

INGENIAS: Desarrollo dirigido por modelos de SMA

Juan Pavón Mestras

jpavon@pdi.ucm.es

Dep. de Ingeniería del Software e Inteligencia Artificial
Universidad Complutense Madrid



<http://grasia.fdi.ucm.es>

Objetivo de INGENIAS

- Proporcionar soluciones de ingeniería para desarrollar SMA
 - Notación
 - Lenguaje visual para expresar el diseño de SMA y agentes
 - Métodos
 - Proceso de desarrollo
 - Organización de entregas
 - Actividades relacionadas
 - Herramientas: Ingenias Development Kit (IDK)
 - Generación de especificación
 - Validación de diseño
 - Generación de código
 - Generación de documentación

Planteamiento de INGENIAS

- Principios
 - Agentes como paradigma de modelado
 - Conceptos de más alto nivel que en objetos y más cercanos al dominio
 - Se pueden considerar adaptaciones específicas a dominios de aplicación particulares
 - Los aspectos organizativos e intencionales reducen el salto de especificación de requisitos al diseño (modelo, solución) de sistema
 - Implementación sobre distintos tipos de plataforma
 - Un modelo de SMA se puede implementar sobre una plataforma de agentes o sobre un entorno de objetos tradicional
 - La metodología facilita y promueve el desarrollo de herramientas de generación de código que faciliten el paso del modelo (análisis y diseño) a la implementación
 - Contempla la evolución de la tecnología de agentes
 - Integración de resultados del área
 - Adaptabilidad a nuevos lenguajes y estándares (p.ej. AUML)
- Todo ello basado en la utilización y manipulación de meta-modelos y transformaciones
 - **Desarrollo dirigido por modelos de SMA**

Desarrollo de software dirigido por modelos (MDD)

- Los modelos dirigen el desarrollo y mantenimiento del sistema
 - Además de:
 - Analizar y experimentar
 - Documentar
 - Facilitar la comunicación entre clientes, usuarios y desarrolladores
 - No sólo como *guía* de implementación
 - Los modelos (y no los programas) pasan a ser el centro y los artefactos fundamentales del proceso software
- Herramientas necesarias:
 - Edición de modelos
 - Generación automática de código
 - Potencialmente sobre múltiples plataformas
 - Verificación y validación de modelos
 - Análisis de propiedades
 - Simulación (ejecución de los modelos)

Motivación para MDD de SMA (1/3)

- La mayoría de las propuestas de ISOA hacen énfasis en el modelado
 - MDD trata sobre modelado
- Actualmente se está estudiando en la comunidad de ISOA la integración de métodos y notaciones
 - MDD, y más concretamente MDA, establece un marco de transformaciones entre modelos
- MDD, y especialmente MDA, aborda los cambios en la tecnología
 - ISOA también

Parece razonable considerar MDD/MDA en ISOA

MDD: Desarrollo Dirigido por Modelos

MDA: Model Driven Architecture del OMG

ISOA: Ingeniería del Software Orientada a Agentes, en inglés, AOSE

Motivación para MDD de SMA (2/3)

- INGENIAS
 - Especificación de metamodelos para SMA
 - Mecanismo para integrar resultados del área
 - Fundamento de las herramientas: INGENIAS Development Kit (IDK)

IDK proporciona la infraestructura necesaria para MDD

IDK: INGENIAS Development Kit

SMA: Sistemas MultiAgente

Motivación para MDD de SMA (3/3)

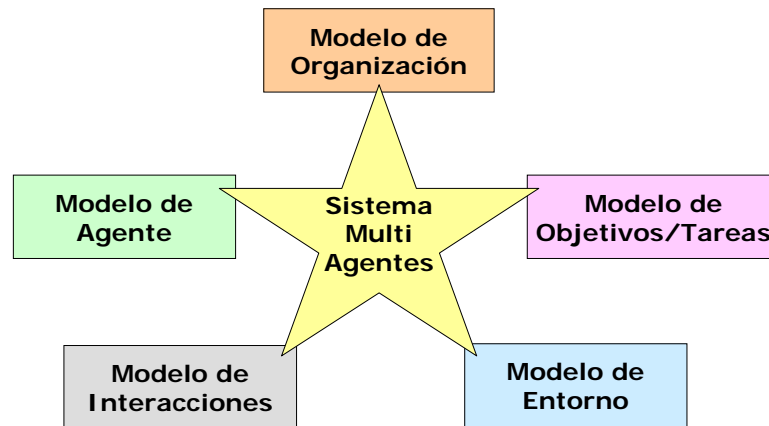
- La experiencia en desarrollos de SMA muestra que
 - **Es conveniente separar modelo de agentes de plataforma de implementación**
 - Las plataformas de implementación son diversas
 - No siempre de agentes
 - **El proceso de desarrollo requiere cambios en las herramientas que manipulan los modelos**
 - Los metamodelos evolucionan
 - Necesidad de incorporar nuevos conceptos
 - Durante el proceso se trata modelado de la aplicación y definición de transformaciones
 - También se va conociendo mejor la plataforma final
 - La reusabilidad es mayor si se aplica también a las actividades del proceso
 - Herramientas
 - Procesos

