Progetto di Sistemi Informativi Distribuiti
Agenti che risiedono su dispositivi mobili. Documentation

Buda Claudio

January, 2005

Contents

1 Abstract

Il progetto consiste nella costruzione di un sistema Client/Server dove agenti risiedenti su dispositivi mobili scandagliano la rete internet alla ricerca di pacchetti software (oppure ipotetici servizi) messi a disposizione da altri agenti. Uno stesso dispositivo può avere all'interno pacchetti SW da scambiare con altri. La complessità del progetto risiede nel reperire moduli di TuCSoN già sviluppati per dispositivi mobili e sviluppare e ottimizzare le parti non ancora complete. In seguito dopo aver raggiunto l'obiettivo di avere un sistema ad agenti che risiede su dispositivi portabili si punterà a sviluppare la semplice applicazione che ricerca software in rete. Per riuscire a stare all'interno dei tempi previsti si vogliono dedicare tre settimane per l'ultimazione e l'utilizzo di una MicroEdition di TuCSoN che permetta perlomeno lo sviluppo dell'applicazione.

2 Considerazioni iniziali

Dopo aver analizzato e testato il progetto della tesi di Ceruti sul "porting di TuCSoN su J2ME" si è giunti all'interrogativo di cosa sviluppare. Grazie alla tesi di Ceruti si parte dalla possibilità di creare agenti anche su J2ME di quelli costruiti per l'infrastruttura d'interesse (TuCSoN), ma niente altro. Manca la possibilità di costruire nodi TuCSoN su dispositivi che hanno una "MicroJVM" e soprattutto gli strumenti apportati dalla tesi sono stati sviluppati per mobili di una certa capacità computazionale e con una discreta risorsa di memoria. Ancora oggi quasi tutti i cellulari, a parte quelli che usciranno nel 2005 sono con un livello di JVM inferiore a quella necessaria per supportare un agente TuCSoN così pensato come nella tesi analizzata. Tutto il lavoro sviluppato deve essere infatti testato grazie ad un simulatore messo a disposizione dalla SUN. Per ovvi limiti di tempo si sono dovute fare scelte drastiche sul procedere del progetto. É stata dapprima scartata la possibilità di sviluppare applicazioni direttamente per mobili oggi in commercio e si è deciso di proseguire nell'utilizzazione di un simulatore, in attesa di avere in commercio dispositivi con maggiori prestazioni. Interessante l'idea di sviluppare quella parte di TuCSoN inerente ai Tuple centres per risiedere sopra una macchina virtuale di capacità ridotte (buon argomento da sviluppare in futuro per una probabile tesi). Siccome il tempo di sviluppo è ristretto l'alternativa è quella di sviluppare una buona applicazione per mobili su un sistema ad agenti ma dove sui mobili risiede solo la parte client (gli agenti) del sistema. Infine ultime ma non per questo poco importanti considerazioni da fare sono le seguenti. Tutto il lavoro svolto nella tesi appena citata è stato fatto avvalendosi TuCSoN 1.4 quindi le modifiche sono state eseguite su di esso. Oggi si può utilizzare la versione TuCSoN 2.0 più performante e con un'architettura diversa (a livelli) dalla versione

precedente. Ripetere tutto il lavoro per rendere MicroEdition TuCSoN 2 necessita una grossa mole di tempo. Ottimizzare la versione TuCSoNME basata su una vecchia architettura non può dirsi pienamente performante perciò aspettando l'avvento (inevitabile) di dispositivi mobili con capacità superiori si decide di utilizzare TuCSoN2 simulandone il suo utilizzo "mobile" su dispositivi che la supportano (laptop). Si consideri che passare da cellulari capaci di utilizzare la versione Micro di TuCSoN già citata a dispositivi che supportano l'ultima versione di TuCSoN è solo questione di pochi anni. Ora il punto di lavoro si sposta da un'area legata all'architettura e alla tecnologia ad un'area legata alla coordinazione di agenti che non risiedono più su dispositivi fissi ma capaci di muoversi nello spazio.

3 Analisi

Gli obiettivi principali di questo progetto sono quelli di sviluppare un sistema ad agenti che possa usufruire di dispositivi di piccola capacità, portabili e mobili nell'ambiente.

