Towards Autonomous Organizations

[DRAFT]

Jubin Jose
hello@josejub.in
linkedin.com/in/jubin-jose-dev

Introduction

Autonomous Organizations are one of the near future promises of current technology and is a buzz word in the field of Decentralized networking and Blockchain oriented Smart Contracts. But when we look closer and analyze the details of it, we see that the technology requirement for engineering Autonomous Organizations is not only limited to aforementioned fields of technology but also many other fields including Artificial Swarm Intelligence, Internet of Things etc. We also see that, the terminology itself is not limited to closed organizations that are controlled by shareholders but also an open network of tiny organizations that are governed by simple rules written by individuals contributing to self identifying, communicating and organizing swarm of collective intelligence. We believe that Web 3.0 and advancements in Artificial Intelligence will be the greatest influencers of this. We observe that the progress towards Autonomous Organizations has already been started and will improve slowly and will win by adapting to and improving iteratively the existing technology, especially the Web. This approach will also ensure the gradual switching of end users without even knowing the technical complexities during the progress.

One of the main challenges that we face today is the lack of proper standards to achieve this goal of building the infrastructure. Currently being at the bleeding edge of all potentially contributing technologies, it is not possible to define a final standard for everything. So in this paper we are going to look into proposed, the high level building blocks of the infrastructure. And we will eventually break them down into smaller components by consulting current technologies and will suggest improvements to be made. We hope that each of them can eventually be standardised with the help of community and keeping open to the community.

Related Work [edit in progress]

One of the oldest remarkable works in Multi-Agent Systems (MAS) is done by the Foundation for Intelligent Physical Agents (FIPA). Which formed in 1996 as an independent Swiss based organization later merged into IEEE CS standards organization. FIPA specifications were originally written for the Multi-Agent communications on the Web. Due to the eventual inactivity of the organization, the specifications are currently inactive and mostly outdated. However the specifications (which are made public and still bound to undisclosed patent restrictions) at its core provide enough inspiration to start building MAS related systems.

FIPA specifications cover three low level modules - Agent Communication, Agent Management, Agent Message Transport upon which a MAS can be built. Agent communication details the data structure of both the message being sent between two agents in a network and the shared Ontology between two agents in communication. A message being exchanged is the fundamental element of MAS which facilitates a Transaction between two agents. Ontology brings in context and background knowledge which enables self identification between two agents. Ontology plays an important role to bring meaning and mutual understanding between agents to accomplish a goal. Agent Management covers agent management services, agent management ontology and agent platform message transport. This specifies a model for the creation, registration, location, communication, migration and retirement of agents, which can be condensed down into an agent lifecycle supported by agent location service (Directory Facilitator). Agent Message Transport describes on-wire protocol for message exchange. We are not interested in Agent Message Transport now because it is outdated and there are better methods to replace it.

Recent advancements in efforts to build Knowledge Graphs over the semi / fully structured Web and Connected Devices led to two popular open standards - Schema.org (RDF is another popular standard) and W3C WoT specification. With which any agent (human or machine) can identify, interpret the ontology and contents of a web oriented site / app or a connected device and interact with it directly without a third party translation.

Schema.org solves the problem of ontology to a large extent for MAS. Adapting the Web to it will eventually create a planet wide, dynamic Knowledge Graph (it is like a database) that any agent can access at any time. It is up to the agent to decide which part of it to be accessed at a time (localize). Today, it is mostly available as embedded information within rich documents. The scope of Schema representations is not limited to documents but to the APIs as well, to make avail ontology description by an agent at a location in the Web.

W3C WoT on the other hand solves the problem of self identification (primitive ontology) and message exchange (Transactions) within a limited scope - electronic devices connected to the Web. It allows the agent to first retrieve the device information - to understand the capabilities of that device (ontology) and then read / update the device parameters or run actions on behalf. Mozilla's adaptation of this specification includes message exchange details as well.

Autonomous organization of virtual agents:

- FIPA (Multi-agent communication standards)
    - Generalization: http://www.fipa.org/subgroups/ROFS-SG-docs/2007-TAAS-specifying-MAS.pdf
    - By spec. status: http://www.fipa.org/repository/standardspecs.html
    - By spec. subject: http://www.fipa.org/repository/bysubject.html
- Web of Things (agents identify others and communicate - API designs)
    - W3C standards: https://www.w3.org/standards/
    - Mozilla's adaptation API spec: https://iot.mozilla.org/wot/
- Categorizing the web (semantic representations (metadata) of objects and services for machine readability and understanding):
    - Google's schema.org: https://schema.org/
    - More: https://en.wikipedia.org/wiki/Metadata_standard

High Level building blocks of Autonomous Organizations [edit in progress]

- Trust, Identity, Transactions (blockchains)
    - Hyperledger for permissioned actions: https://www.hyperledger.org/
    - Etherium for smart contracts: https://www.ethereum.org/
- Decentralization, network latency and clustering
    - Universal protocol stack for decentralized communication: https://ipfs.io/
- Modern engineering (how to engineer, refine above standards for modern web iteratively)
    - Query one API endpoint (very important): https://graphql.org/
    - Couch replication protocol - eventual consistency: http://docs.couchdb.org/en/stable/replication/protocol.html

- Conflict resolution by design - CRDTs: https://en.wikipedia.org/wiki/Conflict-free_replicated_data_type

Building blocks in detail [edit in progress]

FIPA Agent communication specs

- Message structure spec
Parameter	Category of Parameters
performative	Type of communicative acts
sender	Participant in communication
receiver	Participant in communication
reply-to	Participant in communication
content	Content of message
language	Description of Content
encoding	Description of Content
ontology	Description of Content
protocol	Control of conversation
conversation-id	Control of conversation
reply-with	Control of conversation
in-reply-to	Control of conversation
reply-by	Control of conversation