Metamodelado de SMA

- La comunidad de ISOA es receptiva al metamodelado
 - Ejemplos: Ferber (1993) AOR, Odell (2004) AGR, AOSE2004, AgentLink3 (2005)
- Un buen metamodelo como base de un proceso de ingeniería requiere una validación mediante *transferencia tecnológica* en aplicaciones reales
 - Hay muy pocos metamodelos *completos* para SMA
 - INGENIAS, AML, PASSI, Tropos
 - Ejemplos de aplicación de metamodelos y MDD para SMA
 - En papel: Meta-DIMA, Malaca
 - En la práctica: INGENIAS
 - Method engineering (en desarrollo): FAME, PASSI

Conceptos para el modelado con agentes

- Cinco puntos de vista en INGENIAS



Aspectos del SMA

- Modelo de **organización**
 - Estructura del SMA, roles, grupos, relaciones de poder
 - Dinámica: workflows
- Modelo de **agente**
 - Los agentes realizan tareas o persiguen objetivos
 - Responsabilidades, control y estado mental del agente
- Modelo de **objetivos y tareas**
 - Identificación de objetivos generales y descomposición en objetivos más concretos que se pueden asignar a agentes
 - Similarmente con tareas
 - Objetivos: motivación ⇔ Tareas: actividad
- Modelo de **interacción**
 - Qué interacciones existen entre agentes/roles
- Modelo de **entorno**
 - Entidades y recursos con los que interactúa el SMA

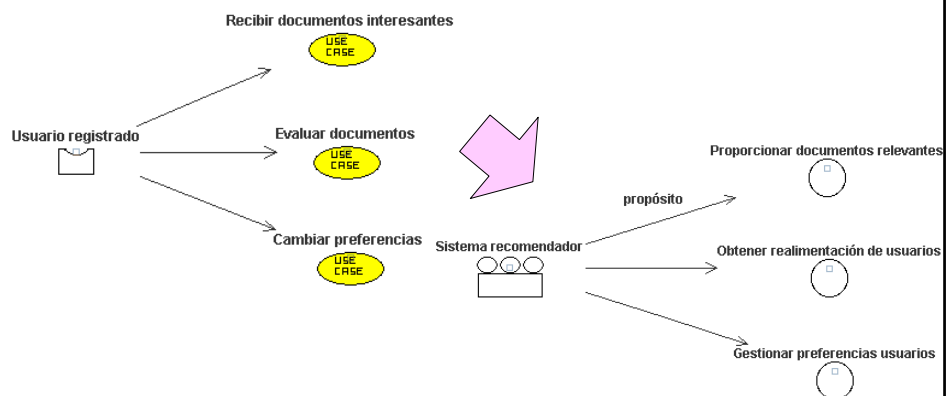
Ejemplo de desarrollo orientado a agentes

- Diseño de un sistema de recomendación usando filtrado colaborativo
 - Los documentos llegan a una comunidad de usuarios
 - Serán evaluados según los gustos mayoritarios de la comunidad de usuarios de gustos similares