Si pensi ad un ambiente dove ci siano tanti modi per collegarsi ad una rete, che sia locale o globale, non ci si vuole soffermarci alle tecnologie con cui collegarsi. In questo ambiente si hanno dispositivi fissi e dispositivi mobili. Per stringere di più il campo e dare use cases sulle applicazioni possibili, si pensi ad una facoltà "intelligente".

Si cerca di localizzare il problema non tanto per definire la struttura nella quale il sistema vive ma soprattutto per definire quali bisogni richiede un sistema del genere. Ora ai dispositivi si può dare un'identità. Si pensi a dispositivi fissi quali PC desktop, server, stampanti, interruttori, sensori in poche parole tutti quegli apparati circuitali con un intelligenza o gestibili da un apparecchio più intelligente. Dispositivi mobili invece sono palmari, cellulari, schede magnetiche, sensori sul corpo. Il PC laptop può essere visto come un dispositivo ibrido.

In un ambiente universitario si può immaginare che ogni persona abbia con se almeno un dispostivo portabile e siano sparsi per la facoltà tanti fissi. In questa prima parte pensiamo alla definizione di agente come a una figura capace di rapppresenatre in un mondo virtuale una persona o uno strumento.

Ad ogni figura (docente, laureando, personale amministrativo) si può associare un agente capace di localizzarsi e fornire agli altri queste informazioni. Un agente "persona" può perciò ricercare dove si trovano fisicamente altri agenti. Si pensi a servizi come la ricerca di un docente da parte di uno studente.

Grazie a questi si può determinare se un docente ha già iniziato le lezioni oppure se si trova nel suo studio o se non è presente. Il docente potrebbe anche interagire con l'agente in modo da comunicare il suo stato oltre alla

sua posizione. Ad esempio a che punto dellla lezione egli sia, altrimenti se si trova nel suo studio ed è disponibile a ricevere oppure no. Lo scenario è vasto. Si pensi ad associare agenti anche a dispositivi fissi. Si potrebbe pensare ad un'agente stampante che attende richieste di stampa da altri utenti. Essa potrebbe comunicare se ha finito la carta, o se ha una lunga coda di stampe e suggerire all'operatore un'altra stampante libera. Un altro dispositivo a cui associare un agente potrebbe essere la serratura delle porte.

Un agente "studente" arrivando in sede e trovandola chiusa (o riservata solo a un certo tipo di personale) potrebbe contrattare con l'agente "serratura" per consentire allo studente di entrare nella facoltà senza l'ausilio di segretari o citofoni. Un nuovo studente entrando in una vasta facoltà smarrendosi potrebbe utilizzare il suo agente personale per collegarsi ad un agente che si occupa delle mappe per farsene fornire una in modo di riuscire a giungere alla lezione. Un ulteriore caso potrebbe essere quello di fornire durante la lezione i lucidi necessari in ordine di spiegazione ai laptop degli studenti da parte del professore. Con un agente che assiste il docente si potrebbe anche gestire il ricevimento degli studenti, basta immaginare ad agenti "studente" che si prenotano e agenti "professore" che fissano l'appuntamento. Non dimentichiamoci di servizi come mail e SMS che possono essere automatizzati e sincronizzati (come per esempio la notifica via SMS dell'arrivo di nuova posta elettronica da parte di un agente "persona").

Volendo però soffermarci sull'utilizzo di dispositivi mobili prendiamo in esame lo use case docente-studente. Pensiamo che queste due figure siano entrambi muniti di un PDA o cellulare (piuttosto evoluto) che si portano sempre con se. Di entrambi viene fatta la localizzazione, tramite l'identificazione dell'access point WLAN a cui sono costantemente collegati oppure grazie anche al servizio GPS (vedi cellulari UMTS); non si escludono altre tecniche hardware. A seconda della persona, del momento, e di altre considerazioni queste informazioni possono venir messe in un contenitore virtuale accessibile a tanti o a pochi. Oltre alla localizzazione si desidera fornire un servizio di stato da comunicare agli altri. Un professore potrebbere essere disponibile per l'amministrazione ma non per gli studenti. La disponibilità poi potrebbe avere varianti quali per un dialogo in rete o di persona. La non disponibilità potrebbe essere causata da una lezione o da un impegno personale.

