



# Desarrollo Basado en Agentes

## Tema 1. Introducción

*Luis Castillo Vidal*

*Dpto Ciencias de la Computación e I.A.  
2016-2017*



# Esquema del tema

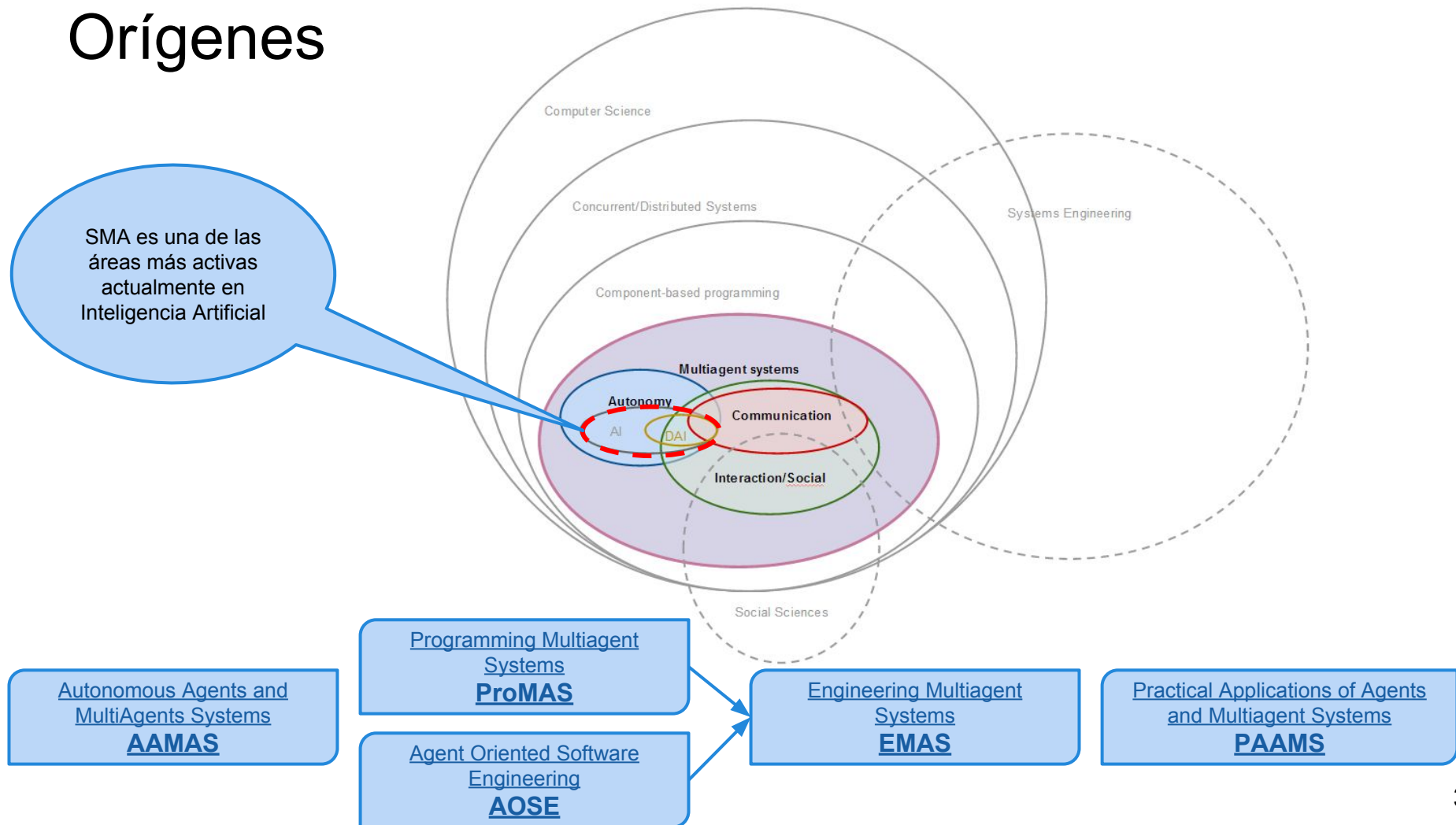
- Historia
- Definiciones previas
- Ingeniería del software orientada a agentes
- Metodologías de desarrollo basadas en agentes



