作为多智能体系统的自动驾驶网络

Autonomous Network as an AI-Driven Multi-Agents System

Min He

2020/08/10 V1

Introduction

AND Core Idea:

Single Domain Autonomy， Multi-domain Orchestration 分域自治， 多域协同

- 什么是自治 (autonomy）？
- 什么是自治域（autonomous domain）？
- 和现有的cloud native 微服有什么关系?
- 如何协同体现群体智能?
- 如何从现有架构迁移到未来适合于“分域自治， 多域协同”的AND架构?
- AI是必须的吗?
- 现有的研究成果能给ADN建设带来什么启发和借鉴?

。。。

Outline

Introduction 引言

- Definition of Autonomous System, Agent, Autonomous Domain
- Autonomous Agents in Microservice Architecture

Agent and Architecture 代理及架构

- Types of Agent
- Reactive Agent Architecture
- Deliberative Agent Architecture
- Hybrid Agent Architecture

Building MAS 开发多智能体系统

- AOP (Agent Programming Language)
- JADE (Java Agent Development Framework)
- WADE (Workflow and Agent Development Framework)

Agent Communications 代理通信

- ACL (Agent Communication Language)
- FIPA-ACL Performatives
- FIPA Predefined Communication Protocols
- Ontology

Summary 总结

AI and MAS

Autonomous System，Agent，and Autonomous Domain 自治系统，代理和自治域

AS (Autonomous System 自治系统) able to make decision based on environment change without human intervention.

- perception 感知
- inference 推理
- decision/action 决策/行动

Agent 自治代理 is used to represent an autonomous actor on behalf of someone or something in an autonomous system which serves a specific purpose. Agent has following characteristics:

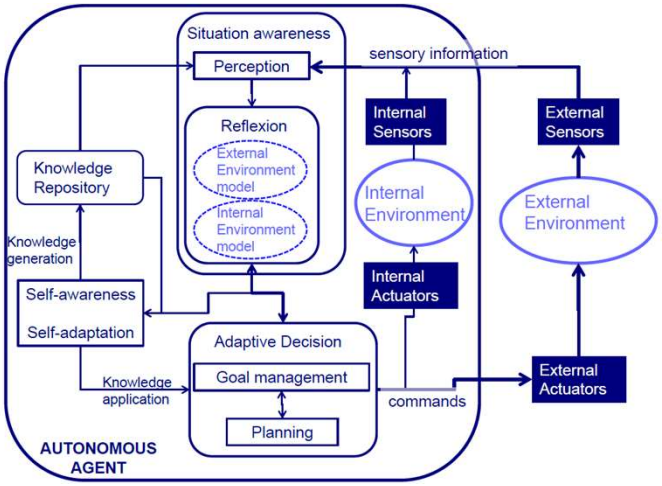
- Only exists with respect to an environment or context. 某种环境中
- Is purposeful. 有目的
- Is autonomous and always acts on its own local, partial view of the world. 局部视野
- Is always on. 总是醒着

Autonomous Domain 自治域 is a complete system that can accomplish difficult tasks or goals in a specific domain with behaviors and responses that display a form on intelligence. It comprises a collection of intelligent agents normally gathered in a hierarchical structure containing many "sub-agents". Intelligent sub-agents process and perform lower level functions. 一系列子自治系统构成能完成复杂操作的智能体系。

Multi-agent System (MAS) 多智能体系统 is one that consists of a number of agents which interact with each other to completed the complex task by: 能完成复杂任务通过多个智能体

- cooperation 合作
- coordination 协调
- negotiation 谈判

Typical Autonomous System



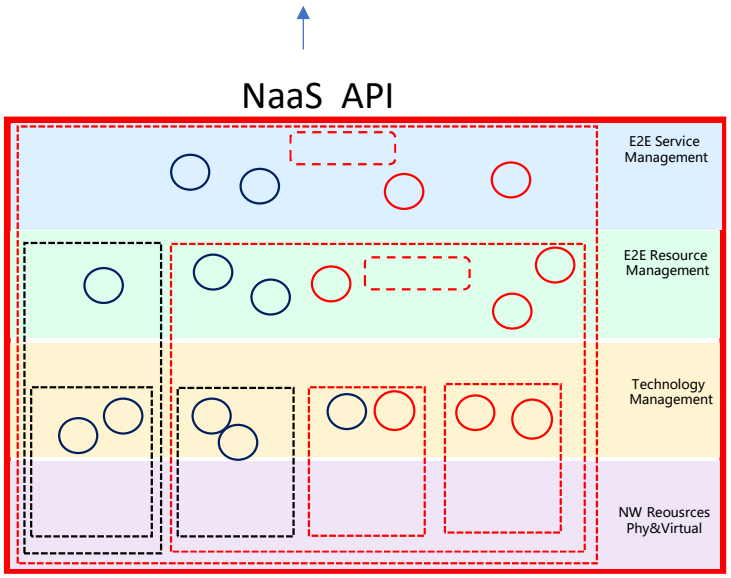
AS:

different shapes and forms
different intelligent levels
different size

In this discussion:

AS = Agent
Autonomous Domain = MAS

Autonomous Agents in Microservice Architecture 在微服架构中的自治代理



AOMS --- Agent-Oriented Microservice

POMS --- Plain-Old Microservice

AD --- Autonomous Domain,

PD --- Passive Domain

MAOCH --- Multi-Agent Orchestrator

An autonomous system or an agent is AOMS (Agent-oriented Microservice).

