한국컴퓨터정보학회 동계학술대회 논문집 제27권 제1호 (2019. 1)

# 안전한 이더리움 분산 어플리케이션 개발을 위한 스테이트 머신 기반의 디자인 패턴

엄현민[O], 이명준[*]
[O*]울산대학교 전기전자컴퓨터공학과
e-mail: lemony3383@gmail.com[O], mjlee@ulsan.ac.kr[*]

# A State Machine Design Pattern for Secure Ethereum Dapp

Hyun-min Eom[O], Myung-Joon Lee[*]
[O*]Dept. of Electrical/Electronic and Computer Engineering, University of Ulsan

● 요 약 ●

최근 블록체인 기반의 어플리케이션이 증가하고 이들을 위한 스마트 컨트랙트가 설계상 오류로 부적절하게 사용될 가능성이 증대되고 있다. 따라서 스마트 컨트랙트의 설계를 보다 안전하게 지원할 수 있는 방안이 필요한 실정이다. 본 논문에서는 State machine을 이용하여 이더리움 스마트 컨트랙트의 기능사용을 보다 안전하게 지원하기 위한 기법을 제안한다. 제안된 기법은 전체 동작의 흐름의 제어하기 위한 Transition Contract와 각각 상태에 대한 스마트 컨트랙트인 State Contract를 이용하여 스마트 컨트랙트의 동작과정을 제어한다.

**키워드:** 스마트 컨트랙트(smart contract), 블록체인(blockchain), 스테이트 머신(state machine)

## I. Introduction

최근 블록체인[1] 기반의 어플리케이션들이 다양한 분야에서 개발되고 있고 그 수가 빠르게 증가하고 있다[2]. 또한 이들을 위한 많은 스마트 컨트랙트가[3] 이더리움[4], EOS[5] 등의 블록체인에 배포되어 사용되고 있고 더욱 빠르게 증가할 것으로 예상된다. 그러나 스마트 컨트랙트의 수가 증가함에 따라 스마트 컨트랙트의 설계상 오류로 스마트 컨트랙트의 기능을 악의적으로 사용할 가능성이 증대되고 있다. 이러한 현상은 블록체인에 잘못된 정보가 저장되거나 복구를 위한 비용이 발생하는 등의 많은 문제점을 야기할 수 있다. 따라서 스마트 컨트랙트의 설계를 보다 안전하게 지원할 수 있는 디자인 패턴이 필요한 실정이다.

본 논문에서는 State machine을 이용하여 이더리움 스마트 컨트랙트의 기능사용을 보다 안전하게 지원하기 위한 기법을 제안한다. 제안하는 기법은 state machine을 스마트 컨트랙트에 적용시켜 상태를 기반으로 스마트 컨트랙트의 기능사용을 제어한다. 이를 위하여 본 논문에서는 전체 동작의 흐름의 제어하기 위한 Transition Contract와 각각 상태에 대한 스마트 컨트랙트인 State Contract를 제안한다. Transition Contract는 어플리케이션의 수행 상태를 관장하고 다음 동작을 수행할 상태를 결정한다. State Contract는 각 상태별 동작을 수행하고 이에 대한 이벤트를 발생시킨다.

## II. Preliminaries

이더리움 분산 어플리케이션은 이더리움 블록체인 네트워크에서 동작하는 어플리케이션이다. 분산 어플리케이션은 블록체인 네트워크에 배포되어 있는 스마트 컨트랙트를 통하여 다양한 기능을 수행한다. 스마트 컨트랙트는 기능 실행을 위한 메소드와 변수 등을 가지며 이러한 변수, 메소드 그리고 기능 실행의 내역 등과 같은 정보들을 블록체인에 기록한다. 또한, 이러한 정보들은 블록체인 네트워크를 구성하는 블록체인 노드에 분산, 복제되어 저장된다. 블록체인 분산 어플리케이션은 스마트 컨트랙트를 통해 기능을 수행하고 이에 대한 정보들을 저장하여 신뢰성 있는 서비스의 제공이 가능하다. 최근 이더리움 어플리케이션이 증가함에 따라 Yakindu에서는[6] statechart 기능을 이더리움 컨트랙트와 연계시키는 작업을 진행 중이다.

