



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0109375
(43) 공개일자 2015년10월01일

(51) 국제특허분류(Int. Cl.)
G06Q 10/06 (2012.01)
(52) CPC특허분류
G06Q 10/0633 (2013.01)
G06Q 10/06311 (2013.01)
(21) 출원번호 10-2015-7020029
(22) 출원일자(국제) 2013년12월20일
심사청구일자 없음
(85) 번역문제출일자 2015년07월22일
(86) 국제출원번호 PCT/US2013/077213
(87) 국제공개번호 WO 2014/116384
국제공개일자 2014년07월31일
(30) 우선권주장
61/755,892 2013년01월23일 미국(US)
61/833,770 2013년06월11일 미국(US)

(71) 출원인
툼슨 라이센싱
프랑스 92130 이씨레물리노 루 잔다르크 1-5
(72) 발명자
워커 마크 프로이
미국 91384 캘리포니아주 캐스텍 캠프리지 애비뉴 30027
클로슨 커트 패트릭
미국 91504 캘리포니아주 버뱅크 델라웨어 로드 537
런델 스티브
미국 91007 캘리포니아주 아라카디아 포틀라 드라이브 1020
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 15 항

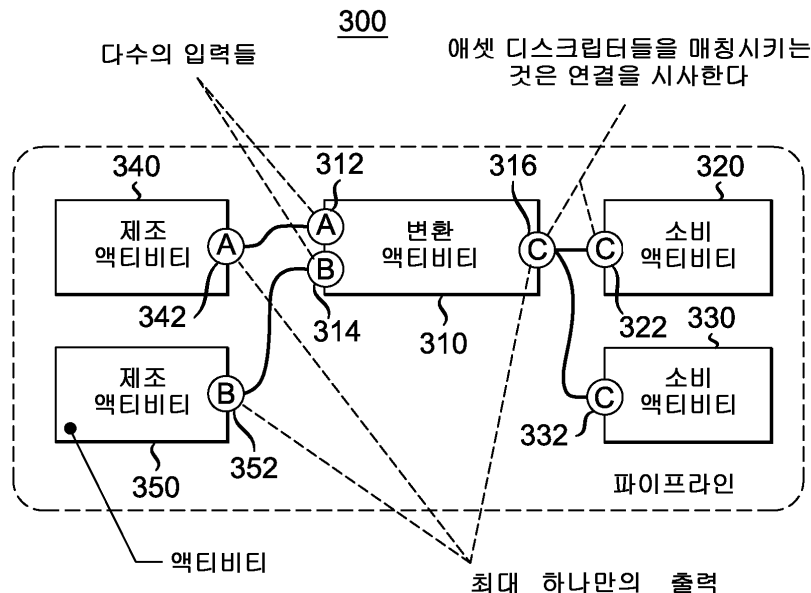
(54) 발명의 명칭 애셋 구동 워크플로우 모델링에서의 세트 핸들링

(57) 요약

애셋 구동 워크플로우 종속성 관리는 각각의 액티비티에 대한 애셋들의 디스크립션들에 기초하여 액티비티들 간의 연결들을 확립한다. 이들 기술적 "계약들"은 원하는 출력을 생성하는데 필수적인 관련 액티비티들을 쉽게 매칭시키는 메카니즘을 제공한다. 원하는 워크플로우의 그래픽 모델을 생성함으로써, 사용자는 워크플로

(뒷면에 계속)

대표도 - 도3



우에 무엇이 수반되는지 뿐만 아니라 문제들 및 중복성들이 어디서 발생하는지의 이해를 제공받는다. 그래픽 모델은 실제 작업 제품들을 설계하고 추적하는데 이용될 수 있다. 액티비티들의 그래픽적 표현들은 벤더들, 설비들, 및 다른 제품 액티비티들을 모델링하는데 이용된다. 액티비티 모델들은 액티비티들 간에 전달되는 상품 (deliverable) 들을 표현하는 애셋들을 제조하고/하거나 소비한다. 액티비티들의 모델들을 이용하여, 제조 파이프라인의 모델이 뒤에서부터 앞으로 구축될 수 있다. 따라서, 최종 결과 액티비티 모델이 먼저 선택되고, 선택된 최종 결과 액티비티에 의해 요구되는 애셋들에 기초하여, 그 요구되는 애셋들을 제조하는 적절한 액티비티가 선택될 수 있다. 이 프로세스는 프로세스 파이프라인의 시작에 도달할 때까지 반복될 수 있다.

그 후, 실세계 프로세스 파이프라인은 모델에 기초하여 형성될 수 있고, 모델은 실세계 제조 파이프라인의 상태를 추적하는데 이용될 수 있다. 이러한 기법들은 보다 효율적인 추적을 위해 애셋들의 세트들의 핸들링을 추가로 지원할 수 있다.

명세서

청구범위

청구항 1

워크플로우 모델에서의 세트를 핸들링하는 방법으로서,

동일한 유형의 다수의 애셋들을 포함하는 제 1 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 제 1 입력을 갖는 제 1 액티비티의 그래픽 표현을 제공하는 단계;

동일한 유형의 다수의 애셋들을 포함하는 제 2 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 2 액티비티의 그래픽 표현을 제공하는 단계로서, 상기 제 2 세트의 멤버들은 상기 제 1 액티비티의 그래픽 표현의 상기 제 1 입력과 연관된 상기 제 1 세트의 멤버들과 매칭하는, 상기 제 2 액티비티의 그래픽 표현을 제공하는 단계; 및

상기 제 1 액티비티의 그래픽 표현의 상기 제 1 입력과 연관된 상기 제 1 세트의 멤버들과 매칭하는, 상기 제 2 액티비티의 그래픽 표현의 상기 출력과 연관된 상기 제 2 세트의 멤버들에 기초하여, 상기 제 1 액티비티의 그래픽 표현의 제 1 입력을 상기 제 2 액티비티의 그래픽 표현의 상기 출력과 연결하는 단계를 포함하는, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 2

제 1 항에 있어서,

그래픽 표현들을 제공하는 단계들은 다수의 가능한 그래픽 표현들로부터 그래픽 표현을 선택하는 단계를 포함하는, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 액티비티의 그래픽 표현은 상기 제 1 입력의 애셋 디스크립터와는 상이한 애셋 디스크립터를 가진 제 2 입력을 갖는, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 4

제 3 항에 있어서,

상기 제 1 액티비티의 그래픽 표현의 상기 제 2 입력의 애셋 디스크립터와 매칭하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 3 액티비티의 그래픽 표현을 제공하는 단계; 및

매칭하는 애셋 디스크립터들에 기초하여 상기 제 1 액티비티의 그래픽 표현의 상기 제 2 입력을 상기 제 3 액티비티의 그래픽 표현의 상기 출력과 연결하는 단계를 더 포함하는, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 5

제 1 항에 있어서,

상기 제 2 액티비티의 그래픽 표현은 연관된 애셋 디스크립터를 가진 적어도 제 1 입력을 더 포함하고,

상기 방법은,

상기 제 2 액티비티의 그래픽 표현의 상기 제 1 입력의 애셋 디스크립터와 매칭하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 3 액티비티의 그래픽 표현을 제공하는 단계; 및

매칭하는 애셋 디스크립터들에 기초하여 상기 제 2 액티비티의 그래픽 표현의 상기 제 1 입력을 상기 제 3 액티비티의 그래픽 표현의 상기 출력과 연결하는 단계를 더 포함하는, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 6

제 1 항에 있어서,

액티비티의 그래픽 표현들 중 적어도 하나는 액티비티 템플릿인, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 7

제 1 항에 있어서,

액티비티의 그래픽 표현들 중 적어도 하나는 액티비티 인스턴스인, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 8

제 7 항에 있어서,

액티비티 인스턴스의 애셋 디스크립터들의 특정된 파라미터들은 상기 액티비티 인스턴스에 연결된 임의의 템플릿 액티비티들로 패스되는, 워크플로우 모델에서의 세트를 핸들링하는 방법.

청구항 9

워크플로우 모델에서의 세트를 핸들링하는 장치로서,

워크플로우 정보를 저장하는 저장부;

프로세싱을 위하여 데이터를 저장하는 메모리;

프로세서를 포함하고,

상기 프로세서는,

동일한 유형의 다수의 애셋들을 포함하는 제 1 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 제 1 입력을 갖는 제 1 액티비티의 그래픽 표현을 제공하고;

동일한 유형의 다수의 애셋들을 포함하는 제 2 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 2 액티비티의 그래픽 표현을 제공하는 것으로서, 상기 제 2 세트의 멤버들은 상기 제 1 액티비티의 그래픽 표현의 상기 제 1 입력과 연관된 상기 제 1 세트의 멤버들과 매칭하는, 상기 제 2 액티비티의 그래픽 표현을 제공하고; 그리고

상기 제 1 액티비티의 그래픽 표현의 상기 제 1 입력과 연관된 상기 제 1 세트의 멤버들과 매칭하는, 상기 제 2 액티비티의 그래픽 표현의 상기 출력과 연관된 상기 제 2 세트의 멤버들에 기초하여, 상기 제 1 액티비티의 그래픽 표현의 제 1 입력을 상기 제 2 액티비티의 그래픽 표현의 상기 출력과 연결하도록 구성되는, 워크플로우 모델에서의 세트를 핸들링하는 장치.

청구항 10

제 9 항에 있어서,

네트워크에 접속하기 위한 네트워크 접속부를 더 포함하는, 워크플로우 모델에서의 세트를 핸들링하는 장치.

청구항 11

제 9 항에 있어서,

제공된 그래픽 표현들은 다수의 가능한 그래픽 표현들로부터 선택된 그래픽 표현을 포함하는, 워크플로우 모델에서의 세트를 핸들링하는 장치.

청구항 12

제 9 항에 있어서,

액티비티의 그래픽 표현들 중 적어도 하나는 액티비티 템플릿인, 워크플로우 모델에서의 세트를 핸들링하는 장

치.

청구항 13

제 9 항에 있어서,

액티비티의 그래픽 표현들 중 적어도 하나는 액티비티 인스턴스인, 워크플로우 모델에서의 세트를 핸들링하는 장치.

청구항 14

제 13 항에 있어서,

액티비티 인스턴스의 애셋 디스크립터들의 특정된 파라미터들은 상기 액티비티 인스턴스에 연결된 임의의 템플릿 액티비티들로 패스되는, 워크플로우 모델에서의 세트를 핸들링하는 장치.

청구항 15

명령들을 포함하는 머신 판독가능 매체로서,

상기 명령들은 실행될 때,

동일한 유형의 다수의 애셋들을 포함하는 제 1 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 제 1 입력을 갖는 제 1 액티비티의 그래픽 표현을 제공하는 단계;

동일한 유형의 다수의 애셋들을 포함하는 제 2 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 2 액티비티의 그래픽 표현을 제공하는 단계로서, 상기 제 2 세트의 멤버들은 상기 제 1 액티비티의 그래픽 표현의 상기 제 1 입력과 연관된 상기 제 1 세트의 멤버들과 매칭하는, 상기 제 2 액티비티의 그래픽 표현을 제공하는 단계; 및

상기 제 1 액티비티의 그래픽 표현의 상기 제 1 입력과 연관된 상기 제 1 세트의 멤버들과 매칭하는, 상기 제 2 액티비티의 그래픽 표현의 상기 출력과 연관된 상기 제 2 세트의 멤버들에 기초하여, 상기 제 1 액티비티의 그래픽 표현의 제 1 입력을 상기 제 2 액티비티의 그래픽 표현의 상기 출력과 연결하는 단계를 수행하는, 명령들을 포함하는 머신 판독가능 매체.

발명의 설명

기술 분야

[0001]

관련 출원들에 대한 참조

[0002]

본 출원은 2013년 1월 23일 출원된 미국 가출원 일련 번호 제 61/755,892 호 및 2013년 6월 11일에 출원된 미국 가출원 일련 번호 제 61/833,770 호의 이익을 주장하며, 여기서는 그 전체 내용을 참조로서 포함한다.

[0003]

본 출원은 또한 명칭이 "ASSET-DRIVEN WORKFLOW MODELING", "FULFILLMENT TRACKING IN ASSET-DRIVEN WORKFLOW MODELING", 및 "METHOD AND APPARATUS FOR MAPPING PROCESS INFORMATION ONTO ASSET DATA" 이고 동시에 제출된 출원들에 관련되며, 여기서는 그 전체 내용을 참조로서 포함한다.

[0004]

기술 분야

[0005]

본 발명은 워크플로우 및/또는 제조 파이프라인들을 모델링하는 것에 관련되며, 보다 구체적으로, 제조 파이프라인을 따라 요구되어 제조되는 애셋들에 기초하여 제조 파이프라인을 모델링하는 방법 및 장치에 관련된다.

배경 기술

[0006]

임의의 영화, 텔레비전, 또는 기록 재생에 대하여, 새로운 애셋들을 새로운 포맷들로 생성할 수도 있는 상이한 제조사들 간에 전달되는 수개의 상이한 포맷들로 되어 있는 수개의 애셋들, 이를 테면, 샷들, 비주얼 효과들, 사운드 등이 존재한다. 이들 애셋들은 텔레비전 쇼, 영화, 레코드 등을 제작하는데 이용되는 구축 블록들이다. 이러한 애셋들의 트랙을 유지하고 애셋들의 위치 및 상태가 무엇인지를 아는 것은 대단히 복잡한 노력

이 든다.

[0007] 프로젝트 매니저 프로그램들이 존재하기는 하지만, 이들은 통상적으로 탑 다운 상태 시스템 설계를 위해 설계되며, 여기에서 각각의 블록은 시스템을 생성하기 위해 사용자에게 의해 프로젝트의 스테이지가 다음 블록에 연관되는 것을 나타낸다. 블록들은 애셋들에 기초하여 함께 링크하지도 않고 프로그램이 이러한 애셋들을 추적하지도 않는다.

[0008] 따라서, 영화, 텔레비전, 음악 앨범 등에 이용된 애셋들에 기초하여 워크플로우 및/또는 제조 파이프라인을 모델링할 수 있는 방법 및 시스템이 요구된다.

발명의 내용

과제의 해결 수단

[0009] 애셋 구동 워크플로우 종속성 관리는 각각의 액티비티에 대한 입력들 및/또는 출력들로서 이용되는 애셋들의 디스크립션들에 기초하여 액티비티들 간의 연결들을 확립한다. 이들 기술적 "계약들"은 원하는 출력을 생성하는데 필수적인 관련 액티비티들을 쉽게 매칭시키는 메카니즘을 제공한다. 원하는 워크플로우의 그래픽 모델을 생성함으로써, 사용자는 워크플로우에 무엇이 수반되는지 뿐만 아니라 문제들 및 중복성들이 어디서 발생하는지의 이해를 제공받는다. 그래픽 모델은 실세계 제품들을 설계하고 추적하는데 이용될 수 있다.

