



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2022년10월17일

(11) 등록번호 10-2455742

(24) 등록일자 2022년10월13일

(51) 국제특허분류(Int. Cl.)
G06Q 10/06 (2012.01) G06Q 10/10 (2022.01)

(52) CPC특허분류
G06Q 10/0633 (2013.01)
G06Q 10/06316 (2013.01)

(21) 출원번호 10-2020-7028125

(22) 출원일자(국제) 2020년09월23일

심사청구일자 2020년11월02일

(85) 번역문제출일자 2020년09월28일

(65) 공개번호 10-2022-0007494

(43) 공개일자 2022년01월18일

(86) 국제출원번호 PCT/US2020/052183

(87) 국제공개번호 WO 2022/010515

국제공개일자 2022년01월13일

(30) 우선권주장

16/924,499 2020년07월09일 미국(US)

16/941,484 2020년07월28일 미국(US)

(56) 선행기술조사문헌

JP2016215303 A*

(뒷면에 계속)

전체 청구항 수 : 총 19 항

심사관 : 심송학

(54) 발명의 명칭 로봇 프로세스 자동화를 위한 로봇 액세스 제어 및 거버넌스

(57) 요약

로봇 프로세스 자동화(RPA)를 위한 로봇 액세스 제어 및 관장이 개시된다. 작업흐름 분석기와 같은, RPA 설계자 애플리케이션의 코드 분석기는 RPA 설계자 애플리케이션을 위한 액세스 제어 및 관장 정책 규칙들을 관독할 수 있고, 액세스 제어 및 관장 정책 규칙들에 대하여 RPA 설계자 애플리케이션의 RPA 작업흐름의 활동들을 분석할 수 있다. RPA 작업흐름의 하나 이상의 분석된 활동들이 액세스 제어 및 관장 정책 규칙들을 위반할 때, 코드 분석기는 RPA 작업흐름이 액세스 제어 및 관장 정책 규칙들을 만족시킬 때까지, RPA 로봇의 생성 또는 RPA 작업흐름의 발행을 방지한다. RPA 작업흐름의 분석된 활동들이 모든 요구된 액세스 제어 및 관장 정책 규칙들을 준수할 때, RPA 설계자 애플리케이션은 RPA 작업흐름을 구현하는 RPA 로봇을 생성할 수 있거나, RPA 작업흐름을 발행할 수 있다.

(52) CPC특허분류

G06Q 10/0637 (2013.01)

G06Q 10/0639 (2013.01)

G06Q 10/10 (2021.08)

(56) 선행기술조사문헌

JP2019079364 A*

KR1020180058563 A*

KR1020180128727 A*

US20200067923 A1*

*는 심사관에 의하여 인용된 문헌

명세서

청구범위

청구항 1

로봇 프로세스 자동화(robotic process automation; RPA)를 위한 실행시간(runtime) 로봇 액세스 제어 및 관장(governance)을 수행하기 위한 컴퓨터-구현 방법(computer-implemented)에 있어서,

액세스 제어 및 관장 규칙에 대하여, RPA 로봇에 의해 수행되어야 할 하나 이상의 액션, 상기 RPA 로봇에 의해 요청된 정보, 또는 양자 모두를 체크하는 단계;

상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙에 기초하여, 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가될 때:

상기 하나 이상의 액션을 인가하거나 상기 요청된 정보를 상기 RPA 로봇에 제공하는 단계; 및

상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙에 기초하여, 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가되지 않을 때:

상기 RPA 로봇이 상기 하나 이상의 액션을 취하는 것 및 상기 요청된 정보를 획득하는 것 중 적어도 하나를 방지하는 단계를 포함하고,

상기 액세스 제어 및 관장 규칙은 하나 이상의 애플리케이션 제약 및 하나 이상의 유니버설 자원 로케이터(universal resource locator; URL) 제약 중 적어도 하나, 하나 이상의 패키지 제약, 하나 이상의 활동 제약, 하나 이상의 활동 속성 요건, 또는 그 조합을 포함하는, 컴퓨터-구현 방법.

청구항 2

제 1 항에 있어서,

상기 RPA 로봇이 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가되지 않을 때, 상기 방법은,

상기 하나 이상의 액션 및 상기 정보에 대한 액세스 중 적어도 하나가 허가되지 않는다는 것을 표시하는 통지를 상기 RPA 로봇 또는 클라이언트-측 애플리케이션으로 전송하는 단계를 더 포함하는, 컴퓨터-구현 방법.

청구항 3

제 1 항에 있어서,

상기 RPA 로봇이 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가되지 않을 때, 상기 방법은,

상기 RPA 로봇의 실행을 자동적으로 종료시키는 단계를 더 포함하는, 컴퓨터-구현 방법.

청구항 4

제 1 항에 있어서,

상기 방법은 클라이언트-측 유효성확인 애플리케이션(validation application)에 의해 수행되는 것인, 컴퓨터-구현 방법.

청구항 5

제 1 항에 있어서,

상기 방법은 서버-측 유효성확인 애플리케이션에 의해 수행되는 것인, 컴퓨터-구현 방법.

청구항 6

제 1 항에 있어서,

컴퓨팅 시스템이 서버-측 지휘자 애플리케이션에 접속할 때, 상기 액세스 제어 및 관장 규칙을 상기 RPA 로봇과 연관된 컴퓨팅 시스템으로 자동적으로 전송하는 단계를 더 포함하는, 컴퓨터-구현 방법.

청구항 7

제 1 항에 있어서,

상기 하나 이상의 액션 중의 적어도 하나는 상기 RPA 로봇에 의해서가 아니라, 인간 사용자를 위하여 허가되는, 컴퓨터-구현 방법.

청구항 8

제 1 항에 있어서,

상기 RPA 로봇이 하나 이상의 레거시 시스템이 하나 이상의 동작을 수행하기 위한 적어도 시간의 주기 동안에 일시정지되는 것을 검증하는 단계; 및

상기 RPA 로봇이 적어도 상기 시간의 주기 동안에 일시정지되지 않을 때:

상기 시간의 주기가 만료될 때까지, 상기 RPA 로봇에 의해 요청된 정보의 획득 또는 상기 RPA 로봇으로부터의 새로운 정보 요청의 수락을 지연시키는 단계를 더 포함하는, 컴퓨터-구현 방법.

청구항 9

제 1 항에 있어서,

상기 액세스 제어 및 관장 규칙의 실행시간 집행은 상기 액세스 제어 및 관장 규칙의 설계 시간 집행에 추가적으로 수행되는 것인, 컴퓨터-구현 방법.

청구항 10

제 1 항에 있어서,

상기 RPA 로봇은 상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙을 획득하거나 상기 액세스 제어 및 관장 규칙을 준수하여 동작하는 것을 강제하는 자동적으로 삽입된 코드를 포함하는, 컴퓨터-구현 방법.

청구항 11

삭제

청구항 12

로봇 프로세스 자동화(robotic process automation; RPA)를 위한 실행시간 로봇 액세스 제어 및 관장을 수행하기 위한 컴퓨터 프로그램을 저장한 비-일시적 컴퓨터-판독가능 매체에 있어서,

상기 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금,

액세스 제어 및 관장 규칙에 대하여, RPA 로봇에 의해 수행되어야 할 하나 이상의 액션, 상기 RPA 로봇에 의해 요청된 정보, 또는 양자 모두를 체크하게 하고;

상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙에 기초하여, 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가되지 않을 때:

상기 RPA 로봇이 상기 하나 이상의 액션을 취하는 것 및 상기 요청된 정보를 획득하는 것 중 적어도 하나를 방지하게 하도록 구성되고,

상기 액세스 제어 및 관장 규칙은 하나 이상의 애플리케이션 제약 및 하나 이상의 유니버설 자원 로케이터(universal resource locator; URL) 제약 중 적어도 하나, 하나 이상의 패키지 제약, 하나 이상의 활동 제약, 하나 이상의 활동 속성 요건, 또는 그 조합을 포함하는, 비-일시적 컴퓨터-판독가능 매체.

청구항 13

제 12 항에 있어서,

상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙에 기초하여, 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가될 때, 상기 컴퓨터 프로그램은 또한, 상기 적어도 하나의 프로세서로 하여금,

상기 하나 이상의 액션을 인가하게 하거나 상기 요청된 정보를 상기 RPA 로봇에 제공하게 하도록 구성되는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 14

제 12 항에 있어서,

상기 RPA 로봇이 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가되지 않을 때, 상기 컴퓨터 프로그램은 또한, 상기 적어도 하나의 프로세서로 하여금,

상기 하나 이상의 액션 및 상기 정보에 대한 액세스 중 적어도 하나가 허가되지 않는다는 것을 표시하는 통지를 상기 RPA 로봇 또는 클라이언트-측 애플리케이션으로 전송하게 하도록 구성되는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 15

제 12 항에 있어서,

상기 하나 이상의 액션 중의 적어도 하나는 상기 RPA 로봇에 의해서가 아니라, 인간 사용자를 위하여 허가되는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 16

제 12 항에 있어서,

상기 컴퓨터 프로그램은 또한, 상기 적어도 하나의 프로세서로 하여금,

상기 RPA 로봇이 하나 이상의 레거시 시스템이 하나 이상의 동작을 수행하기 위한 적어도 시간의 주기 동안에 일시정지되는 것을 검증하게 하고;

상기 RPA 로봇이 적어도 상기 시간의 주기 동안에 일시정지되지 않을 때:

상기 시간의 주기가 만료될 때까지, 상기 RPA 로봇에 의해 요청된 정보의 획득 또는 상기 RPA 로봇으로부터의 새로운 정보 요청의 수락을 지연시키도록 구성되는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 17

제 12 항에 있어서,

상기 RPA 로봇은 상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙을 획득하거나 상기 액세스 제어 및 관장 규칙을 준수하여 동작하는 것을 강제하는 자동적으로 삽입된 코드를 포함하는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 18

로봇 프로세스 자동화(robotic process automation; RPA)를 위한 실행시간 로봇 액세스 제어 및 관장을 수행하기 위한 컴퓨터 프로그램을 저장한 비-일시적 컴퓨터-판독가능 매체에 있어서,

상기 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금,

액세스 제어 및 관장 규칙에 대하여, RPA 로봇에 의해 수행되어야 할 하나 이상의 액션, 상기 RPA 로봇에 의해 요청된 정보, 또는 양자 모두를 체크하게 하고;

상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙에 기초하여, 상기 하나 이상의 액션을 수행하거나, 상기 요청된 정보를 수신하거나, 또는 양자 모두를 행하도록 허가되지 않을 때:

상기 RPA 로봇이 상기 하나 이상의 액션을 취하는 것 및 상기 요청된 정보를 획득하는 것 중 적어도 하나를 방지하게 하고,

상기 하나 이상의 액션 및 상기 정보에 대한 액세스 중 적어도 하나가 허가되지 않는다는 것을 표시하는 통지를 상기 RPA 로봇 또는 클라이언트-측 애플리케이션으로 전송하게 하거나, 상기 RPA 로봇의 실행을 자동적으로 종료시키거나, 또는 양자 모두를 행하게 하도록 구성되고,

상기 액세스 제어 및 관장 규칙은 하나 이상의 애플리케이션 제약 및 하나 이상의 유니버설 자원 로케이터(universal resource locator; URL) 제약 중 적어도 하나, 하나 이상의 패키지 제약, 하나 이상의 활동 제약, 하나 이상의 활동 속성 요건, 또는 그 조합을 포함하는, 비-일시적 컴퓨터-판독가능 매체.

청구항 19

제 18 항에 있어서,

상기 컴퓨터 프로그램은 또한, 상기 적어도 하나의 프로세서로 하여금,

상기 RPA 로봇이 하나 이상의 레거시 시스템이 하나 이상의 동작을 수행하기 위한 적어도 시간의 주기 동안에 일시정지되는 것을 검증하게 하고;

상기 RPA 로봇이 적어도 상기 시간의 주기 동안에 일시정지되지 않을 때:

상기 시간의 주기가 만료될 때까지, 상기 RPA 로봇에 의해 요청된 정보의 획득 또는 상기 RPA 로봇으로부터의 새로운 정보 요청의 수락을 지연시키도록 구성되는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 20

제 18 항에 있어서,

상기 RPA 로봇은 상기 RPA 로봇이 상기 액세스 제어 및 관장 규칙을 획득하거나 상기 액세스 제어 및 관장 규칙을 준수하여 동작하는 것을 강제하는 자동적으로 삽입된 코드를 포함하는 것인, 비-일시적 컴퓨터-판독가능 매체.

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

발명의 설명

기술 분야

[0001] 관련 출원에 대한 상호-참조

[0002] 이 출원은 2020년 7월 9일자로 출원된 미국 정규 특허 출원 제16/924,499호, 및 그 연속 출원인 2020년 7월 28일자로 출원된 미국 정규 특허 출원 제16/941,484호의 이익을 주장한다. 이의 더 이전에 출원된 출원의 발명 요지는 이로써, 그 전체적으로 참조에 의해 편입된다.

[0003] 본 발명은 일반적으로 로봇 프로세스 자동화(robotic process automation; RPA)에 관한 것으로, 더 구체적으로, RPA를 위한 로봇 액세스 제어 및 관장(governance)에 관한 것이다.

배경 기술

[0004] 컴퓨팅 시스템들의 사용자들에 의해 취해진 어떤 액션(action)들은 E.U. 일반 데이터 보호 규정(General Data Protection Regulation; GDPR), U.S. 건강 보험 이동성 및 책무성 법령(Health Insurance Portability and Accountability Act; HIPAA), 제3자 서비스 약관(third party terms of service) 등과 같은, U.S. 및 다른 국가들에서의 법률들 또는 협정들을 위반할 수 있다. 이 법률들 또는 협정들의 위반에 대한 처벌들은 거대할 수 있다. 예를 들어, GDPR에 의해, 더 낮은-레벨의 위반들은 1000만 유로 또는 위반자의 전세계 연간 수입(수익이 아님)의 2 % 중의 더 높은 벌금을 받을 수 있다. 더 심각한 위반은 2000만 유로 또는 위반자의 연간 수입의 4 퍼센트 중의 더 높은 벌금으로 귀착될 수 있다. 이 벌금들은 회사에 금융적으로 손상시킬 수 있거나 심지어 치명적일 수 있다. RPA는 액세스 제어 및 관장에 대한 고유한 도전을 제시한다. 따라서, RPA 액세스 제어 및 관장에 대한 개선된 접근법들이 유익할 수 있다.

발명의 내용

[0005] 본 발명의 어떤 실시예들은 현재의 RPA 기술들에 의해 아직 완전히 식별되거나, 인식되거나, 해결되지 않은 종래 기술에서의 문제들 및 필요성들에 대한 해결책들을 제공할 수 있다. 예를 들어, 본 발명의 일부 실시예들은 RPA를 위한 로봇 액세스 제어 및 관장에 속한다.

[0006] 실시예에서, RPA를 위한 로봇 액세스 제어 및 관장을 수행하기 위한 컴퓨터 프로그램은 비-일시적 컴퓨터-판독 가능 매체 상에서 구체화된다. 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금, RPA 설계자 애플리케이션(designer application)을 위한 액세스 제어 및 관장 정책 규칙들을 관독하게 하고, 액세스 제어 및 관장 정책 규칙들에 대하여 RPA 설계자 애플리케이션의 RPA 작업흐름(workflow)의 활동들을 분석하게 하도록 구성된다. RPA 작업흐름의 하나 이상의 분석된 활동들이 액세스 제어 및 관장 정책 규칙들을 위반할 때, 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금, RPA 작업흐름이 액세스 제어 및 관장 정책 규칙들을 만족시킬 때까지 RPA 로봇의 생성 또는 RPA 작업흐름의 발행(publication)을 방지하게 하도록 구성된다. RPA 작업흐름의 분석된 활동들이 모든 요구된 액세스 제어 및 관장 정책 규칙들을 준수할 때, 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금, RPA 작업흐름을 구현하는 RPA 로봇을 생성하거나 RPA 작업흐름을 발행하게 하도록 구성된다.

[0007] 또 다른 실시예에서, RPA를 위한 로봇 액세스 제어 및 관장을 수행하기 위한 컴퓨터-구현 방법은 코드 분석기(code analyzer)에 의해, 액세스 제어 및 관장 정책 규칙들에 대하여 RPA 설계자 애플리케이션의 RPA 작업흐름의 활동들을 분석하는 단계를 포함한다. RPA 작업흐름의 하나 이상의 분석된 활동들이 액세스 제어 및 관장 정책 규칙들을 위반할 때, 컴퓨터-구현 방법은 코드 분석기에 의해, RPA 작업흐름이 액세스 제어 및 관장 정책 규칙들을 만족시킬 때까지, RPA 로봇의 생성 또는 RPA 작업흐름의 발행을 방지하는 단계를 포함한다.

[0008] 또 다른 실시예에서, RPA를 위한 로봇 액세스 제어 및 관장을 수행하기 위한 컴퓨터 프로그램은 비-일시적 컴퓨터-판독가능 매체 상에서 구체화된다. 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금, 컴퓨팅 시스템의 레지스트리 엔트리(registry entry)로부터 액세스 제어 및 관장 정책 규칙들을 포함하는 파일에 대한 링크(link)를 결정하게 하고, 결정된 링크를 이용하여 파일을 다운로드하게 하거나 지휘자 애플리케이션(conductor application)으로부터 관장 정책 규칙들을 다운로드하게 하도록 구성된다. 컴퓨터 프로그램은 또한, 적어도 하나의 프로세서로 하여금, 다운로드된 파일로부터 RPA 설계자 애플리케이션을 위한 액세스 제어 및 관장 정책 규칙들을 관독하게 하고, 액세스 제어 및 관장 정책 규칙들에 대하여 RPA 설계자 애플리케이션의 RPA 작업흐름의

활동들을 분석하게 하도록 구성된다. RPA 작업흐름의 하나 이상의 분석된 활동들이 액세스 제어 및 관장 정책 규칙들을 위반할 때, 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금, RPA 작업흐름이 액세스 제어 및 관장 정책 규칙들을 만족시킬 때까지 RPA 로봇의 생성 또는 RPA 작업흐름의 발행을 방지하게 하도록 구성된다. RPA 작업흐름의 분석된 활동들이 모든 요구된 액세스 제어 및 관장 정책 규칙들을 준수할 때, 컴퓨터 프로그램은, 적어도 하나의 프로세서로 하여금, RPA 작업흐름을 구현하는 RPA 로봇을 생성하거나 RPA 작업흐름을 발행하게 하도록 구성된다. 액세스 제어 및 관장 정책 규칙들은 하나 이상의 애플리케이션 및/또는 유니버설 자원 로케이터(universal resource locator; URL) 제약, 하나 이상의 패키지 제약, 하나 이상의 활동 제약, 하나 이상의 활동 속성 요건(activity property requirement)들, 또는 그 조합을 포함한다.

도면의 간단한 설명

[0009]

발명의 어떤 실시예들의 장점들이 용이하게 이해되도록 하기 위하여, 위에서 간략하게 설명된 발명의 더 특정한 설명은 첨부된 도면들에서 예시되는 특정 실시예들을 참조하게 될 것이다. 이 도면들은 발명의 오직 전형적인 실시예들을 도시하고, 그러므로, 그 범위의 제한인 것으로 고려되지 않아야 한다는 것이 이해되어야 하지만, 발명은 동반 도면들의 이용을 통해 추가적인 구체성 및 상세함으로 설명되고 기술될 것이다:

도 1은 본 발명의 실시예에 따라, 로봇 프로세스 자동화(RPA) 시스템을 예시하는 아키텍처 다이어그램이다.

도 2는 본 발명의 실시예에 따라, 전개된 RPA 시스템을 예시하는 아키텍처 다이어그램이다.

