AutoCon: Automated Smart Contracts Generation via GUI

Group Members:
Muhammad Abeer(P19-0061)
Aitzaz Tahir(P19-0012)
Najam Aqeel(P19-0035)

Supervisor:
Mr. Muhammad Amin

Table of contents

1. Literature Review

Literature Review

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[1]	2022	Recent Progress of Smart Contract in Blockchain	Data loading methods, and contract execution environment.	Increasing block capacity, directed acyclic graphs, sharding, and combination of zero knowledge proof technology.	Function variables and operation symbols out of bounds problem.
[2]	2022	Incorrect state are data collection and compilation. Data processing, and data misuse.	Confirmation of adding data on block. Also, longer chains issue.	Problems in LAS could be solved using DLT.	Distributed Ledger Technology (DLT).
[3]	2021	Decentralized storage of transactions, autonomous execution of contract codes, and decentralized establishment of the trust.	Identifying Semantic Flaws, Security Check Tools, Trusted Execution Environment (TEE).	The gap between human and smart contracts will be eliminated in future through mobility.	Destroyable Contracts, Exception Disorder, Call Stack Vulnerability.
[4]	2021	Analysis of the current state of research on smart contracts and identifying intellectual structures	Using exploratory factor analysis for co-citation analysis, six different strands of research are identified that concern technical, social, economic and legal disciplines	Structure overview of the main strand of research concerning smart contracts, their development overtime, the relevance of smart contract platforms in research and conceptual connections between publications and discourses are obtained.	N/A
[5]	2021	Challenges faced by developers in developing smart contracts	Interview, survey	Undesirable characteristics/challenges of Solidity language	Wrong conclusions may be drawn from interviews, survey respondents may have provided dishonest answers
[6]	2020	An overview on smart contracts: Challenges, advances, and platforms.	Stellar, Rootstock, and Hyperledger fabric.	Dynamic control flow, trustworthy oracle	Proliferation of smart contracts.
[7]	2020	Minimal transparency, accountability, incoherent data sets.	Ethereum blockchain	removal of middlemen/Brokers.	Certain government rules.
[8]	2020	Intermediaries that could be affected by blockchain protocols.	Heir functions and how can blockchain strengthen the security of these transactions while reducing their time. The author uses a legal methodology to approach it.	Permissioned blockchain controlled by public authorities.	The control of the parties' IDs. The legality of the contract and the verification and protection of rights in rem.
[9]	2020	Testing of Smart Contracts before deployment	sFuzz	Adaptive fuzzer for smart contracts	N/A

Literature Review Contd.

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[10]	2020	Towards automated verification of smart contract fairness.	FairCon	FairCon is effective in detecting property violations and able to prove fairness for common types of contracts.	Fairness Issues in smart contracts.
[11]	2019	Tailoring Gennaro	: Legal methodology to strengthen the security of transitions.	This protocol provides a versatile building block for a range of designs within and beyond the Ethereum ecosystem.	N/A
[12]	2019	Modeling and Verification of the Nervos CKB block Synchronization protocol in UPPAAL	CKB, Block synchronization protocol, UPPAAL	The Blockmaker Automation	UTXO id
[13]	2019	Decentralized storage of transactions, autonomous execution of contract codes, and decentralized establishment of the trust.	Identifying Semantic Flaws, Security Check Tools, Trusted Execution Environment (TEE).	The gap between human and smart contracts will be eliminated in future through mobility.	Destroyable Contracts, Exception Disorder, Call Stack Vulnerability
[14]	2018	Auto-Generation of Smart Contracts from Domain-Specific Ontologies and Semantic Rules	Ontologies and Semantic Rules.	Abstract syntax trees and neural networks as the widely used solutions	N/A
[15]	2018	Recursive calls attack solution	Hard fork	Mature Smart Contracts	Reentrancy vulnerability, Transaction-Ordering Dependence (TOD), Timestamp Dependence
[16]	2018	Decentralized (on-blockchain) and centralized (off–blockchain)	Rinkeby Ethereum	Hybrid architectures	Hybrid architectures are largely unexplored
[17]	2017	Cryptocurrency development	The consensus in the Ethereum network is based on modified GHOST protocol (Greedy Heaviest Observed Subtree).	Overcoming Bitcoin’s limitations.	Bitcoin scalability problem.
[18]	2017	Blockchain and Web3.0	Decentralized	No Border	Regulation is difficult
[19]	2016	Tackling security problem	Artificial Intelligence	Blockchain-based AI prediction	Blockchain-AI decentralized applications
[20]	2016	Propose a mapping that we operationalize using a domain-specific language in order to support the contract modeling process.	Automated Generation of Smart Contracts	Specially designed blockchain VM, called Ethereum Virtual Machine(EVM).	Is it possible to generate on the EVM machine code alone, without the availability of a high-level language such as Solidity?