Ejemplo de desarrollo orientado a agentes

1. Identificar OBJETIVOS

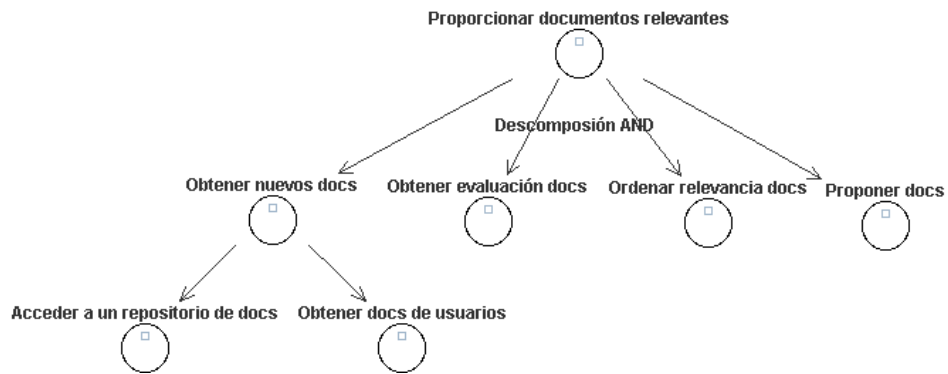
- Análisis: Qué tiene que hacer el sistema (requisitos)
 - De casos de uso a objetivos



Ejemplo de desarrollo orientado a agentes

2. Descomponer los OBJETIVOS en subOBJETIVOS y TAREAS

- Del análisis al diseño:
 - OBJETIVOS identifican requisitos (propósito)



Juan Pavón Mestras, UCM

INGENIAS

13

Modelo de objetivos y tareas

- Qué consecuencias tiene la ejecución de tareas y por qué se deberían ejecutar
 - Justifica la ejecución de tareas basándose en objetivos
 - Que a su vez se van modificando tras su ejecución
- **Objetivo:** Situación deseada
 - Conjunto de estados que el agente quiere lograr, mantener, o evitar
 - Una función de utilidad que maximizar
 - Responde a *¿por qué?*
- **Tarea:** Transiciones de estado
 - Conduce a la consecución de objetivos
 - Responde a *¿cómo?*



Juan Pavón Mestras, UCM

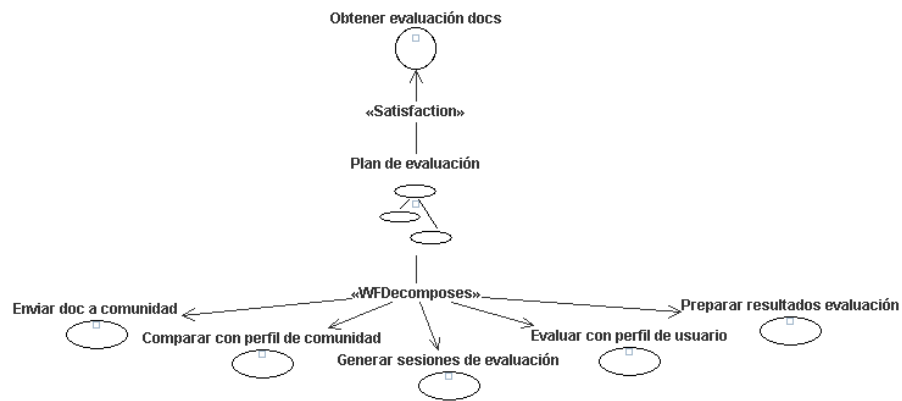
INGENIAS

14

Ejemplo de desarrollo orientado a agentes

2. Descomponer los OBJETIVOS en subOBJETIVOS y TAREAS

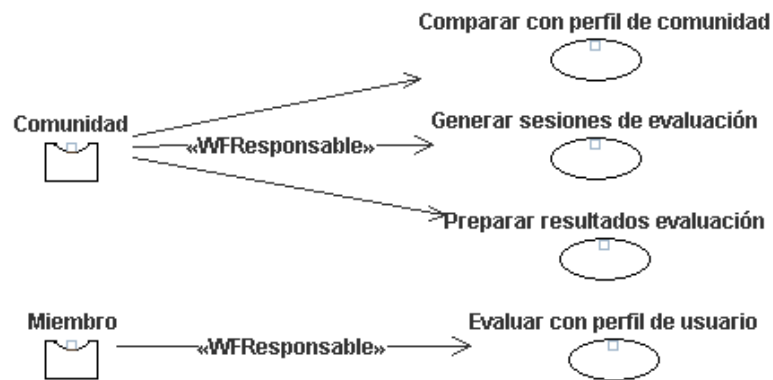
- Del análisis al diseño:
 - TAREAS definen procedimientos para satisfacer OBJETIVOS



