이 사용되어, 다양한 이전 버전 중 하나의 버전에서 최신 버전으로 타깃 파일을 업데이트할 수 있다. 매니페스트(129) 내 정보는, 델타 파일을 식별하는 것에 추가로, 어느 이전 버전으로부터 델타 파일이 사용되어 타깃 파일을 파일의 최신 버전으로 업데이트할 수 있는지를 식별할 수 있다. 이 동작 시나리오에서, 델타 파일(121)은 파일의 제4 버전과 제5 버전 간 차이를 나타내며, 델타 파일(123)은 파일의 제3 버전과 제5 버전 간 차이를 나타내고, 델타 파일(125)은 파일의 제2 버전과 제5 버전 간 차이를 나타낸다. 파일(127)은 파일의 완전한 제5 버전을 나타낸다.
- [0027] 업데이트 서비스(111)는 매니페스트(129)를 콘텐츠 환경(101)으로 그리고 아마도 업데이트(120)를 이용할 수 있는 그 밖의 다른 콘텐츠 환경(도시되지 않음)으로도 전달한다. 업데이트 에이전트(102)는 매니페스트(129)를 조사하여, 업데이트 서비스(111)로부터 다운로드될 델타 파일을 결정할 수 있다. 일부 시나리오에서, 어떠한 델타 파일도 다운로드되지 않는다. 오히려, 업데이트되는 완전한 버전의 파일이 다운로드될 수 있다.

- [0028] 적절한 델타 파일 또는 완전한 파일이 식별되면, 델타 파일 또는 완전한 파일이 업데이트 서비스(111)로부터 콘텐츠 환경(101)으로 다운로드된다. 그 후 따라서 타깃 파일이 업데이트될 수 있다. 이 시나리오에서, 델타 파일(123)이 식별되고 다운로드되어 파일(103)을 제3 버전에서 제5 버전으로 업데이트할 수 있다.
- [0029] 도 2는 도 1에 대해 언급된 동작 시나리오의 맥락에서 업데이트 서비스(111)에 의해 채용될 수 있는 업데이트 프로세스(200)를 도시한다. 도 2에 기재된 단계를 괄호로 참조하면, 업데이트 서비스(111)는 어느 파일이 업데이트 패키지에 포함될지를 식별한다(단계(201)). 이는 업데이트와 연관된 델타 파일과 완전한 파일을 모두 식별하는 것을 포함할 수 있다. 어느 델타 파일이 업데이트에 포함되는지가 소프트웨어 패키지의 업데이트 또는 설치 상태를 기술하는 텔레메트리 데이터를 기초로 자동으로 결정될 수 있다. 또 다른 시나리오에서, 업데이트에 포함되기 위한 델타 파일을 식별하는 것이 사용자 인터페이스를 통해 어느 델타 파일이 포함될 것인지를 특정하는 사용자 입력 또는 그 밖의 다른 지시를 수신하는 것을 포함할 수 있다.
- [0030] 그 후 업데이트 서비스(111)는 업데이트의 내용을 설명하는 매니페스트(129)를 생성한다(단계(203)). 매니페스트는 파일, 메시지, 일련의 메시지, 또는 이러한 정보를 운반하기에 적합한 그 밖의 다른 임의의 데이터 구조일 수 있다. 매니페스트는 업데이트 서비스(111)에 의해 콘텐츠 환경(101)으로 그리고 아마도 그 밖의 다른 콘텐츠 환경으로도 통신된다(단계(205)).
- [0031] 도 3은 도 1과 관련하여 언급된 동작 시나리오의 맥락에서 업데이트 에이전트(102)가 사용할 수 있는 업데이트 프로세스(300)를 더 상세히 도시한다. 도 3에 기재된 단계를 괄호를 이용해 참조하면, 업데이트 에이전트(102)는 매니페스트(129)를 조사하여 콘텐츠 환경(101)에서 어느 파일이 업데이트 대상인지를 식별할 수 있다(단계(301)). 업데이트의 대상인 각각의 파일에 대해, 업데이트 에이전트(102)는 업데이트 패키지에 포함되는 델타 파일 중 임의의 것이 대상 파일을 더 최근 버전의 파일로 업데이트하기에 적합한지 여부를 결정한다(단계(303)). 이는 각각의 델타 파일의 버전 정보를 대상 파일에 대한 업데이트 상태에 비교함으로써 이뤄질 수 있다.
- [0032] 대상 파일을 더 최근 버전으로 업데이트하는 데 사용될 수 있는 델타 파일이 존재하는 경우, 업데이트 에이전트(102)는 업데이트 서비스(111) 또는 그 밖의 다른 일부 콘텐츠 소스로부터의 델타 파일의 다운로드를 개시한다(단계(305)). 그러나 적합한 델타 파일이 이용 가능하지 않는 경우, 파일의 완전한 버전이 다운로드된다.
- [0033] 도 1을 다시 참조하면, 콘텐츠 환경(101)은 파일이 호스팅될 수 있고 파일을 업데이트할 수 있는 사용자 에이전트가 실행될 수 있는 임의의 컴퓨팅 환경일 수 있다. 콘텐츠 환경(101)의 비제한적 예를 들면, 로컬 런-타임 환경, 운영 체제 환경, 웹 브라우저 환경, 가상 머신, 샌드-박스(sand-box) 환경, 설치 환경, 및 가상 환경뿐 아니라 이의 임의의 조합 또는 변형이 있다.
- [0034] 콘텐츠 환경(101)은 단일 컴퓨팅 장치로 구현되거나 복수의 컴퓨팅 장치에 걸쳐 분산될 수 있다. 컴퓨팅 시스템(901)이 한 가지 이러한 컴퓨팅 플랫폼을 나타내며 도 9와 관련하여 이하에서 더 상세히 언급된다. 그러나 콘텐츠 환경(101)을 구현하기에 적합한 컴퓨팅 장치의 예시로는 임의의 다양한 개인 컴퓨터, 가령, 랩톱, 데스크톱, 태블릿, 및 하이브리드 컴퓨터가 있다. 그 밖의 다른 예시로는, 모바일 전화기, 스마트 폰, 게임 장치, 스마트 전화기 및 그 밖의 다른 스마트 가전기구, 웨어러블 컴퓨팅 장치, 서버 컴퓨터, 가상 서버, 가상 머신, 또는 그 밖의 다른 임의의 적합한 컴퓨팅 장치가 있을 수 있다.
- [0035] 업데이트 에이전트(102)는 임의의 소프트웨어 애플리케이션, 유틸리티, 모듈, 컴포넌트, 또는 이들의 모음을 나타내며 업데이트 프로세스(300)를 구현한다. 업데이트 에이전트(102)가 콘텐츠 환경(101)의 맥락 내에서 실행될 때, 컴퓨팅 시스템(901)으로 대표되는 적합한 컴퓨팅 장치 상에 저장되고 이에 의해 실행될 수 있다. 업데이트 에이전트(102)는 독립적으로 실행되거나 그 밖의 다른 애플리케이션, 유틸리티, 모듈, 또는 구성요소로 일체 구성될 수 있다.
- [0036] 업데이트 서비스(111)는 업데이트 에이전트(102)와 통신하고 업데이트 프로세스(200)를 구현할 수 있는 임의의 소프트웨어 애플리케이션 또는 서비스를 나타낸다. 업데이트 서비스(111)는 컴퓨팅 시스템(901)이 대표하는 적합한 컴퓨팅 장치에 저장 또는 이에 의해 실행되는 프로그램 명령으로 구현될 수 있다. 업데이트 서비스(111)는 독립형 서비스이거나 그 밖의 다른 서비스, 가령 생산성 애플리케이션 서비스, 통신 서비스, 게임 서비스, 전자상거래 서비스, 온라인 애플리케이션 상점 또는 그 밖의 다른 임의의 서비스, 서비스 조합, 또는 이들의 변형예와 통합될 수 있다.
- [0037] 파일(103 및 105)은 디지털 콘텐츠로 간주될 수 있고 이따금 업데이트될 수 있는 임의의 유형의 파일을 나타낸다. 예를 들면, 프로그램 애플리케이션 파일, 가령, 실행 파일, 라이브러리 파일, 데이터베이스 파일, 및 운영

