ISSN 1016-9288
제49권 12호
2022년 12월호

전자공학회지

The Magazine of the IEIE

vol.49. no.12

블록체인 기술 동향

- 블록체인 기술과 응용
- 블록체인의 Trilemma
- 블록체인 기술 동향과 Web3.0
- 웹 3.0 활성화를 위한 시나리오 발굴
- 블록체인과 전통적 합의 알고리즘의 이해
- 영지식 증명 이해와 블록체인 활용
- 블록체인 코드와 품질 가시화

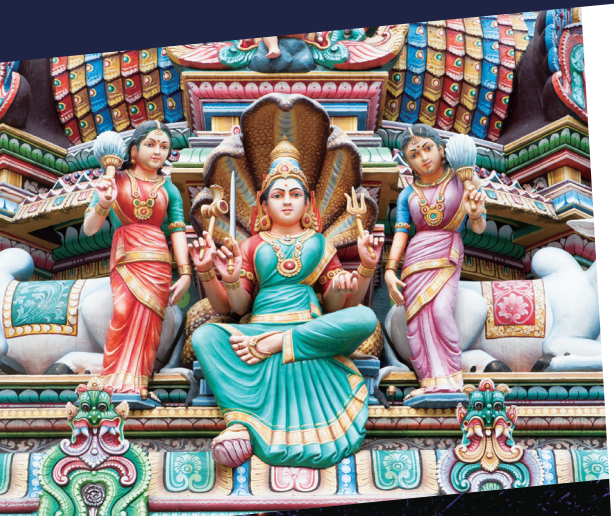
대한전자공학회
The Institute of Electronics and Information Engineers
www.theieie.org

http://www.iceic.org

ICEIC2023

INTERNATIONAL CONFERENCE ON ELECTRONICS, INFORMATION, AND COMMUNICATION 2023

FEB. 5(SUN) – 8(WED) I SHANGRI LA, SINGAPORE



CALL FOR PAPERS

The 22st International Conference on Electronics, Information, and Communication (ICEIC 2023) is a forum open to all the participants who are willing to broaden professional contacts and to discuss the state-of-the-art technical topics.

Regular sessions of ICEIC 2023 will include more than 300 oral and poster presentations. In addition, the conference will offer special sessions, invited talks, keynote speeches, and tutorials to cover a broad spectrum of topics on electronics, information, and communication technologies.

IMPORTANT DATES

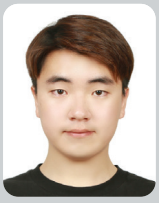
- **Submission of Paper** : September 18, 2022
- **Notification of Acceptance** : October 21, 2022
- **Submission of Camera-Ready Paper** : November 14, 2022

IEIE IEEE

Organized by The Institute of Electronics and Information Engineers (IEIE)

블록체인 코드와 품질 가시화


김장환
홍익대학교


박찬솔
홍익대학교

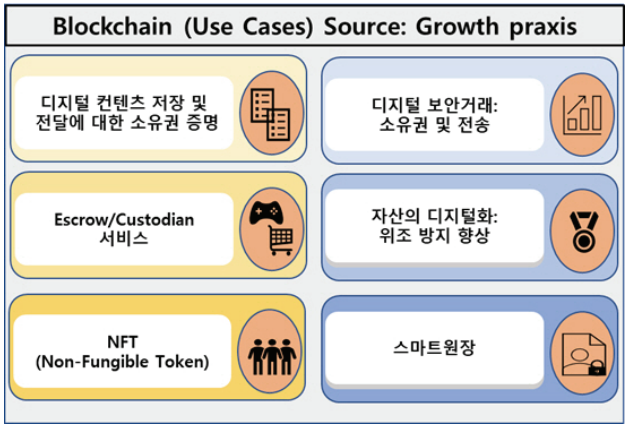

문소영
홍익대학교


장우성
홍익대학교


김영철
홍익대학교

Ⅰ. 서 론

최근 4차 산업혁명이 가장 큰 이슈로 떠오르면서 AI,자율주행, 블록체인 등 새롭고 다양한 시스템들이 등장하고 있다. 그 중에서도 블록체인 기술은 비트코인을 포함한 수많은 가상화폐들이 등장하면서 전 세계의 주목을 한 몸에 받는다. 이러한 현상과 함께 가상화폐(비트코인, 이더리움, 리플 등)가 미래의 화폐가 될 것이라는 주장들이 대두되면서 사람들의 관심이 높아지고 있다. 국내뿐만 아니라 전 세계에서 각종 금융기관 및 미디어는 지속적으로 블록체인과 가상화폐에 대해서 기사를 이어간다. 하지만 가상화폐는 블록체인이라는 기술로 구현하는 것들 중 일부분에 불과할 뿐, 블록체인 기술이 발전하고 점점 더 상용화 된다면 단지 화폐의 대체제로 인기가 높은 것을 떠나 다양한 분야에서 접목시키려는 시도가 더욱 커질 것이다[1].



〈그림 1〉 블록체인 기술 사용사례 도표

〈그림1〉은 여러 분야에서 사용되는 블록체인기술의 사용사례를 나타낸다. 디지털 콘텐츠에 대한 소유권 증명, Escrow/Custodian 서비스, 디지털 보안 거래, 스마트원장, 등 이 분야들 이외에도 여러 분야에서 블록체인 기술을 적용하는 연구가 활발하다.