도 3은 본 발명의 실시예에 따라, 설계자, 활동들, 및 구동기들 사이의 관계를 예시하는 아키텍처 다이어그램이다.

도 4는 본 발명의 실시예에 따라, RPA 시스템을 예시하는 아키텍처 다이어그램이다.

도 5는 본 발명의 실시예에 따라, RPA를 위한 로봇 액세스 제어 및 관장을 수행하도록 구성된 컴퓨팅 시스템을 예시하는 아키텍처 다이어그램이다.

도 6은 본 발명의 실시예에 따라, UiPath Studio™를 위한 Windows® 레지스트리 편집기(Registry Editor)의 폴더(folder)의 뷰(view)를 예시하는 스크린샷이다.

도 7은 본 발명의 실시예에 따라, 파일 SampleGovernanceRules.dll이 UiPath Studio™를 위한 규칙들 폴더에서 설치되는 Windows Explorer® UI의 부분을 예시하는 스크린샷이다.

도 8은 본 발명의 실시예에 따라, 패키지 관리 인터페이스(package management interface)를 예시하는 스크린샷이다.

도 9는 본 발명의 실시예에 따라, 디스에이블된 피드백 전송 아이콘(send feedback icon)을 갖는 RPA 설계자 애플리케이션의 상부 우측 부분을 예시하는 스크린샷이다.

도 10a 내지 도 10d는 본 발명의 실시예에 따라, 작업흐름 분석기 설정들 인터페이스를 예시하는 스크린샷들이다.

도 11은 본 발명의 실시예에 따라, RPA 설계자 애플리케이션 설정들 인터페이스를 예시하는 스크린샷이다.

도 12a 및 도 12b는 본 발명의 실시예에 따라, RPA 작업흐름을 예시하는 스크린샷들이다.

도 13은 본 발명의 실시예에 따라, 에러 리스트 인터페이스(error list interface)를 예시한다.

도 14는 본 발명의 실시예에 따라, 작업흐름 분석기 인터페이스를 예시하는 스크린샷이다.

도 15는 본 발명의 실시예에 따라, 로봇 실행 시에 RPA를 위한 액세스 제어 및 관장을 수행하기 위한 시스템을 예시하는 아키텍처 다이어그램이다.

도 16은 본 발명의 실시예에 따라, RPA 설계자 애플리케이션을 위한 로봇 액세스 제어 및 관장을 수행하기 위한 프로세스를 예시하는 플로우차트이다.

도 17은 본 발명의 실시예에 따라, 실행 RPA 로봇에 대하여 RPA를 위한 로봇 액세스 제어 및 관장을 수행하기 위한 프로세스를 예시하는 플로우차트이다.

이와 다르게 표시되지 않으면, 유사한 참조 문자들은 첨부된 도면들 전반에 걸쳐 대응하는 특징들을 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0010] 일부 실시예들은 RPA를 위한 로봇 액세스 제어 및 관장에 속한다. RPA 관장은 사용자들이 GDPR, HIPAA, 제3자 애플리케이션 서비스 약관들 등을 위반하는 것과 같은, 조직(organization)을 훼손시킬 수 있는 액션들을 취할 수 있는 RPA 로봇들을 개발하는 것을 방지하기 위하여, 정책들을 가동하도록 하기 위한 조직들을 위한 능력이다. 개발자들은 그 자동화들을 수행하면서, 개인정보(privacy) 법률들, 서비스 약관들 등을 위반하는 RPA 로봇들을 생성할 수 있으므로, 일부 실시예들은 로봇 및/또는 로봇 설계 애플리케이션 레벨에서 액세스 제어 및 관장 제약들을 구현한다. 이것은 RPA 개발자들이 보안성 위험들을 도입할 수 있거나 정책들, 규정들, 개인정보 법률들, 및/또는 개인정보 정책들을 위반하는 방식으로 작동할 수 있는 미승인된 소프트웨어 라이브러리들에 대한 종속성들을 취하는 것을 방지함으로써, 일부 실시예들에서, 보안성 및 준수성의 추가된 레벨을 RPA 프로세스 개발 라이프라인에 제공할 수 있다.
- [0011] 일부 실시예들은 RPA 로봇들이 로봇에 의한 것이 아니라 사용자에 의해 허용되는 액션들을 취하는 것을 방지한다. 전통적으로, 조직들은 그 시스템들에 대한 액세스 제어를 통해 사용자가 행할 수 있고 행할 수 없는 것을 제어할 수 있다. 그러나, RPA 및 특히 유인 자동화(attended automation)의 맥락에서, 이 제어는 고장날 수 있다. 로봇은 사용자 대신에 태스크(task)들을 수행하고, 이것은 백엔드 시스템(backend system)이 로봇이 시스템을 액세스하고 있는 것과 사용자를 구별할 수 없다는 것을 의미한다.
- [0012] 이것은 몇몇 이유들로 문제가 있을 수 있다. 예를 들어, 백엔드 시스템들은 사용자가 오용될 수 있는 어떤 데이터를 액세스하는 것을 허용할 수 있다. 그러나, 양호한 시스템 설계는 전형적으로, 사용자들이 하지 않아야 하는 어떤 것을 행하는 것이 느리고 지루하므로, 사용자가 임의의 스케일(scale)에서 쟁점들을 야기시키는 것을 어렵게 한다. 대조적으로, RPA 로봇이 이용될 경우에, 로봇은 그 실행의 속력으로 인해 데이터 오용 또는 정책 위반의 스케일을 대폭 증가시킬 수 있다.
- [0013] 다음의 예들을 고려한다. 조직은 사용자가 "나에게 접촉하지 마세요(do not contact me)" 옵션을 선택하였는지 여부를 체크함으로써, 그리고 개인정보 정책들에 대한 링크들 및 "그들을 잊기(forget them)" 위한 조직에 대하여 요청하기 위한 명령들을 제공함으로써, GDPR의 "옵트 아웃(opt out)" 및 "나를 잊기(forget me)" 정책들 및 다른 사용자 보호 법률들과 같은 다양한 개인정보 옵션들을 완전히 지원하는, 고객들에게 접촉하기 위한 고객 관계 관리(customer relationship management; CRM) 시스템을 이용한다. 조직은 CRM 시스템에 대한 액세스를 피고용자들의 넓은 부분에게 부여하도록 선택하였지만, 시스템에서 이메일(email)들을 전송하기 위한 허가들을 선택된 약간의 피고용자들에게 오직 부여하도록 선택하였다.
- [0014] 그러나, 합법적인 비즈니스 목적들은 사용자들의 이 더 넓은 세트가 이메일 주소를 포함하는 고객 프로파일들을 액세스할 것을 그러나 요구한다(예컨대, 피고용자는 고객과 함께 작업할 때에 시스템에서의 고객들 이메일 주소를 포함하는 오정보(misinformation)를 생성하거나 편집할 수 있음). CRM 시스템을 이용하지 않으면서 고객에게 이메일하는 것은 조직 정책 위반이지만, 일부 피고용자들은 일부 정보를 학습할 것을 기대하고 있고, 그 허가들을 얻거나 적절하게 허가된 피고용자가 그 대신에 그렇게 행하게 하기 위하여 다양한 승인 채널들을 대기하는 것을 원하지 않는다. RPA 로봇이 없다면, 피고용자들은 이 정책을 여전히 위반할 수 있는데, 그것은 시스템으로부터 이메일 주소들을 하나씩 수동적으로 복사하고 붙여넣는 것이 지루하기 때문이다. 피고용자들이 이 정책을 위반하더라도, 위반의 스케일은 작을 것이고, 그러므로, 임의의 위반이 발생할 경우에 조직에 대한 위험을 완화시킬 것이다. 그러나, RPA 로봇은 이메일들을 생성하고 전송하기 위하여 인간보다 많은 자릿수만큼 더 빠리(예컨대, 수 천배 더 빠리, 수 만배 더 빠리, 수십 만배 더 빠리, 수 백만배 더 빠리 등) 이 데이터를 추출할 수 있고, 따라서, 사용자가 정책을 위반하는 5 내지 10 개의 이메일들을 수동적으로 전송할 수 있고 고객에 의한 GDPR "옵트 아웃"을 위반할 위험에 이르기까지 회사를 개발할 경우에, 사용자가 이것을 위한 로봇을 생성하면, 로봇은 큰 수의 위반 이메일들을 전송할 수 있다.
- [0015] 또 다른 예로서, HIPAA 호환 방식으로 모든 데이터를 전송하기 위하여 주의깊게 보호된 시스템들을 이용하는 건강관리 조직(healthcare organization)을 고려한다. 피고용자들은 적절한 애플리케이션(들)의 맥락에서 이 데이터를 합법적으로 액세스할 수 있고 이 데이터로 작업할 수 있다. 시스템의 비효율에 실망하는 피고용자는 환자 데이터를 그 또는 그녀의 데스크톱 상의 비보호된 Excel® 스프레드시트(spreadsheet)로 추출하는 RPA 로봇을 기록하는 것을 원할 수 있다. 이것은 이제 HIPAA 위반이고, 피고용자의 컴퓨터가 도난되고 액세스되는 것과 같은 어떤 일이 그 데이터에 대해 발생할 경우에, 이것은 조직을 상당한 금융적 및 법률적 책임으로까지 개방할 수 있다.

- [0016] 또 다른 예로서, 제3자 애플리케이션 또는 서비스를 위한 서비스 약관들이 애플리케이션 또는 서비스로 작업하기 위하여 자동화를 이용하는 것을 금지하는 경우를 고려한다. 피고용자가 이 서비스 약관들에 대하여 작업하는 RPA 로봇을 기록하는 것을 선택할 경우에, 회사는 법률적으로 책임이 있을 수 있다.
- [0017] 이 문제들 및 다른 것들은 RPA 로봇들을 채용하는 조직들에 실질적인 위험들을 제시할 수 있다. 그러나, 일부 실시예들은 개발자들이 미승인된 라이브러리 및 활동 참조들을 이용하는 것을 방지함으로써, 개발자들로부터 보안성 및 준수성 위험들을 감소시키거나 제거한다. RPA 개발자는 (예컨대, UiPath Studio™를 이용하는) 훈련된 RPA 개발자 또는 (예컨대, UiPath StudioX™를 이용하는) 비훈련된 시민 개발자이든지 간에, 안전하지 않은 것으로 고려되거나 정책/단속 규칙들을 위반하는 방식으로 작동하는 라이브러리들에 대한 참조들을 추가할 수 있다.
- [0018] 다음의 예들을 고려한다. 미국 정부는 미국 정부 컴퓨팅 시스템들 상에서 설치되도록 하기 위한 비친화적인 것으로 고려된 러시아 또는 다른 국가들에서 개발된 기술의 이용을 허용하지 않는다. 그러나, 러시아 회사들에 의해 개발되었던 RPA에서 이용된 일부 보편적인 라이브러리들(예컨대, Abbyy® OCR 엔진)이 있다. 개발자들은 자신의 컴퓨팅 시스템들 상에서 디폴트(default)에 의해 이 라이브러리들이 이용가능하지 않도록 선택할 수 있지만, 개발자들은 사용자가 RPA 프로젝트에서 이 라이브러리들을 편집시키는 종속성을 취하는 것을 명시적으로 방지할 수 없다. RPA 개발자들을 위한 인터넷 액세스는 차단될 수 있고, 이것은 소프트웨어를 개발할 때 많은 이유들로 바람직하지 않지만, 개발자는 유니버설 직렬 버스(universal serial bus; USB) 스틱과 같은 또 다른 메커니즘을 통해 미승인된 라이브러리들을 컴퓨팅 시스템으로 여전히 가져갈 수 있다.
- [0019] 또 다른 예로서, 많은 금융기관들은 어떤 정보가 그 내부 네트워크들 및 방화벽들을 떠날 수 있고 떠나지 않을 수 있는지에 관하여 가동 중인 엄격한 정책들을 가진다. 현대에는, (RPA 활동들을 통해 호출된) 많은 RPA 라이브러리들은 UiPath® 문서 이해 활동들과 같이, 공공 클라우드들 상의 서버들과 통신함으로써 작동한다. 이 금융기관들은 사용자들이 민감한 데이터를 회사 방화벽 외부로 전송할 수 없기 때문에, 이 활동들을 이용하는 것이 허용되지 않는다는 것을 사용자들이 인식하지 않을 것이라는 것을 우려한다. 그러나, 이것이 발생하는 것을 방지하기 위한 이전에 현존하는 효과적인 옵션들이 없다.
- [0020] 여전히 또 다른 우려는 스케일에서의 RPA가 시스템들을 압도하는 것을 방지하고 있다는 것이다. RPA의 대성공을 위한 이유들 중의 하나는 RPA가 그 시스템들에 대한 업그레이드들을 요구하지 않으면서, 레거시 시스템(legacy system)들의 자동화를 가능하게 한다는 것이다. 레거시 시스템들은 자동화되도록 설계되지 않았으므로, 그 설계 및 성능은 인간들이 작업할 수 있는 속력으로 상호작용들을 오직 취급할 수 있다. 그러나, 레거시 애플리케이션들은 RPA 로봇이 시스템과 상호작용할 수 있는 속력을 항상 추종할 수 없을 수 있다. 이 제약들 내에서 동작하도록 설계되지 않는 로봇은 레거시 시스템들이 프로세싱할 수 있는 것보다 더 빠른 요청들로 레거시 시스템들을 압도하는 것으로 필수적으로 종결될 수 있어서, 시스템에 대한 돌발적인 서비스 거부(denial of service; DoS) 공격을 필수적으로 수행한다. 활동들 사이의 지연들을 가능하게 하는 설정들이 UiPath Studio™에서 존재하지만, 개발자들이 RPA 로봇 속력으로 동작할 수 없는 레거시 시스템들로 작업할 때에 이 설정들을 이용한다는 것을 보장하기 위한 현재 현존하는 중앙 집행 메커니즘(central enforcement mechanism)이 없다.
- [0021] 또 다른 우려는 조직 정책 준수성을 보장하는 것이다. 많은 조직들은 이들이 자동화들에 적용하는 것을 원하는 특정 규칙들을 가진다. 예를 들어, 일부 예들은 자동화에 의해 생성된 이메일들이 외부 수신자들이 아니라 내부 피고용자들에게 오직 전송되는 것을 집행하는 것, 자동화에 의해 생성된 모든 이메일들이 인간 검토 없이 전송되는 것이 아니라, 드래프트(draft)로서 저장되는 것을 집행하는 것 등을 포함한다. 적절한 관장이 없으면, 이 정책들은 RPA 로봇들에 의해 위반될 수 있다.
- [0022] RPA에서 발생할 수 있는 액세스 제어 및 관장 문제들의 다양한 문제들을 해결하기 위하여, 일부 실시예들은 관장 정책들을 정의하고 집행하기 위한 조직들을 위한 능력을 제공한다. 예를 들어, 이것은 RPA 로봇이 액세스할 수 있고 및/또는 액세스할 수 없는 것을 정의하는 자바스크립트 오브젝트 표기(JavaScript Object Notation; JSON) 파일을 통해 구현될 수 있다. 일부 실시예들에서, 조직들은 어떤 애플리케이션들 및/또는 유니버설 자원 로케이터(URL)들이 자동화될 수 있고 및/또는 자동화될 수 없는지에 대해 제어들을 설정할 수 있고, 어떤 활동들이 이용될 수 있고 및/또는 이용될 수 없는지에 대해 제어들을 설정할 수 있고, 어떤 패키지들(예컨대, 라이브러리들)이 이용될 수 있고 및/또는 이용될 수 없는지에 대해 제어들을 설정할 수 있고, 관장 능력들을 확대하는 맞춤형 규칙들을 생성하고 전개하기 위한 능력을 제공할 수 있고, 예를 들어, 사용자들이 라이브러리들을 설치하기 위하여 어느 패키지 피드(package feed)들을 이용할 수 있는지를 제어하는 다양한 RPA 설계자 애플리케이션

이션 설정들을 제공할 수 있는 등을 행할 수 있다.

- [0023] 일부 실시예들에서, 관장은 설계 시에(즉, 개발자가 RPA 로봇 작업흐름을 구축하고, 테스트하고, 개정하고 있을 때) 집행될 수 있다. 그러나, 이것은 어떤 정책들을 집행하는 것을 어렵거나 불가능하게 할 수 있다. 예를 들어, 정책이 적용되고 있는 값이 실행시간(runtime)에 오직 알려지는 정책들을 집행하는 것은 어렵거나 불가능할 수 있다. 조직이 예를 들어, 어떤 웹사이트들에 대한 액세스를 방지하는 정책을 가동하도록 하는 것을 희망할 경우에, 설계 시간 분석은 플래그(flag)하기 위한 URL을 알 수 있어야 한다. 그러나, 일부 실시예들에서, 값이 외부 파일 또는 사용자에게 질문하는 프롬프트(prompt)와 같은 동적 소스로부터 로딩되는 그러한 방식으로 RPA 프로그램을 구성하는 것이 가능하다. 이 경우에, 정책을 위반하는 값은 설계 시간에 알려지지 않으므로, 설계 시간 분석은 충분하지 않다. 또 다른 예는 조직이 조직 외부의 고객들에게 이메일하는 것의 자동화를 방지하는 것을 희망할 경우이지만, 이메일 주소들의 리스트는 스프레드시트로부터 나온다. 이 경우에, 이메일들은 설계 시간에 알려지는 것이 아니라, 그 대신에, 실행시간에 평가되어야 하는 시스템에 대한 동적 입력이다.
- [0024] 어떤 실시예들에서, 관장은 설계 시간 집행에 추가적으로 또는 설계 시간 집행 대신에 실행시간에 집행된다. 이것은 예를 들어, 정책들을 구성하고 전개하기 위한 서버-측 애플리케이션을 통해 달성될 수 있다. 정책들은 이 컴퓨팅 시스템들이 예를 들어, 서버-측 지휘자 애플리케이션에 접속될 때에 사용자 컴퓨팅 시스템들로 자동적으로 전송할 수 있다. RPA 설계자 애플리케이션은 RPA 로봇들이 이 정책들을 획득하고 이를 준수하여 동작하는 것을 강제하는 코드를 RPA 로봇들 내로 자동적으로 삽입할 수 있다.
- [0025] 일부 실시예들에서, 개발자들이 RPA 로봇들을 개발할 때에 개발자들이 행할 수 있는 것을 제어하기 위하여, 제어들은 조직, 역할, 그룹, 개별적인 개발자 등에 기초하여 RPA 설계자 애플리케이션을 위하여 정의될 수 있다. 정책들이 정의될 수 있고, 제어들은 RPA 로봇이 자동화를 작동시킬 때에 무엇을 행할 수 있는지를 제어하기 위하여 집행될 수 있다. 이러한 실시예들은 조직들이 법률들, 협정들, 및 회사 정책들에 대한 그 준수성 및 관장 조건들을 더 양호하게 충족시키는 것을 가능하게 할 수 있다.
- [0026] 일부 실시예들에서, 로컬 파일들은 관리자(administrator)에 의해 오직 수정될 수 있다. 이와 같이, 비-관리자 사용자들은 이 파일들을 기각하거나 변경할 수 없어야 한다. 패키지 피드들은 오직 회망된 피드들이 이용가능하고 추가적인 피드들이 추가될 수 없도록 제재(lock down)될 수 있다. 이것은 사용자들이 다른 곳에서 획득되는 패키지들을 이용하는 것을 방지한다. 어떤 실시예들에서, "피드백 전송(send feedback)" 톨은 정책의 일부로서 디스플레이될 수 있다.
- [0027] "패키지 피드"는 본원에서 이용된 바와 같이, 개발자가 그 또는 그녀의 자동화 프로젝트를 위한 추가적인 능력들을 취득하기 위하여 임의적으로 설치하도록 선택할 수 있는 추가적인 라이브러리들에 대한 액세스를 포함하는 위치(원격 또는 로컬의 어느 하나)이다. 라이브러리는 본원에서 이용된 바와 같이, 추가적인 능력들(예컨대, 활동들)을 프로젝트에 제공하는 컴퓨터 코드를 포함하는 패키지이다. 패키지 피드의 예는 UiPath Connect™ 마켓플레이스(marketplace)이다. 이것은 커뮤니티 회원들이 자신들이 기록한 맞춤형 활동들을 포함하는 라이브러리들을 업로딩할 수 있는 개방된 위치이다. UiPath® RPA 개발자가 이 능력들을 이용할 것을 희망할 경우에, 그 개발자는 그 또는 그녀의 프로젝트들에 대한 라이브러리 참조들을 브라우징할 수 있고 추가할 수 있다. 라이브러리의 예는 UiPath GSuite™ 활동들이다. UiPath Studio™의 디폴트 설치하는 RPA 개발자들에게 GSuite™로 작업하기 위한 능력을 부여하지 않는다. 그러나, 개발자들은 Gsuite™(예컨대, Gmail®, Google® 시트들(Sheets), Google® 드라이브(Drive) 등)과 작업하기 위하여 활동들을 이용하기 위한 능력을 그 프로젝트에 이제 부여하는 GSuite™ 활동 라이브러리를 설치할 수 있다.
- [0028] 일부 실시예들에서, 프로그램 분석기 또는 RPA 작업흐름 분석기와 같은 코드 분석기는 RPA 설계자 애플리케이션을 위한 감사 기능(audit function)으로서 제공된다. 작업흐름 분석기는 사용자들이 RPA 설계자 애플리케이션을 작동시키고 정책들을 집행할 수 있을 때에 사용자들을 위하여 허가되는 것을 정의할 수 있다. 예를 들어, 사용자들이 작업할 수 있는 허가된 URL들이 특정될 수 있고, 어떤 URL들은 한정될 수 있는 등과 같다.
- [0029] 본원에서 이용된 바와 같이, "코드 분석기"는 RPA 사용자에게 의해 기록된 코드를 검사하고 애플리케이션이 규칙들을 만족시키지 않을 때에 피드백(예컨대, 메시지들, 경고들, 에러들 등)을 생성하는 일련의 규칙들을 작동시키는 컴퓨터 코드이다. 일부 실시예들에서, 코드 분석기는 RPA 애플리케이션의 벤더(예컨대, UiPath®)에 의해 제공되거나, RPA 애플리케이션을 이용하여 고객들에 의해 기록된다. 코드 분석기 규칙의 예는 조직들이 어떤 데스크톱 애플리케이션들 및 웹사이트 URL들이 자동화 프로젝트에서 이용될 수 있거나 이용되지 않을 수 있는지를 정의하는 것을 허용하는 UiPath Studio™에서의 "App/Url 제약" 규칙이다. 사용자가 이 규칙들 중의 하나를 어길 경우에, 그 또는 그녀는 관장 정책에서 특정된 심각도로 통지를 받을 것이다. 일부 실시예들에서, 이것은