체제 파일이 있다. 그 밖의 다른 예시로는 미디어 파일, 가령, 영화, 음악, 디지털 서적, 디지털 정기 간행물, 또는 그 밖의 다른 임의의 유형의 디지털 콘텐츠 파일이 있다.

- [0038] 도 1-3에 대한 앞서의 설명으로부터, 본 발명에 따르는 개선된 업데이트를 구현할 때 다양한 기술적 효과가 얻어질 수 있다. 앞서 언급된 바와 같이, 다른 경우라면 증가된 업데이트 생성량을 수용하는 데 필요했을 델타 파일의 순수 양이 감소되거나 완화될 수 있다. 덧붙여, 업데이트 에이전트가 델타 파일을 선택적으로 다운로드할 수 있게 하는 것이 네트워크 대역폭 및 로컬 저장 용량을 절약할 수 있다. 본 명세서에 기재된 기술적 효과를 대신하여 또는 추가로 그 밖의 다른 기술적 효과가 가능하며 본 개시 내용의 범위 내에 있는 것으로 간주될 수 있다.
- [0039] 도 4는 또 다른 구현예에서 업데이트 아키텍처(400)를 도시한다. 업데이트 아키텍처(400)는 애플리케이션 환경(401), 애플리케이션 환경(411) 및 애플리케이션 환경(421)을 포함한다. 애플리케이션 환경(401, 411, 및 421)은 업데이트 에이전트(405), 업데이트 에이전트(415), 및 업데이트 에이전트(425)를 각각 포함하며, 이들은 업데이트 서비스(431)와 통신하여 애플리케이션 환경에 설치된 애플리케이션으로의 업데이트를 수신할 수 있다. 애플리케이션은 애플리케이션 환경(401, 411, 및 421) 각각에서 파일(403), 파일(413), 및 파일(423)로 나타내어지는 파일을 포함한다. 업데이트(440)는 파일(403, 413, 및 423)이 업데이트될 수 있는 업데이트 패키지를 나타낸다.
- [0040] 업데이트(440)는 델타 파일(441), 델타 파일(443), 및 델타 파일(445)을 포함한다. 델타 파일(441, 443, 및 445) 각각은 업데이트되는 파일의 이전 버전과 파일의 업데이트될 더 최신 버전 간 차이를 나타낸다. 이 예시에서, 델타 파일(441)은 파일의 제4 버전과 제5 버전 간 차이를 나타내고, 델타 파일(443)은 파일의 제3 버전과 제5 버전 간 차이를 나타내며, 델타 파일(445)은 파일의 제2 버전과 제5 버전 간 차이를 나타낸다. 또한 업데이트(440)는 파일의 완전한 제5 버전을 나타내는 파일(447)을 포함한다.
- [0041] 도 5는 하나의 구현예에서 업데이트 서비스(431)와 애플리케이션 환경(401, 411, 및 421) 간 정보 및 데이터의 흐름을 나타내는 동작 시퀀스(500)를 도시한다. 동작 중에, 업데이트 서비스(431)는 파일 세트 내 어느 파일이 업데이트되도록 스케줄링될 수 있는지를 나타내는 정보를 포함하는 매니페스트를 생성한다. 정보는 또한 업데이트에서 어느 특정 파일이 이용 가능한지, 가령, 업데이트에서 어느 델타 파일이 있는지 그리고 업데이트에 어느 완전한 파일이 있는지를 나타낼 수 있다. 따라서 이 예시 시퀀스에서, 매니페스트가 델타 파일(441, 443, 445) 및 파일(447)을 업데이트하도록 타깃팅된 파일(애플리케이션 환경(401, 411, 및 421) 각각의 파일(403, 413, 및 423)에 대응)을 식별할 수 있다.
- [0042] 업데이트 서비스(431)는 매니페스트를 각각의 애플리케이션(401, 411, 및 421)으로 통신한다. 애플리케이션 환경(401, 411, 및 421) 각각에서의 업데이트 에이전트(405, 415, 및 425)는 매니페스트를 조사하여 업데이트를 진행할 방식을 결정할 수 있는데, 가령, 어느 델타 파일이 다운로드될 것인지를 결정하거나, 적합한 델타 파일이 없는 경우, 어느 완전한 파일이 다운로드될 것인지를 결정할 수 있다.
- [0043] 도 4의 파일(403, 413, 및 423)의 버전은 서로 상이하다. 따라서 업데이트 에이전트(405, 415, 및 425) 각각은, 존재하는 경우 델타 파일(441, 443, 및 445) 중 어느 델타 파일이 다운로드될 것인지에 대한 서로 다른 결정에 도달한다. 업데이트 에이전트(425)는 델타 파일(441)을 다운로드하기로 결정하고 이를 업데이트 서비스(431)로부터 요청한다. 파일(423)의 버전이 버전 4이기 때문에 업데이트 에이전트(425)에 의해 델타 파일(441)이 선택된다. 따라서 파일(423)은 델타 파일(441)에 의해 버전 5로 업데이트될 수 있다. 애플리케이션 환경(411)에서 업데이트 에이전트(415)는 파일(413)을 버전 3에서 버전 5로 업데이트하는 데 델타 파일(443)을 이용하기로 결정하고 이에 따라 델타 파일(443)의 다운로드를 개시할 수 있다.
- [0044] 그러나 업데이트 에이전트(405)는 업데이트(440)에서 이용 가능한 델타 파일(441, 443, 445) 중 어느 것도 파일(403)을 업데이트하는 데 적합하지 않는다고 결정한다. 이는 파일(403)의 버전이 버전 1이지만, 이러한 파일을 버전 1에서 버전 5로 만드는 어떠한 델타 파일도 생성된 적 없기 때문이다. 따라서 애플리케이션 환경(401)에서의 업데이트 에이전트(405)는 파일의 전체 버전, 즉, 파일(447)의 다운로드를 개시하여, 파일을 버전 5로 업데이트할 수 있다. 파일(447)은 파일(403)을 대체하도록 사용될 수 있다.
- [0045] 도 4를 다시 참조하면, 애플리케이션 환경(401, 411, 및 421) 각각이 파일이 호스팅될 수 있고 파일을 업데이트할 수 있는 사용자 에이전트가 실행될 수 있는 임의의 컴퓨팅 환경을 나타낼 수 있다. 애플리케이션 환경(401, 411, 및 421)의 비제한적 예를 들면, 프로그램 애플리케이션, 로컬 런-타임 환경, 운영 체제 환경, 웹 브라우저 애플리케이션, 가상 머신, 샌드-박스 환경, 설치 환경, 및 가상화된 환경뿐 아니라 이들의 임의의 조합 또는 변

