

# Simulazione di agenti BDI basati su Prolog in Alchemist

Tesi in: Sistemi Autonomi

*Relatore:*

Chiar.mo Prof.  
Andrea Omicini

*Presentata da:*

Filippo Nicolini

*Correlatori:*

Dott. Ing. Danilo Pianini  
Dott. Giovanni Ciatto

ALMA MATER STUDIORUM – Università di Bologna  
Campus di Cesena

12 Dicembre 2019

# L'obiettivo della tesi

## Obiettivo della tesi

Unificare piattaforme orientate alla programmazione di agenti con ambienti di simulazione di agenti.

## Interprete Prolog

Estensione modello BDI di AgentSpeak sfruttando tuProlog per poter implementare interpreti sia su ambienti reali che simulati.

## Interprete Alchemist

Realizzazione di un interprete nel simulatore Alchemist che utilizza le logiche del ciclo di ragionamento di Jason.

# AgentSpeak, modello BDI, tuProlog

## Modello BDI

Modello BDI (Beliefs, Desires, Intentions) implementa gli aspetti principali del ragionamento umano per programmare agenti intelligenti.

## AgentSpeak

Linguaggio orientato agli agenti basato su modello BDI e programmazione logica.

## tuProlog

Libreria che permette di utilizzare Prolog all'interno di applicazioni e infrastrutture distribuite sfruttando un core minimale.

# Interprete tuProlog di AgentSpeak

## Formalizzazione

Estensione di AgentSpeak realizzando un'interprete attraverso definizione di alcune sintassi.

## API

- inizializzazione agente
- invocazioni verso linguaggio OO
- gestione 'belief base'
- gestione eventi e posizionamento

# Alchemist

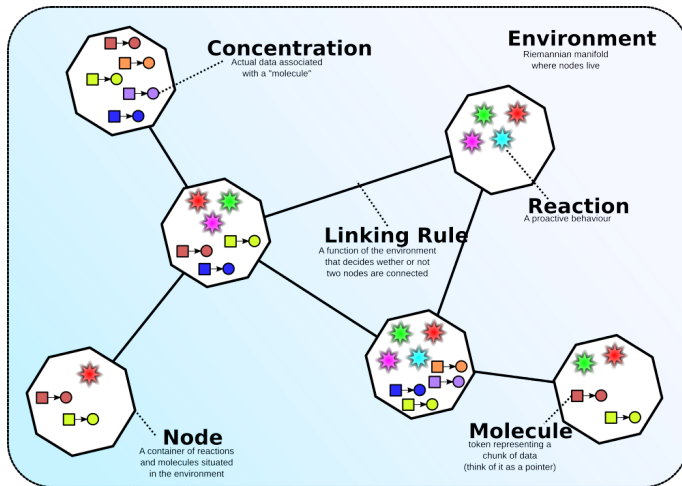
## Alchemist

Simulatore per il calcolo pervasivo, aggregato e naturale che si basa su un meta-modello flessibile e che permette implementazioni di modelli diversi tra loro.

## Meta-modello

- Node
- Environment
- Reaction (Time Distribution, Condition, Action)
- Linking Rule

# Meta-modello



# Spazi di tuple

## LINDA

Modello di coordinazione e comunicazione tra processi paralleli con memoria associativa, virtuale, condivisa.

## Spatial Tuples

Estensione del modello base di tuple per i sistemi distribuiti multi-agente.

Operazioni consentite

- in: legge tupla e la consuma
- rd: legge tupla senza consumarla
- out: inserisce tupla

# Ciclo di ragionamento

## Percezioni

Informazioni ricevute tramite un apparato con le quali l'agente percepisce i cambiamenti dell'ambiente.

## Eventi

Sono relativi a percezioni che l'agente ha ricevuto e possono essere catturati dall'agente.