[0010] 액티비티들의 그래픽적 표현들은 벤더들, 설비들, 및 다른 제품 액티비티들을 모델링하는데 이용된다. 액티비티 모델들은 액티비티들 간에 전달되는 상품 (deliverable) 들을 표현하는 애셋들을 제조하고/하거나 소비한다. 액티비티들의 모델들을 이용하여, 제조 파이프라인의 모델이 뒤에서부터 앞으로 구축될 수 있다. 따라서, 최종 결과 액티비티 모델이 먼저 선택되고, 선택된 최종 결과 액티비티에 의해 요구되는 애셋들에 기초하여, 그 요구되는 애셋들을 제조하는 적절한 액티비티가 선택될 수 있다. 이 프로세스는 프로세스 파이프라인의 시작에 도달할 때까지 반복될 수 있다. 그 후, 실세계 프로세스 파이프라인이 모델에 기초하여 형성될 수 있고, 모델은 실세계 제조 파이프라인의 상태를 추적하는데 이용될 수 있다. 이러한 기법들은 보다 효율적인 추적을 위한 애셋들의 세트들의 핸들링을 추가로 지원할 수 있다.

[0011] 본 개시물의 일 실시형태는 워크플로우 모델에 있어서의 세트들을 핸들링하는 방법을 제공한다. 본 방법은 동일한 유형의 다수의 애셋들을 포함하는 제 1 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 제 1 입력을 갖는 제 1 액티비티의 그래픽 표현을 제공하는 단계; 동일한 유형의 다수의 애셋들을 포함하는 제 2 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 2 액티비티의 그래픽 표현을 제공하는 단계로서, 제 2 세트의 멤버들은 제 1 액티비티의 그래픽 표현의 제 1 입력과 연관된 제 1 세트의 멤버들과 매칭하는, 제 2 액티비티의 그래픽 표현을 제공하는 단계; 및 제 1 액티비티의 그래픽 표현의 입력과 연관된 제 1 세트에서의 멤버들과 매칭하는, 제 2 액티비티의 그래픽 표현의 출력과 연관된 제 2 세트의 멤버들에 기초하여 제 1 액티비티의 그래픽 표현의 제 1 입력을 제 2 액티비티의 그래픽 표현의 출력과 연관시키는 단계를 포함한다.

[0012] 본 개시물의 다른 실시형태는 워크플로우 모델에 있어서의 세트들을 핸들링하는 장치를 제공한다. 본 장치는 저장부, 메모리 및 프로세서를 포함한다. 저장부 및 메모리는 데이터를 저장하기 위한 것이다. 프로세서는, 동일한 유형의 다수의 애셋들을 포함하는 제 1 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 제 1 입력을 갖는 제 1 액티비티의 그래픽 표현을 제공하고; 동일한 유형의 다수의 애셋들을 포함하는 제 2 세트를 표시하는 연관된 애셋 디스크립터를 가진 적어도 출력을 갖는 제 2 액티비티의 그래픽 표현을 제공하는 것으로서, 제 2 세트의 멤버들은 제 1 액티비티의 그래픽 표현의 제 1 입력과 연관된 제 1 세트의 멤버들과 매칭하는, 제 2 액티비티의 그래픽 표현을 제공하고; 그리고 제 1 액티비티의 그래픽 표현의 입력과 연관된 제 1 세트에서의 멤버들과 매칭하는, 제 2 액티비티의 그래픽 표현의 출력과 연관된 제 2 세트의 멤버들에 기초하여 제 1 액티비티의 그래픽 표현의 제 1 입력을 제 2 액티비티의 그래픽 표현의 출력과 연관시키도록 구성된다.

[0013] 목적들 및 이점들은 청구항들 내에 구체적으로 적시된 요소들과 결합들에 의해 실현되고 얻어질 것이다. 개시된 실시형태들은 여기에서의 진보적인 교시들의 다수의 유리한 사용들의 단지 예들에 불과함을 주지하는 것이 중요하다. 상술한 포괄적 설명 및 다음의 상세한 설명 양쪽 모두는 예시적이고 설명적인 것이며 본 발명을 청구하는 바 대로 제한하려는 것이 아님을 이해하여야 한다. 또한, 몇몇 설명들은 본 발명의 진보적 피쳐들에 적용될 수도 있고 그 외의 것에는 적용되지 않을 수도 있다. 일반적으로, 달리 설명되지 않는 한, 일반화의 손실 없이, 단일의 엘리먼트들은 복수일 수도 있고 그 역으로 복수의 엘리먼트들은 단수일 수도 있다. 도면들에서, 유사한 도면부호들은 수개의 도면들 전반에 걸쳐 유사한 부분들을 지칭한다.

도면의 간단한 설명

[0014]

- 도 1 은 애셋 구동 워크플로우 모델링이 일 실시형태에 따라 구현될 수 있는 시스템의 블록 개략도를 나타낸다.
- 도 2 는 일 실시형태에 따라 방법론적 애셋 구동 워크플로우 모델링을 구현하는 전자 디바이스의 블록 개략도를 나타낸다.
- 도 3 은 일 실시형태에 따라 애셋 구동 워크플로우 모델의 블록 개략도를 나타낸다.
- 도 4 는 일 실시형태에 따라 애셋 구동 워크플로우 모델링을 위한 방법론의 예시적인 플로우차트를 나타낸다.
- 도 5 는 일 실시형태에 따라 도 4 의 플로우차트의 단계들을 예시하는 예시도를 나타낸다.
- 도 6 은 일 실시형태에 따라 애셋 구동 워크플로우 모델 구현 세트들의 블록 개략도를 나타낸다.
- 도 7 은 일 실시형태에 따라 액티비트들의 그래픽 표현들의 예시도를 나타낸다.
- 도 8 은 일 실시형태에 따라 애셋 디스크립터들에 기초한 액티비트들의 매칭의 예시도를 나타낸다.
- 도 9 는 일 실시형태에 따라 애셋 디스크립터들에 기초한 액티비티 템플릿들과 액티비티 인스턴스들의 매칭의 예시도를 나타낸다.
- 도 10 은 일 실시형태에 따라 파라미터들에 기초한 예시적인 애셋 디스크립터들 및 이들의 매칭의 테이블을 나타낸다.
- 도 11a 및 도 11b 는 일 실시형태에 따라 애셋 디스크립터들의 파라미터들의 전파 예시도를 나타낸다.
- 도 12 는 일 실시형태에 따라 애셋 구동 워크플로우 모델링에 있어서 애셋들의 상태를 제공하는 방법론의 예시적 플로우차트를 나타낸다.
- 도 13 은 일 실시형태에 따라 도 12 의 플로우차트의 단계들을 예시하는 예시도를 나타낸다.
- 도 14 는 일 실시형태에 따라 공유된 설비들을 수반하는 애셋 추적의 예시도를 나타낸다.
- 도 15 는 일 실시형태에 따라 애셋들, 액티비티들, 벤더들, 및 설비들 간의 관계를 예시하는 예시도를 나타낸다.
- 도 16 은 일 실시형태에 따라 애셋 데이터 상에 프로세스 정보를 매칭시키는 방법론의 예시적인 플로우차트를 나타낸다.
- 도 17 은 일 실시형태에 따라 도 16 의 플로우차트의 단계들을 예시하는 예시도를 나타낸다.
- 도 18 은 일 실시형태에 따라 프로듀서 워크스페이스의 예시적인 스크린샷을 나타낸다.
- 도 19 는 일 실시형태에 따라 도 18 의 프로듀서 워크스페이스의 상품 대시보드의 분리된 스크린샷을 나타낸다.
- 도 20 은 일 실시형태에 따라 도 18 의 프로듀서 워크스페이스의 필터링된 파이프라인의 분리된 스크린샷을 나타낸다.
- 도 21 는 일 실시형태에 따라 도 18 의 프로듀서 워크스페이스의 액티비티 디테일들의 분리된 스크린샷을 나타낸다.
- 도 22 는 일 실시형태에 따라 매니저 워크스페이스의 예시적인 스크린샷을 나타낸다.
- 도 23 은 일 실시형태에 따라 데이터 I/O 워크스페이스의 예시적인 스크린샷을 나타낸다.
- 도 24 는 일 실시형태에 따라 실행 워크스페이스의 예시적인 스크린샷을 나타낸다.
- 도 25 는 일 실시형태에 따라 파이프라인 빌더의 예시적인 스크린샷을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0015]

이하 도 1 로 가면, 애셋 구동 워크플로우 모델링을 구현하는 시스템 (100) 의 일 실시형태의 블록도가 제공된다. 시스템은 서버 (110), 및 인터넷 (150) 을 통하여 서버 (110) 와 통신하는, 하나 이상의 전자 디바이스들, 이를 테면, 스마트 폰들 (120); 퍼스널 컴퓨터들 (PCs)(130), 이를 테면, 데스크톱들 또는 랩톱들; 및 태블릿들 (140) 을 포함한다. 특정 실시형태들에서, 서버 (110) 는 애셋 구동 워크플로우 모델링을 위하여 프로

세싱 및 저장을 포함한 환경을 제공한다. 사용자들은 전자 디바이스들, 이를 테면, 스마트 폰들 (120), PCs (130), 또는 태블릿들 (140) 상에서의 브라우저 또는 애플리케이션을 이용하여 서버 (110) 상에서의 애셋 구동 워크플로우 모델과 인터페이스한다. 다른 실시형태들에서, 애셋 구동 워크플로우 모델링의 일부 또는 전부는 하나 이상의 전자 디바이스들, 이를 테면, 스마트 폰들 (120); 퍼스널 컴퓨터들 (PCs)(130), 이를 테면, 데스크톱들, 또는 랩톱들; 및 태블릿들 (140) 상에서 수행될 수 있다.

[0016] 도 2 는 애셋 구동 워크플로우 모델링을 위한 방법론 및 시스템을 구현하는데 이용될 수 있는 예시적인 서버 (200) 또는 전자 디바이스를 나타낸다. 서버 또는 전자 디바이스는 하나 이상의 프로세서들 (210), 메모리 (220), 저장부 (230) 및 네트워크 인터페이스 (240) 를 포함한다. 이들 엘리먼트들 각각은 아래 보다 자세하게 설명될 것이다.

[0017] 프로세서 (210) 는 서버 (200) 또는 전자 디바이스의 동작을 제어한다. 프로세서 (210) 는 애셋 구동 워크플로우 모델링 애플리케이션의 기능을 제공할 뿐만 아니라 서버 또는 전자 디바이스를 동작시키는 소프트웨어를 실행시킨다. 프로세서 (210) 는 메모리 (220), 저장부 (230) 및 네트워크 인터페이스 (240) 에 접속되고, 이들 엘리먼트들 간의 정보의 전달 및 프로세싱을 핸들링한다. 프로세서 (210) 는 특수 기능에 전용되는 프로세서 또는 범용 프로세서일 수도 있다. 특정 실시형태들에서, 다수의 프로세서들이 있을 수 있다.

[0018] 메모리 (220) 에는 프로세서에 의해 실행되는 데이터 및 명령들이 저장된다. 메모리 (220) 는 휘발성 메모리 (RAM), 비휘발성 메모리 (EEPROM), 또는 다른 적절한 매체를 포함할 수 있다.

[0019] 저장부 (230) 에는, 본 개시물의 콜드 저장 권장 방법론을 실행하는데 있어 프로세서에 의해 이용되고 생성되는 데이터가 저장된다. 저장부는 자기 매체 (하드 드라이브), 광학 매체 (CD/DVD-Rom), 또는 플래시 기반 저장부일 수도 있다. 다른 유형들의 적절한 저장부도 이 개시물의 이익을 받는 당해 기술 분야의 당업자에게 명백할 것이다.

[0020] 네트워크 인터페이스 (240) 는 서버 (200) 또는 전자 디바이스의, 네트워크를 통한 다른 디바이스들과의 통신을 핸들링한다. 적절한 네트워크들의 예들은 이더넷 네트워크들, Wi-Fi 실행가능 네트워크들, 셀룰라 네트워크들 등을 포함한다. 다른 유형들의 적절한 네트워크들도 이 개시물의 이익을 받는 당해 기술 분야의 당업자에게 명백할 것이다.

[0021] 도 2 에 서술된 엘리먼트들은 예시적인 것임을 이해하여야 한다. 서버 (200) 또는 다른 전자 디바이스는 임의의 수의 엘리먼트들을 포함할 수 있고, 특정 엘리먼트들은 다른 엘리먼트들의 기능의 일부 또는 전부를 제공할 수 있다. 다른 가능한 구현예들도 이 개시물의 이익을 받는 당해 기술 분야의 당업자에게 명백할 것이다.

[0022] 도 3 은 애셋 구동 워크플로우의 그래픽 모델 (300) 을 나타낸다. 애셋 구동 워크플로우 종속성 관리는 각각의 액티비티에 대한 입력들 및/또는 출력들로서 이용되는 애셋들의 디스크립션들에 기초하여 액티비티들 간의 연결들을 확립한다. 이들 디스크립터들은 원하는 출력을 생성하는데 필요한 관련 액티비티들을 용이하게 매칭시키는 메카니즘을 제공한다. 도 3 은 변환 액티비티 (310) 가 애셋 A 및 애셋 B 를 입력들 (312, 314) 로서 요구하고 애셋 C 를 출력 (316) 상에 제공된 결과로서 생성하는 것을 도시한다. 소비 액티비티들 (320, 330) 은 입력들 (322, 332) 에서 애셋 C 를 예상하는 한편, 제조 액티비티들 (340, 350) 은 애셋들 A 및 B 를 출력들 (342, 352) 에서 모델링 시스템으로 가져온다. 이 예시적인 파이프라인 (300) 에서, 제조 액티비티들 (340 및 350) 의 출력들 (342, 352) 은 변환 액티비티들 (310) 의 입력들 (312, 314) 에 연결된다. 이어서, 변환 액티비티 (310) 의 출력 (316) 은 소비 액티비티들 (320 및 330) 의 입력들 (322, 332) 에 연결된다.

[0023] **일부 작업 정의들:**

[0024] **파이프라인:** 원하는 출력을 생성하기 위해 함께 연결된 액티비티들의 집합. 파이프라인은 워크플로우의 그래픽 모델을 제공한다. 비디오 또는 필름 제조의 예에서, 파이프라인은 원하는 제품을 제조하는데 필수적인 모든 액티비티들 (이를 테면, 데이터, 특수 샷들, 포맷들, 또는 오디오 트랙들의 생성) 을 표현한다.

[0025] **액티비티:** (상품들, 이를 테면 데이터, 특수 샷들, 포맷들, 또는 오디오 트랙들을 포함하는) 애셋들을 제조, 변환, 또는 소비하는 동작. 각각의 액티비티는 입력들, 출력들 또는 양쪽 모두를 가질 수도 있다. 간단한 가정으로서, (출력이 복잡한 또는 복합의 애셋일 수 있지만) 액티비티들은 통상적으로 단일의 출력만을 갖는다. (소비 액티비티들을 제외한) 액티비티들은 이들의 출력에 의해 쉽게 특정화될 수 있다. 액티비티를 고유하게 하는 것은 주어진 출력을 생성하기 위해 액티비티를 통하여 매핑된 입력들의 특수 구성이다. 상이한

액티비티들은 동일한 출력을 생성할 수도 있고, 이에 따라 오직 하나만이 주어진 파이프라인 내에서 요구된다. 주어진 액티비티의 출력은 다수의 다운스트림 액티비티들 내에 입력을 제공할 수 있다.

