

Sistema multi agente per la georeferenziazione dinamica in assenza di segnale GPS

Cristiana Casanova

12 Marzo 2011

Sommario

La georeferenziazione è attualmente eseguita attraverso i segnali provenienti dai satelliti GPS. La limitazione di questa tecnica è legata all'effettiva ricezione di tali informazioni: infrastrutture e ambienti chiusi rappresentano infatti un problema. Per poter ovviare a tale inconveniente si usano attualmente tecniche di IPS (Indoor Positioning System) e A-GPS (Assisted GPS), dove delle antenne georeferenziate a priori, sono in grado di propagare il segnale verso i dispositivi. Scopo del progetto è l'implementazione di sistemi GPS "intelligenti", in grado in ogni momento di ricostruire la propria posizione, indipendentemente dalla presenza o meno di segnale satellitare.

I ricevitori GPS saranno perciò entità attive, pertanto denominati dispositivi, in grado di riconoscere l'assenza di satelliti necessari alla triangolazione: essi provvederanno quindi a contattare altri ricevitori al fine di poter portare a termine la determinazione della propria posizione, senza la necessità di introdurre entità statiche di propagazione di quest'ultima. La georeferenziazione sarà possibile tramite un numero inferiore di satelliti, aggiungendo dispositivi GPS in modo tale da diminuire la latenza di inizializzazione grazie alla modellazione ad agenti, e così migliorarne le capacità di georeferenziazione.

Indice

1	Introduzione	3
1.1	GPS	3
1.1.1	Funzionamento	3
1.1.2	Trilaterazione	3
1.1.3	Trilaterazione 3D	4
1.1.4	Errori del sistema GPS	5
1.2	A-GPS	6
1.2.1	Funzionamento	6
1.3	IPS	6
2	Sistemi Multi-Agente	7
2.1	Definizione	7
2.2	Supporto MAS: il framework JADE	8
2.3	Architettura di JADE	8
2.4	Scambio di messaggi	10
2.5	Agenti in JADE	11
2.6	Behaviours	11
3	Sviluppo dell'applicazione	13
3.1	Obiettivi	13
3.2	JADE nell'applicazione	13
3.2.1	Modello di comunicazione	13
3.2.2	Yellow Pages	14
3.3	Comportamento dell'applicazione	15
3.4	Architettura	16
3.5	Agenti implementati	18
3.5.1	Main Agent	18
3.5.2	Simulated GPS Agent	19
4	Risultati ottenuti	19
5	Conclusioni	20

1 Introduzione

1.1 GPS

Il GPS consente di ricavare la propria posizione sul globo terrestre, in termini di coordinate geografiche, mediante l'elaborazione del segnale proveniente da una rete di 24 satelliti attivi in orbita attorno alla Terra, disposti in modo non geostazionario e in modo che da ogni punto del pianeta si riceva il segnale di almeno quattro satelliti.

1.1.1 Funzionamento

Il principio di funzionamento del GPS si basa sul calcolo della distanza del ricevitore (ovvero della nostra unità portatile) dai satelliti in orbita che riesce a rilevare. Tale calcolo consiste nel determinare il tempo necessario affinché il segnale trasmesso dal satellite venga ricevuto dal nostro GPS mobile.

Il GPS contiene al suo interno un orologio al quarzo che si sincronizza con gli orologi al cesio dei satelliti, i quali si sincronizzano a loro volta fra loro. Poiché la precisione di un orologio al cesio è molto maggiore rispetto a quella dell'orologio interno del GPS, la precisione richiesta si ottiene confrontando il segnale ed il tempo di percorrenza di più satelliti.

Per il calcolo della posizione, lo strumento deve ricevere il segnale di almeno tre satelliti, anche se si necessiterebbe di un quarto satellite per la verifica ed il calcolo dell'altitudine.

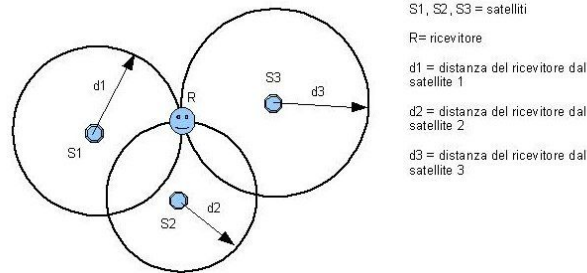
1.1.2 Trilaterazione

Un ricevitore GPS utilizza il processo di trilaterazione per determinare la propria posizione sulla superficie terrestre mediante la ricezione di segnali orari provenienti da tre satelliti del sistema GPS.

Oltre al segnale orario, ogni satellite della costellazione GPS emette segnali a intervalli regolari: questi segnali vengono ricevuti da dispositivi GPS che successivamente calcolano la distanza tra il dispositivo e ciascun satellite in base al ritardo tra il movimento di invio e quello di ricezione del segnale. I segnali, simili a onde radio, viaggiano alla velocità della luce, ma con un ritardo dovuto all'altitudine dei satelliti.

Dopo la ricezione delle distanze relative ad almeno tre satelliti, il dispositivo GPS può eseguire i calcoli in base al processo di trilaterazione.

Esso consente l'individuazione di una posizione geografica su una mappa in base alla distanza precisa da tre punti di riferimento diversi.



Come si può vedere in figura, la posizione da individuare è rappresentata dall'intersezione dei tre cerchi con al centro ciascuno dei punti di riferimento, considerando che il raggio di ogni cerchio corrisponde alla distanza da quest'ultimi.

Nella versione GPS, i calcoli vengono eseguiti nello spazio tridimensionale: il punto da individuare sarà quindi dato dall'intersezione di tre sfere il cui raggio è costituito dalla distanza da ciascuno dei tre satelliti. Se il dispositivo GPS è in grado di ricevere il segnale di un quarto satellite, è possibile ricontrollare le misurazioni.