Anche di uno studente si può definire la disponibilità. Si pensi ad uno studente a lezione, non disponibile per il personale amministrativo, ma che può sicuramente interagire con il professore che tiene la lezione. La pausa pranzo potrebbe essere motivo di non disponibilità per chiunque lo cerchi.

4 Progetto a partire dagli use cases

Tenendo conto delle considerazioni fatte in principio è chiaro che si sta parlando di una infrastruttura Multi Agent System perciò il concetto di agente

può essere quello di un componente autonomo ma che vive in una società, perciò dovrà interagire con altri attraverso artefatti di coordinazione. Nel caso di studio il nostro agente è immerso sia in un ambiente virtuale che in quello fisico, dato dalla mobilità dei dispositivi su cui risiede. Questo ambiente ibrido sarà in futuro quello più comune per sistemi di questo genere.

Questi agenti associati alle persone saranno agenti proattivi, perchè capaci di localizzarsi senza nessuna richiesta, sapranno reagire agli stimoli che arrivano dall'utente umano che richiede un qualche servizio. Non saranno altro che maschere di un umano in un mondo virtuale dove "umano" sarà ogni dispositivo elettronico raffigurato da un agente. É per questo che anche in un ambito virtuale serve un'abilità sociale per interagire con le altre figure virtuali.

Per realizzare gli use cases visti in analisi, si passa perciò alla progettazione di due principali agenti, uno che chiameremo docente e l'altro studente, che risiederanno su dispositivi mobili. Ovviamente ci si avvalrà anche di altri agenti che rapprenteranno l'ambiente o altri dispositivi o che aiuteranno la coordinazione, ma di questi ne parleremo in seguito. Per semplicità ipotizziamo di non avere personale amministrativo che avrebbe altrimenti bisogno a sua volta di un proprio tipo di agente.

4.1 Docente Agent

Ipotizziamo l'agente Docente situato su un palmare. Si può pensare anche ad un agente che esce dal palmare per andare su altri dispositivi di proprietà della stessa persona o di tanti agenti "dispositivi di" che fanno riferimento ad una stessa persona. Per semplificare le cose immaginiamo che ogni professore abbia un solo agente che lo rappresenti ed un solo dispositivo su cui risiede.

Questo dispositivo deve avere la minima pretesa di essere collegato in qualche modo alla rete della facoltà (a questo livello non interessa quale tecnologia lo permetta). Immerso nella rete della facoltà ci sarà il sistema ad agenti.

L'agente in questione dovrà risolvere il problema della localizzazione. Immaginando di avere il palmare collegato alla Wireless Lan si posson reperire informazioni dallo stesso access point. Queste informazini permetteranno di sapere in quale gruppo di stanze il palmare si trova. Se invece si è provvisti di un access point per stanza o di un collegamento Bluetooth in ogni stanza la localizzazione potrà essere ancora più fine. Una volta ottenute le informazioni su dove ci si trova starà all'utente decidere se pubblicarle a tutti o ad un numero ristretto. Ipotizzando questo venga deciso attraverso la selezione di un profilo da parte dell'utente, i dati vengono trasmessi in un centro di raccolta specifico per la localizzazione.

Esso probabilmente si troverà su una macchina fissa all'interno della facoltà. A questo centro possono accedere altri agenti, docenti e non. A seconda dei loro priviegi potranno usufruire delle informazioni o no.

Informazioni sullo stato del docente verranno inserite direttamente dal docente. Come per selezionare il profilo silenzioso o rumoroso di un cellulare, così un docente potrà selezionare il profilo disponibile o no, e per chi. Anche queste informazioni che verranno comunicate all'agente tramite un'interfaccia utente saranno depositate su centro di raccolta informazioni. Quest'ultimo si occuperà dello stato (o disponibilità) delle persone facenti parte del sistema.