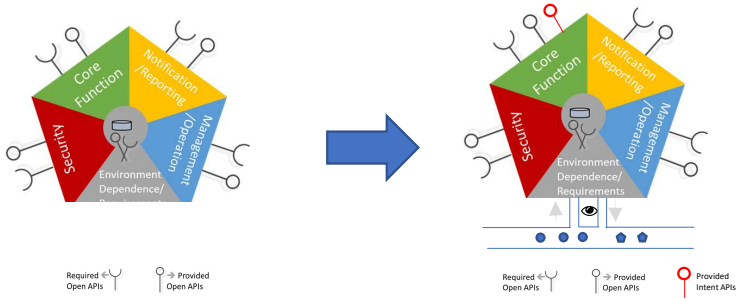
AD (Autonomous Domain) is a Multi-Agent Microservice（MAMS）自治域就是一个多智能体微服

compare POMS and AOMS based on Intelligent MAS criteria

Principles	POMS	AOMS
Autonomy 自治能力	operate without the direct intervention of humans and have some kind of control over their internal state (and actions?).	Agents operate without the direct intervention of humans and have some kind of control over their actions and internal state
Social Ability 社交能力	Interaction between microservices is typically achieved using messages based on RESTful API	interact with other agents using some kind of Agent Communication Language.
Reactivity 反应能力	Respond to incoming HTTP requests in a timely fashion.	perceive their environment and respond in a timely fashion to changes that occur in it.
Proactivity 主动性	do not take the initiative.	don't just respond - they take the initiative

Plain-old Microservice

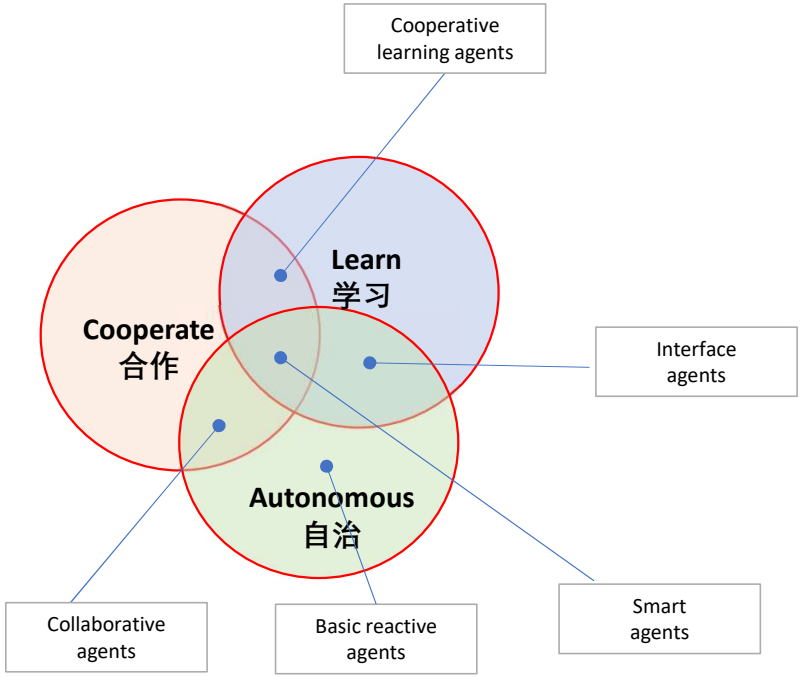
Agent-oriented Microservice



Example: make a POMS to AOMS by adding sensing capability and intent interface
加入环境感知和意图API把POMS变成AOMS

Types of Agents 代理类型

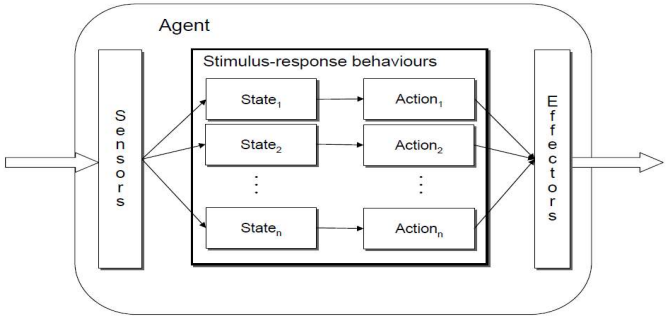
Software Agent Primary Attributes 代理主要能力



Nwana HS, Wooldridge M. Software Agent Technologies. BT Technology Journal, vol. 4, 1996.

- Basic Reactive Agent (Reactive agent) 简单反应式的
  - Reactive to environment change
  - Not much collaboration and learning capability
- Intelligent Agent (Smart Agent) 智能
  - Autonomy 自治能力
  - Reactivity 反应能力
  - Proactivity 主动能力
  - Social Ability 社交沟通能力
- Cooperative leaning agents 协作学习式
  - Responsible for collect info from environment and other agents
  - Also known as information/internet agent.
- Interface Agent 界面式
  - Learn from user, environment and other agent
  - Provide suggestion autonomously
  - Limited reasoning
  - Like personal assistance
- Collaborative Agents (Deliberative Agent) 协同式
  - Take actions after collaborate with other agents.
  - Strong reasoning
  - Limited learning capabilities
- Holonic Agent 整体
  - Comprised of sub-agents.
  - Represent holon to the outside world
  - Sub-agents still have their own goal, not not conflict with holon goal.