1.1.3 Trilaterazione 3D

Consideriamo uno spazio euclideo tridimensionale, centrato al centro della Terra. Vogliamo trovare la posizione della ricevente, ossia $T(t_1; t_2; t_3)$, partendo dai dati trasmessi da tre satelliti, ossia la loro posizione $C_i (c_{i,1}; c_{i,2}; c_{i,3})$, e la loro distanza dalla ricevente, ossia ρ_i . La trilaterazione tridimensionale necessita di quattro punti di riferimento per trovare un quinto punto. La condizione che il punto cercato deve trovarsi sulla Terra ci permette però di determinare la posizione usando solo tre punti di riferimento, poiché l'altro può essere scartato con una semplice condizione.

Si procede quindi nel modo seguente:

- si trova un primo piano radicale delle sfere di centro C_1 e C_2 , notato π_1 ;
- si trova un secondo piano radicale delle sfere di centro C_1 e C_3 , notato π_2 ;
- si intersecano i piani π_1 e π_2 per trovare la retta $r = \pi_1 \cap \pi_2$;
- si interseca la retta r con una delle tre sfere. Si trovano così due punti;
- si tiene presente che il punto cercato deve trovarsi sulla Terra o nella sua atmosfera: in questo modo si elimina il secondo punto, che si trova nello spazio.

Il procedimento è illustrato nella figura seguente:

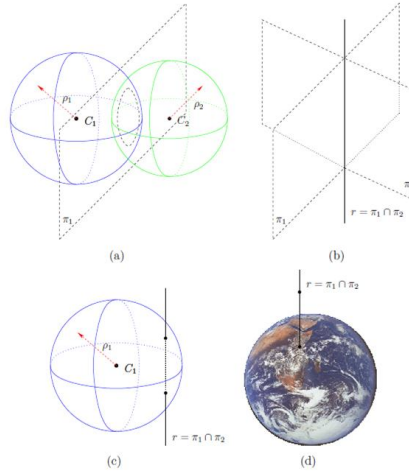


Figura 1: (a) Piano radicale che contiene i punti di intersezione delle due sfere. La stessa cosa viene fatta con altre due sfere in modo da ottenere il secondo piano radicale. (b) Intersezione dei due piani radicali trovati, si ottiene una retta. (c) Intersezione della retta con una delle tre sfere, si ottengono due punti. (d) Il punto più vicino alla Terra è il punto cercato.

1.1.4 Errori del sistema GPS

Il segnale che riceviamo dai satelliti deve giungere in condizioni ottimali, ovvero la trasmissione del segnale attraverso gli strati dell'atmosfera non deve subire variazioni non compensabili.

La correzione dell'errore dovuto ad effetti di rifrazione attraverso gli strati più alti dell'atmosfera richiede però l'elaborazione di due segnali distinti, sfasati tra di loro: questo procedimento è effettuabile solamente con apparati militari che ricevono infatti entrambi i due segnali trasmessi dai satelliti.

Altri errori sono quelli dovuti al calcolo della posizione dei satelliti, compensati per lo più dai satelliti stessi che aggiornano di continuo fra loro le informazioni sulla loro posizione.

Errori che ci riguardano più da vicino sono, ad esempio, quelli dovuti al segnale riflesso da pareti rocciose, da costruzioni elevate, ovvero quando il percorso fatto dal segnale non è diretto ed è maggiore di quello che dovrebbe essere (il tempo di percorrenza è più lungo).

Infine, gli ultimi errori che possiamo avere sono quelli dovuti ad ostacoli naturali e non, che impediscono la ricezione del segnale o consentono la ricezione di un numero insufficiente di satelliti.

Per tutti questi motivi è naturale che la precisione del rilevamento della posizione mediante GPS non sia costante, ma vari in base alle diverse situazioni.

In generale si può quindi dire che la precisione aumenta sempre con l'aumentare dei satelliti osservati, numero che può variare considerevolmente spo-

standosi anche di poco, o in momenti successivi della giornata.

1.2 A-GPS

L'A-GPS è un sistema che consente di diminuire i tempi necessari alla prima localizzazione di un terminale GPS. La sua utilità subentra soprattutto nei cosiddetti “canyon” urbani, quali vie strette o viali alberati, in cui è difficile stabilire con precisione la lista di satelliti in vista.

Uno dei principali problemi dei terminali GPS classici è relativo al cosiddetto *fixing*, ovvero il rilevamento della posizione di un terminale in fase di accensione del dispositivo: essi devono infatti ricavare la lista dei satelliti in vista in quel momento al fine di potersi agganciare e poter risalire alla propria locazione.

Tale processo è alquanto dispendioso in termini di tempo e risorse ed il sistema A-GPS è stato studiato al fine di abbattere tali costi, anche in previsione di un utilizzo su terminali con basse capacità di elaborazione o con risorse energetiche limitate.

Lo scopo principale del sistema è quindi quello di “assistere” il ricevitore GPS nel calcolo della disposizione, fornendogli informazioni sui satelliti in vista con il supporto dell'operatore di telefonia mobile.

1.2.1 Funzionamento

Il sistema si basa sul fatto che ogni cella di telefonia mobile presente sul territorio ha una posizione fissa: sarà quindi la cella stessa a ricavare quali sono i satelliti GPS in vista, istante per istante.

Quando un terminale A-GPS vuole conoscere la propria posizione, si collega tramite la rete cellulare ad un *Assistance Server* (che può essere gestito anche dall'operatore stesso), al quale viene inviata anche l'informazione sulla cella alla quale l'utente risulta essere agganciato.

Dato che i satelliti in vista dalla cella risultano essere noti, si può assumere che anche il terminale A-GPS veda i medesimi satelliti; pertanto il server elaborerà una lista di satelliti in vista e la invierà attraverso la rete cellulare al terminale, che potrà ricavare immediatamente la propria posizione.

1.3 IPS

L'Indoor Positioning System permette di localizzare ed effettuare il tracking di dispositivi all'interno di edifici e costruzioni.

Più in generale, tale meccanismo consente l'individuazione e il posizionamento di oggetti situati in determinate locazioni, che risultano essere marchiati con *token*, etichette, ecc.