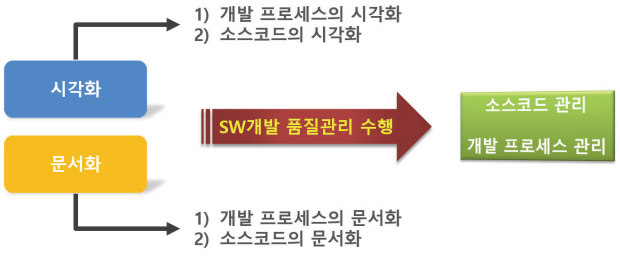
하지만, 이처럼 관심이 커진 만큼 블록체인에 대한 잘못된 정보나 모호한 이해를 통해 블록체인에 대한 잘못된 정보가 확대 재생산 되고 있는 실정이다. 대표적으로 블록체인이 탈중앙화로 인한 데이터의 위 · 변조를 원천봉쇄할 수 있는 데이터 분산기술을 가지고 있기에 사람들은 블록체인 기술이 완벽할 것이라고 생각하는 데, 이것은 잘못된 생각이다. 블록체인은 여러 사용자가 노드를 소유하며 동일한 트랜잭션 정보를 바탕으로 동시에 승인하는 시스템을 구축하기 하는데, 만약 여러 노드 중 한 노드에 부하가 발생하면 해당 트랜잭션 승인에 있어 문제가 발생할 수 있다[2]. 이를 악용해 인위적으로 트랜잭션 승인에 문제를 만든다면 단순히 보안적인 측면뿐만 아니라 수행하는 코드까지 내려가서 문제를 해결할 수 있는 방법을 찾아야한다. 하지만, 이러한 코드들은 문자들의 집합으로 이루어 있기 때문에 코드들의 의미를 파악하고 문제점을 인지하여 개선하기까지는 많은 시간이 걸린다.

또한 가장 중요한 이슈는 새로운 프로그래밍 언어 및 블록체인 관련 코드(Java, Go, Solidity, Lua 등)들에 대한 상용화된 정적분석기들이 존재하고 있지 않다. 따라서 해당 언어의 코드들에 대한 정보를 측정할 수 없기 때문에 해당 언어들의 품질을 보장할 수 없다. 이렇게 소프트웨어의 내부를 잘 파악할 수 없는 특징을 소프트웨어 공학에서는 소프트웨어의 비가시성이라고 한다[3].

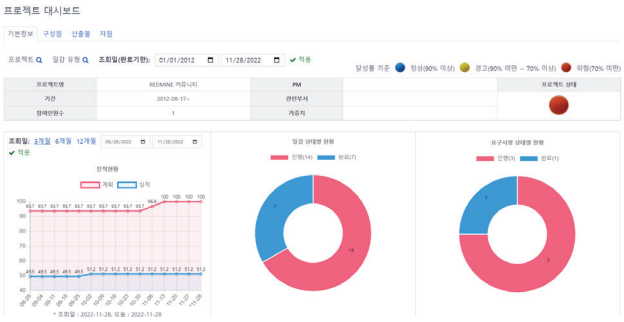
소프트웨어의 비가시성은 소프트웨어 개발자 입문하는데 큰 장애물로 다가온다. 왜냐하면 머릿속으로 생각한 소프트웨어를 코드를 통해 구현할 때 계획한대로 혹은 생각한대로 구현되지 않기 때문이다. 소프트웨어의 규모가 작을 경우 혹은 단순하여 복잡하지 않은 소프트웨어일 경우, 소프트웨어 개발에 방해가 되진 않지만, 반대의 경우 큰 문제를 야기한다. 게다가 지난 세월동안 소프트웨어의 규모는 지속적으로 커져왔고 단일 개발자가 단독으로 소프트웨어의 개발부터 테스트, 유지 및 보수까지 하는 것은 매우 어렵다. 우리는 다양한 분야의 여러 전문가와 이해관계자들이 함께 협업하여 소프트웨어를 개발하는 시대에 살고 있다. 이러한 경우, 스스로 작성하지도 않은 코드를 읽고 분석하는 것은 분명히 어려운 일이다. 이런 문제를 개선하고자 소프트웨어공학에서는 다양한 방향을 통해 문제를 개선하고자 시도하는데 그 방법 중 하나는 바로 소프트웨어 가시화 방법이다.

Ⅱ. 기존 소프트웨어 가시화

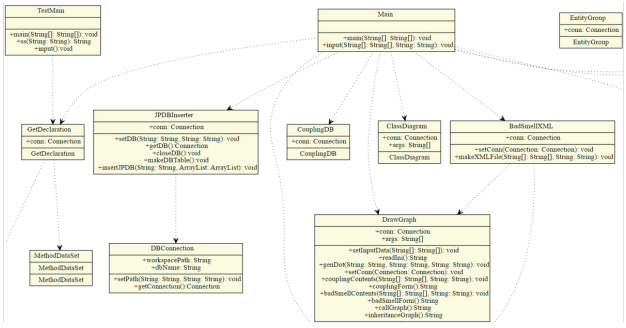
소프트웨어 가시화는 문자 그대로 소프트웨어를 시각화 하여 소프트웨어의 내부를 보여준다. 흔히들, 소프트웨어의 가시화를 생각하면 GUI를 떠올리기 쉽다. 하지만, 소프트웨어 가시화는 GUI와는 다르다. 소프트웨어 내부를 시각화하는 것은 소프트웨어 개발의 가장 어려운 점인 프로세스와 아키텍처의 비가시성을 극복함으로써 개발의 전체 과정을 파악하며, 이를 통하여 품질 관리를 실현하고자 하는데 목적을 둔다. 또한, 이러한 소프트웨



