



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년09월25일
(11) 등록번호 10-2160379
(24) 등록일자 2020년09월21일

(51) 국제특허분류(Int. Cl.)
G06F 11/36 (2006.01)
(52) CPC특허분류
G06F 11/3636 (2013.01)
G06F 11/3688 (2013.01)
(21) 출원번호 10-2020-0047393
(22) 출원일자 2020년04월20일
심사청구일자 2020년04월20일
(56) 선행기술조사문헌
JP2018185808 A*
(뒷면에 계속)

(73) 특허권자
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
최병주
서울특별시 강남구 압구정로 201, 74-1305 (압구정동, 현대아파트)
추현지
경기도 안산시 단원구 광덕동로 78, 607동 1202 (고잔동, 네오빌주공아파트)
(74) 대리인
특허법인(유한)아이시스

전체 청구항 수 : 총 7 항

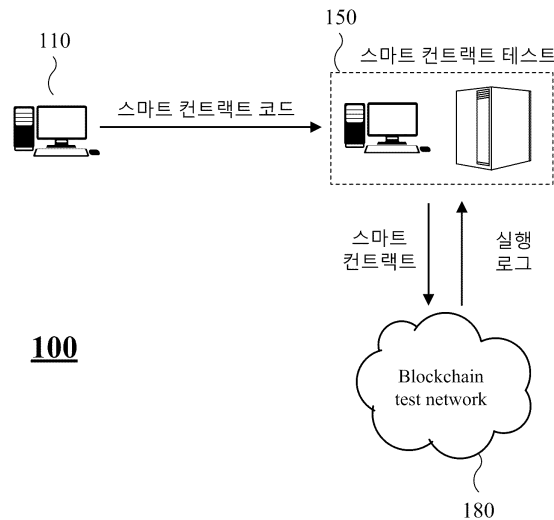
심사관 : 홍경아

(54) 발명의 명칭 **블록체인 기반 분산 애플리케이션 테스트 방법 및 테스트 장치**

(57) 요약

블록체인 기반 분산 애플리케이션 테스트 방법은 테스트 장치가 스마트 컨트랙트(smart contract)의 코드를 입력 받는 단계, 상기 테스트 장치가 상기 코드에 포함된 함수 각각에 대하여 상기 함수의 시작 지점과 종료 지점에 상기 함수에서 사용하는 변수 항목의 값을 출력하는 로그 생성 코드를 삽입하는 단계, 상기 테스트 장치가 상기 스마트 컨트랙트의 실행 과정에서 상기 로그 생성 코드가 출력하는 로그(log)를 획득하는 단계, 상기 테스트 장치가 상기 로그를 이용하여 테스트 데이터를 결정하는 단계 및 상기 테스트 장치가 상기 테스트 데이터를 기준으로 설계 모델의 결과와 상기 스마트 컨트랙트의 실행 결과를 비교하여 상기 스마트 컨트랙트를 테스트하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류
G06F 11/3696 (2013.01)

(56) 선행기술조사문헌
KR1020190139940 A*
WO2019075234 A1
KR1020200105997 A
KR1020190115066 A
*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호	1711103334
과제번호	2017-0-01628-004
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	적응형 블록체인 플랫폼 기술 개발 및 전문 인력 양성
기 여 율	1/1
과제수행기관명	서강대학교산학협력단
연구기간	2020.01.01 ~ 2020.12.31

명세서

청구범위

청구항 1

테스트 장치가 스마트 컨트랙트(smart contract)의 코드를 입력받는 단계;

상기 테스트 장치가 상기 코드에 포함된 함수 각각에 대하여 상기 함수의 시작 지점과 종료 지점에 상기 함수에서 사용하는 변수 항목의 값을 출력하는 로그 생성 코드를 삽입하는 단계;

상기 테스트 장치가 상기 스마트 컨트랙트의 실행 과정에서 상기 로그 생성 코드가 출력하는 로그(log)를 획득하는 단계;

상기 테스트 장치가 상기 로그를 이용하여 테스트 데이터를 결정하는 단계; 및

상기 테스트 장치가 상기 테스트 데이터를 기준으로 설계 모델의 결과와 상기 스마트 컨트랙트의 실행 결과를 비교하여 상기 스마트 컨트랙트를 테스트하는 단계를 포함하되,

상기 테스트 데이터는 트랜잭션 전 데이터 및 트랜잭션 후 데이터로 구분되고,

상기 테스트 데이터를 결정하는 단계에서, 상기 테스트 장치는 (i) 상기 변수 항목 중 변경되지 않는 항목은 상기 로그에서 처음 나타나는 값을 각각 상기 트랜잭션 전 데이터 및 상기 트랜잭션 후 데이터로 결정하고, (ii) 상기 변수 항목 중 변경되는 항목은 상기 로그에서 처음 나타나는 값을 상기 트랜잭션 전 데이터로 결정하고, 상기 로그에서 가장 마지막에 나타나는 값을 상기 트랜잭션 후 데이터로 결정하고,

상기 테스트하는 단계에서, 상기 테스트 장치는 상기 트랜잭션 전 데이터를 상기 설계 모델에 입력하여 출력되는 시뮬레이션 결과와 상기 트랜잭션 후 데이터를 비교하여 상기 스마트 컨트랙트를 테스트하는 블록체인 기반 분산 애플리케이션 테스트 방법.

청구항 2

제1항에 있어서,

상기 변수 항목은 스마트 컨트랙트 상태 변수 값, 트랜잭션 송신 계정 이더량, 컨트랙트 계정 이더량, 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정 및 컨트랙트 계정을 포함하는 블록체인 기반 분산 애플리케이션 테스트 방법.

청구항 3

삭제

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 변경되지 않는 항목은 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정 및 컨트랙트 계정을 포함하고,

상기 변경되는 항목은 트랜잭션 송신 계정 이더량, 컨트랙트 이더량 및 스마트 컨트랙트 상태 변수 값을 포함하는 블록체인 기반 분산 애플리케이션 테스트 방법.

