



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0007496
(43) 공개일자 2022년01월18일

- (51) 국제특허분류(Int. Cl.)
G06Q 10/10 (2022.01) G06F 3/048 (2021.01)
G06Q 10/06 (2012.01)
- (52) CPC특허분류
G06Q 10/10 (2021.08)
G06F 3/048 (2021.01)
- (21) 출원번호 10-2020-7028231
- (22) 출원일자(국제) 2020년09월22일
심사청구일자 2020년10월29일
- (85) 번역문제출일자 2020년09월29일
- (86) 국제출원번호 PCT/US2020/051940
- (87) 국제공개번호 WO 2022/010513
국제공개일자 2022년01월13일
- (30) 우선권주장
16/924,910 2020년07월09일 미국(US)
16/925,544 2020년07월10일 미국(US)
- (71) 출원인
유아이패스, 인크.
미국 뉴욕주 10016 뉴욕 20번 플로어 파크 애비뉴 90
- (72) 발명자
다인스 다니엘
미국 뉴욕주 10016 뉴욕 20번 플로어 파크 애비뉴 90
- (74) 대리인
김태홍, 김진희

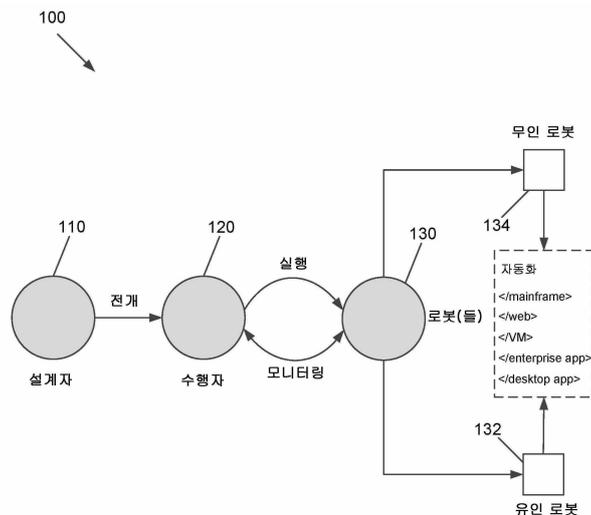
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 제1 세션에서 실행되고 있는 프로세스의 제2 세션에서 실행되고 있는 로봇 프로세스 자동화 로봇을 통한 자동화

(57) 요약

제1 세션에서 실행되고 있는 프로세스의 제2 세션에서 실행되고 있는 로봇 프로세스 자동화(RPA) 로봇(들)을 통한 자동화가 개시된다. 일부 양태들에서, 양식이 사용자 세션에서 디스플레이되지만, 제1 세션에서 애플리케이션을 위한 데이터를 추출하며 그리고/또는 그 데이터와 상호작용하는 하나 이상의 유인 RPA 로봇들이 하나 이상의 다른 세션들에서 실행한다. 이 방식으로, RPA 로봇(들)의 동작은 RPA 로봇(들)이 실행하고 있을 때 사용자가 다른 애플리케이션들 또는 인스턴스들을 사용하는 것을 방지하지 못할 수도 있지만, RPA 로봇(들)에 의해 이루어지거나 또는 용이하게 된 데이터 수정들은 제1 세션 윈도우에서 사용자에게 가시적일 수도 있다.

대표도 - 도1



(52) CPC특허분류

G06Q 10/06316 (2013.01)

G06Q 10/0633 (2013.01)

명세서

청구범위

청구항 1

컴퓨터에 의해 구현되는 방법에 있어서,

컴퓨팅 시스템의 메인 세션에서 로봇 프로세스 자동화(robotic process automation)(RPA) 로봇을 실행시키는 단계;

상기 메인 세션 RPA 로봇에 의해, 프로세스 간 통신(inter-process communication)(IPC)을 통해 클라이언트 세션에서 RPA 로봇을 호출 또는 론칭하는 단계;

상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션 RPA 로봇으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 워크플로우의 적어도 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해, 상기 클라이언트 세션 RPA 로봇으로부터의 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 2

제1항에 있어서, 상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션에서 애플리케이션을 호출 또는 론칭하는 단계;

상기 메인 세션 RPA 로봇에 의해, 상기 애플리케이션의 태스크의 실행 동안 및/또는 후에 IPC를 통해 클라이언트 세션 애플리케이션으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 상기 워크플로우의 적어도 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 상기 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해, 상기 클라이언트 세션 애플리케이션으로부터의 상기 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 3

제2항에 있어서, 상기 클라이언트 세션 애플리케이션이 실행을 완료하기까지 상기 메인 세션 RPA 로봇의 실행을 일시 중지시키는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 4

제2항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 메인 세션 RPA 로봇의 상기 워크플로우에서 두 번 이상 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 5

제2항에 있어서, 상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션에서 추가적인 애플리케이션을 호출 또는 론칭하는 단계;

IPC를 통해 추가적인 클라이언트 세션 애플리케이션으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 상기 워크플로우의 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 상기 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해, 상기 추가적인 클라이언트 세션 애플리케이션으로부터의 상기 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 6

제2항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 메인 세션 RPA 로봇의 상기 워크플로우의 호출 프로세스 활동을 통해 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 7

제6항에 있어서, 상기 호출 프로세스 활동은 상기 클라이언트 세션 애플리케이션이 상기 클라이언트 세션에서 실행된다는 것을 지시하는 플래그를 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 8

제2항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 태스크가 완료된 후에 계속 실행되고 있는, 컴퓨터에 의해 구현되는 방법.

청구항 9

제2항에 있어서, 상기 메인 세션 RPA 로봇과 상기 클라이언트 세션 애플리케이션 사이의 IPC 통신들은 표준화된 통신 포맷으로 되어 있는, 컴퓨터에 의해 구현되는 방법.

청구항 10

제1항에 있어서, 상기 클라이언트 세션 RPA 로봇이 실행을 완료하기까지 상기 메인 세션 RPA 로봇의 실행을 일시 중지시키는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 11

제1항에 있어서, 상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션에서 추가적인 RPA 로봇을 호출 또는 론칭하는 단계;

IPC를 통해 추가적인 클라이언트 세션 RPA 로봇으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 상기 워크플로우의 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 상기 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해, 상기 추가적인 클라이언트 세션 RPA 로봇으로부터의 상기 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 12

제1항에 있어서, 상기 클라이언트 세션 RPA 로봇은 상기 메인 세션 RPA 로봇의 상기 워크플로우에서 두 번 이상 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 13

제1항에 있어서, 상기 클라이언트 세션 RPA 로봇은 상기 메인 세션 RPA 로봇의 상기 워크플로우의 호출 워크플로우 활동을 통해 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 14

제13항에 있어서, 상기 호출 워크플로우 활동은 상기 클라이언트 세션 RPA 로봇이 상기 클라이언트 세션에서 실행된다는 것을 지시하는 플래그를 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 15

제1항에 있어서, 상기 클라이언트 세션 RPA 로봇은 상기 결과들이 상기 메인 세션 RPA 로봇에 제공된 후에 계속 실행되고 있는, 컴퓨터에 의해 구현되는 방법.

청구항 16

제1항에 있어서, 상기 메인 세션 RPA 로봇과 상기 클라이언트 세션 RPA 로봇 사이의 IPC 통신들은 표준화된 통신 포맷으로 되어 있는, 컴퓨터에 의해 구현되는 방법.

청구항 17

컴퓨터에 의해 구현되는 방법에 있어서,

컴퓨팅 시스템의 메인 세션에서 로봇 프로세스 자동화(robotic process automation)(RPA) 로봇을 실행시키는 단계;

상기 메인 세션 RPA 로봇에 의해, 프로세스 간 통신(inter-process communication)(IPC)을 통해 클라이언트 세션에서 애플리케이션을 호출 또는 론칭하는 단계;

상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션 애플리케이션으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 워크플로우의 적어도 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해, 상기 클라이언트 세션 애플리케이션으로부터의 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 18

제17항에 있어서, 상기 클라이언트 세션 애플리케이션이 실행을 완료하기까지 상기 메인 세션 RPA 로봇의 실행을 일시 중지시키는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 19

제17항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 메인 세션 RPA 로봇의 상기 워크플로우에서 두 번 이상 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 20

제17항에 있어서, 상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션에서 추가적인 애플리케이션을 호출 또는 론칭하는 단계;

IPC를 통해 추가적인 클라이언트 세션 애플리케이션으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 상기 워크플로우의 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 상기 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해, 상기 추가적인 클라이언트 세션 애플리케이션으로부터의 상기 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 21

제17항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 메인 세션 RPA 로봇의 상기 워크플로우의 호출 프로세스 활동을 통해 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 22

제21항에 있어서, 상기 호출 프로세스 활동은 상기 클라이언트 세션 애플리케이션이 상기 클라이언트 세션에서 실행된다는 것을 지시하는 플래그를 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 23

제17항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 태스크가 완료된 후에 계속 실행되고 있는, 컴퓨터에 의해 구현되는 방법.

청구항 24

컴퓨터에 의해 구현되는 방법에 있어서,

컴퓨팅 시스템의 메인 세션에서 로봇 프로세스 자동화(robotic process automation)(RPA) 로봇을 실행시키는 단계;

상기 메인 세션 RPA 로봇에 의해, 프로세스 간 통신(inter-process communication)(IPC)을 통해 클라이언트 세

선에서 RPA 로봇을 호출 또는 론칭하는 단계;

상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션에서 애플리케이션을 호출 또는 론칭하는 단계;

메인 세션 RPA 로봇에 의해, IPC를 통해 클라이언트 세션 RPA 로봇 및 클라이언트 세션 애플리케이션으로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 워크플로우의 적어도 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 하나 이상의 애플리케이션들 및/또는 애플리케이션 오브젝트들과 상호작용하기 위해, 또는 둘 다를 위해, 클라이언트 세션 RPA 로봇 및 클라이언트 세션 애플리케이션으로부터의 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 25

제24항에 있어서, 상기 클라이언트 세션 RPA 로봇, 상기 클라이언트 세션 애플리케이션, 또는 둘 다가, 실행을 완료하기까지 메인 세션 RPA 로봇의 실행을 일시 중지시키는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 26

제24항에 있어서, 클라이언트 세션 애플리케이션, 클라이언트 세션 애플리케이션, 또는 둘 다가, 상기 메인 세션 RPA 로봇의 상기 워크플로우에서 두 번 이상 호출 또는 론칭되는, 컴퓨터에 의해 구현되는 방법.

청구항 27

제24항에 있어서, 상기 메인 세션 RPA 로봇에 의해, IPC를 통해 상기 클라이언트 세션에서 추가적인 애플리케이션, 추가적인 RPA 로봇, 또는 둘 다를 호출 또는 론칭하는 단계;

IPC를 통해 상기 추가적인 클라이언트 세션 애플리케이션, 상기 추가적인 RPA 로봇, 또는 둘 다로부터 하나 이상의 메시지들을 수신하는 단계; 및

상기 메인 세션 RPA 로봇에 의해, 상기 메인 세션 RPA 로봇의 상기 워크플로우의 부분을 완료하기 위해, 상기 메인 세션에서 실행되고 있는 하나 이상의 애플리케이션들 및/또는 애플리케이션 오브젝트들 중 적어도 한 쪽과 상호작용하기 위해, 또는 둘 다를 위해, 상기 추가적인 클라이언트 세션 애플리케이션, 상기 추가적인 클라이언트 세션 RPA 로봇, 또는 둘 다로부터 상기 수신된 하나 이상의 메시지들로부터의 결과들을 사용하는 단계를 더 포함하는, 컴퓨터에 의해 구현되는 방법.

청구항 28

제24항에 있어서, 상기 클라이언트 세션 애플리케이션은 상기 메인 세션 RPA 로봇의 상기 워크플로우의 호출 프로세스 활동을 통해 호출 또는 론칭되거나, 상기 클라이언트 세션 RPA 로봇은 상기 메인 세션 RPA 로봇의 상기 워크플로우의 호출 워크플로우 활동을 통해 호출 또는 론칭되거나, 또는 둘 다가 되는, 컴퓨터에 의해 구현되는 방법.

청구항 29

제28항에 있어서, 상기 호출 프로세스 활동은 상기 클라이언트 세션 애플리케이션이 상기 클라이언트 세션에서 실행됨을 지시하는 플래그를 포함하거나, 상기 호출 워크플로우 활동은 클라이언트 세션 RPA 로봇이 클라이언트 세션에서 실행됨을 지시하는 플래그를 포함하거나, 또는 둘 다인, 컴퓨터에 의해 구현되는 방법.

청구항 30

제24항에 있어서, 상기 클라이언트 세션 애플리케이션, 상기 클라이언트 세션 RPA 로봇, 또는 둘 다는, 결과들을 제공한 후 계속 실행되고 있는, 컴퓨터에 의해 구현되는 방법.

발명의 설명

기술 분야

- [0001] 관련 출원들에 대한 교차 참조
- [0002] 본 출원은 2020년 7월 10일자로 출원된 미국 정식 특허출원 제16/925,544호를 우선권 주장하는데, 이 출원은 2020년 7월 9일자로 출원된 미국 정식 특허출원 제16/924,910호의 계속이며 그것의 우선권을 주장한다. 이들 이전에 출원된 출원들의 발명의 주제는 이로써 그 전부가 참조로 본원에 포함된다.
- [0003] 분야
- [0004] 본 발명은 대체로 로봇 프로세스 자동화(robotic process automation)(RPA)에 관한 것이고, 더 상세하게는, 제 1 세션에서 실행되고 있는 프로세스의 제2 세션에서 실행되고 있는 RPA 로봇(들)을 통한 자동화에 관한 것이다.

배경 기술

- [0005] 유인 자동화 RPA 로봇들이, 예를 들어, 해당 사용자와는 동일한 세션에서 사용자에게 의해 동작되는 컴퓨팅 시스템 상에서 통상적으로 실행된다. RPA 로봇들은, 예를 들어, 사용자의 커맨드에서 특정한 태스크들을 완수하기 위해 사용자와 함께 작업할 수도 있다. 그러나, 유인 자동화 시나리오들에서, RPA 로봇은 사용자의 컴퓨팅 시스템을 "장악(take over)"할 수도 있다. 사용자는 로봇이 컴퓨팅 시스템과 상호작용하고 있는 동안 다른 활동들을 수행하기 원할 수도 있지만, 사용자는 그렇게 하는 것이 방지된다. 다르게 말하면, 로봇은 사용자와 동일한 방식으로 사용자 인터페이스(user interface)(UI)를 통해 애플리케이션들을 제어한다(예컨대, 마우스 클릭들 및 키보드 입력을 시뮬레이션한다).
- [0006] 운영 체제(operating system) 또는 그것 상에서 실행되고 있는 애플리케이션들의 완전 또는 부분적 사본들을 생성하는 다양한 기술들이 존재한다. 에뮬레이터들은 수십 년 동안 존재했고 애플리케이션들을 테스트하고 디버깅하는 능력을 개발자들에게 제공할 수도 있다. 예를 들면, 에뮬레이터들은 실행중인 개발 도구들을 직접적으로 지원하지 않는 운영 체제를 사용하는 모바일 애플리케이션들을 테스트하고 디버깅하는 능력을 개발자들에게 제공할 수도 있다. Android[®] 및 iOS[®]는 Android[®] 또는 iOS[®] 애플리케이션을 테스트하고 디버깅하기 위해 개발 머신에서 실행될 수 있는 에뮬레이터들을 제공하는데, 개발자 도구들이 그들 모바일 운영 체제들 상에서 기본적으로 실행될 수 없기 때문이다.
- [0007] 시뮬레이터들은 개발 머신 상에서 수행하는 것이 어렵거나 또는 불가능한 애플리케이션의 거동을 개발자가 테스트하고 디버깅하게 하는 국부 머신 상에서 개발자가 윈도우를 호스팅하는 것을 허용한다. 예를 들어, 시뮬레이터들은 개발자가 시뮬레이터를 회전시키기 위해 버튼을 클릭하는 것을 허용하며, 이는 시뮬레이터 내부에서 실행되고 있는 애플리케이션에게 이들 이벤트들에 응답하는 애플리케이션 거동을 테스트하고 디버깅할 목적으로 디바이스가 회전되었음을 알린다. 다른 일반적인 예가 멀티 터치이다. 많은 개발자 머신들이 터치를 지원하지 않으며, 그래서 시뮬레이터는 애플리케이션이 다수의 터치 포인트들에 응답하는 방법을 개발자가 테스트하고 디버깅하게 한다. Android[®] 및 iOS[®] 에뮬레이터들은 시뮬레이션 능력들을 또한 제공한다. 더욱이, Microsoft[®]는 자신의 유니버설 윈도우즈 플랫폼(Universal Windows Platform)(UWP) 애플리케이션들을 위한 시뮬레이터를 제공한다.
- [0008] 가상 머신(virtual machine)들은 머신 상에서 제2 운영 체제를 호스팅하고 윈도우를 통해 열고 모니터링될 수 있다. 이는 완전히 상이한 운영 체제를 실행하고 호스트 머신과 하드웨어를 공유한다. "게스트" 머신은 설치된 애플리케이션들의 자신 소유의 사본들을 가져야만 하고 사용자 머신과는 공통 리소스들 또는 파일들을 공유하지 않는다.
- [0009] Docker[®] 컨테이너들은 개념적으로 가상 머신의 하이브리드 형태이다. 실행될 것이 필요한 애플리케이션들의 모두가 호스 운영 체제 상에서 직접적으로 실행되는 변경 불가능한 패키지로 패키징된다. 그 패키지는 다른 운영 체제의 완전한 사본이 아니지만, 호스트 머신 상의 애플리케이션들 또는 리소스들 중 임의의 것을 디폴트로 공유하거나 또는 그것에 액세스하지 않는다. 따라서, 사용자 경험 관점에서, Docker[®] 컨테이너들은 가상 머신과 유사하게 느껴지지만, 기술적으로, 컨테이너들은 완전히 별도의 운영 체제 상에서 실행되지 않는다.
- [0010] 그러나, 기존의 에뮬레이터들, 시뮬레이터들, 가상 머신들(VM들), 및 운영 체제(OS) 레벨 가상화를 제공하는 하이브리드 VM들(예컨대, Docker[®] 컨테이너들)은 사용자와 동일한 컴퓨팅 시스템 상에서 동작하는 유인 자동화 로봇들에서 일어나는 문제들을 해결하지 않는다. 따라서, 사용자는 본질적으로는 자신 소유의 컴퓨팅 시스템에 대한 관찰자가 되어, 로봇이 작업하는 것을 보고 사용자 상호작용을 요구하는 머신 상의 다른 애플리케이션들과

상호작용할 수 없게 된다. 그에 따라, 개선된 접근법이 유익할 수도 있다.

발명의 내용

- [0011] 본 발명의 특정한 실시예들은 현재 RPA 기술들에 의해 아직 완전히 식별, 인식, 또는 해결되지 않은 업계에서의 문제들 및 요구들에 대한 해결책들을 제공할 수도 있다. 예를 들어, 본 발명의 일부 실시예들은 제1 세션에서 실행되고 있는 프로세스의 제2 세션에서 실행되고 있는 RPA 로봇(들)을 통한 자동화에 관한 것이다.
- [0012] 일 실시예에서, 컴퓨터 프로그램이 비일시적 컴퓨터 판독가능 매체 상에 수록된다. 컴퓨터 프로그램은 적어도 하나의 프로세서로 하여금, 제1 세션에서 프로세스 간 통신(inter-process communication)(IPC) 촉진자 및 RPA 드라이버를 실행하게 하고 제2 세션에서 RPA 로봇을 실행하게 하도록 구성된다. 컴퓨터 프로그램은 또한 적어도 하나의 프로세서로 하여금, IPC 촉진자에 의해, IPC를 통해 RPA 로봇으로부터 하나 이상의 메시지들을 수신하게 하고, RPA 로봇으로부터의 수신된 하나 이상의 메시지들에 기초하여, IPC 촉진자에 의해, 제1 세션에서 실행되고 있는 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하도록 RPA 드라이버를 제어하게 하도록 구성된다.
- [0013] 다른 실시예에서, 컴퓨터에 의해 구현되는 방법은, 제2 세션에서 실행되고 있는 RPA 로봇에 의해, IPC를 통해 제1 세션에서 실행되고 있는 IPC 촉진자에 하나 이상의 메시지들을 전송하는 단계와, IPC 촉진자에 의해, IPC를 통해 RPA 로봇으로부터 하나 이상의 메시지들을 수신하는 단계를 포함한다. 컴퓨터에 의해 구현되는 방법은 또한 RPA 로봇으로부터의 수신된 하나 이상의 메시지들에 기초하여, IPC 촉진자에 의해, 제1 세션에서 또한 실행되고 있는 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하도록 제1 세션에서 실행되고 있는 RPA 드라이버를 제어하는 단계를 포함한다.
- [0014] 또 다른 실시예에서, 컴퓨터에 의해 구현되는 방법은 제1 세션에서 실행되고 있는 IPC 촉진자에 의해, IPC를 통해 제2 세션에서 실행되고 있는 RPA 로봇으로부터 하나 이상의 메시지들을 수신하는 단계를 포함한다. 컴퓨터에 의해 구현되는 방법은 또한 RPA 로봇으로부터의 수신된 하나 이상의 메시지들에 기초하여, IPC 촉진자에 의해, 제1 세션에서 또한 실행되고 있는 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하도록 제1 세션에서 실행되고 있는 RPA 드라이버를 제어하는 단계를 포함한다.