## Piani

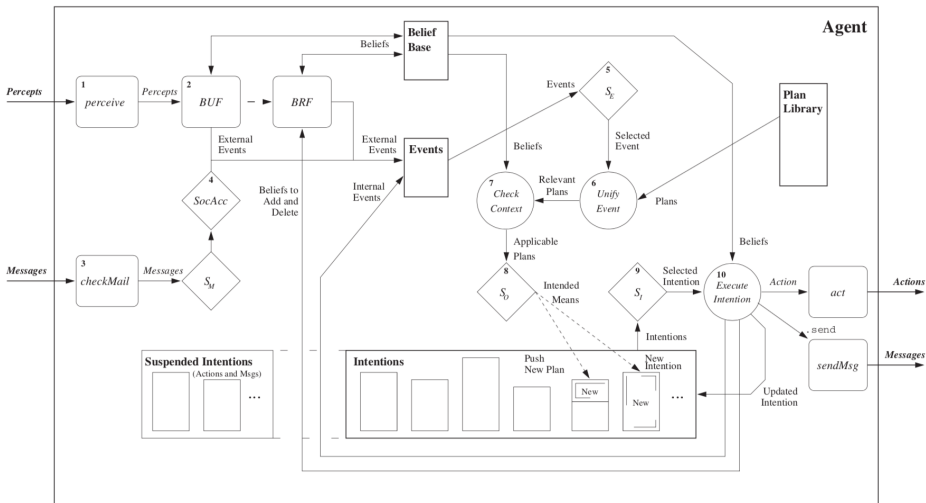
Definiscono come l'agente agisce per raggiungere goal.

## Intenzioni

Operazioni che l'agente vuole eseguire per portare a termine un certo goal.



# Ciclo di ragionamento



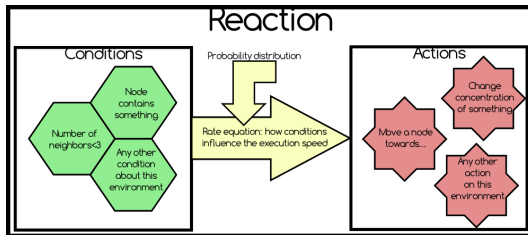
# Unione modelli

## Mapping

Environment → Spazio agenti

Nodo → Contenitore agenti

Reazione → Agente



## Caratteristiche mapping

Ricercata la massima espressività lavorando su più strati: nodo può essere inteso come device in cui operano più agenti.

# Scenario

## Goldminers

Un gruppo di minatori deve recuperare pepite d'oro da miniere sparse nell'ambiente e riportarle in un deposito.

## Entità → ruoli

All'interno del problema si individuano le seguenti entità:

- minatori → agenti
- pepite → tuple
- miniere → spazi di tuple
- deposito → agente

# Realizzazione

## Minatore

Comportamento diviso in 4 stati:

- **ricerca**: spostamento casuale emettendo richieste di tuple;
- **ricezione tupla**: salva posizione miniera e si dirige al deposito;
- **arrivo deposito**: invio pepita e si dirige posizione miniera salvata;
- **arrivo miniera**: torna stato ricerca.

## Miniera

Istanza N tuple all'inizializzazione e risponde alle richieste dei minatori.

## Deposito

Statico nell'ambiente, riceve la pepita tramite un messaggio.

# Conclusioni e lavori futuri

## Conclusioni

- Realizzazione di interprete per programmare e simulare agenti
- Forte espressività modello e interprete (Prolog e OO)
- Estensione agenti in spazi di tuple

## Lavori futuri

- Implementazione interprete OO in piattaforme per ambiente reale
- Ricerca per miglorie sia nell'interprete Prolog che nell'interprete OO

# Simulazione di agenti BDI basati su Prolog in Alchemist

Tesi in: Sistemi Autonomi

*Relatore:*

Chiar.mo Prof.  
Andrea Omicini

*Presentata da:*

Filippo Nicolini

*Correlatori:*

Dott. Ing. Danilo Pianini  
Dott. Giovanni Ciatto

ALMA MATER STUDIORUM – Università di Bologna  
Campus di Cesena

12 Dicembre 2019