# Historia

## Orígenes

SMA es una de las áreas más activas actualmente en Inteligencia Artificial

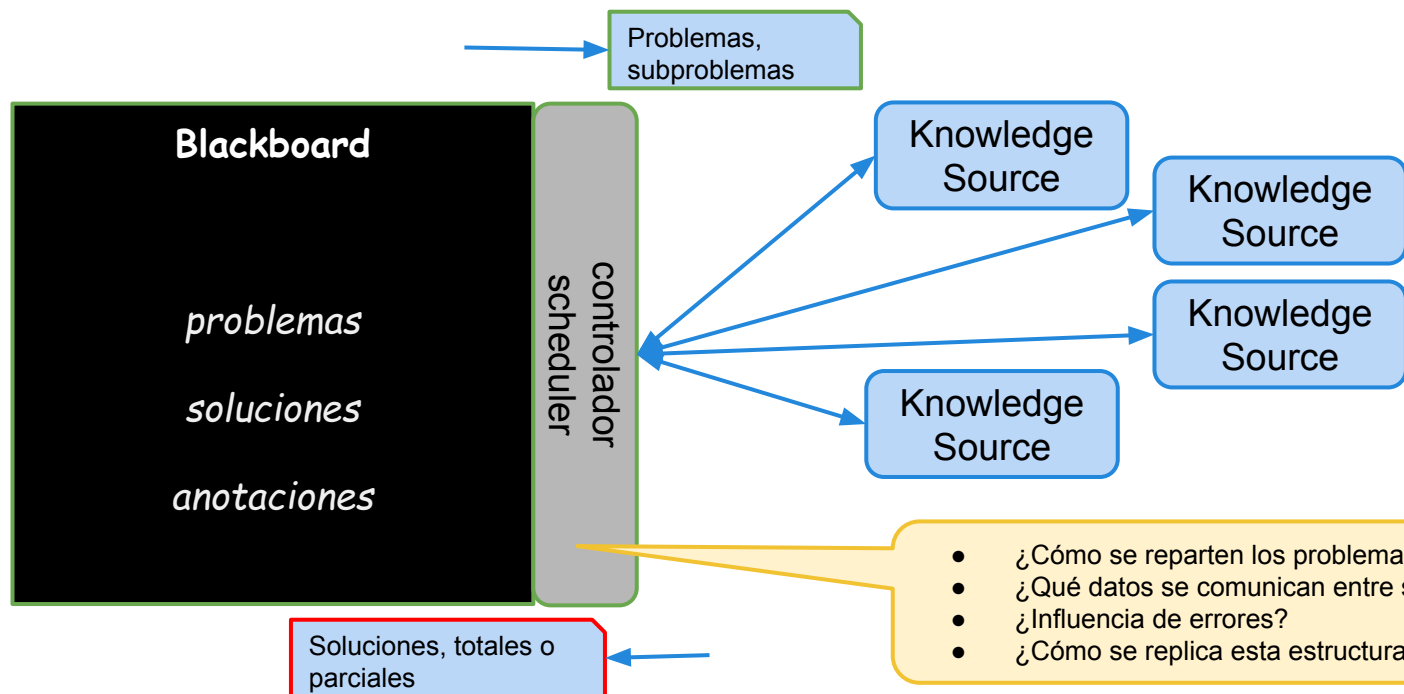


# Historia

- Primera etapa 1956-1990 (inteligencia)
- Década de los 90 (comunicación)
- Finales siglo XX (dimensión social)
- Siglo XXI (ingeniería)