Reactive Agent Architecture 反应式代理架构



- Simple internal model of the world 没有太多模型
- Tight coupling of perception and action 条件反射
- Behavior-based paradigm 基于行为的交流
- Intelligent is the product between the agent and its environment. 智能体现域和环境的互动
- Focused on fast reactions/responses to environment changes。 着重于快速反应

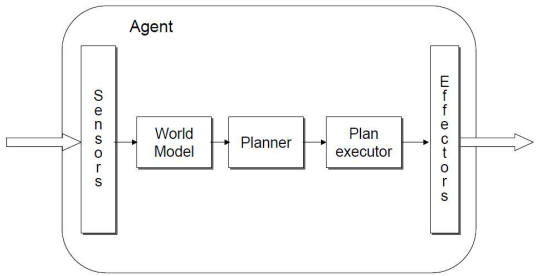
Advantages: 优点

- Simplicity of individual agents。 简单
- Flexibility, adaptability 灵活
  - Ideal in very dynamic and unpredictable environments
- Computational tractability 计算量小
  - Avoiding complex planning/reasoning procedures
  - Avoiding continuous model update
- Robustness against failure 容错率强
  - No central planning component
- Elegance

Limitations: 局限

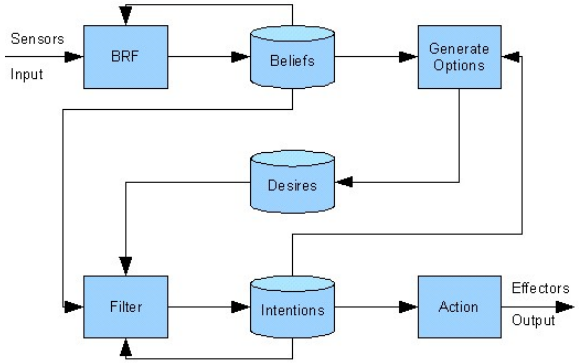
- Short-term view 短视
- Difficult to have “non-local” information. 没有全局关
- No long-term planning capabilities。 没有长期计划
- Not much learning capability 学习能力弱
- Difficult to map global goal to individual agent’s capability. 不容易把要到达到的目标映射成个头的行为
- Limited applicability 应用有限
  Games, simulations, basic robots (insects)

Deliberative Agent Architecture 规划式代理架构



- Explicit symbolic model of the world 有表达环境的模型
- Decisions are made via logical reasoning 逻辑推理
- Sense-plan-act problem-solving paradigm. 感知-计划-行动
- Focused on long-term planning of actions, centred on a set of basic goals 注重基于目标制定长期行动计划

BDI (Believe-Desire-Intent) Reasoning Model 推理模式

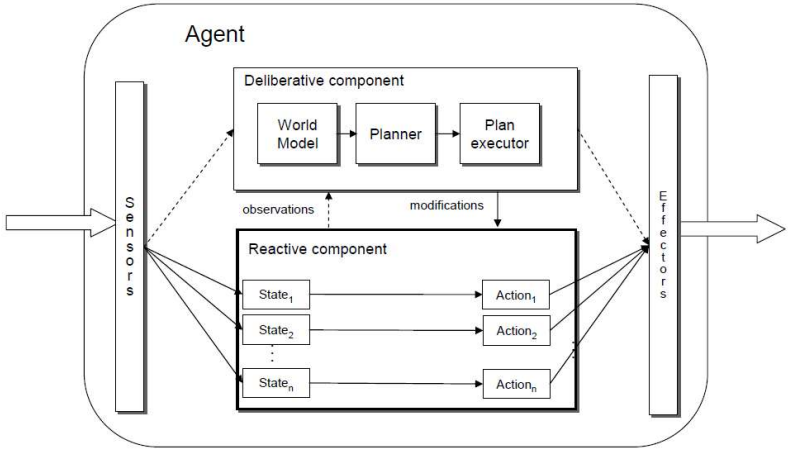


BDI based Agent-Oriented Programming JACK, 2APL, JADEX.. Tools available

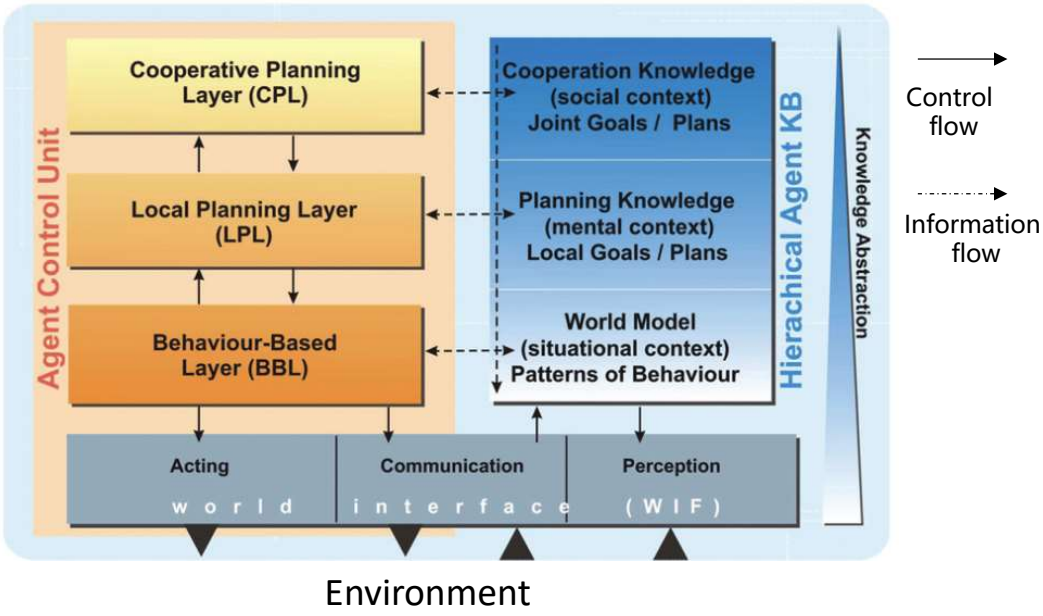
Deliberative Agent Architecture Limitation: 局限性

- Dynamic world 需要不断更新模型
  - Update symbolic world model
  - World changes while planning is being done
- Representation language 对表达语言要求高
  - Expressive enough to be useful in any domain
  - Limited enough to be computationally tractable
- Classical planning => complete, optimal solutions 对算力要求高，反应慢
  - High computational cost
  - Sometimes a sub-optimal low-cost fast reaction can be effective

