

# Synapsis - Middleware per l'integrazione di Game Engine e Sistemi Multi-Agente

Tesi in: Sistemi Autonomi

*Relatore:*

Chiar.mo Prof.  
Andrea Omicini

*Correlatore:*

Dott. Stefano Mariani

*Presentata da:*

Luca Pascucci

ALMA MATER STUDIORUM – Università di Bologna  
Campus di Cesena

10 Ottobre 2019

# L'obiettivo della tesi

## Fase 1

Studiare lo stato dell'arte delle Game Engine (GE) e dei Sistemi Multi-Agente (MAS)

## Fase 2

Analizzare i due sistemi con lo scopo di:

- Evidenziare opportunità per colmare le lacune concettuali/tecniche dei due mondi
- Rilevare astrazioni e meccanismi che forniscono supporto nel riformulare le astrazioni mancanti del MAS e della GE

## Fase 3

Proporre un'infrastruttura di associazione e comunicazione tra MAS e GE, rispettandone il disaccoppiamento e l'integrità concettuale delle astrazioni.

# Concetti preliminari

## Sistema Multi-Agente

Paradigma general-purpose utilizzato per realizzare sistemi complessi. Le astrazioni principali sono:

- Agenti: Entità autonome in grado di comunicare, possono essere situati, intelligenti;
- Società: Gruppo di agenti che cooperano / competono per realizzare goals;
- Ambiente: Il "contenitore" in cui gli agenti eseguono, che li influenza e ne viene influenzato.

## Game Engine

Struttura general-purpose multiplatforma orientata verso ogni aspetto della progettazione e dello sviluppo di videogiochi

# JaCaMo

Framework per la programmazione orientata agli agenti che combina tre tecnologie già affermate e sviluppate da diversi anni

## Jason

Piattaforma di sviluppo agenti basati sull'architettura BDI, implementa AgentSpeak(L).

## CARTAgO

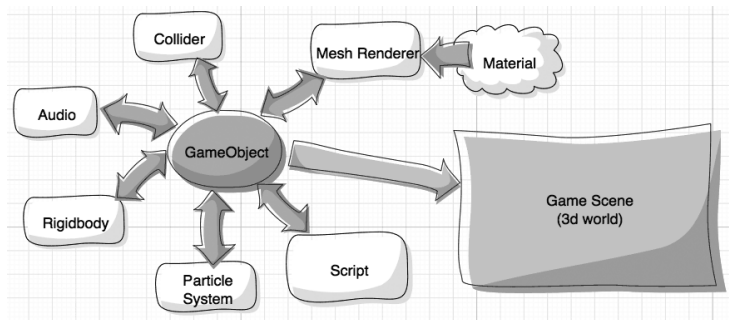
Modello per la progettazione dell'ambiente MAS, basato sul concetto di workspace ed artefatti.

## Moise

Meta-modello organizzativo per MAS basato sulle nozioni di ruoli, gruppi e missioni.

# Unity

Game Engine cross-platform utilizzata per la creazione di videogiochi (sia 2D che 3D), ambienti virtuali e simulazioni interattive.



# Astrazioni concettuali

## Entità

Componente divisibile in due parti, mente e corpo, che collegate riescono a trasmettersi informazioni utilizzate dalla mente per raggiungere i propri obiettivi e dal corpo per diventare "attivo" nell'ambiente in cui si trova.

## Corpo

Componente associato alla nozione di GameObject per avere una rappresentazione fisica dell'entità da realizzare.

## Mente

Componente autonomo che interagisce con l'ambiente per svolgere i propri compiti associato alla nozione di Agente.

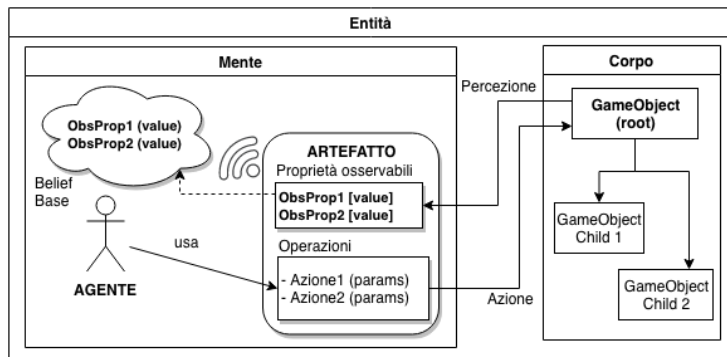
# Integrazione concettuale

## Agente BDI

Focus sul concetto di **Belief** e **Action**

## Artefatto

Focus sul concetto di **Proprietà Osservabile** e **Operazione**

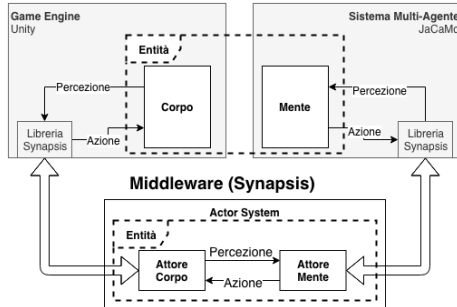


# Middleware Synopsis

## Architettura di sistema

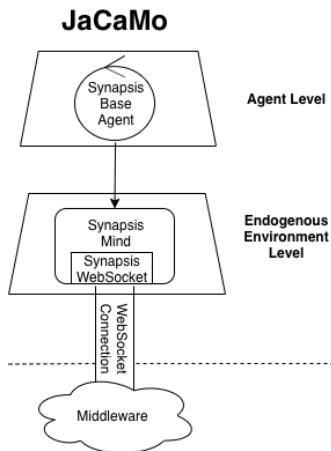
### Sistema ad Attori (Akka, Play)