## III. The Proposed Scheme

제안하는 기법은 State Machine의 상태, 액션, 이벤트 등을 이용하여 스마트 컨트랙트의 기능사용을 제어한다. 이를 위하여 제안하는 기법은 전체 제어의 흐름을 관장하는 Transition Contract와 각각의 상태의 동작을 수행하는 State Contract를 이용한다. Transition Contract는 현재의 상태를 저장하고 각 상태에 따라 해당하는 컨트랙트를 호출하고 이를 제어한다. State Contract는 해당하는 상태에서

수행해야 할 동작을 수행하고 그 결과로 다음 상태로 변경을 알리는 이벤트를 발생시킨다. 그림 1은 제안하는 기법의 동작과정을 나타낸다.

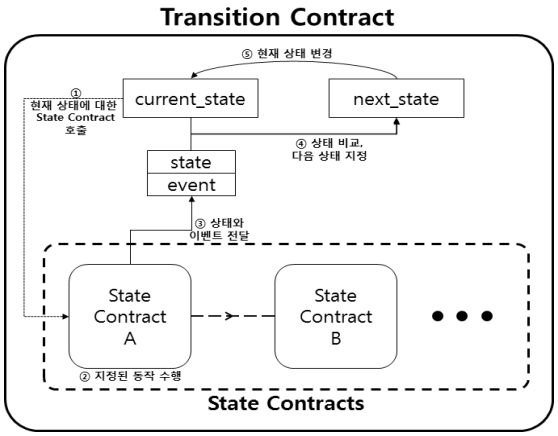


Fig. 1. Process for Smart Contracts

제안하는 기법의 동작과정은 다음과 같다.

(1)Transition Contract는 현재의 상태에 해당하는 State Contract를 호출하여 초기화한다.

(2)호출된 State Contract는 해당하는 상태에서 수행되어야 하는 동작을 수행한 후 Transition Contract에게 현재의 상태와 발생한 이벤트를 전달한다.

(3)Transition Contract는 전달받은 상태가 저장된 현재 상태인지 확인하고 전달받은 이벤트를 통하여 변경될 다음 상태를 지정한다.

(4) Transition Contract는 전달받은 이벤트가 현재 상태에서 발생 가능한 이벤트인지 검사하고 적절한 이벤트일 경우 현재 상태를 지정된 다음 상태로 변경한다.

(5)상태를 변경한 Transition Contract는 변경된 상태에 대한 State Contract를 호출한다.

이를 통하여 제안하는 기법은 Transition Contract에 의해서만 State Contract들이 실행되도록 하고, 또한 적합한 상태와 환경에서만 기능이 실행되도록 제어하여 스마트 컨트랙트의 기능사용이 보다 안전하게 실행되도록 지원한다.

## IV. Conclusions

본 논문에서는 State machine을 이용하여 이더리움 스마트 컨트랙트의 기능사용을 보다 안전하게 지원하기 위한 기법을 제안하였다. 제안된 기법은 Transition Contract와 State Contract를 이용하여 상태를 저장하고 상태에 따른 동작, 이벤트를 통한 상태의 변경을 이용하여 스마트 컨트랙트의 동작과정을 제어한다. 제안된 기법은 정해진 절차에 따라 동작하는 분산 어플리케이션을 위한 스마트 컨트랙트의 신뢰성을 높이기 위한 해결방법으로 활용될 수 있다.

## ACKNOWLEDGEMENT

본 연구는 중소벤처기업부와 한국산업기술진흥원의 “지역특화산업육성사업(R&D, P0006342)”으로 수행된 연구결과입니다. (Corresponding Author: 이명준)

## REFERENCES

[1] G. Zyskind, and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data", Security and Privacy Workshops (SPW), IEEE, pp. 180-184, 2015.

[2] A. Bogner, M. Chanson and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum block-chain.", Proceedings of the 6th International Conference on the Internet of Things, pp. 177-178, 2016.

[3] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts", 2016 IEEE Interna-tional Conference on Consumer Electronics (ICCE), pp. 467-468, 2016.

[4] V. Buterin, "A next-generation smart contract and decentral-ized application platform", Ethereum project white paper, 2014.

[5] EOS, *https://eos.io/*

[6] Yakindu, *https://www.itemis.com/en/yakindu/*