스마트 컨트랙트 프로그래밍 언어 동향 조사

Survey on Smart Contract Programming Languages

김익순 (Ik-Soon Kim, ik-soon.kim@etri.re.kr) 차세대콘텐츠본부 책임연구원

ABSTRACT

Blockchain is an enabling technology for managing data with high trust and transparency among connected computers. Blockchain emerged with the advent of the Bitcoin cryptocurrency, and then, evolved as general-purpose platforms such as Ethereum, EOS, R3 Corda, and IBM Hyperledger Fabric. The application of blockchain covers a broad range of areas such as fintech, decentralized identity, distribution, real estate trading, games, and drone air traffic management. Smart contracts are indispensable for constructing blockchain services. This survey classifies smart contract languages by their features and shows their differences from existing general-purpose programming languages.

KEYWORDS smart contract, blockchain, programming language

Ⅰ. 서론

블록체인은 암호화폐인 비트코인[1]과 함께 등장하였으며, 이후 이더리움[2], EOS[3], R3 Corda [4], IBM 패브릭[5] 등과 같은 범용 블록체인 기술로 발전하였다. 블록체인은 핀테크, 신원 인증, 식자재, 귀금속 및 디지털 콘텐츠 유통, 부동산 거래, 게임, 드론 항공 교통 관리에 이르기까지 다양한 분야에 활용되고 있다.

블록체인 서비스는 스마트 컨트랙트(Contract) 개발을 통하여 이루어진다. 서면으로 작성되던 기존의 계약서와 달리, 스마트 컨트랙트는 프로그램 형태로 작성되고, 계약 이행 요청 시 미리 프로그램된 계약 조건에 따라 계약 내용을 수행한다.

블록체인 서비스의 수행 과정은 스마트 컨트랙트 호출과 이에 대한 수행 결과로 볼 수 있다. 블록체인에 대한 사용자의 요청은 스마트 컨트랙트 호출 과정이며, 이에 대한 수행 순서 및 결과는 합의 과정을 거쳐 블록에 저장되어 배포된다.

본 동향 조사는 다양한 블록체인 시스템에서 사

* DOI: https://doi.org/10.22648/ETRI.2020.J.350512

* 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2020년도 문화기술연구개발 지원사업으로 수행되었음. This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2020

용되고 있는 스마트 컨트랙트 프로그래밍 언어들을 특성에 따라 분류하고, 기존의 프로그래밍 언어와의 차이점에 대해서도 살펴보도록 하겠다.

Ⅱ. 블록체인

블록체인은 네트워크로 연결된 컴퓨터들 사이에서 투명하고 신뢰성 있게 자료를 관리하는 기술이다. 전통적인 클라이언트 서버 모델에서는 특정 기관의 서버 컴퓨터들이 사용자의 요구 사항을 단독적으로 결정하여 자료 연산을 수행한다. 이 연산 과정이 사용자에게 투명하게 공개되지 않으며, 프로그램 오류로 인하여 또는 의도적으로 자료 연산이 조작될 가능성이 있다. 블록체인은 이러한 클라이언트 서버 모델의 문제점들을 방지할 수 있다.

블록체인은 합의 과정을 통해서 신뢰성 있게 자료 연산을 수행한다. 합의 과정은 일반적으로 특정 기관 또는 여러 기관들이 운영하는 다수의 컴퓨터상에서 수행된다. 합의 과정을 수행하는 컴퓨터들은 사용자의 요청 사항에 따라 자료 연산을 수행하고, 이 수행 결과를 합의 알고리즘을 통하여 상호 검증함으로써 연산 결과의 신뢰성을 높인다. 프로그램 오류나 의도적인 결과 조작이 최종 연산 결과에 영향을 미치기 매우 어렵다.

블록체인은 합의 결과를 사용자에게 배포함으로써 자료 연산 과정을 투명하게 공개한다. 사용자에게 배포된 블록체인 내용을 조사함으로써 요청 사항 내용, 요청 사항 처리 순서 그리고 요청 사항 처리 결과를 확인할 수 있다. 또한, 블록체인은 사용자들이 자료를 공유함으로써 블록체인의 자료를 임의로 변경하는 것은 매우 어렵다.