형이 있다. 애플리케이션 환경(401, 411 및 421) 각각은 단일 컴퓨팅 장치에서 구현되거나 복수의 컴퓨팅 장치에 걸쳐 분산될 수 있다. 컴퓨팅 시스템(901)은 이러한 한 가지 컴퓨팅 플랫폼을 나타내고 도 9를 참조하여 이하에서 더 상세히 언급된다.

[0046] 업데이트 에이전트(405, 415, 및 425) 각각은 업데이트 서비스(431)와 통신할 수 있고 업데이트 프로세스를 구현할 수 있는 임의의 소프트웨어 애플리케이션, 유틸리티, 모듈, 구성요소, 또는 이의 조합을 나타낸다. 업데이트 에이전트(405, 415, 및 425) 각각은 상황 애플리케이션 환경(401, 411, 및 421) 내에서 실행될 수 있고 컴퓨팅 시스템(901)으로 대표되는 적합한 컴퓨팅 장치에 저장되거나 이에 의해 실행될 수 있다. 업데이트 에이전트(405, 415, 및 425)는 독립형으로 실행되거나 그 밖의 다른 애플리케이션, 유틸리티, 모듈 또는 구성요소로 통합될 수 있다.

[0047] 업데이트 서비스(431)는 업데이트 에이전트(405, 415, 및 425)와 통신할 수 있는 임의의 소프트웨어 애플리케이션 또는 서비스를 나타내고 업데이트 프로세스를 구현할 수 있다. 업데이트 서비스(431)는 컴퓨팅 시스템(901)으로 대표되는 적합한 컴퓨팅 장치에 저장되거나 이에 의해 실행되는 프로그램 명령으로 구현될 수 있다. 업데이트 서비스(431)는 독립형 서비스이거나 그 밖의 다른 서비스, 가령, 생산성 애플리케이션 서비스, 통신 서비스, 게임 서비스, 전자상거래 서비스, 온라인 애플리케이션 상점, 또는 그 밖의 다른 임의의 유형의 서비스, 서비스의 조합, 또는 이들의 변형과 통합될 수 있다.

[0048] 파일(403, 413, 및 423) 각각은 디지털 콘텐츠로 간주될 수 있고 이따금 업데이트될 수 있는 임의의 유형의 파일을 나타낸다. 예를 들자면, 프로그램 애플리케이션 파일, 가령, 실행 파일, 라이브러리 파일, 데이터베이스 파일, 및 운영 체제 파일이 있다. 그 밖의 다른 예를 들면, 미디어 파일, 가령, 영화, 음악, 디지털 서적, 디지털 정기 간행물, 또는 그 밖의 다른 임의의 유형의 디지털 콘텐츠 파일이 있다.

[0049] 도 4-5에 대한 앞서의 설명으로부터, 본 발명에 따르는 개선된 업데이트를 구현할 때 다양한 기술적 효과가 얻어질 수 있음이 자명할 수 있다. 이 예시에서, 각각의 업데이트 에이전트가 모든 이용 가능한 델타 파일을 다운로드하는 것 대신 각각의 업데이트 에이전트가 서로 다른 개별 델타 파일을 다운로드할 때 대역폭이 보존된다. 덧붙여, 파일을 버전 1에서 버전 5로 업데이트하기 위해 델타 파일이 유지될 필요가 없음으로써, 생산성 및 유지관리 자원이 보존된다.

[0050] 도 6은 복수의 파일이 업데이트의 대상이 되는 하나의 구현예에서 업데이트 아키텍처(600)를 도시한다. 업데이트 아키텍처(600)는 애플리케이션 환경(601), 애플리케이션 환경(611), 및 애플리케이션 환경(621)을 포함한다. 애플리케이션 환경(601, 611, 및 621) 각각은 업데이트 서비스(631)와 통신하여 애플리케이션 환경(601, 611, 및 621) 각각에 설치된 소프트웨어 패키지의 업데이트를 촉진시키는 업데이트 에이전트를 포함한다.

[0051] 특히, 애플리케이션 환경(601)은 소프트웨어 패키지의 일부인 파일(603) 및 파일(604)을 업데이트하는 것을 보조하는 업데이트 에이전트(605)를 포함한다. 애플리케이션 환경(611)은 역시 소프트웨어 패키지에 속하는 파일(613) 및 파일(614)을 업데이트하는 것을 보조하는 업데이트 에이전트(615)를 포함한다. 애플리케이션 환경(621)은 역시 소프트웨어 패키지의 일부인 파일(623) 및 파일(626)을 업데이트하기 위한 업데이트 에이전트(625)를 포함한다. 따라서, 각각의 애플리케이션 환경(601, 611, 및 621)에서 각각의 파일 세트가 각각의 애플리케이션 환경에 설치된 소프트웨어 패키지의 인스턴스를 나타냄이 자명할 수 있다.

[0052] 소프트웨어 패키지의 각각의 인스턴스가 다른 인스턴스에 비해 약간 다를 수 있다. 이는, 예를 들어, 한 인스턴스가 다른 것보다 덜 자주 업데이트되기 때문에 발생할 수 있다. 예를 들자면, 파일(603)이 패키지 내 파일 "alpha.exe"의 제1 버전을 나타내고 파일(604)이 패키지 내 또 다른 파일 "kilo.exe"의 제2 버전을 나타낸다. 애플리케이션 환경(611)에서, 파일(613)은 alpha.exe 파일의 제3 버전을 나타내고, 파일(614)은 kilo.exe 파일의 제3 버전을 나타낸다. 애플리케이션 환경(621) 내에서 파일(623)은 alpha.exe 파일의 제4 버전을 나타내고 파일(626)은 임의의 시점에서 소프트웨어 패키지의 일부였을 수 있는 파일 "sierra.exe"의 임의의 버전을 나타낸다. 이들 예시로부터, 각각의 애플리케이션 환경이 설치된 동일한 소프트웨어 패키지를 포함하는 동안, 패키지 내 각각의 파일과 연관된 업데이트 상태는 환경별로 달라질 수 있고, 실제로는 파일별로도 달라질 수 있음을 알 수 있다.