[0026] **연결:** 액티비티 출력 디스크립션이 하나 이상의 입력 디스크립션들과 매칭할 때 연결이 시사된다. 연결들은 애셋의 전달 및 수신을 위한 이행 동의들 또는 계약들을 표현한다.

[0027] **애셋 디스크립터:** 액티비티들 간의 연결들을 매칭시키고 확립하는데 이용된 액티비티의 입력/출력의 라벨.

[0028] 이들 개념들의 추가의 설명은 이 문서의 아래 제공된다.

[0029] 도 4 는 워크플로우의 그래픽 표현을 생성하는 프로세스의 플로우도 (400) 이다. 기본적으로, 프로세스는 3 개의 단계들을 수반한다. 연관된 애셋 디스크립터를 갖는 적어도 하나의 입력을 가진 제 1 액티비티의 그래픽 표현을 제공하는 것 (단계 410), 제 1 액티비티의 그래픽 표현의 입력의 애셋 디스크립터와 매칭하는 연관된 애셋 디스크립터를 갖는 적어도 하나의 출력을 가진 제 2 액티비티의 그래픽 표현을 제공하는 것 (단계 420), 및 매칭한 애셋 디스크립터들에 기초하여 제 2 액티비티의 그래픽 표현의 출력을 제 1 액티비티의 그래픽 표현의 입력과 그래픽적으로 연결하는 것 (단계 430). 이들 단계들의 그래픽 예 (500) 는 도 5 에서 볼 수 있다.

[0030] **워크플로우의 모델링**

[0031] 도 5 의 그래픽 예 (500) 에 도시된 바와 같이, 단계 (410) 는 제 1 액티비티 (510) 의 그래픽 표현을 제공하는 것으로 시작한다. 이 실시형태에서는, 제 1 액티비티의 그래픽 표현은 원하는 애셋의 디스크립터 (이 경우에 "H") 를 갖는 하나의 입력 (512) 을 갖는다. 다른 실시형태들에서는, 제 1 액티비티 (510) 의 그래픽 표현은 상이한 연관된 애셋 디스크립터들을 갖는 다수의 입력들을 가질 수도 있다. 제 1 액티비티의 제공된 그래픽 표현은 다수의 제공된 액티비티들의 그래픽 표현들로부터 선택된 한 그래픽 표현일 수도 있다. 이 그래픽 표현의 선택은 원하는거나 또는 요구되는 액티비티에 기초하여 시스템 자체에 의해 또는 그래픽 사용자 인터페이스를 이용하여 사용자에게 의해 행해질 수 있다. 일부 환경에서는, 특정 애셋 디스크립터와 매칭할 수 있는 액티비티의 모두를 이용하는 것은 아니다.

[0032] 도 5 의 그래픽 예 (500) 의 단계 (420) 에서, 제 2 액티비티의 적어도 하나의 그래픽 표현이 제공된다. 이 예에서, 시스템은 제 1 액티비티 (510) 의 입력 (512) 의 애셋 디스크립터 ("H") 와 매칭하는 연관된 애셋 디스크립터를 갖는 출력을 가진 액티비티들을 검색한다. 원하는 액티비티를 출력하는 하나 보다 많은 액티비티가 존재할 수 있지만, 오직 하나만이 선택될 필요가 있다. 선택은 사용자에게 의해 또는 시스템에 의해 수행될 수 있고, 이 예에서, 제 1 액티비티 (510) 의 입력 (512) 의 애셋 디스크립터 ("H") 와 매칭하는 연관된 애셋 디스크립터를 갖는 출력을 가진 두개의 가능한 액티비티들 (520, 530) 이 존재한다. 하나의 가능한 제 2 액티비티 (520) 는 상이한 연관된 애셋 디스크립터 ("A") 를 갖는 입력 (522) 뿐만 아니라 매칭하는 애셋 디스크립터 ("IF") 를 갖는 출력 (524) 을 갖는다. 매칭하는 애셋 디스크립터 ("H") 를 갖는 출력 (536) 을 가진 다른 가능한 제 2 액티비티 (530) 는 상이한 연관된 애셋 디스크립터들 ("D" 및 "E") 을 갖는 두개의 입력들 (532, 534) 을 갖는다.

[0033] 제 2 액티비티 (520) 의 출력과 연관된 애셋 디스크립터가 제 1 액티비티 (510) 의 입력 (512) 의 애셋 디스크립터와 매칭하기 때문에 파이프라인으로의 원하는 제 2 액티비티 (이 경우에 액티비티 (520)) 를 선택하는 것은, 액티비티들 (510, 520) 간의 연결을 시사한다. 시사된 연결은 단계 (430) 에서 그래픽 연결 (540) 로서 표현된다.

[0034] 이 실시형태에서, 매칭 및 연결은 애셋 자체가 아닌, 애셋 디스크립터에 기초한다. 이는 실제 애셋이 존재하기 전에 완벽한 파이프라인 모델의 생성을 허용한다. 이러한 애셋 구동 모델링의 일부 이점들은 액티비티들의 명시적 매핑을 포함한다. 이들이 출력 또는 소비하는 애셋들의 디스크립션들에 기초하여, 애셋들의 출처 (provenance) 가 시스템을 통하여 명시적으로 추적될 수 있고 다운스트림 종속값들이 쉽게 계산될 수 있다.

[0035] **워크플로우/파이프라인 모델링 (세트들)**

[0036] 미디어 제조의 세계에서, 보다 큰 세트의 부분으로서 복수의 동일한 엘리먼트들을 제조하는 액티비티들에 직면하는 것이 일반적이다. 예를 들어, "데일리즈" 액티비티는 온셋 캡처된 비디오 및 오디오 "샷들" 을 디렉터 또는 제작자가 리뷰 및 승인하는 쉽게 리뷰가능한 포맷으로 변환하는 것을 담당한다.

[0037] 이 예에서, 데일리즈 액티비티는 수 주일의 기간에 걸쳐 1000 개의 "샷들" 을 프로세싱하는 것을 담당한다. 또한, 이들 "샷들"은 비순차적 방식으로 상이한 카메라 유닛들로부터 유래할 수도 있다. "데일리즈" 액티

비티의 출력은 하루 기반으로 다음 단계까지 규칙적으로 패스된다.

[0038] 이러한 시스템의 모델링을 위하여, 세트들의 이용이 유리할 수도 있다. 세트는 동일한 유형으로 된 하나 이상의 애셋들의 집합들이다. 세트의 각각의 멤버는 고유한 애셋이지만, 세트 내의 다른 애셋들과 동일한 유형 또는 클래스로서 이루어진다. 예를 들어, 세트는 500 개의 샷들로 구성될 수도 있지만, 세트의 각각의 멤버는 한 샷이다. 세트들은 다수의 액티비티들에 걸쳐 작업 제품을 분배하고 추적하는데 이용될 수 있다. 세트들은 또한 서브세트들로 분할될 수 있다. 따라서, 액티비티는 상이한 액티비티들로부터 상이한 서브세트들을 수신할 수도 있거나 또는 액티비티는 제조된 오리지널의 일부를 소비만 하는 것일 수도 있다.

[0039] 세트들을 이용하여 워크플로우를 모델링하는 방법론은 도 4 에 설명된 바와 같은 비세트 애셋 구동 모델링에 대한 방법론과 유사하다. 제 1 및 제 2 액티비티들은 연관된 애셋 디스크립터들에 기초하여 제공되고 연결된다. 그러나 이경우에, 애셋 디스크립터들은 애셋들의 세트들이 이용되고 있음을 표시한다. 이것의 일 예는 도 6 의 워크플로우 모델 (600) 에서 볼 수 있다.

[0040] 도 6 의 워크플로우 모델 (600) 에서, 제 1 액티비티 (610) 의 그래픽 표현이 제공된다. 제 1 액티비티 (610) 는 소비 액티비티이며, 연관된 애셋 디스크립터 (이 경우에 "D") 를 갖는 입력 (612) 을 갖는다. 이 예에서, 애셋 디스크립터는 입력 (612) 에서 수신될 것으로 예상되는 애셋들의 세트 (이 경우에 샷들 (1-25)) 가 존재함을 추가로 표시한다. 제 2 액티비티 (620) 의 그래픽 표현이 또한 제공된다. 제 2 액티비티 (620) 는 변환 액티비티이며, 매칭하는 연관된 애셋 디스크립터 ("D") 를 갖는 출력 (622) 을 갖는다. 그러나, 이 경우에, 애셋 디스크립터는 출력 (622) 상에 제공될 보다 큰 애셋들의 세트 (샷들 (1-100)) 가 존재함을 표시한다. 그러나, 각각의 세트의 멤버 애셋들의 일부가 매칭하기 때문에, 연결이 시사되어 그래픽적으로 표시된다 (670).

[0041] 도 6 의 실시형태에서, 제 3 (630) 및 제 4 (640) 액티비티들의 그래픽 표현들이 또한 제공된다. 제 3 및 제 4 액티비티들은 액티비티들의 세트들을 수신할 것임을 추가로 표시하는 연관 애셋 디스크립터들 ("D") 을 가진 입력들 (632, 642) 을 갖는 소비 액티비티들이다. 제 3 액티비티 (630) 의 경우에, 세트는 샷들 (26-75) 을 포함한다. 제 4 액티비티 (640) 의 경우에, 세트는 샷들 (76-100) 을 포함한다. 제 3 및 제 4 액티비티 (630, 640) 의 세트들이 제 2 액티비티 (620) 의 세트의 서브세트들이기 때문에, 각각의 세트들의 매칭하는 멤버들이 존재하고, 제 2 액티비티 (620) 와 제 3 액티비티 (630) 간에 뿐만 아니라 제 2 액티비티 (620) 와 제 4 액티비티 (640) 간에 연결이 시사되어 각각 그래픽적으로 672, 674 로 표시된다.

[0042] 이전에 설명된 바와 같이, 도 6 의 모델 (600) 에서의 제 2 액티비티 (620) 는 변환 액티비티이다. 이에 의해, 제 2 액티비티 (620) 는 연관된 애셋 디스크립터 (이 경우에 "S") 를 갖는 입력 (624) 을 더 포함한다. 이 예에서, 연관된 애셋 디스크립터는 입력 (624) 상에서 수신될 것으로 예상되는 애셋들의 세트 (이 경우에, 샷들 (1-100)) 가 존재함을 추가로 표시한다. 따라서, 제 2 액티비티 (620) 는 그 입력 (624) 상에서 샷들 (1-100) 을 포함하는 애셋들의 세트 ("S") 를 수신하고 그 출력 (622) 상에서 샷들 (1-100) 을 포함하는 애셋들의 세트 ("D") 를 생성하는 프로세스 또는 오퍼레이터를 모델링한다.

[0043] 제 5 액티비티 (650) 및 제 6 액티비티 (660) 의 그래픽 표현들은 도 6 의 모델 (600) 에 또한 제공된다. 제 5 액티비티 (650) 및 제 6 액티비티 (660) 는 입력들 (652, 662) 이 액티비티들의 세트들을 생성할 것임을 추가로 표시하는 연관된 애셋 디스크립터들 ("S") 을 갖는 출력들 (652, 662) 을 갖는 액티비티들을 생성하고 있다. 제 5 액티비티 (650) 의 경우에, 세트는 샷들 (1-50) 을 포함한다. 제 6 액티비티 (660) 의 경우에, 세트는 샷들 (50-100) 을 포함한다. 제 5 및 제 6 액티비티 (650, 660) 의 세트들이 제 2 액티비티 (620) 의 입력 (624) 상에서 수신될 세트의 서브세트들이기 때문에, 각각의 세트들의 매칭하는 멤버들이 존재하고, 제 2 액티비티 (620) 와 제 5 액티비티 (650) 간에 뿐만 아니라 제 2 액티비티 (620) 와 제 6 액티비티 (660) 간에 연결이 시사되어 각각 그래픽적으로 680, 682 로 표시된다.

[0044] 애셋 디스크립터들

[0045] 알 수 있는 바와 같이, 애셋 디스크립터들은 액티비티들 간의 연결을 생성하고 잠재적인 액티비티 연결들을 식별하기 위하여 입력들 및 출력들을 모델링하는데 이용된다. 특정 실시형태들에서, 애셋 디스크립터들은 애셋 레지스트리에서의 기존 애셋들과 상관되는데 이용될 수 있다.

[0046] 특정 실시형태들에서, 애셋 디스크립터들은 정확한 또는 파라미터화된 매칭을 위해 이용될 수 있고, 여기에서 파라미터화된 매칭은 디스크립터들이 비교될 때 일부 와일드카드형 능력들을 제공한다. 본 실시예들에서, 완전히 정의된 애셋 디스크립터는 예를 들어 원 안에 대문자로 표기된다:

A

[0047]

[0048]

파라미터화된 애셋 디스크립터는 원 안에 "프라임 표시" 된 대문자로 표기될 수 있다. 예를 들어:

A'

[0049]

[0050]

몇몇 추가의 정의들:

[0051]

- **액티비티 인스턴스.** 모든 입력 및 출력 애셋 디스크립터들이 완전하게 정의된 액티비티이며, 정의되지 않은 파라미터가 없음을 의미한다.

[0052]

- **액티비티 템플릿.** 재사용을 용이하게 하기 위하여 하나 이상의 파라미터화된 애셋 디스크립터를 갖는 액티비티이다. 그러나, 이는 필요요건은 아니다.

[0053]

액티비티들은 이들의 입력들 및 출력들의 관점에서 정의된다. 입력들 및 출력들은 이어서 이들의 애셋 디스크립터들에 의해 정의된다. 입력들 및 출력들의 특정 조합은 (이것이 명명된 것과는 무관하게) 액티비티의 "시그너처" 를 결정한다. 도 7 의 예에서, 액티비티 1 (700) 은 입력들 (702, 704) 에서 애셋 (A) 및 애셋 (B) 을 취하고, 출력 (706) 에서 애셋 (C) 을 제공한다. 액티비티 2 (710) 는 출력에서 애셋 (C) 을 제공하지만 입력 (712) 에서 애셋 (X) 을 취한다. 이 예에서, 이들 액티비티들 각각은 이들이 상이한 입력들을 요구하지만 양쪽 모두 동일한 출력을 생성하는 점에서 고유하다.