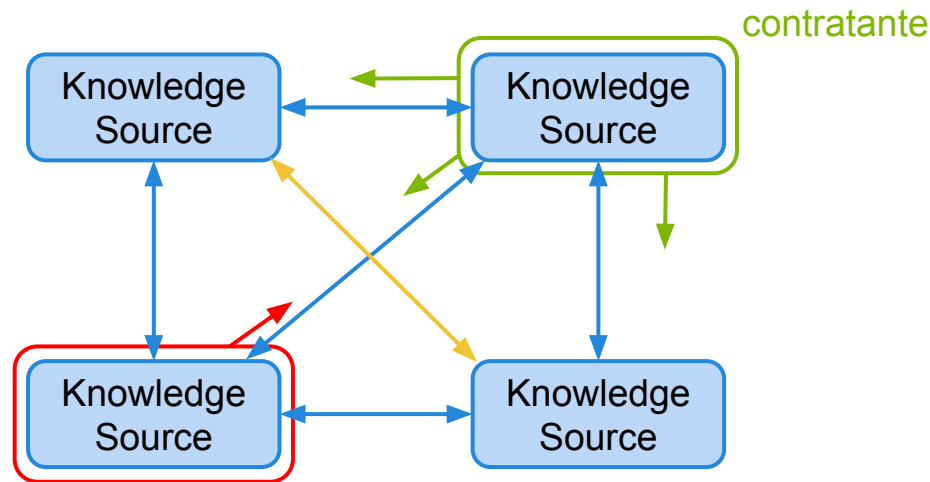
# Historia

- Primera etapa 1956-1990 (inteligencia)
  - Sistemas cooperantes y DAI
    - Arquitectura de pizarra (**centralizado**)



# Historia

- Primera etapa 1956-1990 (inteligencia)
  - Sistemas cooperantes y DAI
    - Red de contratos “contract net” (en red)
      - Cooperación mediante negociación
      - Subcontratos



contratado

# Historia

- Primera etapa 1956-1990 (inteligencia)
  - Arquitecturas **cognitivas** y **reactivas** (vs deliberativas)
    - Brooks '91
      - Agentes simples de estímulo-respuesta: **robots**
      - Estructura jerárquica de comportamientos
      - Inteligencia colectiva emergente (robots)
  - **Organizaciones** de agentes ( ← org. humanas)
    - Modelos de organizaciones
      - Roles fijos
      - Esquemas mixtos
    - Arquitecturas auto-organizadas
      - Comunicación
        - Sintaxis, Semántica
        - Ontologías

# Historia

- Década de los 90 (comunicación)
  - **Internet** reabre el tema de la comunicación entre agentes y habilita un transporte con posibilidades ilimitadas
  - Aplicaciones de agentes orientados como asistentes
    - DARPA (2003) → CALO → SIRI (Apple)
  - Softbots v.s. Robots
  - Agentes móviles → en espacios de información (inet)