청구항 6

삭제

청구항 7

스마트 컨트랙트(smart contract)의 코드를 입력받는 입력장치;

로그 생성 프로그램, 로그 분석 프로그램 및 스마트 컨트랙트의 시뮬레이션 프로그램을 저장하는 저장장치;

스마트 컨트랙트를 블록체인 네트워크에 전송하고, 상기 블록체인 네트워크로부터 스마트 컨트랙트의 로그(log)를 수신하는 통신장치; 및

상기 로그 생성 프로그램을 이용하여 상기 코드에 포함된 함수 각각에 대하여 상기 함수의 시작 지점과 종료 지점에 상기 함수에서 사용하는 변수 항목의 값을 출력하는 로그 생성 코드를 삽입하고, 상기 로그 분석 프로그램을 이용하여 상기 스마트 컨트랙트의 실행 과정에서 상기 로그 생성 코드가 출력하는 로그를 이용하여 테스트 데이터를 결정하고, 상기 테스트 데이터를 기준으로 시뮬레이션 프로그램의 결과와 상기 블록체인 네트워크에서의 상기 스마트 컨트랙트의 실행 결과를 비교하여 상기 스마트 컨트랙트를 테스트하는 연산장치를 포함하되,

상기 테스트 데이터는 트랜잭션 전 데이터 및 트랜잭션 후 데이터로 구분되고,

상기 연산 장치는 상기 변수 항목 중 변경되지 않는 항목은 상기 로그에서 처음 나타나는 값을 각각 상기 트랜잭션 전 데이터 및 상기 트랜잭션 후 데이터로 결정하고,

상기 변수 항목 중 변경되는 항목은 상기 로그에서 처음 나타나는 값을 상기 트랜잭션 전 데이터로 결정하고, 상기 로그에서 가장 마지막에 나타나는 값을 상기 트랜잭션 후 데이터로 결정하고

상기 연산 장치는 상기 트랜잭션 전 데이터를 상기 시뮬레이션 프로그램에 입력하여 출력되는 시뮬레이션 결과와 상기 트랜잭션 후 데이터를 비교하여 상기 스마트 컨트랙트를 테스트하는 블록체인 기반 분산 애플리케이션 테스트 장치.

청구항 8

제7항에 있어서,

상기 변수 항목은 스마트 컨트랙트 상태 변수 값, 트랜잭션 송신 계정 이더량, 컨트랙트 계정 이더량, 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정 및 컨트랙트 계정을 포함하는 블록체인 기반 분산 애플리케이션 테스트 장치.

청구항 9

삭제

청구항 10

삭제

청구항 11

제7항에 있어서,

상기 변경되지 않는 항목은 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정 및 컨트랙트 계정을 포함하고,

상기 변경되는 항목은 트랜잭션 송신 계정 이더량, 컨트랙트 이더량 및 스마트 컨트랙트 상태 변수 값을 포함하는 블록체인 기반 분산 애플리케이션 테스트 장치.

청구항 12

삭제

청구항 13

컴퓨터에서 제1항, 제2항 및 제5항 중 어느 하나의 항에 기재된 블록체인 기반 분산 애플리케이션 테스트 방법을 실행하기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

발명의 설명

기술 분야

[0001] 이하 설명하는 기술은 블록체인 기반의 애플리케이션 테스트 방법에 관한 것이다.

배경 기술

[0002] 데이터의 무결성 및 신뢰성이 요구되는 분야에서 블록체인 기술이 다양하게 활용되고 있다. 암호 화폐, 전자 투표, 전자상거래 플랫폼 등과 같은 다양한 서비스들이 점차 블록체인 기반으로 개발되고 있다. 이더리움(ethereum)은 대표적인 블록체인 개발 및 배포 플랫폼이다. 이더리움은 솔리디티(solidity) 언어를 사용한 스마트 컨트랙트(smart contract)를 지원한다. 개발자는 스마트 컨트랙트를 통해 블록체인 기반의 분산 애플리케이션 DApp(Decentralized Application)을 개발할 수 있다.

[0003] 블록체인 스마트 컨트랙트는 주로 솔리디티 언어로 작성되고 있으며, 스마트 컨트랙트가 점차 다양한 목적으로 개발되면서 개발 언어의 보안상 취약점 또한 지적되고 있다. 최근에는 스마트 컨트랙트 기반 애플리케이션을 테스트하기 위한 다양한 분석 도구도 연구되고 있다.

선행기술문헌

비특허문헌

[0004] (비특허문헌 0001) 문현아 외 2인, 솔리디티 스마트 컨트랙트의 스크립트 기반 시나리오 테스트, 2019년 제 21회 한국 소프트웨어공학 학술대회(KCSE 2019)

발명의 내용

해결하려는 과제

[0005] 종래 정적 테스트 도구는 스마트 컨트랙트가 동작 중에 설계에 부합하게 작동하는지를 확인하기 쉽지 않다. 종래 동적 테스트 도구는 블록체인 트랜잭션(transaction)에 관여된 다수의 계정들의 트랜잭션 전후 상태를 모두 고려한 스크립트 작성이 매우 복잡하다.

[0006] 이하 설명하는 기술은 스마트 컨트랙트가 동작하면서 생성되는 데이터를 분석하여 자동으로 테스트 항목을 결정하는 동적 테스트 도구를 제공하고자 한다.

과제의 해결 수단

[0007] 블록체인 기반 분산 애플리케이션 테스트 방법은 테스트 장치가 스마트 컨트랙트(smart contract)의 코드를 입력받는 단계, 상기 테스트 장치가 상기 코드에 포함된 함수 각각에 대하여 상기 함수의 시작 지점과 종료 지점에 상기 함수에서 사용하는 변수 항목의 값을 출력하는 로그 생성 코드를 삽입하는 단계, 상기 테스트 장치가 상기 스마트 컨트랙트의 실행 과정에서 상기 로그 생성 코드가 출력하는 로그(log)를 획득하는 단계, 상기 테스트 장치가 상기 로그를 이용하여 테스트 데이터를 결정하는 단계 및 상기 테스트 장치가 상기 테스트 데이터를 기준으로 설계 모델의 결과와 상기 스마트 컨트랙트의 실행 결과를 비교하여 상기 스마트 컨트랙트를 테스트하는 단계를 포함한다.