[0054]

액티비티 인스턴스들 간의 연결을 모델링하기 위하여, 업스트림 액티비티의 출력은 다운스트림 액티비티의 입력과 매칭하는 것이 필요하다. 다수의 다운스트림 액티비티들은 동일한 애셋을 소비할 수도 있다. 도 8 의 제 1 모델 (800) 에서, 액티비티 인스턴스 1 (810), 액티비티 인스턴스 2 (820), 액티비티 인스턴스 3 (830), 및 액티비티 인스턴스 4 (840) 가 존재한다. 제 2 모델 (850) 에서, 액티비티 인스턴스 1 (810) 이 애셋 (A) 을 인스턴스들 2 (820) 및 3 (830) 에 제공하는 연결들이 도시되어 있다. 액티비티 인스턴스 3 (830) 은 2 개의 입력들 (A) 및 (B) 을 요구한다. 애셋 (B) 은 액티비티 인스턴스 4 (840) 에 의해 제공된다.

[0055]

액티비티 인스턴스와 액티비티 템플릿 간의 잠재적인 연결을 모델링하기 위하여, 인스턴스의 입력은 템플릿의 출력과의 템플릿 매칭을 행할 수 있다. 이것의 일 예는 도 9 의 모델 (900) 에서 볼 수 있다. 도 9 에서, 액티비티 인스턴스 4 (930) 의 입력 (932) 의 애셋 디스크립터 (A) 는 액티비티 템플릿 1 (910) 의 출력 (912) 의 애셋 디스크립터 (A') 와 매칭하고, 액티비티 인스턴스 5 (940) 의 출력 (942) 상의 애셋 디스크립터 (B) 는 액티비티 템플릿 2 (920) 의 입력 (922) 의 애셋 디스크립터 (B') 와 매칭한다.

[0056]

이 포인트까지 단일의 문자를 이용하는 것은 고 레벨에서의 개시된 개념들을 예시하는 한 접근 방식이었다. 실제로, 애셋을 기술하는데 절대적으로 매우 많은 방법들이 존재한다. 일 실시형태는 애셋 디스크립터들을 생성하기 위한 휴먼 판독가능하고 유연성있는 메카니즘을 제공하기 위해 이름/값 페어들의 집합을 이용한다. 애셋 디스크립터는 애셋 일반 애셋 디스크립터 포맷을 기술하기 위하여 전체적으로 취해진 하나 이상의 이름/값 페어들로 구성될 수 있다:

```
{
    name1:value1,
    name2:value2,
    name3:value3,
    ...
}
```

[0057]

[0058]

예:

```
{
  Title:'The Hobbit',
  Version:'Trailer',
  Type:'Netflix Encoding'
}
```

[0059]

[0060]

파라미터화된 디스크립션은 간단히 하나 이상의 값들을 블랭크로 남겨둔다. 이 예에서, 아래의 "타이틀 (Title)" 및 "버전 (Version)" 양쪽 모두가 파라미터들이다.

[0061]

예:

```
{
  Title:",
  Version:",
  Type:'Netflix Encoding'
}
```

[0062]

[0063]

애셋 식별자들은 애셋 디스크립터의 정규화된 버전들이다. 먼저, 애셋 디스크립터가 정규화되어, 케이스, 및 이름/값 페어 순서가 비교에 영향을 주지 않게 된다. 애셋 디스크립터를 정규화하기 위하여, 이름들 및 값들이 하위 케이스로 강제되고 (선택적), 그 후, 연결 (concatenate) 된 이름 및 결과에 의해 분류된다.

[0064]

예:

[0065]

애셋 디스크립터:

```
{
  Title:'The Hobbit',
  Version:'Trailer',
  Type:'Netflix Encoding'
}
```

[0066]

[0067]

애셋 식별자:

title:'the hobbit',type:'netflix encoding',version:'trailer'

[0068]

[0069]

로 된다.

[0070]

선택적으로, 아래에 도시된 고유 수 (헥스) 식별자를 생성하기 위해, 위의 결과에 대하여 암호화 해시가 수행될 수 있다:

[0071]

147c21df6e470da7879307dbfb2e2a5d3e9c40719ba2a1a840bf71c732f71b2f

[0072]

애셋 식별자의 암호화 해시 변수는 텍스트열 버전이 너무 많은 문제를 가져 편리할 수 없는 HTML 에서 클래스 또는 id 파라미터로서 사용자 인터페이스 엘리먼트들을 식별할 때 특히 유용하다.

[0073]

애셋 레퍼런스는 언더스코어 "_" 문자에 의해 분리되는 오리지널 애셋 디스크립터에서 정의된 순서로 값들을 연결한다. 이는 애셋을 덜 형식적으로 기술하도록 휴먼 관독가능 속기를 제공한다.

- [0074] 위의 예는 다음과 같이 된다:
- [0075] 'The Hobbit_Trailer_Netflix Encoding'
- [0076] 애셋 디스크립터들 및 애셋 식별자들은 완전한 식별을 위해 양쪽 모두 이용될 수 있다. 그러나, 애셋 레퍼런스는 디스플레이 편리성을 위한 것일 뿐, 모호한 레퍼런싱을 위해 의존되지 않아야 한다.
- [0077] 완전 정의된 애셋 디스크립터들의 정확한 매칭은 애셋 식별자들을 이용하여 직접 수행될 수 있다.
- [0078] 완전 정의된 애셋 디스크립터를 파라미터화된 디스크립터와 매칭시킬 때, 다음의 규칙이 이용된다:
- [0079] - 이름/값 페어들이 비교 이전에 하위 케이스로 강제된다.
- [0080] - 파라미터화된 애셋 디스크립터는 완전 정의된 애셋 디스크립터와 정확히 동일한 이름 엔트리들을 가져야 한다. 이름들의 순서는 중요하지 않다.
- [0081] - 파라미터화된 애셋 디스크립터 엔트리가 값에 대하여 블랭크를 가지면, 완전 정의된 애셋 디스크립터에서의 대응하는 값과 무관하게 매칭할 것이다. 파라미터화된 애셋 디스크립터가 값에 대하여 비-블랭크 엔트리를 가지면, (정규화 후에) 정확히 매칭해야 한다.
- [0082] 예들이 도 10 의 예시적인 테이블 (1000) 에 도시된다.
- [0083] 액티비티들의 파이프라인들을 구축할 때, 이는 원하는 출력 (소비 액티비티) 을 선택하고 원하는 파라미터 값들로 채우고 그 후, 파이프라인을 완성하도록 액티비티들이 식별될 때 원하는 값들의 전파를 허용하기에 충분해야 한다. 도 11 의 예는 화살표 (1102) 에 의해 표시되는 바와 같이 종료에서부터 시작으로 구축되는 파이프라인 (1100) 을 세부 설명한다. 파이프라인들은 또한 시작에서 종료로 또는 미들 아웃 (middle-out) 으로 구축될 수도 있다.
- [0084] 도 11b 의 단계 1.0 (1110) 은 종료 액티비티에서 시작한다. 이 예에서, 제공된 액티비티는 입력에 대한 애셋 디스크립터 "A" 에 대한 파라미터들이 특정되어, 액티비티 템플릿 액티비티 인스턴스를 형성하는 선택된 액티비티 템플릿 (1112) 이다. 그 후, 애셋 디스크립터 "A" 에 대하여 특정된 파라미터들이 연결 (1114) 을 통하여 제 2 제공된 액티비티 템플릿 (1122) 으로 패스될 수 있다.
- [0085] 단계 2.0 (1120) 에서, 연결 (1114) 을 통하여 패스되는 특정된 파라미터들은 액티비티 템플릿 액티비티 인스턴스를 형성하는, 제공된 제 2 액티비티 템플릿 (1122) 의 입력들과 연관된 애셋 디스크립터들 ("B" 및 "C") 에 대한 파라미터들을 특정하는데 이용된다. 이 예에서, 애셋 디스크립터 "C"에 대한 언어의 파라미터는 이미 이것이 특정되었기 때문에 패스되지 않는다.
- [0086] 도 11a 의 단계 3.0 (1130) 에서, 제 2 액티비티 인스턴스로부터의 특정된 파라미터들이 연결 (1124) 을 통하여, 제공된 제 3 액티비티 템플릿 (1132) 으로 패스된다. 패스되어진 특정된 파라미터들은 제공된 제 3 액티비티 템플릿 (1132) 의 출력과 연관된 애셋 디스크립터 "C"에 대한 파라미터들을 특정하여 액티비티 템플릿 액티비티 인스턴스를 형성하는데 이용된다.
- [0087] 단계 4.0 (1140) 에서, 제 2 액티비티 인스턴스로부터의 특정된 파라미터들은 연결 (1126) 을 통하여 제공된 제 4 액티비티 템플릿 (1142) 으로 패스된다. 패스되어진 특정된 파라미터들은 제공된 제 4 액티비티 템플릿 (1142) 의 출력과 연관된 애셋 디스크립터 "B" 에 대한 파라미터들을 특정하여 액티비티 템플릿 액티비티 인스턴스를 형성하는데 이용된다.
- [0088] 실제 콘텐츠 생성 및 분배 파이프라인을 모델링하는데 있어서, 다음 휴리스틱들이 유용성을 입증할 수 있다.
- [0089] - 모든 액티비티 디스크립터들은 "타이틀" 및 "버전" 을 포함해야 한다. 이들은 전체 파이프라인 인스턴스에 대한 애셋들을 차별화시킨다.
- [0090] - 모든 액티비티 디스크립터들은 "타입"을 포함해야 한다. 타입 필드는 액티비티에 의해 생성되는 콘텐츠의 유형 (비디오, 오디오, 디지털 시네마 패키지 등) 을 표현한다.
- [0091] - 다른 유용한 애셋 디스크립터 엔트리들은: "언어", "애스펙트비", "DubSubOV", 및 "포맷"이다. 이들은 "타입" 값에 기초하여 관련될 수도 있거나 또는 관련되지 않을 수도 있다. 그 외의 엔트리들이 시간에 따라 사용에 전개될 수도 있다.

- [0092] **애셋 레지스트리**
- [0093] 여기에 설명된 그래픽 모델들의 액티비티들 및 애셋들이 종종 실제 액티비티들 또는 애셋들을 표현할 수 있기 때문에, 이는 애셋 레지스트리를 제공하고 유지하기에 유익할 수 있다. 애셋 레지스트리는 애셋 디스크립터들 (또는 애셋 식별자들) 을 실제 애셋들의 로케이션에 매핑시킨다. 완전 정의된 애셋 디스크립터들에 대하여 기존 애셋들을 레지스터함으로써, 정의시에, 파이프라인으로부터 불필요한 액티비티들을 제거하는 것이 가능하다.
- [0094] 이전에 정의된 파이프라인 구축 전략에 대한 변경으로서, 레지스트리를 검사하는 단계는 액티비티 템플릿들을 매칭하려는 임의의 시도를 진행시킬 수 있다. 주어진 애셋 디스크립터/식별자에 매핑된 상이한 로케이션들에서 애셋의 복수의 카피들을 갖는 것이 가능하다.
- [0095] **이행 모델링**
- [0096] 여기에 설명된 모델링 방법론에 있어서, 액티비티들 및 이들의 연결들은 애셋 종속값들에 기초하여 모델링된다 (디테일들에 대해서는 애셋 디스크립터들 섹션을 참조). 액티비티들 간의 각각의 연결은 후속하는 액티비티의 입력에 대한 한 액티비티의 출력의 논리 종속값을 표현한다. 액티비티에서 액티비티로의 진행을 추적하기 위하여, 연결들이 이행 상태를 가질 수 있다. 워크플로우 모델에서의 이행 상태를 모델링하기 위한 예시적인 방법론은 도 12 의 플로우차트 (1200) 에서 보여질 수 있다.
- [0097] 가장 간단하게 말하면, 방법은 두개의 단계들을 포함한다. 제 1 단계 (1210) 는 적어도 제 1 액티비티의 그래픽 표현 및 제 2 액티비티의 그래픽 표현을 갖는 워크플로우의 모델을 제공하는 것이며, 여기에서 액티비티들은 매칭하는 애셋 디스크립터들에 기초하여 연결된다. 제 2 단계 (1220) 는 제 1 및 제 2 액티비티들의 그래픽 표현들 간의 연결에 적어도 기초하여, 매칭하는 애셋 디스크립터들에 의해 표시되는 적어도 하나의 애셋의 상태를 결정하는 것이다. 단계들은 도 13 을 참조로 보다 자세하게 아래 설명된다.
- [0098] 도 13 의 다이어그램 1300 에서, 워크플로우의 모델은 도 12 의 방법론의 단계 1210 에서 설명된 것으로서 제공된다. 이 예에서, 모델은 제 1 액티비티 (1310)(여기서는 소스 액티비티) 및 제 2 액티비티 (1320)(여기서는 목적지 액티비티) 의 그래픽 표현들을 포함한다. 제 1 (1310) 및 제 2 (1320) 액티비티들은 매칭하는 애셋 디스크립터들에 기초하여 연결 (1330) 된다. 그 후, 연결 (1330) 에 기초하는 매칭하는 애셋 디스크립터들에 의해 표시되는 적어도 하나의 애셋에 대하여 이행 상태 (1340) 가 결정된다.
- [0099] 이행 상태는 한 액티비티로부터 다음 액티비티로 이동하는 물리적/전자적 애셋의 상태를 반영한다. 액티비티가 예상되는 출력 (애셋) 을 생성하면, 이는 종속 다운스트림 액티비티들로 물리적/전자적으로 전송되어, 프로세스가 계속될 수 있다. 이행 메카니즘은 애셋 이동의 상태 (예를 들어, 보류, 전송, 수신, 에러) 를 추적한다. 특정 실시형태들에서, 이행 상태는 모델의 액티비티들 또는 다른 엘리먼트들의 그래픽 표현의 일부로서 그래픽적으로 디스플레이되거나 달리 표시될 수 있다.
- [0100] 다른 특정 실시형태들에서, 액티비티의 상태는 액티비티에 의해 생성 및/또는 소비되는 애셋들의 이행 상태에 기초하여 결정될 수 있다. 추가의 특정 실시형태들에서, 액티비티 상태는 모델의 액티비티들 또는 다른 엘리먼트들의 그래픽 표현의 일부로서 그래픽적으로 디스플레이되거나 달리 표시될 수 있다.
- [0101] 특정 실시형태들에서, 단일의 물리적/전자적 전달은 다수의 다운스트림 액티비티들에 의해 영향을 받을 수 있고, 다수의 이행 레코드들이 중복될 것이다. 이를 해결하기 위하여, 액티비티 대 액티비티 종속성 관계로부터 액티비티 대 설비 관계로 변경이 이루어질 수 있다. 이것의 일 예를 도 14 에서 볼 수 있다.
- [0102] 도 14 의 예시적인 다이어그램 (1400) 에서, 액티비티 B (1420) 및 액티비티 C (1430) 는 동일한 공유되는 설비 (1450)(설비 Y) 에 있다. 따라서, 소스, 액티비티 A (1410) 와, 공유되는 설비 (1450)(설비 Y) 간에 연결 (1402) 이 도시된다. 목적지 액티비티 D (1440) 는 상이한 설비 (설비 Z) 에 있으며, 이에 따라 별도의 연결 (1404) 이 액티비티 A (1410) 와 액티비티 D (1440) 간에 제공된다.
- [0103] 용어 "설비" 는 시스템에 이용된 다른 "로케이션" 레퍼런스들로부터 차별되도록 선택되었다. 또한, 특정 애셋 종속 관계가 여전히 유지되어, 이행이 설비 뿐만 아니라 전달될 특정 애셋에도 종속된다. 이러한 실시형태들에서, 이때 이행 상태는 소스 액티비티로부터 설비로의 특정 애셋의 전달을 표현하고, 이어서 이 설비는 다수의 목적지 액티비티들에 공유될 수도 있다. 이들 엘리먼트들의 상호작용의 다이어그램 (1500) 을 도 15 에서 볼 수 있다.
- [0104] 도 15 의 다이어그램 (1500) 에서는, 워크플로우의 모델이 제공된다. 이 예에서, 모델은 제 1 액티비티

(1510)(여기서는 소스 액티비티) 및 제 2 액티비티 (1520)(여기서는 목적지 액티비티) 의 그래픽 표현들을 포함한다. 제 1 (1510) 및 제 2 (1520) 액티비티들 간의 관계 (1530) 는 매칭하는 애셋 디스크립터들에 기초한다. 그 후, 관계 (1530) 에 기초하는 매칭하는 애셋 디스크립터들에 의해 표시되는 적어도 하나의 애셋에 대하여 이행 상태 (1540) 가 결정된다. 이 방법에 대한 실제적인 제약으로서, 설비 (1560) 는 제 2 액티비티 (1520) 로 레퍼런스되는 벤더 (1550) 와 연관된다. 이 방법으로, 변하는 벤더 정보는 적절한 설비 할당을 야기한다. 주어진 설비는 다수의 벤더들에 의해 레퍼런스될 수도 있다.