Hybrid Agent Architecture 混合式代理架构



- **A deliberative one**, containing a symbolic world model, which develops plans and makes decisions in the way proposed by symbolic AI 。 计划式负责推理计划
- **A reactive one**, which is capable of reacting quickly to events without complex reasoning 反应式负责快速反应。
- **Hierarchy,** with higher layers dealing with information at increasing levels of abstraction。多层架构
- In 2005, Stanly, autonomous car won DARPA’s Grand Challege with hybrid agent architecture. 用此架构斯坦福自动驾驶汽车赢DARPA的大奖



A. Muller InteRRaP vertical two passes hybrid agent architecture
纵向双路混合代理架构

AOP (Agent-Oriented Programming) Language 面向代理的编程语言

- For building MAS.用于构建MAS
- AOP paradigm, the core is “agent” as “object” in OOP. 核心是代理
- Introduced by Yoav Shoham in 90s. 90年代引入
- Heavily influenced d by BDI and reactive planning system。深受BDI推理模式影响
- New concepts:
  - mental categories, reactivity, proactiveness, concurrent execution, meta-level reasoning, agent communications.
- Varies high level abstract defined by language construct:
  - Agent: Belief, Goal, Intention, Plan
  - Organization: Group, Role, Norm, Interactions
  - Environment: Artifacts, Percepts, Actions

Agent-Oriented Software Engineering (AOSE) Methodologies (Prometheus, Gaia, Tropos..) define best practices to build MAS software product.面向代理软件方法论

	OOP	AOP
Basic unit	object	agent
Parameters defining state of basic unit	unconstrained	beliefs, commitments, capabilities, choices, ...
Process of computation	message passing and response methods	message passing and response methods
Types of message	unconstrained	inform, request, offer, promise, decline, ...
Constraints on methods	none	honesty, consistency, ...

OOP

1. abstract class
2. Class
3. member variable
4. Method
5. collaboration (uses)
6. composition (has)
7. inheritance (is)
8. instantiation
9. polymorphism

AOP

1. generic role
2. domain specific role
3. Knowledge, belief
4. Capability
5. Negotiation
6. holonic agents
7. role multiplicity
8. domain specific role + individual knowledge
9. service matchmaking

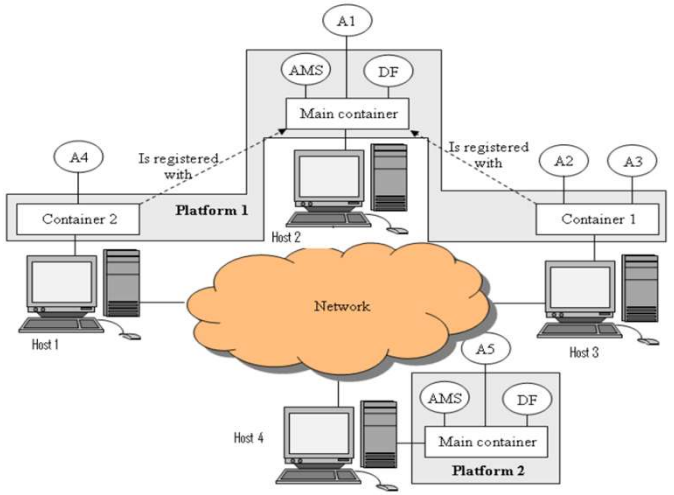
JADE (Java Agent DEvelopment Framework)

JADE (https://jade.tilab.com/) is:

- Open Source MAS Implementation Framework developed and distributed by Telecom Italia. Latest version 4.5.0 released in 8/6/2017.LGPL V2 license. 开源项目
- IEEE FIPA Compliance 。 基于FIPA标准
- Design, runtime, debug and deployment设计，运行，调试和发布环境
- Distributed 分布式
- Pure Java based. Java
- Peer to Peer Communication model。 对等你通信模式

JADE provides: 提供

- Runtime environment 运行环境
- A library of classes to develop an agent。开发代理库
- Remote Agent Management GUI。 远程管理界面



JADE application is composed of **Agents**, Agents lives in **containers**. A collection of containers forms a **platform. 代理在容器里，容器在平台力**

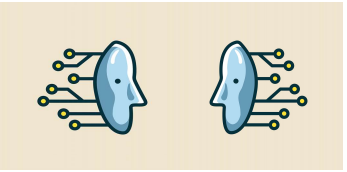
Each platform has one **Main Container. Main Container 每个平台有一个主容器。**

- first to start
- Other container register themselves with the main container.
- **AMS** (Agent Management System) for naming services and **DF** (Directory Facilitator) provides yellow pages service

Agent communication 代理之间使用FIPA-ACL通信

- **Any agent can communicate with any agents regardless containers and platform via asynchronous mode.**
- **Message format is defined by ACL language of FIPA.**

ACL (Agent Communication Language)代理通信语言



- Agent to express its intent to another agent.
- Based on "speech act theory" (言語行為理論) developed by Searle in 60s. Conversation can be expressed in 6 categories:命令,要求,祝愿 / 致歉,询问 / 祈使,邀请,感叹.

- Popular ACLs based on speech act theory:
  - FIPA-ACL
  - KQML (Knowledge Query and Manipulation Language)

- **FIPA** (Foundation for Intelligent Physical Agents), part of IEEE standard committee.