Ⅲ. 스마트 컨트랙트

블록체인 서비스는 스마트 컨트랙트 개발을 통하여 이루어진다. 스마트 컨트랙트는 프로그램 형태로 작성된 계약서로, 계약 이행 요청 시 미리 프로그램된 계약 조건에 따라 계약 내용을 수행한다. 대부분의 블록체인 시스템은 스마트 컨트랙트의 등록, 삭제 기능을 제공한다. 스마트 컨트랙트를 운영 중인 블록체인 시스템에 등록, 삭제할 수 있으며, 등록된 스마트 컨트랙트를 호출함으로써 계약 이행을 요청할 수 있다.

블록체인 서비스의 수행 과정은 스마트 컨트랙트 호출과 이에 대한 수행 결과로 볼 수 있다. 블록체인은 사용자 요청을 트랜잭션 형태로 받아서, 합의 과정을 통하여 트랜잭션 수행 순서 및 수행 결과를 결정하고, 합의 결과를 블록에 저장하여 모든 사용자에게 배포한다. 블록체인의 사용자 요청은 스마트 컨트랙트 호출에 해당하며, 이에 대한 수행 결과는 블록에 저장되어 사용자에게 배포된다.

스마트 컨트랙트 프로그램은 기존의 프로그램과 달리 블록체인의 블록에 자료를 저장하고, 수정할 수 있다. SQL 프로그램이 DBMS의 자료를 관리하듯이, 스마트 컨트랙트 프로그램은 블록의 자료를 관리한다. 스마트 컨트랙트 호출로 블록에 저장된 자료는 호출을 마친 후에도 블록에 그 상태를 계속 유지한다.

Ⅳ. 스마트 컨트랙트 프로그래밍 언어

1. 프로그래밍 패러다임

가. 객체 지향 언어

Solidity[6] 언어는 이더리움에서 사용하는 객체

지향 언어로 정적 타입, 다중 상속, 다양한 자료 타입과 블록에 자동 저장되는 스토리지 변수를 제공하며, EVM 코드(향후 웹어셈블리 지원)로 컴파일하여 수행한다. Solidity 언어의 컨트랙트는 다중 상속을 지원해서 기존에 작성한 컨트랙트를 상속하여 확장된 새로운 컨트랙트를 작성할 수 있다.

나. 함수형 언어

이더리움에서 함수형 언어를 사용하려는 다양한 시도들이 있다. EthereumH는 Haskell 언어를, 이더리움 Idris는 Idris 언어를, Pyramid 언어는 Scheme 언어를, LLL 언어는 Lisp과 유사한 언어를 이더리움 스마트 컨트랙트 개발에 사용할 수 있게 지원하고 있다.

Sophia[7] 언어는 Aeternity에서 사용하는 ML 계열의 함수형 스마트 컨트랙트 언어이다. Liquidity[8] 언어도 Dune/Tezos에서 ML 계열의 함수형 스마트 컨트랙트 언어이다. Liquidity 언어는 스택 기반 중간 언어 Michelson로 컴파일하여 수행한다. Plutus[9] 언어는 Cadano에서 사용하는 스마트 컨트랙트 언어로 Haskell 문법이지만 Eager Evaluation을 지원하는 언어이다.

다. 로직 언어

Logikon[10] 언어는 Prolog 언어와 유사한 로직 언어로 SMT solver 표준 언어인 SMTLib2를 사용할 수 있다. Logikon 프로그램의 의미는 Prolog와 유사하다. Logikon 언어는 컴파일 시간에 정적 검증을 목표로 설계되었고, Solidity의 중간 언어인 YUL을 거쳐 EVM 코드로 컴파일하여 실행한다. 다음은 Logikon 언어로 List 최소값을 계산하는 재귀 함수로 ite는 if-then-else를 의미하는 SMTLib2 언어이다.

```
define recursive min (List) -> Uint
case ([X]) X.
case ([H:T*]) X :-
      (= Y (min T))
      (= X (ite (< H Y) H Y)).
```

라. Concurrent 언어

Rholang[11] 언어는 Rchan[12]에서 사용하는 스마트 컨트랙트 언어로, RHO-calculus[13]에 기반한 언어이다. 비동기 프로세스들 간에 채널을 통한 메시지 전달이 가능하며, 채널을 통하여 프로세스 전달(또는 higher order process 작성)이 가능하다. 이러한 특징들을 이용하여 Concurrent 스마트 컨트랙트를 작성할 수 있다.