〈그림 2〉 소프트웨어 가시화 종류



〈그림 3〉 소프트웨어 프로세스 가시화 대시보드 예시[4]



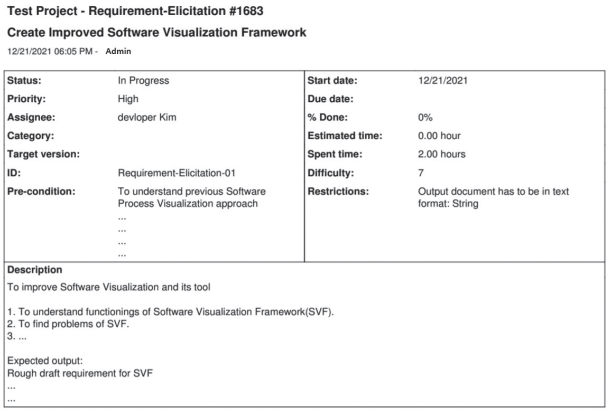
〈그림4〉 코드가시화 결과 다이어그램[5]

어 개발 프로세스를 수행하면 산출되는 문서들을 자동으로 문서화하는 것은 소프트웨어 개발에 중요한 문제이다. 문서화는 기업의 개발과 관련된 업무 이해도 향상과 외부와의 의사소통이 가능하다. 소프트웨어 가시화의 첫 번째는 소프트웨어 프로세스 가시화이다.

〈그림 3〉은 소프트웨어 프로세스를 가시화한 형태로서 대시보드 형식으로 프로세스의 진행사항을 시각적으로 사용자에게 전달하는 것을 나타낸다. 소프트웨어 프로세스 가시화는 기획부터 요구사항 정의 및 분석, 소프트웨어 설계 및 구현, 소프트웨어 검증, 유지 및 보수 등의 단계적 프로세스에 대해 보여준다. 둘째는 소프트웨어 코드 가시화이다.

〈그림 4〉은 코드 가시화로서 타겟 소프트웨어의 코드 구조와 취약점을 도식화해 사용자에게 다이어그램 혹은 그림 형태로 시각화한 것을 나타낸다. 그림의 화살표는 메인함수에서 어떻게 함수들이 연결되어 있는지에 대한 연관관계를 나타낸다.

소스코드의 내부 구조와 취약점을 시각화하기 위해 C.K. Metrics, McCabe의 Cyclomatic Complexity 등



Test Project - Requirement-Elicitation #1683

Create Improved Software Visualization Framework

12/21/2021 06:05 PM - Admin

Status:	In Progress	**Start date:**	12/21/2021
Priority:	High	**Due date:**	
Assignee:	devloper Kim	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	2.00 hours
ID:	Requirement-Elicitation-01	**Difficulty:**	7
Pre-condition:	To understand previous Software Process Visualization approach	**Restrictions:**	Output document has to be in text format: String

Description

To improve Software Visualization and its tool

1. To understand functionings of Software Visualization Framework(SVF).
2. To find problems of SVF.
3. ...

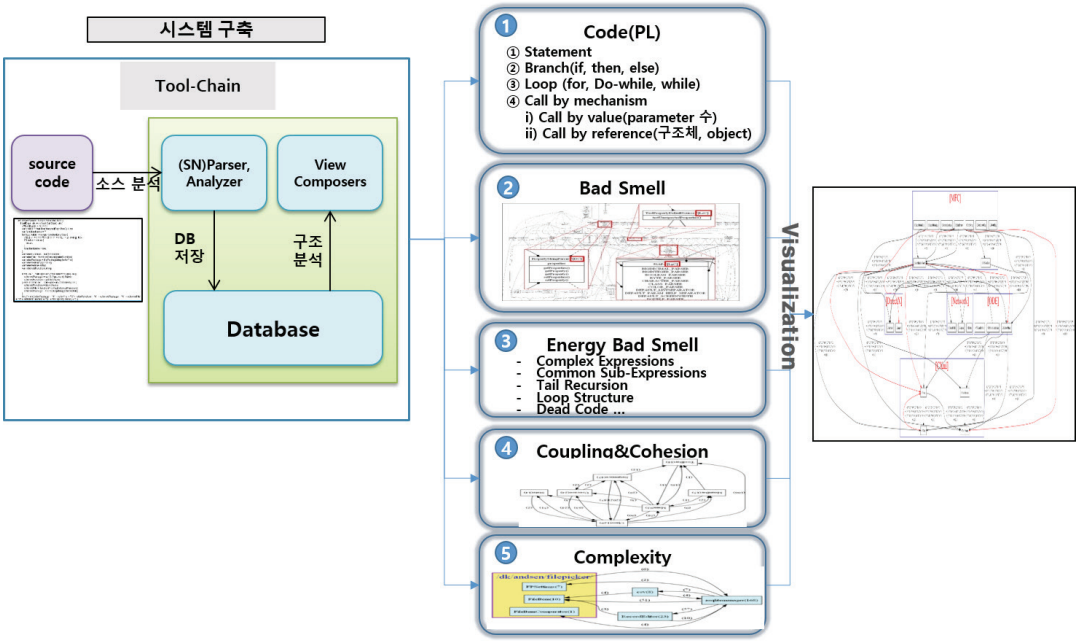
Expected output:
Rough draft requirement for SVF
...
...