# Historia

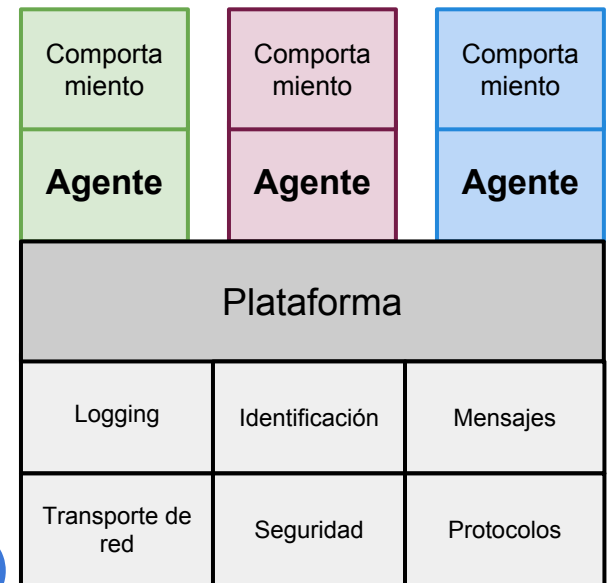
- Década de los 90 (comunicación)
  - Comunicación (actos comunicativos)
    - “Speech act theory” Searle ‘60, Winograd and Flores in the ‘70s
      - Performativas (22) y parámetros
        - **accept-proposal**: aceptar una propuesta recibida previamente
        - **agree**: estar de acuerdo en realizar alguna acción
        - **cancel**: cancelar alguna acción pedida previamente
        - **cfp**: solicitar propuestas para realizar una acción dada
        - **confirm**: informar a un receptor que una proposición es cierta
        - **disconfirm**: informar a un receptor que una proposición es falsa
        - **failure**: informar a otro agente que se intentó una acción pero falló
        - **inform**: informar a un receptor que una proposición es cierta
        - **inform-if**: si el agente que recibe la acción cree que la sentencia es verdadera informará de manera afirmativa, sino indicará que es falsa.
        - **inform-ref**: permite que el emisor informe al receptor de un objeto que cree que corresponde a un descriptor, como puede ser un nombre u otra descripción que lo identifique.
        - **not-understood**: informar a un receptor que el emisor no entendió el mensaje
        - **propagate**: el receptor trata el mensaje como si fuese dirigido directamente a él, y debe identificar los agentes en este descriptor y enviarles el mensaje a ellos
        - **propose**: enviar una propuesta para realizar una cierta acción
        - **proxy**: el receptor debe seleccionar agentes objetivo denotados por una descripción dada, y enviarles un mensaje embebido
        - **query-if**: preguntarle a otro agente si una determinada proposición es cierta
        - **query-ref**: preguntar a otro agente por el objeto referenciado en una expresión
        - **refuse**: rechazar realizar una acción
        - **reject-proposal**: rechazar una propuesta durante una negociación
        - **request**: solicitar a un receptor que realice alguna acción
        - **request-when**: solicitar al receptor que realice alguna acción cuando una proposición dada sea cierta
        - **request-whenever**: solicitar al receptor que realice alguna acción cada vez que una proposición dada sea cierta
        - **subscribe**: una intención persistente de notificar al emisor de un determinado valor, y volver a notificarle cada vez que dicho valor cambie

# Historia

- Década de los 90 (comunicación)
  - Comunicación (actos comunicativos)
    - “Speech act theory” Searle ‘60, Winograd and Flores in the ‘70s
      - Performativas (22) y parámetros
    - DARPA
      - KQML
      - KIF
    - FIPA
      - ACL

# Historia

- Década de los 90 (comunicación)
  - Comunicación (actos comunicativos)
    - “Speech act theory” Searle ‘60, Winograd and Flores in the ‘70s
      - Performativas (22) y parámetros
    - DARPA
      - KQML
      - KIF
    - FIPA
      - ACL
    - Plataformas
      - JADE → secuelas
    - Cualquier otra ( $\neg$ extremismo)

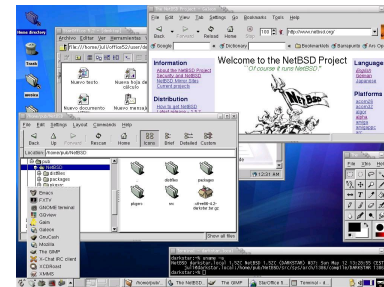
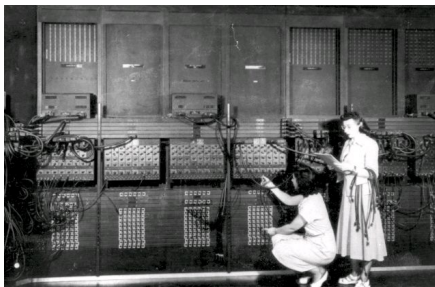


# Historia

- **Finales siglo XX (dimensión social)**
  - Todas las anteriores más una dimensión social
    - Interacciones
    - Decisiones grupales
    - Coaliciones
    - Uso racional de recursos: tiempo, espacio, ...
    - Negociación
    - Argumentación
      - Motivado por la **incertidumbre, incompletitud e imprecisión.**
      - Self explained
        - Convencer a otros
        - Explicar artefactos complejos a otros (humanos)
    - Emociones
    - Benevolencia
    - Veracidad y confianza de unos agentes sobre otros
    - **Aprendizaje**