통상적으로, 자동화가 발행되거나 작동되는 것을 방지할 에러이다.

- [0030] 어떤 실시예들은 로봇 프로세스 자동화(RPA)를 위하여 채용될 수 있다. 도 1은 본 발명의 실시예에 따라, RPA 시스템(100)을 예시하는 아키텍처 다이어그램이다. RPA 시스템(100)은 개발자가 작업흐름들을 설계하고 구현하는 것을 허용하는 설계자(110)를 포함한다. 설계자(110)는 애플리케이션 통합 뿐만 아니라, 제3자 애플리케이션들 자동화, 관리 정보 기술(Information Technology; IT) 태스크들, 및 비즈니스 IT 프로세스들을 위한 해결책을 제공할 수 있다. 설계자(110)는 비즈니스 프로세스의 그래픽 표현인 자동화 프로젝트(automation project)의 개발을 용이하게 할 수 있다. 간단하게 말하면, 설계자(110)는 작업흐름들 및 로봇들의 개발 및 전개(deployment)를 용이하게 한다.
- [0031] 자동화 프로젝트는 실행 순서의 개발자 제어, 및 "활동(activity)들"로서 본원에서 정의된, 작업흐름에서 개발된 단계들의 맞춤형 세트(custom set) 사이의 관계를 제공함으로써 규칙-기반 프로세스들의 자동화를 가능하게 한다. 설계자(110)의 실시예의 하나의 상업적인 예는 UiPath Studio™이다. 각각의 활동은 버튼을 클릭하는 것, 파일을 판독하는 것, 로그 패널(log panel)로 기록하는 것 등과 같은 액션(action)을 포함할 수 있다. 일부 실시예들에서, 작업흐름들은 네스팅(nest)될 수 있거나 내장될 수 있다.
- [0032] 일부 유형들의 작업흐름들은 시퀀스들, 플로우차트들, FSM들, 및/또는 전역적 예외 핸들러(global exception handler)들을 포함할 수 있지만, 이것으로 제한되지는 않는다. 시퀀스들은 선형 프로세스들을 위하여 특히 적당할 수 있어서, 작업흐름을 혼란스럽게 하지 않으면서, 하나의 활동으로부터 또 다른 것으로의 흐름을 가능하게 할 수 있다. 플로우차트들은 더 복잡한 비즈니스 로직에 특히 적당할 수 있어서, 다수의 분기 로직 조각자들을 통해 더 다양한 방식으로 판정들의 통합 및 활동들의 접속을 가능하게 할 수 있다. FSM들은 큰 작업흐름들을 위하여 특히 적당할 수 있다. FSM들은 그 실행에서 유한한 수의 상태(state)들을 이용할 수 있고, 상태들은 조건(즉, 전이) 또는 활동에 의해 트리거링된다. 전역적 예외 핸들러들은 실행 에러(execution error)를 조우할 때에 작업흐름 거동(workflow behavior)을 결정하기 위하여, 그리고 디버깅 프로세스들을 위하여 특히 적당할 수 있다.
- [0033] 일단 작업흐름이 설계자(110)에서 개발되면, 비즈니스 프로세스들의 실행은 설계자(110)에서 개발된 작업흐름들을 실행하는 하나 이상의 로봇들(130)을 편성(orchestrate)하는 지휘자(conductor)(120)에 의해 편성된다. 지휘자(120)의 실시예의 하나의 상업적인 예는 UiPath Orchestrator™이다. 지휘자(120)는 환경에서 자원들의 생성, 모니터링, 및 전개의 관리를 용이하게 한다. 지휘자(120)는 제3자 해결책들 및 애플리케이션들과의 통합 포인트(integration point) 또는 응집 포인트(aggregation point)들 중의 하나로서 작용할 수 있다.
- [0034] 지휘자(120)는 로봇들(130)의 함대(fleet)를 관리할 수 있어서, 중앙집중화된 포인트로부터 로봇들(130)을 접속하고 실행할 수 있다. 관리될 수 있는 로봇들(130)의 유형들은 유인 로봇들(132), 무인 로봇들(134), (무인 로봇들(134)과 유사하지만, 개발 및 테스트 목적들을 위하여 이용된) 개발 로봇들, 및 (유인 로봇들(132)과 유사하지만, 개발 및 테스트 목적들을 위하여 이용된) 비생산 로봇들을 포함하지만, 이것으로 제한되지는 않는다. 유인 로봇들(132)은 사용자 이벤트들에 의해 트리거링되고, 동일한 컴퓨팅 시스템 상에서 인간과 함께 동작한다. 유인 로봇들(132)은 중앙집중화된 프로세스 전개 및 로깅 매체를 위한 지휘자(120)와 함께 이용될 수 있다. 유인 로봇들(132)은 인간 사용자가 다양한 태스크들을 달성하는 것을 도울 수 있고, 사용자 이벤트들에 의해 트리거링될 수 있다. 일부 실시예들에서, 프로세스들은 이 유형의 로봇에 대한 지휘자(120)로부터 시작될 수 없고, 및/또는 프로세스들은 록킹된 스크린 하에서 작동할 수 없다. 어떤 실시예들에서, 유인 로봇들(132)은 로봇 트레이(robot tray)로부터 또는 커맨드 프롬프트(command prompt)로부터 오직 시작될 수 있다. 유인 로봇들(132)은 일부 실시예들에서 인간 감독 하에서 작동해야 한다.
- [0035] 무인 로봇들(134)은 가상적 환경들에서 무인으로 작동하고, 많은 프로세스들을 자동화할 수 있다. 무인 로봇들(134)은 원격 실행, 모니터링, 스케줄링, 및 작업 큐들에 대한 지원을 제공하는 것을 담당할 수 있다. 모든 로봇 유형들에 대한 디버깅(debugging)은 일부 실시예들에서 설계자(110) 내에서 작동될 수 있다. 양자의 유인 및 무인 로봇들은, 메인프레임(mainframe)들, 웹 애플리케이션들, VM들, 기업 애플리케이션들(예컨대, SAP®, Salesforce®, Oracle® 등에 의해 생산된 것들), 및 컴퓨팅 시스템 애플리케이션들(예컨대, 데스크톱 및 랩톱 애플리케이션들, 이동 디바이스 애플리케이션들, 웨어러블 컴퓨터 애플리케이션들 등)을 포함하지만 이것으로 제한되지는 않는 다양한 시스템들 및 애플리케이션들을 자동화할 수 있다.
- [0036] 지휘자(120)는 프로비저닝(provisioning), 전개, 버저닝(versioning), 구성, 큐잉(queueing), 모니터링(monitring), 로깅(logging), 및/또는 상호접속성 제공을 포함하지만 이것으로 제한되지는 않는 다양한 능력들을 가질 수 있다. 프로비저닝은 로봇들(130)과 지휘자(120)(예컨대, 웹 애플리케이션) 사이의 접속들의 생성

및 유지보수를 포함할 수 있다. 전개는 실행을 위한 배정된 로봇들(130)로의 패키지 버전들의 올바른 전달을 보장하는 것을 포함할 수 있다. 버저닝은 일부 실시예들에서 일부 프로세스 또는 구성의 고유한 사례(instance)들의 관리를 포함할 수 있다. 구성은 로봇 환경들 및 프로세스 구성들의 유지보수 및 전달을 포함할 수 있다. 큐잉은 큐들 및 큐 항목들의 관리를 제공하는 것을 포함할 수 있다. 모니터링은 로봇 식별 데이터를 추적하는 것 및 사용자 허가들을 유지하는 것을 포함할 수 있다. 로깅은 로그들을 데이터베이스(예컨대, SQL 데이터베이스) 및/또는 또 다른 저장 메커니즘(예컨대, 큰 데이터세트들을 저장하고 신속하게 질의하기 위한 능력을 제공하는 ElasticSearch®)으로 저장하고 인덱싱하는 것을 포함할 수 있다. 지휘자(120)는 제3자 해결책들 및/또는 애플리케이션들을 위한 통신의 중앙집중화된 포인트로서 작용함으로써 상호접속성을 제공할 수 있다.

[0037] 로봇들(130)은 설계자(110) 내에 구축된 작업흐름들을 작동시키는 실행 에이전트(execution agent)들이다. 로봇(들)(130)의 일부 실시예들의 하나의 상업적인 예는 UiPath Robots™이다. 일부 실시예들에서, 로봇들(130)은 Microsoft Windows® 서비스 제어 관리자(Service Control Manager; SCM)-관리된 서비스를 디폴트(default)로 설치한다. 그 결과, 이러한 로봇들(130)은 로컬 시스템 계정(local system account) 하에서 대화형(interactive) Windows® 세션들을 개방할 수 있고, Windows® 서비스의 권리들을 가질 수 있다.

[0038] 일부 실시예들에서, 로봇들(130)은 사용자 모드에서 설치될 수 있다. 이러한 로봇들(130)에 대하여, 이것은 로봇들이 주어진 로봇(130)이 설치된 사용자와 동일한 권리들을 가진다는 것을 의미한다. 이 특징은 또한, 그 최대 잠재력에서 각각의 머신의 전체 사용을 보장하는 고밀도(High Density; HD) 로봇들을 위하여 이용가능할 수 있다. 일부 실시예들에서, 임의의 유형의 로봇(130)은 HD 환경에서 구성될 수 있다.

[0039] 일부 실시예들에서의 로봇들(130)은 몇몇 컴포넌트들로 분할되고, 각각은 특정한 자동화 태스크에 전용된다. 일부 실시예들에서의 로봇 컴포넌트들은 SCM-관리된 로봇 서비스들, 사용자 모드 로봇 서비스들, 실행자(executor)들, 에이전트들, 및 커맨드 라인(command line)을 포함하지만, 이것으로 제한되지는 않는다. SCM-관리된 로봇 서비스들은 Windows® 세션들을 관리 및 모니터링하고, 지휘자(120)와 실행 호스트들(즉, 로봇들(130)이 그 상에서 실행되는 컴퓨팅 시스템들) 사이의 프록시(proxy)로서 작용한다. 이 서비스들은 로봇들(130)을 위한 크리덴셜(credential)들로 신뢰되고, 로봇들(130)을 위한 크리덴셜들을 관리한다. 콘솔 애플리케이션은 로컬 시스템 하에서 SCM에 의해 기동된다.

[0040] 일부 실시예들에서의 사용자 모드 로봇 서비스들은 Windows® 세션들을 관리 및 모니터링하고, 지휘자(120)와 실행 호스트들 사이의 프록시로서 작용한다. 사용자 모드 로봇 서비스들은 로봇들(130)을 위한 크리덴셜들로 신뢰될 수 있고, 로봇들(130)을 위한 크리덴셜들을 관리할 수 있다. Windows® 애플리케이션은 SCM-관리된 로봇 서비스가 설치되지 않을 경우에 자동적으로 기동될 수 있다.

[0041] 실행자들은 Windows® 세션 하에서 주어진 잡(job)들을 작동시킬 수 있다(즉, 실행자들은 작업흐름들을 실행할 수 있음). 실행자들은 모니터-당(per-monitor) 인치 당 도트들(dots per inch; DPI) 설정들을 인지할 수 있다. 에이전트들은 시스템 트레이 윈도우(system tray window)에서 이용가능한 잡들을 디스플레이하는 Windows® 프리젠테이션 파운데이션(Windows® Presentation Foundation; WPF) 애플리케이션들일 수 있다. 에이전트들은 서비스의 클라이언트일 수 있다. 에이전트들은 잡들을 시작하거나 정지시키고 설정들을 변경하도록 요청할 수 있다. 커맨드 라인은 서비스의 클라이언트이다. 커맨드 라인은 잡들을 시작하도록 요청할 수 있고 그 출력을 대기하는 콘솔 애플리케이션(console application)이다.

[0042] 위에서 설명된 바와 같이 분할된 로봇들(130)의 컴포넌트들을 가지는 것은 개발자들, 지원 사용자들, 및 컴퓨팅 시스템들이 각각의 컴포넌트가 실행하고 있는 것을 더 용이하게 작동시키고, 식별하고, 추적하는 것을 돕는다. 특수한 거동들은 실행자 및 서비스를 위한 상이한 방화벽 규칙(firewall rule)들을 셋업하는 것과 같이, 컴포넌트 당 이러한 방식으로 구성될 수 있다. 실행자는 일부 실시예들에서 모니터 당 DPI 설정들을 항상 인지할 수 있다. 그 결과, 작업흐름들은 작업흐름들이 생성된 컴퓨팅 시스템의 구성에 관계 없이, 임의의 DPI에서 실행될 수 있다. 설계자(110)로부터의 프로젝트들은 또한, 일부 실시예들에서 브라우저 줌 레벨(browser zoom level)에 독립적일 수 있다. DPI-비인지(unaware)이거나 의도적으로 비인지인 것으로서 표기되는 애플리케이션들에 대하여, DPI는 일부 실시예들에서 디스플레이될 수 있다.

[0043] 도 2는 본 발명의 실시예에 따라, 전개된 RPA 시스템(200)을 예시하는 아키텍처 다이어그램이다. 일부 실시예들에서, RPA 시스템(200)은 도 1의 RPA 시스템(100)일 수 있거나 그 일부일 수 있다. 클라이언트 측, 서버 측, 또는 양자 모두는 발명의 범위로부터 이탈하지 않으면서, 임의의 희망된 수의 컴퓨팅 시스템들을 포함할 수 있다는 것이 주목되어야 한다. 클라이언트 측 상에서, 로봇 애플리케이션(210)은 실행자들(212), 에이전트(214),

및 설계자(216)를 포함한다. 그러나, 일부 실시예들에서, 설계자(216)는 컴퓨팅 시스템(210) 상에서 작동하고 있지 않을 수 있다. 실행자들(212)은 프로세스들을 작동시키고 있다. 몇몇 비즈니스 프로젝트들은 도 2에서 도시된 바와 같이 동시에 작동할 수 있다. 에이전트(214)(예컨대, Windows® 서비스)는 이 실시예에서 모든 실행자들(212)을 위한 접촉의 단일 포인트이다. 이 실시예에서의 모든 메시지들은 지휘자(230)로 로깅되고, 지휘자(230)는 데이터베이스 서버(240), 인덱서 서버(indexer server)(250), 또는 양자 모두를 통해 메시지들을 추가로 프로세싱한다. 도 1에 대하여 위에서 논의된 바와 같이, 실행자들(212)은 로봇 컴포넌트들일 수 있다.

[0044] 일부 실시예들에서, 로봇은 머신 명칭과 사용자명(username) 사이의 연관성을 표현한다. 로봇은 다수의 실행자들을 동시에 관리할 수 있다. 동시에 작동하는 다수의 대화형 세션들을 지원하는 컴퓨팅 시스템들(예컨대, Windows® 서버(2012)) 상에서, 다수의 로봇들은 각각 고유한 사용자명을 이용하는 별도의 Windows® 세션에서 동시에 작동하고 있을 수 있다. 이것은 위의 HD 로봇들로서 지칭된다.

[0045] 에이전트(214)는 또한, 로봇의 스테이터스(status)를 전송하고(예컨대, 로봇이 여전히 기능하고 있다는 것을 표시하는 "하트비트(heartbeat)" 메시지를 주기적으로 전송함) 실행되어야 할 패키지의 요구된 버전을 다운로드하는 것을 담당한다. 에이전트(214)와 지휘자(230) 사이의 통신은 일부 실시예들에서 에이전트(214)에 의해 항상 개시된다. 통지 시나리오에서, 에이전트(214)는 커맨드들을 로봇으로 전송하기 위하여(예컨대, 시작, 정지 등) 지휘자(230)에 의해 더 이후에 이용되는 웹소켓(WebSocket) 채널을 개방할 수 있다.

[0046] 서버 측 상에서, 프리젠테이션 계층(웹 애플리케이션(232), 개방 데이터 프로토콜(Open Data Protocol; OData) 대표 상태 전송(Representative State Transfer; REST) 애플리케이션 프로그래밍 인터페이스(Application Programming Interface; API) 엔드포인트(endpoint)들(234), 및 통지 및 모니터링(236)), 서비스 계층(API 구현/비즈니스 로직(238)), 및 지속 계층(persistence layer)(데이터베이스 서버(240) 및 인덱서 서버(250))이 포함된다. 지휘자(230)는 웹 애플리케이션(232), ODATA REST API 엔드포인트들(234), 통지 및 모니터링(236), 및 API 구현/비즈니스 로직(238)을 포함한다. 일부 실시예들에서, 사용자가 (예컨대, 브라우저(230)를 통해) 지휘자(220)의 인터페이스에서 수행하는 대부분의 액션들은 다양한 API들을 호출함으로써 수행된다. 이러한 액션들은 발명의 범위로부터 이탈하지 않으면서, 로봇들 상에서 잡들을 시작하는 것, 큐들에서 데이터를 추가/제거하는 것, 무인으로 작동시키기 위한 잡들을 스케줄링하는 것 등을 포함할 수 있지만, 이것으로 제한되지는 않는다. 웹 애플리케이션(232)은 서버 플랫폼의 시각적 계층이다. 이 실시예에서, 웹 애플리케이션(232)은 하이퍼텍스트 마크업 언어(Hypertext Markup Language; HTML) 및 자바스크립트(JavaScript; JS)를 이용한다. 그러나, 임의의 희망된 마크업 언어들, 스크립트 언어(script language)들, 또는 임의의 다른 포맷들은 발명의 범위로부터 이탈하지 않으면서 이용될 수 있다. 사용자는 지휘자(230)를 제어하기 위한 다양한 액션들을 수행하기 위하여 이 실시예에서 브라우저(220)를 통해 웹 애플리케이션(232)으로부터의 웹 페이지들과 상호작용한다. 예를 들어, 사용자는 로봇 그룹들을 생성할 수 있고, 패키지들을 로봇들에 배정할 수 있고, 로봇 당 및/또는 프로세스 당 로그들을 분석할 수 있고, 로봇들을 시작 및 정지시킬 수 있는 등과 같다.

