Ontological Smart Contracts in OASIS: Ontology for Agents, Systems, and Integration of Services

Domenico Cantone, Carmelo Fabio Longo, Marianna Nicolosi-Asmundo, **Daniele Francesco Santamaria**, Corrado Santoro

Department of Mathematics and Computer Science, University of Catania

16 September, 2021

Introduction

- The blockchain is a peer-to-peer public ledger maintained by a distributed network of computational nodes.
- One of the most important features of a decentralized and publicly shared ledger is the elimination of any third-party intermediaries, since they require clients to put total and unquestioned trust on them.
- The blockchain guarantees the ownership,transparency, traceability, public availability, continuity, and immutability of digital assets, in an efficient and trust-less environment where censorship is hardly achievable.

- One of the most popular applications of the blockchain is a self-executable contract, also called smart contract (SC).
- The smart contract is a way of representing contracts into lines of immutable program codes which are allowed to be self-run on a public ledger.
- Smart contracts are opening new perspectives in several key aspects such as Internet of Things (IoTs), healthcare, insurance, energy, communications, and robotics.
- Smart contracts are used to managed tokens, digital certificates stored on the blockchain representing predetermined rights on certain assets.

- Towards Semantic Blockchains.
- A formal semantic knowledge representation capturing blockchain and smart contracts
  - facilitates the understanding of blockchain concepts,
  - the interlinking with other out-of chain information,
  - enables the automatic discovery of smart contracts.
  - realizes semantic blockchains:
    - A desirable feature of token exchange systems is a precise and intelligent query mechanism capable of determining what, when, and how certain assets have been generated, exchanged or destroyed.
- Semantic web tools and languages aim to reach full machine interoperability, to promote common data formats, to exchange protocols on the web, and to share and reuse data across applications and across enterprise and community boundaries.
- We focus on OWL 2 ontologies.

- Many works aim to to provide ontologies for blockchain contexts, in particular:
  - Blockchain Ontology with Dynamic Extensibility (BLONDiE) project (Ugarte Rojas, 2017) provides a comprehensive vocabulary that covers the structure of different components of three main blockchains, Ethereum, Bitcoin, Hyperledger.
  - Ethon ontology (Pfeffer et al., 2016), providing a semantic interpretation of smart contracts as services.
- The main limitation of the approaches is the poor semantic description of smart contracts, thus hindering the discovery of unknown smart contracts and of the related operations fulfilled during their life-span.

- We extend the ontology **OASIS**, *Ontology for Agents, Systems, and Integration of Services* (Cantone et al., 2019) to semantically represent blockchain, smart contracts, and tokens through Web Ontology Language 2 (OWL 2) ontologies.
- Two steps are required:
  - Extending OASIS with conditionals and ontological smart contracts (OSCs). Conditionals allow one to restrict and limit agent interactions, define activation mechanisms that trigger agent actions, and define constraints and contract terms on OSCs. OSCs are representations of smart contracts that allow to establish responsibilities and authorizations among agents and set agreements. **(In this work)**
  - Extending OASIS with definition of blockchain and the operational semantics of smart contracts, limited to the case study of the Ethereum ERC721 standard protocol for managing non-fungible tokens (NFTs). **(In the next talk)**

OASIS

- OASIS models (web) agents and, in particular, the way they interact and operate in a collaborative environment, regardless of the framework and language adopted for their implementation.
- Agents are mainly represented by means of the mentalistic notion of agent behavior inspired by (Bresciani et al., 2004), encompassing goals and tasks that are achievable by the agent, together with actions, sensors, and actuators used to perform operations.
- OASIS is used to define actions that may be requested to other agents and their related information such as operation inputs and outputs. Such requests are submitted by exchanging suitable fragments of OASIS, whereas agents whose capabilities are compatible with the requested actions are discovered by means of SPARQL queries performed over their behaviors.

- OASIS was applied to build a TRL3 prototype of a home assistant that activates and manages applications, devices, and users interacting with each other within the environment (Cantone et al., 2019).
- OASIS is now part of the project POC4COMMERCE for the NGI ONTOCHAIN European project https://ontochain.ngi.eu/.