[0053] 업데이트 서비스(631)는 애플리케이션 환경(601, 611, 및 621)에 설치된 소프트웨어 패키지의 인스턴스에 업데이트될 수 있는 업데이트를 나타내는 소프트웨어 업데이트(640)를 포함한다. 예를 들어, 소프트웨어 업데이트(640)가 파일 업데이트(650) 및 파일 업데이트(660)로 나타나는 2개의 파일에 대한 업데이트 패키지를 포함한다. 파일 업데이트(650)는 델타 파일(651), 델타 파일(653), 델타 파일(655) 및 완전한 파일(657)로 나타

나는, 애플리케이션 환경(601, 611, 및 621)의 alpha.exe 파일을 파일의 제5 버전으로 업데이트하기 위한 델타 파일 및 완전한 파일을 포함한다. 파일 업데이트(660)는 델타 파일(661), 델타 파일(663), 델타 파일(665), 및 완전한 파일(667)로 나타나는, 애플리케이션 환경(601, 611, 및 621)에서 kilo.exe 파일을 파일의 제5 버전으로 업데이트하기 위한 델타 파일 및 완전한 파일을 포함한다. 소프트웨어 업데이트(640)는 또한 소프트웨어 업데이트(640)의 내용을 기술하는 매니페스트(670)를 포함한다.

[0054] 동작 중에, 소프트웨어 업데이트(640)는 빌드 또는 개발 환경(또는 둘 모두)에서 생성되고 업데이트 서비스(631)에서 선보일 준비가 될 수 있다. 소프트웨어 업데이트(640)가 준비되면, 통지가 애플리케이션 환경(601, 611, 621)으로 제공된다. 업데이트 에이전트(605, 615, 및 625)가 도 7에 상세히 도시된 이러한 상황에서 업데이트 프로세스(700)를 채용한다.

[0055] 도 7을 참조하면, 업데이트 에이전트(가령, 업데이트 에이전트(605, 615, 및 625))가 새 업데이트 패키지가 이용 가능하다는 통지를 수신한다(단계(701)). 이는, 예를 들어, 업데이트 에이전트가 업데이트 서비스를 폴링(polling)할 때 또는 이러한 통지가 업데이트 에이전트로 푸시(push)되거나 그 밖의 다른 방식으로 제공될 때 발생할 수 있다. 이에 응답하여, 업데이트 에이전트는 업데이트의 일부로서 어느 파일이 업데이트될 것인지를 기술하는 업데이트 매니페스트를 요청 및 다운로드한다(단계(703)).

[0056] 매니페스트는 또한 어느 델타 파일 및 어느 완전한 파일이 이용 가능한지를 기술한다. 델타 파일은 각각의 델타 파일이 업데이트할 수 있는 대응하는 파일의 버전과 연관되어 기술된다. 업데이트 에이전트는 매니페스트를 처리하여, 업데이트될 대상으로서 매니페스트 내에 식별된 다양한 파일을 열거할 수 있다(단계(705)). 열거된 각각의 파일이 업데이트에 의해 에이전트로 분석되어, 각각이 파일이 소프트웨어 패키지의 로컬 인스턴스에서 기존 파일에 비해 새 파일인지 여부를 결정할 수 있다(단계(707)). 파일이 새 파일인 경우, 업데이트 에이전트가 전체 파일을 다운로드하고(단계(709)) 이를 이용해 파일의 새 버전을 생성할 수 있다(단계(711)).

[0057] 그러나 파일이 새로운 것이 아닌 경우, 업데이트 에이전트는 매니페스트를 조사하여 델타 파일이 각각의 파일을 업데이트하는 데 적합한 업데이트에 포함되는지 여부를 결정할 수 있다(단계(713)). 그렇지 않은 경우, 업데이트 에이전트가 전체 파일을 다운로드하고(단계(709)) 파일의 새 버전을 생성한다. 적합한 델타 파일이 존재하고 업데이트에 포함되는 경우, 업데이트 에이전트는 델타 파일을 다운로드하고(단계(715)) 델타 파일을 이용해 파일의 업데이트된 버전을 생성한다(단계(717)).

[0058] 새 파일 또는 업데이트된 파일을 생성한 후, 업데이트 에이전트가 매니페스트를 기초로 열거된 그 밖의 다른 임의의 파일이 업데이트 예정인지 여부를 결정한다(단계(719)). 업데이트 예정인 경우, 업데이트 프로세스(700)가 단계(707)로 복귀한다. 그러나 업데이트가 필요한 파일이 남아있지 않은 경우, 업데이트는 완료된다(단계(721)).

[0059] 도 8은 애플리케이션 환경(601, 611, 및 621)에서 업데이트 에이전트(605, 615 및 625)가 도 6에 제공된 시나리오의 맥락에서 업데이트 프로세스(700)를 채용할 때 발생할 수 있는 결과를 도시하는 테이블(800)을 포함한다. 테이블(800)은 업데이트되는 파일의 이름(alpha.exe 및 kilo.exe)으로 정의되는 행 및 각각의 파일이 업데이트되는 애플리케이션 환경에 대응하는 열을 포함한다. 행 및 열에 의해 정의된 각각의 셀이 업데이트 프로세스(700)마다, 특정 애플리케이션 환경 내 특정 파일이 업데이트된 방식을 표현하는 정보를 포함한다.

[0060] 이 예시적 시나리오에서, 파일 업데이트(650)는 alpha.exe 파일과 관련하여 애플리케이션 환경(601)에서 적용될 것이다. 파일(603)을 버전 1에서 버전 5로 업데이트하기 위해 어떠한 적합한 델타 파일도 포함되어 있지 않기 때문에, 업데이트 에이전트(605)는 alpha.exe 파일을 업데이트하기 위해 완전한 파일을 다운로드하기로 결정할 것이다. 반대로, 업데이트 에이전트(605)는 kilo.exe 파일을 버전 2에서 버전 5로 업데이트하기 위해, 델타 파일(665)을 다운로드할 것이다.

[0061] 애플리케이션 환경(611)과 관련하여, 업데이트 에이전트(615)는 파일(613)을 alpha.exe 파일의 제3 버전에서 제5 버전으로 업데이트하기 위해 델타 파일(653)을 다운로드하기로 결정할 것이다. 덧붙여, 업데이트 에이전트(615)는 파일(614)을 kilo.exe 파일의 제3 버전에서 제5 버전으로 업데이트하기 위해 델타 파일(663)을 다운로드하기로 결정할 것이다.