Ejemplo de desarrollo orientado a agentes

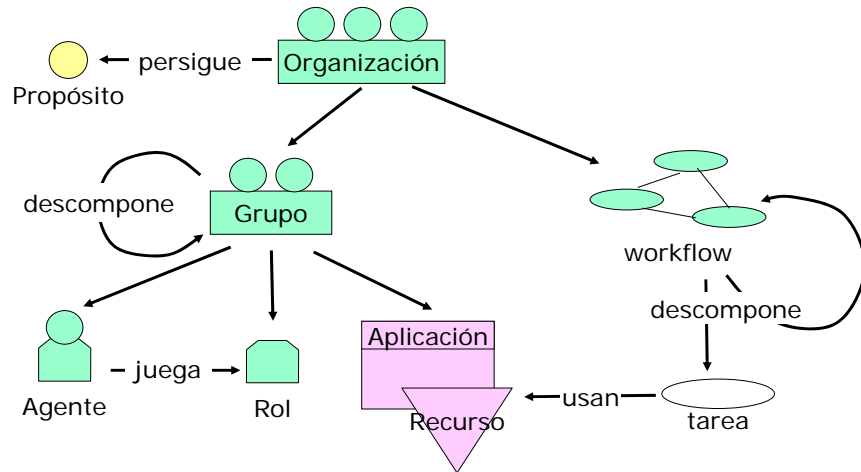
3. Asignar OBJETIVOS a ROLES en la ORGANIZACIÓN

- Diseño:
 - ROLES definen servicios (responsabilidades y funcionalidad esperada)



Modelo de organización

■ Descripción estructural



Juan Pavón Mestras, UCM

INGENIAS

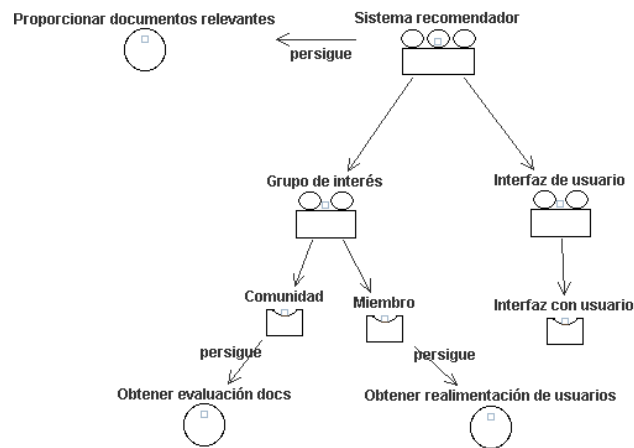
17

Ejemplo de desarrollo orientado a agentes

3. Asignar OBJETIVOS a ROLES en la ORGANIZACIÓN

■ Diseño:

- ORGANIZACIÓN define arquitectura (estructura, normas)



Juan Pavón Mestras, UCM

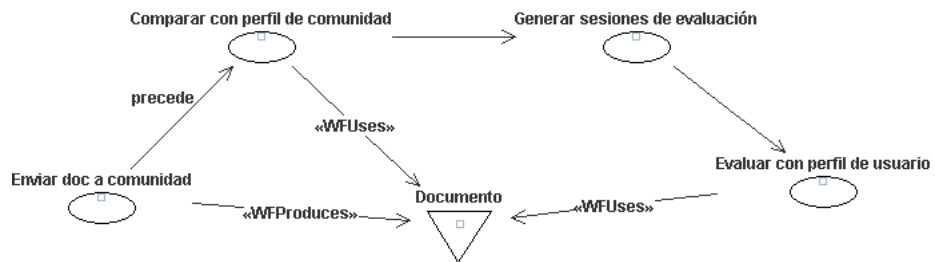
INGENIAS

18

Ejemplo de desarrollo orientado a agentes

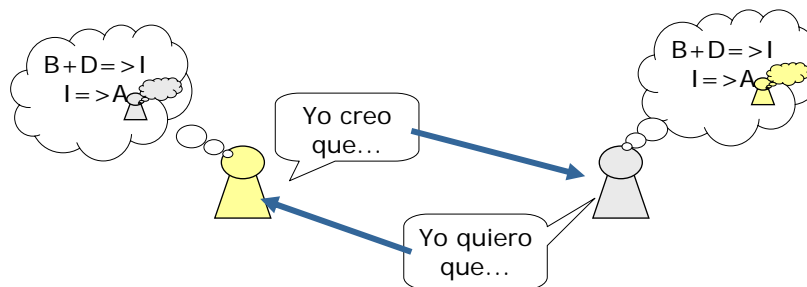
4. Definir WORKFLOWS e INTERACCIONES

- Diseño:
 - WORKFLOWS definen relaciones entre TAREAS, ROLES y RECURSOS



Modelo de interacciones

- Los agentes se **comunican** para
 - Mostrar a otros agentes su estado mental
 - Intentar modificar el estado mental de otros agentes
 - Basado en teoría de los actos del habla (*speech acts*):
 - peformativas del lenguaje