〈그림 5〉 자동으로 생성된 요구사항 도출문서[6]

다양한 품질지표를 바탕으로 해당 소프트웨어에 성격에 따라 소스코드를 분석하고 시각화한다. 마지막으로 문서 가시화이다. 문서 가시화는 소프트웨어 프로세스를 수행하면서 발생하는 각종 산출물들과 결과물들을 문서로 만드는 것을 말한다.

〈그림 5〉는 소프트웨어 프로세스 중 요구사항을 명세한 문서를 자동으로 발생시켜 시각화하여 나타낸 그림이다. 문서 가시화는 요구사항 도출, 소프트웨어 스팩 작성, 요구사항 명세, 설계도, 테스트케이스, 테스트 시나리오 등 소프트웨어 프로세스의 수행에서 발생하는 산출물들을 문서로 자동으로 발생시켜 해당 정보를 저장한다. 이처럼 소프트웨어 가시화 기법은 소프트웨어 개발 공정 전 범위에서 소프트웨어를 가시화하여 사용자와 이해관계자의 이해를 돕는다.

〈그림 6〉는 소프트웨어 코드 가시화 툴체인을 도식화하여 나타낸 것이다. 그림의 왼쪽에는 툴체인 시스템 구축 방법을 통해 분석된 코드 데이터가 어떻게 가시화 되는지에 대한 것을 나타낸다. 코드 가시화는 다양한 오픈소스를 활용하여 코드를 분석하여 결과를 보여줄 수 있는 툴체인(Toolchain)를 구성한다. 〈그림 6〉은 오픈 소스 소프트웨어를 활용하여 소프트웨어 코드가시화의 툴체인과 어떤 품질지표를 분석하여 추출하는지를 보여준다. 정적분석기를 통해 소스 코드를 입력하면, 정적분석기(Source Navigator)는 소스코드를 분석하여 코드의 품질과 관련된 지표들을 추출한다. 정적분석기가 소스코드를 통해 분기문, 반복문, 파라미터 패싱, 코드 나쁜 냄

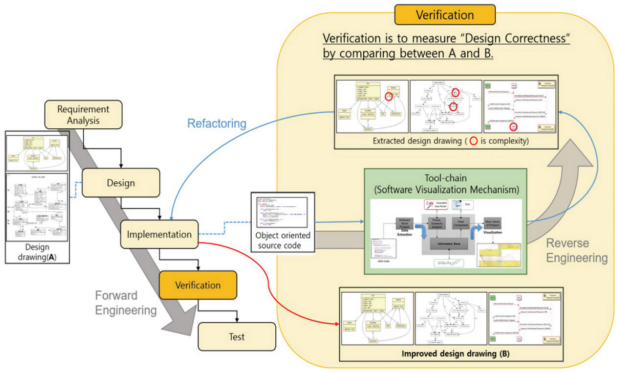


〈그림 6〉 SELab's 코드 가시화 툴체인[6]

새(Code Bad smell)[7], 에너지 나쁜 냄새(Energy bad smell)[8], 결합도&응집도(Coupling and Cohesion)[9], 복잡도(Complexity)[10]와 같은 품질지표를 추출한다. 그런 다음, 정적분석기를 통해 분석된 내용을 데이터베이스에 저장하고, 데이터베이스에서 필요한 품질지표를 질의문(Query Statement)을 통해 원하는 품질지표를 도출할 수 있도록 한다. 코드가시화는 시각화 도구이므로 질의문을 바탕으로 사용자가 원하는 속성에 따라 그래프나 표의 형태로 프로그래밍을 하여 자동으로 코드 구조 추출할 수 있도록 한다.

소프트웨어 개발의 최종 산출물인 소프트웨어는 개발 과정에서 완성된 형태가 명확히 확인되지 않아 오류의 발견 시기를 놓치거나 오류에 대한 해결책을 찾지 못하는 경우가 빈번히 발생한다. 이럴 경우, 뒤늦게 오류를 발견하고 개선하면서 발생하는 유지 및 보수비용은 급격히 증가한다. 그래서 순공학(Forward Engineering) 관점의 가시화는 소프트웨어 프로세스의 현재 단계의 산출물을 기반으로 다음 단계의 산출물을 생산해내는 기법으로 요구사항 분석→설계→구현의 단계를 위에서 아래로 순차적으로 수행되는 것을 가시화한다. 이와는 반대로, 역공학(Reverse Engineering) 관점의 가시화도 존재한다. 역공학 관점의 가시화는 순공학 관점의 가시화와는 반대로 다음 단계의 산출물(코드)을 기반으로 이전 단계의 산출물 추출해내는 기법이다. 이 방법은 구현으로부터 설계를, 설계로부터 요구사항으로 순공학적 어프로치와는 반대로 수행한다. 따라서 역공학 과점의 가시화 방법은 모든 프로세스가 하나로 집약되는 소스코드를 기반으로 품질, 코드의 구조, 코드로부터 설계 요소 추출, 코드의 복잡도 등을 가시화한다. 역공학관점의 가시화는 코드만 존재하고 있고 설계와 문서가 부재할 때 사용하면 더 유용하다.