[0008] 블록체인 기반 분산 애플리케이션 테스트 장치는 스마트 컨트랙트(smart contract)의 코드를 입력받는 입력장치, 로그 생성 프로그램, 로그 분석 프로그램 및 스마트 컨트랙트의 시뮬레이션 프로그램을 저장하는 저장장치, 스마트 컨트랙트를 블록체인 네트워크에 전송하고, 상기 블록체인 네트워크로부터 스마트 컨트랙트의 로그(log)를 수신하는 통신장치 및 상기 로그 생성 프로그램을 이용하여 상기 코드에 포함된 함수 각각에 대하여 상기 함수의 시작 지점과 종료 지점에 상기 함수에서 사용하는 변수 항목의 값을 출력하는 로그 생성 코드를 삽입하고, 상기 로그 분석 프로그램을 이용하여 상기 스마트 컨트랙트의 실행 과정에서 상기 로그 생성 코드가 출력하는 로그를 이용하여 테스트 데이터를 결정하고, 상기 테스트 데이터를 기준으로 시뮬레이션 프로그램의 결과와 상기 블록체인 네트워크에서의 상기 스마트 컨트랙트의 실행 결과를 비교하여 상기 스마트 컨트랙트를 테스트하는 연산장치를 포함한다.

발명의 효과

[0009] 이하 설명하는 기술은 다수의 계정 사이의 복잡한 호출 관계로 얽혀 있는 트랜잭션에 대해서도 손쉽게 스마트 컨트랙트 기반 애플리케이션을 테스트할 수 있다.

도면의 간단한 설명

[0010] 도 1은 블록체인 기반 분산 애플리케이션의 테스트 시스템에 대한 예이다.

도 2는 분산 애플리케이션의 테스트 장치 동작에 대한 예이다.

도 3은 테스트 데이터 구조에 대한 예이다.

도 4는 테스트 데이터 결정 과정에 대한 순서도의 예이다.

도 5는 테스트 장치 동작의 동작 결과에 대한 예이다.

도 6은 분산 애플리케이션의 테스트 장치의 구조에 대한 예이다.

발명을 실시하기 위한 구체적인 내용

[0011] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시례를 가질 수 있는 바, 특정 실시례들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0012] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0013] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설명된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0014] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0015] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0017] 이하 설명에서 사용하는 용어에 대하여 먼저 정의한다.

[0018] 블록체인은 데이터를 거래할 때 중앙집중형 서버에 기록을 보관하는 기존 방식과 달리 거래 참가자 모두에게 내용을 공개하는 분산형 디지털 장부를 말한다. 현재 까지 다양한 블록체인 플랫폼이 개발되었다. 이하 블록체인은 특정한 프로토콜이나 개발 언어를 가리지 않고, 다양한 플랫폼을 포함하는 의미로 사용한다.

[0019] 블록체인 기반 분산 애플리케이션 또는 간단하게 분산 애플리케이션은 블록체인 구조를 활용한 분산 애플리케이션 DApp을 의미한다. 분산 애플리케이션은 애플리케이션의 목적이나 종류를 제한하지 않고, 테스트 대상이 되는 다양한 애플리케이션을 포함할 수 있다.

[0020] 분산 애플리케이션 테스트 도구 또는 간단하게 테스트 도구는 블록체인 기반 분산 애플리케이션에 대한 테스트 애플리케이션을 의미한다. 테스트 도구는 프로그램으로 구현된다.