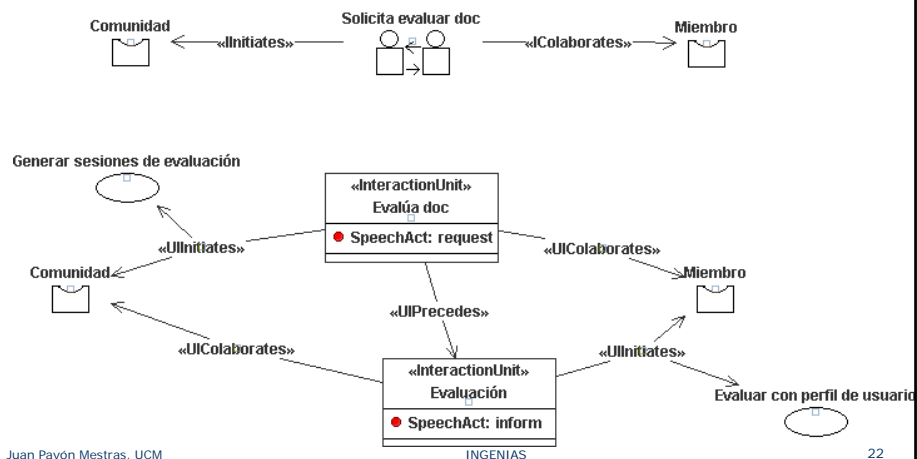
Definición de interacciones

- Qué actores participan en la interacción
 - Cada actor debe mostrar la razón por la que participa
 - Roles iniciador y colaboradores
- Definición de unidades de interacción
 - Mensajes, actos de habla
- Orden de las unidades de interacción
 - Protocolos: contract net, FIPA request, específicos.
 - Diagramas de protocolos AAML
- Acciones ejecutadas en la interacción
 - Criterios para decidir cuándo ejecutar una tarea
 - Consecuencias de la ejecución de una tarea
- Definición del contexto de la interacción
 - Objetivos que persigue la interacción
 - Estado mental de los participantes
- Modelo de control
 - Mecanismos de coordinación

Ejemplo de desarrollo orientado a agentes

4. Definir WORKFLOWS e INTERACCIONES

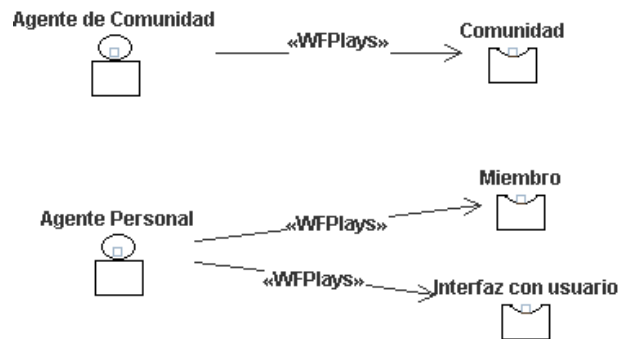
- Diseño:
 - INTERACCIONES definen cómo se comunican los ROLES



Ejemplo de desarrollo orientado a agentes

5. AGENTES desempeñan ROLES

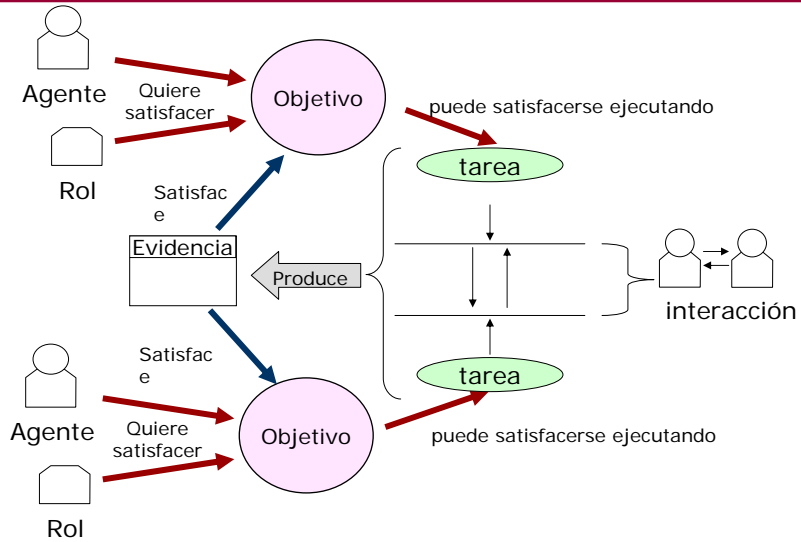
- Diseño:
 - AGENTES tienen capacidades para desempeñar ROLES
 - Cada AGENTE puede tener una estrategia diferente para satisfacer los objetivos del mismo ROLE



