



# Desarrollo Basado en Agentes

## Tema 4. Interacción (II)

*Luis Castillo Vidal*

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



# Contenido

- **Introducción a las sociedades de agentes**
  - Estructuras sociales
  - Instituciones electrónicas
  - Metodologías de desarrollo
- **Temas clave**
  - Coordinación
  - Toma de decisiones en grupo
  - Formación de coaliciones
  - Reserva de recursos
  - Argumentación
  - Emociones



# Metodologías de desarrollo

- La evolución de la Ingeniería del Software en base a las abstracciones que utiliza



# Metodologías de desarrollo

- Características distintivas de los SMAs  
**(Agent Oriented Software Engineering)**
  - Sistemas abiertos y distribuidos
  - Entidades autónomas y proactivas (intencionales)
  - Entidades situadas en un contexto, el cual perciben
  - Entidades interactivas entre sí
    - Comunicación semántica
    - Estructuras sociales
  - Entidades adaptativas

# Metodologías de desarrollo

- AOSE potencia las siguientes abstracciones de forma natural (nuevas)
  - Agente como entidad
  - Sigue siendo difícil definir “agente”
  - Situación en un contexto, percepción
  - Organizaciones y sus normas
  - Roles de agentes, interacciones y comunicación
  - Aprendizaje

# Metodologías de desarrollo

- Un nivel de abstracción superior a la OOP → Metodología general de ingeniería del software, no sólo para agentes
  - Sin memoria compartida
  - No se llama a métodos de agentes
    - Constructor, start()
  - Valores de retorno diferidos en el paso de mensajes
  - Orden de retorno de valores controlable (cola mensajes)
  - Interoperabilidad de datos (serialización)
- Ojo a los excesos, ver final del Tema 1

# Metodologías de desarrollo

- Existen muchas metodologías en la literatura
  - GAIA → Diseño organizacional
  - TROPOS → Goal-based requirements
  - Prometheus → Arquitectura interna del agente
  - ADELFE → Agentes adaptativos
  - MESSAGE → Todo lo anterior (muy complejo)
  - PASSI → AOSE+AI

# Metodologías de desarrollo

- Existen muchas metodologías en la literatura
- Muy usado a nivel de investigación pero verde aún a nivel industrial
- Todas ellas dan por hecho que se va a diseñar un SMA y no pretenden distinguir si un SMA es adecuado o no ←