- **Message format:** defined by FIPA
  - The Sender
  - The List of receivers
  - The communicative intention (aka the performative)
  - The content
  - The content language
  - The ontology
  - conversation-id, reply-with, in-reply-to, reply-by for concurrent conversation control

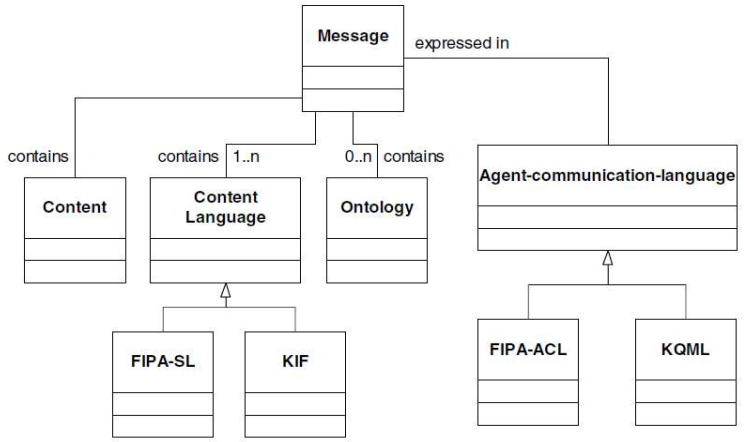


Figure 15: UML - **Message** Elements

Inform message example

```
(inform
                :sender   (agent-identifier:name i)
                :receiver (agent-identifier:name j)
                :content  "door (now, open)"
                :language prolog)
                :ontology house
)
```

FIPA-ACL Performatives 行动说明

FIPA-ACL performatives

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

FIFA-ACL performatives Examples

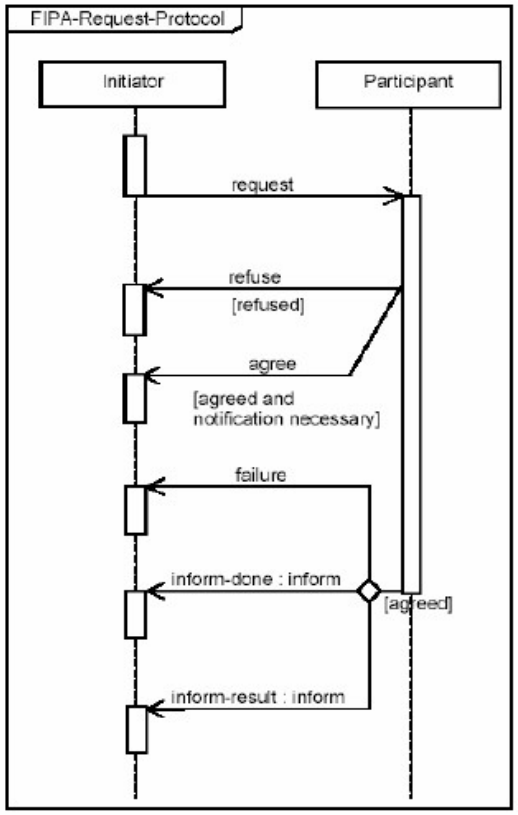
query	Is the door open?
request	Open the door (for me)
agree	OK! I’ ll open the door
inform	The door is open
failure	I am unable to open the door
refuse	I don’ t want to open the door
subscribe	Tell me when the door becomes open
cfp	Does anyone want to open the door?
propose	I can open the door for you... at a price
not-understood	Door? What is that? Do not understand...

...