4.2 Studente Agent

Anche per l'agente studente la localizzazione avviene nella stessa maniera dell'agente professore. Pure le informazioni dello stato saranno gestite nella stessa maniera dei docenti. Ovviamente gli stati non risulteranno gli stessi.

Un servizio aggiuntivo che si vuole fornire agli studenti (non per questo non potrà essere distribuito anche ai docenti) sarà quello di ricerca aule all'interno della facoltà. L'agente richiederà attraverso il centro smistamento mappe il servizio di consegna di una mappa per raggiungere una certa aula data la posizione dello studente.

4.3 Servizio Mappe Agent

Questo è uno degli agenti non mobili che fornisce un servizio di distribuzione mappe dell'edificio a chi si rivolge al centro mappe se ne occupa direttamente. Questo è un esempio di servizio che rimane su un dispositivo fisso che può essere facilmente utilizzato anche da agenti diversi da quelli sopracitati. Basti pensare ad un ipotetico soccorso di feriti o di un incendio sviluppato una centra area della facoltà. I soccorritori potrebbero usufruirne grazie ad agenti speciali che hanno i privilegi necessari per avere le mappe dell'edificio.

4.4 Centro per la Localizzazione

Questo centro si troverà, come già detto, su una macchina fissa all'interno della facoltà. Essa gestirà la coordinazione fra agenti che fanno sapere dove si trovano e agenti che vogliono sapere dove localizzare altri. Il centro non sarà altro che un artefatto di coordinazione fra gli agenti che ci accedono. Sia per rilasciare informazioni sulla posizione di una certa persona sia per reperire informazioni su una certa persona.

4.5 Centro per lo stato di una persona

Il centro che gestirà lo stato delle persone sarà anch'esso gestito con due artifatti di coordinazione. Uno per inserire informazioni e uno per reperirle. É chiaro che non si potranno inserire informazioni su persone diverse e ciò deve essere garantito da un meccanismo di riconoscimento. Si ipotizza in

questo progetto che problemi a livello di sicurezza e riconoscimento siano fatti ad un livello inferiore del sistema. Si pensi di poter consegnare a seconda del tipo di agenti diversi privilegi di interazione con il sistema. Nel caso non si voglia dare informazioni su di se basterà essere non disponibili.

4.6 Centro smistamento mappe

Per avere informazione di come raggiungere una certa area dell'edificio e ottenere una mappa di esso si dovrà utilizzare un artefatto ad hoc che permetta di interagire con l'agente servizio mappe. Questo centro farà da mediatore fra l'agente "persona" e l'agente servizio anche nel caso che i due "parlino" una lingua diversa.

Figure 1: Coordinazione e scambio di informazioni/servizi tra agenti

4.7 Prenotazione a ricevimento

In questa documentazione non verrà descritto il servizio di prenotazione. Si tratta di dare la possibilità a studenti di prenotarsi al ricevimento di professori.

Ad esempio quando un professore passa in modalità "ricevimento" ogni studente avrà la possibilità di prenotare e disdire un appuntamento.

Potrà conoscere il proprio turno e quanti studenti lo precedono, così per potersi organizzare in anticipo. Il professore da canto conoscerà già i nomi degli studenti che deve ricevere.

Questo servizio che offre l'agente docente disporrà di un centro dove registrare le prenotazioni.

Con la disdicitura di un appuntamento da parte di uno studente, tutti gli altri avranno la possibilità di aspettare un turno in meno, quindi ogni attore che partecipa a questo servizio verrà avvisato di modifiche e si comporterà di conseguenza.

5 Un 'nuova' architettura di Agenti

Prima di passare allo sviluppo del prototipo del progetto si vuole dare uno sguardo panoramico a cosa a portato il ragionamento ad agenti, quali problemi a risolto e a cosa si potrà giungere in tempio più lunghi adottando questa tecnologia per sistemi distribuiti.

Sono stati identificati tre tipi di agenti. Il primo è quello personale che interagisce con l'umano che rappresenta. Il secondo è l'agente che fornisce servizi ma non interpreta nessun ruolo fisico nel mondo reale. Il terzo è l'agente che impersona un dispositivo. Ci potrebbe essere conflitto fra agenti "dispositivo" e di tipo "personale" ma si potrebbe immaginare che mentre il primo risiede su ogni apparato hardware, il secondo segue l'umano attraverso i dispositivi.