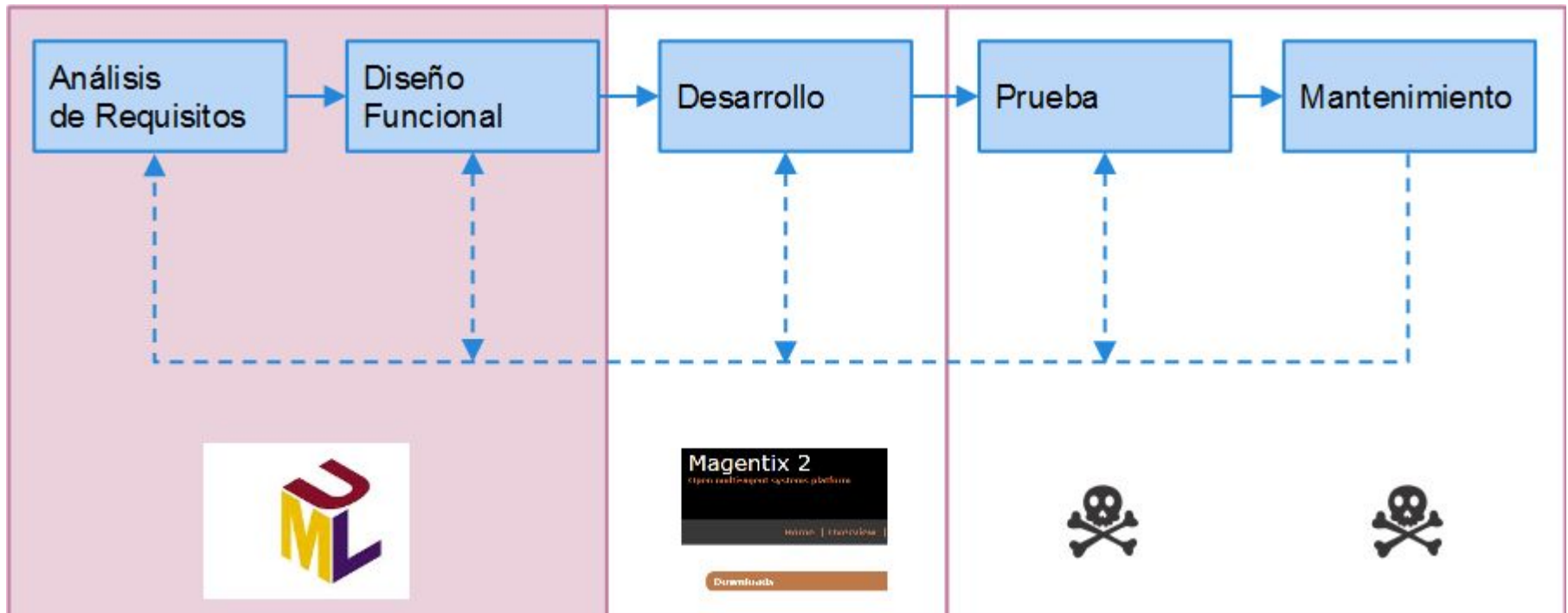


# Metodologías de desarrollo

- ¿Es un SMA el modelo de SW más apropiado?
  - Entornos abiertos, dinámicos, inciertos, complejos
    - SOA: Arquitecturas Orientadas a Servicios
    - Protocolos de comunicación diversos y complejos
  - Fuentes de datos distribuidas
  - Conocimiento experto y/o decisión distribuidos
  - Control inherentemente distribuido
- “Agentificación” de SW obsoleto (encapsulación) para introducirlo en SOA/SMA

# Metodologías de desarrollo

- Paradigma de desarrollo
  - Basado en el modelo clásico “waterfall”



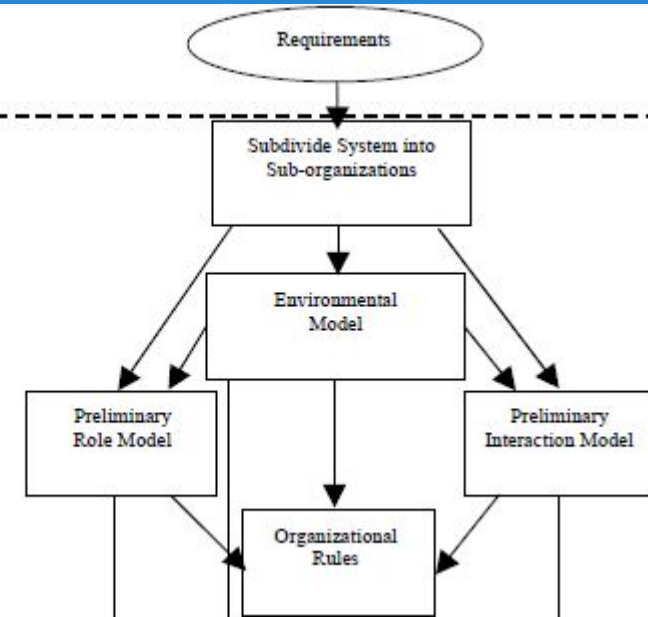
# Metodologías de desarrollo

- Paradigma de desarrollo

- Se elaboran un conjunto de modelos en orden decreciente de abstracción y se articula la implementación en alguna de las plataformas de agentes
- Metodología GAIA: La organización de los agentes es la base del modelado. No está ligada a una notación o implementación concreta (usaremos UML+Magentix)
  - ~~Estudio de requisitos~~
  - **Análisis:** propiedades del problema
  - **Diseño arquitectural:** propiedades de la solución propuesta
  - **Diseño detallado:** propiedades de la implementación
  - ~~Implementación~~