2. Problem Statement

Problem Statement

Absence of an automated and efficient method for development of smart contracts prohibits the time conserving deployment which ensures the avoidance of steep learning curve

3. System Diagram

System Diagram

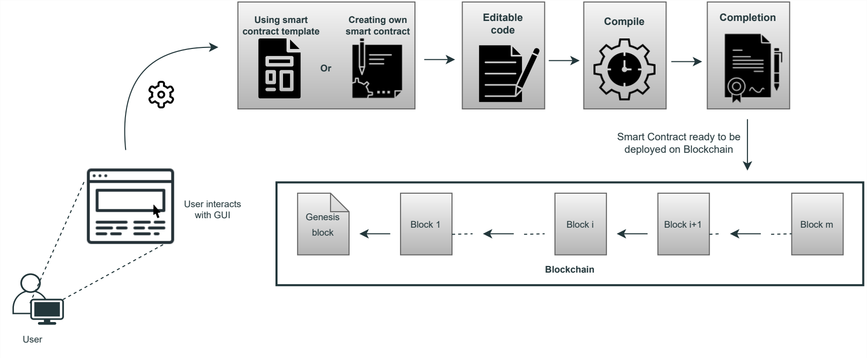

Figure 1: System Diagtam of AutoCon

4. UML Diagrams

Use Case Diagram

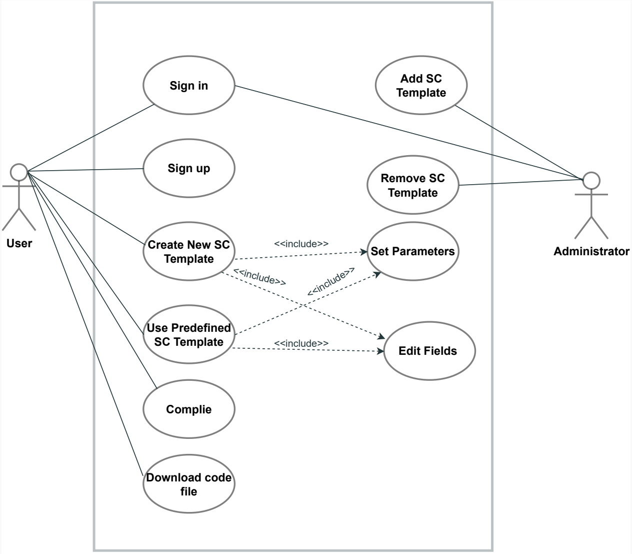

Figure 2: Use Case Diagram of AutoCon

Activity Diagram

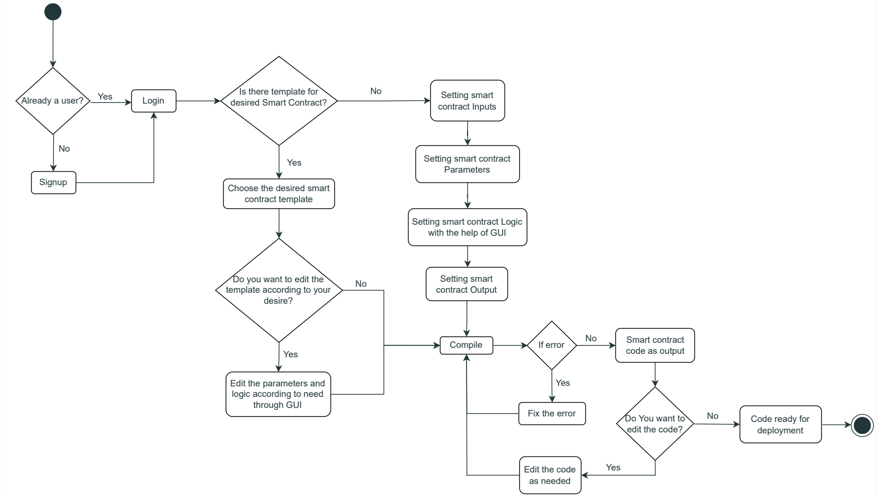

Figure 3: Activity Diagram of AutoCon

5. Objectives

Objectives

- To reduce *time and learning* curve.
- To create a *web2 platform* for smart contracts generation.
- To provide ease of use for end-user by developing a *generalizable* smart contract generator.