[0105] 이행 상태는 애셋 디스크립션을 공유하는 액티비티들의 각각의 페어에 대하여 레퍼런스될 수 있다. 이행 상태를 생성할 때, 목적지 액티비티의 벤더 설비 디스크립터는 이행이 이미 생성되었는지를 결정하는데 이용될 수도 있고, 이미 생성되었다면, 기존의 이행 레코드가 레퍼런스될 수 있고, 생성되지 않았다면 새로운 이행이 생성될 수 있다. 특정 실시형태들에서, 역 도메인 이름 선택스가 이것이 휴먼 관독가능하도록 설비 디스크립터에 이용될 수 있다 (예를 들어, technicolor.perivale, technicolor.perivale.transcodingDept). 고유한 설비들은 별도의 이행들을 보장하지만, 동일한 물리적 로케이션에 다수의 "설비들"이 존재할 수 있다. 이는 이행 상태를 독립적으로 추적하는 별도의 레코드들을 야기한다.

[0106] **애셋 데이터 상으로의 프로세스 정보의 매핑**

[0107] 이 시점까지의 워크플로우 모델링은 모델링된 파이프라인과 일치하는 애셋 생성 프로세스들 (액티비티들) 구동에 초점을 맞추었다. 즉, 시스템은 애셋 디스크립터 방식에 따라 정의된 모델, 어떤 액티비티들이 무엇에 의존하는지, 그리고 애셋들의 강제된 등록에 기초하여 기술되었다. 이하, 기본적인 액티비티들에 직접 영향을 주지 않고 패시브 방식으로 파이프라인 정보의 유사한 레벨을 전달하기 위한 대안의 접근 방식이 제공된다. 일반적인 사상은, 전체적인 프로세스의 상태를 유도하기 위하여 애셋 데이터 상에 파이프라인 모델 (프로세스 데이터) 을 오버레이하는 것이다.

[0108] 애셋이 생성될 때, 이들은 애셋 레지스트리 시스템에 레지스터되는 것으로 본다. 이 방법론은 위에 설명된 애셋 디스크립터들 및 애셋 레지스트리의 개념과 일치하는 추가적인 구조화된 데이터를 요구한다. 애셋 디스크립터들의 이름/값들은 일치하는 것이 필요하다 (예를 들어, 타이틀들, 언어, 애스펙트 비 등).

[0109] 이어서 애셋 디스크립터들을 레퍼런스하는 액티비티들로 구성된 프로세스 모델이 (대체로 프로세스 레지스트리로부터) 획득되며, 파이프라인 상태 정보를 유도하기 위해 애셋 레지스트리에서의 데이터를 뷰하는데 이용된다. 예시적인 방법론은 도 16 의 플로우차트 (1600) 에서 볼 수 있다.

[0110] 가장 간단하게 말하면, 방법은 두개의 단계들을 포함한다. 제 1 단계 (1610) 는 워크플로우의 모델에 요구되는 애셋이 존재함을 결정한다. 제 2 단계 (1620) 는 기존의 애셋을 수반하는 액티비티의 그래픽 표현을 제공하는 것이다. 단계들은 도 17 을 참조로 보다 자세하게 아래 설명된다.

[0111] 도 17 의 다이어그램 (1700) 은 3 개의 부분들: 애셋 레지스트리 (1710), 프로세스 모델 (1720) 및 추론 상태 (1730) 를 갖는다.

[0112] 이는 제 1 단계 (1610) 가 수행되는 애셋 레지스트리 (1710) 에 있다. 애셋이 존재하는지를 결정하기 위하여, 애셋 레지스트리가 쿼리된다. 애셋 레지스트리는 생성되었거나 이전에 존재하였던 애셋들의 집합, 이를 테면, 데이터베이스이다. 이 예에서, 애셋 레지스트리는 주어진 워크플로우에 대한 애셋들만을 포함하는 것으로 본다. 그러나, 당해 기술 분야의 당업자에게는 애셋 레지스트리가 현재 워크플로우 모델의 부분이 아닌 애셋들을 포함하는 임의의 수의 레지스터된 애셋들을 포함할 수 있음이 명백할 것이다. 도 17 의 예에서, 3 개의 애셋들 (A, B, C) 이 이미 존재한다고 결정된다.

[0113] 도 17 의 프로세스 모델 (1720) 에서, 애셋(들)을 수반하는 액티비티들의 그래픽 표현들이 제공된다. 이러한 액티비티들은 애셋들을 생성 또는 소비하는 액티비티를 포함할 수 있다. 특정 실시형태들에서, 추가로 연결될 수 있는 생성 및 소비 액티비티 양쪽 모두의 그래픽 표현들이 제공될 수 있다. 다른 실시형태들에서, 다수의 사전 존재 애셋들이 존재하면, 전체 파이프라인들의 그래픽 표현들은 액티비티들 모두가 연결되는 경우에 제공될 수 있다. 특정 실시형태들에서, 프로세스 또는 모델 레지스트리가 제공될 수 있다. 애셋 레지스트리와 유사한 프로세스 레지스트리는 이미 생성되었거나 또는 이전에 이용된 파이프라인 모델들의 집합, 이를 테면 데이터베이스이다. 일부 이러한 실시형태들에서, 레지스터된 파이프라인 모델들은 애셋 레지스트리에 레지스터된 애셋들과 매칭되거나 또는 달리 링크될 수 있다.

[0114] 추론 상태 (1730) 에서, 파이프라인 모델에서의 각각의 액티비티에 대해, 대응하는 애셋 디스크립터가 애셋 레지스트리로부터 쿼리된다. 디스크립터와 매칭하는 애셋이 발견되면, 그 액티비티는 완료된 것으로 본다.

그 후, 이들 액티비티들에 대한 입력들이 완료되었지만 현재 액티비티의 출력이 완료되지 않은지를 봄으로써 어떤 액티비티들이 진행중에 있어야 하는지를 추론하는 것이 가능하다. 상태 테이블의 일 예는 1740 에서 볼 수 있다.

[0115] 이 기본 메카니즘을 이용하여, 데이터 세트들이 복수의 방식들로 접근될 수 있다.

[0116] 미리 정의된 파이프라인: 이 시나리오에서, 파이프라인의 세부사항들은 미리 공지된다 (즉, 타이틀, 버전, 애스펙트 비 등이 정의된다). 이는 애셋 레지스트리에서의 애셋들에 대한 직접 매핑을 허용한다. 이 접근 방식의 이점은 임의의 대응하는 액티비티들이 아직 등록되지 않은 파이프라인의 상태를 사용자가 뷰할 수 있다는 점이다.

[0117] 기존 애셋들로부터의 추론: 시스템은 레지스트리에 이미 존재하는 애셋들에 대해서만 알려진 파이프라인을 추정할 수 있다. 일 예로서, 트랜슬레이션이 수신되고 특정 타이틀, 버전, 언어에 대하여 레지스터되면, 시스템은 주어진 값들로 획득 트랜슬레이션 액티비티에 대한 인스턴스를 생성하고, 그 후 다른 액티비티 입력들에 출력을 매칭시키고 이어서 후속하는 액티비티들의 입력들에 이들 액티비티들의 출력들을 매칭시킴으로써 나머지 파이프라인을 추론한다. 프로세스는 찾기 액티비티의 임의의 입력들을 매칭시키고 출력들에 대한 입력들의 경로를 추종함으로써 업스트림 방향으로 또한 발생할 수 있다.

[0118] 기존 데이터 상에 프로세스 모델을 오버레이하는 이 방법의 특징은 사용자가 어느 파이프라인 변동이 최상으로 매칭하는지를 보도록 다수의 파이프라인들을 "뷰잉 렌즈들" 로서 얻을 수 있다는 점이다.

[0119] 이전 설명이 워크플로우의 모델을 생성하고 상태를 모니터링하는 것에 초점을 두었지만, 일부 실시형태들에서는, 파이프라인의 오버뷰를 제공하는 것이 유리할 수도 있다. 이에 의해, 사용자 인터페이스는 워크플로우의 그래픽 모델을 제공할 뿐만 아니라 워크플로우 프로세스의 하이 레벨 투시도를 제공하도록 제공될 수도 있다. 이러한 사용자 인터페이스의 예들은 도 18 내지 도 25 에서 볼 수 있다. 이들 예들은 전자 디바이스 상의 웹 브라우저 또는 애플리케이션을 통해서와 같이 시스템과 상호작용할 때 사용자가 제공받을 수 있는 스크린 샷들이다.

[0120] 특정 실시형태들에 따라, 본 개시물의 시스템이 론칭될 때, 사용자는 크리덴셜들에 대해 프롬프트되고 도 18 의 스크린 샷에서 묘사된 바와 같이 프로듀서의 워크스페이스를 제공받는다. 이 워크스페이스 (1800) 는 엔트리들, 뷰잉 상태, 및 종속성들을 액티비티 디테일들과 함께 생성하는 수단을 제공한다. 사용자는 또한 액티비티 디테일 패널로부터 동작 프로세스를 구동할 수 있다. 프로듀서의 워크스페이스 (1800) 는 3 개의 패널들: 상품 대시보드 (1810), 필터링된 파이프라인 뷰 (1820), 및 세부사항 뷰 (1830) 로 구성된다. 다른 워크스페이스 뷰들은 사용자에게 의한 선택 (180) 에 유용하며, 아래 보다 자세하게 설명된다.

[0121] 도 19 는 도 18 의 상품 대시보드 (1810) 의 일 예이다. 대시보드의 각각의 라인은 상품 (1900) 에 관련되며, 상품들과 연관된 액티비티들 (1910) 을 표시한다. 상품 대시보드 (1810) 는 추가로 사용자로 하여금 시스템에 타이틀/버전/포맷 (1920) 을 추가하게 하고, 기존 타이틀들/버전들/포맷들 (1930) 로부터 선택하게 하고, 특정 상품들을 요청하게 하거나 또는 새로운 상품 (1940) 을 추가하게 하고, 그리고 새로운 언어/애스펙트 비/DubSub 라인 (1950) 을 추가하게 한다. 여기에서 정보를 입력하는 것은 요건을 충족시키기 위해 기본적인 시스템이 적절한 파이프라인을 구축하게 한다. 시스템은 상품들 간에 특정 액티비티가 공유되는지를 알기에 충분히 "스마트"하다.

[0122] 도 20 은 필터링된 파이프라인 뷰 (1820) 의 일 예이다. 필터링된 파이프라인 뷰 (1820) 는 액티비티들 및 이들의 상호종속성들의 그래픽 표현을 디스플레이한다. 필터 파이프라인 뷰는 리스트 뷰 (2010) 및 통상적인 파이프라인 모델 뷰 양쪽 모두를 제공한다. 이 뷰에 도시된 액티비티들은 상품 대시보드에서 선택된 로우에 관련된다. 각각의 액티비티 (2030) 는 입력들 (좌측에서의 원들) 및/또는 출력들 (우측에서의 원들) 을 가진 박스로서 그래픽적으로 표현된다. 이 실시형태에서, 애셋들의 이행 상태는, 애셋이 수신 또는 생성되었다는 것을 표시하는 입력들 및/또는 출력들을 채우거나 달리 강조함으로써 표시된다. 특정의 추가 실시형태들에서, 액티비티의 상태는 또한 그래픽적으로 표현될 수 있다. 본 실시예에서, 상태를 표시하도록 채워질 수 있는 좌측 하단 코너부에 박스가 제공된다. 비어 있는 박스는 시작되지 않은 액티비티를 의미하고, 부분적으로 채워진 박스는 액티비티가 진행중에 있음을 의미하고, 채워진 박스는 액티비티가 수행되었음을 의미한다.

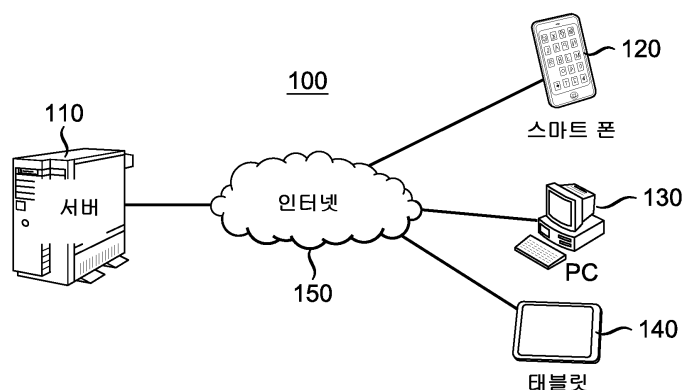
[0123] 이 패널은 또한 실행 워크스페이스에서 이용가능하다. 그 경우, 실행 패널로부터의 액티비티를 선택하는 것은 선택된 액티비티에 대하여 파이프라인을 디스플레이할 것이다.