## ● Análisis

- ## COLLECTION OF REQUIREMENTS

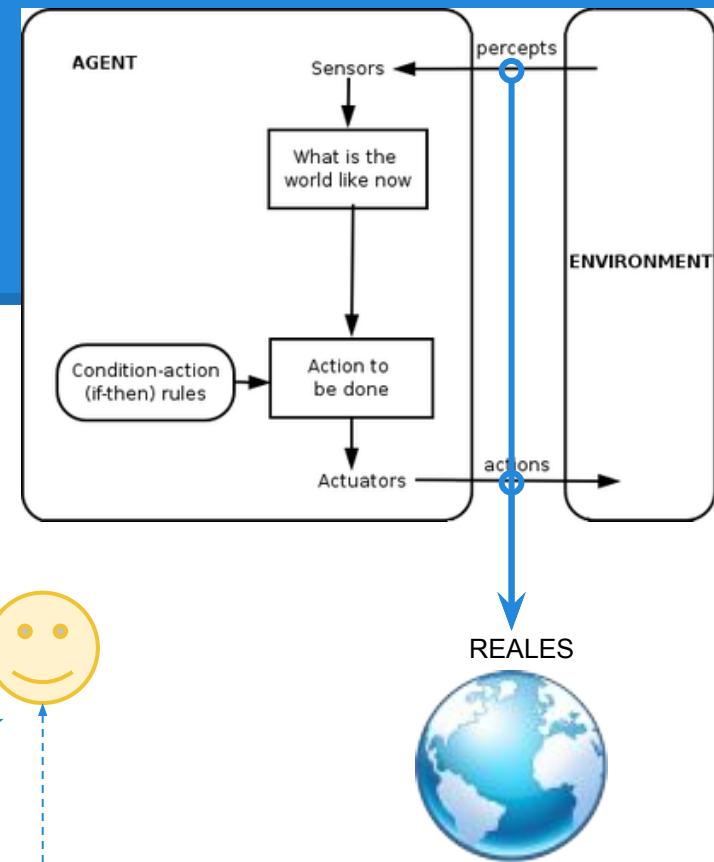
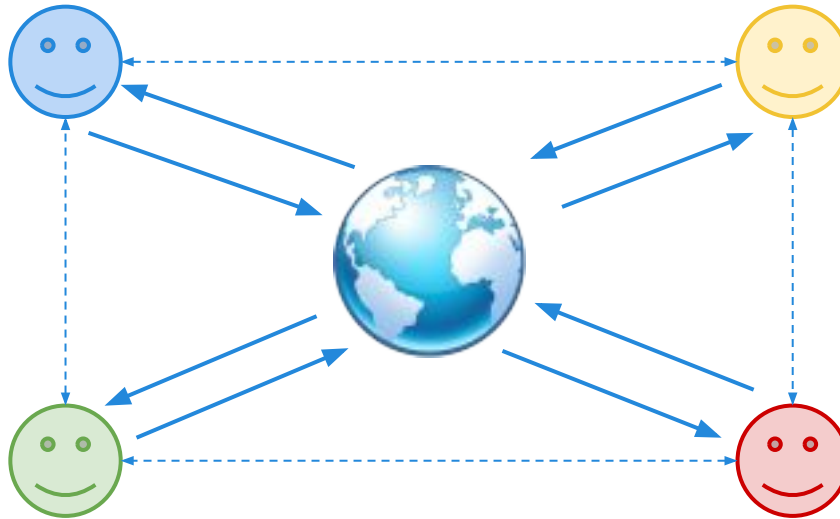