[0047] 웹 애플리케이션(232)에 추가적으로, 지휘자(230)는 또한, OData REST API 엔드포인트들(234)을 노출시키는 서비스 계층을 포함한다. 그러나, 다른 엔드포인트들이 발명의 범위를 이탈하지 않으면서 포함될 수 있다. REST API는 양자의 웹 애플리케이션(232) 및 에이전트(214)에 의해 소비된다. 에이전트(214)는 이 실시예에서 클라이언트 컴퓨터 상의 하나 이상의 로봇들의 감독자이다.

[0048] 이 실시예에서의 REST API는 구성, 로깅, 모니터링, 및 큐잉 기능성을 커버한다. 구성 엔드포인트들은 일부 실시예들에서 애플리케이션 사용자들, 허가들, 로봇들, 자산(asset)들, 배포(release)들, 및 환경들을 정의하고 구성하기 위하여 이용될 수 있다. 로깅 REST 엔드포인트들은 예를 들어, 에러들, 로봇들에 의해 전송된 명시적 메시지들, 및 다른 환경-특정 정보와 같은 상이한 정보를 로깅하기 위하여 이용될 수 있다. 전개 REST 엔드포인트들은 시작 잡 커맨드가 지휘자(230)에서 이용될 경우에 실행되어야 하는 패키지 버전을 질의하기 위하여 로봇들에 의해 이용될 수 있다. 큐잉 REST 엔드포인트들은 데이터를 큐에 추가하는 것, 큐로부터 트랜잭션(transaction)을 획득하는 것, 트랜잭션의 스테이터스를 설정하는 것 등과 같은, 큐들 및 큐 항목 관리를 담당할 수 있다.

[0049] 모니터링 REST 엔드포인트들은 웹 애플리케이션(232) 및 에이전트(214)를 모니터링할 수 있다. 통지 및 모니터링 API(236)는, 에이전트(214)를 등록하고, 구성 설정들을 에이전트(214)로 전달하고, 서버 및 에이전트(214)로부터 통지들을 전송/수신하기 위하여 이용되는 REST 엔드포인트들일 수 있다. 통지 및 모니터링 API(236)는 또한, 일부 실시예들에서 웹소켓 통신을 이용할 수 있다.

[0050] 지속 계층은 이 실시예에서 한 쌍의 서버들 - 데이터베이스 서버(240)(예컨대, SQL 서버) 및 인덱서 서버(250)

- 을 포함한다. 이 실시예에서의 데이터베이스 서버(240)는 로봇들의 구성들, 로봇 그룹들, 연관된 프로세스들, 사용자들, 역할들, 스케줄들 등을 저장한다. 이 정보는 일부 실시예들에서 웹 애플리케이션(232)을 통해 관리된다. 데이터베이스 서버(240)는 큐들 및 큐 항목들을 관리할 수 있다. 일부 실시예들에서, 데이터베이스 서버(240)는 (인덱서 서버(250)에 추가적으로 또는 그 대신에) 로봇들에 의해 로깅된 메시지들을 저장할 수 있다.

[0051] 일부 실시예들에서 임의적인 인덱서 서버(250)는 로봇들에 의해 로깅된 정보를 저장하고 인덱싱한다. 어떤 실시예들에서, 인덱서 서버(250)는 구성 설정들을 통해 디스에이블될 수 있다. 일부 실시예들에서, 인덱서 서버(250)는 개방 소스 프로젝트 전체-텍스트 검색 엔진(open source project full-text search engine)인 ElasticSearch®를 이용한다. (예컨대, 로그 메시지 또는 기록 라인과 같은 활동들을 이용하는) 로봇들에 의해 로깅된 메시지들은 로깅 REST 엔드포인트(들)를 통해 인덱서 서버(250)로 전송될 수 있고, 여기서, 메시지들은 미래의 사용을 위하여 인덱싱된다.

[0052] 도 3은 본 발명의 실시예에 따라, 설계자(310), 활동들(320, 330), 및 구동기들(340) 사이의 관계(300)를 예시하는 아키텍처 다이어그램이다. 상기에 따라, 개발자는 로봇들에 의해 실행되는 작업흐름들을 개발하기 위하여 설계자(310)를 이용한다. 작업흐름들은 사용자-정의된 활동들(320) 및 UI 자동화 활동들(330)을 포함할 수 있다. 일부 실시예들은 이미지에서의 비-텍스트 시각적 컴포넌트들을 식별할 수 있고, 이것은 본원에서 컴퓨터 비전(CV)으로 칭해진다. 이러한 컴포넌트들에 속하는 일부 CV 활동들은 클릭(click), 타이핑(type), 텍스트 얻기(get text), 호버(hover), 엘리먼트 존재, 범위 리프레시(refresh scope), 강조표시 등을 포함할 수 있지만, 이것으로 제한되지는 않는다. 일부 실시예들에서의 클릭은 예를 들어, CV, 광학적 문자 인식(OCR), 퍼지 텍스트 정합(fuzzy text matching), 및 멀티-앵커(multi-anchor)를 이용하여 엘리먼트를 식별하고, 그것을 클릭한다. 타이핑은 상기한 것을 이용하여 엘리먼트를 식별할 수 있고, 엘리먼트에서 타이핑한다. 텍스트 얻기는 특정 텍스트의 위치를 식별할 수 있고, OCR을 이용하여 그 텍스트를 스캔할 수 있다. 호버는 엘리먼트를 식별할 수 있고, 엘리먼트 상에서 호버링할 수 있다. 엘리먼트 존재는 위에서 설명된 기법들을 이용하여 스크린 상에 엘리먼트가 존재하는지 여부를 체크할 수 있다. 일부 실시예들에서는, 설계자(310) 내에 구현될 수 있는 수백 또는 심지어 수천 개의 활동들이 있을 수 있다. 그러나, 임의의 수 및/또는 유형의 활동들은 발명의 범위로부터 이탈하지 않으면서 이용가능할 수 있다.

[0053] UI 자동화 활동들(330)은, 더 낮은 레벨의 코드로 기록되는 특수한 더 낮은 레벨의 활동들(예컨대, CV 활동들)의 서브세트(subset)이고, 스크린과의 상호작용들을 용이하게 한다. UI 자동화 활동들(330)은 로봇이 회망된 소프트웨어와 상호작용하는 것을 허용하는 구동기들(340)을 통해 이 상호작용들을 용이하게 한다. 예를 들어, 구동기들(340)은 OS 구동기들(342), 브라우저 구동기들(344), VM 구동기들(346), 기업 애플리케이션 구동기들(348) 등을 포함할 수 있다.

[0054] 구동기들(340)은 후크들을 검색하고, 키들에 대해 모니터링하는 등의 낮은 레벨에서 OS와 상호작용할 수 있다. 구동기들은 Chrome®, IE®, Citrix®, SAP® 등과의 상호작용을 용이하게 할 수 있다. 예를 들어, "클릭" 활동은 구동기들(340)을 통해 이 상이한 애플리케이션들에서 동일한 역할을 수행한다.

[0055] 도 4는 본 발명의 실시예에 따라, RPA 시스템(400)을 예시하는 아키텍처 다이어그램이다. 일부 실시예들에서, RPA 시스템(400)은 도 1 및/또는 도 2의 RPA 시스템들(100 및/또는 200)일 수 있거나 이들을 포함할 수 있다. RPA 시스템(400)은 로봇들을 작동시키는 다수의 클라이언트 컴퓨팅 시스템들(410)을 포함한다. 컴퓨팅 시스템들(410)은 그 상에서 작동하는 웹 애플리케이션을 통해 지휘자 컴퓨팅 시스템(420)과 통신할 수 있다. 지휘자 컴퓨팅 시스템(420)은 궁극적으로, 데이터베이스 서버(430) 및 임의적인 인덱서 서버(440)와 통신할 수 있다.

[0056] 도 1 및 도 3에 대하여, 웹 애플리케이션이 이 실시예들에서 이용되지만, 임의의 적당한 클라이언트/서버 소프트웨어는 발명의 범위로부터 이탈하지 않으면서 이용될 수 있다는 것이 주목되어야 한다. 예를 들어, 지휘자는 클라이언트 컴퓨팅 시스템들 상에서 비-웹-기반(non-web-based) 클라이언트 소프트웨어 애플리케이션들과 통신하는 서버-측 애플리케이션을 작동시킬 수 있다.

[0057] 도 5는 본 발명의 실시예에 따라, RPA를 위한 로봇 액세스 제어 및 관장을 수행하도록 구성된 컴퓨팅 시스템(500)을 예시하는 아키텍처 다이어그램이다. 일부 실시예들에서, 컴퓨팅 시스템(500)은 본원에서 도시되고 및/또는 설명된 컴퓨팅 시스템들 중의 하나 이상일 수 있다. 컴퓨팅 시스템(500)은 정보를 통신하기 위한 버스(505) 또는 다른 통신 메커니즘, 및 정보를 프로세싱하기 위하여 버스(505)에 결합된 프로세서(들)(510)를 포함한다. 프로세서(들)(510)는 중앙 프로세싱 유닛(Central Processing Unit; CPU), 애플리케이션 특정 집적 회로(Application Specific Integrated Circuit; ASIC), 필드 프로그래밍가능 게이트 어레이(Field Programmable

Gate Array; FPGA), 그래픽 프로세싱 유닛(Graphics Processing Unit; GPU), 그 다수의 사례들, 및/또는 그 임의의 조합을 포함하는 임의의 유형의 일반 또는 특정 목적 프로세서일 수 있다. 프로세서(들)(510)는 또한, 다수의 프로세싱 코어들을 가질 수 있고, 코어들의 적어도 일부는 특정 기능들을 수행하도록 구성될 수 있다. 멀티-병렬(multi-parallel) 프로세싱은 일부 실시예들에서 이용될 수 있다. 어떤 실시예들에서, 프로세서(들)(510) 중의 적어도 하나는 생물학적 뉴런(neuron)들을 모방하는 프로세싱 엘리먼트들을 포함하는 뉴로모픽 회로(neuromorphic circuit)일 수 있다. 일부 실시예들에서, 뉴로모픽 회로들은 폰 노이만(Von Neumann) 컴퓨팅 아키텍처의 전형적인 컴포넌트들을 요구하지 않을 수 있다.

[0058] 컴퓨팅 시스템(500)은 프로세서(들)(510)에 의해 실행되어야 명령들 및 정보를 저장하기 위한 메모리(515)를 더 포함한다. 메모리(515)는 랜덤 액세스 메모리(Random Access Memory; RAM), 판독 전용 메모리(Read Only Memory; ROM), 플래시 메모리, 캐시, 자기적 또는 광학 디스크와 같은 정적 스토리지, 또는 임의의 다른 유형들의 비-일시적 컴퓨터-판독가능 매체들의 임의의 조합 또는 그 조합들로 이루어질 수 있다. 비-일시적 컴퓨터-판독가능 매체들은 프로세서(들)(510)에 의해 액세스될 수 있고 휘발성 매체들, 비-휘발성 매체들, 또는 양자 모두를 포함할 수 있는 임의의 이용가능한 매체들일 수 있다. 매체들은 또한, 분리가능, 비-분리가능, 또는 양자 모두일 수 있다.

[0059] 추가적으로, 컴퓨팅 시스템(500)은 무선 및/또는 유선 접속을 통해 통신 네트워크에 대한 액세스를 제공하기 위한, 트랜시버(transceiver)와 같은 통신 디바이스(520)를 포함한다. 일부 실시예들에서, 통신 디바이스(520)는 발명의 범위로부터 이탈하지 않으면서, 주파수 분할 다중 액세스(Frequency Division Multiple Access; FDMA), 단일 캐리어 FDMA(Single Carrier FDMA; SC-FDMA), 시간 분할 다중 액세스(Time Division Multiple Access; TDMA), 코드 분할 다중 액세스(Code Division Multiple Access; CDMA), 직교 주파수 분할 멀티플렉싱(Orthogonal Frequency Division Multiplexing; OFDM), 직교 주파수 분할 다중 액세스(Orthogonal Frequency Division Multiple Access; OFDMA), 이동 통신들을 위한 글로벌 시스템(Global System for Mobile communications; GSM), 범용 패킷 라디오 서비스(General Packet Radio Service; GPRS), 유니버설 이동 전기통신 시스템(Universal Mobile Telecommunications System; UMTS), cdma2000, 광대역 CDMA(Wideband CDMA; W-CDMA), 고속 다운링크 패킷 액세스(High-Speed Downlink Packet Access; HSDPA), 고속 업링크 패킷 액세스(High-Speed Uplink Packet Access; HSUPA), 고속 패킷 액세스(High-Speed Packet Access; HSPA), 롱텀 에볼루션(Long Term Evolution; LTE), LTE 어드밴스드(LTE Advanced; LTE-A), 802.11x, Wi-Fi, 지그비(Zigbee), 초광대역(Ultra-WideBand; UWB), 802.16x, 802.15, 홈 노드-B(Home Node-B; HnB), 블루투스(Bluetooth), 라디오 주파수 식별(Radio Frequency Identification; RFID), 적외선 데이터 연계(Infrared Data Association; IrDA), 근접장 통신(Near-Field Communications; NFC), 제5 세대(fifth generation; 5G), 뉴 라디오(New Radio; NR), 그 임의의 조합, 및/또는 임의의 다른 현재 현존하거나 미래-구현된 통신 표준 및/또는 프로토콜을 이용하도록 구성될 수 있다. 일부 실시예들에서, 통신 디바이스(520)는 발명의 범위로부터 이탈하지 않으면서, 단일체, 어레이(array)되고, 페이징(phase)되고, 스위칭되고, 빔포밍(beamforming), 빔조향(beamsteering), 그 조합, 및/또는 임의의 다른 안테나 구성인 하나 이상의 안테나들을 포함할 수 있다.

[0060] 프로세서(들)(510)는 플라즈마 디스플레이, 액정 디스플레이(Liquid Crystal Display; LCD), 발광 다이오드(Light Emitting Diode; LED) 디스플레이, 필드 방출 디스플레이(Field Emission Display; FED), 유기 발광 다이오드(Organic Light Emitting Diode; OLED) 디스플레이, 플렉시블 OLED 디스플레이, 플렉시블 기판 디스플레이, 투사 디스플레이(projection display), 4K 디스플레이, 고해상도 디스플레이(high definition display), Retina® 디스플레이, 평면내 스위칭(In-Plane Switching; IPS) 디스플레이, 또는 정보를 사용자에게 디스플레이하기 위한 임의의 다른 적당한 디스플레이와 같은 디스플레이(525)에 버스(505)를 통해 추가로 결합된다. 디스플레이(525)는 저항성, 용량성, 표면-음향파(surface-acoustic wave; SAW) 용량성, 적외선, 광학적 이미징, 분산 신호 기술, 음향 펄스 인식, 전반사 장애(frustrated total internal reflection) 등을 이용하여 터치(햅틱(haptic)) 디스플레이, 3 차원(3D) 터치 디스플레이, 멀티-입력 터치 디스플레이, 멀티-터치 디스플레이 등으로 구성될 수 있다. 임의의 적당한 디스플레이 디바이스 및 햅틱 I/O는 발명의 범위로부터 이탈하지 않으면서 이용될 수 있다.

[0061] 키보드(530) 및, 컴퓨터 마우스, 터치패드 등과 같은 커서 제어 디바이스(535)는 사용자가 컴퓨팅 시스템(500)과 인터페이스하는 것을 가능하게 하기 위하여 버스(505)에 추가로 결합된다. 그러나, 어떤 실시예들에서, 물리적 키보드 및 마우스는 존재하지 않을 수 있고, 사용자는 전적으로 디스플레이(525) 및/또는 터치패드(도시되지 않음)를 통해 디바이스와 상호작용할 수 있다. 입력 디바이스들의 임의의 유형 및 조합은 설계 선택의 문제로서 이용될 수 있다. 어떤 실시예들에서, 물리적 입력 디바이스 및/또는 디스플레이는 존재하지 않는다. 예

를 들어, 사용자는 그 통신하는 또 다른 컴퓨팅 시스템을 통해 원격으로 컴퓨팅 시스템(500)과 상호작용할 수 있거나, 컴퓨팅 시스템(500)은 자율적으로 동작할 수 있다.

[0062] 메모리(515)는 프로세서(들)(510)에 의해 실행될 때에 기능성을 제공하는 소프트웨어 모듈들을 저장한다. 모듈들은 컴퓨팅 시스템(500)을 위한 오퍼레이팅 시스템(540)을 포함한다. 모듈들은 본원에서 설명된 프로세스들의 전부 또는 일부 또는 그 파생부들을 수행하도록 구성되는 로봇 액세스 제어 및 관장 모듈(545)을 더 포함한다. 컴퓨팅 시스템(500)은 추가적인 기능성을 포함하는 하나 이상의 추가적인 기능적 모듈들(550)을 포함할 수 있다.

[0063] 당해 분야의 당업자는 발명의 범위로부터 이탈하지 않으면서, "시스템"이 서버, 내장된 컴퓨팅 시스템, 개인용 컴퓨터, 콘솔, 개인 정보 단말(personal digital assistant; PDA), 셀 전화, 태블릿 컴퓨팅 디바이스, 퀀텀(quantum) 컴퓨팅 시스템, 또는 임의의 다른 적당한 컴퓨팅 디바이스, 또는 디바이스들의 조합으로서 구체화될 수 있다는 것을 인식할 것이다. 위에서 설명된 기능들을 "시스템"에 의해 수행되는 것으로서 제시하는 것은 본 발명의 범위를 임의의 방식으로 제한하도록 의도된 것이 아니라, 본 발명의 많은 실시예들 중의 하나의 예를 제공하도록 의도된다. 실제로, 본원에서 개시된 방법들, 시스템들, 및 장치들은 클라우드 컴퓨팅 시스템들을 포함하는 컴퓨팅 기술과 부합하는 국소화된 및 분산된 형태들로 구현될 수 있다.

[0064] 이 명세서에서 설명된 시스템 특징들의 일부는 그 구현 독립성을 더 상세하게 강조하기 위하여 모듈들로서 제시되었다는 것이 주목되어야 한다. 예를 들어, 모듈은 맞춤형 초고밀도 집적(very large scale integration; VLSI) 회로를 또는 게이트 어레이들, 로직 칩들과 같은 기성품(off-the-shelf) 반도체들, 트랜지스터들, 또는 다른 개별 컴포넌트들을 포함하는 하드웨어 회로로서 구현될 수 있다. 모듈은 또한, 필드 프로그래밍가능 게이트 어레이들, 프로그래밍가능 어레이 로직, 프로그래밍가능 로직 디바이스들, 그래픽 프로세싱 유닛들 등과 같은 프로그래밍가능 하드웨어 디바이스들로 구현될 수 있다.