Si vuole dare una visione dove ogni figura che compie azioni in un ambiente virtuale è un agente.

5.1 Agente Personale

Siamo partiti dallo use case della facoltà per munire docenti e professori di un assistente personale. A livello progettuale si identifica un comune tipo di agente, che rappresenta agenti di professori e studenti. Nasce perciò l'idea di un agente personale, capace di interagire con chi rappresenta e con tutti gli altri agenti. É la rappresentazione della volontà della persona in un mondo virtuale dove ogni dispositivo e servizio è vivo. Essi sopravvivono in un ambiente sociale interagendo non solo con altri agenti ma anche con gli umani.

5.2 Agente Servizio

Questo tipo di agente, non ha vita, o meglio non rappresenta nessuno e serve solo per soddisfare esigenze di altri. Può iniziare il suo task da se o grazie alla chiamata di altri, ma può essere mobile. Proprio in concomitanza

con il concetto di servizi software, questo agente viene visto come tale (vedi servizio mappe), in grado di compiere mansioni che altri delegano a lui. In un sistema evoluto potrebbe essere visto come di proprietà di un agente più "intelligente" capace di venderlo ad un altro su richiesta.

5.3 Agente Dispositivo

Nasce dall'esigenza di dare vita hai dispositivi, in modo che essi non siano solo fruitori di servizi ma siano anche capaci di ragionare. In una classifica di intelligenza fra i tre tipi di agente, questi sarebbero in testa, perchè non hanno bisogno di dritte umane, non devono rappresentare nessuno se non loro stessi. Forniscono la rappresentazione umana (reale) di una macchina nel mondo fisico (robot). Forse rimane discutibile la dicitura "Agente Dispositivo", basti pensare ad un gioco virtuale. Mentre il personaggio virtuale che rappresenta l'umano verrebbe gestito da un Agente Personale, i personaggi non rappresentati da nessun umano verrebbero a fare parte di questi agenti. In conclusione sono agenti che impersonano dispositivi o personaggi virtuali. Nel primo caso rimerrebbero fissi su un apparato, nel secondo potrebbero essere mobili.

6 Sviluppo

Come già detto in precedenza l'infrastruttura che si andrà ad utilizzare è quella di TuCSoN, e per come è stato concepito, gli artefatti di coordinazioni saranno i Centri di Tuple programmabili dove verranno implementate le regole per coordinare e fare interagire più agenti.

Per motivo di tempo e praticità in questo primo prototipo gli agenti non verranno schematicamente suddivisi come citato nelle precedenti considerazioni, in attesa dell'uscita di una nuova versione di TuCSoN, capace di identificare il tipo di agente per consegnare ad esso un interfaccia di interazione con il sistema con privilegi differenti.

Si parte cercando dove localizzare il sistema ovvero i nodi di TuCSoN e perciò i loro centri di tuple. É ovvio che questi centri di tuple dovranno essere sempre raggiungibili e stabili nel sistema. Perciò dispositivi fissi, collegati wired alla rete, quali PC desktop all'interno della facoltà sono i candidati principali. Dei centri di tuple citati non è possibile costruirne più copie, questo per univocità delle informazioni. Si decide quindi di depositarli su un unico nodo, che si ipotizza sempre stabile ed incrollabile. Non conviene spargere i centri su più nodi perchè a questa versione del sistema la mancanza di un solo centro causa in ogni modo il crash dell'applicazione.

Di seguito si vuole dare una rappresentazione figurativa di come si potrebbe rappresentare fisicamente l'ambiente nel quale il sistema vivo. Si immagini ogni studente e docente munito di assistente digitale (PDA cellulare laptop) che può portare con se mantenendolo collegato ad una rete wireless. Su di

esso risiederà l'agente personale. Alla rete wireless sarà collegato anche un server che conterrà il nodo con i centri di tuple raggiungibile da ogni agente e alcuni agenti servizio di uso comune (come il servizio fornitura mappe).

