Developing a Model-Driven Toolchain for the Formal Verification of DAOs

Valentini Simone[1], Avanzo Sowelu[2], Norta Alex[3], Schifanella Claudio[2], and Riccobene Elvinia[1]

[1]University of Milan
[2]University of Turin
[3]Tallinn University

June 18, 2025

Abstract

We present the research plan concerning the development of an integrated toolchain that facilitates the specification, development and verification of DAOs with suitable organizational structures. While existing approaches focus on some of the mentioned aspects in isolation, the complexity of the DAO domain requires an end-to-end toolchain supporting all the phases of the development lifecycle. Hence, we address this gap by first developing a formal verification approach based on Abstract State Machines. Subsequently, we automate the verification of DAOs and the generation of their smart contracts from visual models by means of a dedicated tool support. Unlike previous efforts, the proposed toolchain shall combine visual modeling with formal verification of governance properties of DAO systems.

1 Introduction

Ethereum [12] stands out as the first blockchain to put into practice the concept of smart contracts, originally envisioned by Nick Szabo in 1996. This technology combines a distributed ledger with the capacity to execute unchangeable code automatically when specific conditions are met, removing the need for a trusted third party. This combination has unlocked numerous potential new applications. Today, smart contracts are widely used to automate and enforce the terms of agreements between parties in various fields. These applications range from financial trades and services to insurance, credit authorization, legal processes, and decentralized organizations.

Decentralized Autonomous Organizations (DAOs) are a specific kind of Decentralized Application (DApp). They use smart contracts to help manage governance procedures. Creating DAOs is particularly difficult because designing and confirming the effectiveness and conformance of their token economies and governance systems is complex.

Our ongoing work involves investigating potential solutions to address the difficulties associated with designing and verifying DAOs. This document is structured in the following manner: Section 2 reviews the currently available solutions for DAO design and verification, outlining their difficulties and limitations. Following this, Section 3 outlines the immediate goals of this project. Lastly, Section 4 describes our objectives for the longer term.

2 Related works

Numerous studies aim to tackle the complexities involved in developing DAO smart contracts. They do this by creating graphical and textual Domain Specific Languages (DSLs). An example is IContractML 2.0 [8] However, many current DSLs lack formal verification methods, or they continue to be challenging for non-technical stakeholders to effectively use and understand.

Recent works focused on visual specification of the decentralized governance properties of DAOs. In particular, DECENT addresses the specification of decentralized governance processes [9] DAO-ML addresses the design of their organizational structures [2], and Extended T-DM [3] addresses the design of token economies of DAOs. Still, these nascent works lack formal verification capabilities.

Other approaches try to create DSLs and modelling languages that could be verified and vali-

dated. In particular, FsolidM/Verisolid [10] aims to provide a graphical DSL based on Final State Machines, it allows to visually or textually define the different states a smart contract can be in, the transitions between these states, the conditions (guards) that must be met for a transition to occur. Moreover it provides a verification mechanism based on the NuSMV [7].

However, the mentioned approaches lack suitability to verify the governance properties of DAO systems. Hence, we aim to address this gap in two steps, which we outline below.

3 Short-term Goals

Based on previous work on ASMETA [6] and on DAO-ML [2], we plan to enable verification of DAO-ML models and the corresponding smart contracts implementing the access control logic of a DAO.

This will be achieved by following the steps outlined below. First, we shall specify the properties DAOs modeled using DAO-ML should have to conform to security requirements. The security requirements of DAOs shall be elicited based on extant literature and system specifications. Subsequently, we plan to extend the existing library of attackers included in the ASMETA project [6] to verify the elicited security properties of DAOs modeled using DAO-ML. The work in [6] defines the model of the attacker performing for reentrancy attacks, known vulnerability of the infamous The DAO project, exploited in 2016. However, the framework currently lacks attacker models that address privilege escalations in the organizational structure of DAOs. This particularly concerns the need to ensure that, given a configuration of roles and permission assignments in a DAO, users in no occurrence can alter the organizational structure to obtain additional powers than the ones intended in the design phase. This aspect is particularly critical in the presence of complex DAO models specified using DAO-ML, such as those presented in [2]. These contain control relations that entail the capability of roles to add or remove subordinate roles or alter their permissions. The definition of dedicated attacker models shall facilitate the verification of the smart contracts generated by means of the translator from DAO-ML visual diagrams to Solidity, which was recently implemented. We see this as a necessary step towards a security aware model-driven approach for DAOs.