## Recursos

# Metodología GAIA

- **Análisis**

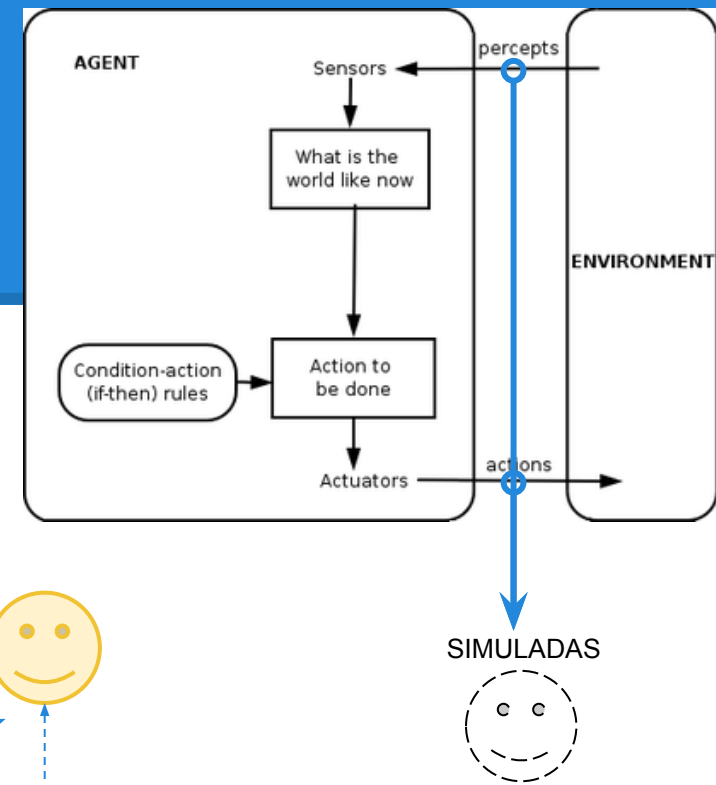
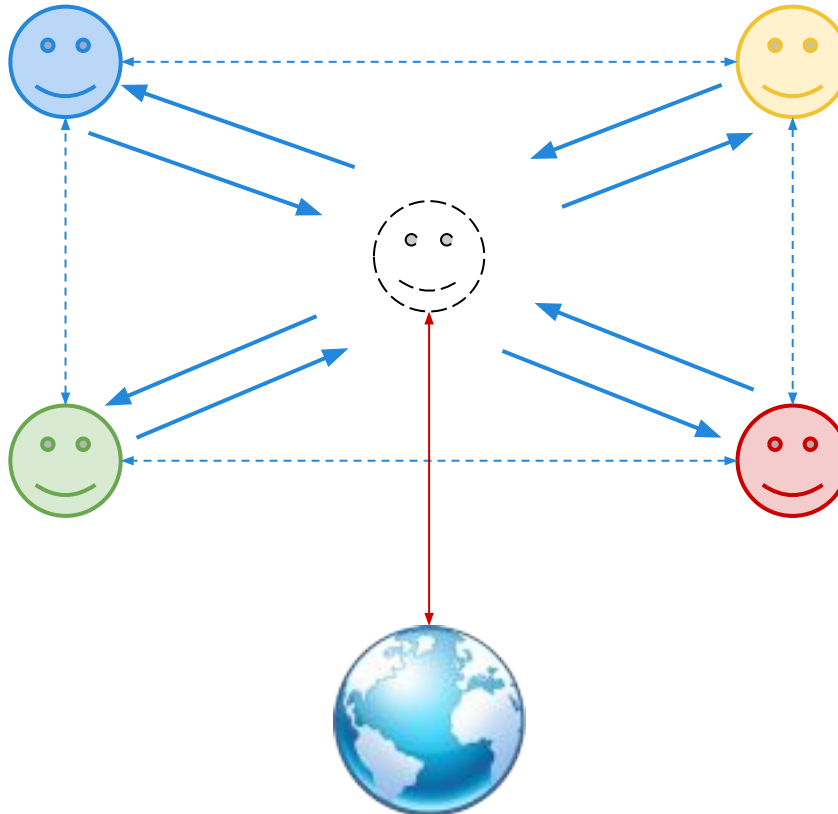
- Modelo del entorno
  - Real