순공학 프로세스에서 소프트웨어를 설계한 설계도면



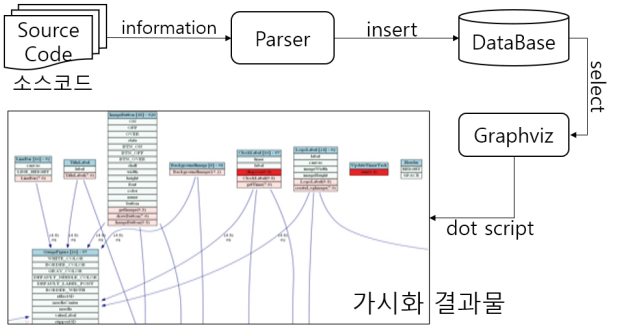
〈그림 7〉 역공학 기반 설계 검증

(A)와 역공학 프로세스를 통해 소프트웨어 Tool-chain을 통해 얻어진 설계도면(B)을 비교하여 설계 정확도를 검증한다. 두 설계의 비교를 통해 현재 설계(B)의 복잡도를 측정하고 설계를 개선해야할 부분을 확인할 수 있다. 이를 바탕으로 이상적인 설계도면(A)에 가까워지도록 소스코드를 변경하는 소프트웨어 리팩토링을 수행하여 설계 복잡도를 낮춘다. 결과적으로 '설계→구현→설계' 과정의 반복으로 소프트웨어 설계를 검증하여 완성도를 지속적으로 개선하여 소프트웨어 품질을 높인다. 〈그림 7〉은 소프트웨어 개발 프로세스 중 구현단계에서 작성한 소스코드 통해 역공학을 기반으로 설계도를 복원하여 검증 하는 프로세스를 나타낸다.

III. 블록체인 관련 코드

1. 블록체인

블록체인은 분산 컴퓨팅 기술을 기반으로 한 데이터 위, 변조 방지 기술을 말한다. 블록체인은 저장할 데이터를 블록 단위로 나누고 이를 체인으로 연결하여 저장한다. 이렇게 저장된 정보는 블록체인 네트워크에 참여하는 모든 구성원에게 공유되고 저장된다. 이를 통해 데이터를 탈중앙화 하여 저장하고, 데이터가 여러 곳에 분산 저장되어 있으므로 데이터의 위변조 또한 막을 수 있다. 또한 새로운 블록체인에 새로운 데이터를 기록함에 있어, 합의 알고리즘을 통한 구성원의 합의 이후 기록한다. 이를 통해 누군가가 악의적으로 데이터를 수정하는 것을 방지하여 데이터의 무결성과 신뢰성을 보장한다. 블록체인은 2009년 비트코인(Bitcoin) 개발을 시작으로 활성화되기 시작했다. 블록체인 기반의 암호 화폐 구현을 블록체인 1세대라고 칭하며, 대표적인 블록체인 네트워크로는 비트코인이 있다. 스마트 계약 구현을 통해 암호 화폐를 통한 거래를 자동화한 블록체인을 2세대라고 칭하며, 대표적으로 이더리움이 있다. 블록체인 2세대는 스마트 계약(Smart contract)을 통해 블록체인을 기반으로 하는 플랫폼을 구축했으며, 이러한 플랫폼에서 동작하는 애플리케이션이 탈중앙화 애플리케이션(dApp)이다. 블록체인 3세대는 기존 블록체인이 가진 단점들을 극복하고 실제



〈그림 8〉 Java 언어에 대한 코드가시화 기법

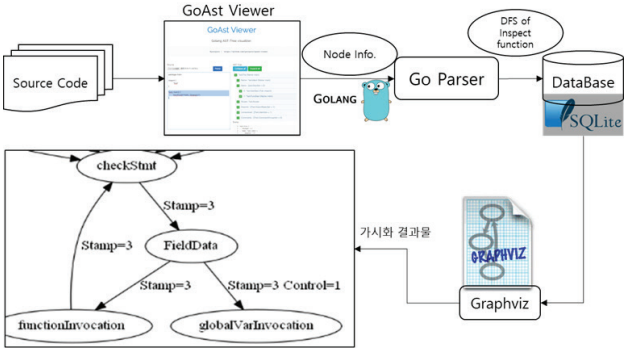
상용화하여 일상생활 속에서도 무리 없이 사용할 수 있는 블록체인 네트워크를 말하며 다양한 블록체인 네트워크들이 이를 목표로 발전 중에 있다.