# Historia

- Siglo XXI (Dagstuhl Roadmap 2012 → Engineering Multi-Agent Systems)
  - Ubicuidad e Interconexión
  - Inteligencia y cognición
  - Delegación (NASA RAX, UAV, Google Car)
  - Orientado a humanos
    - Más abstractos
      - interruptores → command line → GUI → Habla



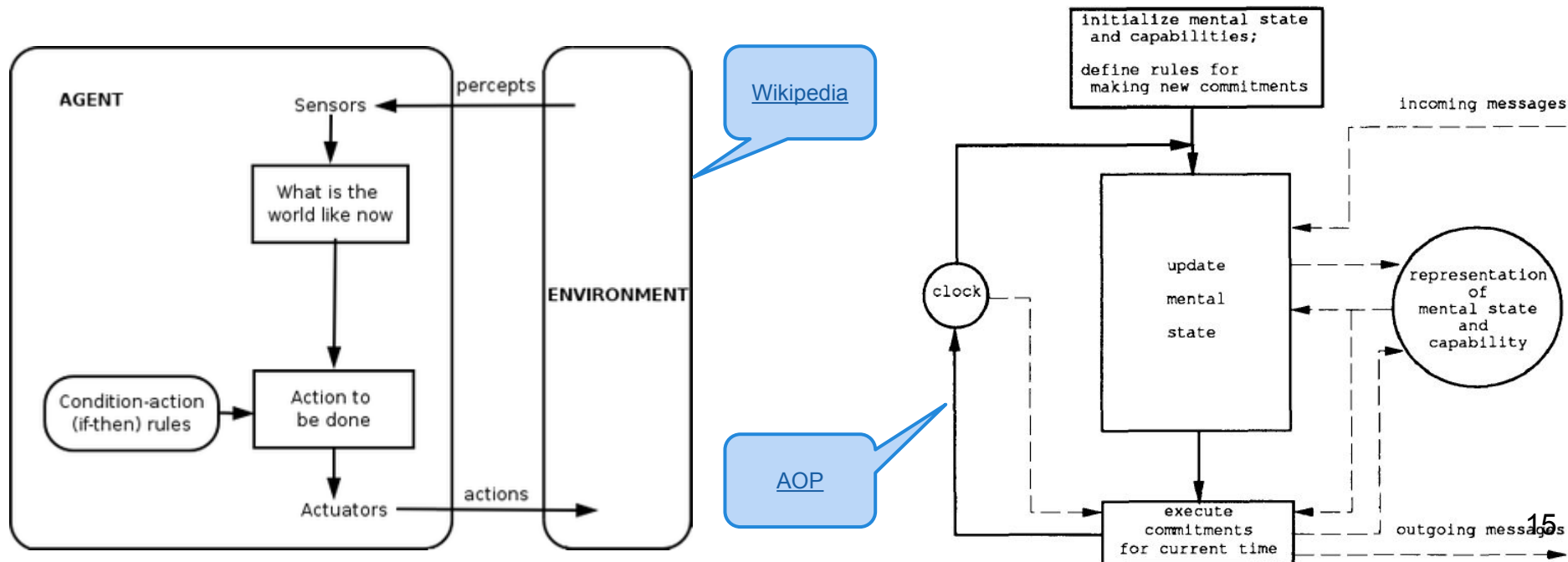
# Historia

- Siglo XXI (Dagstuhl Roadmap 2012 → Engineering Multi-Agent Systems)
  - Orientado a humanos
    - En beneficio de las personas
    - Agentes como una herramienta para comprender sociedades humanas (simulación)
      - **Agent-Based Modeling** → Marketing, Redes sociales, ...
  - **Independencia de la plataforma (FIPA/DARPA)**
  - **Incluso del lenguaje de programación → ACL**
  - Tecnológicamente
    - Metodologías de desarrollo estándar
    - Herramientas de desarrollo y benchmarking