도면의 간단한 설명

- [0015] 본 발명의 특정한 실시예들의 장점들이 쉽사리 이해되도록 하기 위하여, 위에서 간략히 설명된 본 발명의 더 특정한 설명이 첨부된 도면들에서 예시되는 특정 실시예들을 참조하여 제공될 것이다. 이들 도면들은 본 발명의 전형적인 실시예들만을 묘사하고 그러므로 본 발명의 범위의 제한으로 간주되지 않는 것으로 이해되어야 하지만, 본 발명은 첨부 도면들의 사용을 통해 추가의 특이성 및 세부사항과 함께 기재되고 설명될 것이며, 도면들 중:
 - 도 1은 본 발명의 일 실시예에 따른 로봇 프로세스 자동화(RPA) 시스템을 예시하는 아키텍처 도면이다.
 - 도 2는 본 발명의 일 실시예에 따른 전개된(deployed) RPA 시스템(200)을 예시하는 아키텍처 도면이다.
 - 도 3은 본 발명의 일 실시예에 따른 설계자, 활동들, 및 드라이버들 사이의 관계를 예시하는 아키텍처 도면이다.
 - 도 4는 본 발명의 일 실시예에 따른 RPA 시스템을 예시하는 아키텍처 도면이다.
 - 도 5는 본 발명의 일 실시예에 따른 RPA 로봇들을 위한 세션 간 자동화를 용이하게 하도록 구성되는 컴퓨팅 시스템을 예시하는 아키텍처 도면이다.
 - 도 6a 내지 도 6g는 본 발명의 일 실시예에 따른, 로봇 세션에서 실행되고 있는 RPA 로봇, IPC 촉진자, 및 RPA 드라이버를 사용하여 사용자 세션에서 양식을 완성하는 예를 예시한다.
 - 도 7a 내지 도 7g는 본 발명의 일 실시예에 따른, 직접 변수 수정을 통해 로봇 세션에서 실행되고 있는 RPA 로봇을 사용하여 사용자 세션에서 양식을 완성하는 예를 도시한다.
 - 도 8은 본 발명의 일 실시예에 따른, 로봇 세션에서 실행되고 있는 RPA 로봇, IPC 촉진자, 및 RPA 드라이버를 통해 사용자 세션에서 실행되고 있는 프로세스의 자동화를 위한 프로세스를 예시하는 흐름도이다.
 - 도 9는 본 발명의 일 실시예에 따른, 직접 변수 수정을 통해 사용자 세션에서 실행되고 있는 프로세스의 로봇

세션에서 실행되고 있는 RPA 로봇을 통한 자동화를 위한 프로세스를 예시하는 흐름도이다.

도 10은 본 발명의 일 실시예에 따른, 메인 세션 애플리케이션과 상호작용하는 다수의 클라이언트 세션 RPA 로봇들의 일 예를 도시한다.

도 11은 본 발명의 일 실시예에 따른, 클라이언트 세션에서 유인 자동화 RPA 로봇을 실행하기 위한 프로세스를 예시하는 흐름도이다.

도 12는 본 발명의 일 실시예에 따른, 메인 세션 로봇(M1), 클라이언트 세션 애플리케이션(A1), 및 한 쌍의 클라이언트 세션 로봇들(C1 및 C2) 사이의 다중 로봇 집단 워크플로우의 실행을 예시하는 흐름도이다.

도 13은 본 발명의 일 실시예에 따른, 메인 세션 RPA 로봇 및 클라이언트 세션 RPA 로봇의 실행을 병렬로 예시하는 흐름도이다.

도 14는 본 발명의 일 실시예에 따른, 메인 세션 애플리케이션에 대한 트리거에 기초한 클라이언트 세션 RPA 로봇의 실행을 예시하는 흐름도이다.

도 15는 본 발명의 일 실시예에 따른, 세션 간 자동화를 수행하기 위한 프로세스를 예시하는 흐름도이다.

달리 지시되지 않는 한, 유사한 참조 부호들은 결부된 도면들의 전체에 걸쳐 일관되게 대응하는 특징들을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0016] 일부 실시예들은 제1 세션에서 실행되고 있는 프로세스의 제2 세션에서 실행되고 있는 RPA 로봇(들)을 통한 자동화에 관한 것이다. 예를 들면, 양식(예컨대, 웹 페이지, 이메일 애플리케이션(예컨대, Outlook[®]), 스프레드시트 애플리케이션(예컨대, Excel[®]), CRM(customer relationship management) 시스템 애플리케이션(예컨대, Salesforce[®]), ERM(enterprise resource management) 시스템, 공급 체인 관리 시스템, 맞춤형 컴퓨터 애플리케이션, 모바일 애플리케이션, 채움가능 피드들을 갖는 다른 애플리케이션 등)이 제1 세션에서 디스플레이될 수도 있다. 일부 실시예들에서, 제1 세션은 사용자 세션(또한 본 개시에서 메인 세션 또는 부모 세션이라 불림)일 수도 있다. 제1 세션에서 애플리케이션을 위한 데이터를 추출하며 그리고/또는 그 데이터와 상호작용하는 하나 이상의 RPA 로봇들은 하나 이상의 다른 세션들(본 개시에서는 또한 로봇 세션들, 제2 세션들, 또는 이차 세션들이라 불림)에서 실행될 수도 있다. 특정 실시예들에서, 로봇 세션은 제1 세션의 자식 세션일 수도 있다.

[0017] 일부 실시예들에서, 클라이언트 세션에서 실행되고 있는 RPA 로봇은 프로세스 간 통신(IPC)을 통해 메인 세션에서 실행되고 있는 RPA 드라이버에 메시지들을 전송한다. RPA 로봇으로부터의 IPC 통신물(들)에 기초하여, RPA 드라이버는 컴퓨팅 시스템과의 다양한 운영 체제(OS) 및/또는 애플리케이션 프로그래밍 인터페이스(application programming interface)(API) 레벨 상호작용들을 구현할 수도 있다. 예를 들면, 드라이버는 본 발명의 범위로부터 벗어나는 일 없이 마우스를 이동시키거나, 버튼을 클릭하거나, 메뉴와 상호작용하거나, 필드에 텍스트를 입력하거나, 윈도우를 열거나 또는 닫거나, 윈도우를 이동시키며 그리고/또는 크기 조정하거나, 또는 임의의 다른 적합한 상호작용을 수행할 수도 있다.

[0018] 이 기능을 구현하기 위하여, 드라이버는 IPC 촉진자를 포함할 수도 있거나 또는 그 IPC 촉진자에 의해 제어 또는 호출될 수도 있다. IPC 촉진자는 일부 실시예들에서 별도의 애플리케이션 또는 프로세스일 수도 있거나 또는 드라이버의 하위 프로세스일 수도 있다. IPC 촉진자는 클라이언트 세션에서 RPA 로봇으로부터 IPC 메시지들을 청취(예컨대, 트리거를 청취)하며, RPA 로봇에 통신물들을 전송하고 RPA 로봇으로부터 통신물들을 수신하며, RPA 로봇 실행 스테이터스를 모니터링하며, 그것들의 조합 등을 수행할 수도 있다. IPC 촉진자는 또한 드라이버로 하여금 RPA 로봇으로부터의 IPC 통신물(들)에 기초하여 부모 세션에서 원하는 기능을 구현하도록 할 수도 있다. 특정 실시예들에서, IPC 촉진자는 주어진 RPA 로봇을 위한 스크립트로 구성될 수도 있다. 일부 실시예들에서, IPC 촉진자는 부모 세션에서 수행될 기능을 위해 맞춤 설계될 수도 있다.

[0019] 일부 실시예들에서, 클라이언트 세션에서의 RPA 로봇은 컴퓨팅 시스템 상의 메모리에 저장되는 변수들을 직접 수정할 수도 있다. 이들 변수들에 액세스하는 부모 세션에서 실행되고 있는 애플리케이션은 그 다음에 디스플레이가 리프레시할 때에 디스플레이된 값들을 업데이트할 수 있거나 또는 그 애플리케이션은 그렇지 않으면 자신의 UI를 리로드(reload)한다. 예를 들어, Excel[®] 스프레드시트의 변수들은 클라이언트 세션에서 실행되고 있는 RPA 로봇에 의해 직접 수정될 수 있는데, 이는 부모 세션에서 디스플레이된 값들 역시 변경시킬 것이다.

- [0020] 일부 실시예들에서, RPA 로봇(들)의 동작은 RPA 로봇이 실행되고 있을 때 사용자가 다른 애플리케이션들 또는 인스턴스들을 사용하는 방지하지 못할 수도 있지만, RPA 로봇(들)에 의해 이루어진 데이터 수정들은 애플리케이션 디스플레이가 사용자 세션 윈도우에서 리프레시됨에 따라 사용자에게 가시적이 될 수도 있다. 그러나, 특정 실시예들에서, RPA 로봇(들)은 사용자 인터페이스를 갖지 않는 애플리케이션들과 상호작용할 수도 있다.
- [0021] 일부 실시예들에서, 프로세스는 다음과 같이 동작할 수도 있다. 사용자가 버튼을 클릭하거나 또는 아니면 RPA 로봇으로 하여금 사용자 세션(예컨대, 사용자가 자신의 컴퓨팅 시스템을 시동할 때 론칭되는 부모 또는 메인 세션)의 자식 세션에서 론칭하게 할 수도 있다. 특정 실시예들에서, 로봇 세션은 이미 실행되고 있을 수도 있거나, 또는, 예를 들어, 로봇 트레이로부터 사용자에게 의해 별도로 론칭될 수도 있다. 사용자는 로봇으로 하여금 자신의 워크플로우를 자식 세션에서 실행하게 할 수도 있고, 그러면 로봇은 부모 세션에서 실행되고 있는 IPC 촉진자와 통신함으로써 메인 세션에서 하나 이상의 애플리케이션들과 상호작용할 수도 있다. 그 다음에 IPC 촉진자는 RPA 드라이버로 하여금 부모 세션에서 원하는 기능을 수행하게 할 수도 있다.
- [0022] 일부 실시예들에서, 로봇 세션에서 실행되고 있는 RPA 로봇은 워크플로우가 관련되는 적어도 하나의 애플리케이션이 론칭되지 않았으면 동작을 일시 정지시키고 메시지를 제공할 수도 있고, IPC 촉진자는 애플리케이션을 론칭할 수도 있으며 그래서 로봇은 동작을 계속할 수 있다. 일단 로봇이 자신의 워크플로우를 실행하는 것을 시작하며, 애플리케이션에서 디스플레이된 데이터에 대한 변경들은 이러한 변경들이 IPC 촉진자 및 RPA 드라이버를 통해 이루어질 때 메인 세션 윈도우에서 가시적이 될 수도 있다. 예를 들면, 클라이언트 세션에서 RPA 로봇에 의해 제공되는 데이터에 기초하여 드라이버가 데이터를 채움(예컨대, IPC를 통해, 플랫폼 파일(flat file) 또는 데이터베이스 등에 데이터를 저장함)에 따라 데이터는 나타날 수도 있다.
- [0023] 비제한적인 예로서, 일부 실시예들에서, RPA 로봇으로 하여금 클라이언트 세션에서 론칭되게 하거나, 또는 클라이언트 세션에서 이미 실행중인 RPA 로봇과 상호작용하게 하는 스크립트가 Salesforce®에서 생성될 수도 있다. 사용자가 스크립트를 클릭할 때, RPA 로봇은 자신의 RPA 워크플로우에서 설계된 바와 같은 RPA 프로세스를 실행하는 것을 시작한다. 자신의 실행(예컨대, 데이터베이스로부터 정보를 추출하는 것, 인공지능(artificial intelligence)(AI)/머신 러닝(machine learning)(ML) 모델을 호출하고 결과들을 획득하는 것, 다수의 레거시 시스템들로부터 정보를 획득하는 것 등)의 결과들에 기초하여, RPA 로봇은 그러면 IPC 촉진자와 상호작용하는데, 그러면 IPC 촉진자는 그에 따라 부모 세션에서 실행중인 Salesforce® 애플리케이션 인스턴스와 상호작용한다. 이는 실행중인 Salesforce® 애플리케이션 인스턴스에서 디스플레이된 정보의 일부로 하여금, 예를 들어 RPA 드라이버가 상호작용들을 수행함에 따라, 변경되게 하고 사용자에게 가시적이 되게 할 수도 있다.
- [0024] 특정 실시예들에서, 클라이언트 세션에서 실행되고 있는 RPA 로봇은 부모 세션에서의 애플리케이션이 IPC 촉진자를 통해 열리게 할 수도 있다. 그러면 IPC 촉진자는, RPA 드라이버를 통해, 애플리케이션에 연관된 데이터에 액세스하고 추가, 변경, 및/또는 삭제를 할 수도 있다. 이런 방식으로, RPA 로봇이 부모 세션에서 실행되고 있지 않더라도, RPA 로봇은 부모 세션 자체에서 실행되고 있는 RPA 로봇과 유사하게 나타나지만 기능적으로는 상이한 방식으로, 부모 세션 애플리케이션들에서 상호작용들을 여전히 유발할 수 있다.
- [0025] 일부 실시예들에서, 사용자가 웹사이트로 이동하고 얼마간의 정보를 수집하는 클라이언트 세션에서 로봇으로 하여금 워크플로우를 실행하게 할 수도 있다. 그러면 RPA 로봇은 이 정보를 IPC 촉진자에게 제공할 수도 있으며, 그러면 IPC 촉진자는 RPA 드라이버로 하여금, 예를 들어 메인 세션에서 가시적이 되는 스프레드시트에, 수집된 정보를 입력하게 한다. 특정 실시예들에서, Salesforce®와 같은 애플리케이션이, 메인 세션에서 열린다. 그러면 사용자는, 예를 들어, 현재 클라이언트 ID를 읽고, 웹사이트(예컨대, HubSpot®)로 이동하고, 웹사이트와의 클라이언트 상호작용들에 관계된 정보를 수집하는 자동화를 클라이언트 세션에서 실행한다. 그러나, 유인 RPA 로봇들은 본 발명의 범위로부터 벗어나는 일 없이 RPA 촉진자를 통해 임의의 적합한 애플리케이션(들)과 상호작용할 수도 있으며 그리고/또는 임의의 적합한 소스(예컨대, 데이터베이스, 다른 애플리케이션 등)으로부터 데이터를 획득할 수도 있다.
- [0026] 일부 실시예들의 애플리케이션들은 에뮬레이터들, 시뮬레이터들, VM들, OS 레벨 가상화를 제공하는 하이브리드 VM들 (예컨대, Docker® 컨테이너들)을 비제한적으로 포함한다. 일부 실시예들은 유인 자동화 프로세스에 의해 제어되는 애플리케이션들의 UI들을 포함하는 윈도우로서 하나 이상의 로봇 세션들을 생성하고 호스팅한다. 특정 실시예들에서, 로봇(들)과 상호작용하는 애플리케이션들의 인터페이스들만이 보이고 있다. 일부 실시예들에

서, 클라이언트 세션 윈도우가 전혀 보이지 않는다. 본 개시에서 사용되는 바와 같이, "윈도우"가 메인 UI 내에 도시된 UI를 나타내는 윈도우, 컴퓨팅 시스템의 제2 디스플레이의 제2 스크린, 가상 데스크톱, 격리된 환경(즉, 격리된 환경 내부에서 론칭되는 모든 애플리케이션들("자식"이라고 지칭됨)의 UI들을 그리고 호스팅 세션의 맥락에서 그것들을 실행하는 윈도우("호스트"라고 지칭됨) 등에, 본 발명의 범위로부터 벗어나는 일 없이 적용될 수도 있다. 실행중인 다수의 세션들은 사용자가 제1 세션(예컨대, 부모 세션)과 상호작용하는 동안 로봇(들)이 자신들의 세션(들)에서 동작하는 것을 허용한다. 따라서 사용자는 로봇이 데이터와 상호작용하는 동안 다양한 애플리케이션들과 상호작용할 수 있다.

[0027] 일부 실시예들에서, 임의의 수의 로봇들을 위한 임의의 원하는 수의 세션들이 본 발명의 범위로부터 벗어나는 일 없이 생성되고 사용될 수 있다. 예를 들면, 사용자는 제1 세션에서 동작할 수도 있으며, 제1 로봇이 제2 세션에서 동작할 수도 있으며, 제2 로봇이 제3 세션에서 동작할 수도 있다는 등이다. 특정 실시예들에서, 다수의 로봇들은 RPA 촉진자를 통해 하나 이상의 공통 애플리케이션들과 상호작용하기 위해 잠재적으로 번갈아 가며 단일 세션에서 동작할 수도 있다.

[0028] 세션을 생성하기 위한 기능은, 예를 들어, 사용자가 로그 아웃하지 않고서도 사용자 소유의 머신으로 되돌아가는 세션을 생성할 수 있는 Windows[®] 단말 서비스들 자식 세션들을 통해 구현될 수도 있다. 새로 생성된 세션은 자식 윈도우로서 나타나고 사용자의 세션에 존재하는 애플리케이션들을 포함하고 론칭한다. 다르게 말하면, 사용자와 로봇 사이의 분리는 UI 레벨에서 일어난다. 예를 들어, 파일이 삭제되면, 이는 컴퓨팅 시스템 상에서 실행되고 있는 모든 세션들에 걸쳐 발생한다.

[0029] 특정한 실시예들은 로봇 프로세스 자동화(RPA)를 위해 채용될 수도 있다. 도 1은 본 발명의 일 실시예에 따른 RPA 시스템(100)을 예시하는 아키텍처 도면이다. RPA 시스템(100)은 개발자가 워크플로우들을 설계하고 구현하는 것을 허용하는 설계자(110)를 포함한다. 설계자(110)는 애플리케이션 통합, 뿐만 아니라 서드파티 애플리케이션들, 관리 정보 기술(IT) 태스크들, 및 비즈니스 IT 프로세스들의 자동화를 위한 솔루션을 제공할 수도 있다. 설계자(110)는 비즈니스 프로세스의 그래픽 표현인 자동화 프로젝트의 개발을 용이하게 할 수도 있다. 간단히 말해서, 설계자(110)는 워크플로우들 및 로봇들의 개발 및 전개를 용이하게 한다.

[0030] 자동화 프로젝트는 본 개시에서 "활동들(activities)"로서 정의되는 워크플로우로 개발된 사용자지정 세트의 단계들 사이의 실행 순서 및 관계의 체어를 개발자에게 제공함으로써 규칙 기반 프로세스들의 자동화를 가능하게 한다. 설계자(110)의 일 실시예의 하나의 상업적 예가 UiPath Studio[™]이다. 각각의 활동은 버튼 클릭, 파일 관독, 로그 패널에의 기입 등과 같은 액션을 포함할 수도 있다. 일부 실시예들에서, 워크플로우들은 네스트되거나 또는 삽입될 수도 있다.

[0031] 일부 유형들의 워크플로우들은 시퀀스들, 흐름도들, FSM들, 및/또는 전역 예외 핸들러들을 비제한적으로 포함할 수도 있다. 시퀀스들은 워크플로우를 혼잡하게 하는 일 없이 하나의 활동에서부터 다른 활동으로의 흐름을 가능하게 하는 선형 프로세스들에 특히 적합할 수도 있다. 흐름도들은 더 복잡한 비즈니스 로직에 특히 적합하여, 다수의 분기 로직 오퍼레이터들을 통해 더 다양한 방식으로 활동들의 연결과 결정들의 통합을 가능하게 할 수도 있다. FSM들은 큰 워크플로우들에 특히 적합할 수도 있다. FSM들은 조건(즉, 전이(transition)) 또는 활동에 의해 트리거되는 유한한 수의 상태들을 자신들의 실행에 사용할 수도 있다. 전역 예외 핸들러들은 실행 예러와 마주칠 때에 워크플로우 거동을 결정하는데 그리고 프로세스들을 디버깅하는데 특히 적합할 수도 있다.

[0032] 일단 워크플로우가 설계자(110)에서 개발되면, 비즈니스 프로세스들의 실행은 수행자(conductor)(120)에 의해 오케스트레이션되는데, 수행자는 설계자(110)에서 개발된 워크플로우들을 실행하는 하나 이상의 로봇들(130)을 오케스트레이션한다. 수행자(120)의 일 실시예의 하나의 상업적 예는 UiPath Orchestrator[™]이다. 수행자(120)는 환경에서 리소스들의 생성, 모니터링 및 전개의 관리를 용이하게 한다. 수행자(120)는 서드파티 솔루션들 및 애플리케이션들을 이용하여, 통합 지점, 또는 집성 지점들 중 하나의 집성 지점으로서 역할을 할 수도 있다.