- [0021] 분산 애플리케이션 테스트 장치 또는 테스트 장치는 테스트 도구를 이용하여 개발자가 개발한 분산 애플리케이션을 테스트하는 장치이다. 테스트 장치는 물리적으로 다양한 형태로 구현될 수 있다. 예컨대, 테스트 장치는 네트워크 서버, PC, 스마트 기기 등과 같은 형태를 가질 수 있다.
- [0022] 이하 사용자 또는 개발자는 개발한 분산 애플리케이션을 테스트하는 주체를 의미한다.
- [0024] 도 1은 블록체인 기반 분산 애플리케이션의 테스트 시스템(100)에 대한 예이다. 도 1은 개략적인 분산 애플리케이션의 테스트 시스템이다.
- [0025] 클라이언트 장치(110)는 개발자가 분산 애플리케이션 개발에 사용하는 장치이다. 개발자는 클라이언트 장치(110)를 이용하여 분산 애플리케이션을 코딩하고 테스트 대상이 될 분산 애플리케이션 코드를 완성한다. 클라이언트 장치(110)는 스마트 컨트랙트 코드를 테스트 장치(150)에 전달한다.
- [0026] 테스트 장치(150)는 PC, 서버, 스마트 기기 등과 같은 형태를 가질 수 있다. 도 1은 클라이언트 장치(110)와 테스트 장치(150)를 별개의 객체로 도시하였다. 테스트 도구는 프로그램 형태로 개발된다. 따라서, 개발자는 분산 애플리케이션 코드를 개발하는 도구와 테스트 도구를 하나의 컴퓨터 장치에 설치하여 코드 개발 및 테스트를 수행할 수도 있다. 즉, 도 1의 클라이언트 장치(110)와 테스트 장치(150)가 물리적으로 동일한 객체일 수도 있다.
- [0027] 테스트 장치(150)는 스마트 컨트랙트를 일정하게 분석하고 처리하여 실행 로그(log)를 생성하도록 코드를 변경한다. 테스트 장치(150)는 변경된 스마트 컨트랙트를 블록체인 네트워크(180)에 배포한다. 테스트 장치(150)는 블록체인 네트워크(180)에서 실행되는 스마트 컨트랙트에서 출력되는 실행 로그를 수신한다. 테스트 장치(150)는 실행 로그를 분석하여 테스트 데이터를 결정하고, 시뮬레이션 결과와 비교하여 해당 스마트 컨트랙트를 테스트한다.
- [0029] 도 2는 분산 애플리케이션의 테스트 장치(200)의 동작에 대한 예이다. 도 2에서 테스트 장치(200)는 기능적 구성으로 구분한 예이다. 테스트 장치(200)는 로그 생성 모듈(210), 로그 분석 모듈(220) 및 테스트 검증 모듈(230)을 포함한다.
- [0030] DApp 개발자(이하 개발자라 함)는 분산 애플리케이션에 대한 스마트 컨트랙트를 완성한다. ① 테스트 장치는 개발자는 스마트 컨트랙트 코드를 입력받는다.
- [0031] 로그 생성 모듈(210)은 트랜잭션 실행 중 변수 값이 변화하는 내역을 실행 로그로 생성하기 위해 입력된 스마트 컨트랙트 코드를 분석하여 상태 변수를 식별한다. 상태 변수는 스마트 컨트랙트 내 최상위단에 선언된 변수를 의미한다. 상태 변수는 EVM(Ethereum Virtual Machine) 내 메모리가 아닌 스토리지 영역에 영구 저장되고, 각 트랜잭션 실행 이후의 스마트 컨트랙트의 최종 상태를 나타내는 지표가 된다. 트랜잭션에 의한 스마트 컨트랙트 상태 전이는 상태변수의 변화에 의해 일어나다고 볼 수 있고, 이는 각 함수가 호출될 때마다 발생한다.
- [0032] 로그 생성 모듈(210)은 입력받은 스마트 컨트랙트 코드의 각 함수마다 시작 지점과 종료 지점에 변수 값을 실행 로그로 출력하는 코드를 삽입한다. 로그를 출력하는 코드를 로그 생성 코드라고 명명한다. ② 로그 생성 모듈(210)은 로그 생성 코드가 추가된 스마트 컨트랙트를 블록체인 테스트 네트워크에 배포한다.
- [0033] 한편, ③개발자는 테스트를 위하여 사전에 가상의 DApp 사용자 계정을 생성한다. ④ 로그 생성 코드가 추가된 스마트 컨트랙트가 블록체인 네트워크에 배포된 후 개발자는 가상의 DApp 사용자 계정으로 트랜잭션을 요청한다.
- [0034] ⑤ 테스트 장치(200)는 요청 트랜잭션에 대하여 생성되는 실행 로그를 수신한다. 로그 생성 코드는 트랜잭션마다 함수의 시작 시점에서의 각 상태 변수 값과 함수 종료 시점에서의 상태 변수 값을 트랜잭션 송신 계정 이더리움, 컨트랙트 계정 이더리움과 함께 실행 로그로 출력되게 한다. 한편, 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정, 컨트랙트 계정은 이더리움이 로그 내에 기본적으로 출력하는 항목이므로 별도로 출력 코드를 사용할 필요가 없다.
- [0035] 로그 분석 모듈(220)은 수신한 실행 로그를 분석한다. 로그 분석 모듈(220)은 실행 로그를 분석하여 테스트 데이터를 분류 및 결정한다.
- [0036] 로그 분석 모듈(220)은 블록체인 테스트 네트워크로부터 출력된 실행 로그로부터 테스트 항목을 정의하고 테스트 데이터로 생성한다. 테스트 데이터는 트랜잭션 전 데이터와 트랜잭션 후 데이터로 구분된다. 트랜잭션 전 데이터는 트랜잭션이 전송되기 전 각 계정의 상태를 나타낸다. 트랜잭션 후 데이터는 스마트 컨트랙트 코드 실행에 의해 각 계정의 상태가 변화한 결과값을 나타낸다. 도 3은 테스트 데이터 구조에 대한 예이다. 트랜잭션 기

본 정보와 트랜잭션 송신 계정 정보는 가상의 DApp 사용자 계정 정보 및 트랜잭션 요청 정보에 대응한다. 컨트랙트 상태 정보는 테스트 항목에 대응한다. 로그 분석 모듈(220)은 추출된 실행 로그에 기반하여 테스트 항목을 자동으로 정의하게 된다.