Ejemplo: diseño de los agentes

- Descripción de las capacidades de los agentes
 - Autonomía
 - El agente puede decidir por sí mismo qué agentes contactar para el proceso de compra de billete. Su propósito es asistir al usuario
 - Inteligencia
 - El agente decidirá, de acuerdo a ciertos parámetros (tiempo de respuesta, precio, calidad del cine, etc.) si le interesa o no aceptar un trato
 - Una manera de hacerlo es elaborando árboles de decisión y un algoritmo de inducción (como ID3) para determinar los criterios de selección de un buen cine
 - Además el agente podría ser capaz de aprender nuevas tareas
- En INGENIAS todo ello se hace gestionando y procesando un Estado Mental

Control del agente



Ejemplo de desarrollo orientado a agentes

6. Generación de código

- Simulación: validar el comportamiento del sistema multi-agente
 - INGENIAS Agent Framework (IAF)
- Realización e implantación de un sistema ejecutable
 - Módulos de generación de código del IDK
- Generación de documentación
 - Módulo de generación de documentación HTML del IDK

Proceso de desarrollo orientado a agentes

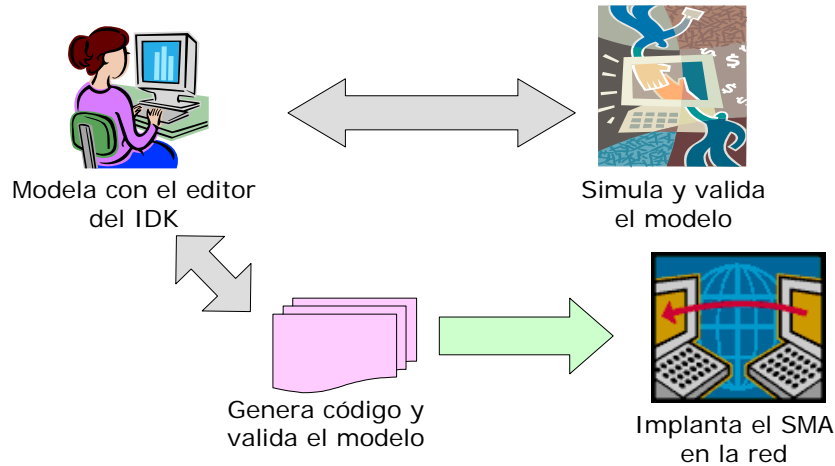
- Hay otras posibilidades
 - Centrarse y comenzar con los workflows
 - Cuando la organización está orientada a procesos
 - Centrarse en la coordinación y las interacciones
 - Cuando el problema es la definición de un algoritmo distribuido
 - Sistemas cooperativos
 - Centrarse en el entorno
 - Sistemas empotrados
 - Robótica
 - Centrarse en los actores
 - Simulación social

Proceso de desarrollo con INGENIAS

- El uso del IDK determina un proceso de desarrollo característico, que es MDD, que complementa procesos software
 - Proceso centrado en el modelado
 - La implementación se deriva de los modelos mediante módulos del IDK para generación de código
 - Aparte del role del desarrollador clásico (usando CASE tools) hay que considerar el role del ingeniero de metamodelos
 - Puede extender el lenguaje de modelado y realizar nuevos módulos del IDK

Actores en un desarrollo con INGENIAS (1)

- El modelo de desarrollo con INGENIAS
 - El desarrollador de SMA: realiza la aplicación