[0033] 수행자(120)는 로봇들(130)의 플릿(fleet)을 관리하여, 중앙 지점으로부터의 로봇들(130)을 연결하고 실행할 수도 있다. 관리될 수도 있는 로봇들(130)의 유형들은 유인 로봇들(132), 무인 로봇들(134), 개발 로봇들(무인 로봇들(134)과 유사하지만, 개발 및 테스트 목적으로 사용됨), 및 비생산 로봇들(유인 로봇들(132)과 유사하지만, 개발 및 테스트 목적으로 사용됨)을 비제한적으로 포함한다. 유인 로봇들(132)은 사용자 이벤트들에 의해 트리거되거나 또는 자동으로 일어나도록 스케줄링될 수도 있고 동일한 컴퓨팅 시스템 상의 사람과 함께 동작할 수도 있다. 유인 로봇들(132)은 중앙집중형 프로세스 전개 및 로깅 매체를 위해 수행자(120)와 함께 사용될 수

도 있다. 유인 로봇들(132)은 인간 사용자가 다양한 태스크들을 수행하도록 도울 수도 있고, 사용자 이벤트들에 의해 트리거될 수도 있다. 일부 실시예들에서, 프로세스들은 이 유형의 로봇에 대한 수행자(120)로부터 시작될 수 없으며 그리고/또는 그들 프로세스들은 잠긴 스크린 하에서 실행될 수 없다. 특정 실시예들에서, 유인 로봇들(132)은 오직 로봇 트레이 또는 커맨드 프롬프트로부터만 시작될 수 있다. 유인 로봇들(132)은 일부 실시예들에서 인간의 감독 하에 실행되어야 한다.

[0034] 무인 로봇들(134)은 가상 환경들에서 무인으로 실행되고 많은 프로세스들을 자동화할 수 있다. 무인 로봇들(134)은 원격 실행, 모니터링, 스케줄링, 및 작업 큐들에 대한 지원의 제공을 담당할 수도 있다. 모든 로봇 유형들에 대한 디버깅은 일부 실시예들에서 설계자(110)에서부터 실행될 수 있다. 유인 로봇 및 무인 로봇 둘 다 는 메인프레임들, 웹 애플리케이션들, VM들, 엔터프라이즈 애플리케이션들(예컨대, SAP[®], Salesforce[®], Oracle[®] 등에 의해 생성된 것들), 및 컴퓨팅 시스템 애플리케이션들(예컨대, 데스크톱 및 랩톱 애플리케이션들, 모바일 디바이스 애플리케이션들, 착용가능 컴퓨터 애플리케이션들 등)을 비제한적으로 포함하는 다양한 시스템들 및 애플리케이션들을 자동화할 수도 있다.

[0035] 수행자(120)는 프로비저닝(provisioning), 전개, 버저닝(versioning), 구성, 큐잉, 모니터링, 로깅, 및/또는 상호연결성의 제공을 비제한적으로 포함하는 다양한 능력들을 가질 수도 있다. 프로비저닝은 로봇들(130)과 수행자(120)(예컨대, 웹 애플리케이션) 사이의 연결들의 생성 및 유지보수를 포함할 수도 있다. 전개는 패키지 버전들의 배정된(assigned) 로봇들(130)로의 올바른 전달을 실행을 위해 보장하는 것을 포함할 수도 있다. 버저닝은 일부 실시예들에서 일부 프로세스 또는 구성의 고유 인스턴스들의 관리를 포함할 수도 있다. 구성은 로봇 환경들 및 프로세스 구성들의 유지보수 및 전달을 포함할 수도 있다. 큐잉은 큐들 및 큐 아이템들의 관리를 제공하는 것을 포함할 수도 있다. 모니터링은 로봇 식별 데이터의 추적을 유지하는 것과 사용자 허가들을 유지하는 것을 포함할 수도 있다. 로깅은 데이터베이스(예컨대, SQL 데이터베이스) 및/또는 다른 저장 메커니즘(예컨대, 큰 데이터세트들을 저장하고 빠르게 쿼리 할 수 있는 기능을 제공하는 ElasticSearch[®])에 대한 로그들의 저장 및 인덱싱을 포함할 수도 있다. 수행자(120)는 서드파티 솔루션들 및/또는 애플리케이션들에 대한 중앙 통신 지점으로서 역할을 함으로써 상호연결성을 제공할 수도 있다.

[0036] 로봇들(130)은 설계자(110)에 내장된 워크플로우들을 실행하는 실행 에이전트들이다. 로봇(들)(130)의 일부 실시예들의 하나의 상업적 예는 UiPath Robots[™]이다. 일부 실시예들에서, 로봇들(130)은 Microsoft Windows[®] 서비스 제어 관리자(Service Control Manager)(SCM) 관리 서비스를 디폴트로 설치한다. 그 결과, 이러한 로봇들(130)은 국부 시스템 계층 하에 상호작용 Windows[®] 세션들을 열 수 있고, Windows[®] 서비스의 권한을 가진다.

[0037] 일부 실시예들에서, 로봇들(130)은 사용자 모드로 설치될 수 있다. 이러한 로봇들(130)의 경우, 이는 그들 로봇들이 주어진 로봇(130)이 설치한 사용자와 동일한 권한을 가짐을 의미한다. 이 특징은 또한 각각의 머신의 완전한 이용을 자신의 최대 잠재력에서 보장하는 고밀도(High Density)(HD) 로봇들에 이용 가능할 수도 있다. 일부 실시예들에서, 임의의 유형의 로봇(130)은 HD 환경에서 구성될 수도 있다.

[0038] 일부 실시예들에서의 로봇들(130)은 각각의 컴포넌트가 특정 자동화 태스크에 전용되는 여러 컴포넌트들로 분할된다. 일부 실시예들에서의 로봇 컴포넌트들은 SCM 관리 로봇 서비스들, 사용자 모드 로봇 서비스들, 실행자들, 에이전트들, 및 커맨드 라인을 비제한적으로 포함한다. SCM 관리 로봇 서비스들은 Windows[®] 세션들을 관리하고 모니터링하며 수행자(120)와 실행 호스트들(즉, 로봇들(130)이 실행되는 컴퓨팅 시스템들) 사이의 프록시로서 역할을 한다. 이들 서비스들은 로봇들(130)에 대한 자격증명을 신뢰하고 관리한다. 콘솔 애플리케이션이 국부 시스템 하에서 SCM에 의해 시작된다.

[0039] 일부 실시예들에서의 사용자 모드 로봇 서비스들은 Windows[®] 세션들을 모니터링하고 관리하며 수행자(120)와 실행 호스트들 사이의 프록시로서 역할을 한다. 사용자 모드 로봇 서비스들은 로봇들(130)에 대한 자격증명을 신뢰하고 관리한다. SCM 관리 로봇 서비스가 설치되지 않으면 Windows[®] 애플리케이션이 자동으로 시작될 수도 있다.

[0040] 실행자들은 Windows[®] 세션 하에서 주어진 작업들을 실행할 수도 있다(즉, 그것들은 워크플로우들을 실행할 수도 있다). 실행자들은 모니터별 DPI(dots per inch) 설정들을 알고 있을 수도 있다. 에이전트들은 시스템 트레이 윈도우에 가용 작업들을 디스플레이하는 Windows[®] 프레젠테이션 파운데이션(WPF) 애플리케이션들일 수도 있다.

에이전트들은 서비스의 클라이언트일 수도 있다. 에이전트들은 작업들을 시작하거나 또는 중지하는 것과 설정들을 변경하는 것을 요청할 수도 있다. 커맨드 라인인 서비스의 클라이언트이다. 커맨드 라인은 작업들을 시작할 것을 요청할 수 있고 그 작업들의 출력을 기다리는 콘솔 애플리케이션이다.

[0041] 위에서 설명된 바와 같이 로봇들(130)의 컴포넌트들을 분리하면 개발자들, 지원 사용자들, 및 컴퓨팅 시스템들은 각각의 컴포넌트가 실행되고 있는 것을 더 쉽게 실행하며, 식별하고, 추적할 수 있게 한다. 실행자 및 서비스에 대해 상이한 방화벽 규칙들을 설정하는 것과 같이, 특수한 거동들은 이러한 방식으로 컴포넌트별로 구성될 수도 있다. 실행자는 일부 실시예들에서 모니터링별 DPI 설정들을 항상 알고 있을 수도 있다. 그 결과, 워크플로우들은 그들 워크플로우들이 생성되었던 컴퓨팅 시스템의 구성에 상관없이 임의의 DPI에서 실행될 수 있다. 설계자(110)로부터의 프로젝트들은 또한 일부 실시예들에서 브라우저 줌 레벨과는 독립적일 수도 있다. DPI 비인식 또는 의도적으로 비인식으로 마킹된 애플리케이션들의 경우, DPI는 일부 실시예들에서 디스에이블될 수도 있다.

[0042] 도 2는 본 발명의 일 실시예에 따른 전개된 RPA 시스템(200)을 예시하는 아키텍처 도면이다. 일부 실시예들에서, RPA 시스템(200)은 도 1의 RPA 시스템(100)일 수도 있거나, 또는 그 RPA 시스템의 일부일 수도 있다. 클라이언트 측, 서버 측, 또는 양쪽 모두는 발명의 범위로부터 벗어나는 일 없이 임의의 원하는 수의 컴퓨팅 시스템들을 포함할 수도 있다는 것에 주의해야 한다. 클라이언트 측에서, 로봇 애플리케이션(210)이 실행자들(212), 에이전트(214), 및 설계자(216)를 포함한다. 그러나, 일부 실시예들에서, 설계자(216)는 컴퓨팅 시스템(210) 상에서 실행되지 않을 수도 있다. 실행자들(212)은 프로세스들을 실행하고 있다. 여러 비즈니스 프로젝트들은, 도 2에서 도시된 바와 같이, 동시에 실행될 수도 있다. 에이전트(214)(예컨대, Windows® 서비스)가 이 실시예에서의 모든 실행자들(212)에 대한 단일 접촉 지점이다. 이 실시예에서의 모든 메시지들은 수행자(230)에 로깅되며, 수행자는 그들 메시지들을 데이터베이스 서버(240), 인덱서 서버(250), 또는 둘 다를 통해 추가로 프로세싱한다. 도 1을 참조하여 위에서 논의된 바와 같이, 실행자들(212)은 로봇 컴포넌트들일 수도 있다.

[0043] 일부 실시예들에서, 로봇이 머신 이름과 사용자 이름 사이의 연관을 표현한다. 그 로봇은 동시에 다수의 실행자들을 관리할 수도 있다. 동시에 실행되고 있는 다수의 상호작용 세션들을 지원하는 컴퓨팅 시스템들(예컨대, Windows® 서버 2012) 상에서, 다수의 로봇들은 고유 사용자이름을 사용하는 별도의 Windows® 세션에서 각각을 동일한 시간에 실행되고 있을 수도 있다. 이는 위에서 논의된 HD 로봇들이라고 지칭된다.

[0044] 에이전트(214)는 (예컨대, 로봇이 여전히 기능하고 있음을 지시하는 "심박" 메시지를 주기적으로 전송하는) 로봇의 상태를 전송하고 실행될 패키지의 요구된 버전을 다운로드하는 역할을 또한 담당한다. 에이전트(214)와 수행자(230) 사이의 통신이 일부 실시예들에서 에이전트(214)에 의해 항상 개시된다. 통지 시나리오에서, 에이전트(214)는 수행자(230)에 의해 나중에 사용되는 WebSocket 채널을 로봇에 커맨드들(예컨대, 시작, 중지 등)을 전송하기 위해 열 수도 있다.

[0045] 서버 측에는, 프레젠테이션 계층(웹 애플리케이션(232), 오픈 데이터 프로토콜(Open Data Protocol)(OData) REST(Representative State Transfer) 애플리케이션 프로그래밍 인터페이스(Application Programming Interface)(API) 엔드포인트들(234), 그리고 통지 및 모니터링(236)), 서비스 계층(API 구현예/비즈니스 로직(238)), 및 영속 계층(데이터베이스 서버(240) 및 인덱서 서버(250))가 포함된다. 수행자(230)는 웹 애플리케이션(232), OData REST API 엔드포인트들(234), 통지 및 모니터링(236), 그리고 API 구현예/비즈니스 로직(238)을 포함한다. 일부 실시예들에서, 사용자가 수행자(230)의 인터페이스에서 (예컨대, 브라우저(220)를 통해) 수행하는 대부분의 액션들은 다양한 API들을 호출함으로써 수행된다. 이러한 액션들은 발명의 범위를 벗어나는 일 없이 로봇들 상의 작업들을 시작하는 것, 큐들에서 데이터를 추가/제거하는 것, 무인 실행을 위한 작업들을 스케줄링하는 것 등을 비제한적으로 포함할 수도 있다. 웹 애플리케이션(232)은 서버 플랫폼의 시각적 계층이다. 이 실시예에서, 웹 애플리케이션(232)은 하이퍼 텍스트 마크업 언어(Hypertext Markup Language)(HTML)와 자바스크립트(JavaScript)(JS)를 사용한다. 그러나, 임의의 원하는 마크업 언어들, 스크립트 언어들, 또는 임의의 다른 포맷들이 발명의 범위로부터 벗어나는 일 없이 사용될 수도 있다. 사용자는 다양한 액션들을 수행하여 수행자(230)를 제어하기 위하여 이 실시예에서 브라우저(220)를 통해 웹 애플리케이션(232)으로부터의 웹 페이지들과 상호작용한다. 예를 들면, 사용자는 로봇 그룹들을 생성하며, 패키지들을 로봇들에 배정하며, 로봇별 및/또는 프로세스별 로그들을 분석하며, 로봇들을 시작하고 정지하는 등을 할 수도 있다.

[0046] 웹 애플리케이션(232) 외에도, 수행자(230)는 OData REST API 엔드포인트들(234)을 노출시키는 서비스 계층을

또한 포함한다. 그러나, 다른 엔드포인트들은 발명의 범위로부터 벗어나는 일 없이 포함될 수도 있다. REST API는 웹 애플리케이션(232) 및 에이전트(214) 둘 다에 의해 소비된다. 에이전트(214)는 이 실시예에서 클라이언트 컴퓨터 상의 하나 이상의 로봇들의 감독자이다.

[0047] 이 실시예에서의 REST API는 구성, 로깅, 모니터링, 및 큐잉 기능을 커버한다. 구성 엔드포인트들은 일부 실시예들에서 애플리케이션 사용자들, 허가들, 로봇들, 자산들, 릴리스들, 및 환경들을 정의하고 구성하는데 사용될 수도 있다. 로깅 REST 엔드포인트들은, 예를 들면, 에러들, 로봇들에 의해 전송된 명시적 메시지들, 및 다른 환경 특정 정보와 같은 상이한 정보를 로깅하는데 사용될 수도 있다. 전개 REST 엔드포인트들은 시작 작업 커맨드가 수행자(230)에서 사용되면 실행되어야 하는 패키지 버전을 쿼리하기 위해 로봇들에 의해 사용될 수도 있다. 큐잉 REST 엔드포인트들은 큐들과 데이터를 큐에 추가하는 것, 큐로부터 트랜잭션을 획득하는 것, 트랜잭션의 스테이더스를 설정하는 것 등과 같은 큐 아이템 관리를 담당할 수도 있다.

[0048] 모니터링 REST 엔드포인트들은 웹 애플리케이션(232) 및 에이전트(214)를 모니터링할 수도 있다. 통지 및 모니터링 API(236)는 에이전트(214)를 등록하는데, 구성 설정들을 에이전트(214)에 전달하는데, 그리고 서버 및 에이전트(214)로부터의 통지들을 전송/수신하는데 사용되는 REST 엔드포인트들일 수도 있다. 통지 및 모니터링 API(236)는 일부 실시예들에서 WebSocket 통신을 또한 사용할 수도 있다.

[0049] 영속 계층(persistence layer)은 이 실시예에서 한 쌍의 서버들 - 데이터베이스 서버(240)(예컨대, SQL 서버) 및 인덱서 서버(250) - 을 포함한다. 이 실시예에서의 데이터베이스 서버(240)는 로봇들, 로봇 그룹들, 연관된 프로세스들, 사용자들, 규칙들, 스케줄들 등의 구성을 저장한다. 이 정보는 일부 실시예들에서 웹 애플리케이션(232)을 통해 관리된다. 데이터베이스 서버(240)는 큐들 및 큐 아이템들을 관리할 수도 있다. 일부 실시예들에서, 데이터베이스 서버(240)는 (인덱서 서버(250)에 더하여 또는 그 인덱서 서버 대신에) 로봇들에 의해 로깅된 메시지들을 저장할 수도 있다.

[0050] 인덱서 서버(250)는, 일부 실시예들에서 옵션적인 것으로, 로봇들에 의해 로깅된 정보를 저장하고 인덱싱한다. 특정 실시예들에서, 인덱서 서버(250)는 구성 설정들을 통해 디스에이블될 수도 있다. 일부 실시예들에서, 인덱서 서버(250)는 ElasticSearch[®]를 사용하는데, 이는 오픈 소스 프로젝트 풀 텍스트 검색 엔진이다. 로봇들(예컨대, 로그 메시지 또는 기입 라인과 같은 활동들을 사용함)에 의해 로깅된 메시지들은 로깅 REST 엔드포인트(들)를 통해 인덱서 서버(250)에 전송될 수도 있으며, 인덱서 서버에서 그 메시지들은 추가의 이용을 위해 인덱싱된다.

[0051] 도 3은 본 발명의 일 실시예에 따른 설계자(310), 활동들(320, 330), 및 드라이버들(340) 사이의 관계(300)를 예시하는 아키텍처 도면이다. 위와 같이, 개발자가 설계자(310)를 사용하여 로봇들에 의해 실행되는 워크플로우들을 개발한다. 워크플로우들은 사용자 정의 활동들(320)과 UI 자동화 활동들(330)을 포함할 수도 있다. 일부 실시예들은 본 개시에서 컴퓨터 비전(computer vision)(CV)이라 불리는, 이미지에서의 비 텍스트 시각적 컴포넌트들을 식별할 수 있다. 이러한 컴포넌트들에 관련된 일부 CV 활동들은 클릭, 타이핑(type), 텍스트 가져오기, 호버(hover), 엘리먼트 존재, 범위 리프레시, 강조표시 등을 비제한적으로 포함할 수도 있다. 일부 실시예들에서의 클릭은, 예를 들어, CV, 광학적 문자 인식(OCR), 퍼지 텍스트 매칭, 및 멀티 앵커를 사용하여 엘리먼트를 식별하고, 그 엘리먼트를 클릭한다. 타이핑은 위의 것을 사용하여 엘리먼트를 식별하고 그 엘리먼트에 입력할 수도 있다. 텍스트 가져오기는 특정 텍스트의 로케이션을 식별하고 그 특정 텍스트를 OCR을 사용하여 스캔할 수도 있다. 호버는 엘리먼트를 식별하고 그것 위에 호빙할 수도 있다. 엘리먼트 존재는 위에서 설명된 기법들을 사용하여 엘리먼트가 스크린 상에 존재하는지의 여부를 체크할 수도 있다. 일부 실시예들에서, 설계자(310)에서 구현될 수 있는 수백 또는 심지어 수천 개의 활동들이 있을 수도 있다. 그러나, 임의의 수 및/또는 유형의 활동들은 발명의 범위로부터 벗어나는 일 없이 이용 가능할 수 있다.

[0052] UI 자동화 활동들(330)은 하위 레벨 코드(예컨대, CV 활동들)로 작성되고 스크린과의 상호작용들을 용이하게 하는 특수한, 하위 레벨 활동들의 서브세트이다. UI 자동화 활동들(330)은 로봇이 원하는 소프트웨어와 상호작용하는 것을 허용하는 드라이버들(340)을 통해 이들 상호작용들을 용이하게 한다. 예를 들면, 드라이버들(340)은 OS 드라이버들(342), 브라우저 드라이버들(344), VM 드라이버들(346), 엔터프라이즈 애플리케이션 드라이버들(348) 등을 포함할 수도 있다.

[0053] 드라이버들(340)은 후크들을 찾고 키들을 모니터링하는 등 낮은 레벨에서 OS와 상호작용할 수도 있다. 그것들은 Chrome[®], IE[®], Citrix[®], SAP[®] 등과와의 통합을 용이하게 할 수도 있다. 예를 들면, "클릭" 활동은 드라이버들(340)을 통해 이들 상이한 애플리케이션들에서 동일한 역할을 수행한다.