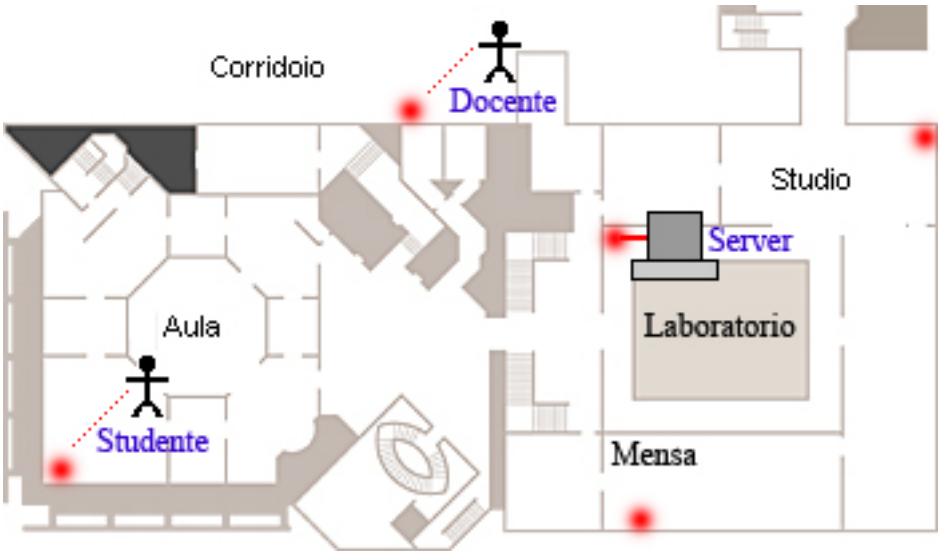

Figure 2: Use case di ambiente dove cooperano gli agenti

Si può intuire che si necessitano principalmente di tre centri di tuple, uno per la localizzazione, uno per lo stato delle persone e uno per il servizio di reperimento mappe. Prima di andare a programmare i centri di tuple (fondamentale aspetto di TuCSoN), si vuole fare un ultimo briefing implememantativo sugli agenti. Nel corso ci sono stati forniti gli strumenti necessari per costruire agenti sia in Prolog che in Java. Questa motivazione spinge il progetto nella direzione di scegliere una delle due maniere per costruirli.

6.1 Docente Agent

Questo è un agente del tipo personale e verrà scritto in Java, non tanto perchè dovrà risiedere su una Micro JavaVirtualMachine, non si dimentichi il fatto che esiste già un motore prolog che gira su JVM (tuProlog) e sul fatto che esista una versione Micro (tuPrologME), quanto più per facilitare l'interazione con interfacce utente e reperire informazioni a livelli sottostanti (localizzazione).

Questa prima versione dell'agente docente sarà in grado solamente di fornire il proprio stato e la propria posizione. Per limiti di tempo si è deciso di fornire solo l'agente studente di strumenti per visualizzare queste informazioni.

L'agente terminerà con la chiusura della propria interfaccia grafica.

6.1.1 Localizzazione

Al seguente agente si vuole permettere la localizzazione di se stesso e rendere disponibile ad altri queste informazioni. Per reperire le informazioni dovrà esserci un costante monitoraggio delle rete per definire a quale punto di accesso si è collegati. Avendo supposto un palmare connesso alla WLAN come dispositivo personale di docenti, allora sarà l'identificatore dell'access point a definire dove ci si trova. Quindi ogni volta che vengono variate informazioni inerenti a chi fornire informazioni sulla localizzazione verranno modificate le specifiche del centro di tuple, ed ogni volta che avviene handover fra un access point ed un altro la tupla che contiene l'area di presenza verrà variata. Le tuple che verranno inserite nel centro saranno del tipo:

location_teacher('teacher','area')

6.1.2 Stato

L'agente deve definire anche lo stato (o profilo) del professore. Per semplicità ipotizziamo che qualsiasi agente potrà conoscere lo stato del professore che sarà lo stesso indipendentemente da chi lo vuole conoscere. Questo verrà impostato dal docente attraverso l'interfaccia utente in comunicazione con l'agente. Le tuple inserite nel centro saranno del tipo:

teacher_profile('teacher','profile')

6.2 Studente Agent

Un agente studente avrà la possibilità di localizzare il professore e sapere in che stato si trova, in più potrà avvalersi del servizio di mappe all'interno dell'edificio.