I GPS o i GNSS non risultano essere troppo adatti all'utilizzo di tale meccanismo a causa della comunicazione via onde radio che comporta la dispersione del segnale: nonostante questo, ci sono innumerevoli studi al fine di poter introdurre tale meccanismo anche in questo ambito.

2 Sistemi Multi-Agente

Un sistema multi-agente è un sistema composto da più agenti che convivono e che hanno un obiettivo più o meno comune. A livello progettuale, i sistemi multi-agente (MAS) sono costituiti dagli agenti e dall'ambiente in cui essi vivono, di cui devono essere specificate le caratteristiche spaziali e temporali.

Essi possono modellare sistemi di qualunque complessità, ma risultano essere particolarmente adatti in situazioni in cui non è facile avere una visione d'insieme del sistema completo.

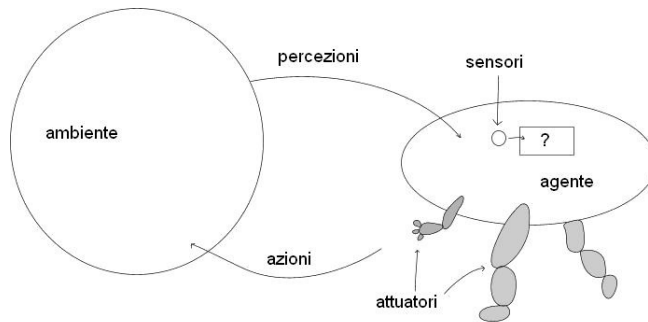
Un agente può essere concepito come qualunque cosa in grado di percepire il proprio ambiente attraverso sensori e di agire in quell'ambiente attraverso attuatori. In ambito informatico, un agente è essenzialmente un componente software autonomo, in grado di eseguire azioni e interagire con l'ambiente circostante in modo indipendente seguendo i propri obiettivi.

2.1 Definizione

Le principali caratteristiche di un agente sono l'autonomia e la sua capacità di essere flessibile; ogni agente è:

- *collocato in un ambiente* (anche dinamico e complesso). Gli agenti possono percepire e agire sull'ambiente. Quest'ultimo è dunque parte integrante del MAS e può determinare, in concomitanza con le azioni svolte dagli agenti, l'evoluzione del sistema in una tra le tante direzioni possibili.
- *autonomo*, poiché ogni agente ha il controllo sulle proprie azioni e sul proprio stato interno: a differenza degli oggetti, risultano essere entità attive e non passive. Una volta che sono stati definiti gli obiettivi e le regole di comportamento, gli agenti sono in grado di operare in modo totalmente indipendente all'interno del proprio ambiente, prendendo decisioni e potendo interagire con eventuali altri agenti.
- *sociale*, perché coopera con l'uomo o con altri agenti per raggiungere i propri obiettivi, attraverso un linguaggio di comunicazione ad agenti e grazie alle strategie di coordinamento e negoziazione.
- *reattivo*, poiché percepisce l'ambiente in cui è collocato e risponde entro un certo tempo ai cambiamenti che avvengono al suo interno, modificando il proprio comportamento. Gli obiettivi dell'agente possono evolversi e cambiare anch'essi, in conseguenza alle variazioni dell'ambiente, rendendo così tale tecnologia molto adatta in situazioni in cui le operazioni da compiere sono legate allo stato dell'ambiente.
- *proattivo*, poiché agisce in risposta ai cambiamenti dell'ambiente, ma è capace di mostrare un comportamento "goal-directed", prendendo l'iniziativa.

Di seguito viene mostrato uno schema riassuntivo dell'interazione tra agente e ambiente.



2.2 Supporto MAS: il framework JADE

JADE (Java Agent Development Framework) è un framework software per la realizzazione di sistemi multi-agente, sviluppato da Telecom Italia e distribuito con licenza LGPL. E' sviluppato interamente in linguaggio Java ed è conforme alle specifiche FIPA per lo sviluppo di sistemi ad agenti; esso mette a disposizione strumenti utili alla modellazione del problema e alla creazione del MAS, al controllo della piattaforma e al debugging dell'applicazione.

2.3 Architettura di JADE

La componente principale del framework JADE è costituita da un *Runtime Environment* (detto Platform), che ha il compito di garantire tutta la serie di servizi relativi allo scambio di messaggi descritti da FIPA. All'interno della platform possono esistere uno o più Container, che risultano essere dei veri e propri contenitori dove possono "vivere" gli agenti creati; oltre a questi è sempre presente il cosiddetto *Main Container*, che offre i servizi di cui JADE dispone.

Gli elementi fondamentali di una piattaforma secondo lo standard FIPA sono:

- l'*agente*: esso rappresenta l'entità principale che opera all'interno della piattaforma, ed è quindi in grado di svolgere un determinato insieme di azioni all'interno di essa, tra cui l'interazione con altri agenti, con la piattaforma stessa o con il software esterno ad essa. Ad ogni agente deve essere associato un proprietario (owner), che può essere un utente umano oppure un'entità software. Ad ogni agente è inoltre assegnato un AID (Agent Identifier), ovvero un identificatore univoco che permette di individuarlo con certezza all'interno della piattaforma in cui opera: esso è in genere costituito da una stringa, da un numero o da una combinazione di questi due elementi. Inoltre, a causa del fatto che un agente può utilizzare