[0065] 모듈은 또한, 다양한 유형들의 프로세서들에 의한 실행을 위하여 소프트웨어로 적어도 부분적으로 구현될 수 있다. 예를 들어, 실행가능 코드의 식별된 유닛은, 예를 들어, 오브젝트(object), 프로시저(procedure), 또는 함수(function)로서 조직화(organize)될 수 있는 컴퓨터 명령들의 하나 이상의 물리적 또는 논리적 블록들을 포함할 수 있다. 그럼에도 불구하고, 식별된 모듈의 실행가능(executable)들은 함께 물리적으로 위치될 필요가 없지만, 함께 논리적으로 병합될 때, 모듈을 포함하고 모듈을 위한 기재된 목적을 달성하는 상이한 위치들에서 저장된 이질적인 명령들을 포함할 수 있다. 또한, 모듈들은 예를 들어, 발명의 범위로부터 이탈하지 않으면서, 하드 디스크 드라이브, 플래시 디바이스, RAM, 테이프, 및/또는 데이터를 저장하기 위하여 이용된 임의의 다른 이러한 비-일시적 컴퓨터-관독가능 매체일 수 있는 컴퓨터-관독가능 매체 상에서 저장될 수 있다.

[0066] 실제로, 실행가능 코드의 모듈은 단일 명령 또는 다수의 명령들일 수 있고, 몇몇 상이한 코드 세그먼트들 상에서, 상이한 프로그램 사이에서, 그리고 몇몇 메모리 디바이스들에 걸쳐 심지어 분산될 수 있다. 유사하게, 동작 데이터는 모듈들 내에서 본원에서 식별 및 예시될 수 있고, 임의의 적당한 형태로 구체화될 수 있고, 임의의 적당한 유형의 데이터 구조 내에서 조직화될 수 있다. 동작 데이터는 단일 데이터 세트로서 수집될 수 있거나, 상이한 저장 디바이스들 상부를 포함하는 상이한 위치들 상에서 분산될 수 있고, 적어도 부분적으로, 시스템 또는 네트워크 상의 단지 전자 신호들로서 존재할 수 있다.

[0067] 일부 실시예들에서, RPA 설계자 애플리케이션을 위한 관장 정보는 RPA 개발자에 의해 편집될 수 없는 파일에서 저장될 수 있다. 예를 들어, 도 6은 RPA 설계자 애플리케이션(이 경우에, UiPath Studio™)을 위한 Windows® 레지스트리 편집기(600)의 폴더(600)의 뷰를 예시하는 스크린샷이다. 경로 Computer\HKEY_CURRENT_USER\Software\UiPath 하에서, 연관된 링크(620)를 통해 이용가능한 원격 정책을 지시하는 GovernanceSource(610)로 명명된 레지스트리 키(registry key)가 (예컨대, RPA 설계자 애플리케이션이 설치될 때) 설정된다. 이 링크에서 이용가능한 정책은 컴퓨팅 시스템으로 오직 관독될 수 있고, 어떤 허가들을 관장하기 위하여 RPA 설계자 애플리케이션 또는 RPA 로봇에 의해 이용될 것이다. 이 정책은 개별적인 사용자 레벨, 그룹 레벨, 역할 레벨, 조직 레벨 등에서 정의될 수 있다. 원격 링크를 통해 정책을 포함함으로써, 정책에 대한 업데이트들이 희망될 때마다 행해질 수 있고, 업데이트된 정책은 RPA 로봇들 및/또는 RPA 설계자 애플리케이션들을 소비함으로써 자동적으로 액세스될 것이다. 이 예에서, 정책 파일은 개발 프로파일을 위한 정책들의 상이한 세트들을 정의하기 위한 능력을 제공한다. 일부 실시예들에서, 상이한 정책들은 숙련된 프로그래머들을 위하여, 그리고 이 개발자들이 상이한 필요성들을 가질 수 있으므로 프로그래밍 지식을 가지지 않을 수 있는 시민 개발자들을 위하여 설정될 수 있다. 개발 사용자들, 비즈니스 사용자들, 및 StudioPro 사용자들을 위한 규칙들을 정의하는 UiPath Studio™를 위한 일 예의 정책 파일의 부분이 이하에서 포함된다. 굵게 표시된

섹션들은 추가적인 규칙들 또는 카운터들이 이 예에서 어디에서 정의될 수 있는지를 특정한다.

```
{
  "Metadata": {
    "Version": "1.0",
  },
  "Info": {
    "Name": "Sample Policy Registry Key"
  },
  "Profiles": {
    "Development": {
      "Shell": {
        "IsFeedbackEnabled": true,
        "HideGettingStartedScreen": false
      },
      "SourceControl": {
        "CheckInBeforePublish": false
      },
      "Workflow": {
        "DockedAnnotations": true,
        "AnalyzeOnPublish": false,
        "AnalyzeOnRun": false
      }
    }
  }
}
```

[0068]

```

    },
    "PackageManager": {
        "AllowAddRemoveFeeds": false,
        "AllowEnableDisableFeeds": false,
        "AllowOrchestratorFeeds": true,
        "Feeds": [
            {
                "Name": "Local",
                "Source": "C:\\Program Files (x86)\\UiPath\\Studio\\Packages",
                "IsEnabled": true
            },
            {
                "Name": "Official",
                "Source": "https://www.myget.org/F/workflow/",
                "IsEnabled": true
            },
            {
                "Name": "Connect",
                "Source": "https://gallery.uipath.com/api/v2",
                "IsEnabled": true
            }
        ]
    },
    },

```

[0069]

```

"Analyzer": {
  "AllowEdit": false,
  "ReferencedRulesConfigFile": null,
  "EmbeddedRulesConfig": {
    "Rules": [
      {
        "Id": "ST-NMG-001",
        "IsEnabled": true,
        "Parameters": [
          {
            "Name": "Regex",
            "Value": null
          }
        ],
        "ErrorLevel": "Off"
      },
      <여기에서 정의된 추가적인 규칙들>
    ],
    "Counters": [
      {
        "Id": "ST-NMG-001",
        "IsEnabled": true,
        "Parameters": []
      }
    ]
  }
}

```

[0070]

```

    },
    <여기에서 정의된 추가적인 카운터들>
  ]
}
}
},
"Business": {
  "Shell": {
    "IsFeedbackEnabled": true,
    "HideGettingStartedScreen": false
  },
  "SourceControl": {
    "CheckInBeforePublish": false
  },
  "Workflow": {
    "DockedAnnotations": true,
    "AnalyzeOnPublish": false,
    "AnalyzeOnRun": false
  },
  "PackageManager": {
    "AllowAddRemoveFeeds": false,
    "AllowEnableDisableFeeds": false,
    "AllowOrchestratorFeeds": true,

```

[0071]

```

"Feeds": [
  {
    "Name": "Local",
    "Source": "C:\\Program Files (x86)\\UiPath\\Studio\\Packages",
    "IsEnabled": true
  },
  {
    "Name": "Official",
    "Source": "https://www.myget.org/F/workflow/",
    "IsEnabled": true
  },
  {
    "Name": "Connect",
    "Source": "https://gallery.uipath.com/api/v2",
    "IsEnabled": true
  }
]
},
"Analyzer": {
  "AllowEdit": false,
  "ReferencedRulesConfigFile": null,
  "EmbeddedRulesConfig": {
    "Rules": [

```

[0072]

```

{
  "Id": "ST-NMG-001",
  "IsEnabled": true,
  "Parameters": [
    {
      "Name": "Regex",
      "Value": null
    }
  ],
  "ErrorLevel": "Off"
},
<여기에서 정의된 추가적인 규칙들>
],
"Counters": [
  {
    "Id": "ST-NMG-001",
    "IsEnabled": true,
    "Parameters": []
  },
  <여기에서 정의된 추가적인 카운터들>
]
}
}

```

[0073]

```

    },
    "StudioPro": {
      "Shell": {
        "IsFeedbackEnabled": true,
        "HideGettingStartedScreen": false
      },
      "SourceControl": {
        "CheckInBeforePublish": false
      },
      "Workflow": {
        "DockedAnnotations": true,
        "AnalyzeOnPublish": false,
        "AnalyzeOnRun": false
      },
      "PackageManager": {
        "AllowAddRemoveFeeds": false,
        "AllowEnableDisableFeeds": false,
        "AllowOrchestratorFeeds": true,
        "Feeds": [
          {
            "Name": "Local",
            "Source": "C:\\Program Files (x86)\\UiPath\\Studio\\Packages",
            "IsEnabled": true
          }
        ]
      }
    }
  }
}

```

[0074]

```

    },
    {
      "Name": "Official",
      "Source": "https://www.myget.org/F/workflow/",
      "IsEnabled": true
    },
    {
      "Name": "Connect",
      "Source": "https://gallery.uipath.com/api/v2",
      "IsEnabled": true
    }
  ]
},
"Analyzer": {
  "AllowEdit": false,
  "ReferencedRulesConfigFile": null,
  "EmbeddedRulesConfig": {
    "Rules": [
      {
        "Id": "ST-NMG-001",
        "IsEnabled": true,
        "Parameters": [
          {

```

[0075]

```
"Name": "Regex",  
    "Value": null  
},  
],  
    "ErrorLevel": "Off"  
},  
<여기에서 정의된 추가적인 규칙들>  
],  
    "Counters": [  
        {  
            "Id": " ST-NMG-001",  
            "IsEnabled": true,  
            "Parameters": []  
        },  
<여기에서 정의된 추가적인 규칙들>  
    ]  
}  
}  
}  
}  
}
```

[0076]

[0077] 여기에서 도시된 규칙들 및 카운터들은 각각의 사용자 유형에 대하여 동일하지만, 이것은 반드시 그러할 필요가 없고, 전형적으로, 실제적인 구현예에서 그렇지 않을 것이라는 것이 주목되어야 한다. 또한, 제어 및 관장 정책 규칙들을 포함하는 파일이 일부 실시예들에서 레지스트리 엔트리를 이용하여 획득되지만, 어떤 실시예들에서는, 파일 또는 다른 정책 정보 포맷이 지휘자 애플리케이션에 의해 전달된다는 것이 주목되어야 한다. 어떤 실시예들에서, 서버 UI는 관리자가 접속을 통해(예컨대, .json 페이로드로서) 전달될 정책을 정의하는 것을 허용한다. 그러나, 이 정책은 반드시, 관리자가 일부 실시예들에서 명시적으로 저술(author)하는 "파일"이 아닐 수 있다.

[0078] 일부 실시예들에서, 맞춤형 관장 규칙들은 작업흐름 분석기 프로세스의 일부로서 개발될 수 있다. 예를 들어, 일부 실시예들에서, RPA 설계자 애플리케이션을 위한 설치 스크립트(installation script)는 맞춤형 관장 규칙들을 사용자의 컴퓨팅 시스템에 설치할 수 있다. 예를 들어, 파일 SampleGovernanceRules.dll(710)이 UiPath Studio™을 위한 규칙들 폴더에서 설치되는 Windows Explorer® UI의 부분을 도시하는 도 7의 스크린샷(700)을 참조한다. SampleGovernanceRules.dll(710)은 프로그램 파일들 폴더에서 위치되므로, 오직 관리자가 그것을 수정할 수 있다. 그 또는 그녀의 머신 상에서 관리 특권들을 갖지 않는 최종 사용자는 그렇게 할 수 없다. 이에 따라, 이러한 특권들을 갖지 않는 최종 사용자는 관장 정책을 가질 것이고, 그것을 기각하거나 변경할 수 없을

것이다.

- [0079] 도 8은 본 발명의 실시예에 따라, 패키지 관리 인터페이스(800)를 예시하는 스크린샷이다. 이 실시예에서는, 개발자를 위한 패키지 피드들이 제재되었다. 더 구체적으로, 개발자는 공식적인 패키지 피드(810)에 대해 오직 액세스하고, 개발자는 공식적인 패키지 피드(810)를 인에이블하거나 디스에이블할 수 없다. 개발자는 또한, 사용자-정의된 패키지를 위한 명칭(820) 또는 소스(source)(830)를 기입하거나, 사용자-정의된 패키지를 추가하기 위하여 추가 버튼(840)을 이용하거나, 수정된 패키지 설정들을 저장하기 위하여 저장 버튼(850)을 이용하는 것을 할 수 없다. 이것은 개발자들이 조직이 개발자들에게 원하지 않는 어떤 곳으로부터 활동 패키지들을 얻는 것을 방지한다.
- [0080] 일부 조직들은 개발자들이 개발 동안에 민감한 정보를 RPA 설계자 애플리케이션의 제공자에게 잠재적으로 누설하는 것에 대한 우려들을 가질 수 있다. 따라서, 일부 실시예들에서, 피드백 전송 기능성은 정책의 사항으로서 RPA 설계자 애플리케이션에서 디스에이블된다. 예를 들어, 디스에이블된 피드백 전송 아이콘 및 이것이 그러하다는 것을 표시하는 메시지(910)를 갖는 RPA 설계자 애플리케이션의 상부 우측 부분(900)을 예시하는 스크린샷인 도 9를 참조한다.
- [0081] 도 10a는 본 발명의 실시예에 따라, 작업흐름 분석기 설정들 인터페이스(1000)를 예시하는 스크린샷이다. 일부 실시예들에서, 작업흐름 분석기 설정들은 정책 구성 파일로부터 취해진다. 도 10a에서 보여질 수 있는 바와 같이, 다양한 규칙들은 개개의 코드, 명칭, 범위, 및 디폴트 액션을 갖는 규칙들 페인(pane)(1002)에서 리스팅된다. 선택된 규칙에 대한 특정 설정들 옵션들은 규칙 설정들 페인(1004)에서 도시된다.
- [0082] 이 실시예에서, 4 개의 유형들의 액세스 제약들 - app/URL 제약들(1010), 패키지 제약들(1020), 활동 제약들(1030), 및 활동 속성 요건들(1040)이 있다. 그러나, 임의의 수 및/또는 유형의 액세스 제약들은 발명의 범위로부터 이탈하지 않으면서 이용될 수 있다. App/URL 제약들(1010)은 RPA 설계자 애플리케이션의 사용자가 어느 애플리케이션들 및/또는 URL들로 작업하도록 허용되고 및/또는 허용되지 않는지를 제어한다. 예를 들어, 도 10b를 참조하면, 금지된 애플리케이션들은 임의의 것이 금지될 경우에 금지된 애플리케이션들 필드(1012)에서 도시될 것이고, 허용된 애플리케이션들은 허용된 애플리케이션들 필드(1014)에서 도시된다. 사용자가 스크롤 다운(scroll down)할 경우에, 금지된 URL들 및 허용된 URL들은 또한, 개개의 필드들에서 도시될 것이다. 당연히, 주어진 애플리케이션 또는 URL은 양자 모두 금지될(즉, 블랙리스트됨) 수 없고 허용될(즉, 화이트리스트됨) 수 없다. 어떤 실시예들에서, 허용된 애플리케이션들이 특정될 경우에, 사용자는 그 애플리케이션들로 오직 작업할 수 있다. 이 실시예에서, 이 필드들은 오직 판독된다.
- [0083] 패키지 제약들(1020)은 어느 패키지들이 금지되는지를 제어한다. 예를 들어, 도 10c를 참조하면, 금지된 패키지들은 금지된 패키지들 필드(1022)에서 리스팅될 수 있다. 이것은 조직이 사용자가 일부 실시예들에서 액세스하는 것을 원하지 않는 패키지들의 카테고리들을 한정할 수 있다. 어떤 실시예들에서, 허용된 패키지들은 금지된 패키지들 필드(1022)에 추가적으로 또는 금지된 패키지들 필드(1022) 대신에 리스팅될 수 있다. 금지된 패키지들 필드(1022)는 이 실시예에서 오직 판독된다.
- [0084] 활동 제약들(1030)은 사용자가 어느 활동들을 이용하도록 허용되는지 및/또는 이용하도록 허용되지 않는지를 제어한다. 예를 들어, 도 10d를 참조하면, 금지된 활동들은 임의의 것이 금지될 경우에 금지된 활동들 필드(1032)에서 도시될 것이고, 허용된 활동들은 임의의 활동들이 허용된 것으로서 리스팅될 경우에 허용된 활동들 필드(1034)에서 도시될 것이다. 이에 따라, 제약들이 전체 패키지의 레벨에서 희망되는 것이 아니라, 허가되지 않아야 하는 어떤 활동들이 있고 및/또는 오직 어떤 활동들이 허용되어야 할 경우에, 이것은 활동 제약들(1030)을 통해 특정될 수 있다. 이 실시예에서, 이 필드들은 오직 판독된다.
- [0085] 일부 실시예들에서, 활동들이 허용되고 및/또는 불허되는 것으로서 특정되지 않을 경우에, 이것은 모든 활동들이 허용되는 것으로 디폴트된다. 어떤 실시예들에서는, 양자 모두가 아니라, 오직 허용되거나 불허되는 것이 집행된다. 예를 들어, 이러한 실시예들에서, 양자의 허용된 및 불허된 활동들이 포함될 경우에, 허용된 활동들을 구체적으로 특정하는 것은 더 한정적인 것으로 고려되므로, "허용된" 세트가 우위를 취할 수 있고 불허된 세트는 무시될 수 있다.
- [0086] 활동 속성 요건들(1040)은 조직이 맞춤형 규칙들을 정의하는 것을 허용한다. 도 10a로 돌아가면, 규칙 구성 필드(1042)는 "Property:*Draft*,Activity:*MailX,Value:True"의 값을 가진다. 이 예에서, UiPath StudioX™에서의 Outlook®로 작업하는 MailX 활동들에 대하여, MailX 활동들은 참(True)으로 설정된 드래프트 속성들을 가져야 한다. 다른 제약 유형들에서와 같이, 규칙 구성 필드(1042)는 사용자를 위하여 오직 판독된다. 도 10a

내지 도 10d에서 보여질 수 있는 바와 같이, 일부 규칙들은 에러들을 디폴트 액션으로서 제공하도록 설정되고, 다른 것들은 경고들을 제공하도록 설정된다.