- [0054] 도 4는 본 발명의 일 실시예에 따른 RPA 시스템(400)을 예시하는 아키텍처 도면이다. 일부 실시예들에서, RPA 시스템(400)은 도 1 및/또는 도 2의 RPA 시스템들(100 및/또는 200)일 수도 있거나 또는 그것들을 포함할 수도 있다. RPA 시스템(400)은 로봇들을 실행하는 다수의 클라이언트 컴퓨팅 시스템들(410)을 포함한다. 컴퓨팅 시스템들(410)은 자신들 상에서 실행되고 있는 웹 애플리케이션을 통해 수행자 컴퓨팅 시스템(420)과 통신할 수 있다. 수행자 컴퓨팅 시스템(420)은, 결국, 데이터베이스 서버(430) 및 옵션적인 인덱서 서버(440)와 통신할 수 있다.
- [0055] 도 1 및 도 3에 관해, 웹 애플리케이션이 이들 실시예들에서 사용되지만, 임의의 적합한 클라이언트/서버 소프트웨어는 발명의 범위로부터 벗어나는 일 없이 사용될 수 있다는 점에 주의해야 한다. 예를 들면, 수행자는 클라이언트 컴퓨팅 시스템들 상의 비 웹 기반 클라이언트 소프트웨어 애플리케이션들과 통신하는 서버측 애플리케이션을 실행할 수도 있다.
- [0056] 도 5는 본 발명의 일 실시예에 따른 RPA 로봇들을 위한 세션 간 자동화를 용이하게 하도록 구성되는 컴퓨팅 시스템(500)을 예시하는 아키텍처 도면이다. 일부 실시예들에서, 컴퓨팅 시스템(500)은 본 개시에서 묘사되며 및/또는 설명되는 컴퓨팅 시스템들 중 하나 이상일 수도 있다. 컴퓨팅 시스템(500)은 정보를 통신하기 위한 버스(505) 또는 다른 통신 메커니즘과, 버스(505)에 커플링되어 정보를 프로세싱하는 프로세서(들)(510)를 포함한다. 프로세서(들)(510)는 중앙 프로세싱 유닛(Central Processing Unit)(CPU), 주문형 집적회로(Application Specific Integrated Circuit)(ASIC), 필드 프로그램가능 게이트 어레이(Field Programmable Gate Array)(FPGA), 그래픽 프로세싱 유닛(Graphics Processing Unit)(GPU), 그것들의 다중 인스턴스들, 및/또는 그것들의 임의의 조합을 포함하는 임의의 유형의 일반 또는 특정 목적 프로세서일 수도 있다. 프로세서(들)(510)는 다수의 프로세싱 코어들 또한 가질 수도 있고, 코어들 중 적어도 일부는 특정 기능들을 수행하도록 구성될 수도 있다. 다중 병렬 프로세싱이 일부 실시예들에서 사용될 수도 있다. 특정 실시예들에서, 적어도 하나의 프로세서(들)(510)는 생물학적 뉴런들을 흉내내는 프로세싱 엘리먼트들을 포함하는 신경모방(neuromorphic) 회로일 수도 있다. 일부 실시예들에서, 신경모방 회로들은 폰 노이만 컴퓨팅 아키텍처의 전형적인 컴포넌트들을 요구하지 않을 수도 있다.
- [0057] 컴퓨팅 시스템(500)은 프로세서(들)(510)에 의해 실행될 정보 및 명령들을 저장하기 위한 메모리(515)를 더 포함한다. 메모리(515)는 임의의 조합의 랜덤 액세스 메모리(Random Access Memory)(RAM), 판독 전용 메모리(Read Only Memory)(ROM), 플래시 메모리, 캐시, 자기 또는 광학적 디스크와 같은 정적 스토리지, 또는 임의의 다른 유형들의 비일시적 컴퓨터 판독가능 매체 또는 그 조합들로 구성될 수 있다. 비일시적 컴퓨터 판독가능 매체는 프로세서(들)(510)에 의해 액세스될 수 있고 휘발성 미디어, 비휘발성 미디어, 또는 둘 다를 포함할 수도 있는 임의의 가용 매체일 수도 있다. 그 매체들은 또한 탈착식, 비탈착식, 또는 둘 다일 수도 있다.
- [0058] 덧붙여, 컴퓨팅 시스템(500)은 무선 및/또는 유선 연결을 통해 통신 네트워크에 대한 액세스를 제공하는 통신 디바이스(520), 이를테면 트랜시버를 포함한다. 일부 실시예들에서, 통신 디바이스(520)는 본 발명의 범위로부터 벗어나는 일 없이 주파수 분할 다중 접속(Frequency Division Multiple Access)(FDMA), SC-FDMA(Single Carrier FDMA), 시분할 다중 접속(Time Division Multiple Access)(TDMA), 코드 분할 다중 접속(Code Division Multiple Access)(CDMA), 직교 주파수 분할 다중화(Orthogonal Frequency Division Multiplexing)(OFDM), 직교 주파수 분할 다중 접속(Orthogonal Frequency Division Multiple Access)(OFDMA), GSM(Global System for Mobile) 통신들, 일반 패킷 무선 서비스(General Packet Radio Service)(GPRS), 유니버설 이동 통신 시스템(Universal Mobile Telecommunications System)(UMTS), cdma2000, 광대역 CDMA(W-CDMA), 고속 다운링크 패킷 접속(High-Speed Downlink Packet Access)(HSDPA), 고속 업링크 패킷 접속(High-Speed Uplink Packet Access)(HSUPA), 고속 패킷 접속(High-Speed Packet Access)(HSPA), LTE(Long Term Evolution), LTE-A(LTE Advanced), 802.11x, Wi-Fi, Zigbee, 초광대역(UWB), 802.16x, 802.15, 홈 노드-B(HnB), 블루투스, 무선 주파수 식별(RFID), IrDA(Infrared Data Association), 근접장 통신들(NFC), 5세대(5G), 새 무선(New Radio)(NR), 그것들의 임의의 조합, 및/또는 임의의 다른 현재 존재하거나 또는 장래에 구현되는 통신 표준 및/또는 프로토콜을 사용하도록 구성될 수도 있다. 일부 실시예들에서, 통신 디바이스(520)는 본 발명의 범위로부터 벗어나는 일 없이 단일, 어레이, 위상, 스위칭, 빔포밍, 빔조향(beamsteering), 그것들의 조합, 및 또는 임의의 다른 안테나 구성인 하나 이상의 안테나들을 포함할 수도 있다.
- [0059] 프로세서(들)(510)는 디스플레이(525), 이를테면 플라즈마 디스플레이, 액정 디스플레이(Liquid Crystal Display)(LCD), 발광 다이오드(Light Emitting Diode)(LED) 디스플레이, 전계 방출 디스플레이(Field Emission Display)(FED), 유기 발광 다이오드(Organic Light Emitting Diode)(OLED) 디스플레이, 플렉시블 OLED 디스플레이

레이, 플렉시블 기관 디스플레이, 프로젝션 디스플레이, 4K 디스플레이, 고품위 디스플레이, 망막[®] 디스플레이, IPS(In-Plane Switching) 디스플레이, 또는 사용자에게 정보를 디스플레이하기 위한 임의의 다른 적합한 디스플레이에 버스(505)를 통해 추가로 커플링된다. 디스플레이(525)는 저항성, 용량성, 표면 음향 파(surface-acoustic wave)(SAW) 용량성, 적외선, 광학적 이미징, 분산적 신호 기술, 음향 펄스 인식, 좌절된 내부 전반사(frustrated total internal reflection) 등을 사용하여 터치(햅틱) 디스플레이, 3차원(3D) 터치 디스플레이, 멀티 입력 터치 디스플레이, 멀티 터치 디스플레이 등으로서 구성될 수도 있다. 임의의 적합한 디스플레이 디바이스 및 햅틱 I/O는 발명의 범위로부터 벗어나는 일 없이 사용될 수도 있다.

[0060] 키보드(530)와 커서 제어 디바이스(535), 이를테면 컴퓨터 마우스, 터치패드 등은 사용자가 컴퓨팅 시스템(500)와 인터페이스하는 것을 가능하게 하기 위해 버스(505)에 추가로 커플링된다. 그러나, 특정 실시예들에서, 물리적 키보드 및 마우스가 존재하지 않을 수도 있고, 사용자는 디스플레이(525) 및/또는 터치패드(도시되지 않음)를 통해 단독으로 디바이스와 상호작용할 수도 있다. 입력 디바이스들의 임의의 유형 및 조합이 설계 선택 사항으로서 사용될 수도 있다. 특정 실시예들에서, 물리적 입력 디바이스 및/또는 디스플레이가 존재하지 않는다. 예를 들면, 사용자는 통신하는 다른 컴퓨팅 시스템을 통해 원격으로 컴퓨팅 시스템(500)과 상호작용할 수도 있거나, 또는 컴퓨팅 시스템(500)은 자율적으로 동작할 수도 있다.

[0061] 메모리(515)는 프로세서(들)(510)에 의해 실행될 때 기능을 제공하는 소프트웨어 모듈들을 저장한다. 그 모듈들은 컴퓨팅 시스템(500)을 위한 운영 체제(540)를 포함한다. 그 모듈들은 본 개시에서 설명되는 프로세스들의 전체 또는 부분 또는 그것의 파생물들을 수행하도록 구성되는 세션 간 자동화 모듈(545)을 더 포함한다. 컴퓨팅 시스템(500)은 추가적인 기능을 포함하는 하나 이상의 추가적인 기능성 모듈들(550)을 포함할 수도 있다.

[0062] 본 기술분야의 통상의 기술자는 "시스템"이 본 발명의 범위로부터 벗어나는 일 없이 서버, 내장형 컴퓨팅 시스템, 개인용 컴퓨터, 콘솔, 개인 정보 단말기(personal digital assistant)(PDA), 셀 전화기, 태블릿 컴퓨팅 디바이스, 양자 컴퓨팅 시스템, 또는 임의의 다른 적합한 컴퓨팅 디바이스, 또는 디바이스들의 조합으로서 실시될 수 있다는 것을 이해할 것이다. 위에서 설명된 기능들을 "시스템"에 의해 수행되는 것으로서 제시하는 것은 본 발명의 범위를 어떠한 방법으로도 제한할 의도는 아니며, 본 발명의 많은 실시예들 중 하나를 제공하려는 의도이다. 실제로, 본 개시물에서 개시되는 방법들, 시스템들, 및 장치들은 클라우드 컴퓨팅 시스템들을 포함하여, 컴퓨팅 기술과 일치하는 국부적이고 분산된 형태들로 구현될 수도 있다.

[0063] 이 명세서에서 설명되는 시스템 특징들의 일부는 그것들의 구현 독립성을 더 크게 강조하기 위하여 모듈들로서 제시되었다는 것에 주의해야 한다. 예를 들어, 모듈이 맞춤형 VLSI(very large scale integration) 회로들 또는 게이트 어레이들, 로직 칩들과 같은 기성(off-the-shelf) 반도체들, 트랜지스터들, 또는 다른 개별 부품들을 포함하는 하드웨어 회로로서 구현될 수도 있다. 모듈이 필드 프로그램가능 게이트 어레이들, 프로그램가능 어레이 로직, 프로그램가능 로직 디바이스들, 그래픽 프로세싱 유닛들 등과 같은 프로그램가능 하드웨어 디바이스들로 또한 구현될 수도 있다.

[0064] 모듈이 다양한 유형들의 프로세서들에 의한 실행을 위해 적어도 부분적으로는 소프트웨어로 또한 구현될 수도 있다. 식별된 실행가능 코드 유닛이, 예를 들면, 오브젝트, 절차, 또는 함수로서 편성될 수도 있는, 예를 들면, 컴퓨터 명령들의 하나 이상의 물리적 또는 논리적 블록들을 포함할 수도 있다. 그럼에도 불구하고, 식별된 모듈의 실행가능물(executable)들은 물리적으로 함께 위치될 필요는 없지만, 논리적으로 함께 연결될 때, 모듈을 포함하고 그 모듈에 대한 언급된 목적을 성취하는 상이한 로케이션들에 저장된 이질적인 명령들을 포함할 수도 있다. 게다가, 모듈들은 본 발명의 범위로부터 벗어나는 일 없이 데이터를 저장하는데 사용되는, 예를 들면, 하드 디스크 드라이브, 플래시 디바이스, RAM, 테이프, 및/또는 임의의 다른 이러한 비일시적 컴퓨터 판독가능 매체일 수도 있는 컴퓨터 판독가능 매체 상에 저장될 수도 있다.

[0065] 실제로, 실행가능 코드 모듈이 단일 명령, 또는 다수의 명령들일 수도 있고, 심지어, 상이한 프로그램들 중에서, 그리고 여러 메모리 디바이스들에 걸쳐 여러 상이한 코드 세그먼트들을 통해 분산될 수도 있다. 마찬가지로, 운용 데이터(operational data)는 본 명세서에서는 모듈들 내에서 식별되고 예시될 수도 있고, 임의의 적합한 형태로 구체화되고 임의의 적합한 유형의 데이터 구조 내에서 조직될 수도 있다. 운용 데이터는 단일 데이터 세트로서 수집될 수도 있거나, 또는 상이한 저장 디바이스들에 걸쳐를 포함하여 상이한 로케이션들에 걸쳐 분산될 수도 있고, 시스템 또는 네트워크 상의 단지 전자 신호들로서 적어도 부분적으로 존재할 수도 있다.

[0066] 기존에, 유인 자동화에서, 인간 사용자들은 동일한 세션에서 실행되고 있는 RPA 로봇이 자신의 태스크들을 완료하는 동안 기다린다. 그러나, 일부 실시예들은 하나 이상의 로봇 세션들을 생성하여 그것들 안에서 RPA 로봇들을 호스팅하고 실행시킨다. 현존 RPA 시스템들과는 달리, 사용자들은 로봇(들)이 로봇 세션(들)에서 자동화를

실행하는 동안 그들의 컴퓨팅 시스템과 상호작용하는 능력으로부터 이익을 얻을 수 있다. 사용자는 일부 실시예들 로봇이 무엇을 하는지를 여전히 모니터링하고 로봇 세션(들) 동안 호스트 자동화 윈도우(들)를 통해 로봇과 상호작용할 수도 있다. 일부 실시예들에서, 사용자 및 로봇 세션(들)은 사용자의 컴퓨팅 시스템에 의해 제어되는 원격 머신 상에서 실행되고 있을 수도 있다.

[0067] 그러나, 일부 실시예들에서, RPA 로봇은 사용자가 IPC 촉진자를 통해 사용하고 있는 애플리케이션들과 상호작용하지 않는다. 대신 RPA 로봇은 사용자에게 가시적이지 않거나 또는 아니면 RPA 촉진자 및 RPA 드라이버를 통해 사용자에게 의해 사용되지 않는 애플리케이션 또는 다른 프로세스와 상호작용할 수도 있다. 이러한 로봇들은 통상적으로 애플리케이션들과의 사용자 상호작용들을 직접 늘리지 않고 무인 로봇들로 간주될 수도 있다.

[0068] 특정 실시예들에서, RPA 로봇은 원격 런타임을 통해 (예컨대, UiPath Remote Runtime™을 통해) 사용자의 컴퓨팅 시스템 및 구동 원격 컴퓨팅 시스템 상에서 실행되고 있을 수도 있다. UiPath Remote Runtime™은 원격 애플리케이션 또는 데스크톱, 이를테면 Citrix Virtual Apps and Desktops™와, 전용 UiPath® 확장판(예컨대, Citrix®를 위한 UiPath® 확장판 또는 Windows® 원격 데스크톱을 위한 UiPath® 확장판) 사이의 통신을 용이하게 하는 컴포넌트이다. UiPath Remote Runtime™은 원격 애플리케이션들의 타겟이 된 UI 엘리먼트들에 관계된 정보를 수집하고 이 정보를 대응하는 확장판에 전송하여서 선택기들이 UI Explorer™에서 기본적으로 생성되게 한다.

[0069] 도 6a 내지 도 6g는 본 발명의 일 실시예에 따른, 로봇 세션에서 실행되고 있는 RPA 로봇, IPC 촉진자, 및 RPA 드라이버를 사용하여 사용자 세션에서 양식을 완성하는 예를 예시한다. 도 6a에는, 사용자가 UI에서 애플리케이션들과 상호작용할 수 있고 현재 실행되고 있는 로봇은 없는 사용자 세션 윈도우(600)가 도시되어 있다. 로봇 트레이 아이콘(610)이 사용자 세션 윈도우(600)의 우측 하부 부분에서 가시적이다. RPA 드라이버(640)는 이 실시예에서 하위 프로세스로서 IPC 촉진자(642)를 포함한다. 그러나, 특정 실시예들에서, IPC 촉진자(642)는 본 발명의 범위로부터 벗어나는 일 없이 RPA 드라이버(640)와 통신하는 별도의 애플리케이션 또는 프로세스일 수도 있다. 이러한 실시예들에서, IPC 촉진자(642)은 IPC를 통해, API 호출을 통해 등으로 RPA 드라이버(640)와 통신할 수도 있다. 사실상, 특정 실시예들에서, IPC가 사용되지 않을 수도 있다.

[0070] 도 6b에서, 사용자는 웹 브라우저를 론칭하고 인보이스 생성 웹 페이지(620)를 방문한다. 도 6c에서, 사용자는 (예컨대, 로봇 트레이 아이콘(610)을 클릭함으로써) 로봇 트레이(612)를 끌어 올리고 자신의 컴퓨팅 시스템 상에서 실행할 클라이언트 세션 로봇 옵션(614)을 선택한다. 실행될 로봇을 선택한 후, 도 6d에 도시된 바와 같이, 로봇 세션을 위한 로봇 세션 윈도우(630)가 스크린 상에 자식 윈도우로서 나타난다. RPA 로봇은 로봇 세션에서 동작할 것이다. 이 실시예에서, 클라이언트 세션 로봇을 위한 윈도우(632)는 로봇 세션 윈도우(630) 내에서 자동으로 론칭되고, 사용자 세션에서 웹 페이지(620)를 위한 데이터를 추출하기 위해 양식 채우기 워크플로우를 실행하기 위한 버튼(634)을 포함한다.

[0071] 일부 실시예들에서, 로봇 세션 윈도우는 디스플레이되지 않을 수도 있고, 로봇은 사용자에게 가시적이 되는 일 없이 자동으로 론칭되며, 동작되고, 닫힐 수도 있다. 특정 실시예들에서, 로봇은 자신의 워크플로우를 완료한 후 자신의 세션을 닫을 수도 있다. 일부 실시예들에서, 로봇 트레이로부터 론칭하는 대신, 로봇 세션이 개시될 수도 있고 로봇은 (예컨대, 사용자가 메인 세션의 애플리케이션에서 버튼을 클릭하는 것으로 인해) 로봇 트레이(612)의 사용 없이 론칭되고 동작한다.

[0072] 도 6e로 가면, 사용자가 버튼(634)을 클릭한 후, 로봇은 웹 페이지(620)에서 양식 필드들을 채우기 위한 데이터를 추출하는 것을 시작한다. 로봇이 데이터의 부분들을 추출함에 따라, 데이터는 IPC 촉진자(642)에 전송되며, 그러면 IPC 촉진자는 RPA 드라이버(640)로 하여금 데이터를 웹 브라우저(620)에 입력하게 한다. 예를 들면, RPA 드라이버(640)는 마우스를 주어진 필드 위로 이동시키고 그 필드로의 데이터로부터의 텍스트의 입력을 유발시킬 수도 있다. 도 6e에서, RPA 드라이버(640)는 이미 완성된 인보이스 번호 필드를 가지고 이제 마우스를 인보이스 금액 필드로 이동시키며, 그것을 클릭하고, 숫자 1을 입력하였다. 캐럿은 도 6e에서 숫자 1 뒤에 나타난다. 특정 실시예들에서, RPA 촉진자(642)는 RPA 드라이버(640)가 웹 페이지(620)의 필드들을 채우도록 하기 전에 로봇(632)으로부터 완전한 데이터 세트를 수신할 수도 있다.

[0073] 위와 같이, 로봇 세션에서 실행되고 있는 로봇과 사용자 세션에서 실행되고 있는 IPC 촉진자(642) 사이의 통신은 IPC 프로토콜을 사용하여 완수된다. IPC 프로토콜들은 네트워크, 파이프들, COM(Component Object Model), RPC(Remote Procedure Calls), 소켓들 등을 통해 통신을 용이하게 할 수도 있다. 적합한 세션 생성 메커니즘들 및 IPC 프로토콜들은 지원되는 경우 다른 운영 체제들에서도 사용될 수도 있다. 로봇은 스테이터스 통지들(예컨대, 로봇이 시작, 실행중, 임시 정지됨 등을 지시함), 추출된 데이터, 에러 메시지들, 커맨드들, 또는 다른

통신물들을 IPC 프로토콜을 통해 다시 IPC 촉진자(642)에 전송할 수도 있다.