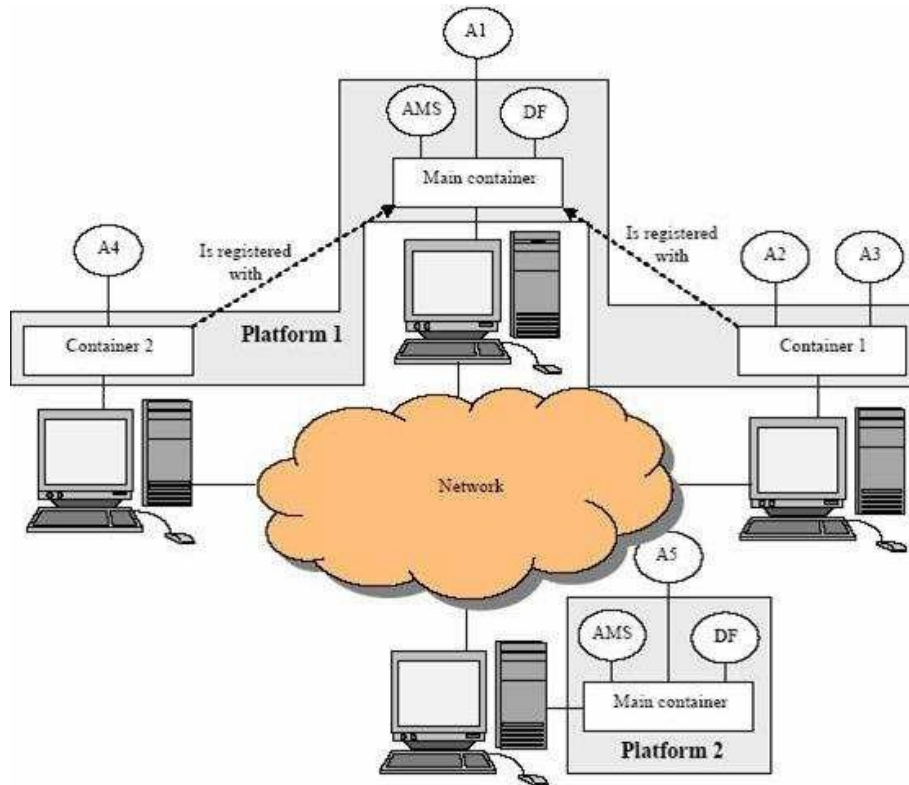
vari servizi di trasporto per spostarsi da un nodo ad un altro, ad esso viene anche associato un indirizzo di trasporto, per poterlo identificare univocamente, questa volta all'interno della rete in cui opera;

- l'*Agent Management System (AMS)*: è un componente che ha come compito quello di supervisionare gli accessi alla piattaforma da parte di entità esterne, oltre che di monitorare le azioni che essi compiono; è fondamentale, secondo le specifiche FIPA, che all'interno di una piattaforma ve ne sia uno soltanto. Quindi, se ad esempio un agente (esterno o appena creato) volesse operare all'interno della piattaforma, è obbligato ad effettuare una registrazione presso l'AMS: quest'ultimo concederà all'agente l'autorizzazione per poter operare all'interno della piattaforma stessa e provvederà all'assegnazione di un AID opportuno.

Per poter svolgere questo tipo di compito, l'AMS mantiene al suo interno un elenco di tutti gli AID assegnati e, per ognuno di essi, ulteriori informazioni utili alla gestione degli agenti stessi (come ad esempio il loro indirizzo). Un'altra funzione accessoria svolta da questo componente è quella del servizio di pagine bianche (*White Page Service*): ogni agente può utilizzarlo per conoscere quali agenti sono presenti all'interno della piattaforma in un determinato istante e in quale luogo si trovano;

- il *Directory Facilitator (DF)*: come per l'AMS, anche la presenza di questo componente è obbligatoria all'interno della piattaforma. Il suo compito principale è quello di fornire il cosiddetto servizio di pagine gialle (*Yellow Page Service*): a differenza delle pagine bianche, qui viene mantenuto un elenco relativo ai servizi offerti da ogni agente presente all'interno della piattaforma. In questo modo, un agente che offre un determinato servizio può comunicarlo al DF, che provvederà a memorizzare tale informazione. Simmetricamente, un altro agente che necessita di tale servizio può effettuare una query al DF per conoscere il nome e il luogo dell'agente che lo rende disponibile, per poi, eventualmente, contattarlo;
- il *message transport system (MTS)*: ha il compito di gestire e coordinare lo scambio di informazioni tra agenti; in particolare questi ultimi possono essere situati nella stessa piattaforma o anche su piattaforme diverse. La classe che implementa il messaggio è la classe *ACLMessage*, ed offre tutte le funzioni necessarie alla creazione e alla gestione dei messaggi.
- l'*agent platform*: è la piattaforma per gli agenti mobili vera e propria, ovvero un componente che fornisce un ambiente uniforme in cui gli agenti possono svolgere i propri compiti, comunicare e spostarsi. È da notare come, secondo lo standard FIPA, ne facciano parte anche il sistema operativo e il calcolatore fisico su cui essa è in esecuzione.
- il *software esterno*: viene utilizzato come riferimento per indicare tutto il software non basato sulla tecnologia ad agenti software, ma che offre servizi a cui gli agenti della piattaforma possono accedere.

JADE dà inoltre la possibilità di creare applicazioni distribuite, potendo creare più piattaforme che rappresenteranno i vari computer connessi in rete. Ogni platform anche in questo caso, dovrà necessariamente avere al suo interno un *Main Container* che conterrà gli agenti citati precedentemente.



Per facilitare la gestione della piattaforma, è stata sviluppata una GUI (Graphical User Interface) che permette di monitorare lo stato degli agenti e dei container in esecuzione in un dato istante. Quest'ultima prende il nome di *RMA* (*Remote Monitoring Agent*) ed è stata implementata anch'essa come un agente; attraverso tale GUI è possibile lanciare gli agenti creati e inviare messaggi, monitorando graficamente il processo.

2.4 Scambio di messaggi

Gli agenti sono in grado di comunicare tra di loro, come accennato precedentemente, grazie all'implementazione dell'*Agent Communication Language*(*ACL*) definito dalle specifiche FIPA, il quale supporta tutti i protocolli di trasporto

da esso definiti. Il tipo di protocollo utilizzato varia a seconda dei casi: in particolare si possono delineare tre specifiche casistiche:

- gli agenti si trovano all'interno dello stesso container
- gli agenti sono situati in container differenti all'interno della stessa piattaforma
- gli agenti sono su piattaforme differenti

Nel caso in questione gli agenti adotteranno il meccanismo di invocazione remota *RMI*, protocollo utilizzato solitamente nella seconda casistica.