# Definiciones previas

## ● Agente

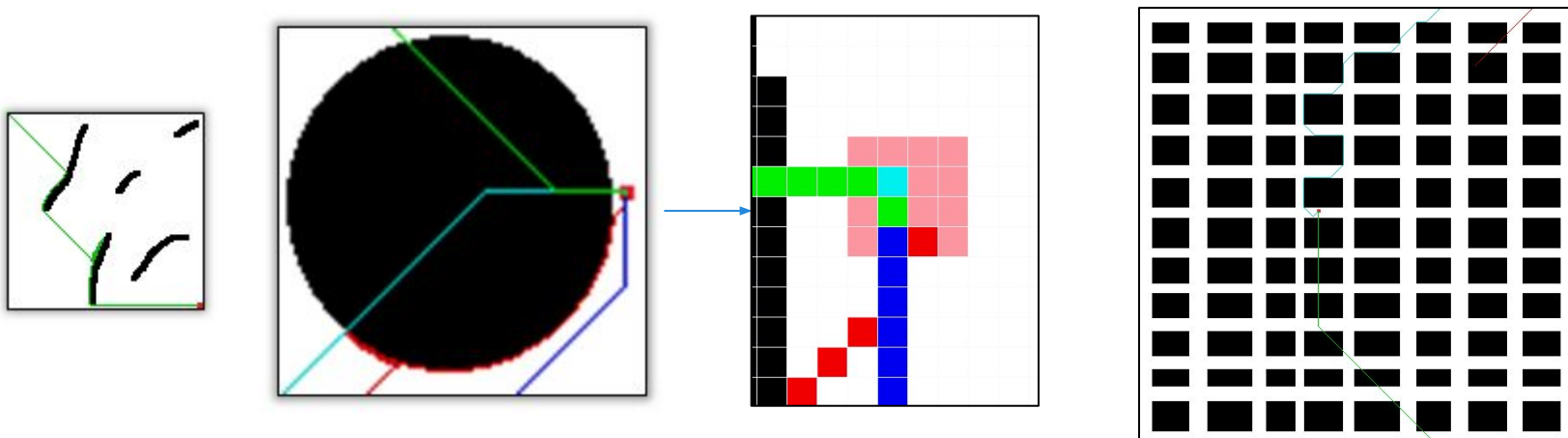
Un sistema computacional **situado en un entorno** y capaz de actuar de forma independiente o **autónoma** para conseguir sus **propios objetivos** (sin que una persona tenga que decirle el cómo), **en representación** de otra instancia (otro agente o una persona o él mismo)



# Definiciones previas

- **Sistemas Multiagente**

- Es un sistema que se compone de un conjunto de agentes, los cuales **interactúan** los unos con los otros, para la consecución de sus **propios objetivos** o de **objetivos comunes**, para lo cual necesitan **comunicarse, cooperar, coordinarse y negociar**.





# Definiciones previas

- Bond & Gasser '88: artículo sobre DAI
  - a. ¿Cómo formular a un agente el problema a resolver?
  - b. ¿Cómo se deben comunicar los agentes e interactuar entre ellos?
  - c. ¿Como garantizar la coherencia de la actuación de un agente?
  - d. ¿Cómo permitir a un agente razonar sobre problemas, acciones y conocimiento de otros agentes? ¿Cómo razonar sobre el estado de procesos coordinados?
  - e. ¿Cómo resolver conflictos entre agentes que se coordinan?

# AOSE

## Agent-Oriented Software Engineering

- Programación orientada a objetos?
- Sistemas concurrentes y distribuidos?
- Ingeniería de sistemas complejos?
- Inteligencia Artificial? Sistemas Expertos?
- Teoría económica?
- Teoría de juegos?
- Ciencias sociales?

AOSE se sitúa en la intersección de todos ellos

# AOSE

## Agent-Oriented Software Engineering

- AOP se puede ver como una especialización de OOP
  - Parecidos
    - Encapsulación del estado
    - Comunicación mediante mensajes
    - Puede realizar ciertas operaciones: métodos
  - Diferencias
    - Concepto de autonomía más potente → decisión autónoma
    - Inteligencia (reactivos, proactivos, sociales)
    - No pasividad
    - $AOP \subseteq OOP$ 
      - Los agentes no comparten estructuras de datos comunes
  - [+Info](#)