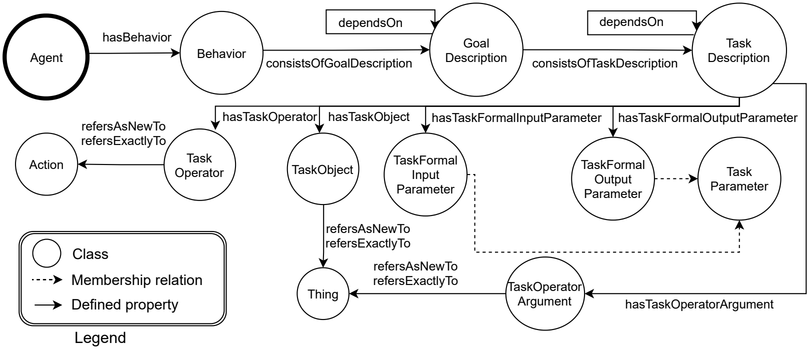

Figure 1: OASIS agent behavior schema

Agents are represented through three steps: a) behavior templates, b) concrete behavior, c) agent requests and agent actions.

OASIS conditionals

Conditional are constituted by a consequent (head) and an antecedent (body), both formed by a conjunctive set of atoms. Atoms in their turn comprise the subject of the conditional, the object, the operator describing the action and, possibly, an operator parameter and argument.