- [0037] ⑥ 테스트 검증 모듈(230)은 로그 분석 모듈(220)이 결정한 테스트 데이터를 수신한다. 테스트 검증 모듈(230)은 트랜잭션 전 데이터를 블록체인 설계 모델 내지 시뮬레이션 모델(시뮬레이션 프로그램)에 입력하여 출력 결과를 얻는다. 그리고, 테스트 검증 모듈(230)은 시뮬레이션 모델에서 출력되는 값과 트랜잭션 후 데이터를 비교하여 결과가 동일한지 확인한다.
- [0038] ⑦ 테스트 장치(200)는 테스트 결과를 사용자에게 전달할 수 있다.
- [0040] 도 4는 테스트 데이터 결정 과정(300)에 대한 순서도의 예이다. 도 4의 설명과 함께 연구자가 실제 테스트한 결과를 같이 설명한다. 도 4는 도 3의 테스트 검증 모듈(230)이 동작하는 과정에 해당한다.
- [0041] 테스트 장치는 블록체인 네트워크로부터 전송한 실행 로그를 수신한다. 테스트 장치는 실행 로그를 파싱하고 관련된 변수를 추출한다(310).
- [0042] 변수 항목은 스마트 컨트랙트 상태 변수 값, 트랜잭션 송신 계정 이더량, 컨트랙트 계정 이더량, 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정 및 컨트랙트 계정을 포함할 수 있다.
- [0043] 테스트 장치는 추출한 변수 항목이 변경되는 변수인지 여부를 확인한다(320). 이하, 현재 테스트 데이터를 결정하는 대상이 되는 변수를 현재 변수라고 명명한다.
- [0044] 현재 변수가 변경되지 않는 변수 항목이라면(320의 YES), 테스트 장치 해당 변수를 트랜잭션 전 데이터 및 트랜잭션 후 데이터에 추가한다(330). 즉, 변경되지 않는 변수 세트(제1 변수 세트)는 트랜잭션 전 데이터와 동시에 트랜잭션 후 데이터로 분류된다. 변경되지 않는 변수 항목은 트랜잭션 해시, 트랜잭션 입력 값, 트랜잭션 송신 계정 및 컨트랙트 계정을 포함한다. 따라서, 테스트 장치는 현재 변수가 제1 변수 세트에 속하면, 해당 변수를 트랜잭션 전 데이터 및 트랜잭션 후 데이터에 추가한다.
- [0045] 한편, 변경되는 변수 세트(제2 변수 세트)는 트랜잭션 송신 계정 이더량, 컨트랙트 이더량 및 스마트 컨트랙트 상태 변수 값을 포함한다. 현재 변수가 제2 변수 세트에 속하면(320의 NO), 테스트 장치는 이후 실행 로그에 해당 변수가 등장하는 시기에 따라 테스트 데이터를 결정한다.
- [0046] 테스트 장치는 현재 변수가 실행 로그에 최초로 등장하는 변수인지 판단한다(340). 현재 변수가 실행 로그에 최초로 등장하는 변수라면(340의 YES), 테스트 장치는 현재 변수를 트랜잭션 전 데이터에 추가한다(350).
- [0047] 현재 변수가 실행 로그에 최초로 등장하는 변수가 아니라면(340의 NO), 테스트 장치는 현재 변수가 실행 로그에서 마지막(최종)으로 출력되는 변수인지 판단한다(360).
- [0048] 현재 변수가 실행 로그의 최종 변수라면(360의 YES), 테스트 장치는 현재 변수를 트랜잭션 후 데이터에 추가한다(370). 현재 변수가 실행 로그의 최종 변수가 아니라면(360의 NO), 테스트 장치는 계속 실행 로그를 분석하여 테스트 데이터를 결정하는 과정을 반복한다.
- [0050] 도 5는 테스트 장치 동작의 동작 결과에 대한 예이다. 도 5는 연구자가 연구실에서 개발한 테스트 도구를 테스트한 결과이다.
- [0051] 도 5(A)는 테스트 장치가 개발자가 완성한 스마트 컨트랙트에 로그 생성 코드를 자동으로 추가하는 예이다. 도 5(A)에서 좌측은 원본 스마트 컨트랙트이고, 우측은 일부 로그 생성 코드가 추가된 스마트 컨트랙트를 도시한다. 테스트 장치가 스마트 컨트랙트 내의 함수(deposit())의 시작 지점과 종료 지점에서 각각 변수의 값을 나타내는 로그를 출력하는 코드(로그 생성 코드)를 추가했음을 알 수 있다.
- [0052] 도 5(B)는 테스트 장치가 실행 로그에서 변수를 식별하고 분석하는 과정을 나타낸다. 도 5(B)에서 붉은색 텍스트는 해당 변수의 정보를 별도로 표시한 것이다. 테스트 장치는 전송한 바와 같이 실행 로그에서 등장하는 변수를 불변 변수와 가변 변수로 구분한다. 테스트 장치는 불변 변수는 트랜잭션 전 데이터 및 트랜잭션 후 데이터로 분류한다. 테스트 데이터는 최종적으로 테스트에 사용되므로, 테스트 장치는 트랜잭션 전 데이터와 트랜잭션 후 데이터를 구분한다. 테스트 장치는 가변 변수에 대해서는 실행 로그에 등장하는 시점(최초, 최종)에 따라 트랜잭션 전 데이터 또는 트랜잭션 후 데이터로 분류한다.
- [0053] 도 5(C)는 테스트 데이터를 트랜잭션 전 데이터와 트랜잭션 후 데이터를 구분한 예이다. 테스트 데이터는 트랜

잭션 전 데이터와 트랜잭션 후 데이터로 구성된다. 도 5(C)에서 상부 테이블은 트랜잭션 전 데이터를 나타내고, 하부 테이블은 트랜잭션 후 데이터를 나타낸다.

