



ARES-MAS

Autonomous Rovers for Exploration and Safety

A Hybrid Symbolic System for Multi-Agent Planetary Exploration

Intelligent Systems Engineering Project

Marco Costantini

1. Separation of Concerns

Ipotesi architetturale: Nessun singolo paradigma simbolico è sufficiente per modellare l'intera complessità di un MAS.

Integrazione delle 3 Tecnologie Core



COMPONENTE 01

Jason / AgentSpeak(L)

BDI Architecture

Gestisce l'intero ciclo Belief-Desire-Intention. Permette agli agenti di agire in modo autonomo per raggiungere i propri obiettivi. Inoltre, gestisce la collaborazione tra i rover.



COMPONENTE 02

tuProlog

Logic-Based Safety Guardrails

Motore di inferenza logica per la valutazione a runtime di vincoli di sicurezza (es. evitare collisioni o esaurimento batteria).



COMPONENTE 03

STRIPS / Kotlin

Automated Forward Planning

Motore logistico basato su una ricerca in ampiezza (BFS). Interviene, su delega del livello BDI, per calcolare rotte di navigazione complesse, generando un piano che l'agente esegue fisicamente passo dopo passo.

2. Proprietà MAS



PROPRIETÀ 01

Autonomia

Gli agenti mantengono una propria base di conoscenza (belief base) e prendono decisioni in modo indipendente, valutando autonomamente le aste e perseguendo i propri task senza uno scheduler centrale



PROPRIETÀ 02

Coordinazione

Non essendoci memoria condivisa tra gli agenti, ogni operazione di coordinamento è mediata dallo scambio di messaggi standardizzati FIPA-ACL.



PROPRIETÀ 03

Situatedness

Gli agenti sono fisicamente ancorati a una griglia spaziale 2D: ogni ragionamento e decisione è rigorosamente vincolata alle reali limitazioni fisiche e di batteria imposte dall'ambiente.



PROPRIETÀ 04

Reattività

Gli agenti sono orientati agli obiettivi (esplorano o estraggono proattivamente), ma reagiscono dinamicamente ai cambiamenti del mondo: un'azione fisica negata dall'ambiente innesca immediatamente un piano di recupero.

3. Deliberazione BDI e Contract Net Protocol

Workflow decisionale: Architettura integrata in AgentSpeak per la gestione di percezioni e aste

Fasi Operative e funzionamento dell'asta



PERCEZIONE

Scoperta Geospaziale

Il riconoscimento avviene fisicamente tramite l'operatore di *belief revision* (`--at(X,Y)`): il piano di scoperta si attiva solo quando la posizione spaziale dello Scout coincide esattamente con quella del minerale.



CNP - PASSO 01

Broadcast CFP

Inizio Negoziazione

Una volta bloccata la risorsa tramite lock esclusivo nell'ambiente, lo Scout avvia il protocollo FIPA-ACL inviando un messaggio di broadcast a tutti gli Harvester per annunciare il task di estrazione.



CNP - PASSO 02

Calcolo Euristicica

Distanza di Manhattan

Alla ricezione della CFP, ogni Harvester calcola autonomamente la propria offerta di estrazione. L'offerta è ancorata allo stato fisico del mondo utilizzando l'euristica della Distanza di Manhattan tra la base e il target.



CNP - PASSO 03

Valutazione Offerte

Operatori Logici Jason

Selezione dell'offerta vincente tramite operatori nativi Jason (`.findall` / `.min`). Infine, notifica l'assegnazione del task esclusivamente all'agente vincitore.

4. Integrazione tuProlog

Prevenzione di collisioni ed esaurimento risorse: Motore logico temporaneo e regole dichiarative



MOVEMENT GUARDRAIL

safe_to_move/2

Valutato prima di ogni mossa fisica. La query viene eseguita su un **motore Prolog**.

I fatti relativi al pericolo (`hazard(X, Y)`) sono costantemente sincronizzati con la tempesta di sabbia.

Previene collisioni fatali garantendo:

`\+ hazard(X, Y)`



EXTRACTION GUARDRAIL

safe_to_extract/3

Funge da blocco di sicurezza di ultima istanza prima dello scavo.