2. 블록체인 코드 Java 언어

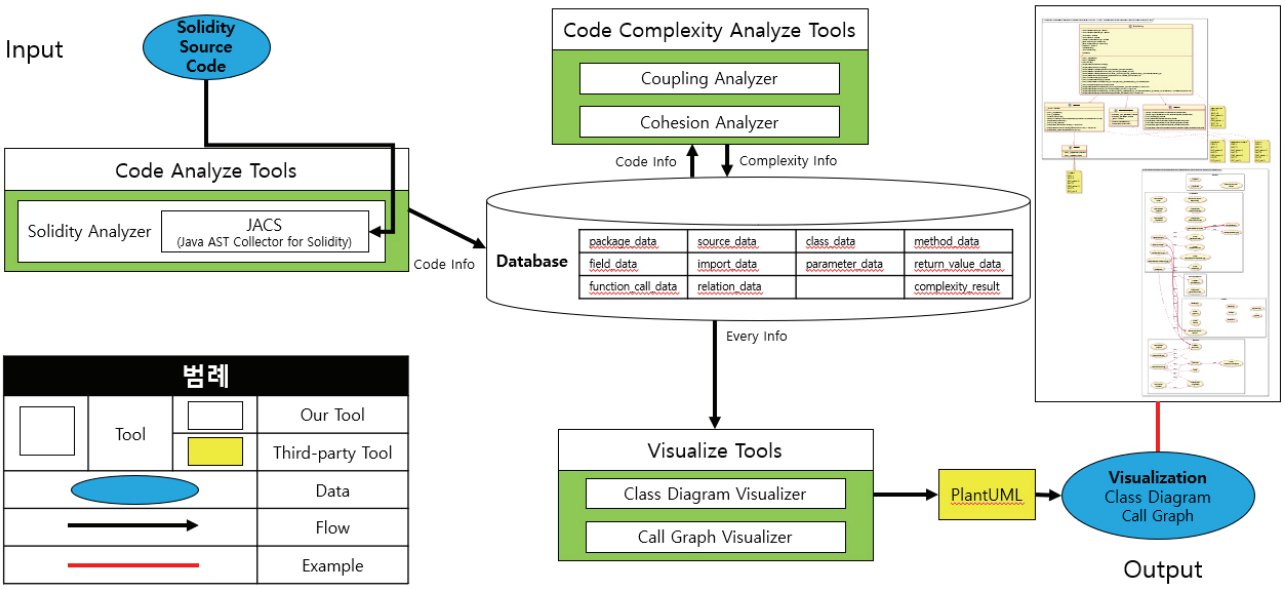
소프트웨어 가시화 방법을 Java에 접목하기 위한 방법으로 다음과 같은 코드 가시화 기법을 적용한다. 소스코드를 Java parser를 이용해 소스코드의 정보를 추출한다. 파서를 통해 추출한 정보들은 데이터베이스에 저장하여 저장된 정보들을 언제든지 다시 재사용할 수 있게 한다. 특정한 지표를 시각화하기 위해서 데이터베이스에 있는 정보들을 추출해 Graphviz라는 도구를 사용하여 dot script를 이용해 그래프 형태로 나타낸다[8].

3. 블록체인 코드 Go 언어

2009년 11월 구글에서 발표한 프로그래밍 언어로, 높은 생산성과 빠른 컴파일링, 실행속도를 자랑하는 Go라는 언어를 개발한다. 이후 이 언어는 Golang으로 불린



〈그림 9〉 Go 언어에 대한 코드가시화 기법



〈그림 10〉 Solidity에 대한 아키텍처 가시화 툴체인[12,14,15]

다. 이 언어는 컴파일언어에 속하지만 언어의 문법구조의 개선을 통해 빠른 컴파일리을 할 수 있는 언어로 개발되었다. 새로운 언어답게 코드는 매우 간결하며 쉽게 배울 수 있는 장점을 지녔다.

Go언어는 스마트 원장(Smart Contract)을 지원하는 언어로서 블록체인 기술을 개발 할 수 있다. 〈그림 9는 Go 언어에 대한 코드가시화 기법을 나타낸다. 소스코드로부터 GoAST Viewer를 통해 해당 소스코드의 AST를 추출한다. 추출한 AST정보는 파서를 통해 분석되며, 분석된 정보는 데이터베이스를 통해 저장하여 재사용한다. 데이터베이스에 저장된 정보는 다양한 가시화 도구(PlantUML, StarUML, GravphViz 등)을 이용해 시각화할 수 있다.

4. 블록체인 코드 Solidity 언어

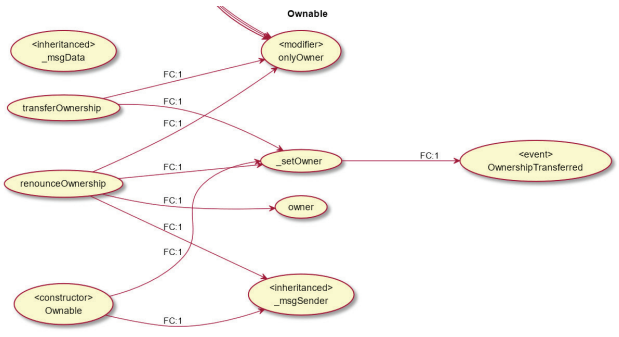
솔리디티는 이더리움 플랫폼에서 스마트 계약을 구현하는 언어이자 DApp에서 중추적인 역할을 한다. Solidity는 이더리움 플랫폼에 특화된 객체지향 프로그래밍 언어로서, 중간 언어인 Yul 언어를 거쳐 OPCODE로 컴파일된다. OPCODE로 컴파일 된 스마트 계약을 배포하면 이더리움의 계약계정(CA)에 저장된다. 솔리디티로 작성된 스마트 계약은 계약 그 자체로 동작하지 않고, 이더리움 네트워크 내부에서의 트랜잭션 혹은 이더리움 네트워크 외부에서의 원격 프로시져 호출(RPC)을 통해 동작한다[10, 11].

IV. 블록체인 코드 가시화 사례

〈그림 11〉은 Java 언어로 작성된 스마트 계약의 호출 그래프를 나타낸다. FC는 Function call을 의미하며, 소스코드의 함수가 다른 함수를 부르는 것들을 파악할 수 있다. 이런 정보들을 통해서 호출관계, 결합도, 응집도 등 Java언어에 대한 품질지표에 대해 파악할 수 있다.