L'agente terminerà con la chiusura della propria interfaccia grafica.

6.2.1 Localizzazione

Tutto è come nel caso dell'agente docente. La localizzazione, ipotizzando che lo studente abbia come dispositivo personale un cellulare UMTS, verrà effettuata recuperando informazioni attraverso il GPS in aree esterne e attraverso Bluetooth in aree interne. Le tuple che verranno inserite nel centro saranno del tipo:

location_student('student','area')

Per localizzare un docente non dovrà fare altro che accedere al centro di tuple di localizzazione e leggere le tuple inserite dal professore che interessa. Grazie al motore prolog che pulsa dentro i centri di TuCSoN le tuple da

leggere saranno del tipo:

location_teacher('teacher',Area)

Area sarà una variabile che unificherà con il luogo dove il docente si trova. Questa informazione verrà in seguto visualizzata allo studente.

6.2.2 Stato

Anche la parte sullo stato non si differenzia da quella del professore se non per il fatto che esisterannno profili diversi da scegliere. Le tuple che verranno inserite nel centro saranno del tipo:

student_profile('student','area')

Queste informazioni in questo prototipo non verranno reperite da nessuno ma si crede che in una seguente versioni siano utili per l'interazione fra studenti. Per conoscere lo stato di un professore da visualizzare allo studente la procedura è la stessa di quella eseguita per la localizzazione. La lettura della tupla inerente nel centro di tuple che gestisce gli stati.

teacher_profile('teacher',Profile)

6.2.3 Servizio Mappe

Per poter interagire con il servizio mappe si dovrà fare in modo di fare una richiesta attraverso una tupla inserita nel centro di tuple inerente. Un provider di mappe localizzerà la richiesta e farà in modo di fornire una mappa a chi la necessita. La tupla di richiesta servizio inserita dall'agente studente sarà del tipo:

map_service('student','mappa')

Ovviamente si potrà scegliere fra diverse mappe, differenziandole fra piani e topologie. Una volta emessa la richiesta ci si mette in attesa di una risposta.

Nella prima versione di prototipo l'attesa era bloccante ma ciò comprometteva il funzionamento dell'agente (si pensi ai problemi che possono insorgere se è venuto a mancare il servizio di consegna mappe). Nella versione finale quasi nessuna richiesta al centro di tuple è bloccante proprio per non compromettere il funzionamento degli agenti. Saranno bloccanti quegli accessi solo per agenti che non devono fare niente altro. Ad esempio l'agente che fornisce il servizio di mappe sta continuamente in attesa di una tupla.

In questo specifico caso, viene bloccato il thread per un secondo per dare la possibilità al servizio mappe di fornire la mappa. La risposta del servizio che si preleverà dal centro di tuple "mappe" (lo stesso dove si è fatta la richiesta) sarà la tupla:

map_url('student',Url)

La variabile Url avrà unificato con un indirizzo http dove andare a recuperare la mappa richiesta dallo studente student. Un interfaccia grafica la visualizzerà allo studente.

6.3 Servizio Mappe Agent

Questo agente farà parte di quella tipologia di agenti che è stata definita di servizio, ovvero che svolgeranno compiti richiesti da altri agenti. Esso verrà costruito in prolog, siccome non ci sono interfaccie e umani con cui interagire il linguaggio logico per gestire delle tuple è molto più semplice da utilizzare. Ricordiamoci che l'agente risiederà su un dispositivo fisso e avrà con se un database di mappe dell'edificio da fornire agli agenti che ne necessitano. In realtà a seconda della richiesta fornirà un Url che rappresenta la localizzazione dell'immagine mappa desiderata. Esso starà continuamente in attesa di richieste da parte di qualche agente sul centro di tuple mappe. Questi tipi di agenti sono agenti task oriented che hanno come unico scopo di vita il soddisfacimento di un certo compito. Appena avverrà la richiesta il servizio sarà reso disponibile dall'agente che invierà la mappa all'interessato. La tupla che l'agente attende di consumare è

map_service('mappa','locUsr','Usr')