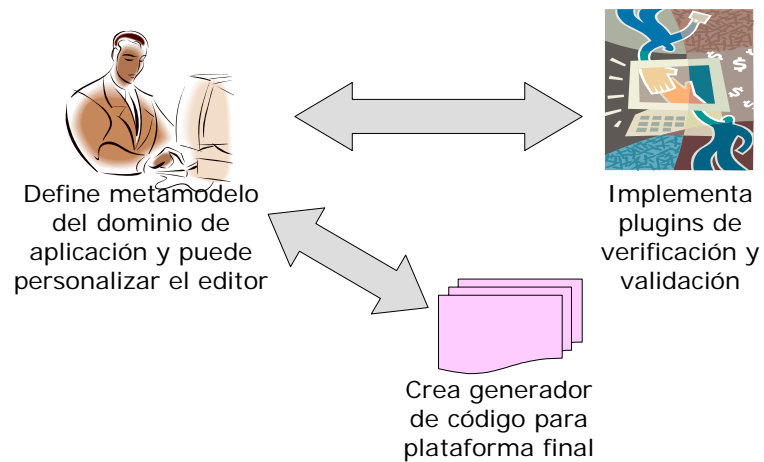
Juan Pavón Mestras, UCM

INGENIAS

29

Actores en un desarrollo con INGENIAS (2)

- El modelo de desarrollo con INGENIAS
 - Ingeniero INGENIAS: prepara las herramientas



Juan Pavón Mestras, UCM

INGENIAS

30

INGENIAS Development Kit (IDK)

- Editor de modelos
 - Herramienta visual (notación INGENIAS)
 - Inicialmente basado en herramienta de meta-modelado (METAEDIT+)
 - Actualmente 100% Java
 - Generación de modelos siguiendo los meta-modelos
 - Integración con agentes (en desarrollo)

- Módulos:
 - Para la generación de código
 - Plantillas configurables, marcadas con XML, para distintas plataformas de agentes
 - Jade, Robocode, Servlets, Agentes gracia!
 - Para verificación y validación de especificaciones
 - Para generar documentación (HTML)
 - Armazón para desarrollar módulos personalizados

Editor del IDK

- El editor del IDK permite
 - Crear y modificar modelos de SMA
 - Generar documentación (HTML)
 - Sacar snapshots de los diagramas para utilizarlos en otras aplicaciones
 - Procesar las especificaciones mientras se están generando con el editor o una vez grabadas en un fichero
 - Introducir explicaciones en lenguaje natural de los diferentes diagramas y de cada elemento en los diagramas, así como añadir etiquetas de texto

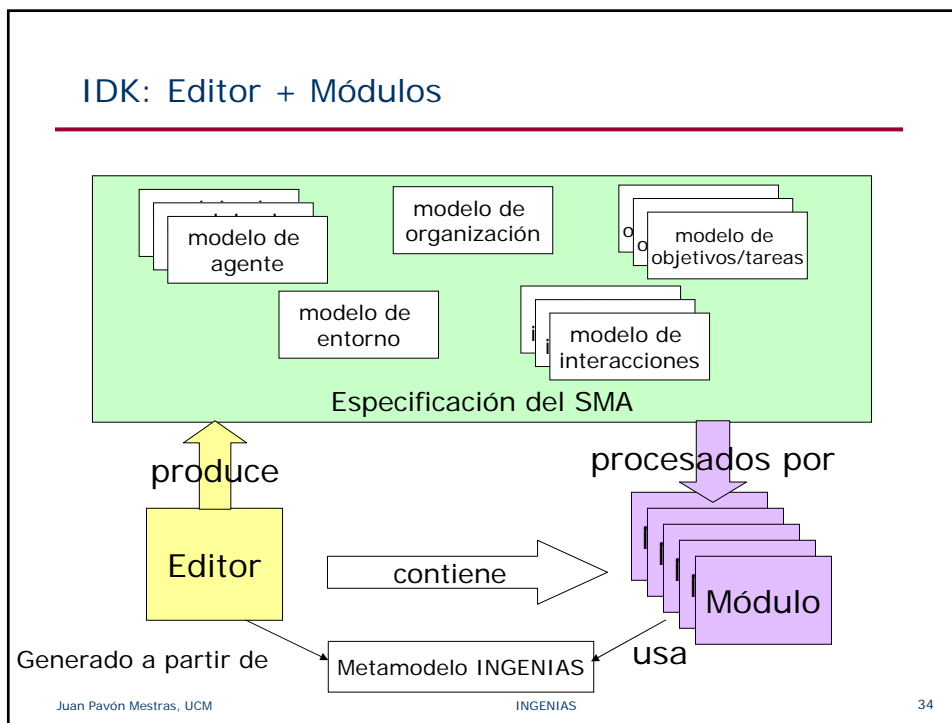
- Invocar módulos de generación de código, validación y verificación de las especificaciones