6. Expected Output

Expected Output

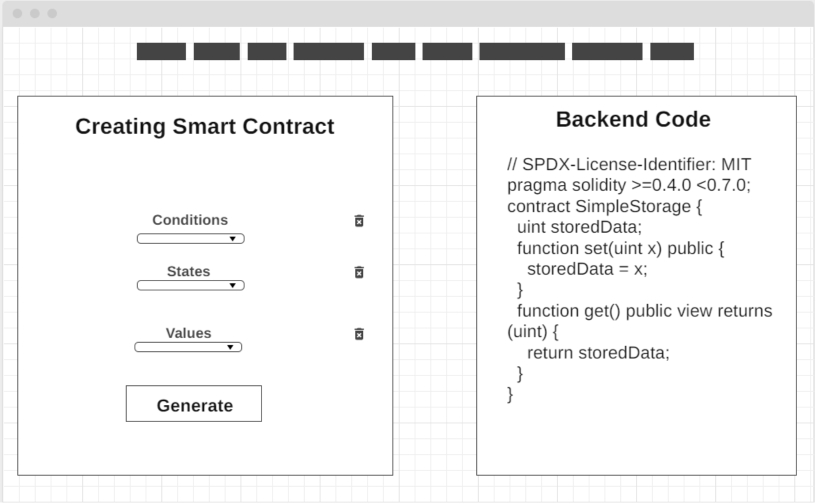

Figure 4: Expected Output

7. Gantt Chart

Gantt Chart

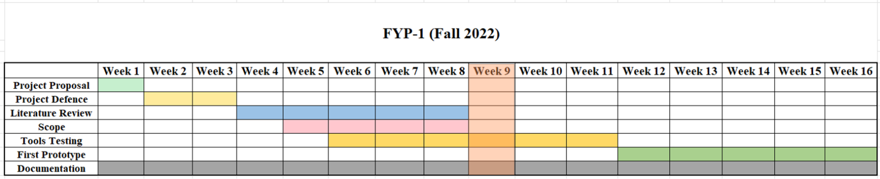

Figure 5: Gantt Chart

8. References

[1] Canghai Wu, Jie Xiong, Huanliang Xiong, Yingding Zhao, and Wenlong Yi. A review on recent progress of smart contract in blockchain. *IEEE Access*, 10:50839–50863, 2022.

[2] Miroslav Stefanovi, Dorde Prulj, Sonja Risti, Darko Stefanovi, and Danilo Nikoli. Smart contract application for managing land administration system transactions. *IEEE Access*, 10:39154–39176, 2022.

References [2]

[3] Tharaka Mawanane Hewa, Yining Hu, Madhusanka Liyanage, Salil S. Kanhare, and Mika Ylianttila. Survey on blockchain-based smart contracts: Technical aspects and future research. *IEEE Access*, 9:87643–87662, 2021.

[4] Lennart Ante. Smart contracts on the blockchain – a bibliometric analysis and review. *Telematics and Informatics*, 57:101519, 2021.

[5] Weiqin Zou, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Xia, Yang Feng, Zhenyu Chen, and Baowen Xu. Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering*, 47(10):2084–2106, 2021.

References [3]

[6] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:129–146, 2020.

[7] Archana Sahai and Rajiv Pandey. Smart contract definition for land registry in blockchain. In *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 230–235, 2020.

[8] Rosa M. Garcia-Teruel. Legal challenges and opportunities of blockchain technology in the real estate sector. *Journal of Property, Planning and Environmental Law*, 12:475–491, 2020.

References [4]

[9] Tai D. Nguyen, Long H. Pham, Jun Sun, Yun Lin, and Quang Tran Minh. Sfuzz: An efficient adaptive fuzzer for solidity smart contracts. ICSE ’20, page 778–788, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Ye Liu, Yi Li, Shang-Wei Lin, and Rong Zhao. Towards automated verification of smart contract fairness. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, page 666–677, New York, NY, USA, 2020. Association for Computing Machinery.

References [5]

[11] Philipp Schindler, Aljosha Judmayer, Nicholas Stifter, and Edgar R. Weippl. Ethdkg: Distributed key generation with ethereum smart contracts. *IACR Cryptol. ePrint Arch.*, 2019:985, 2019.

[12] Eranga Bandara, Wee Keong Ng, Nalin Ranasinghe, and Kasun De Zoysa. Aplos: Smart contracts made smart. In *International Conference on Blockchain and Trustworthy Systems*, pages 431–445. Springer, 2019.

[13] Florian Daniel and Luca Guida. A service-oriented perspective on blockchain smart contracts. *IEEE Internet Computing*, 23(1):46–53, 2019.

References [6]

[14] Olivia Choudhury, Nolan Rudolph, Issa Sylla, Noor Fairoza, and Amar Das. Auto-generation of smart contracts from domain-specific ontologies and semantic rules. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 963–970, 2018.

[15] Shuai Wang, Yong Yuan, Xiao Wang, Juanjuan Li, Rui Qin, and Fei-Yue Wang. An overview of smart contract: Architecture, applications, and future trends. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 108–113, 2018.

References [7]

[16] Carlos Molina-Jimenez, Ioannis Sfyrakis, Ellis Solaiman, Irene Ng, Meng Weng Wong, Alexis Chun, and Jon Crowcroft. Implementation of smart contracts using hybrid architectures with on and off–blockchain components. In *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)*, pages 83–90, 2018.

[17] Dejan Vujičić, Dijana Jagodić, and Siniša Ranđić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–6, 2018.

[18] Iuon-Chang Lin and Tzu-Chun Liao. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.*, 19:653–659, 2017.

[19] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 254–269, New York, NY, USA, 2016. Association for Computing Machinery.

[20] Christopher K. Frantz and Mariusz Nowostawski. From institutions to code: Towards automated generation of smart contracts. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 210–215, 2016.