# Metodología GAIA

- **Análisis**

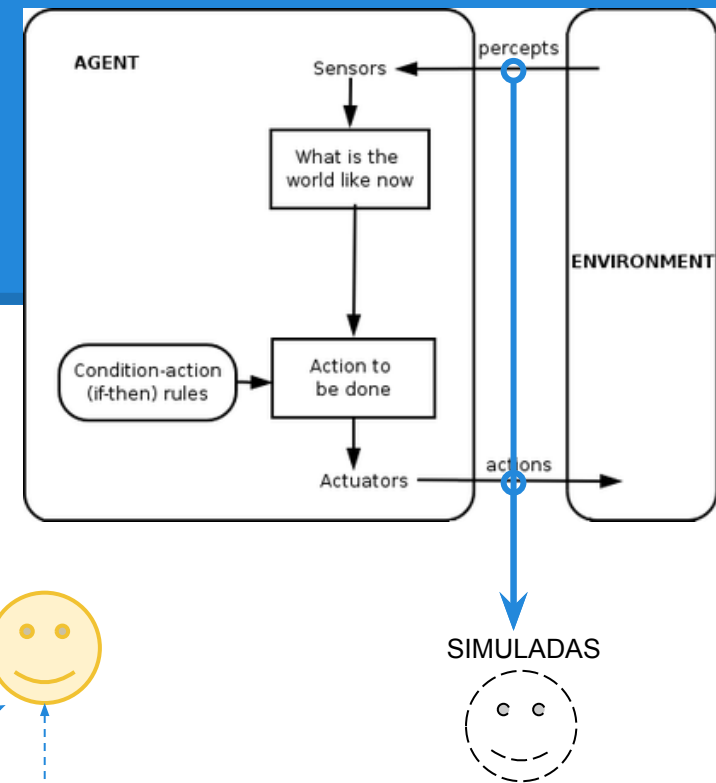
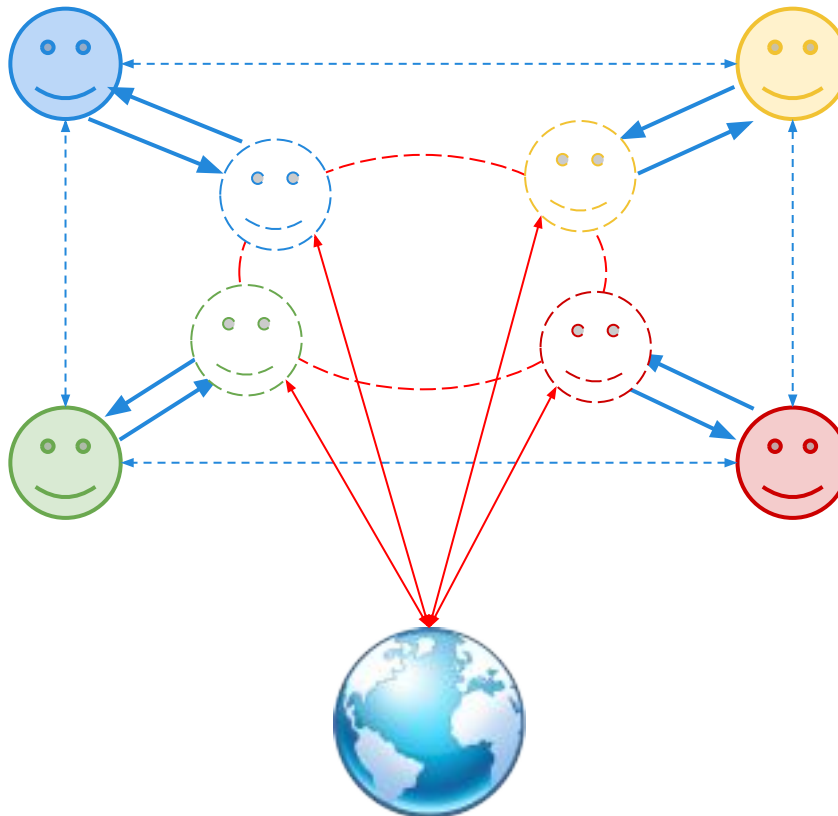
- Modelo del entorno
  - Simulado Centralizado