[0062] 애플리케이션 환경(621)과 관련하여, 업데이트 에이전트(625)는 파일(623)을 alpha.exe 파일의 제4 버전에서 제5 버전으로 업데이트하는 데 사용되도록 델타 파일(651)을 다운로드하기로 결정할 것이다. 그러나 kilo.exe 파일의 로컬 버전을 업데이트하는 것과 관련하여, 업데이트 에이전트(625)는 파일이 로컬하게 존재하지 않았던 상황에 직면했을 것이다. 따라서 업데이트 에이전트(625)는 완전한 파일(667)을 다운로드하여 제5 버전인

kilo.exe 파일의 완전한 버전을 획득할 수 있다.

- [0063] 테이블(800)이 또한 선택사항으로서 업데이트 에이전트(625)가 sierra.exe라는 이름의 파일(626)을 삭제할 수 있음을 보여준다. 이는 업데이트 패키지가 소프트웨어 제품군에 이전에 포함됐던 파일을 더는 포함하지 않는 일부 상황에서 발생할 수 있다. 일부 시나리오에서, 업데이트 에이전트는 매니페스트에서 파일 부재에 의해 상기 파일이 삭제됐을 수 있다고 결론 내릴 수 있다. 또 다른 시나리오에서, 매니페스트는 제거될 파일을 명시적으로 특정할 수 있다.
- [0064] 도 6을 다시 참조하며, 애플리케이션 환경(601, 611, 및 621) 각각이 파일이 호스팅될 수 있고 파일을 업데이트할 수 있는 사용자 에이전트가 실행될 수 있는 임의의 컴퓨팅 환경일 수 있다. 애플리케이션 환경(601, 611, 및 621)의 비제한적 예를 들면, 프로그램 애플리케이션, 로컬 런-타임 환경, 운영 체제 환경, 웹 브라우저 애플리케이션, 가상 머신, 샌드-박스 환경, 설치 환경, 및 가상화된 환경뿐 아니라 이들의 임의의 조합 또는 변형까지 일 수 있다. 애플리케이션 환경(601, 611 및 621) 각각은 단일 컴퓨팅 장치로 구현되거나 복수의 컴퓨팅 장치 간에 분산될 수 있다. 컴퓨팅 시스템(901)은 한 가지 이러한 컴퓨팅 플랫폼을 나타내며 도 9와 관련하여 이하에서 더 상세히 언급된다.
- [0065] 업데이트 에이전트(605, 615, 및 625) 각각이 업데이트 서비스(631)와 통신할 수 있고 업데이트 프로세스를 이행할 수 있는 임의의 소프트웨어 애플리케이션, 유틸리티, 모듈, 컴포넌트 또는 이들의 모음을 나타낸다. 업데이트 에이전트(605, 615, 및 625) 각각은 애플리케이션 환경(601, 611, 및 621)의 맥락 내에서 각각 실행될 수 있고 컴퓨팅 시스템(901)으로 대표되는 적합한 컴퓨팅 장치 상에 저장되고 이에 의해 실행될 수 있다. 업데이트 에이전트(605, 615, 및 625)는 독립적인 방식으로 실행되거나 그 밖의 다른 애플리케이션, 유틸리티, 모듈 또는 구성요소로 일체 구성될 수 있다.
- [0066] 업데이트 서비스(631)는 업데이트 에이전트(605, 615, 625)와 통신할 수 있고 업데이트 프로세스(700)를 구현할 수 있는 임의의 소프트웨어 애플리케이션 또는 서비스를 나타낸다. 업데이트 서비스(631)는 컴퓨팅 시스템(910)으로 대표되는 적합한 컴퓨팅 장치에 저장되거나 이에 의해 실행되는 프로그램 명령으로 구현될 수 있다. 업데이트 서비스(631)는 독립형 서비스이거나 다른 서비스, 가령, 생산성 애플리케이션 서비스, 통신 서비스, 게임 서비스, 전자상거래 서비스, 온라인 애플리케이션 스토어, 또는 그 밖의 다른 임의의 유형의 서비스, 서비스의 조합, 또는 이들의 변형과 일체 구성될 수 있다.
- [0067] 파일(603, 604, 613, 614, 623, 및 624) 각각은 디지털 콘텐츠로 간주될 수 있고 이따금 업데이트될 수 있는 임의의 유형의 파일을 나타낸다. 예시로는, 프로그램 애플리케이션 파일, 가령, 실행 파일, 라이브러리 파일, 데이터베이스 파일, 및 운영 체제 파일이 있다. 또 다른 예로는 미디어 파일, 가령, 영화, 음악, 디지털 서적, 및 디지털 정기 간행물이 있다.
- [0068] 도 6-8의 설명으로부터 본 발명에 따르는 개선된 업데이트를 구현할 때 다양한 기술적 효과가 달성될 수 있음이 자명할 수 있다. 다른 가능한 이점들보다도, 완전한 파일보다 델타 파일을 제공함으로써, 대역폭이 보존된다. 또한 제한된 델타 파일 세트가 생산 및 유지관리 자원을 보존한다.
- [0069] 도 9는 본 명세서에 기재된 다양한 동작 아키텍처, 시나리오 및 프로세스가 구현될 수 있는 임의의 시스템 또는 시스템 모음을 나타내는 컴퓨팅 시스템(901)을 도시한다. 컴퓨팅 시스템(901)의 비제한적 예를 들면, 스마트폰, 랩톱 컴퓨터, 태블릿 컴퓨터, 데스크톱 컴퓨터, 하이브리드 컴퓨터, 게임 머신, 가상 머신, 스마트 텔레비전, 스마트 시계 및 그 밖의 다른 웨어러블 장치, 및 이의 임의의 변형 또는 조합이 있다. 그 밖의 다른 예시로는 서버 컴퓨터, 랙 서버(rack server), 웹 서버, 클라우드 컴퓨팅 플랫폼, 및 데이터 센터 장비, 및 그 밖의 다른 임의의 유형의 물리적 또는 가상 서버 머신, 및 이의 임의의 변형 또는 조합이 있다.
- [0070] 컴퓨팅 시스템(901)은 단일 장치, 시스템, 또는 장치로서 구현되거나 복수의 장치, 시스템, 또는 장치로서 분산된 방식으로 구현될 수 있다. 컴퓨팅 시스템(901)의 비제한적 예를 들면, 처리 시스템(902), 저장 시스템(903), 소프트웨어(905), 통신 인터페이스 시스템(907) 및 사용자 인터페이스 시스템(909)이 있다. 처리 시스템(902)은 저장 시스템(903), 통신 인터페이스 시스템(907) 및 사용자 인터페이스(909)와 동작 가능하게 연결된다.
- [0071] 처리 시스템(902)은 저장 시스템(903)으로부터 소프트웨어(905)를 로딩하고 실행한다. 소프트웨어(905)는 적어도, 상기 구현예와 관련하여 언급된 업데이트 프로세스(200), 업데이트 프로세스(300) 및 업데이트 프로세스(700)를 나타내는 업데이트 프로세스(906)를 포함한다. 디지털 콘텐츠가 업데이트되는 방식을 개선하기 위해 처리 시스템(902)에 의해 실행될 때, 소프트웨어(905)는 적어도 다양한 프로세스, 동작 시나리오, 및 시퀀스에 대해 본 명세서에서 기재된 바와 같이 동작하도록 처리 시스템(902)에게 지시한다. 컴퓨팅 시스템(901)은 선택사

항으로서 추가 장치, 특징부, 또는 기능부를 포함할 수 있으며, 이들은 간략성을 위해 언급되지 않는다.