- [0074] 로봇에 의해 추출된 데이터로부터의 연관된 텍스트는 웹 페이지(620)가 최소화되지 않거나 또는 다른 윈도우에 의해 덮이지 않을 때 그것이 RPA 드라이버(640)에 의해 입력됨에 따라 사용자에게 가시적이 된다. RPA 드라이버(640)가 그 양식에 대한 데이터 상호작용들을 완료하는 동안, 사용자는 다른 애플리케이션들과 상호작용하고 계속 생산적일 수도 있거나, 또는 심지어 일부 실시예들에서 웹 페이지(620) 자체와 상호작용할 수 있지만, 사용자가 정보를 입력한 동일한 필드에 대한 데이터를 RPA 드라이버(640)가 수정하면 사용자의 입력들은 덮어쓰일 수도 있다.
- [0075] 도 6f로 가면, 로봇이 자신의 워크플로우를 완료한 후, 메시지가 윈도우(632)에 디스플레이된다. 배경에서 알 수 있는 바와 같이, 로봇은 양식 필드들에 대한 데이터 추출을 완료하였고, 양식 필드들을 위한 새로운 엔트리들이 RPA 드라이버(640)에 의한 입력 후에 웹 페이지(620)에서 가시적이 된다. 그 다음에 사용자는 로봇 세션 윈도우(630)를 닫을 수도 있거나, 로봇은 로봇 세션 윈도우(630)(와 잠재적으로 로봇 세션 자체)를 자동으로 닫을 수도 있거나, 또는 로봇 세션 윈도우(630)는 열린 채로 유지될 수도 있다. 그러면 사용자는 완성된 양식을 제출할 수도 있다. 도 6g를 참조한다.
- [0076] 도 7a 내지 도 7g는 본 발명의 일 실시예에 따른, 직접 변수 수정을 통해 로봇 세션에서 실행되고 있는 RPA 로봇을 사용하여 사용자 세션에서 양식을 완성하는 예를 도시한다. 도 7a에는, 사용자가 UI에서 애플리케이션들과 상호작용할 수 있고 현재 실행되고 있는 로봇이 없는 사용자 세션 윈도우(700)가 도시되어 있다. 로봇 트레이 아이콘(710)이 사용자 세션 윈도우(700)의 우측 하부 부분에서 가시적이다.
- [0077] 도 7b에서, 사용자는 웹 브라우저를 론칭하고 인보이스 생성 웹 페이지(720)를 방문한다. 도 7c에서, 사용자는 (예컨대, 로봇 트레이 아이콘(710)을 클릭함으로써) 로봇 트레이(712)를 끌어 올리고 자신의 컴퓨팅 시스템 상에서 실행할 클라이언트 세션 로봇 옵션(714)을 선택한다. 실행될 로봇을 선택한 후, 도 7d에 도시된 바와 같이, 로봇 세션을 위한 로봇 세션 윈도우(730)가 스크린 상에 자식 윈도우로서 나타난다. RPA 로봇은 로봇 세션에서 동작할 것이다. 이 실시예에서, 클라이언트 세션 로봇을 위한 윈도우(732)는 로봇 세션 윈도우(730) 내에서 자동으로 론칭되고, 사용자 세션에서 웹 페이지(720)를 위한 양식 채우기 워크플로우를 실행하기 위한 버튼(734)을 포함한다.
- [0078] 일부 실시예들에서, 로봇 세션 윈도우는 디스플레이되지 않을 수도 있고, 로봇은 사용자에게 가시적이 되는 일 없이 자동으로 론칭되며, 동작되고, 닫힐 수도 있다. 특정 실시예들에서, 로봇은 자신의 워크플로우를 완료한 후 자신의 세션을 닫을 수도 있다. 일부 실시예들에서, 로봇 트레이로부터 론칭하는 대신, 로봇 세션이 개시될 수도 있고 로봇은 (예컨대, 사용자가 메인 세션의 애플리케이션에서 버튼을 클릭하는 것으로 인해) 로봇 트레이(712)의 사용 없이 론칭되고 동작한다.
- [0079] 도 7e로 가면, 사용자가 버튼(734)을 클릭한 후, 로봇은 웹 페이지(720)의 양식 필드들을 위한 변수들에 액세스하고 상호작용함으로써 웹 페이지(720)에서의 양식 필드들을 채우기 시작한다. 예를 들면, 로봇은 본 발명의 범위로부터 벗어나는 일 없이 RAM, 영구적 국부 스토리지, 데이터베이스, 또는 임의의 다른 저장 유형 및/또는 로케이션에 저장되는 필드들을 위한 메모리 값들을 변경시킬 수도 있다. 특정 실시예들에서, 변수들은 웹 양식(720)에 연관된 웹 애플리케이션에 할당되고 해당 웹 애플리케이션에 의해 사용되는 메모리의 일부로서 저장될 수도 있다.
- [0080] 로봇에 의해 변경된 각각의 메모리 값들은 웹 페이지(720)가 리프레시될 때 웹 페이지(720)의 각각의 필드들로 하여금 업데이트 되게 한다. 데이터 수정들을 통해 연관된 텍스트는 웹 페이지(720)가 다른 윈도우에 의해 덮이지 않거나 또는 최소화되지 않을 때 사용자에게 가시적이 된다. RPA 드라이버(640)가 API 레벨 및/또는 네이티브 메커니즘들을 통해 텍스트를 입력하고 마우스를 조작하는 도 6a 내지 도 6g와는 달리, 데이터는 이러한 상호작용들 없이 웹 페이지(720)의 양식 필드들에서 나타난다. 로봇이 그 양식에 대한 데이터 상호작용들을 완료하는 동안, 사용자는 다른 애플리케이션들과 상호작용하고 계속 생산적일 수도 있거나, 또는 심지어 일부 실시예들에서 웹 페이지(720) 자체와 상호작용할 수 있지만, 사용자가 정보를 입력한 동일한 필드에 대한 데이터를 로봇이 수정하면 사용자의 입력들은 덮어쓰일 수도 있다.
- [0081] 도 7f로 가면, 로봇이 자신의 워크플로우를 완료한 후, 메시지가 윈도우(732)에 디스플레이된다. 배경에서 알 수 있는 바와 같이, 로봇은 양식 필드들에 연관된 각각의 메모리 변수들에 대한 데이터 수정들을 완료하였고, 양식 필드들에 대한 새로운 엔트리들은 웹 페이지(720)에서 이 웹페이지가 리프레시된 후에 가시적이 된다. 그 다음에 사용자는 로봇 세션 윈도우(730)를 닫을 수도 있거나, 로봇은 로봇 세션 윈도우(730)(와 잠재적으로 로

봇 세션 자체)를 자동으로 닫을 수도 있거나, 또는 로봇 세션 윈도우(730)는 열린 채로 유지될 수도 있다. 그러면 사용자는 완성된 양식을 제출할 수도 있다. 도 7g를 참조한다.

[0082] 일부 실시예들에서, 로봇 세션은 자식 세션일 수도 있고, 자식 세션은 운영 체제의 자식 세션 API를 통해 생성될 수도 있다. 운영 체제에 의해 제공되는 Windows[®] 단말 서비스들 자식 세션들 또는 다른 자식 세션 API는 본 발명의 범위로부터 벗어나는 일 없이 제2 세션을 생성하기 위해 일부 실시예들에서 사용될 수도 있다. 로봇 트레이 애플리케이션(예컨대, UiPath[®] 로봇 에이전트 데스크톱) 또는 로봇(들)을 론칭하도록 구성되는 다른 애플리케이션은 그 다음에 해당 자식 세션에서 로봇 프로세스를 시작하기 위한 적절한 인수들을 가지고서 운영 체제에서 생성 프로세스 API들을 사용할 수도 있다. 로봇 트레이 애플리케이션 또는 다른 적합한 애플리케이션은 그 다음에 적합한 프로토콜(예컨대, 명명된 파이프들 상에 구축된 것)을 사용하여 로봇 프로세스와 통신할 수도 있다.

[0083] 도 8은 본 발명의 일 실시예에 따른, 로봇 세션에서 실행되고 있는 RPA 로봇, IPC 촉진자, 및 RPA 드라이버를 통해 사용자 세션에서 실행되고 있는 프로세스의 자동화를 위한 프로세스(800)를 예시하는 흐름도이다. 그 프로세스는 810에서 사용자 세션 윈도우를 론칭하는 것으로 시작한다. 이는, 예를 들어, 사용자 컴퓨팅 시스템 상에서 실행되고 있는 운영 체제에 연관되는 메인 윈도우일 수도 있다. 그 다음에 로봇 세션 윈도우는 820에서 사용자 세션 윈도우의 자식 윈도우로서 론칭된다. 일부 실시예들에서, 로봇 세션 윈도우는, 예를 들어, 개시되는 로봇에 응답하여 론칭되거나 또는 달리 론칭될 수도 있다. 그 다음에 로봇은 830에 로봇 세션에서 개시된다.

[0084] 로봇은 840에서 실행되고 IPC 촉진자와는 IPC를 통해 통신한다. IPC 촉진자는 IPC를 통해 로봇으로부터 스테이 터스 통지들, 추출된 데이터, 에러 메시지들, 커맨드들, 또는 다른 통신물들을 수신할 수도 있다. IPC 촉진자는 로봇에 메시지들(이들테면 IPC 촉진자 또는 RPA 드라이버 등으로부터의 커맨드들, 스테이 터스 업데이트들, 에러 메시지들)을 또한 전송할 수도 있다.

[0085] 로봇으로부터 추출된 데이터를 사용하여, RPA 촉진자는 850에서 추출된 데이터 및/또는 로봇으로부터의 다른 통신물들을 사용하여 RPA 드라이버를 제어하거나 또는 아니면 RPA 드라이버가 애플리케이션 상호작용들을 수행하게 한다. 예를 들면, RPA 드라이버는 마우스 이동, 필드들 클릭, 텍스트 입력, 버튼 클릭, 메뉴 내비게이션, 애플리케이션 또는 윈도우를 열기 또는 닫기 등을 하도록 유발될 수도 있다. 로봇이 자신의 실행을 완료한 후, 860에서 로봇 세션은 종료될 수도 있고 일부 실시예들에서 로봇 세션 윈도우는 자동으로 닫힐 수도 있다.

[0086] 도 9는 본 발명의 일 실시예에 따른, 직접 변수 수정을 통해 사용자 세션에서 실행되고 있는 프로세스의 로봇 세션에서 실행되고 있는 RPA 로봇을 통한 자동화를 위한 프로세스(900)를 예시하는 흐름도이다. 그 프로세스는 910에서 사용자 세션 윈도우를 론칭하는 것으로 시작한다. 이는, 예를 들어, 사용자 컴퓨팅 시스템 상에서 실행되고 있는 운영 체제에 연관되는 메인 윈도우일 수도 있다. 그 다음에 로봇 세션 윈도우는 920에서 사용자 세션 윈도우의 자식 윈도우로서 론칭된다. 일부 실시예들에서, 로봇 세션 윈도우는, 예를 들어, 개시되는 로봇에 응답하여 론칭되거나 또는 달리 론칭될 수도 있다. 그 다음에 로봇은 930에서 로봇 세션에서 개시되고 로봇은 940에서 로봇 세션으로부터 사용자 세션에서 실행되고 있는 애플리케이션(들)(예컨대, 웹 페이지들, 스프레 드시트들, ERP 애플리케이션들, 판매 애플리케이션들 등)에 연관된 데이터와 상호작용한다. 예를 들면, 로봇은 사용자 세션에서 애플리케이션(들)에 의해 액세스되고 사용되는 공통 저장된 데이터에서 변경들이 발생하게 하도록 컴퓨팅 시스템 메모리에서의 저장된 정보를 생성, 변경, 또는 삭제할 수도 있다. 로봇이 자신의 실행을 완료한 후, 950에서 로봇 세션은 종료될 수도 있고 일부 실시예들에서 로봇 세션 윈도우는 자동으로 닫힐 수도 있다.

[0087] 일부 실시예들에서, 다수의 실행중인 자동화 프로세스들(예컨대, UI 자동화 프로세스들)이 클라이언트 세션에서 격리 될 수도 있다. 이는, 예를 들어, 이들 프로세스들이 IPC를 통한 메인 세션으로부터의 요구 시 호출되는 것을 허용한다. 이런 방식으로, 다수의 RPA 로봇 프로세스들은 메인 세션 애플리케이션, 메인 세션 RPA 로봇, 다른 클라이언트 세션 RPA 로봇 등으로부터 호출되는 요구 시 이용 가능할 수도 있다. 클라이언트 세션에서 실행되고 있는 RPA 로봇들은 또한 메인 세션에서부터 데이터를 수집할 수도 있다. 특정 실시예들에서, 클라이언 트 세션에서 실행되고 있는 RPA 로봇들은 그것들이 그것들의 자동화 또는 자동화의 일부를 수행하기 전에 메인 세션으로부터의 특정한 트리거를 기다릴 수도 있다.

[0088] 도 10은 본 발명의 일 실시예에 따른, 메인 세션 애플리케이션과 상호작용하는 다수의 클라이언트 세션 RPA 로 봇들의 일 예를 도시하는 아키텍처 도면(1000)이다. 도 10에서, 메인 세션(1010)과 클라이언트 세션(1020)이

실행되고 있다. 클라이언트 세션(1020)은 클라이언트 세션 RPA 로봇 1(1030)과 클라이언트 세션 RPA 로봇 2(1032)를 포함한다.

- [0089] 클라이언트 세션 RPA 로봇들(1030, 1032)은 별도의 IPC 촉진자 애플리케이션(1040) 또는 내장된 RPA 촉진자 프로세스 또는 드라이버(1050)의 일부인 서브루틴(1052)과 통신한다. 일부 실시예들에서, 드라이버(1050)는, 예를 들어, 도 3의 드라이버(340)일 수도 있다. IPC 촉진자(1040 또는 1052)는 클라이언트 세션 RPA 로봇(1030 또는 1032)의 각각의 워크플로우 로직에 의해 묘사되는 방식으로 드라이버(1050)로 하여금 메인 세션 애플리케이션 및/또는 연관된 애플리케이션 오브젝트(1060)(예컨대, Microsoft Windows[®] 애플리케이션을 위한 COM(Component Object Model) 오브젝트)와 상호작용하게 한다. 예를 들면, 드라이버(1050)는 메인 세션 애플리케이션(1060)에서 마우스를 이동시키며, 양식 데이터를 채우며, 하나 이상의 버튼들을 클릭하며, 메뉴와 상호작용하며, 그것들의 조합을 하도록 유발될 수도 있다. 특정 실시예들에서, RPA 촉진자 애플리케이션(1040)은 메인 세션 애플리케이션 및/또는 연관된 애플리케이션 오브젝트(1060)와 직접 통신할 수도 있다.
- [0090] 특정 실시예들에서, 클라이언트 세션에서 실행되고 있는 RPA 로봇들은 유인 자동화 로봇들일 수도 있다. 예를 들면, 클라이언트 세션에서 실행되고 있는 RPA 로봇은 메인 세션에서 사용자에게 입력을 요청할 수도 있다. 이는 메인 세션 로봇, 드라이버 등으로 IPC를 통해 완수될 수도 있다.
- [0091] 도 11은 본 발명의 일 실시예에 따른, 클라이언트 세션에서 유인 자동화 RPA 로봇을 실행하기 위한 프로세스(1100)를 예시하는 흐름도이다. 이 실시예에서, 사용자는 1120에서 클라이언트 세션 RPA 로봇 프로세스로 하여금 IPC를 통해 시작되게 하는 메인 세션 애플리케이션(1110)을 이용한다. 일부 실시예들에서, 메인 세션 애플리케이션(1110)은 메인 세션에서의 사용자가 클라이언트 세션에서의 유인 RPA 로봇과 상호작용하는 것을 허용하는 RPA 보조 애플리케이션일 수도 있다. 예를 들어, RPA 보조 애플리케이션의 대시보드가 로봇에 의해 지원되는 현재 RPA 프로세스들을 보여줄 수도 있고, 사용자는 실행될 프로세스를 선택할 수 있다. 특정 실시예들에서, RPA 보조 애플리케이션은 클라이언트 세션 RPA 로봇이 실행되고 있는 동안 클라이언트 세션 워크플로우의 단계들 및/또는 실행 스테이터스를 디스플레이할 수도 있다.
- [0092] 클라이언트 세션 RPA 로봇이 1120에서 시작한 후, 클라이언트 세션 RPA 로봇은 1130에서 자신의 로직의 첫 번째 부분을 실행한다. 1130에서의 로직의 첫 번째 부분의 실행 후, 클라이언트 세션 RPA 로봇은 1140에서 사용자 입력을 요청한다. 이는 메인 세션 애플리케이션과 상호작용하는 IPC 촉진자 및 드라이버를 통해 나타나는 메인 세션에서 실행되고 있는 애플리케이션에서의 메시지, 메인 세션에서 실행되고 있는 RPA 로봇으로부터의 메시지 등을 통해 일어날 수 있다. 그러면 사용자는 요청된 입력을 제공하며, 그 입력은 IPC를 통해 클라이언트 세션 RPA 로봇에 전송되고, 로직의 두 번째 부분은 1150에서 이 입력을 사용하여 실행된다. 클라이언트 세션 RPA 로봇 프로세스가 1160에서 종료된 후, 메인 세션 애플리케이션(1110)은 RPA 로봇 프로세스가 종료했다는 통지를 수신할 수도 있다.
- [0093] 특정 실시예들에서, 자동화의 부분들은 메인 세션에서 실행될 수도 있고 자동화의 다른 부분들은 클라이언트 세션에서 실행될 수도 있다. 예를 들면, 메인 세션 애플리케이션 또는 RPA 로봇은 메인 세션에서 특정한 동작들을 수행한 다음, 메인 세션 애플리케이션 또는 로봇이 자신의 실행에서의 특정한 지점에 도달하며, 트리거를 수신하며, 논리적 조건이 충족되는 동일 때 클라이언트 세션에서 실행되고 있는 RPA 로봇으로 하여금 자동화의 다른 일부를 실행하게 할 수도 있다. 클라이언트 세션 RPA 로봇은 자신의 자동화의 일부를 메인 세션 애플리케이션 또는 로봇과 순차적으로 또는 병행하여 수행할 수 있다. 그 다음에 클라이언트 세션 RPA 로봇은 지시, 요청된 데이터, 실행 결과들 등을 메인 세션 애플리케이션 또는 로봇에 제공할 수 있다.
- [0094] 도 12는 본 발명의 일 실시예에 따른, 메인 세션 로봇(M1), 클라이언트 세션 애플리케이션(A1), 및 한 쌍의 클라이언트 세션 로봇들(C1 및 C2) 사이의 다중 로봇 집단 워크플로우의 실행을 예시하는 흐름도(1200)이다. A1, C1, 및/또는 C2는 동일한 클라이언트 세션 또는 상이한 클라이언트 세션들에 있을 수도 있다. M1은 자신의 워크플로우를 실행하는 것을 시작하고 워크플로우를 완료하기 위해 C1을 호출하는 활동에 도달한다. 이는 일부 실시예들에서 클라이언트 세션에서 RPA 프로세스를 실행하도록 설정된 플래그를 갖는 "호출 워크플로우(invoked workflow)" 활동에 의해 수행될 수도 있다. M1은 C1으로 하여금 자신의 워크플로우를 IPC를 통해 실행하게 하고 호출 활동은 대기한다. 이 시간 동안, M1은 일부 실시예들에서 다른 태스크들을 수행할 수도 있다.
- [0095] C1이 자신의 워크플로우를 완료한 후, C1은 IPC를 통해 M1에게 통지하고, M1은 태스크의 실행을 시작하거나 또는 그 태스크를 완료하기 위해 클라이언트 세션에서의 프로세스인 A1을 호출하는 활동에 도달하기까지 실행을 재개한다. M1은 causes A1으로 하여금 IPC를 통해 태스크를 실행 및/또는 수행하게 하고 호출 활동(예컨대, "호출 프로세스(invoked process)" 활동)은 대기한다. 호출 프로세스 활동은 일부 실시예들에서는 클라이언트 세

선에서 A1을 실행하도록 설정된 플래그를 가질 수도 있다. 다시 한번, M1은 일부 실시예들에서 A1을 기다리는 동안 다른 태스크들을 수행할 수도 있다.

[0096] A1이 자신의 프로세스 또는 태스크를 완료한 후, A1은 IPC를 통해 M1에게 통지하고, M1은 워크플로우를 완료하기 위해 C2를 호출하는 활동에 도달하기까지 실행을 재개한다. M1은 C2로 하여금 IPC를 통해 자신의 워크플로우를 실행하게 하고 호출 활동은 대기한다. 다시 한번, M1은 일부 실시예들에서 C2를 기다리면서 다른 태스크들을 수행할 수도 있다.