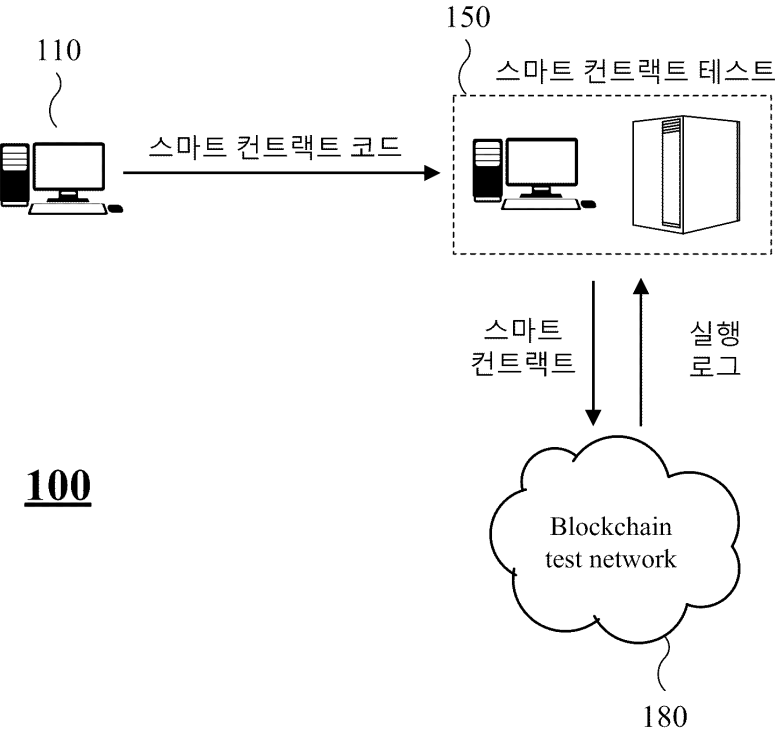
- [0054] 불변 변수인 트랜잭션 기본 정보(트랜잭션 해시 및 트랜잭션 입력 값), 트랜잭션 송신 계정 및 컨트랙트 계정은 트랜잭션 전 데이터와 트랜잭션 후 데이터에 모두 포함된다. 가변 변수인 트랜잭션 송신 계정 이더량, 컨트랙트 이더량 및 스마트 컨트랙트 상태 변수 값은 실행 로그에 등장하는 시점에 따라 트랜잭션 전 데이터 또는 트랜잭션 후 데이터로 구분된다.
- [0056] 도 6은 분산 애플리케이션의 테스트 장치(400)의 구조에 대한 예이다. 도 6은 테스트 장치(400)는 도 1의 테스트 장치(150)에 해당한다. 테스트 장치(400)는 물리적으로 다양한 형태로 구현될 수 있다. 예컨대, 테스트 장치(400)는 PC, 스마트기기, 컴퓨터 장치, 네트워크의 서버, 데이터 처리 전용 칩셋 등의 형태를 가질 수 있다.
- [0057] 테스트 장치(400)는 저장장치(410), 메모리(420), 연산장치(430), 인터페이스 장치(440), 통신장치(450) 및 출력장치(460)를 포함할 수 있다.
- [0058] 저장장치(410)는 분산 애플리케이션 테스트를 위한 프로그램(들)을 저장한다. 프로그램은 테스트 과정의 구성에 따라 개별 모듈로 구성될 수도 있다. 예컨대, 테스트 프로그램은 로그 생성 프로그램, 로그 분석 프로그램 및 테스트 결과 프로그램을 저장할 수 있다. 로그 생성 프로그램은 개발자가 제공하는 스마트 컨트랙트의 함수에 변수 값을 출력하는 로그 생성 코드를 자동으로 삽입하는 프로그램이다. 로그 분석 프로그램은 블록체인 테스트 네트워크에서 전달되는 실행 로그를 분석하는 프로그램이다. 로그 분석 프로그램은 실행 로그에 존재하는 변수들을 식별하여 테스트 데이터를 구성한다. 테스트 결과 프로그램은 트랜잭션 전 데이터를 설계 모델(예컨대, Simulink)에 입력하여 출력 결과를 얻는다. 이를 위해 저장장치(410)는 설계 모델(또는 시뮬레이션 프로그램)을 더 저장할 수도 있다. 테스트 결과 프로그램은 설계 모델의 출력 값과 트랜잭션 후 데이터를 비교하여 스마트 컨트랙트를 테스트할 수 있다.
- [0059] 저장장치(410)는 스마트 컨트랙트를 저장할 수 있다. 저장장치(410)는 분류한 테스트 데이터를 저장할 수 있다. 또한 저장장치(410)는 테스트 결과를 저장할 수도 있다.
- [0060] 메모리(420)는 테스트 장치가 데이터 처리 및 분석 과정에 생성되는 임시 데이터를 저장할 수 있다.
- [0061] 인터페이스장치(440)는 외부로부터 일정한 명령 및 데이터를 입력받는 장치이다. 인터페이스장치(440)는 물리적으로 연결된 입력장치 또는 외부 저장장치로부터 스마트 컨트랙트를 입력받을 수 있다. 인터페이스장치(440)는 분산 애플리케이션 테스트를 위한 프로그램을 입력받을 수 있다.
- [0062] 통신장치(450)는 유선 또는 무선 네트워크를 통해 일정한 정보를 수신하고 전송하는 구성을 의미한다. 통신장치(450)는 로그 생성 코드가 삽입된 스마트 컨트랙트를 블록체인 테스트 네트워크에 배포할 수 있다. 통신장치(450)는 블록체인 테스트 네트워크로부터 실행 로그를 수신할 수 있다.
- [0063] 통신장치(450)는 외부 객체로부터 스마트 컨트랙트를 입력받을 수 있다. 통신장치(450)는 데이터 처리를 위한 프로그램 및 데이터를 수신할 수 있다. 통신장치(450)는 분산 애플리케이션 테스트 결과를 외부 객체로 송신할 수도 있다.
- [0064] 통신장치(450) 내지 인터페이스장치(440)는 외부로부터 일정한 데이터 내지 명령을 전달받는 장치이다. 통신장치(450) 내지 인터페이스장치(440)를 입력장치라고 통칭할 수 있다.
- [0065] 출력장치(460)는 일정한 정보를 출력하는 장치이다. 출력장치(460)는 분산 애플리케이션 테스트 과정에 필요한 인터페이스, 테스트 결과 등을 출력할 수 있다.
- [0066] 연산 장치(430)는 스마트 컨트랙트 코드에 로그 생성 코드를 삽입할 수 있다. 연산 장치(430)는 로그 생성 프로그램으로 스마트 컨트랙트 코드를 분석하여, 각 함수마다 시작 시점과 종료 시점에 변수값을 출력하는 코드를 추가할 수 있다.
- [0067] 연산 장치(430)는 로그 생성 코드가 삽입된 스마트 컨트랙트를 블록체인 테스트 네트워크에 배포되게 제어할 수 있다.
- [0068] 연산 장치(430)는 블록체인 테스트 네트워크로부터 수신되는 실행 로그를 분석할 수 있다. 연산 장치(430)는 로그 분석 프로그램을 이용하여 스마트 컨트랙트의 실행 과정에서 로그 생성 코드가 출력하는 실행 로그를 기준으로 테스트 데이터를 결정(분류)할 수 있다. 전송한 바와 같이 연산 장치(430)는 불변 변수인 트랜잭션 기본 정보(트랜잭션 해시 및 트랜잭션 입력 값), 트랜잭션 송신 계정 및 컨트랙트 계정을 트랜잭션 전 데이터와 트랜잭

션 후 데이터에 포함되도록 분류한다. 또한, 연산 장치(430)는 가변 변수인 트랜잭션 송신 계정 이더량, 컨트랙트 이더량 및 스마트 컨트랙트 상태 변수 값을 실행 로그에 등장하는 시점에 따라 트랜잭션 전 데이터 또는 트랜잭션 후 데이터에 포함되도록 분류한다. 연산 장치(430)는 가변 변수가 실행 로그에 최초 등장하면, 해당 변수를 트랜잭션 전 데이터로 분류한다. 연산 장치(430)는 가변 변수가 실행 로그에 최종 등장하면, 해당 변수를 트랜잭션 후 데이터로 분류한다.

