

Towards Autonomous Organizations

[DRAFT]

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
- Trust, Identity, Transactions (blockchains)
 - Hyperledger for permissioned actions: <https://www.hyperledger.org/>
 - Ethereum 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
- 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

POINTS:

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

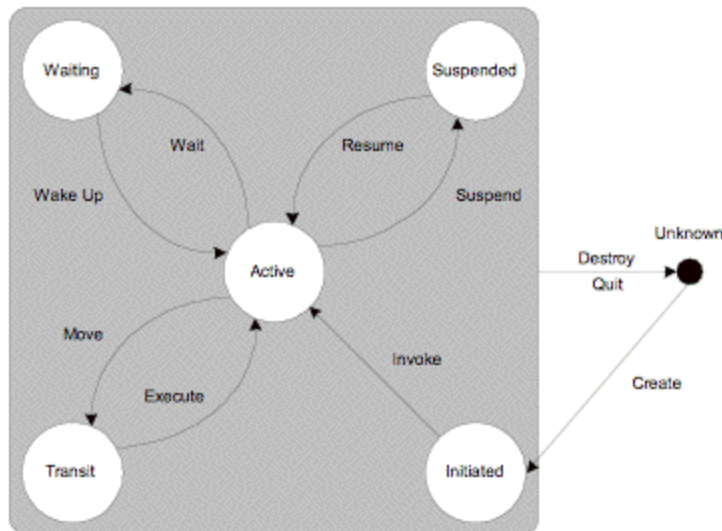
Extension	When O1 extends the ontology O2
Identical	When the two ontologies O1 and O2 are identical
Equivalent	When the two ontologies O1 and O2 are equivalent
Weakly-Translatable	When the source ontology O1 is weakly translatable to the target ontology O2
Strongly-Translatable	When the source ontology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

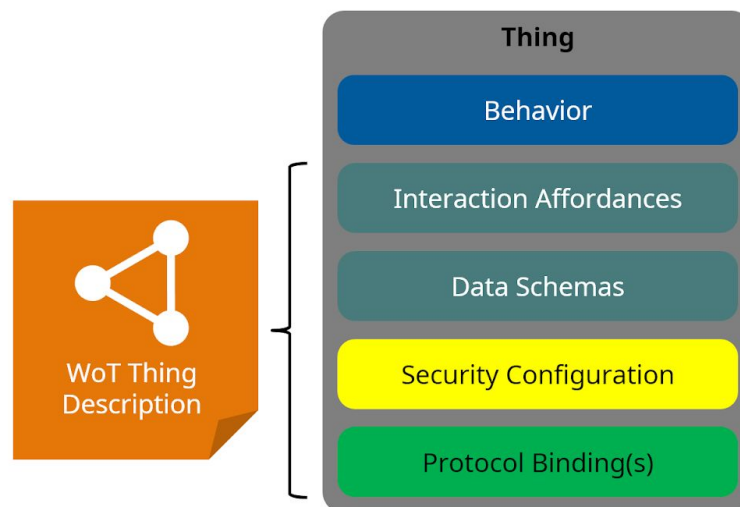
- <http://www.fipa.org/repository/applicationspecs.php3>
- Ontology is defined in action level. Do we need an Ontology defined this way? Or in a very basic form like Schema.org?

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](https://www.w3.org/TR/wot-architecture/#sec-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](https://www.w3.org/TR/wot-architecture/#sec-interaction-model) 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>