- [0072] 도 9를 다시 참조하면, 처리 시스템(902)은 저장 시스템(903)으로부터 소프트웨어(905)를 불러오고 실행하는 마이크로프로세서 및 그 밖의 다른 회로를 포함할 수 있다. 처리 시스템(902)은 단일 처리 장치 내에서 구현될 수 있지만, 또한 프로그램 명령을 실행하는 데 협력하는 복수의 처리 장치 또는 서브-시스템 간에 분산될 수 있다. 처리 시스템(902)의 예시로는 범용 중앙 처리 장치, 주문형 프로세서, 로직 장치뿐 아니라, 그 밖의 다른 임의의 유형의 처리 장치, 이들의 조합 또는 변형이 있다.
- [0073] 소프트웨어 시스템(903)은 처리 시스템(902)에 의해 관독될 수 있고 소프트웨어(905)를 저장할 수 있는 임의의 컴퓨터 관독형 저장 매체를 포함할 수 있다. 저장 시스템(903)은 정보, 가령, 컴퓨터 관독형 명령, 데이터 구조, 프로그램 모듈 또는 그 밖의 다른 데이터의 저장을 위한 임의의 방법 또는 기법으로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함할 수 있다. 저장 매체의 예시로는 랜덤 액세스 메모리, 리드 온리 메모리, 자기 디스크, 광 디스크, 플래시 메모리, 가상 메모리 및 비-가상 메모리, 자기 카세트, 자기 테이프, 자기 디스크 저장 또는 그 밖의 다른 자기 저장 장치, 또는 그 밖의 다른 임의의 적합한 저장 매체가 있다. 어느 경우라도 전파되는 신호는 컴퓨터 관독형 저장 매체이다.
- [0074] 컴퓨터 관독형 저장 매체에 추가로, 일부 구현예에서, 저장 시스템(903)은 소프트웨어(905) 중 적어도 일부가 내부적으로 또는 외부적으로 통신될 때 통하는 컴퓨터 관독형 통신 매체를 더 포함할 수 있다. 저장 시스템(903)은 단일 저장 장치로서 구현될 수 있지만, 함께 위치하거나 서로 분산되어 있는 복수의 저장 장치 또는 서브-시스템에 걸쳐 구현될 수도 있다. 저장 시스템(903)은 처리 시스템(902) 또는 가능하면 다른 시스템과 통신할 수 있는 추가 요소, 가령, 제어기를 포함할 수 있다.
- [0075] 소프트웨어(905)는 프로그램 명령으로 구현될 수 있고 그 밖의 다른 기능이, 처리 시스템(902)에 의해 실행될 때, 본 명세서에 도시된 다양한 동작 시나리오, 시퀀스, 및 프로세스에 대해 기재된 바와 같이 동작하도록 처리 시스템(902)에게 지시할 수 있다. 예를 들어, 소프트웨어(905)는 업데이트 에이전트 또는 업데이트 서비스 및 이들 각자의 기능을 구현하기 위한 프로그램 명령을 포함할 수 있다.
- [0076] 특히, 프로그램 명령은 본 명세서에 기재된 다양한 프로세스 및 동작 시나리오를 수행하도록 협력 또는 그 밖의 다른 방식으로 대화하는 다양한 구성요소 또는 모듈을 포함할 수 있다. 다양한 구성요소 또는 모듈은 컴파일된 또는 해석된 명령으로 구현되거나 명령의 그 밖의 다른 변형 또는 조합으로 구현될 수 있다. 단일 스레드 환경 또는 멀티-스레드 환경에서 또는 이의 조합에 따라, 다양한 구성요소 또는 모듈은 동기식 또는 비동기식으로, 직렬로, 또는 병렬로 실행될 수 있다. 소프트웨어(905)는 업데이트 프로세서(906)에 추가로 추가 프로세스, 프로그램 또는 구성요소, 가령, 운영 체제 소프트웨어 또는 그 밖의 다른 애플리케이션 소프트웨어를 포함할 수 있다. 소프트웨어(905)는 또한 펌웨어 또는 처리 시스템(902)에 의해 실행 가능한 머신-관독형 처리 명령의 그 밖의 다른 형태를 더 포함할 수 있다.
- [0077] 일반적으로 소프트웨어(905)는, 처리 시스템(902)으로 로딩되고 실행될 때, (컴퓨팅 시스템(901)으로 대표되는) 적합한 장치, 시스템 또는 디바이스를 범용 컴퓨팅 시스템에서 디지털 콘텐츠를 위한 개선된 업데이트를 촉진하도록 커스텀화된 특수 목적 컴퓨팅 시스템으로 변환할 수 있다. 실제로, 저장 시스템(903) 상의 인코딩 소프트웨어(905)는 저장 시스템(903)의 물리 구조를 변환할 수 있다. 물리 구조의 특정 변환이 이 기재의 서로 다른 구현의 다양한 인자들에 따라 달라질 수 있다. 이러한 인자들의 비제한적 예시로는 저장 시스템(903)의 저장 매체를 구현하는 데 사용되는 기법, 컴퓨터-저장 매체가 주 기억장치인지 보조 기억장치인지 여부, 및 그 밖의 다른 인자를 포함할 수 있다.
- [0078] 예를 들어, 컴퓨터 관독형 저장 매체가 반도체 기반 메모리로서 구현되는 경우, 프로그램 명령이 인코딩될 때, 가령, 트랜지스터, 커패시터, 또는 반도체 메모리를 구성하는 그 밖의 다른 이산 회로 요소의 상태를 변환함으로써, 소프트웨어(905)는 반도체 메모리의 물리적 상태를 변환할 수 있다. 자기 또는 광학 매체와 관련하여 유사한 변환이 발생할 수 있다. 본 발명의 범위 내에서 물리 매체의 그 밖의 다른 변환이 가능하며, 상기의 예시는 본 발명의 설명을 용이하게 하기 위한 목적으로 제공된 것에 불과하다.
- [0079] 예시로서 도 1-3을 다시 참조하면, 컴퓨팅 시스템(901)으로 대표되는 컴퓨팅 시스템의 동작을 통해, 업데이트 아키텍처(100) 및 구체적으로 파일(103)과 관련하여 변환이 수행될 수 있다. 예를 들어, 파일(103)의 업데이트 상태가 버전 3으로 업데이트된 후일 수 있다. 업데이트 서비스(111)가 업데이트 프로세스(200)를 이용하고 업데이트 에이전트(102)가 업데이트 프로세스(300)를 채용하면, 델타 파일(123)이 다운로드되고 사용되어, 파일(103)을 버전 5로 업데이트함으로써, 새 업데이트 상태로 변경할 수 있다.