Prolog verifica 3 requisiti vitali:

- **Ruolo corretto dell'agente**
- **Cella libera dalla tempesta**
- **Energia residua sufficiente**



GESTIONE FALLIMENTO

Recovery Autonomo

Se la query Prolog fallisce, l'Ambiente (Kotlin) nega l'esecuzione restituendo **false**.

L'agente Jason intercetta il fallimento dell'azione fisica.

Innesca autonomamente un **piano di recovery** per ripristinare il flusso decisionale.

5. Dinamiche Ambientali

Gestione Ambientale e Concorrenza: Thread di background, modellazione fisica e protezione della teoria Prolog



GESTIONE AMBIENTALE

Hazard Dinamico (Sandstorm)

Gestito da un thread in background indipendente dal ciclo BDI.

A ogni spostamento, la teoria Prolog viene ricostruita dinamicamente.

Protetto tramite un blocco `synchronized(engineLock)` per evitare letture corrotte durante i controlli di sicurezza.



SITUATEDNESS

Modello Fisico

La batteria non è un concetto astratto, ma è gestita nativamente nell'ambiente Kotlin.

Tracciata tramite la struttura dati `mutableMapOf<String, Int>`.

Il consumo fisico viene interrogato a runtime dal guardrail `tuProlog` prima di ogni estrazione.



ATOMIC CLAIMING

Gestione Concorrenza

I minerali attraversano tre stati rigidi:

`Available Claimed Extracted`

Le race conditions tra più Scout che scoprono la stessa risorsa sono prevenute da un'operazione atomica protetta da `synchronized(resourceLock)`.

6. Flusso di Esecuzione

01

FASE 1

Scoperta e Claiming

Lo Scout scopre fisicamente la risorsa.

L'**Environment** risolve la concorrenza con un blocco atomico per garantire l'accesso esclusivo.

02

FASE 2

Negoziazione FIPA-ACL

Lo Scout avvia il **Contract Net Protocol (CFP)**.

Gli Harvester calcolano le offerte e lo Scout assegna il task al miglior offerente.

03

FASE 3

Esecuzione e Guardrail

L'Harvester naviga verso il target e tenta l'estrazione.

Le azioni fisiche vengono intercettate e validate dinamicamente dalle regole Prolog (**safe_to_move** e **safe_to_extract**).

04

FASE 4

Fallback Logistico

Conclusa l'estrazione (o in caso di fallimento per batteria scarica), l'Harvester invoca l'Internal Action **STRIPS**.

Genera la rotta di ritorno **BFS** ed esegue il piano passo dopo passo.

7. Explainability (XAI) e Dashboard Streamlit

TRACCIAMENTO CAUSALE

Serializzazione Causale (`events.jsonl`)

Ogni cambiamento di stato (**CLAIM**, **NEGOTIATION**, **VIOLATION**) viene salvato come oggetto JSON. Gli eventi sono sincronizzati tramite un **tick counter** crescente che garantisce l'ordinamento causale.

RIPRODUZIONE TEMPORALE

Time Machine

Il frontend non comunica con Jason/Prolog. Legge i log e filtra dinamicamente l'array degli eventi in base al tick selezionato dall'utente (`[e for e in events if e['tick'] <= current_tick]`), ricostruendo visivamente percorsi, minerali e hazard.

NARRATIVA E ANALISI

Mission Log e CNP Analysis

Traduzione del JSON grezzo in narrativa **human-readable** e renderizzazione di grafici a barre per il confronto delle bid del Contract Net Protocol.

ISPEZIONE LOGICA

AI Safety Reasoning

Il sistema distingue tra violazioni **safe_to_move** e **safe_to_extract**. Costruisce uno **pseudo proof-tree** visuale che espone la condizione logica fallita (es. soglia batteria < 20 o coordinate bloccate) estraendo i dati strutturati dai log.

8. Esecuzione Live: MAS Console



REAL-TIME SYSTEM MONITORING

Integrazione Ibrida in Real-Time

La console unifica i log dell'ambiente fisico, le azioni BDI e l'inizializzazione del motore tuProlog.



FIPA-ACL PROTOCOL

Negoziabile Tracciabile

Dimostra l'esecuzione del Contract Net Protocol, dal calcolo dell'euristica (Distanza di Manhattan) fino alla raccolta e valutazione delle Proposals.



SAFETY & EMERGENCY RECOVERY

Guardrail e Recovery Attivo

Blocco effettivo delle azioni non sicure (**GUARDRAIL ACTIVE**) e innesco automatico dei piani di emergenza (**Triggering emergency return**).