    Table 1: FIPA ACL Message Parameters
- Ontology service spec (refers to scheme.org or WoT description)
  - Agents that communicate together should be aware of the Ontology - capabilities of each of them

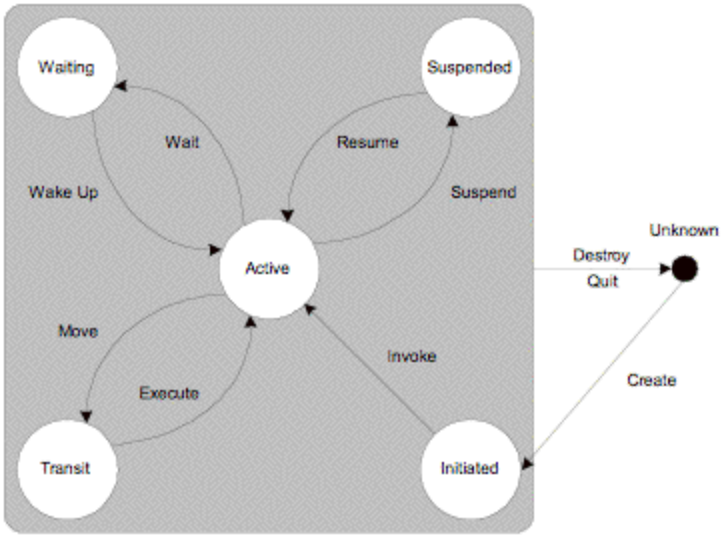
- Ontologies of agents can be related in multiple ways

Extension	When O1 extends the ontology O2
Identical	When the two ontologies O1 and O2 are identical
Equivalent	When the two ontologies O1 and O2 are equivalent
Weakly-Translatable	When the source ontology O1 is weakly translatable to the target ontology O2
Strongly-Translatable	When the source ontology O1 is strongly translatable to the target ontology O2
Approx-Translatable	When the source ontology O1 is approximately translatable to the target ontology O2

Table 1: Ontology Relationship Levels

  - It is common and good engineering practice to build a new ontology by extending or combining existing ones.
  - Translation between Ontologies is necessary in this case. More: http://www.fipa.org/specs/fipa00086/XC00086D.html

FIPA agent management spec

- Covers: agent management services, agent management ontology and agent platform message transport
- It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agents.
- Directory Facilitator - a directory listing for agents with UID and nickname
  - API: register, deregister, modify, search, get-description agents
  - Federated Directory Facilitators: network of DFs
  - 

agent lifecycle

FIPA Architecture

- http://www.fipa.org/specs/fipa00001/SC00001L.html
- Agents & services - core components. A service can be an agent or an RPC function
- Agent & service directory service (indexing) Data structure: [name, type, locator]
- Agent message structure & encryption

FIPA Applications & Examples

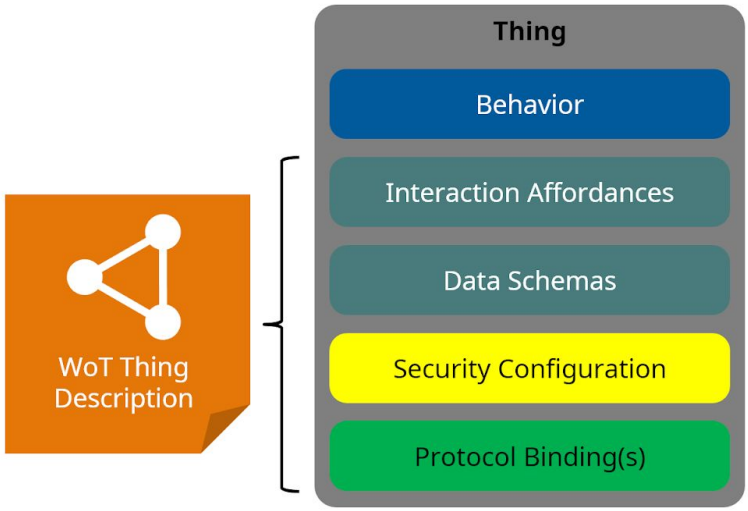
- http://www.fipa.org/repository/applicationspecs.php3
- Ontology is defined in action level. Do we need an Ontology defined this way? Or in a very basic form like Schema.org?

W3C web thing description

- Property, Action, Event - Interaction affordances
- Each Interaction affordance is located with a URI
- Description of a thing: https://www.w3.org/TR/wot-thing-description/#thing

W3C web thing architecture

- A web thing identifies itself with thing description
- A thing can be standalone or linked with other things. The thing description will contain links to others if there's any. There can be terminating or intermediate things as well. (https://www.w3.org/TR/wot-architecture/#sec-architecture-overview)



-
- The Interaction Model of W3C WoT introduces an intermediate abstraction that formalizes the mapping from application intent to concrete protocol operations and also narrows the possibilities how Interaction Affordances can be modeled.
- Each interaction model in detail: https://www.w3.org/TR/wot-architecture/#sec-interaction-model
- Communication patterns between things: https://www.w3.org/TR/wot-architecture/#sec-wot-servient-architecture-high-level

Mozilla WoT

- Contexts: http://iotschema.org/, https://iot.mozilla.org/schemas

User experience and Value generation ecosystem [edit in progress]

- User experience
    - A virtual assistant per person
    - No more knowledge aggregation in one place - like Alexa, Google Assistant, Siri but a swappable assistants, like you can use any web browser to get the same experience
    - Any capability of Assistant is plug-in - which includes AI (cognitive) services
- Privacy
    - Data stays on device
    - Assistants see data on device and behave dynamically (intelligence downloaded to device)
- Value and ecosystem
    - Addons generate value
    - Search service, app store