# AOSE

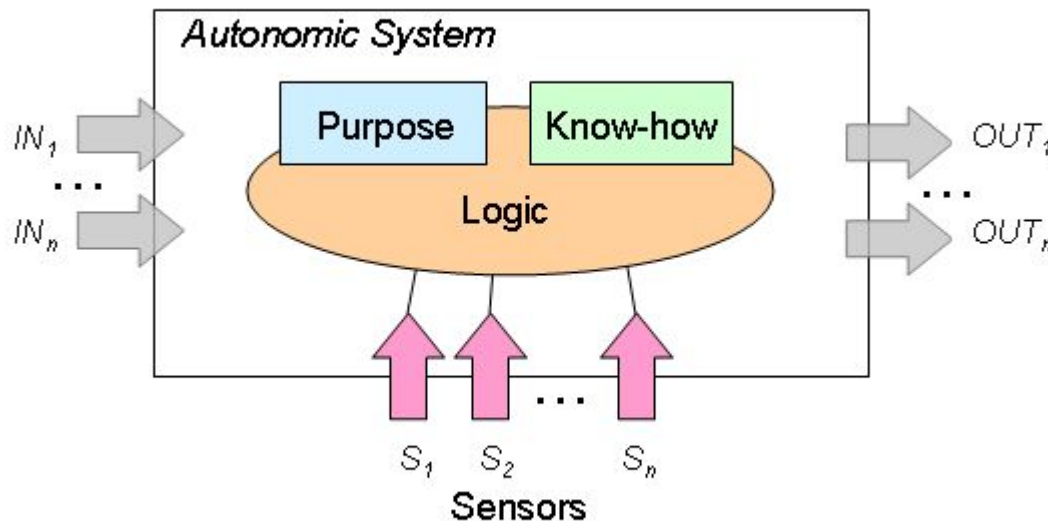
## Agent-Oriented Software Engineering

- Adecuabilidad como paradigma de SE
  - Técnicamente (estricto) aporta poco a las metodologías OOP
  - Semánticamente es mucho más completo
- Autonomía → ya visto en Inteligencia Artificial
- Interacción es la característica más compleja
  - Agentes que perciben el entorno en el que están situados
  - Agentes egoístas o interesados o de mercado
  - Agentes que dialogan de forma compleja
    - El estado del agente depende del estado del diálogo
  - Grid computing: sistemas abiertos que colaboran en un mismo problema
  - Computación ubicua (internet de las cosas, IoT)

# AOSE

## Agent-Oriented Software Engineering

- **Autonomic computing** ([IBM](#)): self-healing, self-configuring, self-optimising, self-protecting hardware and software in IT infrastructures.



# AOSE

## Agent-Oriented Software Engineering

- **Ciberseguridad** ([US NIST](#)): Protección de infraestructuras IT como un sistema multiagente de defensa (a imitación del [sistema inmune](#))



- Requiere el uso de **metodologías** de desarrollo específicas

# Metodologías de desarrollo

- Fases separadas de **modelado y desarrollo**
- **Modelado (UML)**
  - Tipos de agentes (Diagrama de clases)
    - Funcionamiento interno (Diagrama de actividad)
  - Tipos de roles (Diagramas de clases)
  - Comunicación
    - Por cada iniciador de mensajes (Diagrama de secuencia)
      - Protocolo de comunicación y agentes involucrados
- **Desarrollo**
  - Proyecto de implementación
    - Plataforma de agentes
    - Lenguaje de programación