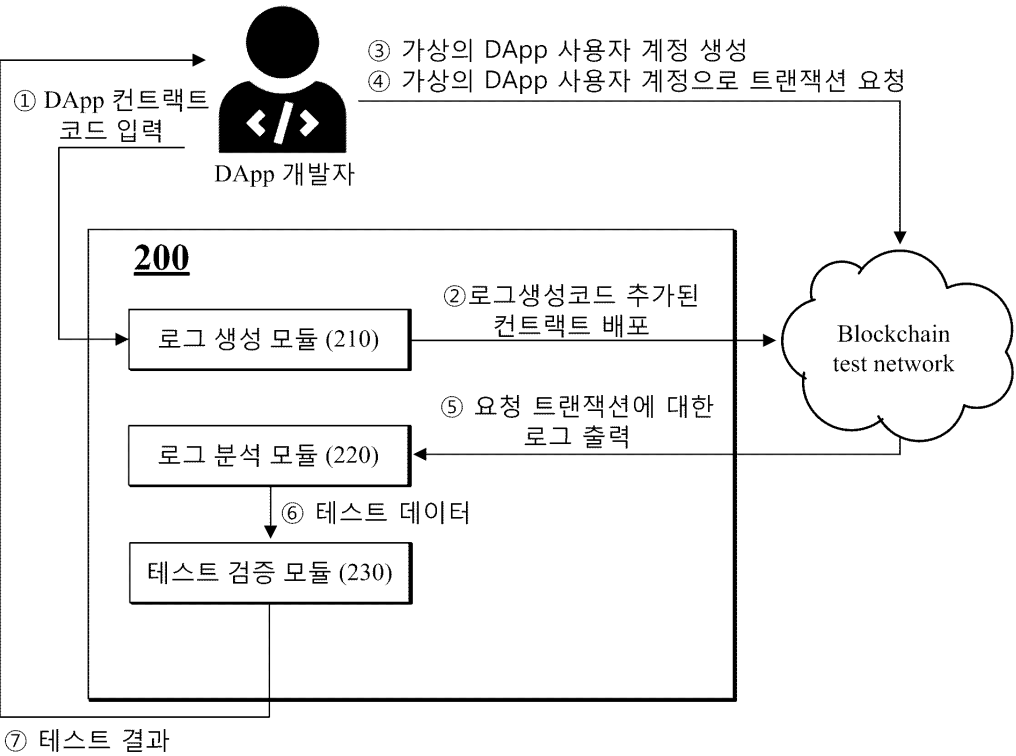
- [0069] 연산 장치(430)는 트랜잭션 전 데이터를 설계 모델에 입력하여, 설계 모델로부터 일정한 결과값을 획득한다. 연산 장치(430)는 설계 모델의 출력값과 트랜잭션 후 데이터의 변수값을 비교하여 두 개의 값이 동일한지 확인한다. 동일하다면, 연산 장치(430)는 테스트 대상인 분산 애플리케이션의 코드를 정확하게 동작했다고 판단할 수 있다. 동일하지 않다면, 연산 장치(430)는 테스트 대상인 분산 애플리케이션의 코드에 에러가 있다고 판단할 수 있다.
- [0070] 연산 장치(430)는 데이터를 처리하고, 일정한 연산을 처리하는 프로세서, AP, 프로그램이 임베디드된 칩과 같은 장치일 수 있다.
- [0072] 또한, 상술한 바와 같은 블록체인 기반 분산 애플리케이션 테스트 방법은 컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 어플리케이션)으로 구현될 수 있다. 상기 프로그램은 일시적 또는 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.
- [0073] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM (read-only memory), PROM (programmable read only memory), EPROM(Erasable PROM, EPROM) 또는 EEPROM(Electrically EPROM) 또는 플래시 메모리 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.
- [0074] 일시적 판독 가능 매체는 스태틱 램(Static RAM, SRAM), 다이내믹 램(Dynamic RAM, DRAM), 싱크로너스 디램(Synchronous DRAM, SDRAM), 2배속 SDRAM(Double Data Rate SDRAM, DDR SDRAM), 증강형 SDRAM(Enhanced SDRAM, ESDRAM), 동기화 DRAM(SyncLink DRAM, SLDRAM) 및 직접 램버스 램(Direct Rambus RAM, DRRAM) 과 같은 다양한 RAM을 의미한다.
- [0076] 본 실시례 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