2. 실행 방식

특정 블록체인 시스템들은 기존의 범용 프로그래밍 언어를 그대로 스마트 컨트랙트 언어로 사용하고 있다. 이러한 블록체인 시스템들은 별도의 SDK를 제공하여 스마트 컨트랙트 프로그램이 실행 시간에 블록체인 엔진과 통신할 수 있는 방법을 제공한다. IBM Fabric은 Java, JavaScript, Go 언어를 이용하여 스마트 컨트랙트를 개발할 수 있다.

자체적인 스마트 컨트랙트 언어를 제공하는 블록체인 시스템은 컨트랙트 프로그램을 바이트 코드로 컴파일한 후 가상 기계에서 수행한다. 비트코인은 비트코인 스크립트라는 스택 기반 언어를 제공하고 있으며, 이를 바이트 코드로 변환하여 비트코인 엔진에서 수행한다. 이더리움은 다양한 스마트 컨트랙트 언어를 제공하는데, 이러한 언어로 작성한 컨트랙트 프로그램을 바이트 코드로 컴파일하여 EVM 가상 기계에서 수행된다. 향후, 이

더리움은 자체적인 바이트 코드 대신 W3C에서 표준화 중인 웹어셈블리[14] 코드를 사용할 예정이다.

한편, EOS는 기존의 범용 프로그래밍 언어인 C++를 사용하여 스마트 컨트랙트를 개발한다. 개발자는 C++언어와 EOS 제공 라이브러리를 사용하여 프로그램을 작성하고 이를 웹어셈블리로 컴파일하여 수행한다.

웹어셈블리는 바이너리 프로그램 로딩 시 프로그램의 안전성을 검사할 수 있다. 뿐만 아니라, 향후 표준화된 웹어셈블리로 변환될 다른 프로그래밍 언어를 스마트 컨트랙트 언어로 사용할 수 있다는 장점도 있다.

3. 튜링 완전성

튜링 기계는 대표적인 계산 모델로, 컴퓨터의 참조 모델로 많이 사용되고 있다. 어떤 프로그래밍 언어가 튜링 완전하다는 것은 튜링 기계로 계산할 수 있는 모든 것을 그 프로그래밍 언어를 이용하여 계산할 수 있다는 의미이다.

스마트 컨트랙트 프로그램 수행이 지나치게 오래 걸리거나 멈추지 않으면 이는 블록체인의 성능에 커다란 영향을 끼칠 수 있다. 많이 블록체인 시스템들은 이러한 문제를 방지하기 위하여 노력하고 있다.

이더리움은 스마트 컨트랙트 수행 시 비용(gas 값)을 지불하도록 함으로써 스마트 컨트랙트 수행 시간을 제한하고 있다. 다른 블록체인 시스템들은 스마트 컨트랙트 언어를 튜링 불완전하게 설계하여 스마트 컨트랙트 프로그램 수행이 오래 걸리지 않도록 하고 있다.

비트코인 스크립트[15]는 비트코인에서 사용하는 Forth(또는 Postscript)와 유사한 스택 기반의 언어로 loop을 지원하지 않는 튜링 불완전 언어이다. 비트코인 상에서 스마트 컨트랙트 수행 과정이 지나치게 오래 걸리거나 무한 반복에 빠지지 않도록 loop을 지원하지 않는다. 비트코인 스크립트는 loop을 지원하지 않아서 계산 능력 많이 제한된다는 단점이 있다.

Pact[16] 언어는 Kadena에서 사용하는 튜링 불완전 스마트 컨트랙트 언어이다. 스마트 컨트랙트 수행 시 무한 반복에 빠지지 않도록 재귀 호출을 금지하고, 리스트 기반의 연산자(map, fold, filter등)로 loop을 대신하고 있다.