Dove Usr è l'utente che ha fatto la richiesta e locUsr la sua posizione. Questo perchè il servizio fornisce mappe diverse di uno stesso luogo a seconda di dove il richiedente si trova in modo da facilitarne la lettura. Si può notare che la tupla da consumare non è la stessa di quelle inserite dagli studenti. Sarà a questo punto che il centro di tuple mappe, essendo programmabile attraverso le specifiche respect fungerà da mediatore fra client e provider del servizio che non parlano lo stesso linguaggio. La tupla che il servizio restituirà con l'url della mappa sarà:

map_url('usr','mappa')

Si nota che questa tupla è identica a quella che lo studente consuma per reperire la mappa.

6.4 Localizzazione Tuple Center

Il tuple center sulla localizzazione sarà lo stesso per professori e studenti. Esso conterrà le tuple che individuano la posizione di studenti e professori.

6.5 Stato Tuple Center

Il tuple center dei profili sarà lo stesso per professori e studenti. Esso conterrà le tuple che individuano lo stato di studenti e professori.

6.6 Servizio Mappe Tuple Center

Infine il seguente tuple center che fungerà da mediatore fra il fornitore e l'utilizzatore del servizio potrà essere programmato dall'esterno nel caso in cui i due agenti interagenti non parlino la stessa lingua. Si pensi ad un servizio mappe che richiede oltre alla mappa la localizzazione del richiedente per facilitarne l'uso. Allora bisognerebbe che il centro di tuple conoscendo il client (agente studente) riuscisse a fornire indicazione sulla sua posizione. Lo si potrebbe fare ad esempio rilegando ad un altro agente ausiliare il compito di spostarsi nel tuple center di localizzazione per prelevare questa informazione. Questo è proprio quello che le specifiche respect permettono di fare. Rilegando ad un agente di servizio (SetSpec.java) la scrittura delle specifiche nel centro di tuple, esse consitono in due reazioni, la prima alla richiesta di servizio dello studente:

reaction(out(map_service(St,Map)),(
spawn(findStud,prolog('find'),[St]))).

Infatti in seguito all'immissione della tupla di richiesta servizio, viene lanciato un agente di servizio "find" il quale avendo come parametro lo studente andrà nel centro di localizzazione per recuperare la sua posizione. In seguito facendo la out della tupla

location_student(St,'area')

comunicherà al centro mappe la posizione dello studente.

La seconda reazione del centro di tuple mappe sarà a questa immissione di tupla:

reaction(out(location_student(St,Loc)),(
in_r(map_service(St,Map)),
in_r(location_student(St,Loc)),
out_r(map_service(Map,Loc,St)))).

A questo punto il centro consuma sia la tupla dello studente per il servizio, sia la tupla che lo localizza per fornire al servizio mappe una tupla per esso comprensibile con i dati necessari (mappa,localizzazione del richiedente e identificazione del richiedente).

7 Simulazione

Siccome si tratta di un prototipo e si è deciso di non trascorrere troppo tempo sulle difficoltà tecnologiche quanto più sulla coordinazione degli agenti, il problema della localizzazione è stato simulato. Con tecnologie innovative che usciranno nei prossimi mesi e scarse possibilità per ora di testare l'applicazione su terminali quali PDA e cellulari si è deciso di utilizzare PC sui quali far girare l'applicazione e interfaccie grafiche per simulare gli spostamenti. Viene simulato l'handover fra un punto di accesso ed un altro alla rete, in realtà si può benissimo pensare di simulare la presenza di sensori nell'ambiente. A questo livello non interessa la tecnologia che si utilizzerà per la localizzazione. Perciò per ogni agente di tipo "personale" un'intefaccia permette di scegliere in che punto di facoltà virtuale ci si trova.

La simulazione verrà testata su un unica macchina. Per rendere distribuito il sistema basta caricare gli agenti su macchine diverse e definire all'interno di ognuno l'indirizzo della macchina in cui sta il nodo TuCSoN per utilizzare i centri di tuple.