- [0087] 도 11의 스크린샷에서 도시된 바와 같이, RPA 설계자 애플리케이션 설정들 인터페이스(1100)에서, 다양한 설정들은 RPA 작업흐름 설계를 위하여 집행될 수 있다. 이 경우에, 작업흐름은 작동 전 및 발행 전의 양자 모두에 작업흐름 분석기를 통과해야 한다. 이 집행은 각각 설정들(1110 및 1120)에서 보여질 수 있다. 이것은 작업흐름 분석기 설정들 인터페이스(1000)에서의 규칙들로서의 관장 정책들 세트에 대하여, 작업흐름은 비준수성의 경우에 에러를 투하하도록 설치되는 모든 규칙들을 준수해야 한다.
- [0088] 도 12a 및 도 12b는 본 발명의 실시예에 따라, RPA 작업흐름(1200)을 예시하는 스크린샷들이다. 사용자는 링크(1210)(여기서, "https://www.uipath.com/")에서 콘텐츠를 자동화하기 위하여 노력하고 있다. 이것은 작업흐름 분석기의 규칙들에 기초한 허가된 URL이 아닌 경우를 고려한다. 또한, "save as draft(드래프트로서 저장)" 체크박스(1220)가 선택될 것을 요구하는 규칙이 존재하는 경우를 고려한다. 사용자가 작업흐름을 유효성확인(validate)하는 것을 추구할 때, 에러 리스트 인터페이스(1300)가 이 실시예에서 나타난다. 도 13을 참조한다. 에러 리스트 인터페이스(1300)는 "드래프트로서 저장" 속성이 조직의 가이드라인들을 충족시키지 않는다는 것과, https://www.uipath.com/이 UI 자동화를 위하여 조직에 의해 정의된 허용된 애플리케이션이 아니라는 것을 표시한다.
- [0089] 상기에 따라, 정책들은 일부 실시예들에서, 사용자의 기술적 정교함에 기초하여 정의될 수 있다. 예를 들어, 더 복잡한 규칙들은 임의의 레벨의 프로그래밍 지식 또는 심지어 그것을 결여한 개인들에 의해 이용될 수 있는 UiPath StudioTM이 아니라, 프로그래밍 지식을 갖는 RPA 개발자들에 의해 이용되는 경향이 있는 UiPath StudioTM을 위하여 정의될 수 있다. 예를 들어, 기술적으로 더 정교한 RPA 설계자 애플리케이션의 사용자들은 더 많은 패키지 피드들을 액세스할 수 있고, 피드들을 디스에이블하기 위한 능력을 가질 수 있고, 규칙들의 더 풍부한 세트들을 가질 수 있는 등과 같다.
- [0090] 도 14의 작업흐름 분석기 인터페이스(1400)를 고려한다. 이 경우에, 맞춤형 규칙들을 포함하는 규칙들의 풍부한 세트는 RPA 개발자를 위하여 제공된다. 맞춤형 패키지 버전 요건들 규칙(1410)이 구현된다. 메일(mail) 활동들 "UiPath.Mail.Activities"은 규칙 구성 필드(1412)에서 1.6.0의 최소 버전 및 1.6.0의 최대 버전을 가져야 하고, 에러는 또 다른 버전이 참조될 경우에 투하될 것이다. 이것은 UiPath.Mail.Activities가 정확하게 이 버전을 가져야 한다는 것을 요구한다. 또한, 사전해제 허용 패키지들 필드(1414)는 "True"로 설정된다.
- [0091] 상기에 따라, 일부 실시예들에서, 관장은 RPA 설계자 애플리케이션을 위한 설치 스크립트의 일부로서 가능하게 된다. 어떤 실시예들에서, RPA 설계자 애플리케이션이 관장 정책들의 현재의 버전을 항상 가지도록, 원격 정책 파일은 등록 키를 통해 설정될 수 있다. 대안적으로, 정책 파일은 RPA 설계자 애플리케이션이 상주하는 로컬 컴퓨팅 시스템 상의 보호된 위치에 있을 수 있다. 일부 실시예들에서, 맞춤형 규칙들은 설치 스크립트의 일부로서 보호된 "규칙들" 폴더로 전개된다.
- [0092] 일부 실시예들에서, 별도의 정책들은 기술적으로 정통한 사용자들 및 실질적인 프로그래밍 지식을 갖지 않는 사용자들을 위하여 집행된다. 어떤 실시예들에서, 상이한 정책들은 주어진 사용자, 사용자의 역할, 사용자의 그룹 등에 기초하여 적용된다. 일부 실시예들에서, 사용자들은 비호환 RPA 작업흐름들을 작동시키거나 발행할 수 없다.
- [0093] 도 15는 본 발명의 실시예에 따라, 로봇 실행 시에 RPA를 위한 액세스 제어 및 관장을 수행하기 위한 시스템(1500)을 예시하는 아키텍처 다이어그램이다. 시스템(1500)은 데스크톱 컴퓨터(1502), 태블릿(1504), 및 스마트폰(1506)과 같은 사용자 컴퓨팅 시스템들을 포함한다. 그러나, 스마트 시계들, 랩톱 컴퓨터들, 사물 인터넷(Internet-of-Things; IoT) 디바이스들, 차량 컴퓨팅 시스템들 등을 포함하지만 이것으로 제한되지는 않는 임의의 희망된 컴퓨팅 시스템이 발명의 범위로부터 이탈하지 않으면서 이용될 수 있다.
- [0094] 각각의 컴퓨팅 시스템(1502, 1504, 1506)은 하나 이상의 자동화들을 실행하는, 그 상에서 작동되는 RPA 로봇(1510)을 가진다. 그러나, 어떤 실시예들에서, 컴퓨팅 시스템들(1502, 1504, 1506)은 상이한 로봇들을 실행할 수 있다. 그 프로세스를 실행할 때, RPA 로봇은 네트워크(1520)(예컨대, 로컬 영역 네트워크(local area network; LAN), 이동 통신 네트워크, 위성 통신 네트워크, 인터넷, 그 임의의 조합 등)를 통해 서버(1530)로부터 어떤 정보를 요청한다(예컨대, 요청들을 서버(1530) 상에서 작동되는 지휘자 애플리케이션으로 전송함). 예를 들어, RPA 로봇(1510)은 발명의 범위로부터 이탈하지 않으면서, 데이터를 전송하거나 수신하기 위하여(예컨대, 데이터 큐(data queue)로부터 푸싱/풀링), 보안성 있게 저장된 크리덴셜(credential)/자산 정보(예컨대, 사

용자명들/패스워드들, 액세스 토큰들 등)를 취출(retrieve)하기 위하여, "하트비트 스테이터스"(예컨대, RPA 로봇(1510)이 올바르게 작동하고 있다는 것)를 보고하기 위하여, "편성 커맨드들"(예컨대, "이것을 작동", "실행 포기/취소" 등)을 체크하기 위하여, 또는 임의의 다른 적당한 통신들을 위하여, 서버(1530)와 통신할 수 있다. 일부 실시예들에서, 서버(1530)는 지휘자 애플리케이션을 작동시킬 수 있고, 데이터는 하트비트 메시지의 일부로서 주기적으로 전송될 수 있다. 서버(1530)는 하나 이상의 파일들(1532)로부터, 데이터베이스(1540)로부터, 또는 양자로부터 주어진 로봇을 위한 액세스 제어 및 관장 규칙들을 획득한다.

[0095] 서버(1530)는 액세스 제어 및 관장 규칙들에 대하여 로봇(1510)에 의해 수행되어야 할 액션(들) 및/또는 RPA 로봇에 의해 요청된 정보를 체크한다. 로봇(1510)이 액션을 취하거나 정보를 획득하도록 허가되지 않을 경우에, 서버(1530)는 요청된 정보를 로봇(1510)에 제공하고, 및/또는 액션이 수락가능하다는 것을 표시하는 정보를 개개의 컴퓨팅 시스템 상의 유효성확인 애플리케이션(validation application)으로 전송한다. 그렇지 않을 경우에, 서버(1530)는 요청된 정보를 로봇(1510)에 제공하지 않고, 및/또는 로봇(1510)에 의해 시도된 액션이 수락가능하지 않다는 것을 표시하는 정보를 개개의 컴퓨팅 시스템 상의 유효성확인 애플리케이션으로 전송한다. 어떤 실시예들에서, 유효성확인 애플리케이션은 로봇(1510)과 연관된 프로세스를 종료시킬 수 있다.

[0096] 일부 실시예들에서, 허가되지 않은 액션은 RPA 로봇이 아니라, 인간 사용자를 위하여 허가된 액션일 수 있다. 어떤 실시예들에서, 서버(1530)는 로봇(1510)이 하나 이상의 레거시 시스템들을 위하여 충분히 길게 일시정지하는 것을 검증할 수 있고, 로봇에 의해 요청된 정보를 획득하는 것, 또는 지연 주기가 만료할 때까지 로봇으로부터의 새로운 정보 요청들을 수락하는 것을 지연시킬 수 있다. 어떤 실시예들에서, 관장은 시간 집행을 설계하는 것에 추가적으로, 실행시간에 집행된다. 정책들은 일부 실시예들에서, 컴퓨팅 시스템들(1502, 1504, 1506)이 예를 들어, 서버(1530)의 서버-측 지휘자 애플리케이션에 접속할 때에, 컴퓨팅 시스템들(1502, 1504, 1506)로 자동적으로 전송될 수 있다. RPA 설계자 애플리케이션은 RPA 로봇들이 일부 실시예들에서, 이 정책들을 획득하고 이를 준수하여 동작하는 것을 강제하는 코드를 RPA 로봇들 내로 자동적으로 삽입할 수 있다.

[0097] 도 16은 본 발명의 실시예에 따라, RPA 설계자 애플리케이션을 위한 로봇 액세스 제어 및 관장을 수행하기 위한 프로세스(1600)를 예시하는 플로우차트이다. 프로세스는 1605에서, 컴퓨팅 시스템의 레지스트리 엔트리로부터 액세스 제어 및 관장 정책 규칙들을 포함하는 파일에 대한 링크를 결정하는 것, 및 1610에서 결정된 링크를 이용하여 파일을 다운로드하는 것으로 시작된다. 일부 실시예들에서, 액세스 제어 및 관장 정책 규칙들은 RPA 설계자 애플리케이션을 위한 설치 스크립트를 통해 구현된다. 작업흐름 분석기는 1615에서, 파일로부터 RPA 설계자 애플리케이션을 위한 액세스 제어 및 관장 정책 규칙들을 판독한다.

[0098] 일부 실시예들에서, 액세스 제어 및 관장 정책 규칙들은 어느 애플리케이션들 및/또는 URL들이 자동화될 수 있고 및/또는 자동화되지 않을 수 있는지에 대한 제어들, 어떤 활동들이 RPA 작업흐름에서 이용될 수 있고 및/또는 이용되지 않을 수 있는지에 대한 제어들, 어떤 패키지들이 RPA 작업흐름을 위하여 이용될 수 있고 및/또는 이용되지 않을 수 있는지에 대한 제어들, 또는 그 조합을 포함한다. 어떤 실시예들에서, 액세스 제어 및 관장 정책 규칙들은 조직, 역할, 그룹, 개별적인 개발자, 또는 그 조합에 기초하여 RPA 설계자 애플리케이션을 위하여 정의된다. 일부 실시예들에서, 액세스 제어 및 관장 정책 규칙들은 RPA 설계자 애플리케이션이 실행되는 컴퓨팅 시스템의 오퍼레이팅 시스템에 의해 집행된 바와 같은 RPA 설계자 애플리케이션의 사용자에 의해 수정될 수 없다. 어떤 실시예들에서, 액세스 제어 및 관장 정책 규칙들은 하나 이상의 애플리케이션 및/또는 URL 제약들, 하나 이상의 패키지 제약들, 하나 이상의 활동 제약들, 하나 이상의 활동 속성 요건들, 또는 그 조합을 포함한다.

[0099] 일부 실시예들에서는, 1620에서, 다양한 인터페이스들이 디스플레이될 수 있고, 비허가된 사용자 수정들이 방지될 수 있다. 예를 들어, RPA 작업흐름의 활동들에 의해 액세스될 수 있는 패키지들을 포함하는 패키지 관리 인터페이스가 디스플레이될 수 있고, RPA 설계자 애플리케이션의 사용자는 액세스 제어 및 관장 정책 규칙들에 기초하여, 허가된 패키지들을 추가하거나 허가되지 않은 새로운 패키지들을 추가하는 것이 방지될 수 있다. 또 다른 예로서, 액세스 제어 및 관장 정책 규칙들을 리스팅하는 작업흐름 분석기 설정들 인터페이스가 디스플레이될 수 있고, RPA 설계자 애플리케이션의 사용자는 액세스 제어 및 관장 정책 규칙들을 수정하는 것이 방지될 수 있다.

[0100] RPA 설계자 애플리케이션의 RPA 작업흐름의 활동들은 1625에서, 액세스 제어 및 관장 정책 규칙들에 대하여 분석된다. 일부 실시예들에서, RPA 작업흐름의 활동들의 분석은 RPA 작업흐름 활동에서 액세스되어야 할 하나 이상의 라이브러리들이 화이트리스트에서 포함되거나 블랙리스트에서 포함되지 않는지 여부를 검증하는 것을 포함한다. 유효성확인이 1630에서 성공할 경우에(즉, RPA 작업흐름 활동들이 모든 요구된 액세스 제어 및 관장 정

책 규칙들을 준수함), 1635에서, RPA 작업흐름을 구현하는 RPA 로봇이 생성되거나 RPA 작업흐름이 발행된다. 그러나, 유효성확인이 1630에서 실패할 경우에(즉, RPA 작업흐름의 하나 이상의 분석된 활동들이 액세스 제어 및 관장 정책 규칙들을 위반함), RPA 로봇의 생성 또는 RPA 작업흐름의 발행은 RPA 작업흐름이 액세스 제어 및 관장 정책 규칙들을 만족시킬 때까지 방지된다. 이 실시예에서, RPA 작업흐름 분석기/RPA 설계자 애플리케이션은 1640에서, 사용자가 RPA 작업흐름에 대한 수정들을 행하는 것을 대기한다. 프로세스는 그 다음으로, RPA 작업흐름의 수정된 활동들을 분석하기 위하여 단계(1625)로 복귀한다. 일부 실시예들에서는, 수정되었던 활동들만이 다시 분석된다.

[0101] 도 17은 본 발명의 실시예에 따라, 실행 RPA 로봇에 대하여 RPA를 위한 로봇 액세스 제어 및 관장을 수행하기 위한 프로세스(1700)를 예시하는 플로우차트이다. 프로세스는 1710에서, 액세스 제어 및 관장 규칙들에 대하여 RPA 로봇에 의해 수행되어야 할 액션(들) 및/또는 RPA 로봇에 의해 요청된 정보를 체크하는 것으로 시작된다. 체크는 예를 들어, 클라이언트-측 또는 서버-측 유효성확인 애플리케이션에 의해 수행될 수 있다. RPA 로봇이 1720에서, 액션(들)을 취하고 및/또는 정보를 획득하도록 허가될 경우에, 1730에서, 요청된 정보가 RPA 로봇에 제공되고, 및/또는 액션(들)이 인가된다. 유효성확인 애플리케이션은 그 다음으로, 1740에서, 다음 RPA 로봇 통신이 수신될 때까지 대기한다.

[0102] RPA 로봇이 1720에서, 액션(들)을 취하고 정보를 획득하도록 허가되지 않을 경우에, 유효성확인 애플리케이션은 1750에서, RPA 로봇이 정보를 획득하고 및/또는 액션(들)을 취하는 것을 방지한다. 일부 실시예들에서는, 1760에서, 정보에 대한 액세스 및/또는 액션(들)이 허가되지 않는다는 통지가 RPA 로봇으로 전송될 수 있다. 어떤 실시예들에서, 유효성확인 애플리케이션은 1770에서, (예컨대, RPA 로봇과 연관된 프로세스를 종결시킴으로써) RPA 로봇의 실행을 자동적으로 종료시킬 수 있다.

[0103] 일부 실시예들에서, 허가되지 않은 액션은 RPA 로봇이 아니라, 인간 사용자를 위하여 허가될 액션일 수 있다. 어떤 실시예들에서, 유효성확인 애플리케이션은 RPA 로봇이 하나 이상의 레거시 시스템들을 위하여 충분히 길게 일시정지하는 것을 검증할 수 있고, RPA 로봇에 의해 요청된 정보를 획득하는 것, 또는 지연 주기가 만료할 때까지 RPA 로봇으로부터의 새로운 정보 요청들을 수락하는 것을 지연시킬 수 있다.

[0104] 도 16 및 도 17에서 수행된 프로세스 단계들은 프로세서(들)가 본 발명의 실시예들에 따라 도 16 및 도 17에서 설명된 프로세스(들)의 적어도 일부를 수행하기 위한 명령들을 인코딩하는 컴퓨터 프로그램에 의해 수행될 수 있다. 컴퓨터 프로그램은 비-일시적 컴퓨터-판독가능 매체 상에서 구체화될 수 있다. 컴퓨터-판독가능 매체는 하드 디스크 드라이브, 플래시 디바이스, RAM, 테이프, 및/또는 데이터를 저장하기 위하여 이용된 임의의 다른 이러한 매체 또는 매체들의 조합일 수 있지만, 이것으로 제한되지는 않는다. 컴퓨터 프로그램은 컴퓨터-판독가능 매체 상에서 또한 저장될 수 있는, 도 16 및 도 17에서 설명된 프로세스 단계들의 전부 또는 일부를 구현하기 위하여 컴퓨팅 시스템의 프로세서(들)(예컨대, 도 5의 컴퓨팅 시스템(500)의 프로세서(들)(510))를 제어하기 위한 인코딩된 명령들을 포함할 수 있다.

[0105] 컴퓨터 프로그램은 하드웨어, 소프트웨어, 또는 하이브리드 구현으로 구현될 수 있다. 컴퓨터 프로그램은, 서로 통신 동작하고 있고, 정보 또는 명령들을 디스플레이로 전달하도록 설계되는 모듈들로 구성될 수 있다. 컴퓨터 프로그램은 범용 컴퓨터, ASIC, 또는 임의의 다른 적당한 디바이스 상에서 동작하도록 구성될 수 있다.

[0106] 본 발명의 다양한 실시예들의 컴포넌트들은 본원에서의 도면들에서 일반적으로 설명되고 예시된 바와 같이, 폭넓게 다양한 상이한 구성들로 배열 및 설계될 수 있다는 것이 용이하게 이해될 것이다. 이에 따라, 본 발명의 실시예들의 상세한 설명은 첨부된 도면들에서 표현된 바와 같이, 청구된 바와 같은 발명의 범위를 제한하도록 의도된 것이 아니라, 발명의 선택된 실시예들을 단지 대표한다.

[0107] 이 명세서의 전반에 걸쳐 설명된 발명의 특징들, 구조들, 또는 특성들은 하나 이상의 실시예들에서 임의의 적당한 방식으로 조합될 수 있다. 예를 들어, "어떤 실시예들", "일부 실시예들", 또는 유사한 언어에 대한 이 명세서의 전반에 걸친 참조는 실시예와 관련하여 설명된 특정한 특징, 구조, 또는 특성이 본 발명의 적어도 하나의 실시예 내에 포함된다는 것을 의미한다. 이에 따라, 이 명세서의 전반에 걸친 어구들 "어떤 실시예들에서", "일부 실시예에서", "다른 실시예들에서", 또는 유사한 언어의 출현들은 실시예들의 동일한 그룹을 반드시 모두 지칭하는 것은 아니고, 설명된 특징들, 구조들, 또는 특성들은 하나 이상의 실시예들에서 임의의 적당한 방식으로 조합될 수 있다.

[0108] 특징들, 장점들, 또는 유사한 언어에 대한 이 명세서의 전반에 걸친 참조는 본 발명으로 실현될 수 있는 특징들 및 장점들의 전부가 발명의 임의의 단일 실시예 내에 있어야 하거나 있다는 것을 암시하지는 않는다는 것이 주

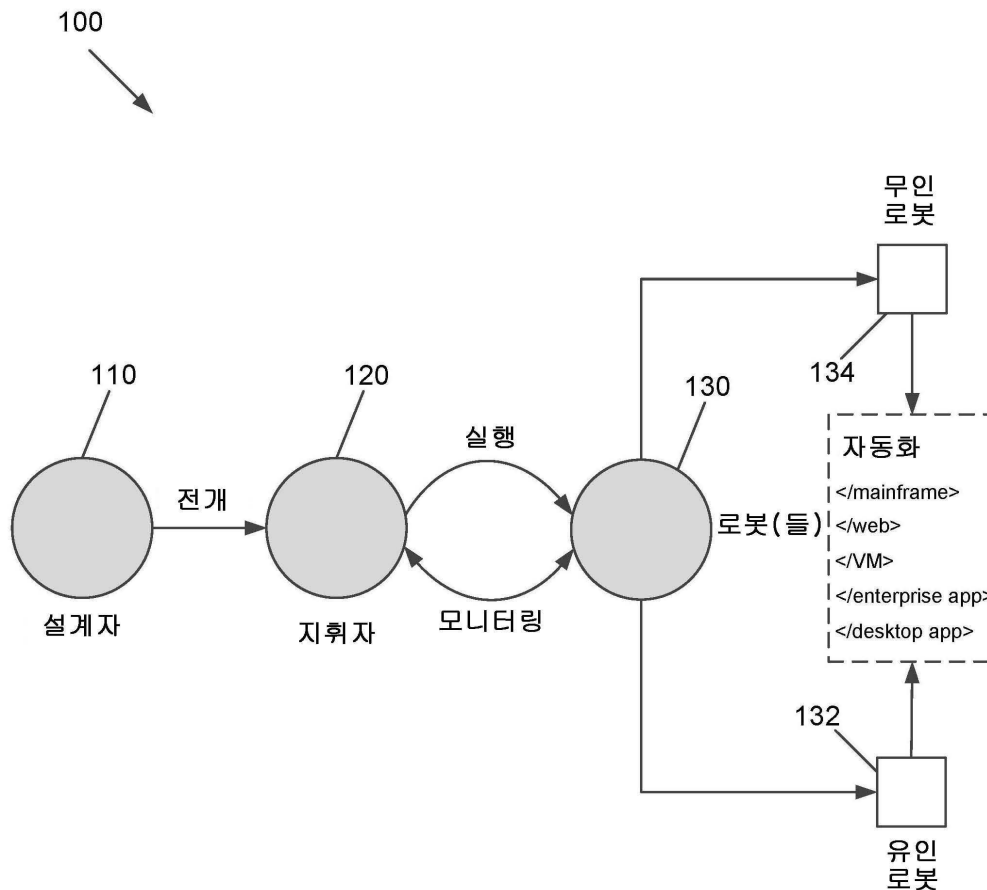
목되어야 한다. 오히려, 특징들 및 장점들을 지칭하는 언어는 실시예와 관련하여 설명된 특정 특징, 장점, 또는 특성이 본 발명의 적어도 하나의 실시예 내에 포함된다는 것을 의미하도록 이해된다. 이에 따라, 이 명세서의 전반에 걸친 특징들 및 장점들, 및 유사한 언어의 논의는 동일한 실시예를 지칭할 수 있지만, 반드시 그러한 것은 아니다.