- [0124] 도 21 은 세부 뷰 (1830) 의 일 예이다. 디테일 뷰 패널 (1830) 은 버튼들이 제공될 수 있는 3 개의 간단한 액션들에 기초하여 액티비티의 상태를 업데이트하는 능력을 제공한다.
- [0125] - **준비로 설정** (2120) 은 액티비티가 완료되었음을 표시하는데 이용된다.
- [0126] - **전송** (도시 생략) 은 애셋이 다음 액티비티 또는 액티비티들에 전송될 때를 시스템이 알게 하는데 이용된다.
- [0127] - **수신** (도시 생략) 은 애셋이 업스트림 액티비티로부터 수신될 때를 시스템이 알게 하는데 이용된다.
- [0128] 특정 실시형태들에서, **리바이스** 액션은 액티비티가 준비로 일단 설정되면 제공될 수 있다. 리바이스는 사용자 하여금 변경의 영향을 알고 나서 그 변경이 재실행되도록 커밋하는 것을 허용한다.
- [0129] 이들 액션들에 의해 제공된 정보는 애셋들의 이행 상태 뿐만 아니라 액티비티들의 상태 자체를 결정하는데 이용될 수 있다.
- [0130] 디테일 뷰 패널 (1830) 의 현재 디스플레이에서, 강조된 "ACTIONS" 탭으로 표시되는 액션들이 디스플레이되고 있다. "DETAILS" 탭이 선택되면, 날짜 및 벤더 정보로 인한 정보가 디스플레이된다. 날짜가 설정되거나 날짜를 놓치면 (예를 들어, 기일까지 행하지 못함), 시스템은 이것을 이슈로서 플래그한다.
- [0131] 다른 가능한 워크스페이스는 도 22 의 스크린 샷에 묘사된 매니저 워크스페이스 (2200) 이다. 매니저 워크스페이스 (2200) 는 매니저 패널 (2210) 및 액티비티 디테일 패널 (1830) 로 구성된다.
- [0132] 매니저 패널 (2210) 은 특정 액티비티를 경험하는 모든 작업을 제시한다. 매니저 패널 (2210) 은 사용자로 하여금 필드 (2220) 를 이용하여 특정 액티비티를 선택하는 것을 허용한다. 그 후, 선택된 액티비티와 연관된 작업 또는 테스크들은 패널 (2210) 에 디스플레이된다. 필터 (2230) 는 무엇이 디스플레이되고 있는지를 조정하는데 이용될 수 있다. 디폴트 필터는 작업중에 필요한 일들만을 보여주지만, 필터들은 무엇이 오고 무엇이 이미 완료되었는지를 보여주도록 설정될 수 있다. 작업이 행해지면, "준비로 설정" (2240) 이 선택될 수 있고 (필터들이 달리 설정되지 않으면) 테스크들은 보드로부터 드롭 오프된다. 매니저 워크스페이스 (2200) 의 액티비티 디테일 패널 (1830) 은 도 21 을 참조로 설명된 바와 같이 동작한다.
- [0133] 다른 가능한 워크스페이스는 도 23 의 스크린 샷에 묘사된 바와 같은 데이터 I/O 워크스페이스 (2300) 이다. 데이터 I/O 워크스페이스 (2300) 는 데이터 I/O 패널 (2310) 및 액티비티 디테일 패널 (1830) 로 구성된다.
- [0134] 데이터 I/O 패널 (2310) 은 특정 설비에 관련된 모든 액션들을 보여주도록 설계된다. 이는 설비의 안과 밖으로 애셋들을 이동시키는 사람이 애셋들을 생성 또는 변경하는 작업을 수행하는 사람들과 상이할 수도 있다는 아이디어를 수용하는 액션들의 전송 및 수신을 보여주도록 기본적으로 설계된다. 데이터 I/O 패널 (2310) 은 사용자로 하여금 필드 (2320) 를 이용하여 특정 액티비티를 선택하는 것을 허용한다. 그 후, 선택된 액티비티와 연관된 작업 또는 테스크들 뿐만 아니라 이들의 상태 (2340) 가 패널 (2310) 에 디스플레이된다. 필터 (2330) 는 무엇이 디스플레이되고 있는지를 조정하는데 이용될 수 있다. 디폴트 필터는 작업중에 필요한 일들만을 보여주지만, 필터들은 무엇이 오고 무엇이 이미 완료되었는지를 보여주도록 설정될 수 있다. 액션, 이를 테면, "준비로 설정", "전송", 및 "수신" (2350) 이 선택될 수 있고, 상태 (2340) 가 이에 따라 업데이트된다. 데이터 I/O 워크스페이스 (2300) 의 액티비티 디테일 패널 (1830) 은 도 21 을 참조로 설명된 바와 같이 동작한다.
- [0135] 다른 가능한 워크스페이스는 도 24 의 스크린 샷에 묘사된 실행 워크스페이스 (2400) 이다. 실행 워크스페이스 (2400) 는 실행 패널 (2410), 필터링된 파이프라인 패널 (1820), 및 액티비티 디테일 패널 (1830) 로 구성된다.
- [0136] 실행 패널 (2410) 은 그래프들 및 테스크 리스트들과 같은 오버뷰 데이터를 제공한다. 필터 (2420) 는 무엇이 디스플레이되고 있는지를 조정하는데 이용될 수 있다. 이 예에서, 상단 박스는 무엇이 필터링되는지를 결정하며, 하단 박스는 이용할 값을 설정한다. 결과들은 셀렉터 (2430) 를 이용하여 그룹화될 수 있다. 그룹핑 헤더들 (2440) 를 선택하는 것은 그룹들이 팽창, 또는 소멸 (collapse) 되는 것을 허용한다. 패널 (2450) 에서의 아이템을 선택하는 것은 관련 액티비티들 및 테스크들이, 필터링된 파이프라인 패널 (1820) 및 액티비티 디테일 패널 (1830) 에 디스플레이되는 것을 야기한다.
- [0137] 실행 워크스페이스 (2200) 의 필터링된 파이프라인 패널 (1820) 은 도 20 을 참조로 설명된 바와 같이 동작한다. 실행 워크스페이스 (2200) 의 액티비티 디테일 패널 (1830) 은 도 21 을 참조로 설명된 바와 같이 동작한다.

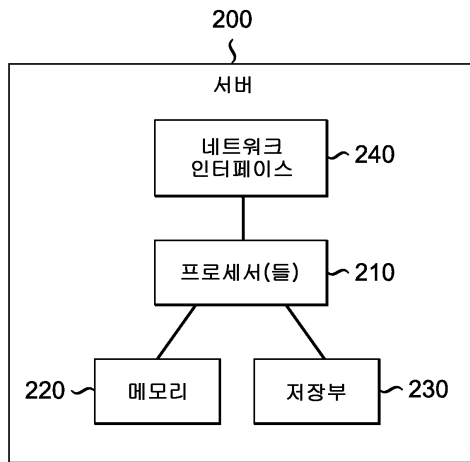
- [0138] 최종의 예시적인 워크스페이스는 도 25 의 스크린 샷에 설명된 바와 같은 파이프라인 빌더 (2500) 이다. 파이프라인 빌더 (2500) 는 템플릿 리스트 (2510) 및 워크스페이스 (2520) 로 구성된다.
- [0139] 템플릿 리스트 (2510) 는 프로젝트 또는 워크플로우를 선택하기 위한 필드 (2512) 를 제공한다. 그 후, 선택된 프로젝트 또는 워크플로우에 대한 관련 액티비티 템플릿들이 템플릿 리스트에 제공된다. 이들 결과들은 추가로 필터 기능 (2514) 을 이용하여 필터링될 수 있다. 필요에 따라, 새로운 템플릿이 생성 툴 (2516) 을 이용하여 생성될 수 있다.
- [0140] 워크스페이스 (2520) 는 이 개시물 전반에 걸쳐 개시된 바와 같이 파이프라인 모델을 구축하기 위한 기능을 제공한다. 이 실시형태에서, 액티비티 (2530) 의 그래픽 표현의 입력 또는 출력을 선택하는 것은 템플릿 리스트 (2510) 에서의 결과가 입력 또는 출력과 연관된 애셋 디스크립터들에 기초하여 필터링되게 한다.
- [0141] 여기에 개시된 여러 실시형태들은 하드웨어, 펌웨어, 소프트웨어 또는 이들의 임의의 조합으로서 구현될 수도 있다. 또한, 소프트웨어는 컴퓨터 판독가능 매체 또는 프로그램 저장 매체 상에서 실제적으로 구현되는 애플리케이션 프로그램으로서 바람직하게 구현된다. 애플리케이션 프로그램은 임의의 적절한 아키텍처를 포함하는 머신에 업데이트되어 머신에 의해 실행될 수도 있다. 바람직하게 머신은 하드웨어, 이를 테면, 하나 이상의 중앙 프로세싱 유닛들 ("CPUs"), 메모리 및 입력/출력 인터페이스들을 갖는 컴퓨터 플랫폼 상에서 구현된다. 컴퓨터 플랫폼은 또한 오퍼레이팅 시스템 및 마이크로명령 코드를 포함할 수도 있다. 여기에 설명된 여러 프로세스들 및 기능들은 마이크로명령의 일부 또는 애플리케이션 프로그램의 일부 또는 이들의 임의의 조합일 수도 있고, 이는 이러한 컴퓨터 또는 프로세서가 명시적으로 도시되든 아니든 간에 CPU 에 의해 실행될 수도 있다. 추가로, 여러 다른 주변 유닛들이 컴퓨터 플랫폼, 이를 테면, 추가적인 데이터 저장 유닛 및 프린팅 유닛에 접속될 수도 있다.
- [0142] 여기에 이용된 모든 예들 및 조건적인 언어는 당해 기술을 발전시키도록 발명자에 의해 기여된 개념들 및 실시 형태들의 원리들을 독자들이 이해하는데 있어 도움을 주는 교육적 목적으로 의도되며, 이러한 인용된 예들 및 조건들로 제한되지 않는 것으로 간주되어야 한다. 또한, 원리들, 양태들, 및 여러 실시형태들 뿐만 아니라 이들의 특정 예들을 인용하는 여기에서의 모든 설명들은 이들의 기능적 그리고 구조적 등가물 양쪽 모두를 포함하도록 의도된다. 추가로, 이러한 등가물들은 현재 알려진 등가물들 뿐만 아니라 미래 개발될 등가물들, 즉, 구조와 관련없이 동일한 기능을 수행하도록 개발된 임의의 엘리먼트들, 양쪽 모두를 포함하도록 의도된다.