2.5 Agenti in JADE

In Jade ogni agente è un thread, che viene creato, terminato, sospeso e riattivato dall'AMS o per mezzo di una richiesta ad esso rivolta. Una volta che l'AMS ha garantito l'unicità dell'identificatore dell'agente ed ha creato il thread, si passa alla fase di inizializzazione dell'agente stesso. Questa fase prevede l'esecuzione di un metodo *setup()*, dove si eseguono le procedure di inizializzazione (lettura dei parametri di configurazione, connessione a database, ecc), e si attivano i cosiddetti *behaviours*, cioè comportamenti che l'agente è in grado di eseguire.

2.6 Behaviours

I Behaviours sono il “cuore” di ciò che fa un agente: ciascuno è pensato per eseguire un particolare compito.

Ad ogni agente è associata una lista di behaviours, ognuno dei quali ha il compito di eseguire determinate azioni nel momento dell'attivazione.

Esistono vari tipi di behaviours a seconda delle situazioni:

- *GenericBehaviour*: tale compito è caratterizzato dalla presenza di un indicatore di stato: al variare del valore dell'indicatore si esegue un'azione diversa. Il behaviour termina quando si è raggiunto un determinato stato.
- *OneShotBehaviour*: l'azione viene eseguita una sola volta, terminando senza che sia necessario verificare la condizione di completamento.
- *CyclicBehaviour*: l'azione in questo caso viene eseguita continuamente e il behaviour non termina mai. In termini di risorse quali CPU e memoria centrale, tale behaviour è estremamente costoso, ma JADE mette a disposizione istruzioni in modo tale da mettere in stato di blocco l'agente.
- *TickerBehaviour*: uguale al precedente con la differenza che gli intervalli con cui viene messo in esecuzione sono periodici.
- *FSMBehaviour*: tale behaviour associa un'azione da compiere in base allo stato attuale dell'agente. Questo tipo di comportamento risulta essere utile quando si ha a che fare con behaviour complessi formati da più protocolli di comunicazione che interagiscono tra di loro.

- *SequentialBehaviour*: questa classe permette di creare un behaviour composto da uno o più sotto-behaviour, che possono essere aggiunti tramite il metodo *addSubBehaviour()*. Questi ultimi verranno eseguiti in sequenza nell'ordine specificato, ed il behaviour terminerà solo quando saranno terminati tutti i sotto-behaviour.
- *ParallelBehaviour*: permette di eseguire i sotto-behaviour specificati in modo concorrente.

Quando gli agenti terminano i loro compiti, oppure nel caso in cui l'AMS lo richieda, gli agenti vengono terminati. Prima dell'effettiva cancellazione dalla lista degli agenti attivi, ogni agente esegue il metodo *takeDown()*. In questo modo si deregistrano i servizi dell'agente dal DF e si provvede alla cancellazione di altre eventuali tracce che potrebbero creare problemi all'interno dell'applicazione.

3 Sviluppo dell'applicazione

Di seguito vengono spiegate l'architettura e la progettazione dell'applicazione.

3.1 Obiettivi

Scopo dell'applicazione è quello di introdurre un comportamento dinamico rispetto alle metodologie illustrate precedentemente che si avvalgono di antenne e ripetitori statici posizionati sul territorio terrestre, in modo tale da rendere autonome le entità GPS, le quali potranno così comunicare anche con dispositivi non statici.

In particolare i GPS possono risalire alla propria posizione in qualsiasi momento e in qualsiasi luogo, grazie alla comunicazione con altre entità GPS e sfruttando in questo modo le informazioni ricevute.

Si è utilizzato allo scopo il linguaggio di programmazione JADE, in quanto il più consono per l'implementazione di tale programma, e ci si è avvalsi del plug-in di Eclipse *EJADE*.

3.2 JADE nell'applicazione

Come mostrato precedentemente, JADE dispone di servizi ottimali per quanto riguarda la comunicazione e interazione tra agenti. Di conseguenza è stato scelto JADE come linguaggio di programmazione al fine di poter sfruttare tali servizi e implementare l'applicazione; in particolare si sono utilizzati:

- il modello di comunicazione
- le *yellow pages*

3.2.1 Modello di comunicazione

Il comportamento alla base del programma è la comunicazione che avviene tra le entità GPS.

Con il linguaggio di programmazione JADE gli agenti sono in grado di comunicare gli uni con gli altri, collaborare, negoziare, ecc.

Il modello di comunicazione adottato è l'*Asynchronous Message Passing* in cui ad ogni agente è associata una coda di messaggi ricevuti dagli altri agenti, aggiornata ogni qualvolta arrivi un nuovo messaggio. Il formato dei messaggi scambiati segue le specifiche del linguaggio ACL definito dal FIPA.

Affinchè gli agenti di una piattaforma possano comunicare, è necessario che essi condividano non solo linguaggi e protocolli di comunicazione, ma anche un comune vocabolario di termini utilizzati nei messaggi scambiati.

Ogni messaggio che viene scambiato è composto dalle seguenti proprietà:

- il mittente del messaggio
- una lista di destinatari

- l'atto comunicativo che indica qual'è l'intenzione del mittente
- il contenuto
- il linguaggio che indica la sintassi utilizzata nel contenuto
- l'ontologia che indica il vocabolario dei simboli utilizzati nel contenuto
- un protocollo al quale il messaggio appartiene

Nel caso specifico, ogni agente è in grado in ogni istante di ricevere informazioni dalle altre entità, sebbene questo comportamento sia limitato alla disponibilità o meno del GPS in questione. Infatti i GPS provvederanno a contattare, in caso di assenza del segnale, solamente le entità che risulteranno possedere informazioni utili.

Il GPS specificherà quindi, in accordo con le proprietà sopra elencate:

- i destinatari: che risulteranno essere i GPS più prossimi a lui
- il contenuto: ogni GPS (che risulti essere in assenza di segnale satellitare) invierà una stringa di richiesta ai destinatari prescelti per ottenere la loro posizione

Una volta inviata la stringa, i GPS destinatari provvederanno alla ricezione del messaggio, in primis controllando che sia di richiesta della posizione, successivamente inoltrando la risposta con la propria posizione.