[0097] C2가 자신의 워크플로우를 완료한 후, M1은 IPC를 통해 통지를 받고, 워크플로우를 완료하기 위해 C1을 다시 한번 호출하는 활동에 도달하기까지 실행을 재개한다. 이는 이전에 실행된 또는 상이한 워크플로우 또는 활동 세트와는 동일한 워크플로우 활동들일 수도 있다. M1은 C1로 하여금 자신의 워크플로우를 IPC를 통해 실행하게 하고 호출 활동은 대기한다. 또 다시, M1은 일부 실시예들에서 C1을 기다리는 동안 다른 태스크들을 수행할 수도 있다. C1이 자신의 워크플로우를 완료한 후, M1은 IPC를 통해 통지를 받고 M1의 워크플로우가 완료되기까지 실행을 재개한다. 특정 실시예들에서, C1은 처음에 호출된 후 실행을 유지할 수도 있고 M1에 의해 다시 호출될 때 여전히 실행중일 수도 있다.

[0098] 위와 같이, 일부 실시예들에서, M1은 클라이언트 세션에서 RPA 로봇(들)(예컨대, C1 및/또는 C2) 및/또는 다른 프로세스들(예컨대, A1)을 호출할 수도 있는 하나 이상의 "호출 워크플로우" 활동들 및/또는 "호출 프로세스" 활동들을 포함할 수도 있다. 이러한 활동은 각각의 RPA 로봇(들) 및/또는 프로세스(들)를 시작하며, 이미 실행 중인 RPA 로봇(들) 및/또는 프로세스(들)와 통신하는 등을 할 수도 있다. 이들 활동들은 RPA 개발자가 M1을 위한 워크플로우를 개발할 때 생성될 수도 있다. 특정 실시예들에서, RPA 로봇(들)은 RPA 촉진자 또는 다른 적합한 기능을 통해 클라이언트 세션에서부터 메인 세션으로 RPA 로봇들 및/또는 프로세스들을 호출할 수도 있다.

[0099] 실행 동안, M1은 IPC를 통해 A1, C1, 및 C2와 통신할 수도 있고, 반대의 경우일 수도 있다. 예를 들면, M1은 A1, C1, 및 C2에 코멘트들을 전송하고 요청할 수도 있고, M1은 A1, C1, 및 C2로부터 스테이터스 메시지들, 결과들, 및 에러 메시지들을 수신할 수도 있다는 등이다. 이런 방식으로, M1은 클라이언트 세션에서 클라이언트 세션 RPA 로봇들 및/또는 다른 프로세스들(예컨대, 애플리케이션들)을 제어하는 메인 RPA 로봇으로서 역할을 할 수도 있다. 특정 실시예들에서, M1, A1, C1, 및/또는 C2는 병행하여 동작할 수도 있다.

[0100] 일부 실시예들에서, M1은 메인 세션에서 실행되고 있는 애플리케이션들과 모든 상호작용들을 수행할 수도 있으며, 이는 IPC 촉진자의 필요성을 제거할 수도 있다. 따라서, C1 및 C2는 다양한 데이터 추출 및 프로세싱 태스크들을 수행할 수도 있고 M1은 UI 상호작용 태스크들을 수행할 수도 있다. 특정 실시예들에서, M1, C1, 및/또는 C2의 적어도 일부 워크플로우 활동들은 병행하여 실행될 수도 있다. 하나의 이러한 예가 도 13의 흐름도(1300)에서 도시되는데, 거기서 C1은 M1의 로직에 기초하여 실행되고 실행은 병행하여 발생한다. 통신물들은 M1과 C1 사이에서 C1의 섹션 동안 전송될 수도 있다. 그러나, 특정 실시예들에서, C1은 자신의 실행을 독립적으로 완료할 수도 있고, M1이 종료된 후 자신의 워크플로우의 실행을 완료할 수도 있다.

[0101] 일부 실시예들에서, 모든 RPA 로봇 활동들은 클라이언트 세션에서 실행된다. RPA 로봇들은, 예를 들면, UiPath[®] 로봇 트레이, IPC 촉진자, 다른 애플리케이션 등과 같은 로봇 실행자에 의해 메인 세션으로부터 호출될 수도 있다. RPA 로봇들의 실행의 결과들 및 다른 통신물들은 IPC를 통해 로봇 실행자 또는 다른 애플리케이션에 제공될 수 있다.

[0102] 특정 실시예들에서, 표준화된 통신 포맷이 클라이언트 세션과 메인 세션 사이의 IPC 통신들을 위해 사용될 수도 있다. 예를 들면, 메인 세션 및 클라이언트 세션 프로세스들은 XAML 데이터를 교환할 수도 있다. 이는 통신물들이 각각의 프로세스들이 인식할 수 있는 포맷으로 있는 것을 보장하는데 도움이 된다.

[0103] 일부 실시예들에서, IPC는 동기화 목적으로 동일한 세션에서 실행되고 있는 RPA 프로세스들 사이에서 사용될 수도 있다. 예를 들면, 다수의 RPA 프로세스들은 메인 세션에서의 포어그라운드 프로세스와는 백그라운드에서 병행하여 실행될 수도 있다. IPC는 백그라운드 프로세스들과 포어그라운드 또는 메인 프로세스 사이에 정보를 교환하는 동기화 메커니즘을 제공할 수도 있다.

[0104] 특정 실시예들에서, 드라이버(예컨대, 도 3의 드라이버(340))는 메인 RPA 워크플로우 프로세스에서 로드될 것이다. 지식 RPA 프로세스와 메인 RPA 워크플로우 프로세스 사이의 통신은 IPC를 통해 발생할 수도 있다. 이는 메인 RPA 워크플로우 프로세스가 드라이버를 이용하여 IPC 정보에 기초하여 클라이언트 세션 로봇을 위한 동작들을 수행하는 것을 허용한다.

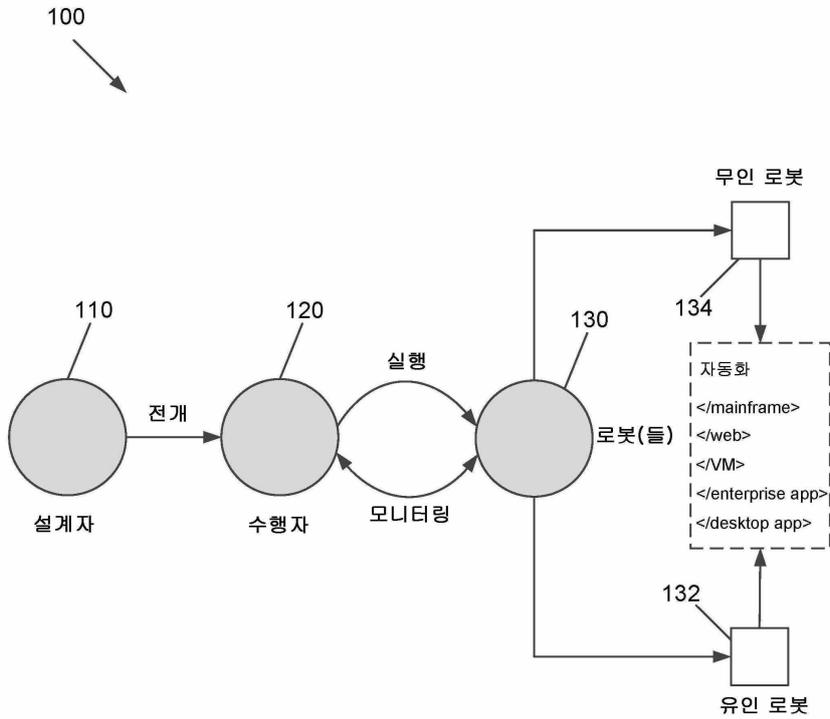
- [0105] 일부 실시예들에서, 클라이언트 세션 RPA 로봇은 메인 세션에서 열린 애플리케이션들과 함께 작업할 수도 있다. 예를 들면, 클라이언트 세션 RPA 로봇은 메인 세션에서 열린 Excel[®] 스프레드시트와 함께 작업할 수도 있다. 이는, 예를 들어, 클라이언트 세션 RPA 로봇과 RPA 촉진자 사이에 IPC를 통해 발생할 수 있다. RPA 촉진자는 드라이버로 하여금 클라이언트 세션 RPA 로봇에 의해 제공된 정보에 기초하여 Excel[®] 스프레드시트와 상호작용(들)(예컨대, 스프레드시트에 데이터를 입력하는 것, 테이블 값들을 수정하는 것 등)을 수행하게 할 수도 있다. 특정 실시예들에서, 활동들(예컨대, Microsoft Office[®] 활동들, 웹 브라우저 활동들 등)은 양 세션들 사이에서 애플리케이션 데이터를 자동으로 캡처하는 RPA 워크플로우에 포함될 수도 있다. 이런 방식으로, 클라이언트 세션 RPA 로봇은 본질적으로 메인 세션에서 실행되고 있는 애플리케이션들과는 그것들이 RPA 로봇 소유의 세션에서 실행되고 있다면 함께 작업한다. 실제로, 일부 실시예들에서, RPA 개발자는 자신이 개발하고 있는 워크플로우들 또는 그 일부들이 클라이언트 세션에서 실행될 것임을 알지 못할 수도 있다.
- [0106] 특정 실시예들에서, 예를 들어, 설계자 애플리케이션을 사용하여 RPA 로봇을 설계할 때, RPA 개발자는 "호출 워크플로우" 활동 또는 "호출 프로세스" 활동에 대해 클라이언트 세션 플래그를 설정할 수도 있다. 호출 워크플로우 활동은 RPA 워크플로우를 호출한다. 호출 프로세스 활동은 국부 머신에 이용 가능한 프로세스를 실행한다. 이 플래그를 설정하는 것은 RPA 워크플로우 또는 프로세스가 메인 세션보다는 클라이언트 세션에서 호출되게 할 수도 있다.
- [0107] 일부 실시예들에서, 클라이언트 세션 RPA 로봇의 실행은 메인 세션 애플리케이션 및/또는 RPA 로봇으로부터의 트리거에 기초 할 수도 있다. 도 14는 본 발명의 일 실시예에 따른, 메인 세션 애플리케이션에 대한 트리거에 기초한 클라이언트 세션 RPA 로봇의 실행을 예시하는 흐름도(1400)이다. 메인 세션 애플리케이션(A1)(이는 일부 실시예들에서 RPA 로봇일 수도 있음)이 실행을 시작하고 트리거를 기다린다. 트리거가 수신된 후, A1은 IPC를 통해 클라이언트 세션 RPA 로봇(C1)의 실행을 시작한다. 통신물들은 C1의 섹션 동안 A1과 C1 사이에서 전송될 수도 있다. 그 다음에 A1은 다른 트리거를 기다리고 C1은 자신의 워크플로우를 실행한다.
- [0108] 도 15는 본 발명의 일 실시예에 따른, 세션 간 자동화를 수행하기 위한 프로세스(1500)를 예시하는 흐름도이다. 그 프로세스는 1510에서 컴퓨팅 시스템의 메인 세션에서 RPA 로봇을 실행하는 것으로 시작한다. RPA 로봇 또는 애플리케이션이 1520에서 메인 세션 RPA 로봇에 의해 IPC를 통해 클라이언트 세션에서 호출 또는 론칭된다. 메시지들은 1530에서 IPC를 통해 클라이언트 세션 RPA 로봇 또는 애플리케이션과 메인 세션 RPA 로봇 사이에 통신된다. 일부 실시예들에서, 메인 세션 RPA 로봇은 1540에서 클라이언트 세션 RPA 로봇 또는 애플리케이션의 실행이 종료되기까지 자신의 실행을 일시 정지시킨다.
- [0109] 클라이언트 세션 RPA 로봇 또는 애플리케이션의 실행의 결과(들)은 1550에서 메인 세션 RPA 로봇에 의해 IPC를 통해 수신된다. 그 다음에 메인 세션 RPA 로봇은 메인 세션 RPA 로봇의 워크플로우의 적어도 부분을 완료하기 위해, 메인 세션에서 실행되고 있는 애플리케이션 또는 애플리케이션 오브젝트와 상호작용하기 위해, 또는 둘 다를 위해 1560에서 결과(들)를 사용한다. 일부 실시예들에서, 프로세스는 1570에서 다음 클라이언트 세션 RPA 로봇 또는 클라이언트 세션 애플리케이션에 대해 반복되고, 프로세스는 단계 1520로 복귀한다.
- [0110] 일부 실시예들에서, 클라이언트 세션 RPA 로봇, 클라이언트 세션 애플리케이션, 또는 둘 다는, 메인 세션 RPA 로봇의 워크플로우에서 두 번 이상 호출되거나 또는 론칭된다. 특정 실시예들에서, 클라이언트 세션 애플리케이션은 메인 세션 RPA 로봇의 워크플로우의 호출 프로세스 활동을 통해 호출 또는 론칭되거나, 클라이언트 세션 RPA 로봇은 메인 세션 RPA 로봇의 워크플로우의 호출 워크플로우 활동을 통해 호출 또는 론칭되거나, 또는 둘 다가 행해진다. 일부 실시예들에서, 호출 프로세스 활동은 클라이언트 세션 애플리케이션이 클라이언트 세션에서 실행됨을 지시하는 플래그를 포함하거나, 호출 워크플로우 활동은 클라이언트 세션 RPA 로봇이 클라이언트 세션에서 실행됨을 지시하는 플래그를 포함하거나, 또는 둘 다이다. 특정 실시예들에서, 클라이언트 세션 애플리케이션, 클라이언트 세션 RPA 로봇, 또는 둘 다는, 결과들을 제공한 후 계속 실행되고 있다. 일부 실시예들에서, 메인 세션 RPA 로봇과 클라이언트 세션 RPA 로봇 및/또는 클라이언트 세션 애플리케이션 사이의 IPC 통신들은 표준화된 통신 포맷으로 되어 있다.
- [0111] 도 8 내지 도 15에서 수행되는 프로세스 단계들은 본 발명의 실시예들에 따라, 도 8 내지 도 15에서 설명된 프로세스(들) 중 적어도 일부를 수행하기 위한 프로세서(들)를 위한 명령들을 인코딩하는 컴퓨터 프로그램에 의해 수행될 수도 있다. 컴퓨터 프로그램은 비일시적 컴퓨터 판독가능 매체 상에 수록될 수도 있다. 컴퓨터 판독가능 매체는 데이터를 저장하는데 사용되는 하드 디스크 드라이브, 플래시 디바이스, RAM, 테이프, 및/또는 임의의 다른 이러한 매체 또는 매체들의 조합일 수도 있지만 이것들로 제한되지 않는다. 컴퓨터 프로그램은 컴퓨터

관독가능 매체 상에 또한 저장될 수도 있는 도 8 내지 도 15에서 설명되는 프로세스 단계들의 전체 또는 부분을 구현하기 위해 컴퓨팅 시스템의 프로세서(들)(예컨대, 도 5의 컴퓨팅 시스템(500)의 프로세서(들)(510))를 제어하기 위한 인코딩된 명령들을 포함할 수도 있다.

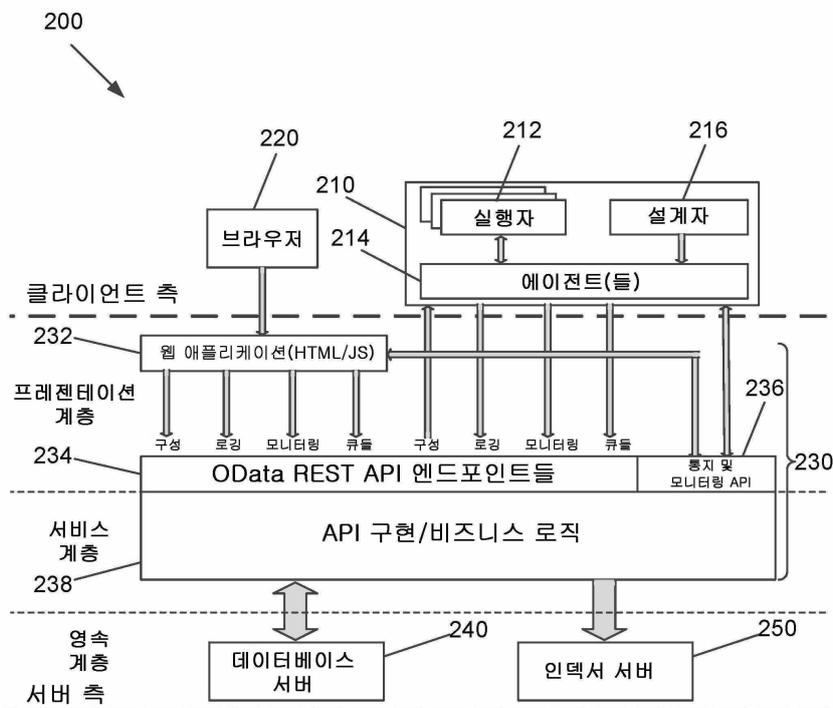
- [0112] 컴퓨터 프로그램은 하드웨어, 소프트웨어, 또는 하이브리드 구현으로 구현될 수 있다. 컴퓨터 프로그램은 서로 동작적으로 통신하는, 그리고 정보 또는 명령들을 디스플레이하기 위해 전달하도록 설계되는 모듈들로 구성될 수 있다. 컴퓨터 프로그램은 범용 컴퓨터, ASIC, 또는 임의의 다른 적합한 디바이스 상에서 동작하도록 구성될 수 있다.
- [0113] 본 발명의 다양한 실시예들의 컴포넌트들은, 본 개시에서 일반적으로 설명되고 도면들에서 예시된 바와 같이, 매우 다양한 상이한 구성들로 배열되고 설계될 수도 있다는 것이 쉽사리 이해될 것이다. 따라서, 본 발명의 실시예들의 상세한 설명은, 첨부된 도면들에서 나타난 바와 같이, 청구된 바와 같은 발명의 범위로 제한될 의도는 아니고, 본 발명의 선택된 실시예들을 나타낼 뿐이다.
- [0114] 이 명세서의 전체에 걸쳐 설명되는 본 발명의 특징들, 구조들, 또는 특성들은 하나 이상의 실시예들에서 임의의 적합한 방식으로 결합될 수도 있다. 예를 들어, "특정한 실시예들", "일부 실시예들", 또는 유사한 언어표현에 대한 이 명세서의 전체에 걸친 언급은 실시예에 관련하여 설명된 특정 특징, 구조, 또는 특성이 본 발명의 적어도 하나의 실시예에 포함되는 것을 의미한다. 따라서, 이 명세서의 전체에 걸친 "특정 실시예들에서", "일부 실시예에서", "다른 실시예들에서", 또는 유사한 언어표현의 문구들의 출현은 모두가 반드시 실시예들의 동일한 그룹을 지칭하는 것은 아니고 설명된 특징들, 구조들, 또는 특성들은 하나 이상의 실시예들에서 임의의 적합한 방식으로 결합될 수도 있다.
- [0115] 특징들, 장점들, 또는 유사한 언어에 대한 이 명세서의 전체에 걸친 언급은 본 발명으로 실현될 수도 있는 특징들 및 장점들의 모두가 본 발명의 임의의 단일 실시예이거나 또는 그러한 실시예 안에 있음을 의미하지 않는다는 것에 주의해야 한다. 오히려, 특징들 및 장점들을 언급하는 언어표현은 일 실시예에 관련하여 설명되는 특정 특징, 장점, 또는 특성이 본 발명의 적어도 하나의 실시예에 포함된다는 것을 의미하는 것으로 이해된다. 따라서, 이 명세서의 전체에 걸친 특징들 및 장점들, 및 유사한 언어의 논의는 동일한 실시예를 참조하지만 반드시 그런 것은 아닐 수도 있다.
- [0116] 더욱이, 본 발명의 설명된 특징들, 장점들, 및 특성들은 하나 이상의 실시예들에서 임의의 적합한 방식으로 결합될 수도 있다. 본 기술분야의 통상의 기술자는 본 발명이 특정 실시예의 특정 특징들 또는 장점들 하나 이상 없이도 실시될 수 있다는 것을 인식할 것이다. 다른 경우들에서, 추가적인 특징들 및 장점들은 본 발명의 모든 실시예들에서 존재하지 않을 수도 있는 특정 실시예들에서 인식될 수도 있다.
- [0117] 본 기술분야의 통상의 기술을 가진 자는 위에서 논의된 바와 같은 본 발명이 상이한 순서의 단계들로, 그리고/또는 개시된 것들과는 상이한 구성들의 하드웨어 엘리먼트들로 실시될 수도 있다는 것을 쉽사리 이해할 것이다. 그러므로, 본 발명이 이들 바람직한 실시예들에 기초하여 설명되었지만, 특정한 수정들, 변경들, 및 대안적 구성들이 본 발명의 정신 및 범위 내에 있으면서도 명백할 것이라는 것이 본 기술분야의 통상의 기술자에게 명백할 것이다. 그러므로, 본 발명의 한계와 경계를 결정하기 위하여, 첨부된 청구항들이 참조되어야 한다.