- [0080] 일반적으로 컴퓨팅 시스템(901)은 개선된 디지털 콘텐츠 업데이트를 구현하기 위해 소프트웨어(905)가 배포되고 실행될 수 있는 컴퓨팅 시스템을 대표하는 것을 의도됨이 이해될 수 있다. 그러나 컴퓨팅 시스템(901)은 또한 소프트웨어(905)가 스테이징될 수 있는 임의의 컴퓨팅 시스템으로 적합할 수 있으며, 하나 또는 둘 모두가 배포, 실행, 또는 추가 분산을 위해 분산, 전송, 다운로드, 또는 그 밖의 다른 방식으로 또 다른 컴퓨팅 시스템으로 제공될 수 있다.
- [0081] 통신 인터페이스 시스템(907)은 통신 네트워크(도시되지 않음)를 통해 다른 컴퓨팅 시스템(도시되지 않음)과의 통신을 가능하게 하는 통신 연결 및 장치를 포함할 수 있다. 시스템들 간 통신을 가능하게 하는 연결 및 장치의 예시로는 네트워크 인터페이스 카드, 안테나, 전력 증폭기, RF 회로, 트랜시버 및 그 밖의 다른 통신 회로가 있다. 연결 및 장치는 통신 매체, 가령, 금속, 유리, 공기, 또는 그 밖의 다른 적합한 통신 매체를 통해 타 컴퓨팅 시스템 또는 시스템 네트워크와 통신 신호를 교환할 수 있다. 앞서 언급된 매체, 연결, 및 장치가 잘 알려져 있으며 본 명세서에서 더는 언급될 필요가 없다.
- [0082] 사용자 인터페이스 시스템(909)은 선택사항이며, 예를 들면, 키보드, 마우스, 음성 입력 장치, 사용자로부터의 터치 제스처를 수신하기 위한 터치 입력 장치, 비-터치 제스처 및 그 밖의 다른 사용자 모션을 검출하기 위한 모션 입력 장치, 및 사용자로부터의 사용자 입력을 수신할 수 있는 그 밖의 다른 유사한 입력 장치 및 연관된 처리 요소가 있을 수 있다. 출력 장치, 가령, 디스플레이, 스피커, 햅틱 장치, 및 그 밖의 다른 유형의 출력 장치가 또한 사용자 인터페이스 시스템(909)에 포함될 수 있다. 일부 경우, 입력 장치와 출력 장치가 단일 장치로 결합될 수 있다. 앞서 언급된 사용자 입력 및 출력 장치가 종래 기술에 알려져 있으며 본 명세서에서 더는 설명될 필요가 없다.
- [0083] 사용자 인터페이스 시스템(909)은 앞서 언급된 다양한 사용자 입력 및 출력 장치를 지원하는 처리 시스템(902)에 의해 실행 가능한 연관된 사용자 인터페이스 소프트웨어를 더 포함할 수 있다. 개별적으로 또는 서로 그리고 타 하드웨어 및 소프트웨어 요소와 결합하여, 사용자 인터페이스 소프트웨어 및 사용자 인터페이스 장치가 그래픽 사용자 인터페이스, 자연 사용자 인터페이스, 또는 그 밖의 다른 임의의 유형의 사용자 인터페이스를 지원할 수 있다.
- [0084] 컴퓨팅 시스템(901)과 타 컴퓨팅 시스템(도시되지 않음) 간 통신이 통신 네트워크를 통해, 그리고 다양한 통신 프로토콜, 프로토콜 조합, 또는 이들의 변형에 따라 발생할 수 있다. 예를 들면, 인트라넷, 인터넷, 인터넷(Internet), 로컬 영역 네트워크, 광역 네트워크, 무선 네트워크, 유선 네트워크, 가상 네트워크, 소프트웨어 정의 네트워크, 데이터 센터 버스, 컴퓨팅 백플레인, 또는 그 밖의 다른 임의의 유형의 네트워크, 네트워크 조합, 또는 이의 변형이 있다. 앞서 언급된 통신 네트워크 및 프로토콜이 잘 알려져 있으며 본 명세서에서 더는 상세히 기재되지 않는다. 그러나 사용될 수 있는 일부 통신 프로토콜의 비제한적 예로는, 인터넷 프로토콜(IP, IPv4, IPv6 등), 전송 제어 프로토콜(TCP), 및 사용자 데이터그램 프로토콜(UDP)뿐 아니라 그 밖의 다른 임의의 적합한 통신 프로토콜, 이의 변형 또는 조합까지 있다.
- [0085] 데이터, 콘텐츠, 또는 그 밖의 다른 임의의 유형의 정보가 교환되는 앞서 언급된 예시들 중 임의의 예시에서, 정보의 교환이 다양한 프로토콜, 가령, FTP(file transfer protocol), HTTP(hypertext transfer protocol), REST(representational state transfer), WebSocket, DOM(Document Object Model), HTML(hypertext markup language), CSS(cascading style sheets), HTML5, XML(extensible markup language), JavaScript, JSON(JavaScript Object Notation), 및 AJAX(Asynchronous JavaScript and XML), 그 밖의 다른 임의의 적합한 프로토콜, 이의 변형 또는 조합 중 임의의 프로토콜에 따라 발생할 수 있다.
- [0086] 도 1-9가 일반적으로 비교적 적은 동작 시나리오 및 시퀀스를 도시하지만, 본 명세서에 개시된 개념이 반복적으로 적용될 수 있음을 알 수 있다. 예를 들어, 본 명세서에 개시된 업데이트 서비스는 임의의 개수의 콘텐츠 및 애플리케이션 환경의 지원 하에 배포될 수 있다.
- [0087] 도면에 제공된 기능적 블록도, 동작 시나리오 및 시퀀스, 흐름도가 본 발명의 신규한 양태를 수행하기 위한 예시적 시스템, 환경, 및 방법을 나타낸다. 설명의 단순성을 위해, 본 명세서에 포함된 방법이 기능적 다이어그램, 동작 시나리오 또는 시퀀스 또는 흐름도의 형태로 존재하고 일련의 동작으로 기재될 수 있지만, 방법이 동작의 순서로 한정되지 않고, 일부 동작은 서로 다른 순서로 발생하고 및/또는 본 명세서에 도시되고 기재된 타 동작과 동시에 발생할 수 있음이 이해되어야 한다. 예를 들어, 해당 분야의 통상의 기술자라면 방법은 대안적으로 가령, 상태 다이어그램의 일련의 상호 관련된 상태 또는 이벤트로서 표현될 수 있다. 덧붙여, 하나의 방법으로 도시되는 모든 동작이 신규한 구현예를 위해 반드시 필요한 것은 아닐 수 있다.

