

# Sistemas MultiAgente

O-MaSE (AgentTool)

**Daniel Alfaro**

# ¿Qué es O-MaSE?

- Organization - based Multiagent System Engineering.
- Evolución de MaSE:
  - Diseñada para desarrollar Multiagentes de propósito general.
  - Define metas a partir de lo que se necesita.
  - Define roles para satisfacer las metas.
  - Define clases basadas en los roles.

# ¿Qué es O-MaSE?

- Basado en OPEN Process Framework.
- Definido en términos de:
  - Metamodelo,
  - Un conjunto de Fragmentos metodológicos,
  - Un conjunto de directrices.

# Metamodelo

- El metamodelo de O-MaSE, determina un conjunto de análisis, diseño, conceptos de implementación y un conjunto de restricciones entre ellas.
- Basado en un enfoque organizacional.
- Organización: Metas, Roles, Agentes, Modelo de dominio, políticas(reglas).

# Fragmentos metodológicos

- Define como un conjunto de análisis y diseño de producto que puede ser creado y usado dentro de la estructura (Framework).

# Directrices

- Las directrices definen como los fragmentos metodológicos se pueden combinar para crear una instancia completa de la metodología O-MaSE.

# AgentTool(1)

- Es un entorno de desarrollo gráfico basado en Java, el cual permite a los usuarios analizar, diseñar e implementar sistemas multiagente.
- Diseñado para dar soporte O-MaSE.
- Se define el comportamiento del sistema de manera gráfica utilizando la flexible metodología de O-MaSE.

# AgentTool(2) - Instalación

- Para instalar AgentTool se debe tener previamente :
  - Java Runtime Environment 1.5.0 o superior.
  - Eclipse 3.4.2, o superior(Ganymede).
- Instalar nuevo software (help)
  - Bajar e instalar todos los paquetes de "**<http://agenttool.cis.ksu.edu/update/>**".



# Instalar librería O-MaSE

- Para instalar la librería O-MaSE, se debe tener previamente:
  - Java Runtime Environment 1.5.0 or higher.
  - Eclipse SDK 3.4 (Ganymede).
  - GEF SDK 3.2 or above.
- [Instalación](#)

# AgentTool + librería O-MaSE

The screenshot displays the AgentTool interface with the O-MaSE library. The interface is divided into several panes:

- Library (1):** A tree view on the left showing the O-MaSE library structure, including Method Content, Content Packages, Roles, and Tasks.
- Example 1 (2):** A tree view in the center showing the task structure for Example 1, including Inception Phase, iter1, Requirements, Analysis, iter2, and Design.
- APE - Process Management (5):** A tree view on the right showing the process management structure for Example 1, including Inception, iter1, Requirements, Analysis, iter2, and Design.
- APE - Task Constraints (3):** A panel on the right showing task constraints for the 'new\_task' task, including Mandatory Input(s) (GMOs), Optional Input(s) (Role Model, Capability Model), and Precondition(s) (Role Model and Capability Model).
- GoalDiagram (6):** A diagram showing a goal hierarchy with a root goal '«Goal» 0 Goal0' and two child goals '«Goal» 1 Goal1' and '«Goal» 2 Goal2'.
- APE - Process Consistency (4):** A table at the bottom showing process consistency information.

Index	Description	Delivery Process	Path
(3)	new_task	The following workproduct(s) is(are) required :=GMOs, Role Model	Test1 /Users/jga

# Conclusiones

- O-MaSE -> dinámica y personalizable, el diseñador elige los módulos que considere necesarios para trabajar en la definición de su SMA.
- AgentTool -> Mantiene centralizada la especificación de los SMA.