도면

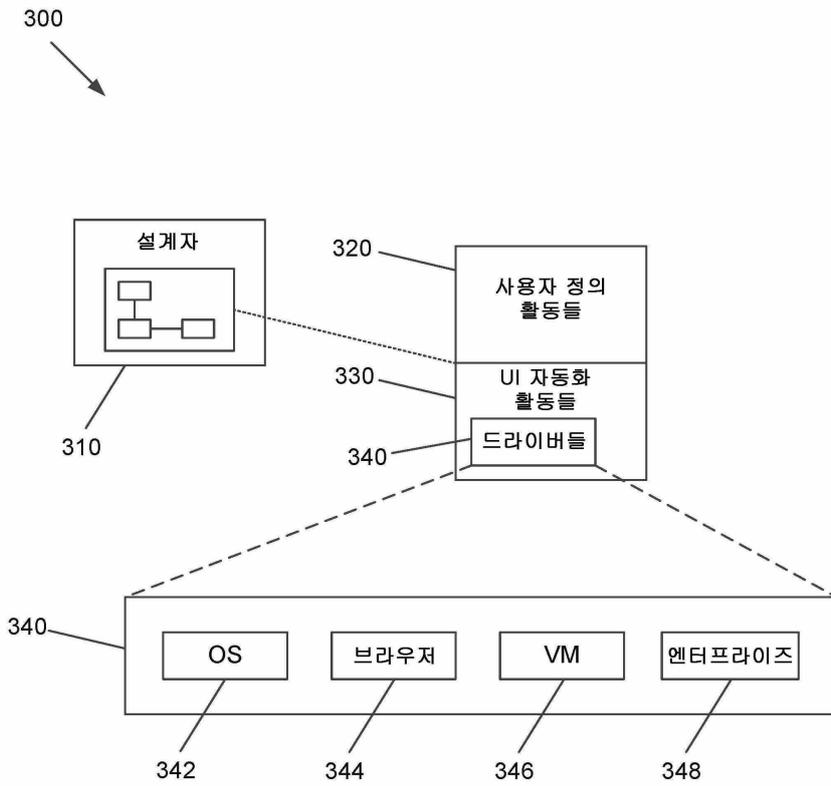
도면1



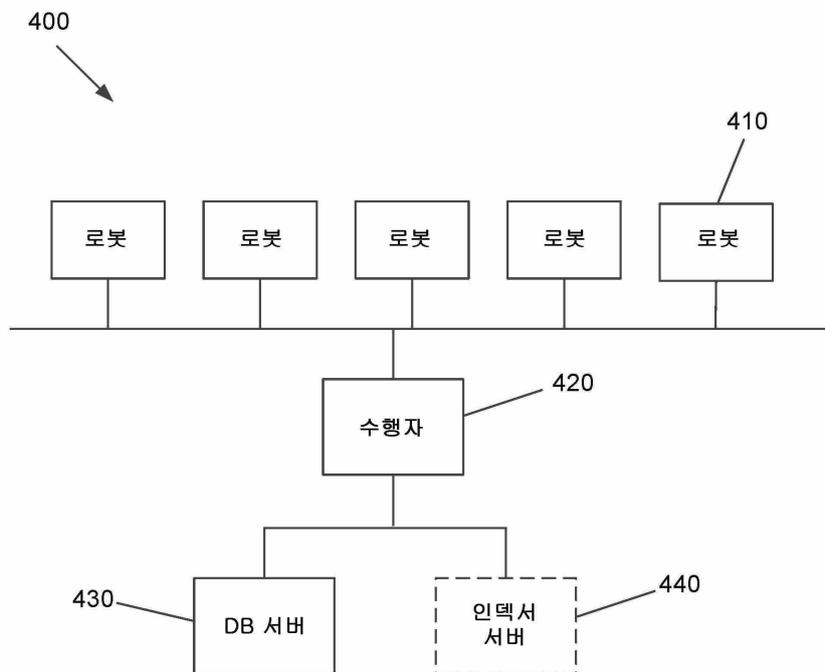
도면2



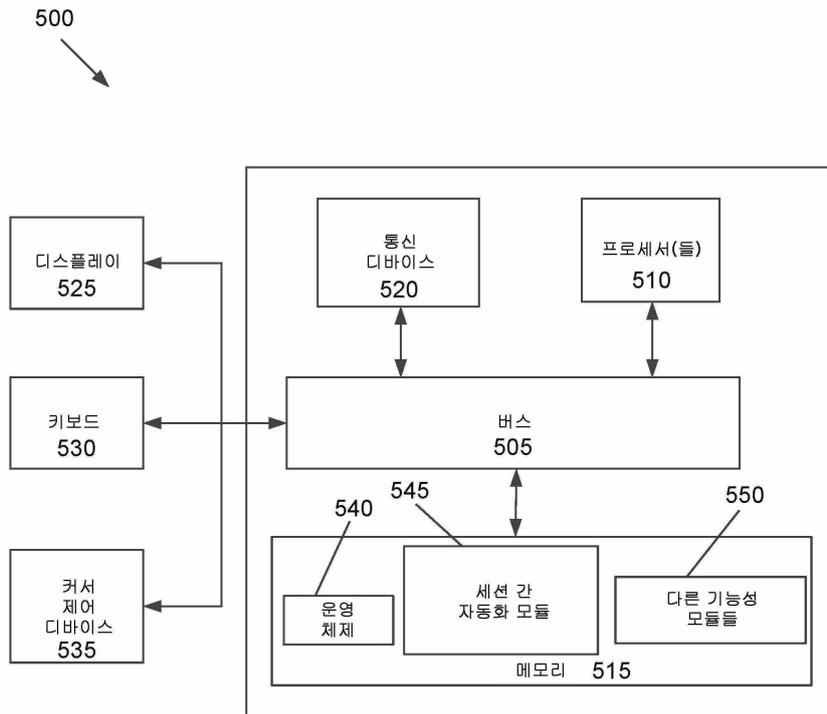
도면3



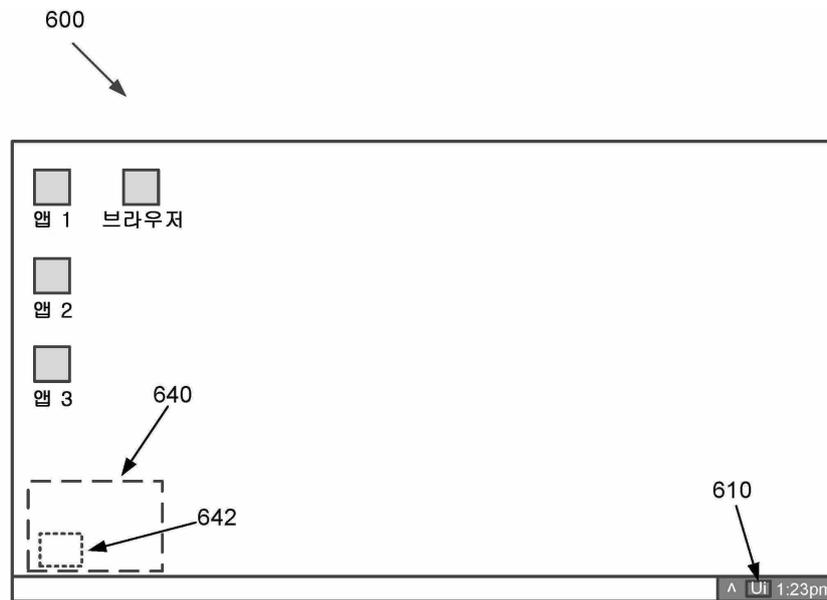
도면4



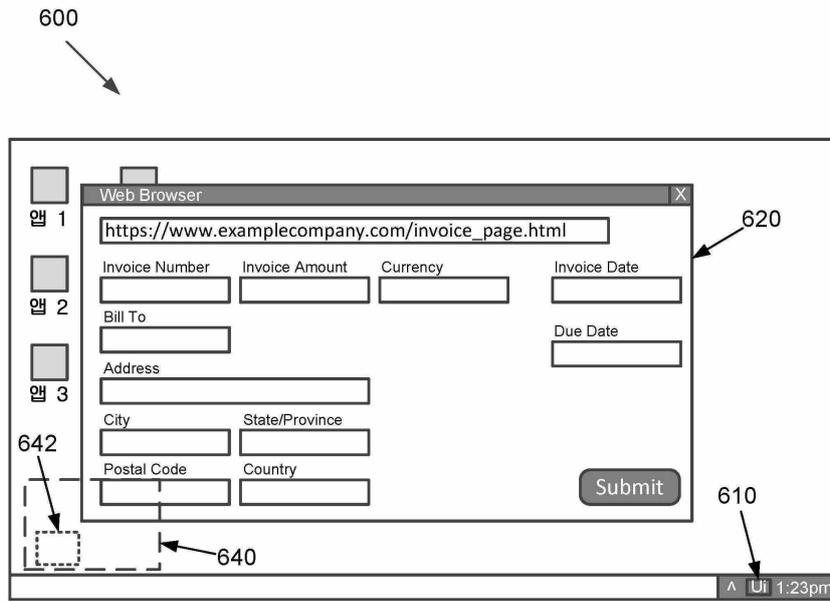
도면5



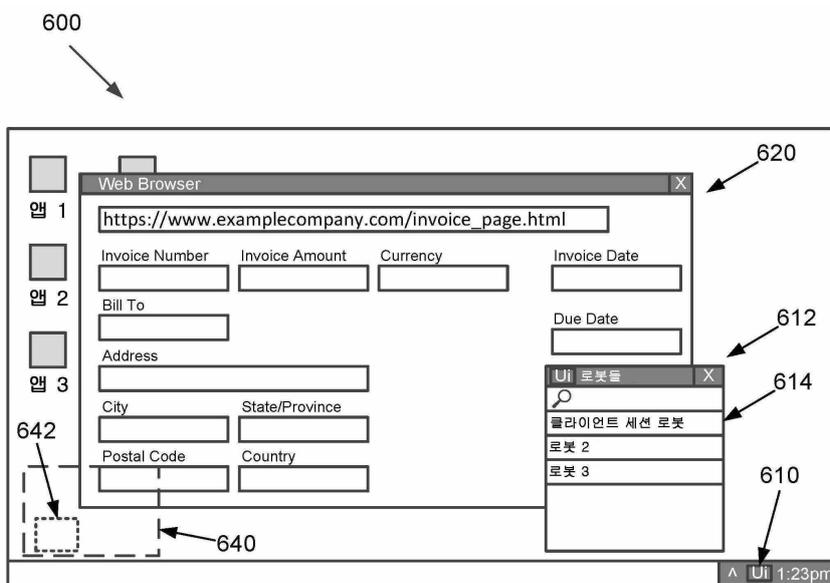
도면6a



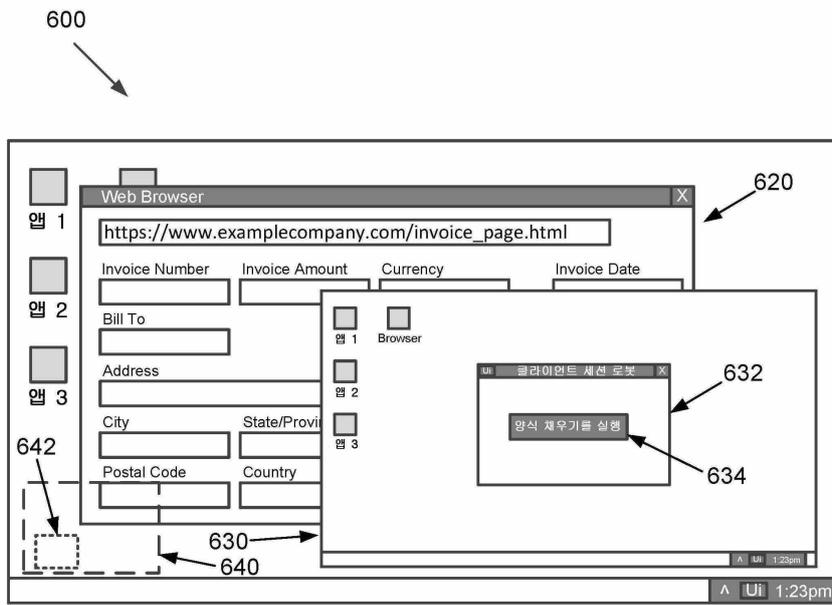
도면6b



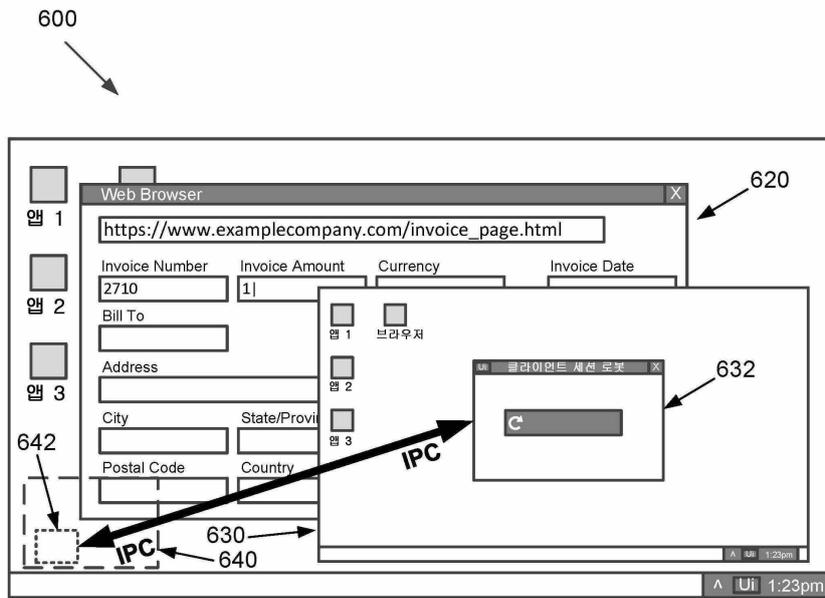
도면6c



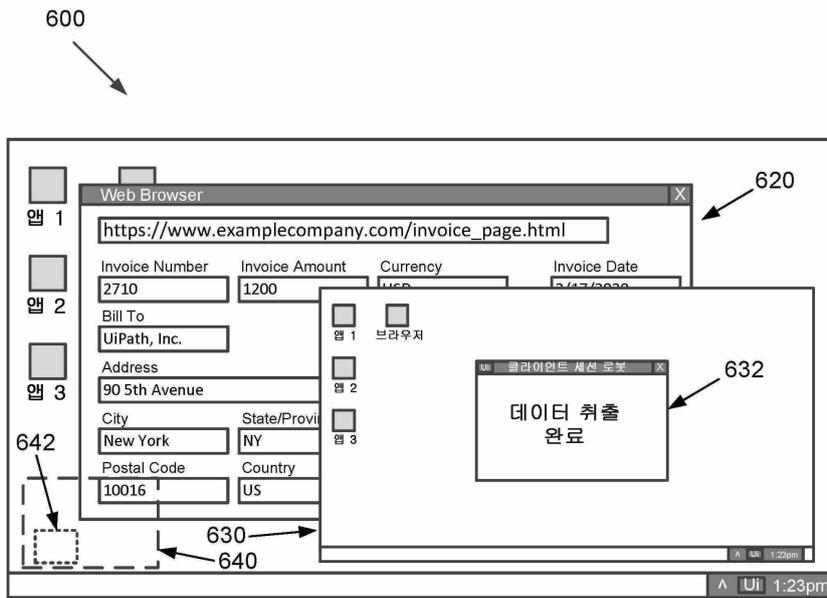
도면6d



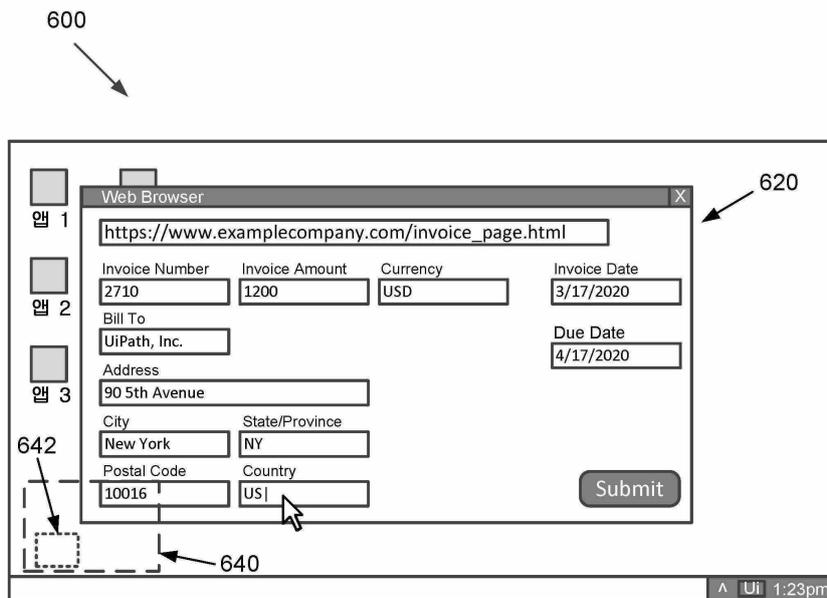
도면6e



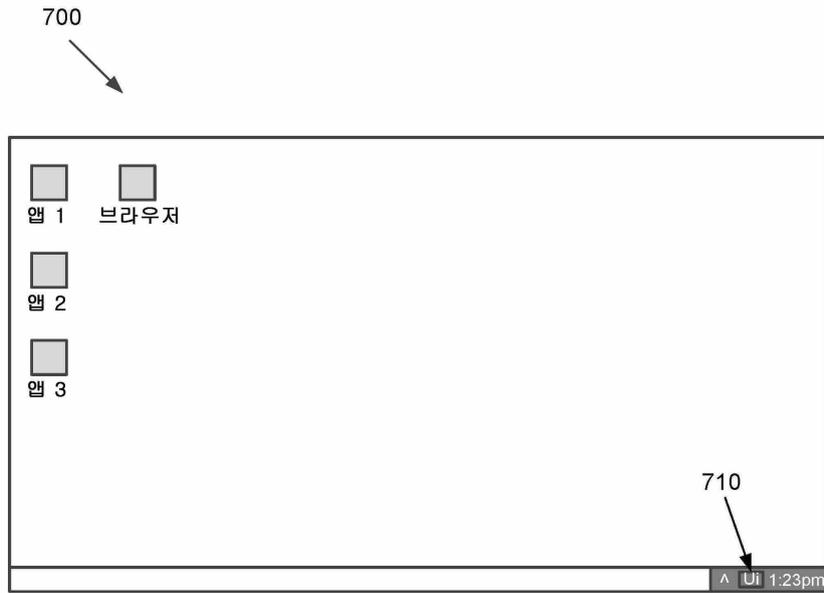
도면6f



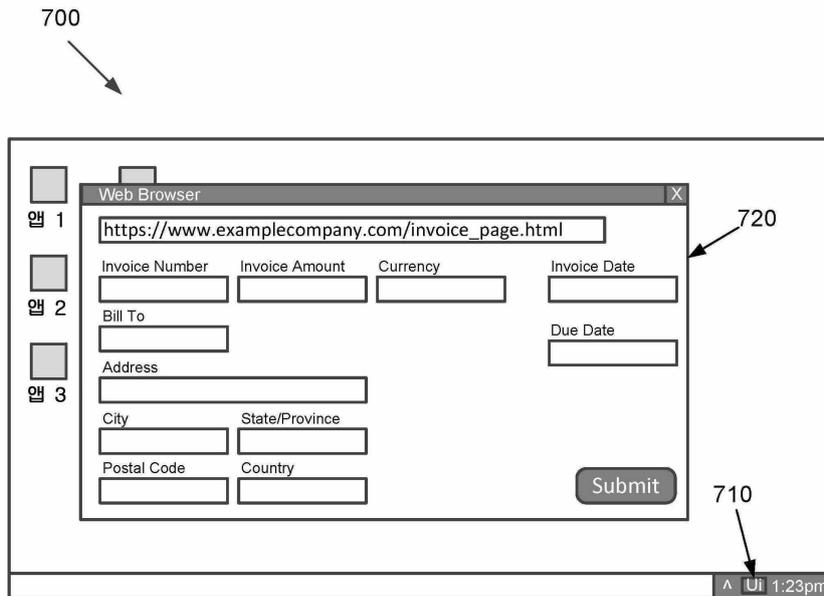
도면6g



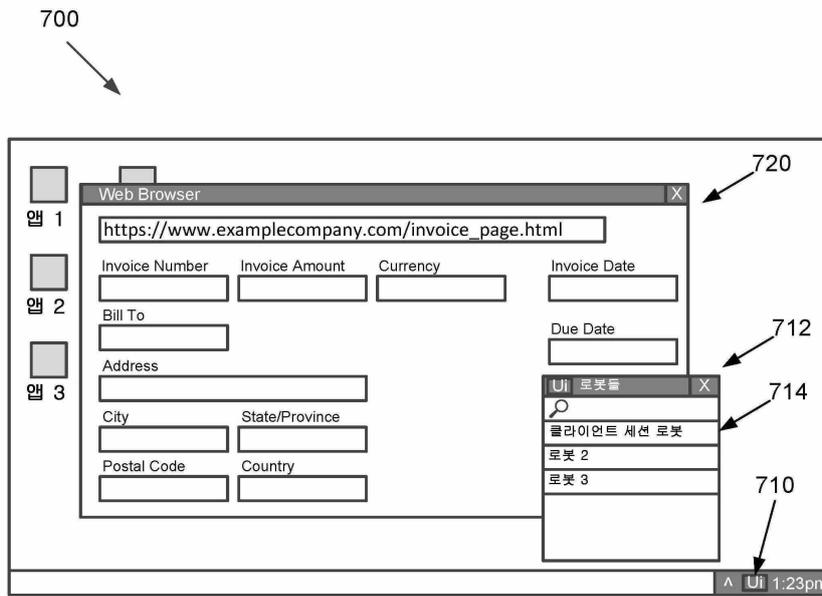