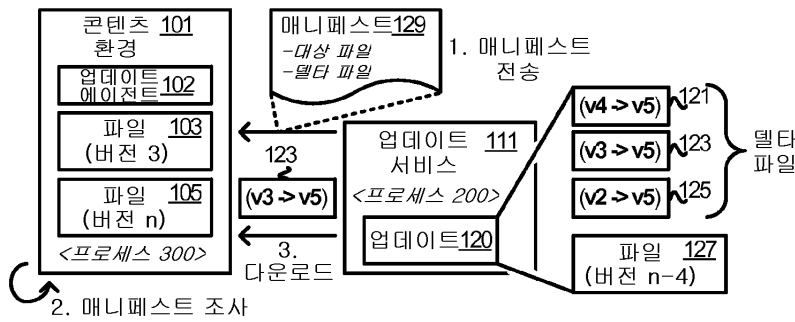
[0088]

본 명세서에 포함되는 기재 및 도면이 최상의 옵션을 제작하고 사용하는 방식을 해당 분야의 통상의 기술자에게 설명하기 위한 특정 구현예를 도시한다. 본 발명의 원리를 설명하기 위한 목적으로, 일부 종래의 양태는 단순화되거나 생략됐을 수 있다. 해당 분야의 통상의 기술자라면 이들 구현예가 본 발명의 범위 내에 속하는 것으로 이해될 것이다. 해당 분야의 통상의 기술자라면 앞서 기재된 특징을 다양한 방식으로 조합되어 복수의 구현예를 형성할 수 있음을 알 것이다. 따라서, 본 발명은 앞서 기재된 특정 구현예에 한정되지 않으며, 청구항 및 이의 균등물에 의해서만 한정된다.

도면

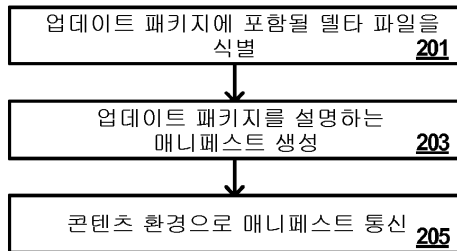
도면1

100



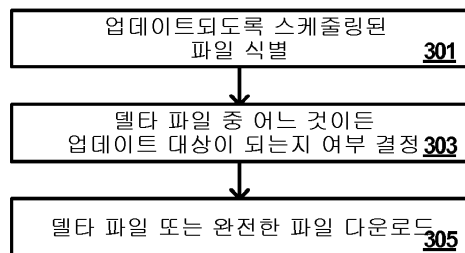
도면2

200

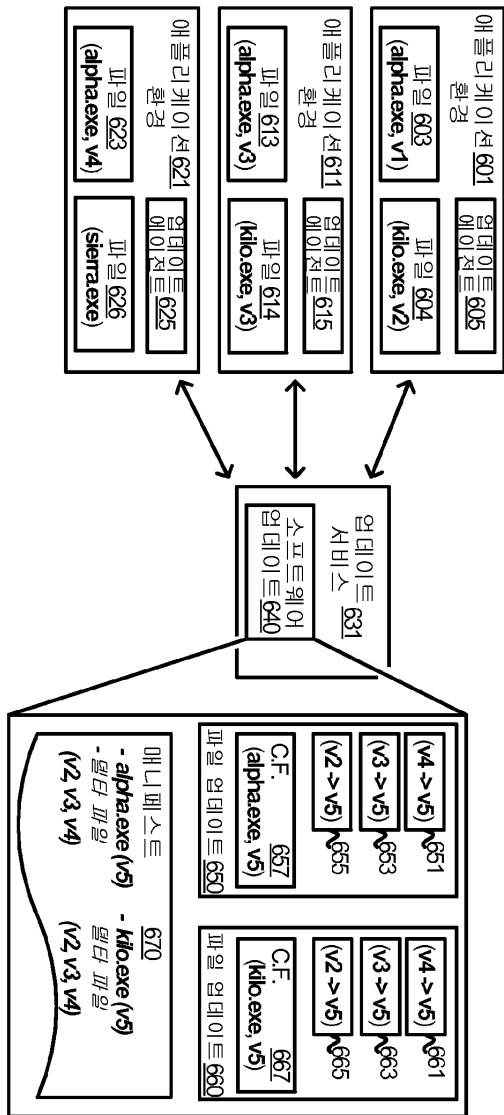


도면3

300



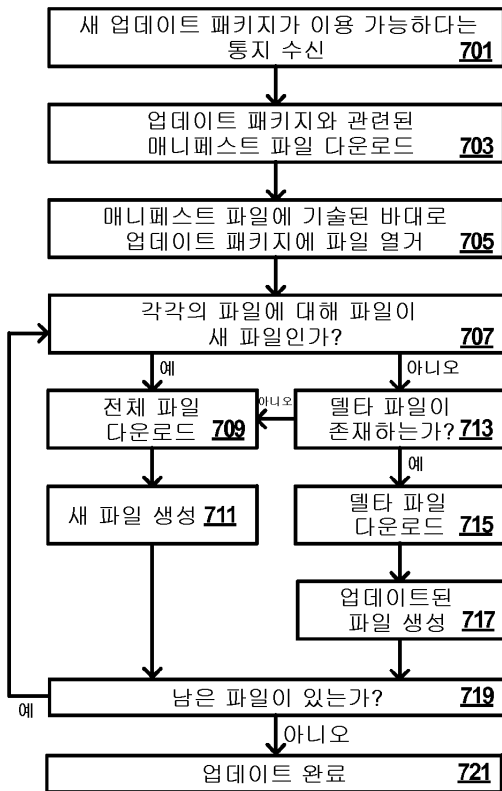
도면6



600

도면7

700



도면8

800

업데이트 \ 환경	애플리케이션 환경601	애플리케이션 환경611	애플리케이션 환경621
파일 업데이트650 (alpha.exe, v5)	완전한 파일 다운로드 (파일 657)	델타 파일 다운로드 (델타 파일 653)	델타 파일 다운로드 (델타 파일 651)
파일 업데이트660 (kilo.exe, v5)	델타 파일 다운로드 (델타 파일 665)	델타 파일 다운로드 (델타 파일 663)	완전한 파일 다운로드 (파일 667)
			sierra.exe 삭제

도면9