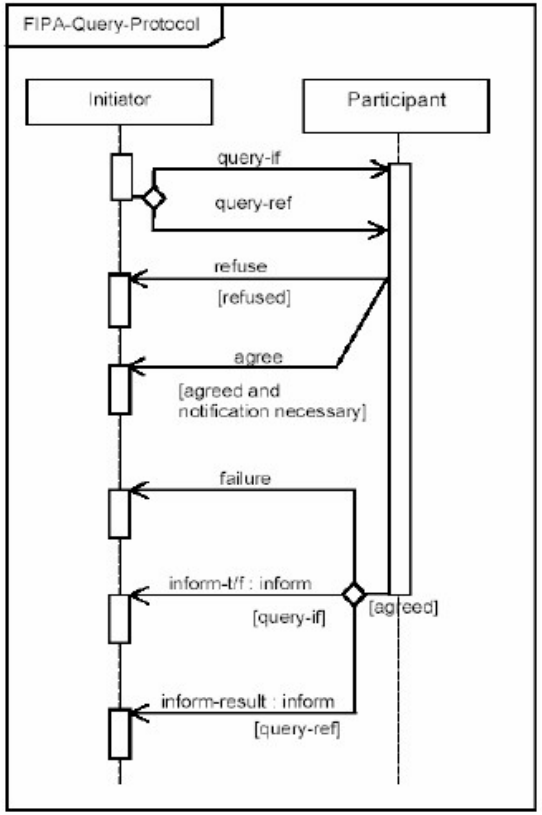
http://www.fipa.org

Predefined Communication Protocols 通信协议

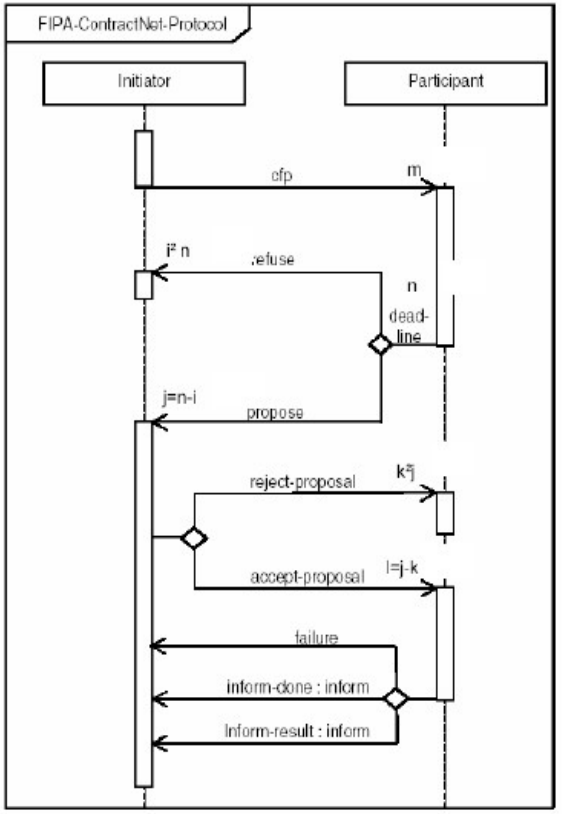
FIPA Request Protocol



FIPA Query Protocol



FIPA Contract Net Protocol



Ontology 本体

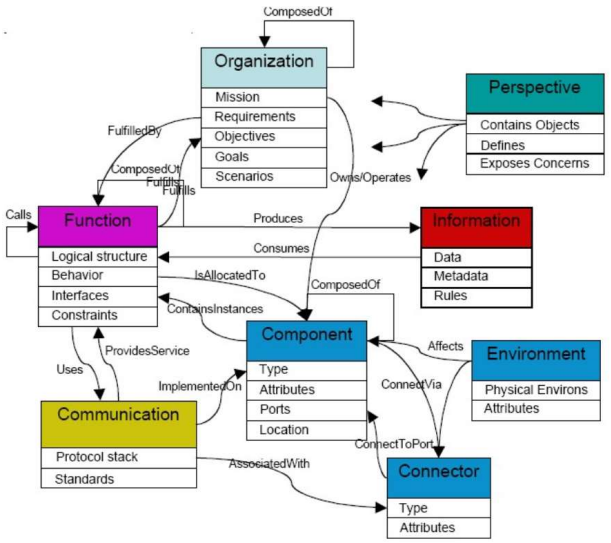
What is ontology? 什么是本体

- An ontology is an explicit description of a domain 描述某领域的
  - Concepts 概念
  - Properties and attributes of concepts 属性
  - Relations 关系
- An ontology defines 本体定义了
  - A common vocabulary 共同术语
  - A shared understanding among a set of agents. 共享知识

Why we need ontology? 为什么需要本体

- To share a common understanding of structure of information
  - Among people, among agents 代理之间共享认知
- To make domain assumptions explicit 明确表达
- To enable to reuse of domain knowledge 允许知识重复使用

Ontology example



ontology + data = knowledge graph
知识图谱是本体的具体化

WADE (Workflows and Agents Development framework)

WADE is JADE platform with workflow capability extension。 JADE的工作流延申

Workflow 工作流

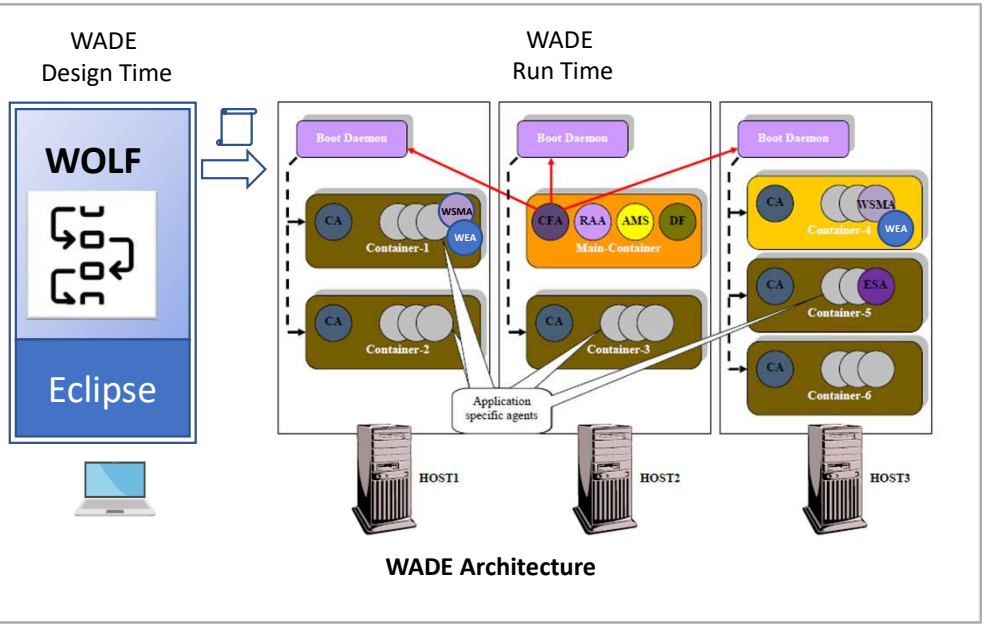
- Execution steps as well as their sequencing are made explicit 执行步骤
- Graphic representation 图形表达
- Potential for businesspeople to define 商业人士可以定义
- Use embedded Micro workflow engine in each agent。使用微工作流引擎。
- Requestor agent “launch” workflow defined in actuator agent.
- WOLF provide graphical workflow interface。 WOLF提供工作流设计界面

Agent Pool 代理池

- Manage a collection of the same agents.
- Agent pool created agent but not deployed to containers. RAA responsible for deploy to container based on policy

Agent and container fault tolerance 提供代理和容器的高容错

- JADE already provide agent and container fault tolerance at platform level.
- WADE provides fault tolerance at application level. CA (control agent) automatically restarting all agents that suddenly disappear due to HW or SW. leader CA responsible to restart the whole container