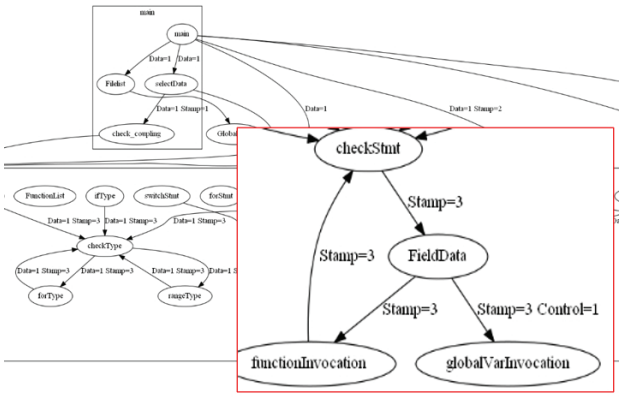
Case 2번은 Go언어를 분석하여 앞서 언급한 Go언어

Case 1: Java 블록체인 코드 가시화 사례



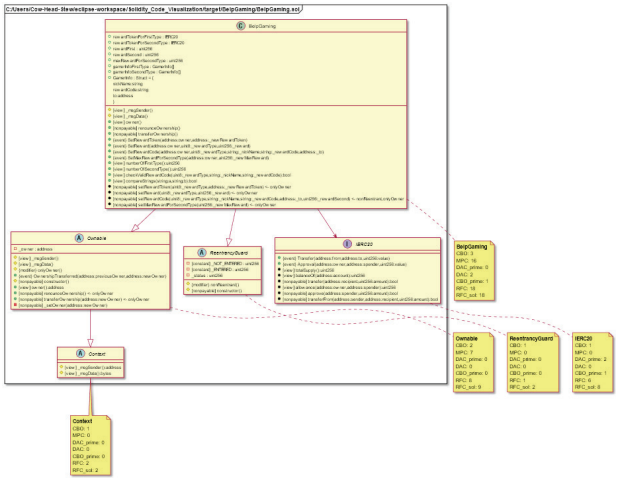
〈그림 11〉 Java 언어 코드 가시화 사례

Case 2: Go 언어 블록체인 코드 가시화 사례



〈그림 12〉 Go 언어 코드가시화 사례

Case 3: Solidity 언어 블록체인 코드 가시화 사례[12]



〈그림 13〉 Solidity 스마트계약 Class Diagram 추출[14]

가시화 툴체인에 소스코드를 입력하여 생성한 결과물을 나타낸다. 동그란 타원은 함수를 나타내며 함수가 다른 함수를 호출하는 관계에 대한 결합도를 점수로 표현한다.

〈그림 13〉은 Solidity로 작성된 예제 스마트 계약에 대해 Class Diagram 형태로 설계를 추출한 것이다. Class Diagram을 통해 계약의 상태를 저장하는 변수 및 계약의 함수, 이벤트 등의 동작 요소들을 확인 할 수 있다. 또한 각각의 계약에 대해 계산한 코드 복잡도를 표시해 줌으로서 계약이 얼마나 복잡한지 나타낸다.

솔리디티의 품질로서 주로 고려되는 사항은 크게 두 가지가 있다. 첫 번째로는 가스 소모량이 품질로서 고려되고 있다[13]. 이더리움 네트워크는 하나의 서버를 통해 동작하는 것이 아닌, 여러 개의 분산된 노드들을 통해 동작하는 분산 네트워크이다. 이러한 노드들 중 몇몇 노드는 채굴 노드로서 이더리움 네트워크에 계산 리소스를 제공하는 대가로서 수수료를 받는다. 스마트 계약을 배포 및 실행하는 경우 이러한 명령들에 대한 수수료로서 가스(Gas)를 소모하게 되고, 가스 소모량과 가스당 단가를 곱한 양의 코인을 채굴 노드에게 제공한다. 이러한 가스 소모량은 스마트 컨트렉트 및 DApp을 운영하기 위한 유지비가 되는 셈이다. 따라서 이러한 유지비를 감소시키기 위한 방법 및 코딩 패턴에 대한 연구가 진행되고 있다.

두번째로는 취약점이 품질로서 고려되고 있다. Solidity는 토큰 혹은 NFT 그리고 Ether와 같이 가치가 있다고 판단되는 재화를 직접적으로 거래할 수 있는 언어이다. 이러한 언어를 통해 작성한 프로그램에 존재하는 결함 및 취약점은 곧 계약의 배포자 및 계약에 참여한 당사자들의 금전적 손실로 이어질 수 있기 때문에 대부분의 계약은 Audit 서비스를 통해 취약점에 대한 검토 및 Audit 인증서를 발급받아 계약의 신뢰도를 보장 받는다.

V. 전망과 결론

블록체인과 그 기술에 적용하는 소프트웨어에 대한 관심이 증가하고 블록체인 기반 소프트웨어에 대한 개념이 변화되고 있다. 또한 블록체인에 대한 응용분야가 확대됨에 따라 이를 구현하고 탑재하는 소프트웨어에 대한 연구도 큰 관심을 받기 시작하고 있다.

블록체인 기술은 매우 그 출발이 매우 뛰어난 기술임에도 불구하고 여전히 가상 화폐나 다른 이슈들로 인해 그 빛이 가려져있다.