Editor del IDK

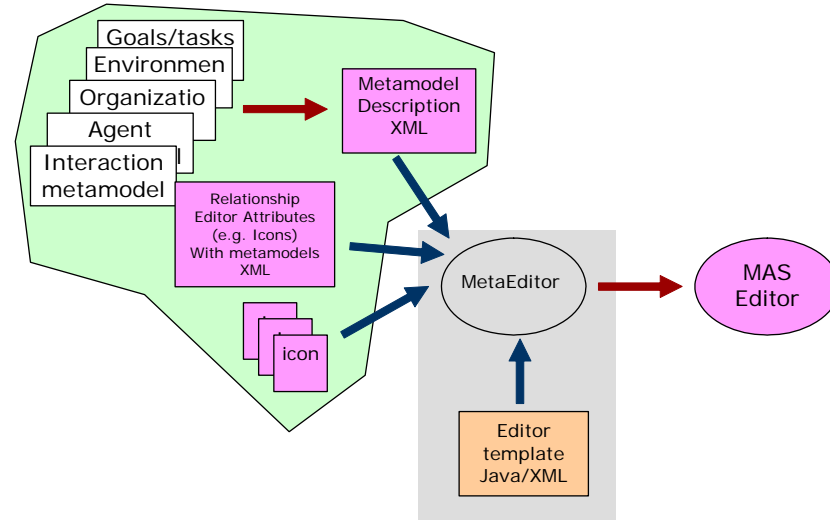
Juan Pavón Mestras, UCM

33

IDK: Editor + Módulos



Generación del editor

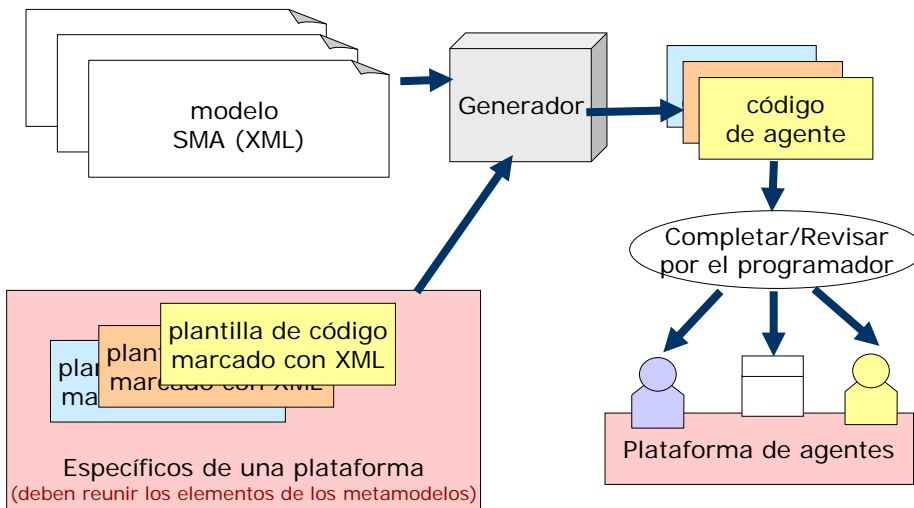


Juan Pavón Mestras, UCM

INGENIAS

35

Generación de código



Juan Pavón Mestras, UCM

INGENIAS

36

Resumen

- INGENIAS proporciona
 - Una notación para modelar un SMA desde cinco puntos de vista:
 - Organización
 - Agente
 - Objetivos/Tareas
 - Interacciones
 - Entorno
 - Modelos de proceso de desarrollo dirigido por modelos
 - Herramientas de soporte: Ingenias Development Kit (IDK)
<http://ingenias.sourceforge.net>
 - Generación de código sobre distintas plataformas
 - En el futuro tiene que evolucionar para:
 - Mejorar usabilidad y documentación
 - Facilitar trabajo en equipo (esto requiere, entre otras facilidades, control de versiones)
 - Identificación de patrones de diseño orientado a agentes

Bibliografía

- Ana Mas (2005). *Agentes Software y Sistemas Multiagente. Conceptos, Arquitecturas y Aplicaciones*. Pearson-Prentice Hall
- Brian Henderson-Sellers y Paolo Giorgini, editores (2005). *Agent-Oriented Methodologies*. Idea Group Publishing