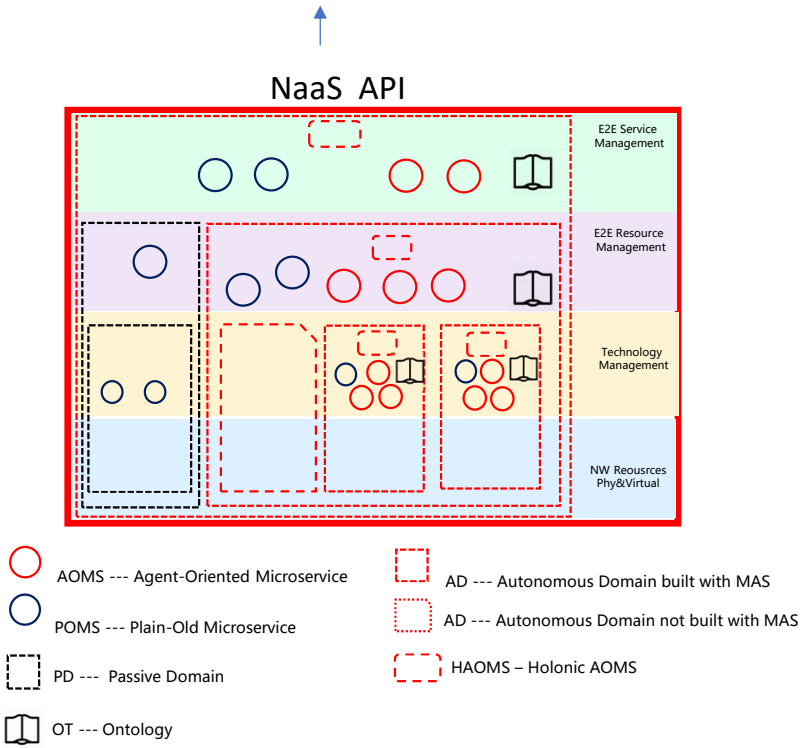


WADE Architecture

RAA -- Runtime Allocation Agent
WSMA -- Workflow Status Manager Agent.
ESA -- Event System Agent . This agent can be allocated in whatever container and implements the WADE Event System that allows a workflow to suspend waiting for an event.
WEA -- Workflow Engine Agent..
WOLF -- WOrkflow LiFe cycle environmentle management. Eclipse plugin for workflow creation GUI.

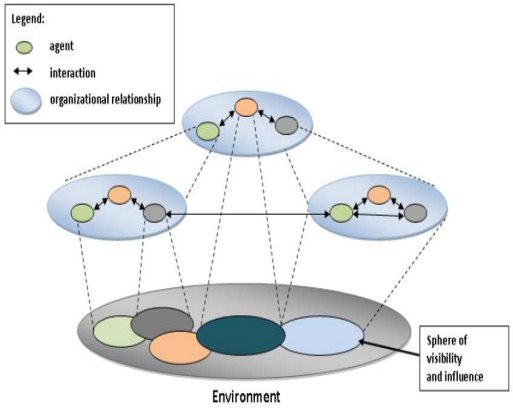
https://jade.tilab.com/wade/doc/WADE-User-Guide.pdf

ADN as a MAS System 按多智能体系统方法构建ADN



AD (Autonomous Domain) can be built with or without MAS technology， but if follows “single domain autonomy， multi domain orchestration” principle，， overall AND， as a NaaS provider， likely be a Hierarchical Holonic MAS。 自治域可以用MAS方法也可以不用MAS方法构建， 但如果遵循“单域自治，多域协同”的原则，AND作为有自治能力的NaaS功能提供者面对的是一个典型的分层整体MAS问题。MAS的方法论可以借鉴：

- Utilizing Hybrid Agent Architecture 采用代理混和架构
- Container-based and Distributed 基于容器的分布式布局
- Hierarchical Ontology System 分层次的本体模型（知识库）
- Agent Communication use ACL（broader than Intent API）定义代理通信语言而不是简单的“Intent API”
- Follow AOSE （Agent-Oriented Software Engineering）methodologies and tools 借鉴AOSE方法和工具



CANONICAL VIEW OF AN AGENT-BASED SYSTEM [JEN00].

代理系统的规范视图

Table 1. Lifecycle and Agent Mapping of AOSE Methodologies

S.NO	METHODOLOGIES	LIFECYCLE	AGENT MAPPING
1.	Prometheus	Analysis Design Implementation Debugging	Goal Plan
2.	MaSE	Analysis Design	Role
3.	Tropos	Analysis Design Implementation GOST	Actor Goal
4.	GAIA	Analysis Design	Role
5.	SODA	Analysis Design	Role
6.	PASSI	Analysis Design Implementation Unit Testing	Role
7.	ODAM	Analysis Design Implementation	Role
8.	MESSAGE	Analysis Design	Goal Role
9.	ROADMAP	Analysis Design Implementation Testing	Role
10.	MAS CommonKADS	Analysis Design	Goal

AOSE 方法流派

Summary 总结

- 构建AN系统需要从新的角度来思考因为：
  - 自治域是有自动性的， 自治带来了不确定性。
  - 自治域之间如何组织，协同， 交流， 竞争都是需要重新考虑。
  - 如何将独立有自己小目标的自治域整合成能完成复杂任务的大目标， 通过以往的“协同层”的方法不一定合适。
  - 需要从面向对象的转换到面向智能代理的架构思维。
- AN系统是一个典型的MAS系统。
- 90年代开始的MAS （Multi-Agents System）技术对AND系统的构建有非常好的借鉴作用。 这个领域都MAS的方方面面都有一定的探索结果和工具。
- 建议用AOSE （Agent-Oriented Software Engineering）方法论来审视TMF TA的思路和产品开发思路。