[0109] 또한, 발명의 설명된 특징들, 장점들, 및 특성들은 하나 이상의 실시예들에서 임의의 적당한 방식으로 조합될 수 있다. 관련 기술분야의 당업자는 발명이 특정한 실시예의 특정 특징들 또는 장점들 중의 하나 이상 없이 실시될 수 있다는 것을 인식할 것이다. 다른 사례들에서, 추가적인 특징들 및 장점들은 발명의 모든 실시예들에서 존재하지 않을 수 있는 어떤 실시예들에서 인식될 수 있다.

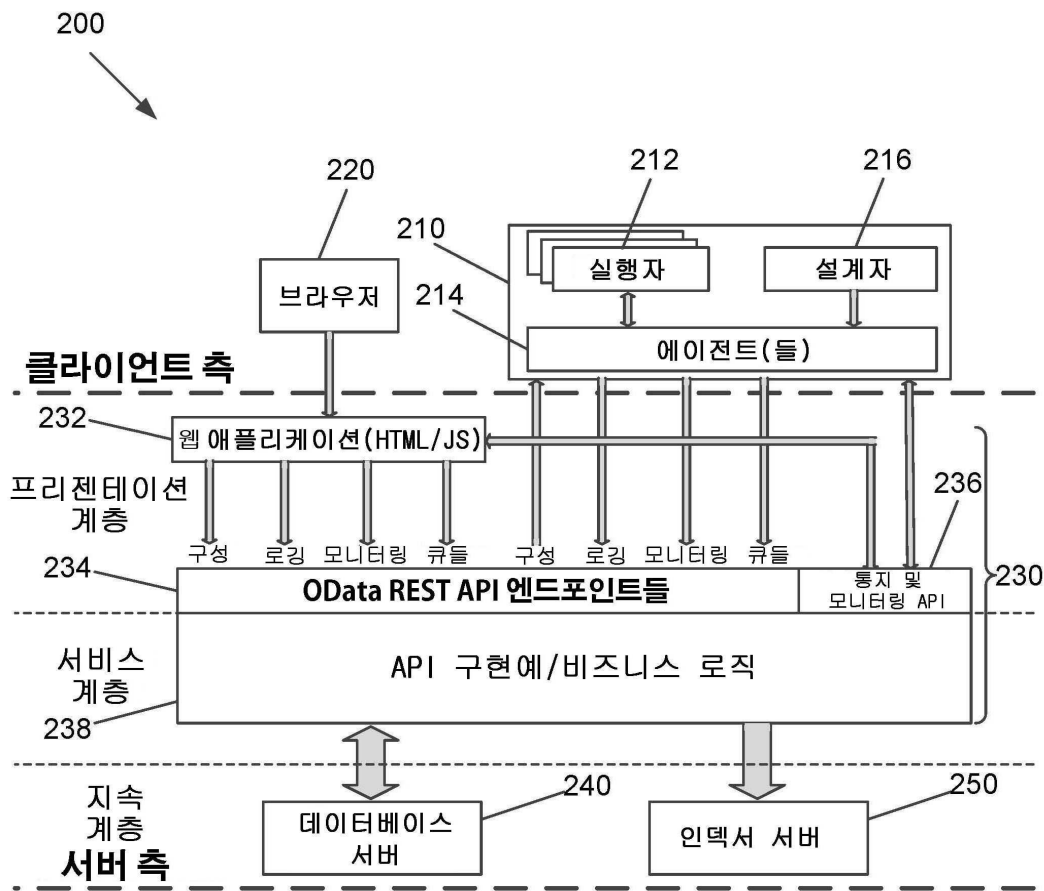
[0110] 당해 분야의 당업자는 위에서 논의된 바와 같은 발명이 상이한 순서인 단계들로, 및/또는 개시되는 것들과 상이한 구성들인 하드웨어 엘리먼트들로 실시될 수 있다는 것을 용이하게 이해할 것이다. 그러므로, 발명은 이 바람직한 실시예들에 기초하여 설명되었지만, 발명의 사상 및 범위 내에서 유지되면서, 어떤 수정들, 변동들, 및 대안적인 구성들이 분명할 것이라는 것이 당해 분야의 당업자들에게 분명할 것이다. 그러므로, 발명의 한계들 및 경계들을 결정하기 위하여, 첨부된 청구항들에 대해 참조가 행해져야 한다.