- attori = entità computazionali (re)attive
- message passing = asincronia, mailbox
- fault-tolerance, scalabilità





# Synopsis: JaCaMo client



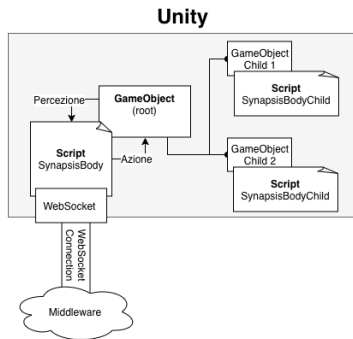
## Contenuto

- Agente "SynapsisBaseAgent"
- Arfettato "SynapsisMind"

## API

- Invio azione generica
- Presenza di azioni già delineate (Trova, Vai a, Prendi ...)
- Gestione delle percezioni ricevute dal corpo
- Connessione WebSocket al middleware

# Synopsis: Unity client



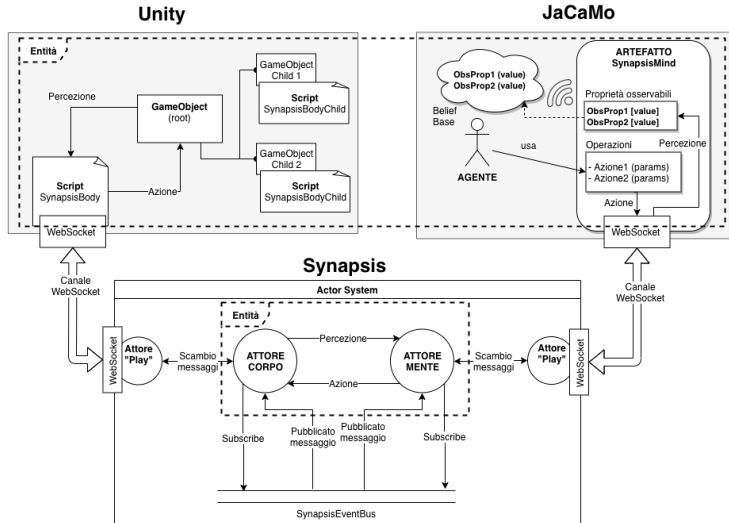
## Contenuto

- Script "SynopsisBody"
- Script "SynopsisBodyChild"

## API

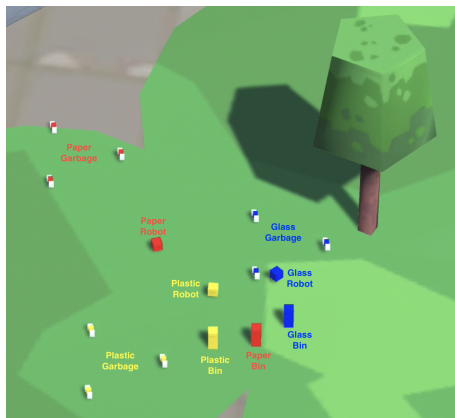
- Invio percezione generica
- Presenza di percezioni già delineate (Trovato, Arrivato, Preso ...)
- Invio automatico di percezioni fisiche (contatto con altri oggetti in scena)
- Connessione WebSocket al middleware

# Synopsis: zoom architettura



# Recycling Robots

Scenario contenente robot con il compito di riciclare la spazzatura presente nell'ambiente portandola al rispettivo bidone.



# Riassumendo...

## Conclusioni

- Definite linee guida per l'integrazione di Game Engine e Sistemi Multi-Agente
- Realizzato middleware e librerie che permettono la costruzione di scenari relativamente complessi

## Possibili sviluppi futuri

- Integrare il modello di coordinazione degli agenti tramite spazio di tuple e primitive Linda
- Realizzare nuove librerie per espandere la lista di GE e MAS collegabili
- Utilizzare le caratteristiche dei sistemi presenti per realizzare lo stesso scenario distribuito su più elaboratori

# Synapsis - Middleware per l'integrazione di Game Engine e Sistemi Multi-Agente

Tesi in: Sistemi Autonomi

*Relatore:*

Chiar.mo Prof.  
Andrea Omicini

*Correlatore:*

Dott. Stefano Mariani

*Presentata da:*

Luca Pascucci

ALMA MATER STUDIORUM – Università di Bologna  
Campus di Cesena

10 Ottobre 2019