우리는 단순한 블록체인 기술의 발전이 아니라 블록체인 관련 코드(Java, Go, Solidity, Lua 등)에 대한 고품질화를 통한 신뢰성 있는 코드를 만들고자 한다. 현실은 새로운 언어에 대한 상용화된 정적/동적 분석기가 존재하지 않기 때문에 품질을 고려하기 어렵다. 이를 해결하기 위해 기존의 소프트웨어 가시화 기법을 블록체인 코드에 품질가시화를 위해 적용하는 방법이 있다.

그러한 기술에 기반이 되는 기술, 또한, 블록체인 기술

이 적용되어 함께 상생할 수 있는 기술의 발전에 집중해야할 때라고 생각한다. 이러한 기반 기술의 발전과 확대가 더 나은 블록체인 기술 발전과 더불어 세계를 선도하는 대한민국 형 블록체인 기술이 많이 보급될 수 있으면 하는 바람이며 이에 대한 연구와 지원이 확대되기를 기대한다.

참 고 문 헌

[1] 안현식. "블록체인코드 복잡도 분석을 위한 정적분석기 개발과 저전력 및 성능 개선 방법." 국내석사학위논문 弘益大學校 大學院, 2021. 서울

[2] 김장환. (2019). 블록체인 기반 시스템의 구조적 분석과 취약점 도출. 한국소프트웨어감정평가학회논문지, 15(1), 115-121.

[3] Brooks, F. P. Jr. (1987). "No Silver Bullet - Essence and Accidents of Software Engineering" (PDF). Computer. 20 (4): 10-19. CiteSeerX 10.1.1.117.315. doi:10.1109/MC.1987.1663532. S2CID 372277.

[4] http://www.redmine.or.kr/projects/community/project_dashboard

[5] Park, B.K.; Kang, G.-H.; Son, H.S.; Jeon, B.; Kim, R.Y.C. Code Visualization for Performance Improvement of Java Code for Controlling Smart Traffic System in the Smart City. Appl. Sci. 2020, 10, 2880. https://doi.org/10.3390/app10082880

[6] 문소영. "순/역공학 기반 비용 예측 및 자동 검증의 요구사항 프레임워크." 국내박사학위논문 홍익대학교 대학원, 2019. 서울

[7] Fowler, Martin. Refactoring: improving the design of existing code. Addison-Wesley Professional, 2018.

[8] 박지훈. "정적 분석 기반 블록체인 코드 취약성의 자동 가시화 연구." 국내석사학위논문 홍익대학교 대학원, 2019. 서울

[9] https://go.dev/

[10] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." white paper 3.37 (2014): 2-1.

[11] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.

[12] Chansol Park et al. "Design Validation Practices on Design Extraction of Solidity's Smart Contract Code based on Reverse Engineering". aacl17: IIBC, 2021, pp. 26-31.

[13] Janghwan Kim et al. "Best Practices on Improving Gas Consumption through Simplifying Quality Complexity of Solidity code for Smart Contracts in Distributed Network Environments". ICGHIT, 2022.

[14] 박찬솔 외. "Smart Contract Auditing을 위한 코드 복잡도 추출". 2022 종합학술대회 : 스마트미디어학회, 2022, pp. 63-65.

[15] Chansol Park et al. "Applying Code Visualization into Solidity for Auditing of Smart Contract". 2022 ICAEIC: ICT-Advanced Engineering Society, Vol.5 No.2, 2022, pp. 333-336.

김 장 환

- 2019년 12월 아이다호 주립대학교 학사
- 2022년 2월 홍익대학교 소프트웨어융합학과 석사
- 2022년 3월 ~ 현재 홍익대학교 박사과정

〈관심 분야〉
Software Visualization, Blockchain, Requirement Engineering, Web 3.0, Software Quality

박 찬 솔

- 2022년 2월 홍익대학교 소프트웨어융합학과 학사
- 2022년 3월 ~ 현재 홍익대학교 석사과정

〈관심 분야〉
소프트웨어공학, 블록체인, 객체지향 SW

장우성

- 2022년 8월 홍익대학교 전자전산공학과 석/박사
- 2011년 8월 ~ 2015년 1월 (주)한백전자 연구원
- 2015년 9월 ~ 2022년 6월 홍익대학교 강사
- 2022년 9월 ~ 현재 홍익대학교 외래교수

〈관심 분야〉
소프트웨어공학, 요구공학, 객체지향 SW, 인공지능 시스템, IoT/Embedded 시스템, 소프트웨어 모델링, 자연어기반의 테스트 케이스

김영철

- 2000년 2월 일리노이공대 박사
- 2001년 3월 ~ 2002년 2월 LG산전 부장
- 2002년 3월 ~ 현재 홍익대학교 교수

〈관심 분야〉
Software Visualization, TMM, Requirement Engineering.

문소영

- 2018년 8월 홍익대학교 전자전산공학과 석/박사
- 2007년 3월 ~ 2012년 2월 ㈜액트
- 2017년 12월 ~ 현재 NIPA SP 선임심사원
- 2019년 3월 ~ 현재 홍익대학교 초빙교수

〈관심 분야〉
Software Visualization, Requirement Engineering.


첨단기술로
더 나은 환경을 만듭니다
더 나은 미래를 열어가는 기술
환경문제를 해결하는 기술
반도체의 미래를 준비하는 일
지금, SK하이닉스가 하고 있습니다
We Do Green Technology
We Do Technology
첨단기술의 중심, 더 나은 세상을 만듭니다
SK hynix