도면

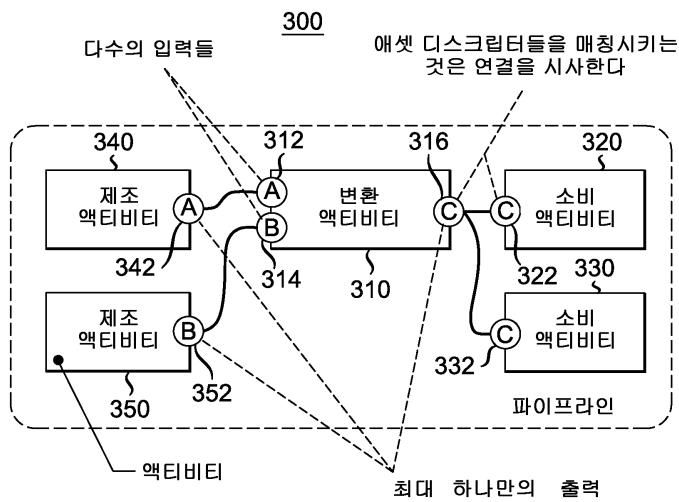
도면1



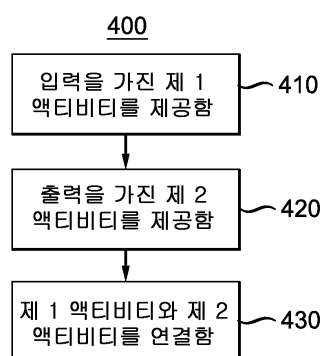
도면2



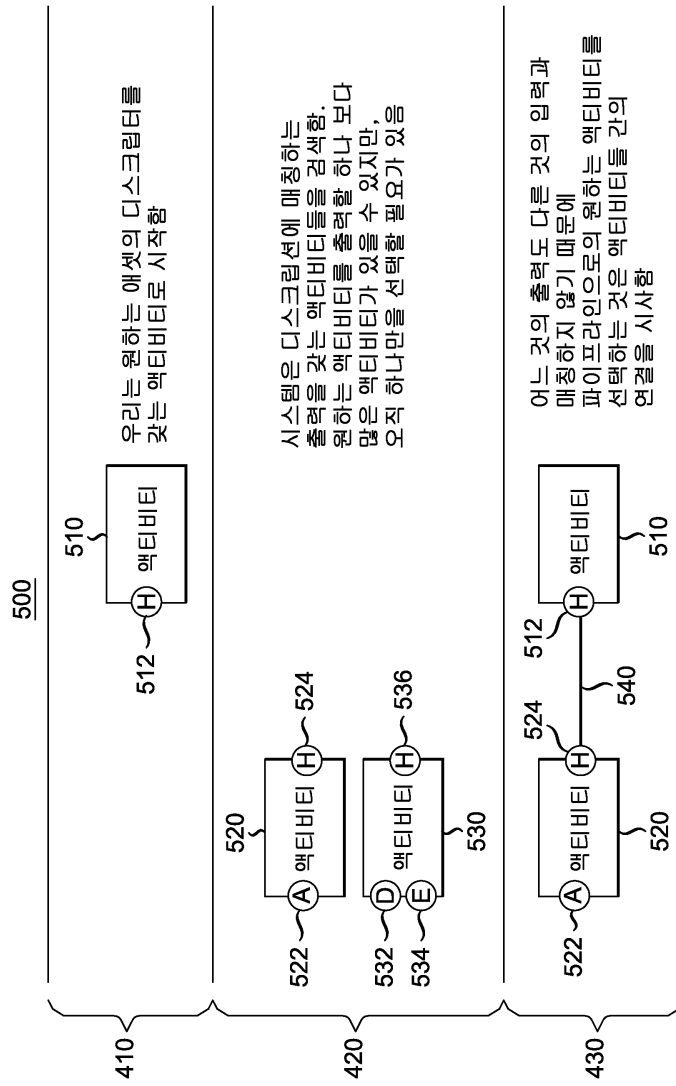
도면3



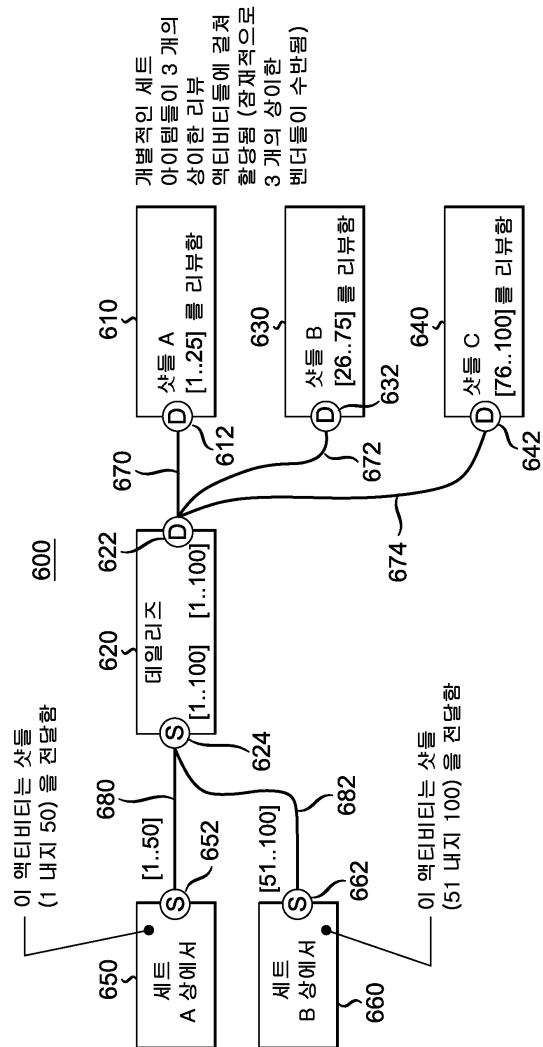
도면4



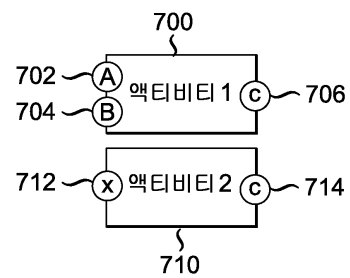
도면5



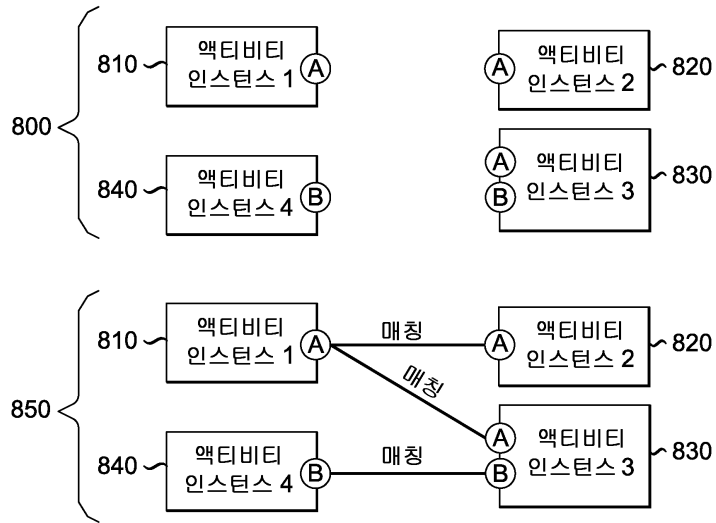
도면6



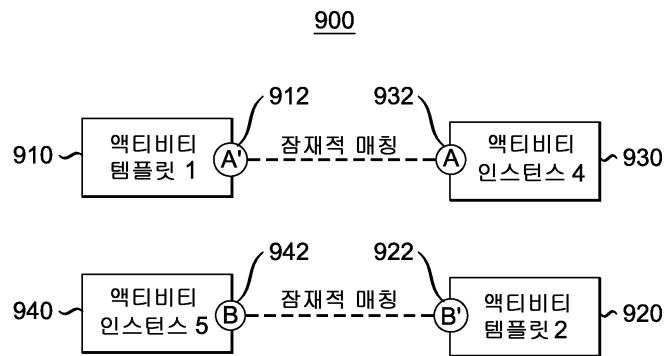
도면7



도면8



도면9

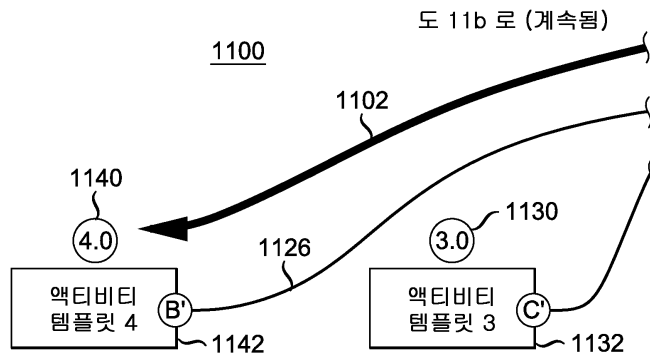


도면10

1000

원전하게 정의된 애셋 디스크립터	파라미터화된 애셋 디스크립터
{ Title:'The Hobbit', Version:'Trailer', Type:'Netflix Encoding' }	MATCHES 블랭크들이 와일드카드임
{ Title:'The Hobbit', Version:'Trailer', Type:'Netflix Encoding' }	NOT MATCHED 이름/값 페어들의 상이한 세트들 (엑스트라)
{ Title:'The Hobbit', Version:'Trailer', Type:'Netflix Encoding' }	MATCHES 정확하게 매칭하는 값들로 채워짐
{ Title:'The Hobbit', Version:'Trailer', Type:'SOMETHING ELSE' }	NOT MATCHED 정확하게 매칭하지 않는 값들로 채워짐

도면11a



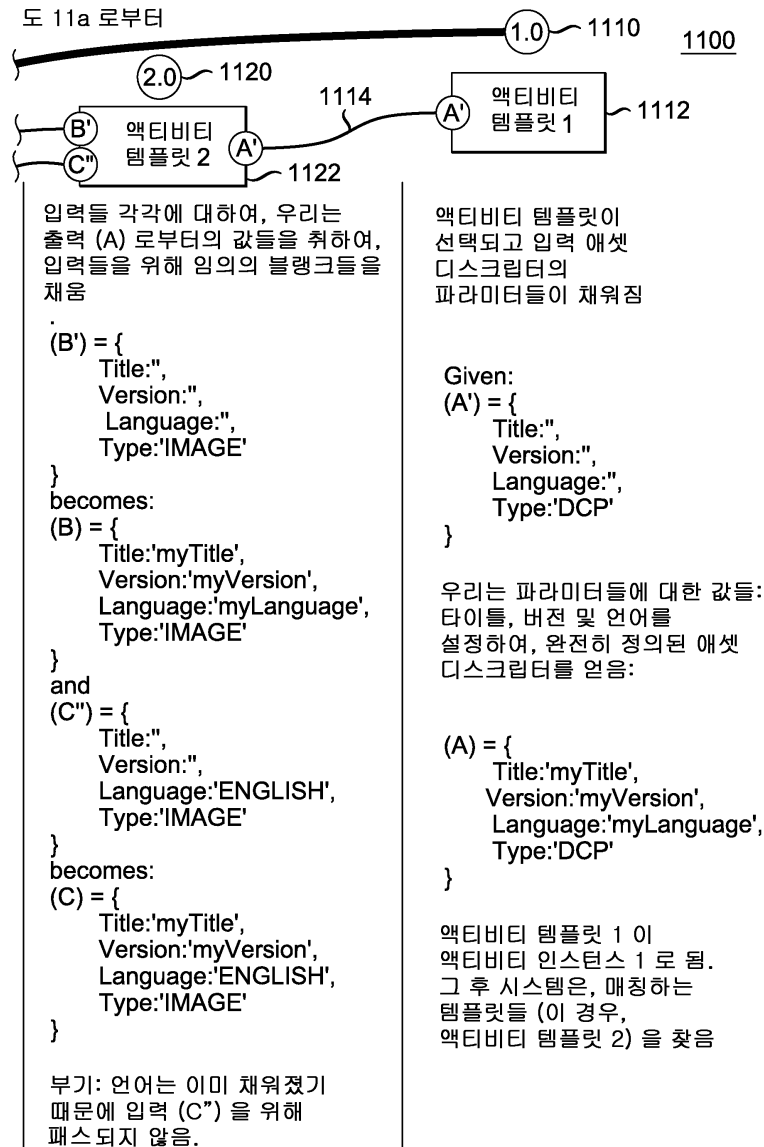
그리고 최종적으로 액티비티 템플릿 4 출력 (B') 은 액티비티 2 의 입력 (B) 에 의해 확립되는 파라미터들을 수용함

```
(B') = {
  Title:",
  Version:",
  Language:",
  Type:'IMAGE'
}
becomes:
(B) = {
  Title:'myTitle',
  Version:'myVersion',
  Language:'myLanguage',
  Type:'IMAGE'
}
```

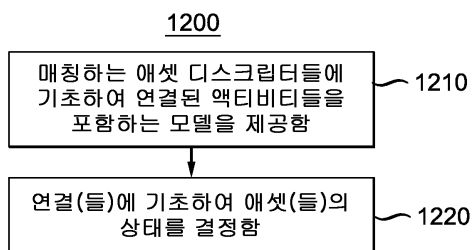
(C') 가 이전 단계에서 생성된 (C) 와 매칭하기 때문에 액티비티 템플릿 3 이 선택됨. 그러나, 패스된 언어는 (C') 에 의해 확립되는 하드코딩된 언어임

```
(C') = {
  Title:",
  Version:",
  Language:",
  Type:'IMAGE'
}
becomes:
(C) = {
  Title:'myTitle',
  Version:'myVersion',
  Language:'ENGLISH',
  Type:'IMAGE'
}
```