도면7a



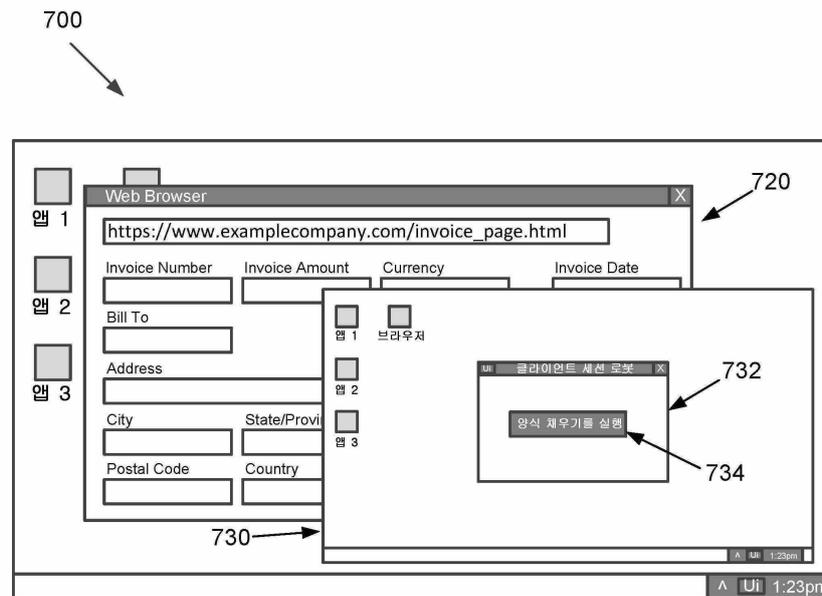
도면7b



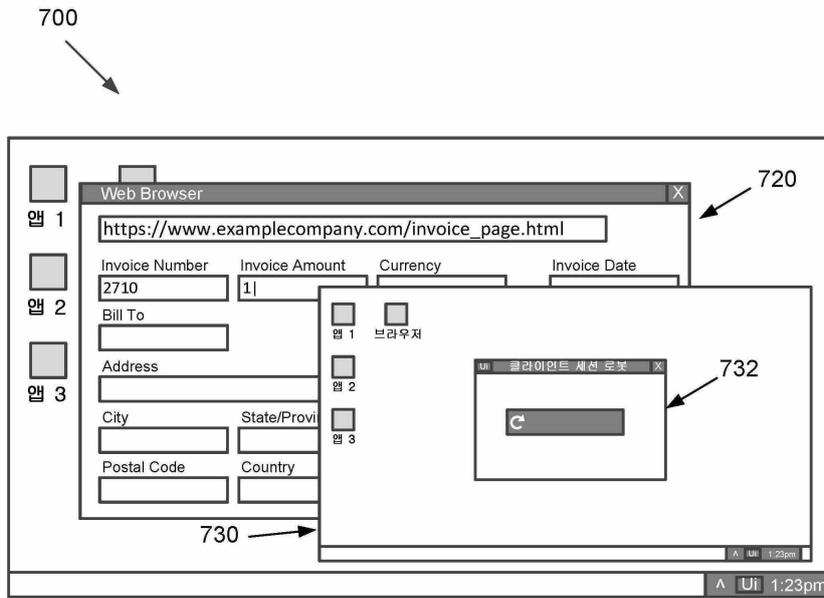
도면7c



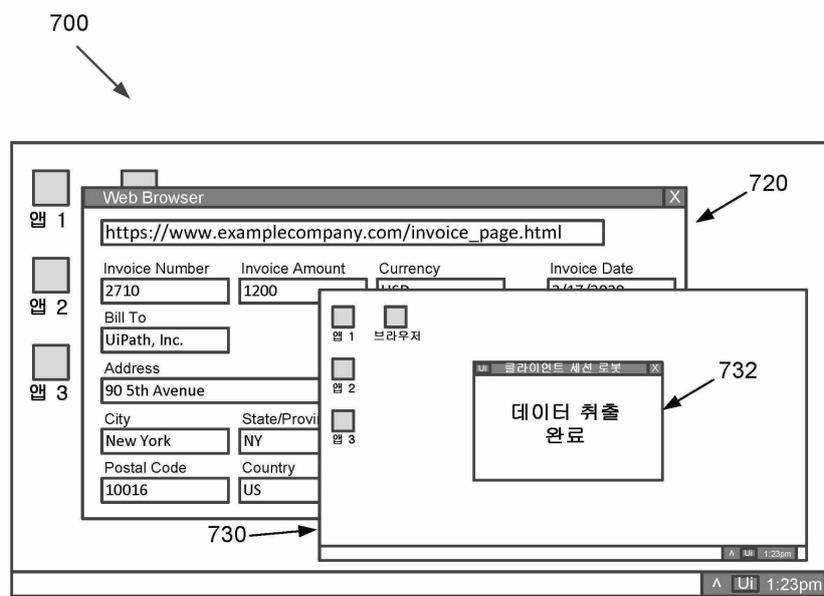
도면7d



도면7e

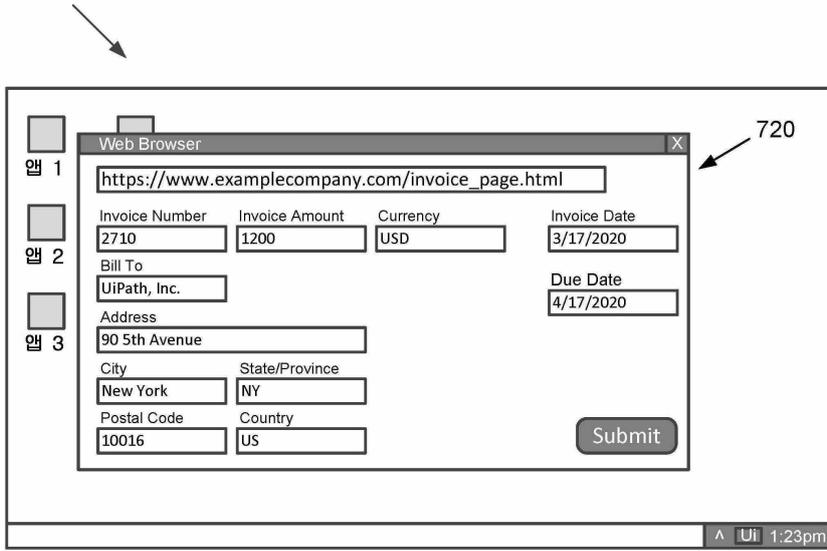


도면7f



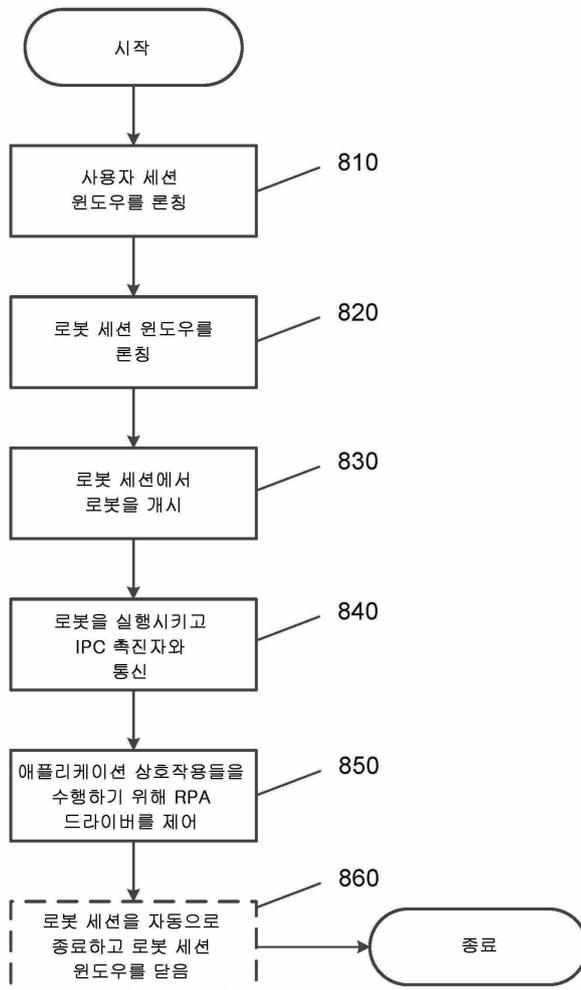
도면7g

700



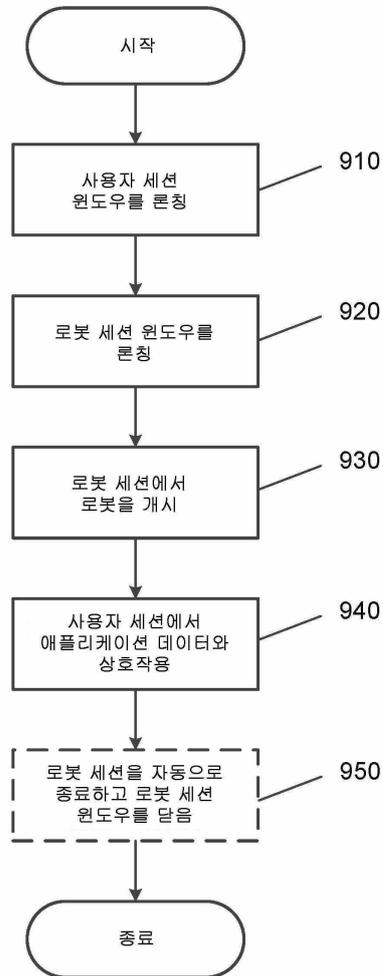
도면8

800

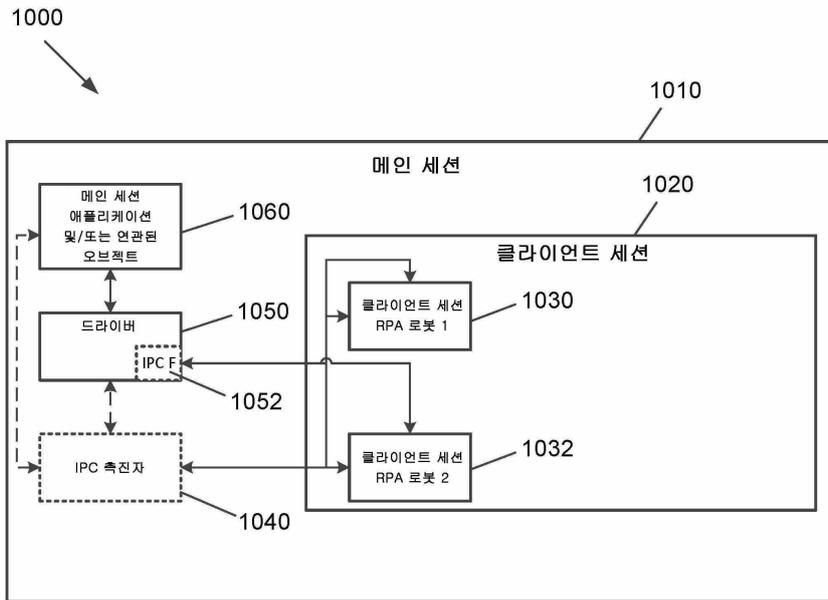


도면9

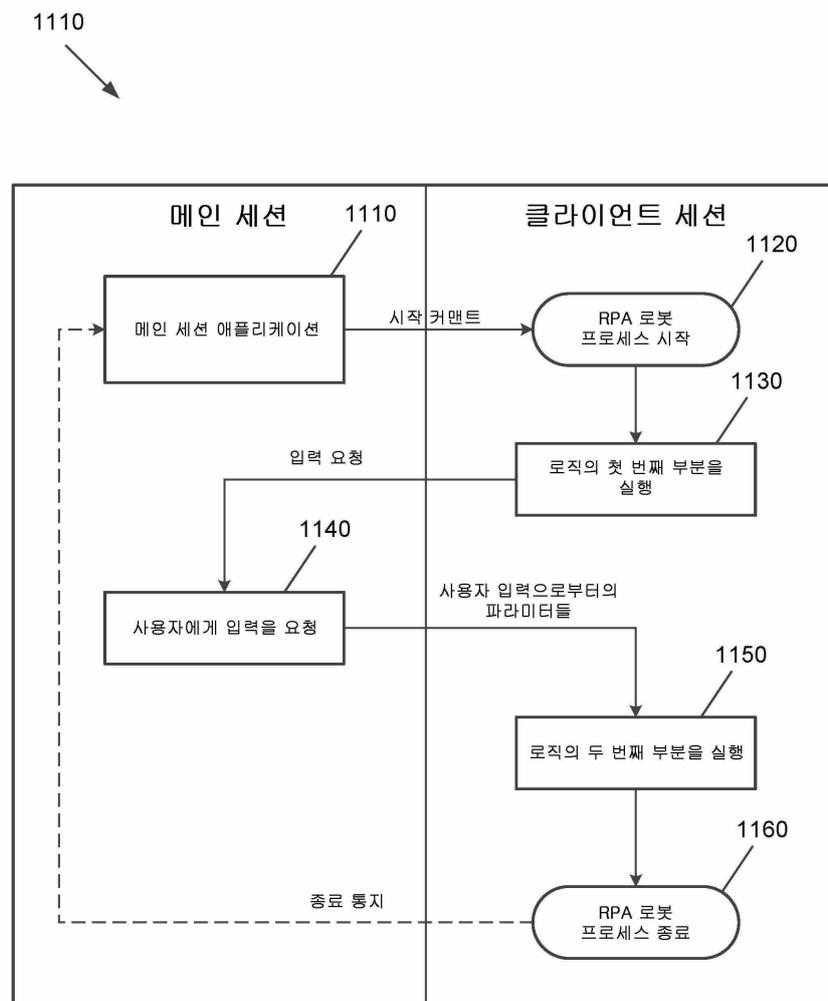
900



도면10

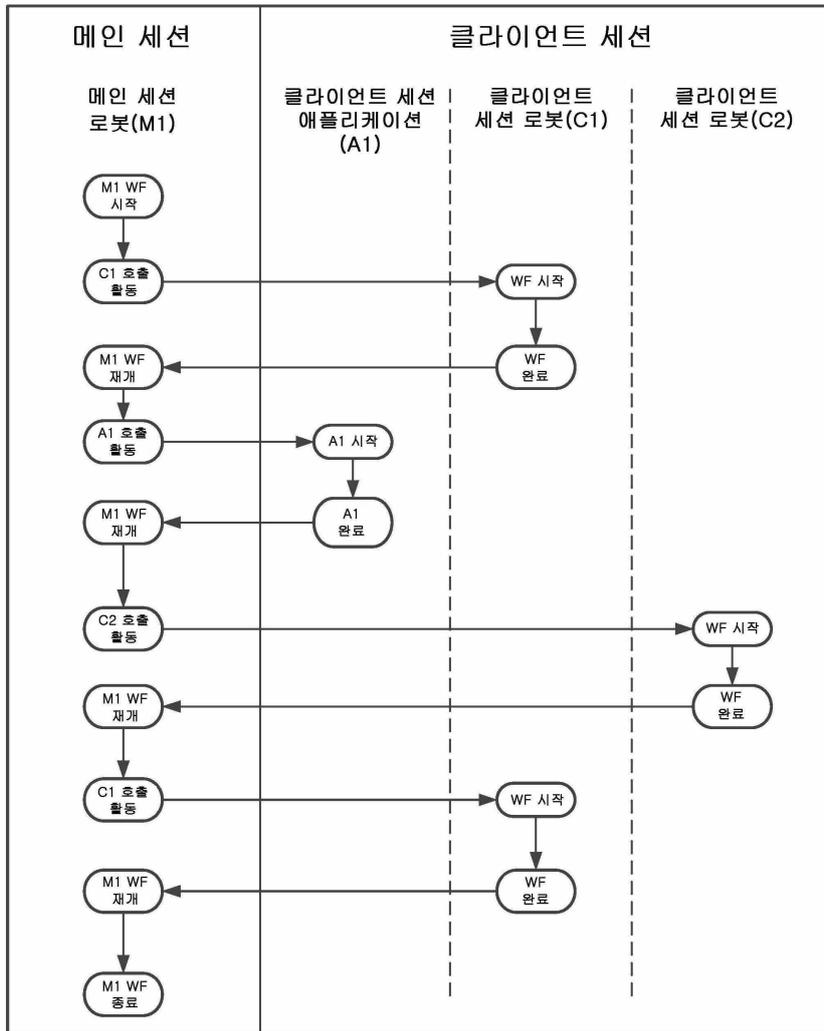


도면11



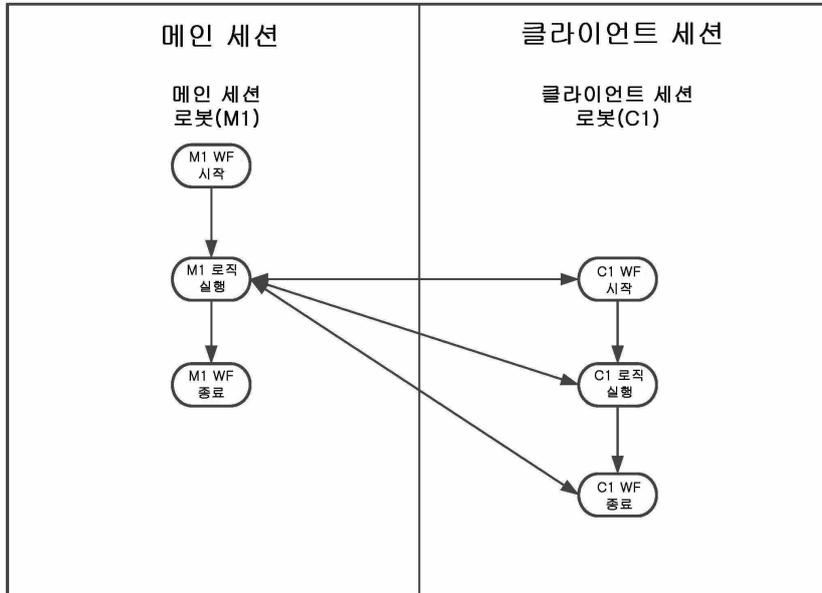
도면12

1200



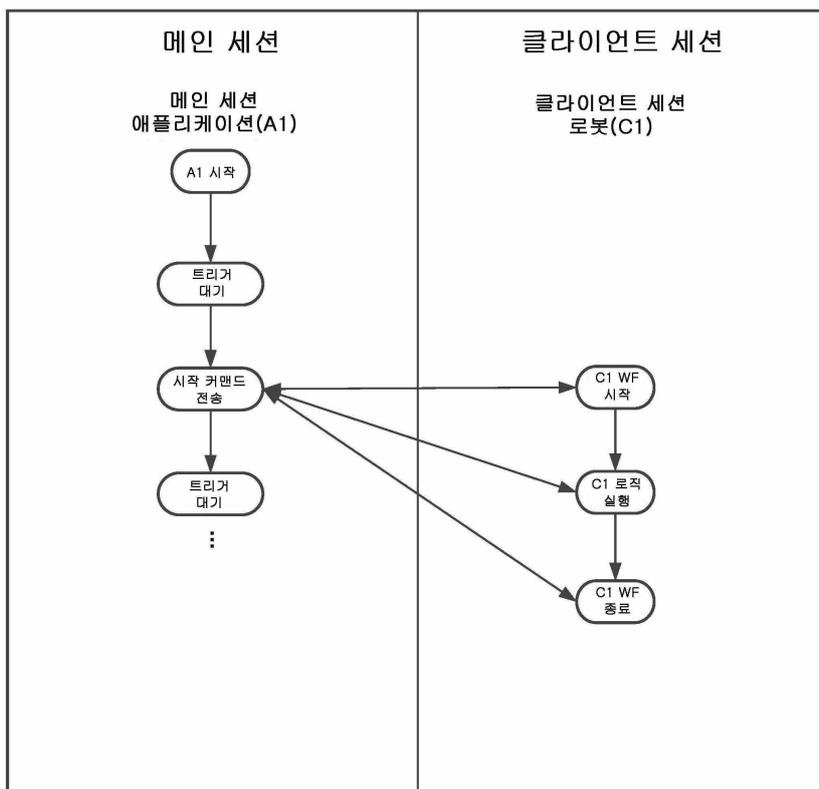
도면13

1300



도면14

1400



도면15