도면

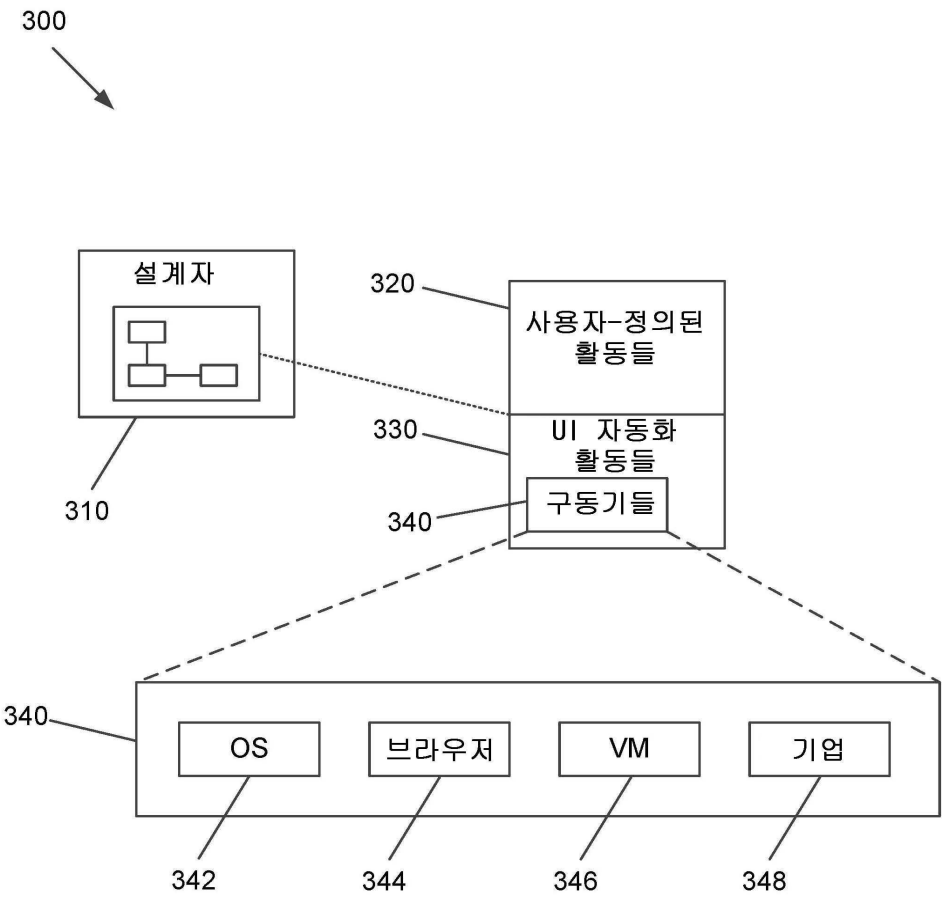
도면1



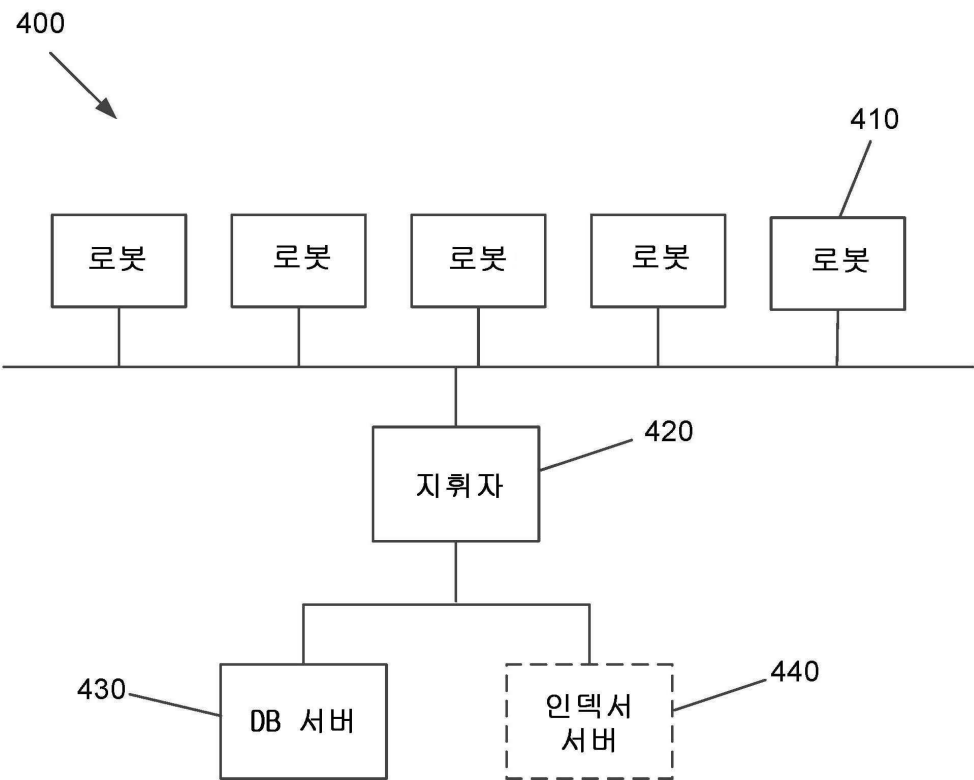
도면2



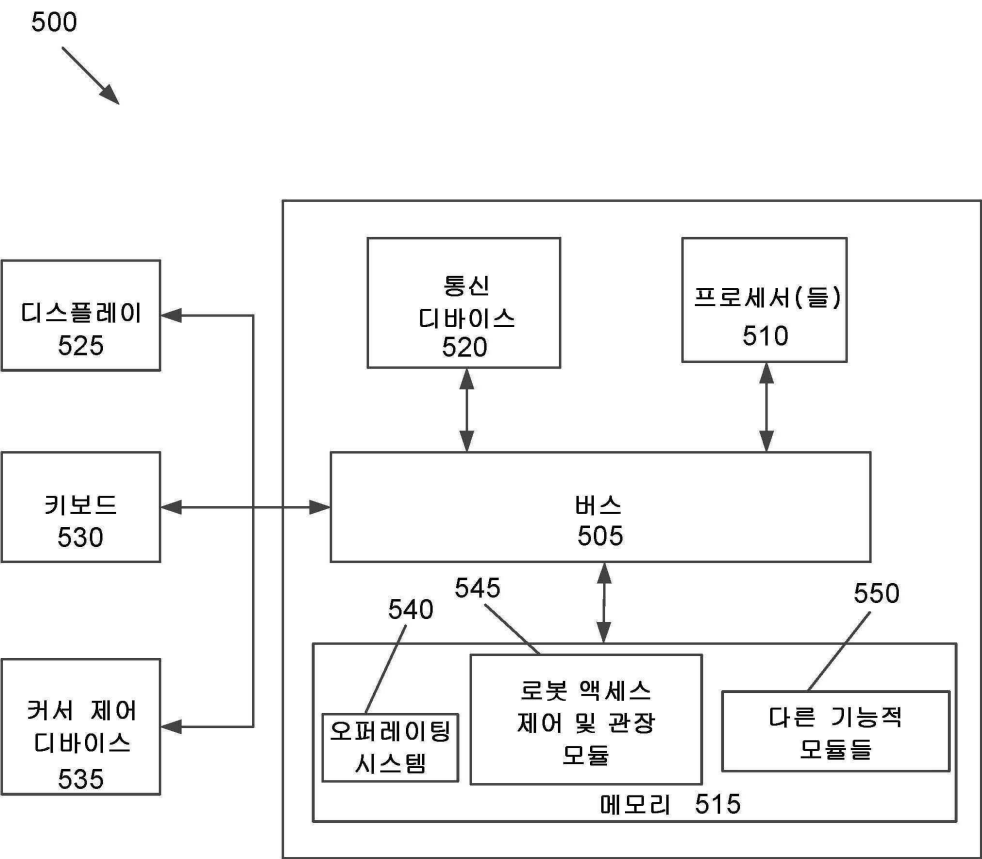
도면3



도면4



도면5



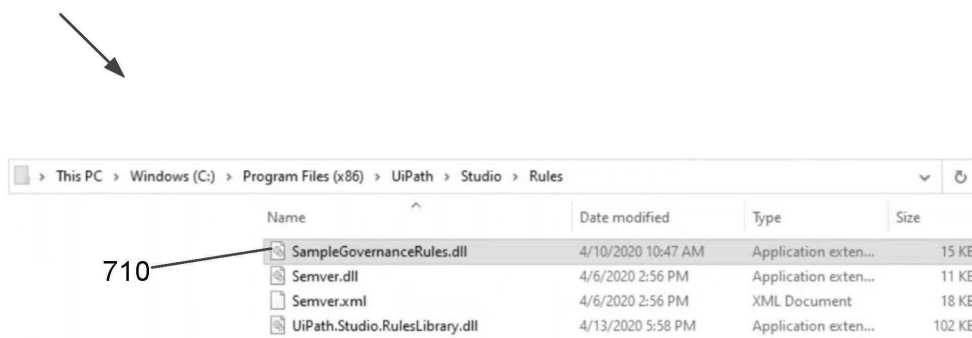
도면6

600



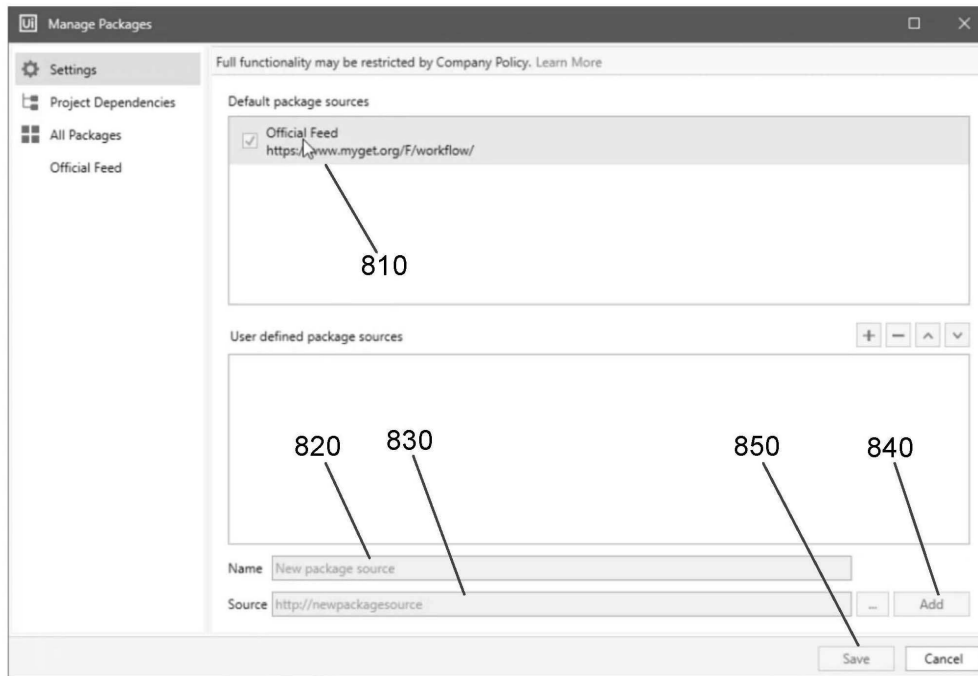
도면7

700



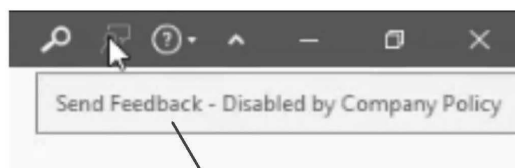
도면8

800



도면9

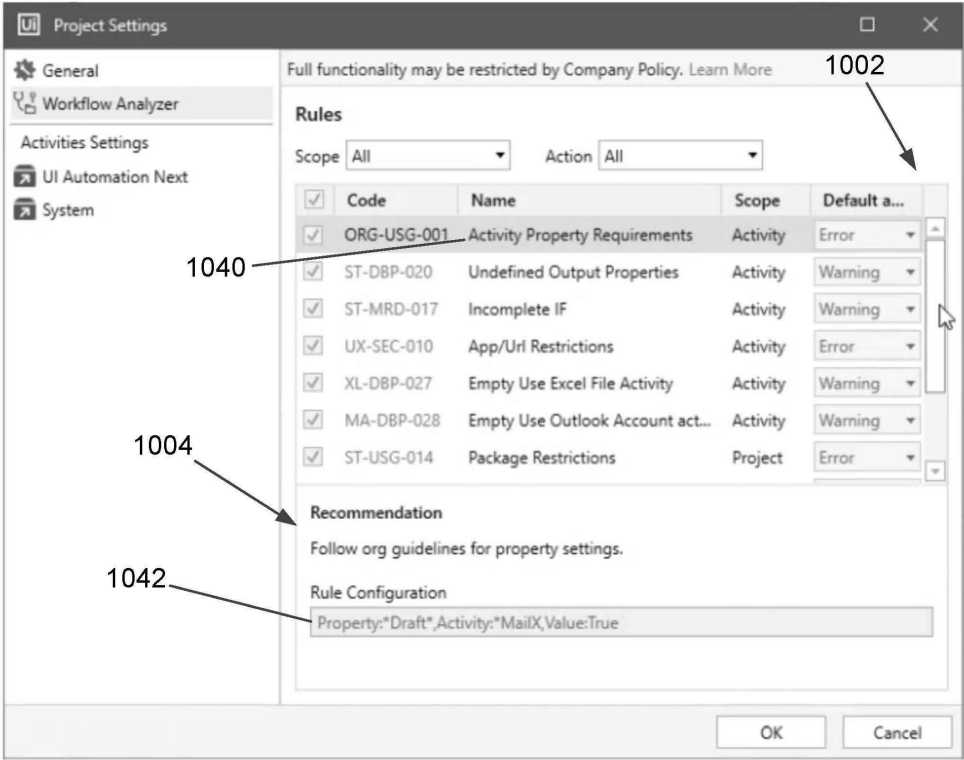
900



910

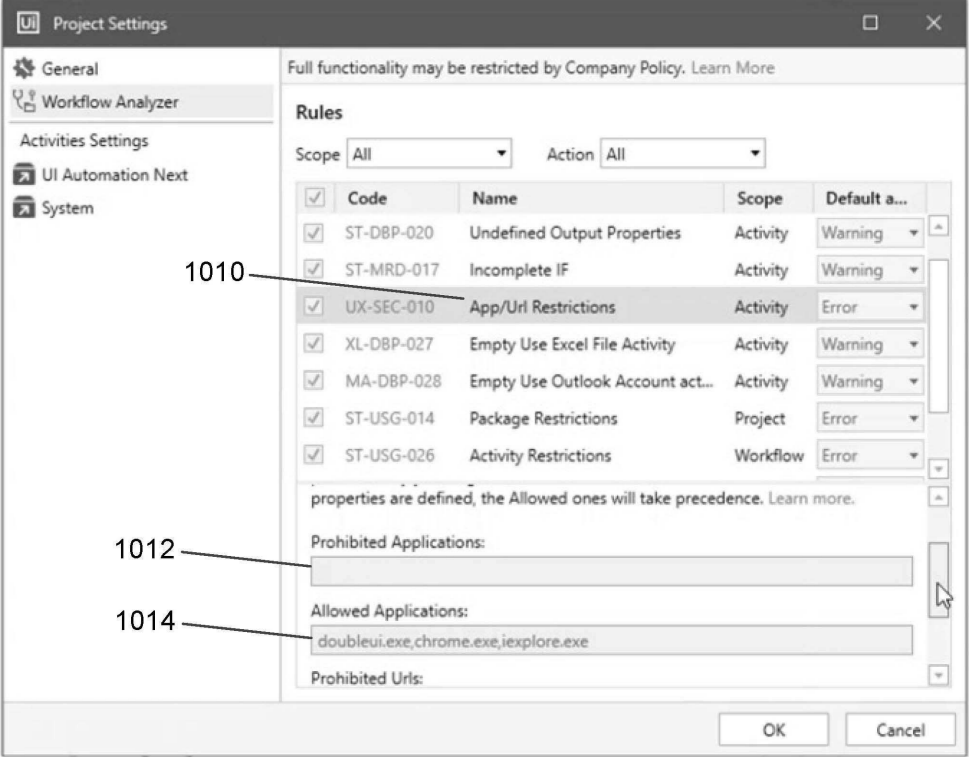
도면10a

1000



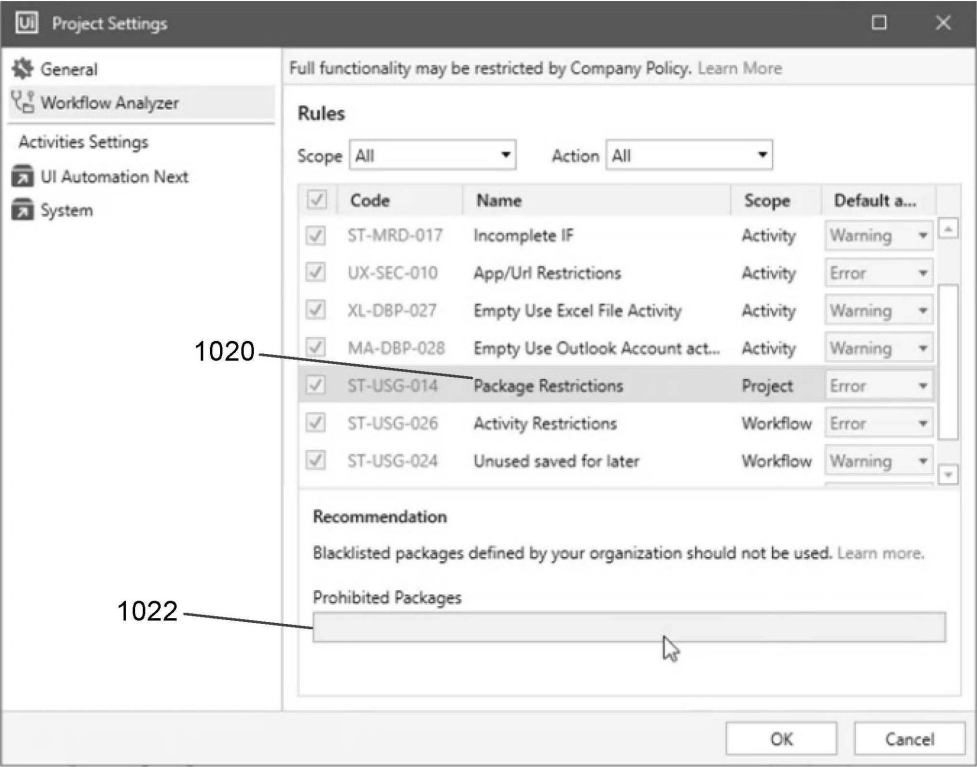
도면10b

1000



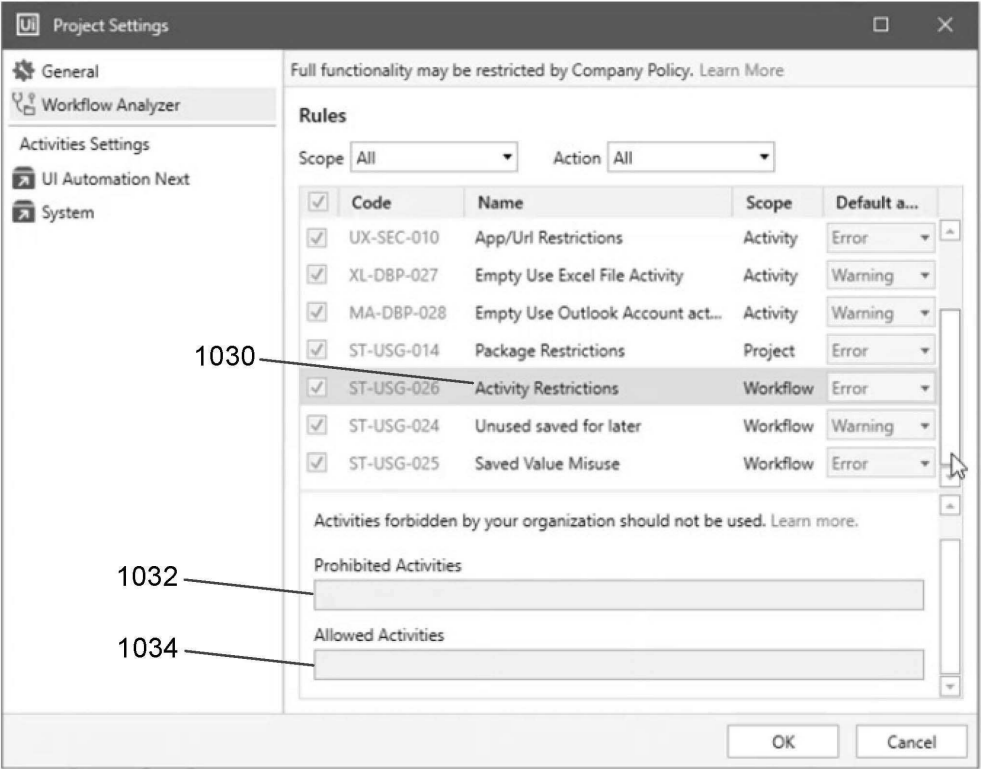
도면10c

1000



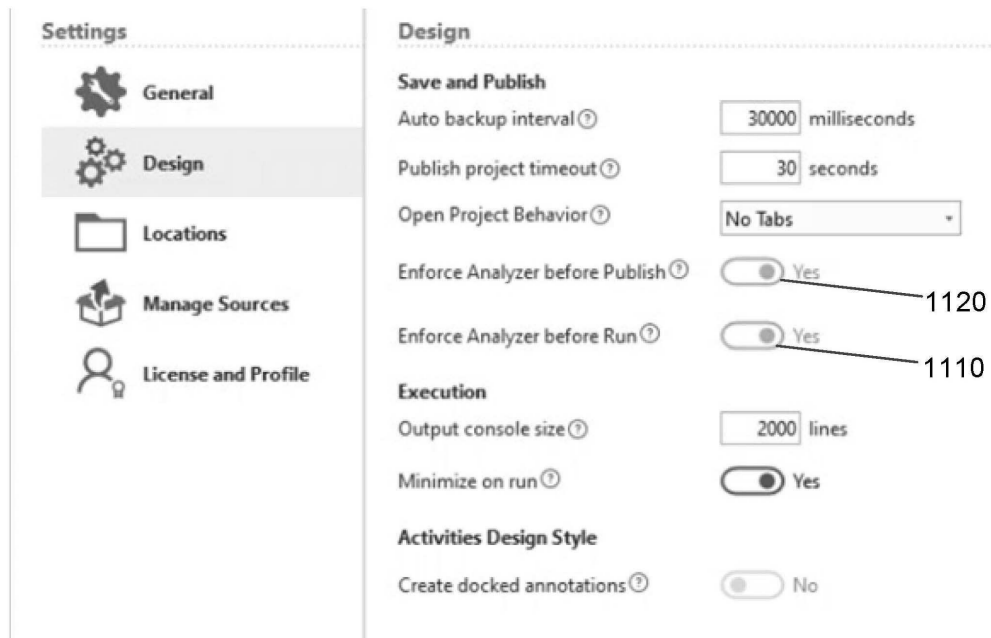
도면10d

1000



도면11

1100



도면 12a

1200

The screenshot displays a UiPath RPA interface with a sequence of steps:

- Step 1: Use Browser Chrome: Robotic Process Automation | UiPath**
 - Browser URL:
- Step 2: Use Outlook Account**
 - Account:
 - Reference as:
- Step 3: Send Outlook Email**
 - Account:
 - To:
 - Cc:
 - Subject:
 - Body: ☒ Text ☐ Use Word Document
 -
 - ☐ Save as draft

Arrows indicate the following labels:

- 1200** points to the top-left corner of the interface.
- 1210** points to the Browser URL input field.
- 1220** points to the "Save as draft" checkbox.

도면 12b

1200



Browser URL

https://www.uipath.com/

1210

Use Outlook Account

Account

Default Email Account

Reference as

Outlook

Send Outlook Email

Account

Outlook

To

me@mailinator.com

Cc

Select a value

Subject

Select a value

Body ☒ Text ☐ Use Word Document

Select a value

☐ Save as draft

Attachments ☒ Files ☐ Folder

Select a value

1220

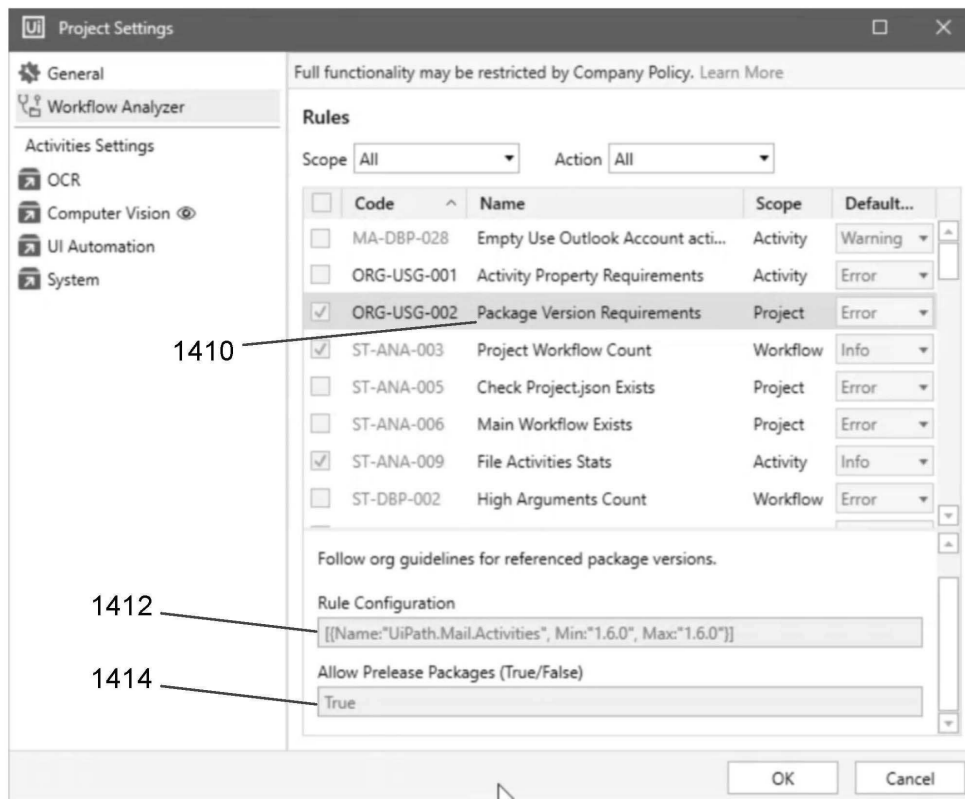
도면13

1300

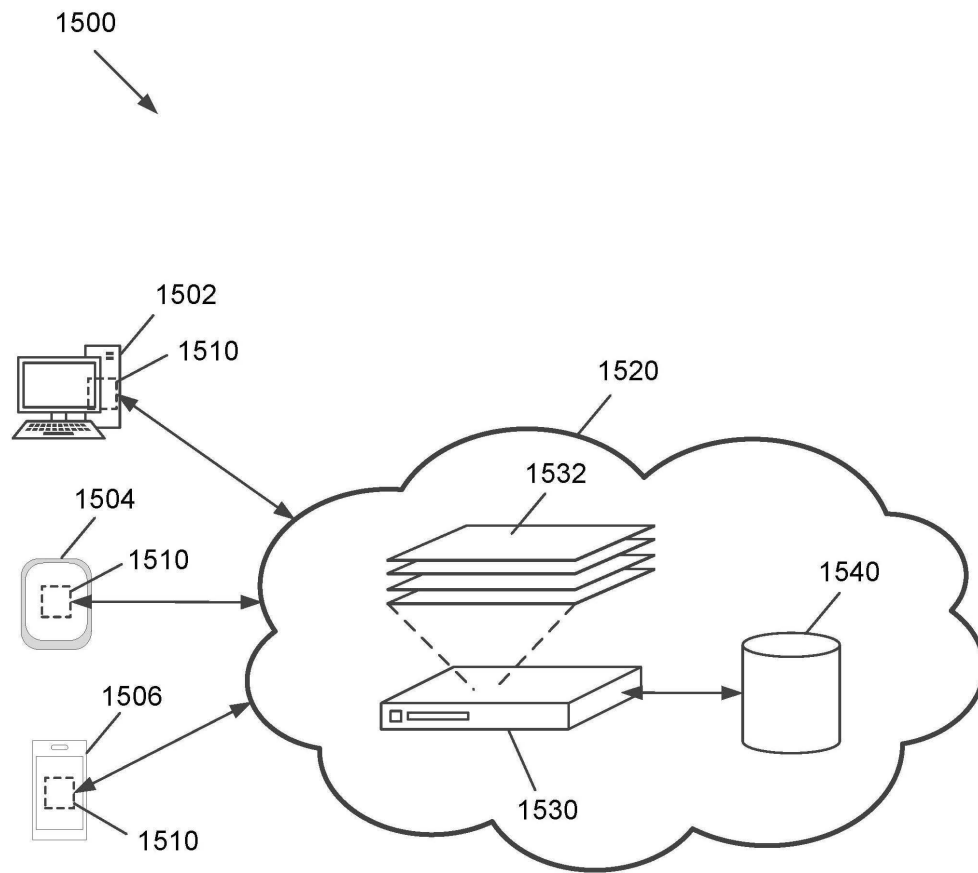
에러 리스트				
범위: 프로젝트▼		규칙 모든 규칙들▼		
v 코드	명칭	설명	파일	
▶ △ ORG-USG-001	활동 속성 요건들	활동의 '드래프트로서의 저장' 속성은 당신의 조직의 가이드라인들을 충족시키지 않음	Main.xml	
▶ △ UX-SEC-010	App/Url 제약들	UI 자동화는 당신의 조직에 의해 정의된 바와 같은 허용된 애플리케이션이 아닌 'https://www.uipath.com' 상에서 검출됨	Main.xml	

도면14

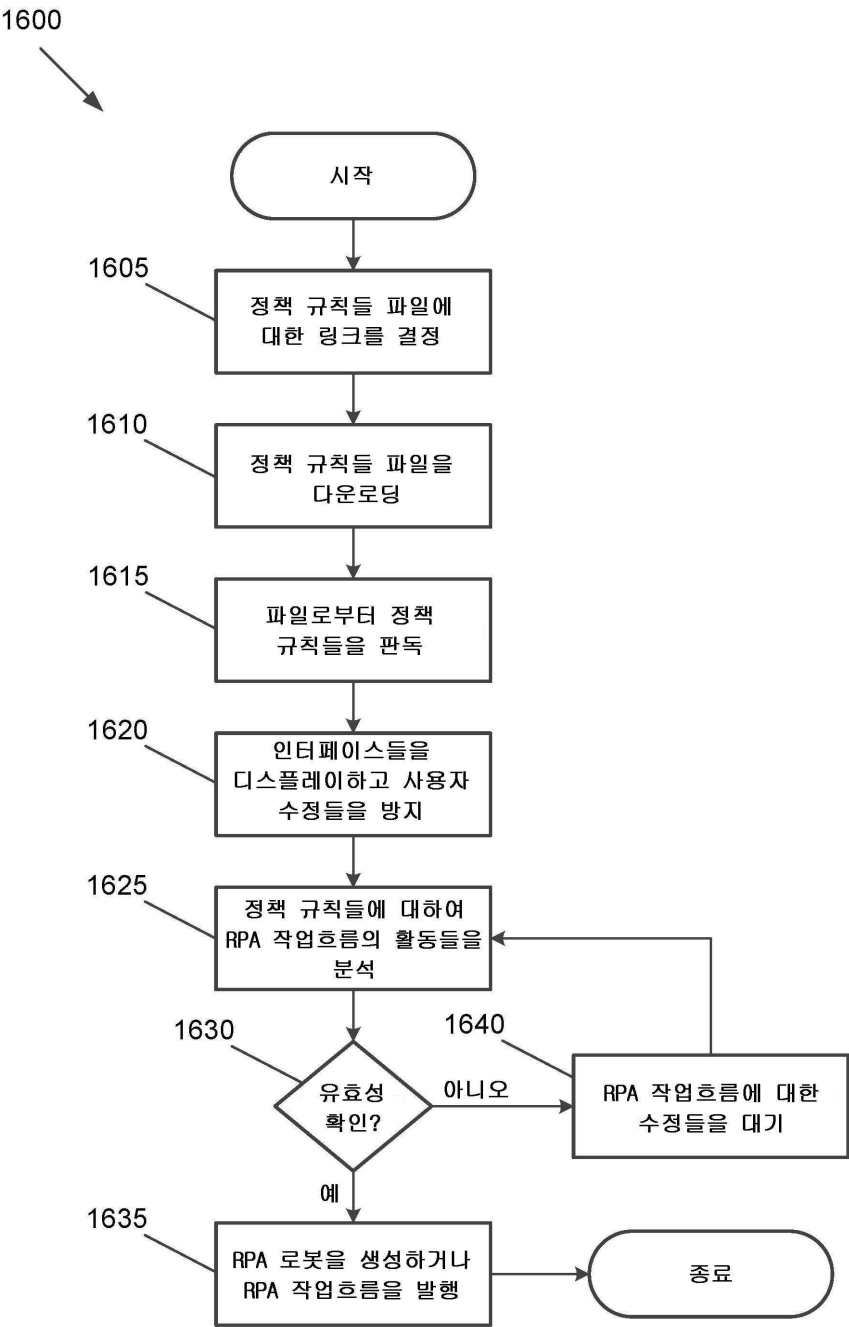
1400



도면15



도면16



도면17