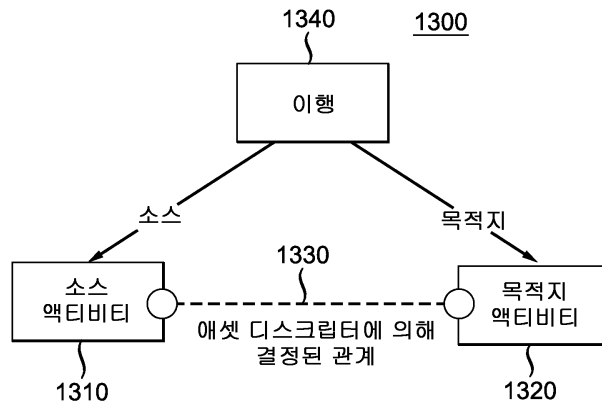
도면11b



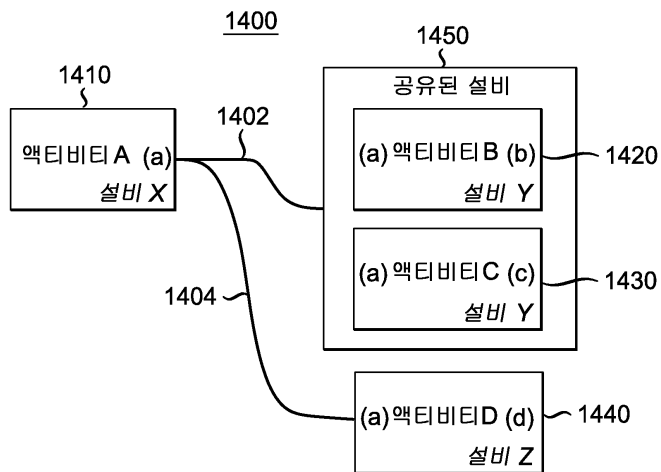
도면12



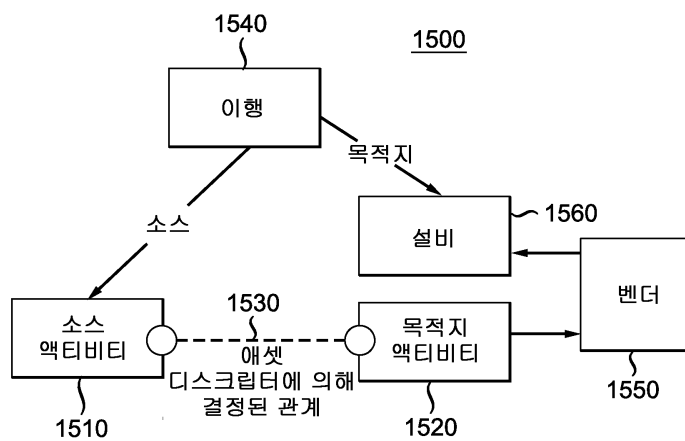
도면13



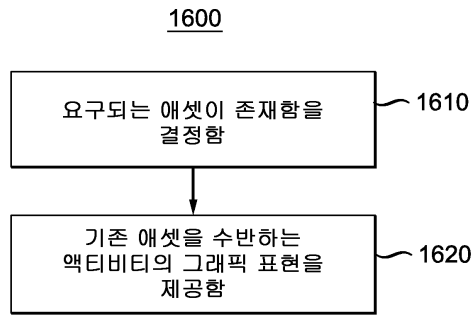
도면14



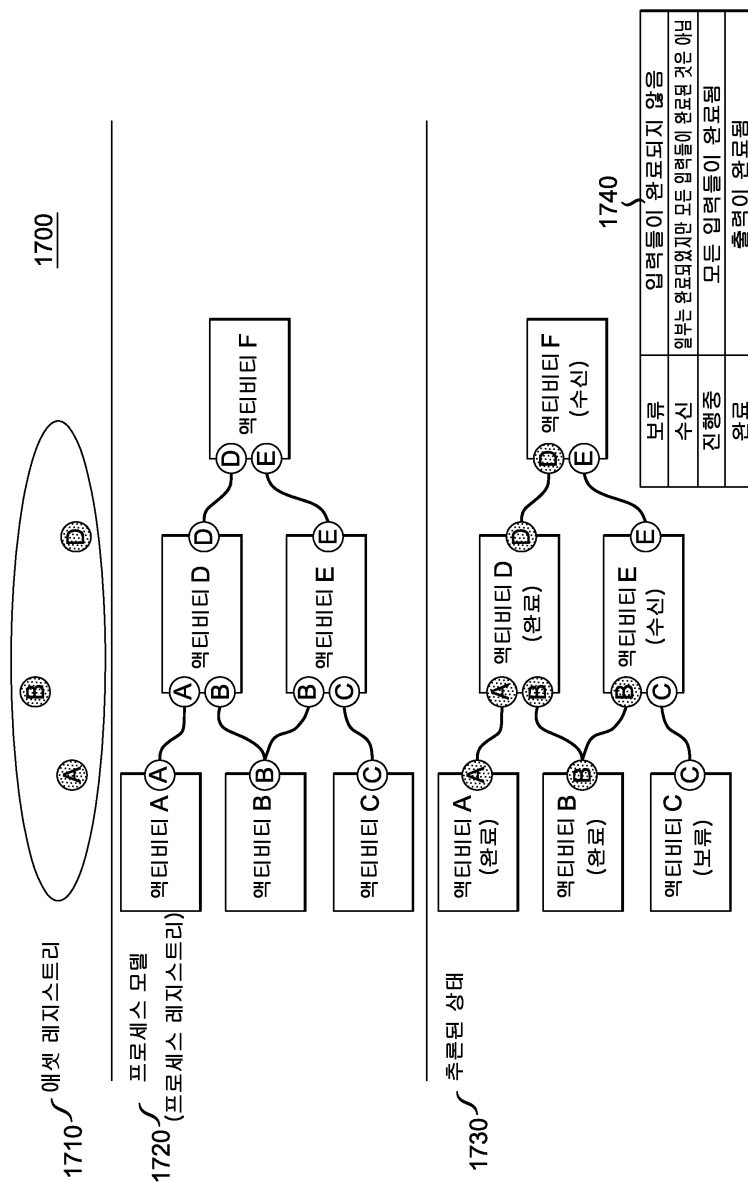
도면15



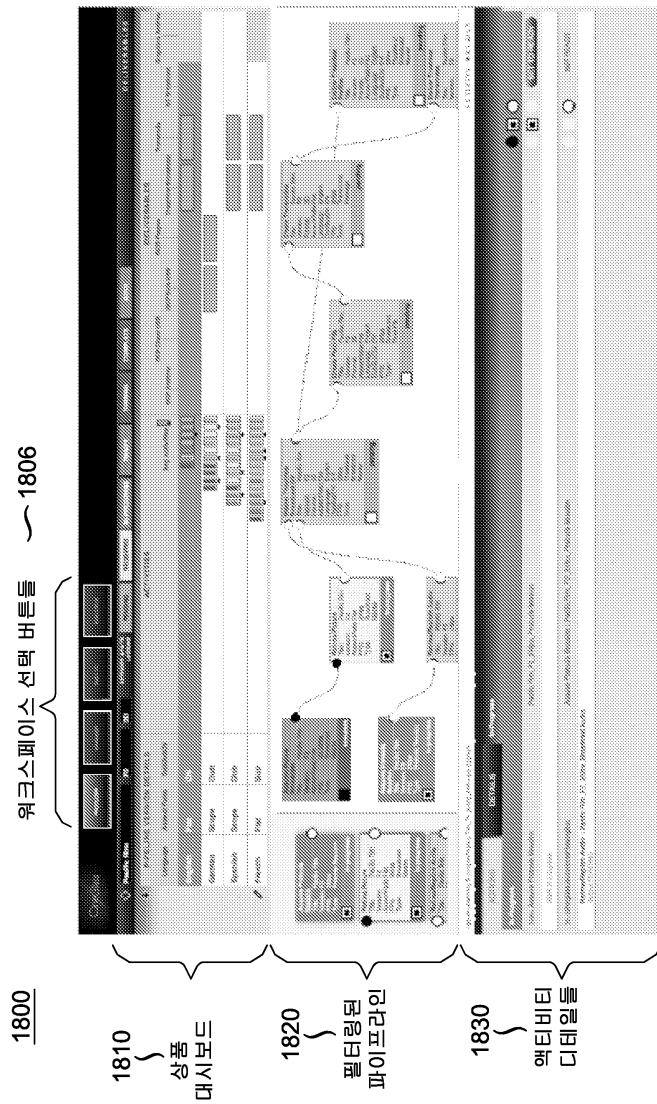
도면16



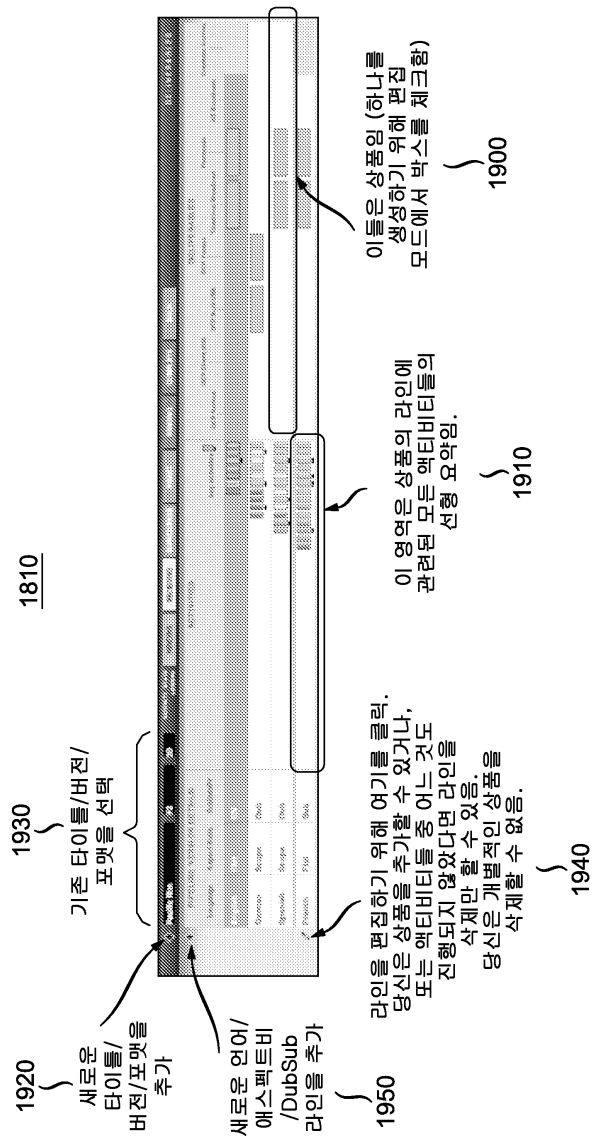
도면17



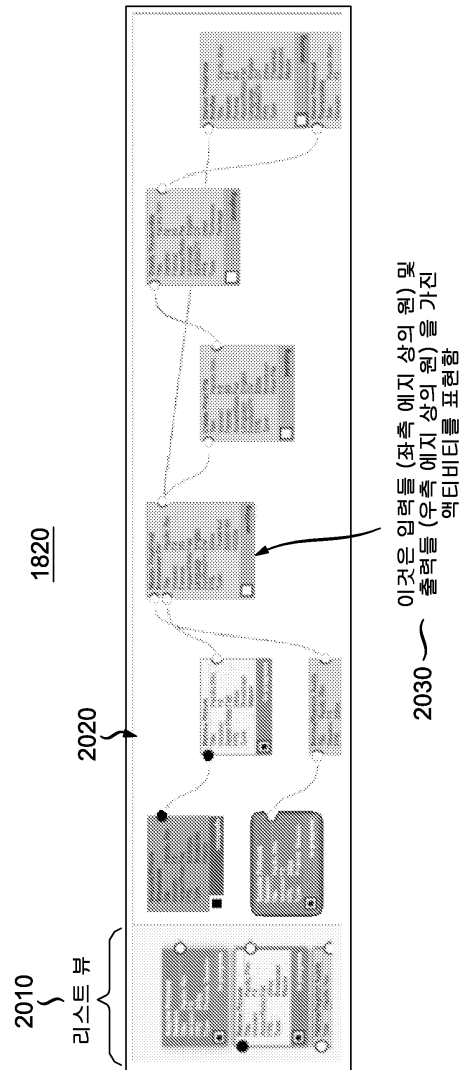
도면18



도면19



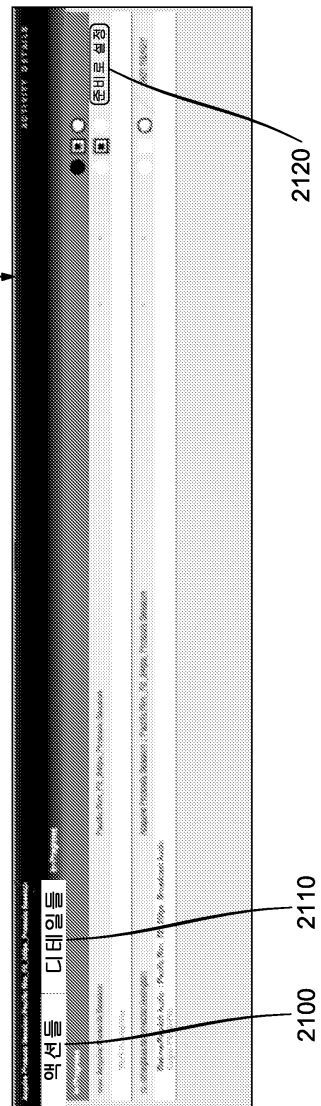
도면20



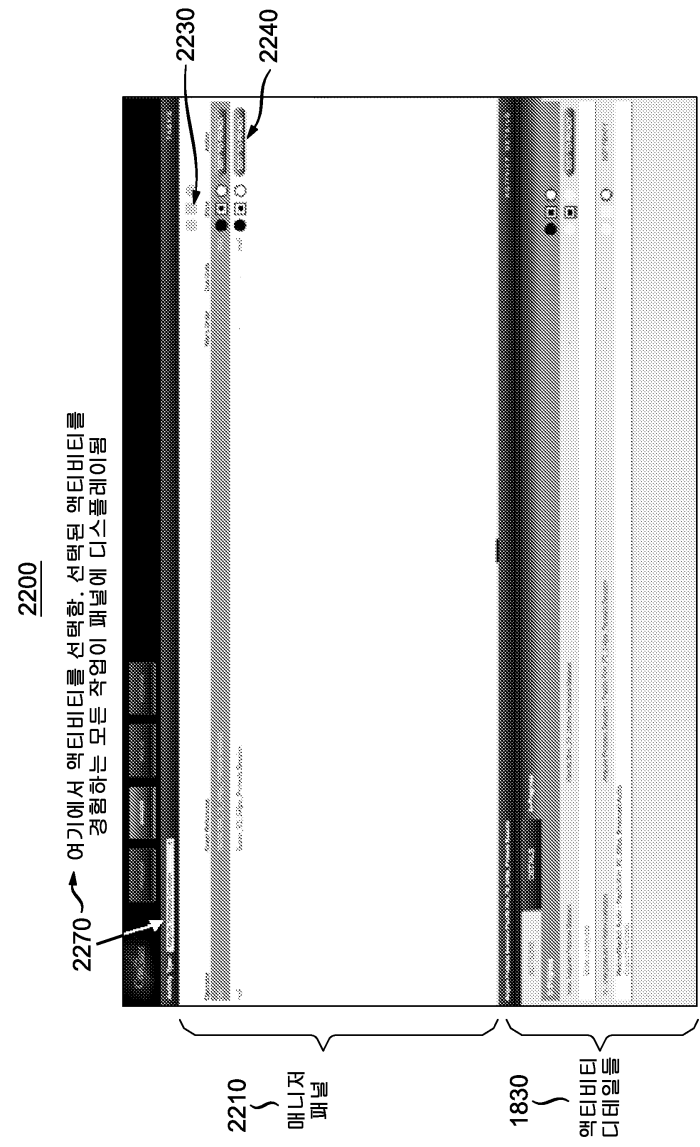
도면21

2130 부기: 액티비티가 완료되면, REVISE 버튼을
 눌러서 나태날 것임. 리바이스는당신이
 변경의 영향을 알고 나서 그 변경을
 재실행하게끔 커밋하는것을 허용함.

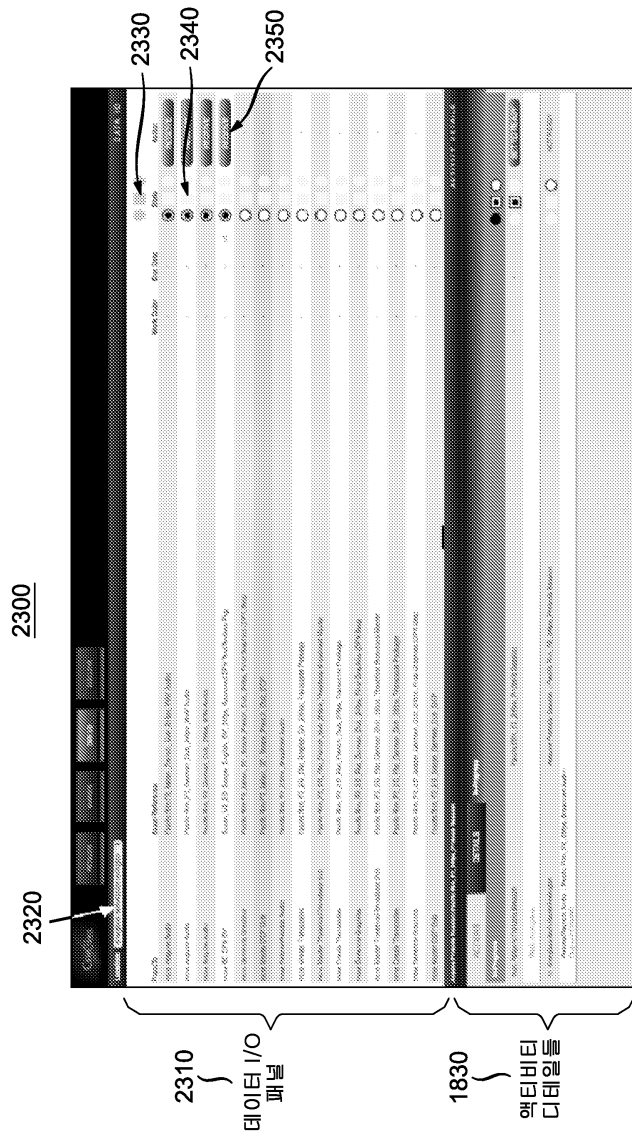
1830



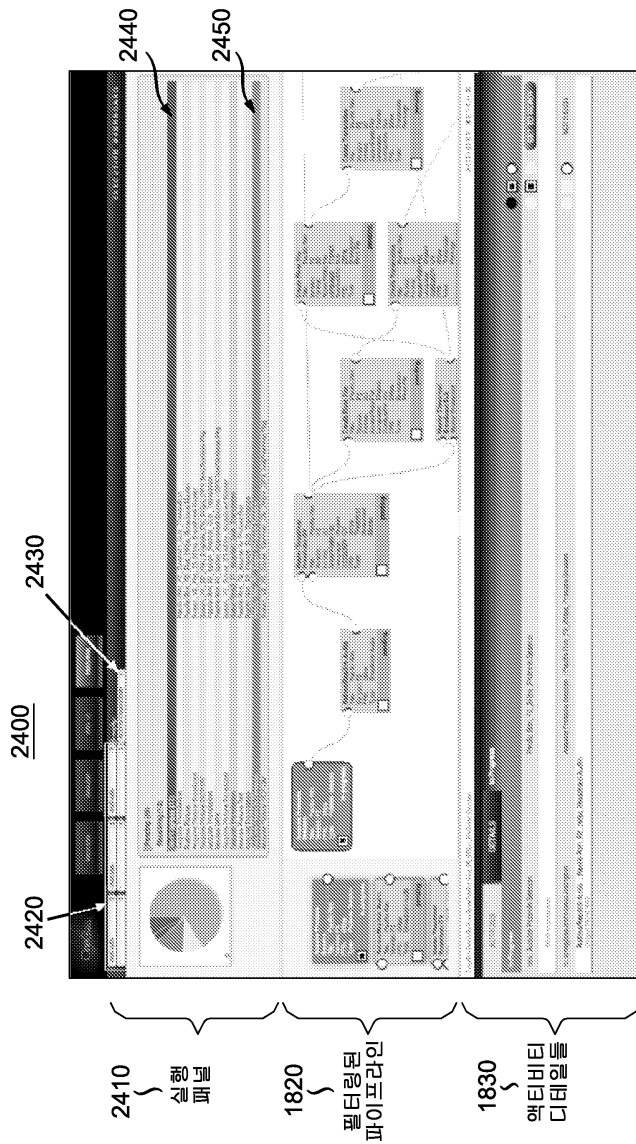
도면22



도면23



도면24



도면25