Tale procedimento viene effettuato ogni secondo, dal momento che le entità GPS sono in continuo movimento e potrebbero perdere il segnale satellitare in qualsiasi istante.

3.2.2 Yellow Pages

Il secondo servizio fornito da JADE utilizzato all'interno dell'applicazione risulta essere quello delle *yellow pages*: tale servizio consente ad un agente di rendere noti i servizi che eroga potendosi registrare al suo interno e specificandone nome e AID dell'agente stesso.

Il servizio che nell'applicazione in questione ci interessa, è quello di *GPS available*, che sta ad indicare se il GPS in questione ha informazioni utili da fornire alle altre entità GPS incapaci di ricevere il segnale satellitare.

Le *yellow pages* sono aggiornate ogni secondo dal momento che le entità GPS si muovono in ogni istante: ogni entità provvederà quindi alla registrazione o deregistrazione del servizio in ogni intervallo di tempo.

Ogni volta che un GPS vorrà contattare gli altri GPS al fine di potersi triangolare, consulterà quindi le *pagine gialle* e contatterà quelli che risulteranno essere disponibili e più vicini a lui.

A seconda del numero di canali mancanti, i GPS mittenti provvederanno al controllo del numero di servizi disponibili all'interno delle *pagine gialle*: nel caso in cui il numero di servizi sia inferiore rispetto al numero di segnali di

cui si necessita, il GPS in questione rimarrà in una situazione di attesa. Esso continuerà il proprio moto nello spazio, ma le informazioni relative alla propria posizione e visibili tramite GUI non saranno più disponibili.

In un secondo momento, o per la successiva registrazione di un GPS all'interno delle *pagine gialle* (se il GPS in questione risulta essere ancora in una zona priva di segnale satellitare), o per lo spostamento in un'area dove risulta essere disponibile tale segnale, i dati visibili via GUI torneranno ad essere disponibili.

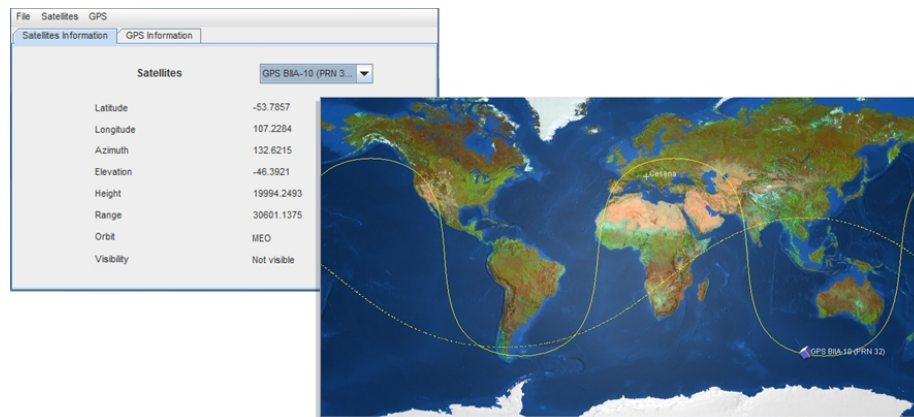
3.3 Comportamento dell'applicazione

In primis, l'applicazione provvederà a caricare i dati del NORAD (il quale fornisce un quadro di insieme sulla situazione come ad esempio la natura, la posizione e la direzione di ogni aviogetto od oggetto spaziale), memorizzando i *TLE* (*Two-Line-Element*) dei satelliti GPS esistenti.

I TLE sono uno standard per la definizione dei parametri orbitali, necessari all'applicazione in questione per effettuare il tracking satellitare e poterne calcolare le orbite.

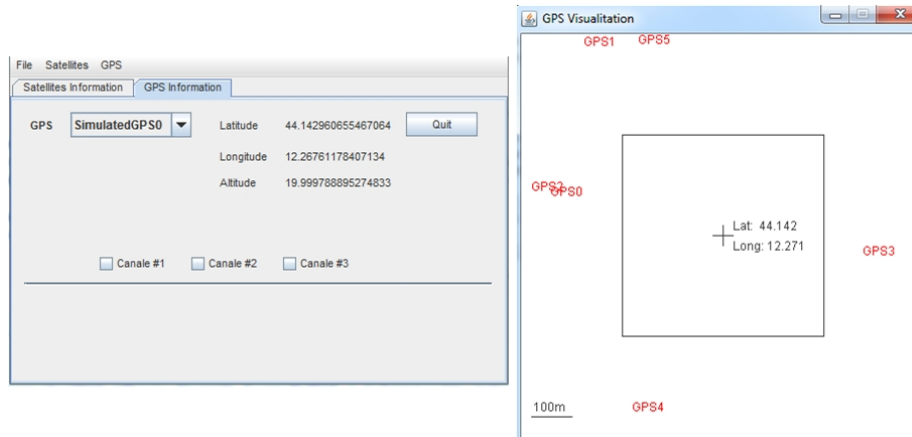
In particolare il tracking si basa sull'utilizzo di un propagatore orbitale (strumento utilizzato per poter predire la posizione di un satellite nel suo moto di rotazione attorno alla Terra) chiamato *SGP4*: esso viene comunemente usato per satelliti con orbite basse.

L'applicazione prevede la possibilità di poter monitorare visivamente tutti i satelliti GPS, mostrandone le informazioni relative rispetto al punto di monitoraggio prescelto: in particolare è stata definita Cesena come punto di riferimento, memorizzando i dati dei satelliti che risulteranno essere visibili da quella posizione e aggiornandoli in tempo reale.



Successivamente l'utente potrà selezionare il numero di antenne GPS che si intende simulare, le quali saranno contenute all'interno di un pannello rappresentante un km². La visualizzazione comprende inoltre un'area di forma quadrata

all'interno della quale, ai fini dimostrativi dell'applicazione, si presuppone che i GPS non ricevano segnale, simulando in tal modo un canyon urbano.