도면

도면1



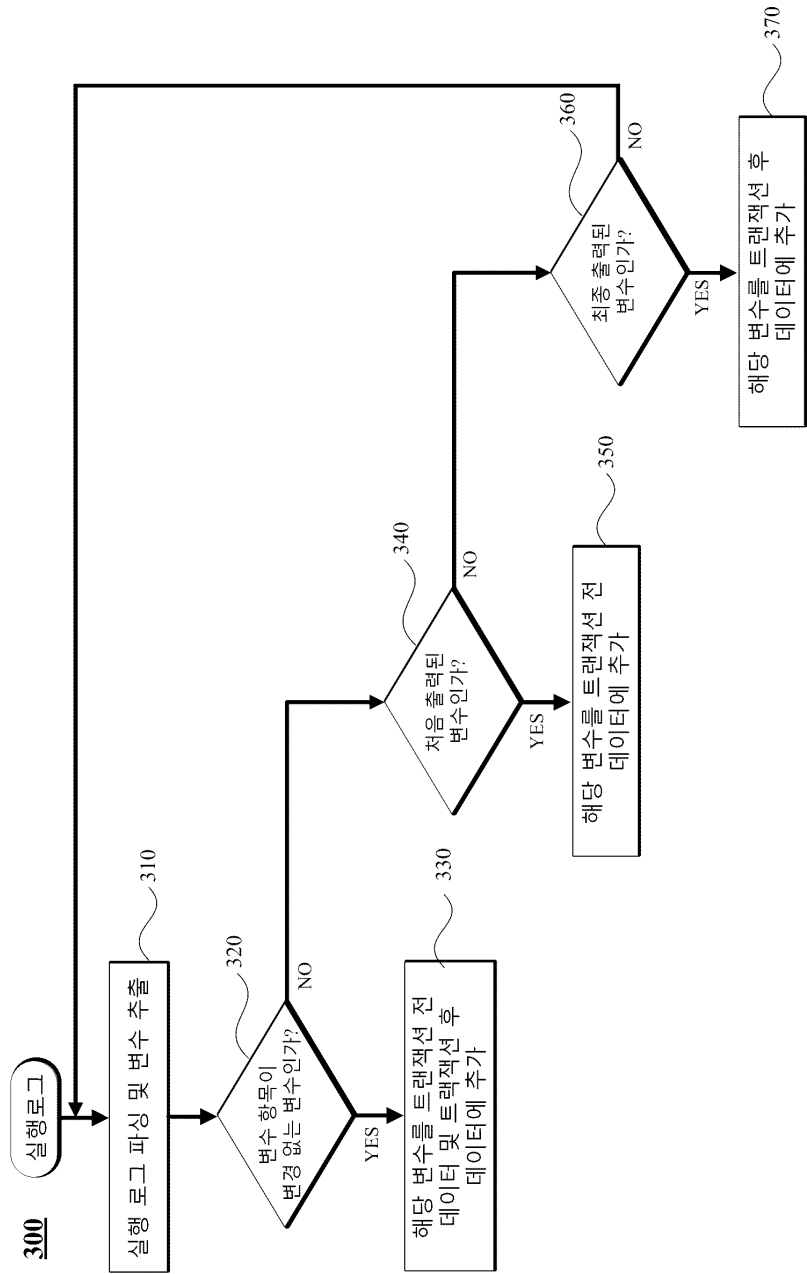
도면2



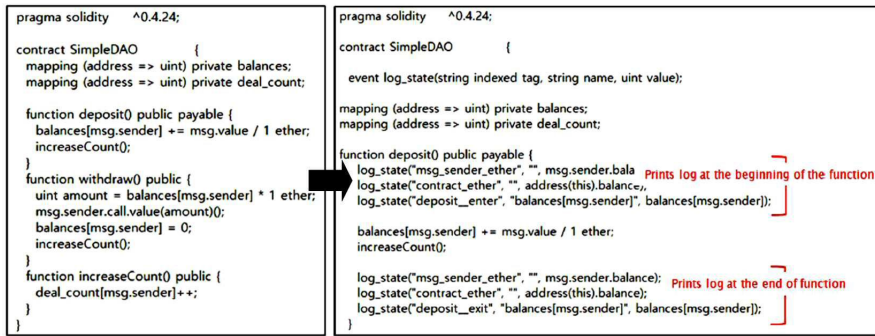
도면3

구분	항목
트랜잭션 기본 정보	- 트랜잭션 해시 - 트랜잭션 입력 값
트랜잭션 송신 계정 정보	- 트랜잭션 송신 계정 - 트랜잭션 송신 계정 이더량
컨트랙트 상태 정보	- 컨트랙트 계정 - 컨트랙트 계정 이더량 - 상태 변수 값

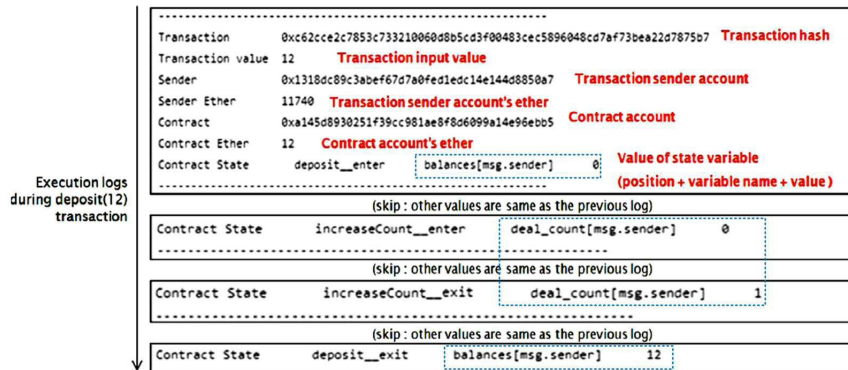
도면4



도면5



(A)



(B)

<Data before transaction>

Tx.hx_hash	Tx.tx_value	Tx.tx_sender_state tx_sender_address	Tx.tx_sender_state tx_sender_ether	Tx.tx_contract_state tx_contract_address	Tx.tx_contract_state tx_contract_ether	Tx.tx_contract_state balances	Tx.tx_contract_state deal_count
0xc62cce2c7853c733210060d8b5cd3f00483cec5896048cd7af73bea22d7875b7	12	0x1318dc89c3abef67d7a0fed1edc14e144d8850a7	11740	0xa145d8930251f39cc981ae8f8d6099a14e96ebb5	12	0	0

<Data after transaction>

Tx.hx_hash	Tx.tx_value	Tx.tx_sender_state tx_sender_address	Tx.tx_sender_state tx_sender_ether	Tx.tx_contract_state tx_contract_address	Tx.tx_contract_state tx_contract_ether	Tx.tx_contract_state balances	Tx.tx_contract_state deal_count
0xc62cce2c7853c733210060d8b5cd3f00483cec5896048cd7af73bea22d7875b7	12	0x1318dc89c3abef67d7a0fed1edc14e144d8850a7	11740	0xa145d8930251f39cc981ae8f8d6099a14e96ebb5	12	12	1

Transaction basic Info Transaction sender account Info Contract state Info

(C)

도면6