LOGICAL PLANNING

Invocazione STRIPS

Mostra il log di attivazione dell'Internal Action logistica e il piano spaziale restituito all'agente prima dell'esecuzione iterativa.

```
MAS Console - ares_mas

MAS Console - ares_mas

Runtime Services (RTS) is running at 127.0.1.1:35431
Agent mind inspector is running at http://127.0.1.1:3272
Scenario loaded: 2 entities found.
tuProlog Guardrail Initialized successfully.
[harv2] Harvester online, awaiting auctions...
[harv1] Harvester online, awaiting auctions...
[scout1] Scout online, starting exploration!
[scout2] Scout online, starting exploration!
Action executed by scout1: move to [1, 1] {Battery: 98}
Action executed by scout2: move to [8, 1] {Battery: 98}
Sandstorm moved to [6, 5] (1/8)
Action executed by scout1: move to [3, 4] {Battery: 96}
Action executed by scout2: move to [7, 4] {Battery: 96}
[scout1] I physically found a mineral during my patrol at [3,4]!
scout1 claimed mineral at [3, 4]
[scout1] Claim successful for [3,4]! Launching CFP...
[harv1] CFP Receipt from scout1 for the mineral to [3, 4]
[harv2] CFP Receipt from scout1 for the mineral to [3, 4]
[harv2] Calculated Manhattan distance is 7. Bidding cost: 24...
[harv1] Calculated Manhattan distance is 7. Bidding cost: 9...
GUARDRAIL ACTIVE: Movement of scout2 to [6, 5] blocked by tuProlog!
[scout2] Patrol route blocked by safety guardrail! Diverting to alternative target...
Action executed by scout2: move to [8, 8] {Battery: 94}
[scout1] Proposals collected: [offer(9,harv1),offer(24,harv2)]
[scout1] Winner: harv1 with cost 9
[harv1] I won the contract for [3, 4]! I'm going to extract...
Sandstorm moved to [6, 6] (2/8)
Action executed by scout1: move to [2, 7] {Battery: 94}
[scout1] Exploration completed (Route A)!
Action executed by harv1: move to [3, 4] {Battery: 19}
[harv1] Attempting to extract at [3, 4]...
GUARDRAIL ACTIVE: Extraction by harv1 at [3, 4] blocked — battery 19 < 20
[harv1] Extraction BLOCKED by tuProlog guardrail! Triggering emergency return.
STRIPS Engine activated for route [3, 4] to [0, 0]
[harv1] STRIPS returned the following plan: [move(3,3),move(3,2),move(3,1),move(3,0),move(2,0),move(1,0),move(0,0)]
Action executed by harv1: move to [3, 3] {Battery: 17}
Action executed by harv1: move to [3, 2] {Battery: 15}
Action executed by harv1: move to [3, 1] {Battery: 13}
Action executed by harv1: move to [3, 0] {Battery: 11}
Sandstorm moved to [5, 6] (3/8)
Action executed by harv1: move to [2, 0] {Battery: 9}
Action executed by harv1: move to [1, 0] {Battery: 7}
Action executed by harv1: move to [0, 0] {Battery: 5}
[harv1] Safely returned to base!
Mission complete signal from harv1 — sandstorm patrol halted.
```

9. ARES-MAS (Demo Live)

01

FOCUS 01

Esplorazione e Scoperta

Esplorazione situata dell'ambiente operativo e rilevamento dinamico delle risorse minerarie.

Tracciamento in tempo reale delle coordinate e dello stato fisico dell'agente Scout.

02

FOCUS 02

Negoziazione FIPA-ACL

Apertura ed esecuzione dell'asta mediante protocollo **Contract Net (CFP)**.

Valutazione euristica delle offerte degli Harvester e assegnazione distribuita del task.

03

FOCUS 03

Hard-Safety Guardrail

Intercettazione e validazione deterministica delle azioni tramite regole **tuProlog**.

Blocco immediato delle operazioni non sicure in presenza di hazard o autonomia critica.

04

FOCUS 04

Rotta di Emergenza

Attivazione dell'Internal Action logistica basata su pianificazione **STRIPS**.

Calcolo e rientro guidato alla base mediante percorso ottimo **BFS**.