# Metodología GAIA

- **Análisis**

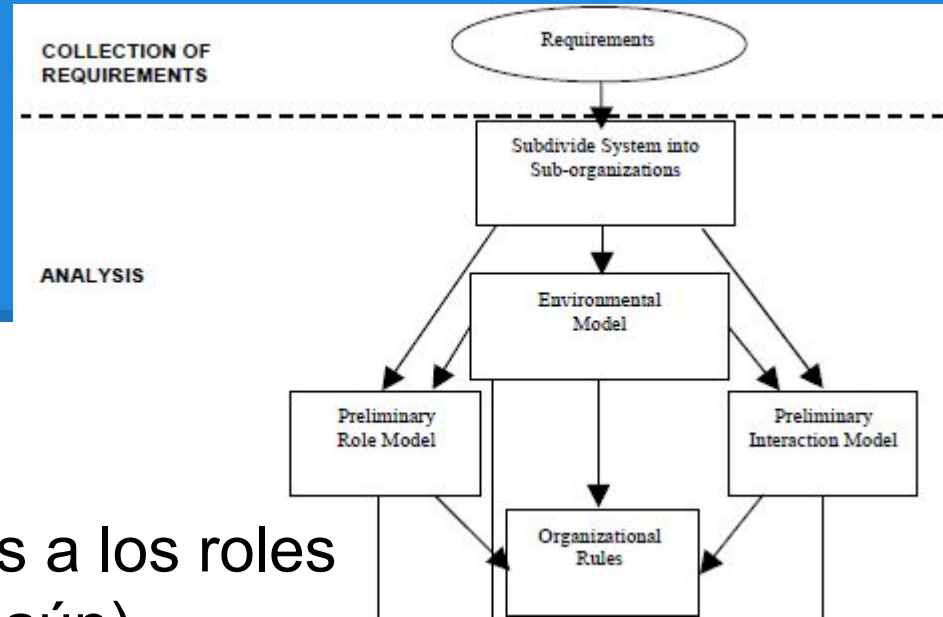
- Modelo del entorno
  - Simulado Distribuido



# Metodología GAIA

## ● Análisis

- Modelo de roles preliminar
  - Asignar las capacidades a los roles operativos (no agentes aún)
    - Permisos, Responsabilidades, Capabilities
  - No contemplar aún la estructura de la organización