AOSE 专题图

AI/ML and MAS

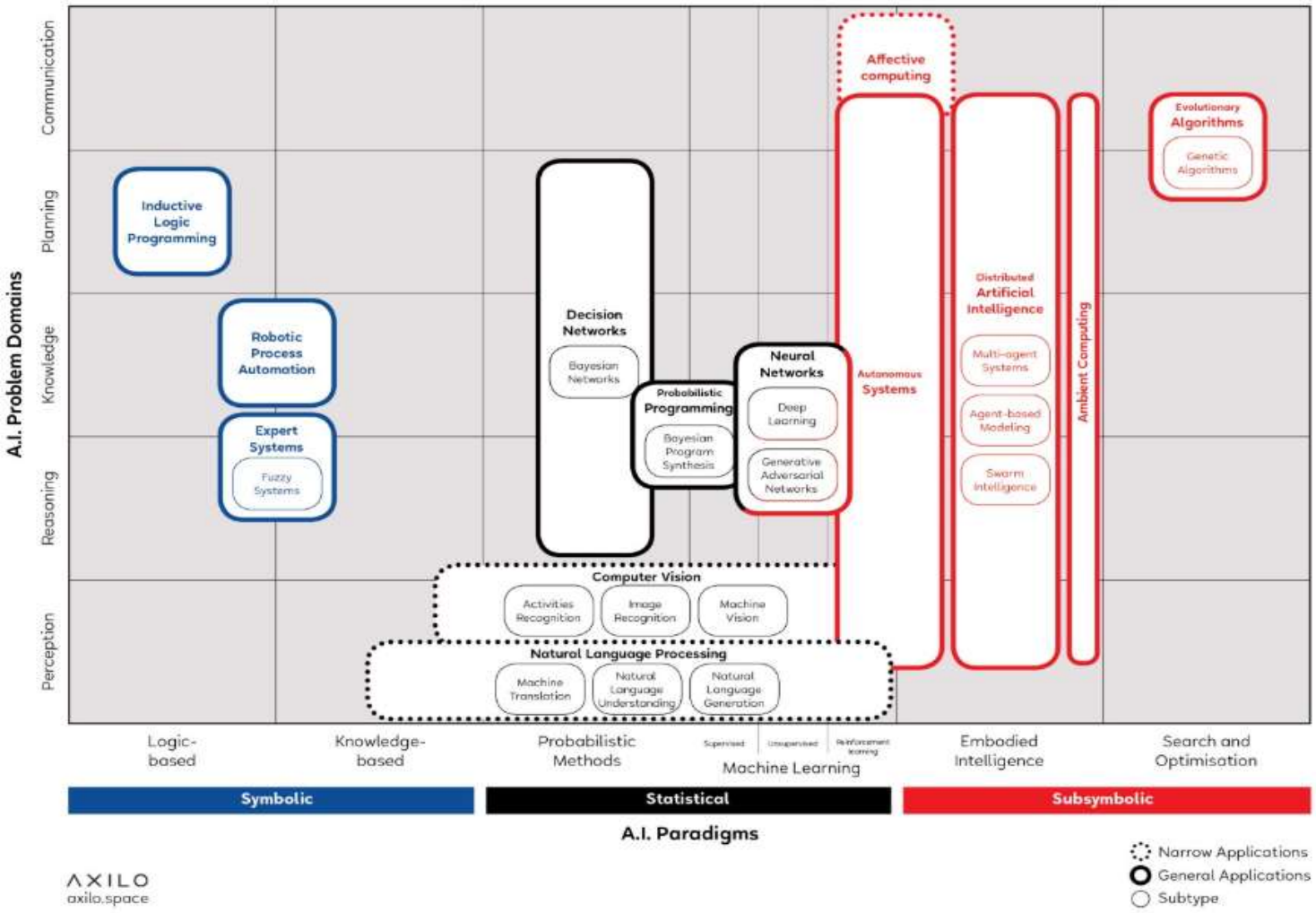
- MAS not necessarily involved AI but MAS is a paramount stepstone toward a better AI.
  MAS 不一定需要AI来实现，但MAS是走向更高级AI的主要垫脚石。

- More recently MAS and AI combine to form a field called Distributed AI (DAI). 分布式AI

- DAI studies followings： DAI 特点:
  - Agents
    - Granularity differs
    - Knowledge level differs
  - Distribution of tasks between agents 如何分配任务
  - Distribution of powers 如何分配权力
  - Communications of agents. 如何沟通交流

- Swarm Intelligent studies the special situation: 群体智能
  - Many fairly homogeneous agents。 大量相同类似代理
  - basic rules to communicate with each other or via environment 简单通信或无通信
  - group demonstrate emergent intelligence, show self organize and fault tolerant capabilities. 显示出群体智能，自组织，高容错特点
  - embeds some degree of randomness in the decision process of each agent. 总是包括一些随机因子。

- More studies on MAS with ML (specifically reinforcement learning, deep learning and deep convolutional networks), since ML can use Agent Based Model as an environment and a reward generator while Agent Based Model can use ML to refine the internal models of the agents.



Single Autonomous System needs AI from perception, reasoning, knowledge, planning, communication domains

Distributed AI requires additional MAS, ABM and Swarm intelligent technologies.

https://medium.com/@Francesco_AI/ai-knowledge-map-how-to-classify-ai-technologies-6c073b969020

References

- R. Collier, et al., 2019 MAMS Multi Agent Microservice
- J Muller The Agent Architecture InteRRaP Concept and Application
- https://site.ieee.org/pes-mas/agent-technology/applications/
- Agent-oriented Programming
- WADE-User-Guide.pdf
- Francesco Corea, 2019 A primer for MAS, ABM and Swarm Intelligent
- Francesco Corea, 2018 AI knowledge map: how to classify ai technologies
- C. Gerber, 1999 Flexible autonomy in holonic agent system
- Latika Kharb，Enes Sukic 2015 An Agent Based Software Approach towards Building Complex Systems
- Michael E. Bratman Belief-Desire-Intention(BDI) reasoning model