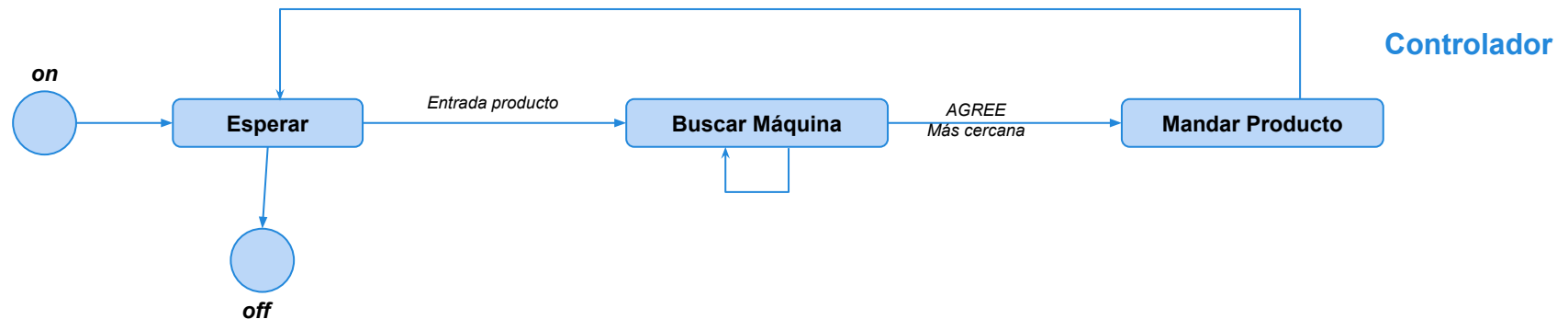
# Jerarquías de agentes





# Metodologías de desarrollo

## Modelo interno de cada agente



# Modelo de comunicación entre agentes



# Metodologías de desarrollo

## Desarrollo de cada agente

```
import ...

public class Maquina extends SingleAgent{

    private int estado;
    private final int ESPERAR = 0;
    private final int PROCESAR = 1;
    private final int BUSCAR = 2;
    private final int ENVIAR = 3;

    private int subestado;
    private final int ESPERANDO_CFP = 0;
    private final int ESPERANDO_ACC_REJ = 1;
    private final int ESPERANDO_AGR_REF = 2;
    private final int OCUPADO = 3;

    private int id, fila, columna, tipo;
    private AgentID maquinas[];
    private Entorno entorno;
    private ACLMessage mensaje;
    private MessageQueue colaMensajes = null;
    private String producto_actual;

    private boolean libre;

    public Maquina(AgentID aid, int i, int fil, int col, int t, Entorno e) throws Exception {...}

    public void setMaquinas(AgentID[] m) {...}

    public void init() {...}

    public void execute() {...}

    public void finalize() {}
```

# Metodologías de desarrollo

- Los cuatro pasos de las vocales AEIOU A(rquitectura) E(ntorno) I(nteracción) O(rganización) U(tilidad)
  - Arquitectura
    - Interna del agente
  - Entorno
    - Percepción del agente
  - Interacción
    - Comunicación, diálogo
  - Organización
    - Entre los agentes
  - Utilidad
    - Objetivos
- Agent design (microlevel) → society design (macrolevel)
- + Info en Tema 4: Otras metodologías (GAIA)

# ¡Atención!

1. **Exceso de confianza**
  - a. You Get Religious
    - i. Believe Agents = Silver Bullet
    - ii. See agents everywhere
    - iii. Too Many Agents
    - iv. Too few agents
2. **Desconocer las capacidades**
  - a. Don't Know Why You Want Agents
  - b. Don't Know What Agents Are Good For
  - c. Use Too Much AI
  - d. Not Enough AI
3. **Es un proyecto de software**
  - a. Confuse Prototypes with Systems
  - b. Forget it's Software
  - c. Forget its distributed
  - d. Forget it is usually networked
  - e. Reinventar la rueda
    - i. Don't Exploit Related Technology
    - ii. The tabula rasa
      1. Want Your Own Architecture
      2. Implementing infrastructure
  - f. Don't exploit concurrency
  - g. System is anarchic
  - h. Confuse simulated with real parallelism
4. **Ignore de facto standards**



# Glosario

- DAI. Distributed Artificial Intelligence
- OOP. Object-Oriented Programming
- AOP. Agent-Oriented Programming
- AOSE. Agent-Oriented Software Engineering
- DARPA. Defense Advanced Research Projects Agency
- KQML. Knowledge Query and Manipulation Language
- KIF. Knowledge Interchange Format
- FIPA. Foundation for Intelligent Physical Agents
- JADE. Java Agent DEvelopment framework
- ACL. Agent Communications Language