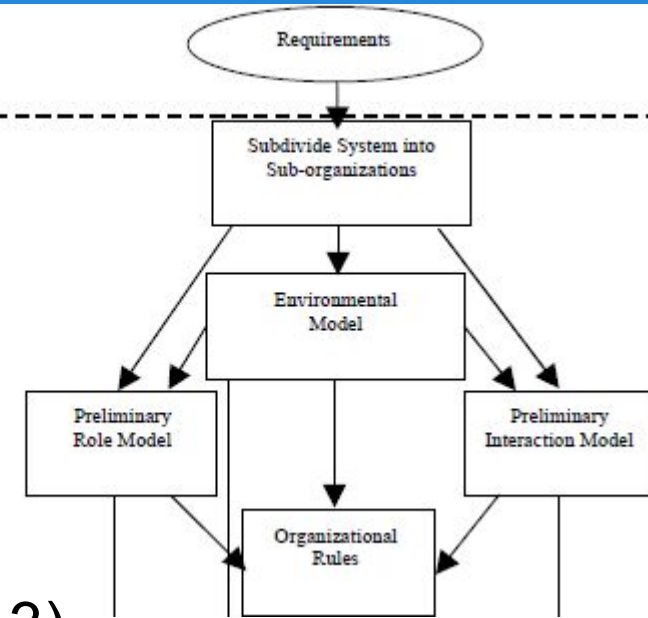




# Metodología GAIA

COLLECTION OF  
REQUIREMENTS

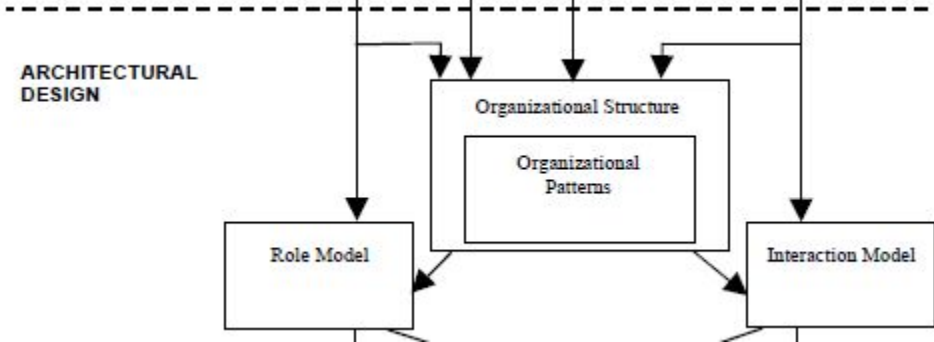
ANALYSIS



## ● Análisis

- Modelo de interacción preliminar
  - Dependencias entre los roles preliminares → Protocolo (Tema 3)
  - No contemplar aún la estructura de la organización
- Reglas de la organización
  - Restricciones sobre las capacidades de los roles
  - Restricciones sobre las interacciones

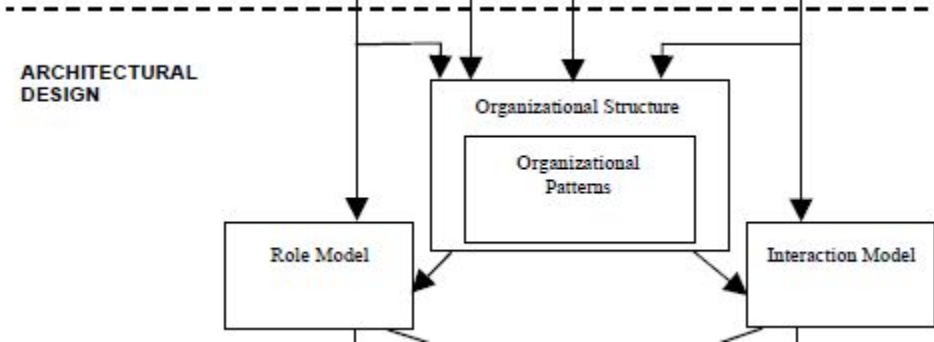
# Metodología GAIA



## ● Diseño arquitectural

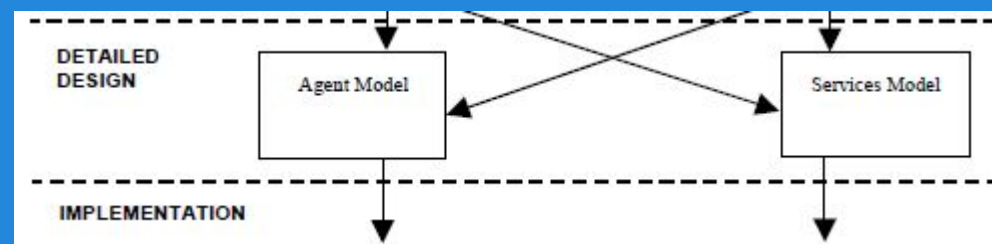
- Estructura de la organización
  - Topología ( ← catálogos de organizaciones)
  - Régimen de control
    - Estructura social
    - Modelo de acceso al entorno
- Modelo de roles completo
  - Basado en la estructura de la organización
  - “Separar bien” la parte proveniente del análisis (estática) de la que se ha diseñado (dinámica)
    - Roles del problema
    - Roles organizativos

# Metodología GAIA



- **Diseño arquitectural**
  - Modelo de interacción completo
    - Idem
      - Protocolos del problema
      - Protocolos organizativos

# Metodología GAIA



## ● Diseño detallado

- Modelo de agentes
  - Clases de agentes
  - Relación “clase de agente”  $\leftarrow \rightarrow$  Rol
    - 1-1, 1-n, n-n
- Modelo de servicios
  - DIAGRAMAS DE ACTIVIDAD: Identificar los bloques de actividades principales que llevará a cabo cada agente ( $\approx$  métodos)
    - Disparados por peticiones de otros agentes
    - Disparados por el propio agente
- Componentes ( $\approx$  métodos)
  - Entradas y Precondiciones
  - Salidas y Postcondiciones

TDA: Diagramas de clase (OOP),  
texto libre, control del entorno

Texto libre (Dignum, Findler)

Diagramas de clase (interface)

Diagramas de secuencia (puede  
haber más de uno)

Diagramas de clase (Agentes)

Diagramas de actividades+Texto  
libre. Uno por cada tipo de agente