4 Long-term Goals

Following the completion of the short-term goals detailed in Section 3, our long-term aims to focus on supplementing these outcomes with developing a new textual and graphical notations of a DSL. This DSL will be defined for the specification DAO smart contracts in an independent paltfom format. More precisely, we intend to utilize frameworks designed for DSL development, such as Xtext [11] for textual languages or Cinco Cloud [4] for graphical languages.

Subsequently, to effectively verify that a model correctly adheres to its functional requirements, we aim to utilize a formal method capable of performing validation and verification. Specifically, we intend to implement an automatic translator. This tool will convert models written in from the DSL into Abstract State Machines (ASM) models [5] using the ASMETA [1] framework. This will enable us to employ the model checker, validator, and other tools included in the ASMETA toolset for rigorous analysis.

Finally, we plan to integrate the proposed translator into the recently implemented Cinco Cloud-based visual editor for DAO-ML models to provide an integrated toolchain that provides visual modeling, verification, and code generation capabilities. In summary, the long-term objectives include the following points:

- Implementing a DSL specifically for the creation of DAO smart contracts;
- Developing an automatic translator to convert a DSL model into an ASM model.
- Integrating the translator in existing tool support for DAO-ML.

References

[1] Asmeta - Overview — asmeta.github.io. https://asmeta.github.io/. [Accessed 17-04-2025].

[2] Sowelu Avanzo, Alex Norta, Julio Linares, Claudio Schifanella, and Marie Hattingh. Dao-ml: A modelling language for the specification of decentralized autonomous organization governance. *methods*, 13:16, 2024.

[3] Sowelu Avanzo, Alex Norta, Julio Linares, Claudio Schifanella, and Marie Hattingh. Extending trusted dapp modeling for decentralized autonomous organization development. In *Proceedings of the International Congress on Blockchain and Applications*, 2024.

[4] Alexander Bainczyk, Daniel Busch, Marco Krumrey, Daniel Sami Mitwalli, Jonas Schürmann, Joel Tagoukeng Dongmo, and Bernhard Steffen. Cinco cloud: A holistic approach for web-based language-driven engineering. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Software Engineering*, pages 407–425, Cham, 2022. Springer Nature Switzerland.

[5] Egon Börger and Robert Stärk. *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer Verlag, 2003.

[6] Chiara Braghin, Elvinia Riccobene, and Simone Valentini. Modeling and verification of smart contracts with abstract state machines. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pages 1425–1432, 2024.

[7] Alessandro Cimatti, Edmund M. Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. NuSMV 2: An OpenSource Tool for Symbolic Model Checking. In *Proceedings of the 14th International Conference on Computer Aided Verification*, CAV '02, pages 359–364, Berlin, Heidelberg, 2002. Springer-Verlag.

[8] Mohammad Hamdaqa, Lucas Alberto Pineda Metz, and Ilham Qasse. icontractml: A domain-specific language for modeling and deploying smart contracts onto multiple blockchain platforms. In *Proceedings of the 12th System Analysis and Modelling Conference*, pages 34–43, 2020.

[9] Fadime Kaya, Francisco Perez, Joris Dekker, and Jaap Gordijn. Decent: A domain specific language to design governance decisions. In *International Conference on Research Challenges in Information Science*, pages 603–610. Springer, 2023.

[10] Anastasia Mavridou, Aron Laszka, Emmanouela Stachtiari, and Abhishek Dubey. Verisolid: Correct-by-design smart contracts for ethereum. In *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers*, page 446–465, Berlin, Heidelberg, 2019. Springer-Verlag.

[11] Miro Spoenemann. Xtext - Language Engineering Made Easy! — eclipse.dev. https://eclipse.dev/Xtext/. [Accessed 17-04-2025].

[12] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.