OASIS conditional

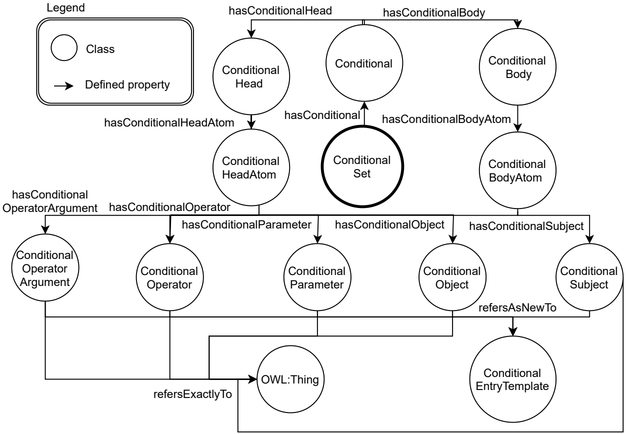

Figure 2: OASIS conditional schema

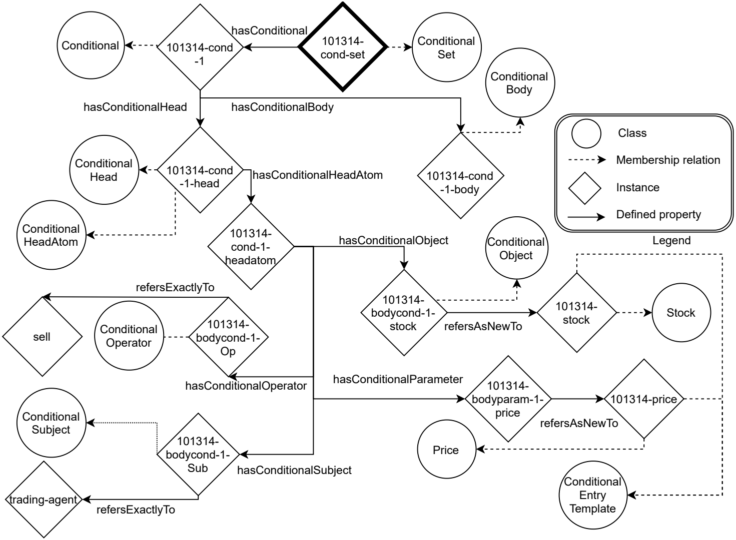

Figure 3: OASIS conditional example

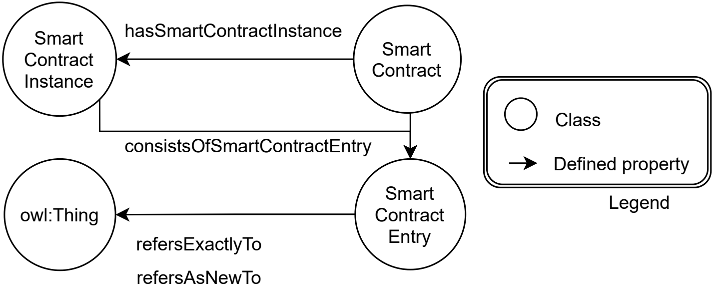

Figure 4: OASIS OSC schema

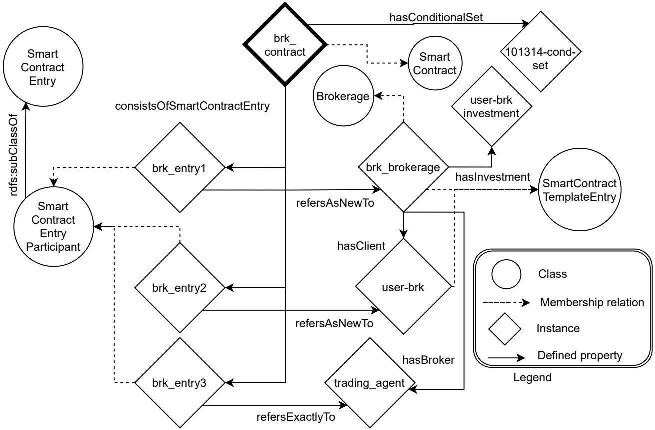

Figure 5: OASIS OSC example

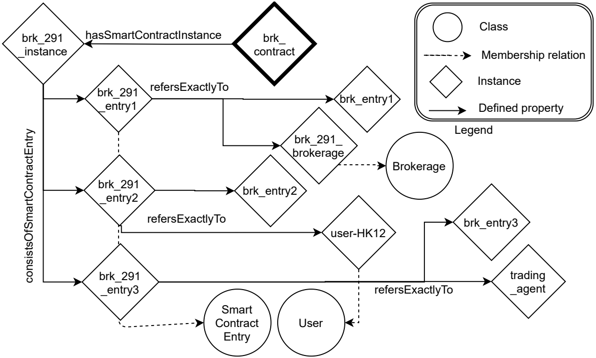

Figure 6: OASIS OSC instance example

- Storing ontologies inside blockchains is unfeasible.
- Ontologies can be stored on IPFS and managed through a suitable Ethereum smart contract.

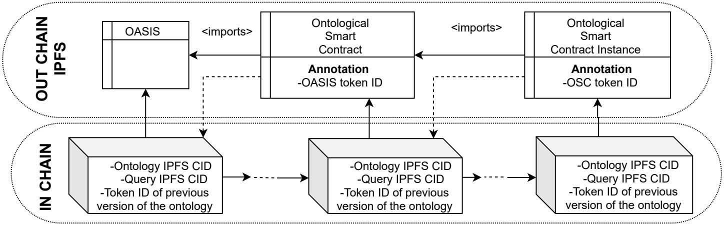

Figure 7: Sketch of the architecture of an OASIS OSC-based application

- The smart contract is reachable at 0x36194ab80f7649572cab9ec524950df32f638b08 of the Ethereum main network.
- A Java API to publish and retrieve OSC is available at https://github.com/dfsantamaria/ProfOnto/tree/master/profonto/src/main/java/dmi/unict/it/osc/core (check the readme for the latest version)

References

- (Ugarte Rojas, 2017), A more pragmatic web 3.0: Linked blockchain data, in Google Scholar.
- (Pfeffer et al., 2016) Ethon - an ethereum ontology (2016), available on-line: https://ethon.consensys.net/index.html.
- (Cantone et al., 2019) Towards an Ontology-Based Framework for a Behavior-Oriented Integration of the IoT, Proc. of the 20th Workshop From Objects to Agents, 26-28 June, 2019, Parma, Italy, CEUR Vol. 2404, pp. 119–126, 2019.
- (Bresciani et al. 2004), Tropos: An agent-oriented software development methodology. Autonomous Agents Multi Agent Systems 8(3), 203–236 (2004).