E' prevista inoltre la possibilità, nel caso in cui il GPS si trovi al di fuori dell'area definita, di poter oscurare da uno a tre canali riceventi: in questo modo si presuppone che un numero di satelliti definito da utente smetta improvvisamente di trasmettere dati; in questo caso il GPS effettuerà una trilaterazione cosiddetta *ibrida* (che verrà spiegata successivamente).

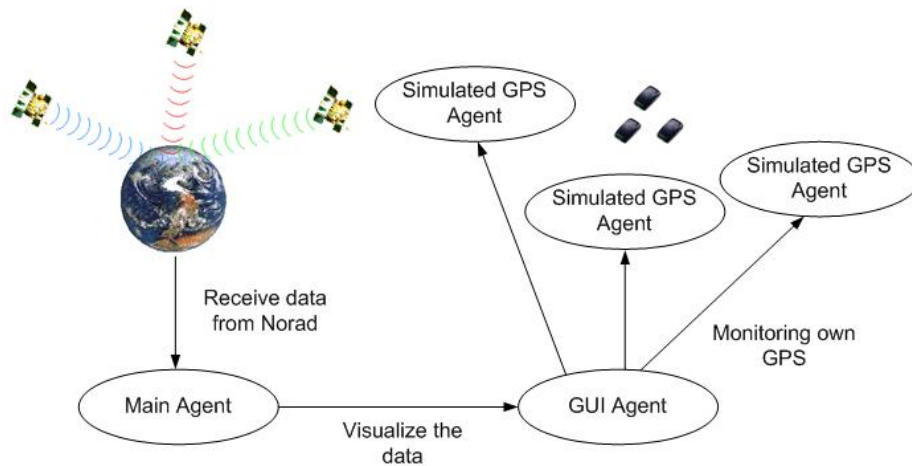
Nel momento in cui un GPS entrerà all'interno del *quadrato*, provvederà a contattare altri GPS che dovranno risultare essere necessariamente al di fuori di questo: la ricerca avviene mediante il meccanismo delle *yellow pages* spiegato precedentemente.

Le informazioni relative alla posizione dei GPS risulteranno essere visibili anch'esse via GUI e l'utente potrà decidere, a seconda del numero di GPS che si è inteso simulare, quali di essi si vuole monitorare.

Il movimento delle antenne GPS è stato simulato, per semplicità, utilizzando un moto rettilineo uniforme nel quale le condizioni iniziali sono definite in maniera randomica, sia per quanto riguarda la posizione che per la velocità. Per evitare inoltre che i GPS visualizzati fuoriuscissero dal pannello di visualizzazione, sono stati introdotti dei vincoli che prevedono l'inversione del moto una volta raggiunto il confine.

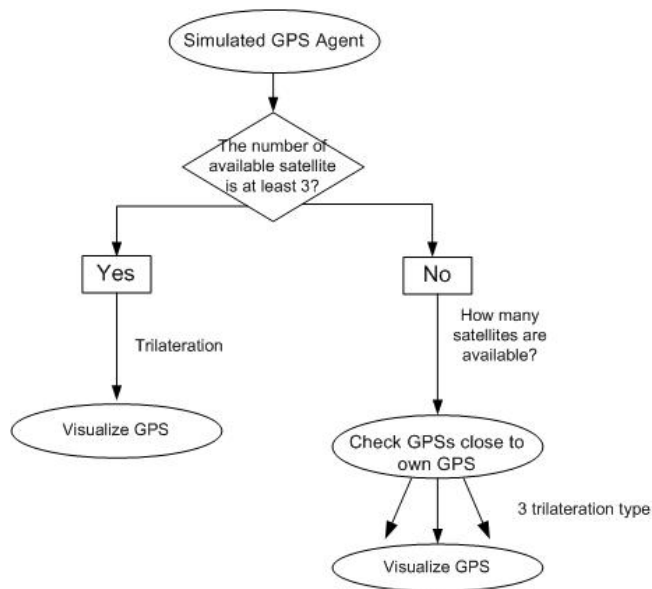
3.4 Architettura

L'architettura dell'applicazione risulta essere la seguente:



In particolare il *Main Agent* è l'agente che si occupa della gestione dell'applicazione nella sua globalità: i dati pervenuti verranno mostrati all'interno dell'interfaccia grafica, gestita anch'essa dal *Main Agent* e comunicante con quest'ultimo mediante un protocollo specifico di JADE.

Il comportamento dei singoli *Simulated GPS Agent* è mostrato nella figura seguente:



in particolare essi controlleranno che il numero di satelliti visibili sia pari a tre e, in caso affermativo, provvederanno alla triangolazione con successiva visualizzazione dei GPS in questione.

In caso contrario, a seconda del numero di satelliti visibili, completeranno la mancanza con un numero di GPS tale che la somma finale sia sempre tre.

In questo caso possiamo avere tre tipi di trilaterazione:

- la prima che prevede la trilaterazione con soli satelliti nel caso in cui questi siano tutti disponibili
- la seconda che prevede la trilaterazione “ibrida” tra satelliti e GPS
- la terza infine che prevede la trilaterazione con sole entità GPS

Il controllo della visibilità dei satelliti viene anch'esso effettuato ogni secondo aggiornando la visualizzazione grafica in base ai calcoli effettuati.

3.5 Agenti implementati

In questa sezione si mostrerà nel dettaglio da quali agenti è costituito il sistema.

3.5.1 Main Agent

Il *Main Agent*, come visto precedentemente, è colui che si occupa della gestione dell'interfaccia grafica e del calcolo dei dati satellitari iniziali; inoltre è a partire da questo che si possono lanciare i *Simulated GPS Agent* che si intende simulare.

Altra funzionalità del Main Agent è quella di gestire i *Container* di tutta l'applicazione nel caso di terminazione di quest'ultima: esso infatti fa in modo che non vi sia proliferazione nella creazione di Container, eliminandoli ogni volta che viene chiuso il pannello di visualizzazione per poter eseguire una successiva simulazione.