Ride[17] 언어는 waves에서 사용하는 함수형 스마트 컨트랙트 언어로 재귀함수, loop, goto를 지원하지 않는 튜링 불완전 언어이다.

4. 기타

블록체인이 제공하는 스마트 컨트랙트 언어들은 Atomic 수행을 지원한다. 스마트 컨트랙트 실행 중 오류가 발생하면, 현재까지의 스마트 컨트랙트의 실행 결과를 무시하고, 블록체인 상태를 스마트 컨트랙트 호출 이전으로 복구한다. 가령, 이더리움의 경우 gas 비가 부족하거나 오류가 발생하면 스마트 컨트랙트 실행 결과는 무시되고, 호출 이전의 상태로 복구된다. 또한, Solidity 언어와 Pact 언어의 경우는 명시적으로 현재 처리하고 있는 작업을 롤백할 수 있다.

일부 스마트 컨트랙트 언어에서 SQL 기능을 제공한다. Kadena의 Pact 언어는 SQL을 이용하여 블록 정보에 대한 자료 연산이 가능하다. Aergo는 lua, SCL 등의 스마트 컨트랙트 언어를 지원하지만, JDBC를 통하여 SQL 기능을 사용할 수 있다.

V. 결론

블록체인 기술은 핀테크, 신원 인증, 식자재, 귀금속 및 디지털 콘텐츠 유통, 부동산 거래, 게임, 드론 항공 교통 관리 등 다양한 분야로 응용 분야를 넓혀가고 있다. 이러한 블록체인 서비스를 구축하기 위하여 스마트 컨트랙트 개발이 필수적이다.

최근 스마트 컨트랙트 언어들은 스마트 컨트랙트 개발의 편의성 및 성능 향상을 위하여 Concurrency를 비롯한 다양한 패러다임의 언어 및 DSL이 등장하고 있다. 최근 스마트 컨트랙트 언어는 웹어셈블리로 변환되어 실행되기도 한다. 또한, 최근 스마트 컨트랙트 언어들은 안전한 스마트 컨트랙트 프로그램 작성을 위하여 Formal Method 기반의 프로그램 작성 및 증명 방법이 적용되고 있다.

최근 등장하는 스마트 컨트랙트 언어는 기존의 프로그래밍 언어에 대한 연구 주제들이 적용되고 있으며, 블록체인의 필요성이 증가하면서 이러한 추세는 계속 이어질 것으로 예상된다.

약어 정리

DSL	Domain Specific Language
EVM	Ethereum Virtual Machine
ML	Meta Language
RHO	Reflexive Higher Order
SQL	Structured Query Language

참고문헌

[1] Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cach system," 2008년 12월.

[2] 이더리움, https://ethereum.org

[3] EOSIO-Blockchain software architecture, https://eos.io

[4] Corda, https://r3.com

[5] Hyperledger Fabric, https://www.hyperledger.org

[6] Solidity, https://solidity.readthedocs.io

[7] Sophia, https://aeternity.com/#sophia

[8] Liquidity, https://liquidity-lang.org

[9] Plutus, https://github.com/input-output-hk/plutus

[10] Logikon, https://github.com/logikon-lang/logikon

[11] Rholang, https://rchain.coop/platform

[12] Rchain, https://rchain.coop

[13] Meredith, L. G.; Radestock, Mattias, "A Reflective Higher-Order Calculus," Electronic Notes in Theoretical Computer Science. 141 (5), 2005년 12월,

[14] Andreas Haas Andreas Rossberg Derek L. Schuff, Ben L. Titzer, Michael Holman, Dan Gohman Luke Wagner Alon Zakai, JF Bastien, "Bringing the Web up to Speed with WebAssembly," PLDI, 2017년.

[15] 비트코인 스크립트, https://en.bitcoin.it/wiki/Script

[16] Pact, https://pact-language.readthedocs.io/ko/latest/

[17] Ride, https://docs.waves.tech/en/ride/