Affinchè sia possibile la comunicazione tra agenti e GUI, JADE mette a disposizione la classe astratta *GuiAgent* che estende la classe *Agent*. Questa classe ha due metodi specifici:

- *postGuiEvent*
- *onGuiEvent*

Questi metodi permettono di gestire le interazioni tra un'interfaccia grafica e un agente; quest'ultimo dovrà estendere la classe *GuiAgent* e riceverà gli eventi tramite il metodo *onGuiEvent*, mentre la GUI li notificherà tramite il metodo *postGuiEvent*.

Nella fase iniziale di sviluppo dell'applicazione era stato adottato questo meccanismo per la comunicazione tra agente e GUI, dal momento che il comportamento degli agenti non era stato ancora automatizzato ed il controllo di entità GPS disponibili e di selezione dei destinatari veniva fatto totalmente graficamente. In un secondo momento, si è eliminata tale procedura di comunicazione in quanto si è reso automatico il comportamento dei GPS. Ciò nonostante, è tramite il Main Agent che è possibile visualizzare la parte grafica dell'applicazione e caricare i dati iniziali del NORAD.

3.5.2 Simulated GPS Agent

Ogni entità GPS risulta essere un agente denominato *Simulated GPS Agent*.

In particolare esso ha due specifici *Behaviour*:

- la ricezione
- la registrazione sulle *yellow pages*

I behavior implementati sono di tipo generico, seppur con caratteristiche cicliche: in questo modo è stato possibile specificare l'intervallo di tempo ogni quanto devono essere eseguiti tali compiti, alleggerendo la CPU.

Sono poi implementate le funzioni di *sendMessage*, *register* e *deregister*, che vengono richiamate a seconda della posizione effettiva del GPS e della successiva comunicazione con un'eventuale entità.

Ogni Simulated GPS Agent importa la classe *Simulated GPS Trilateration* che implementa i tre tipi di trilaterazione previsti.

Ad ogni Simulated GPS Agent è associato un oggetto GPS che è possibile visualizzare mediante interfaccia grafica e monitorarne altitudine, longitudine e latitudine.

4 Risultati ottenuti

Gli obiettivi prefissati inizialmente per lo svolgimento dell'applicazione sono stati raggiunti: i GPS infatti riescono a ricostruire la propria posizione in totale o parziale assenza di segnale satellitare. Durante le simulazioni si è potuto osservare che l'errore ottenuto nella ricostruzione della posizione attraverso l'utilizzo di satelliti, antenne GPS oppure in maniera ibrida, risulta essere dello stesso ordine di grandezza.

In questa implementazione si è supposto che il numero di satelliti necessari alla trilaterazione sia tre: in realtà questo non è del tutto corretto: infatti come spiegato nella fase introduttiva della relazione, si necessiterebbe di un quarto satellite relativo alla sincronizzazione temporale tra i GPS. Inoltre gli errori dei GPS, anch'essi discussi precedentemente, vengono assunti nulli e non si sono introdotti eventuali simulazioni di ritardo nella comunicazione tra agenti. Tali semplificazioni potrebbero portare ad un aumento dell'errore di stima della posizione ma è plausibile, facendo uso dei risultati ottenuti, che esso possa rimanere in linea con quello attualmente misurabile.

Attraverso la tecnologia software ad agenti ed incrementando il numero di canali (con conseguente trilaterazione con un numero maggiore di antenne o satelliti), si arriverebbe però ad un possibile incremento di precisione nella georeferenziazione delle antenne GPS.

5 Conclusioni

Attraverso lo svolgimento del progetto, è stata dimostrata la possibilità di implementare dei dispositivi intelligenti, basati su un linguaggio ad agenti, in grado di ricostruire la propria posizione senza la necessità di ricevere informazioni dalla costellazione di satelliti GPS. Attraverso l'utilizzo di *Jade 4 Android* e *Jade Leap* è pensabile anche di poter implementare tale progetto per dispositivi mobili e sfruttarne le potenzialità.

Jade Leap in particolare rimpiazza alcune parti del kernel JADE formando un ambiente runtime modificato che può essere impiegato in un'ampia gamma di dispositivi che vanno dai server fino ai cellulari dotati di ambiente J2ME.

Jade 4 Android subentra invece al fine di sfruttare le grandi potenzialità del sistema operativo in questione, sviluppato interamente in Java: con questo tool il team di Jade intende fornire il supporto necessario per utilizzare una piattaforma Jade-Leap su una piattaforma Android.

Purtroppo vi sono ancora delle limitazioni legate alle prestazioni delle risorse disponibili su un dispositivo mobile: in primis perché le risorse che un dispositivo mobile mette a disposizione non sono paragonabili con quelle di un pc; in secondo luogo perché tale add-on risulta ancora avere alcune limitazioni (come l'assenza di migrazione degli agenti o l'esecuzione di più agenti all'interno dello stesso container lanciato su Android).

Riferimenti bibliografici

- [1] C. Casanova, "Progetto e stesura preliminare di un S/W per il tracking satellitare in ambiente Java," Relazione tirocinio, 2008,
- [2] Z. Biacs, G. Marshall, M. Moeglein, W. Riley, "The Qualcomm/SnapTrack Wireless-Assisted GPS Hybrid Positioning System and Results from Initial Commercial Deployments," 2002,
- [3] C. Randell, H. Muller, "Low Cost Indoor Positioning System," University of Bristol,
- [4] D. Niculescu, B. Nath, "Ad Hoc Positioning System (APS) Using AOA," IEEE, 2003,
- [5] F. Bellifemine, G. Caire, D. Greenwood, "Developing Multi-Agent Systems with JADE," 2007,
- [6] M. Mehrtash, "GPS Navigation Toolbox," Matlab Central, 2008,
- [7] S. Mani, "Global Positioning System," 2007,
- [8] M. Rosada, "Realizzazione su piattaforma ad agenti mobili Jade di un sistema di autenticazione biometrica," 2011,
- [9] A. Pierangeli, "Progetto e sviluppo di un sistema informativo sanitario ad agenti," 2009