UNIVERSITÀ DEGLI STUDI DI TRENTO

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea in Informatica

Elaborato Finale

Estensione del sistema di carpooling Andiamo per comunità di utenti co-localizzate

Relatore:
Prof. Paolo Giorgini

Laureando:
Paolo Cesari

ANNO ACCADEMICO 2006-2007

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Negli ultimi decenni abbiamo assistito ad un continuo e progressivo sviluppo tecnologico, che ci ha portato ad una realtà in cui la diffusione di apparecchi mobili, quali cellulari e palmari, è a tal punto elevata che sembra di poterne fare meno. Viviamo infatti in un mondo in cui la tecnologia sembra essere diventata un bisogno primario dell'uomo, senza la quale l'individuo medio si sente escluso ed emarginato: è un dato ormai consolidato che la maggior parte delle comunicazioni interpersonali avvenga tramite cellulari, connessioni internet e dispositivi Bluetooth.
Sulla scia del successo di questi modelli comunicativi, le grandi compagnie si sono spinte alla ricerca di nuovi metodi per gestire i rapporti interpersonali, creando in tal modo cellulari sempre più tecnologici, in grado non solo di mettere in comunicazione via voce due persone, ma anche di consentire lo scambio di informazioni in formato elettronico e la memorizzazione su cellulare di un sempre più elevato quantitativo di dati.
Tuttavia non sempre tale potenziale tecnologico risulta essere completamente sfruttato: sovente infatti, ove esistono gli strumenti, non ci sono le idee per adoperarli in modo costruttivo. In altre parole allo stato attuale si tende a considerare i dispositivi di cui siamo in possesso come mero svago e passatempo anzichè come possibile mezzo per contribuire alla risoluzione di alcune problematiche attuali; infatti proprio gli utenti che utilizzano tali dispositivi formano delle comunità virtuali mobili, ovvero scambiando informazioni tra membri geograficamente co-localizzati, accomunati dal fatto che condividono interessi comuni offerti da utenti attivi in una stessa zona. In questo contesto si inserisce appunto il presente progetto: il nostro scopo consiste nella creazione di un sistema che consenta di sfruttare al meglio la posizione dell'utente all'interno del sistema Andiamo per contribuire, nel nostro piccolo, sia al miglioramento del problema dell'inquinamento atmosferico derivante dalle emissioni delle autovetture, sia all'ottimizzazione del servizio del sistema offerto al cliente in un determinato luogo.
Con il trascorrere del tempo la questione dell'inquinamento atmosferico è diventata sempre più attuale: è un dato infatti incontrovertibile che con il passare degli anni si stia andando incontro ad un progressivo e sensibile peggioramento non solo della salubrità dell'aria, ma anche della qualità della vita in generale, questione questa decisamente da non trascurare.
Numerose sono le cause dell'inquinamento atmosferico che una buona politica di contenimento dello stesso dovrebbe tenere in considerazione: si pensi ad esempio all'elevato utilizzo di impianti di riscaldamento, alle emissioni inquinanti di fabbriche e inceneritori e, last but not least, alle emissioni di autovetture. Il

presente progetto si concentra proprio su quest'ultimo fattore e intende creare un sistema in grado di ottimizzare gli spostamenti effettuati dalle persone che compiono lo stesso tragitto condividendo il proprio mezzo di trasporto(privato), usufruendo del sistema di car-pooling Andiamo al fine di ridurre il congestionamento del traffico nelle aree urbane e non.
E'infatti evidente che una politica idonea ad incentivare gli automobilisti (che compiono il medesimo tragitto) a coordinarsi tra loro, per adoperare in comune una sola autovettura (anzichè un mezzo a persona)e servirsi di un servizio che tenga conto della posizione dei vari fruitori, genera numerose esternalità positive quali la limitazione delle emissioni inquinanti, la riduzione della congestione del traffico, e in ultimo la creazione di una nuova occasione di socializzazione da parte delle persone di una zona, rendendo in tal modo il tragitto più piacevole per i soggetti coinvolti.
Di qui la scelta di implementare un sistema semplice che sia in grado di sfruttare la tecnologia già presente, per mettere in comunicazione tra loro persone dislocate nel medesimo territorio, raggruppate da tragitti in comune, in modo da aggregare i possibili utenti in un servizio di carpooling circoscritto: sì è in altre parole progettato e sviluppato un sistema in grado di far incontrare persone colocalizzate che ricercano-offrano passaggi. Per ottenere tale risultato si è deciso di avvalersi dell'ampia diffusione di dispositivi mobili altamente tecnologici tra la popolazione: in altre parole si tenta di adoperare la tecnologia per risolvere i problemi che da essa stessa sono stati generati!
Questo lavoro di tesi presenta quindi un sistema costituito da un server contenente un database e access-point Bluetooth, accessibili attraverso l'utilizzo di telefoni cellulari/PDA, sfruttando la tecnologia Bluetooth; proprio questa via di comunicazione è il punto cardine del lavoro poichè permette di localizzare gli utenti all'interno del sistema e di raggrupparli in quanto frequentano uno stesso ambiente per poi poter fornire loro un servizio in comune.
Il lavoro si articola come di seguito: il primo capitolo servirà per introdurre Andiamo, analizzando il problema, il contesto sociale in cui il sistema prenderà piede ed infine si affronterà il tema delle comunità virtuali mobili. Il secondo capitolo verrà trattata l'architettura generale, i componenti del progetto e le scelte di implementazione adottate; proseguendo si descriverà invece un esempio di applicazione del sistema progettato,fino a giungere all'ultimo capitolo dove verrà fatto il punto della situazione sul progetto affrontato, traendone le conclusioni e proponendo possibili ulteriori sviluppi futuri del sistema.

Capitolo 1

Il sistema di carpooling Andiamo

Appare opportuno, prima di trattare delle specificità del presente progetto, soffermarsi brevemente su alcune questioni preliminari quali ad esempio il contesto sociale entro cui dovrà inserirsi e le principali componenti tecnologiche che sono state adoperate per la costruzione del sistema di carpooling Andiamo. In quest'ottica il primo capitolo sarà interamente dedicato all'analisi del problema che ha portato a sviluppare tale applicazione, affrontando (per questioni logistiche solo per sommi capi) soprattutto i provvedimenti che sono stati adottati in materia di riduzione delle emissioni inquinanti delle autovetture. In particolare verrà descritto nello specifico il sistema Andiamo considerando gli aspetti della sua architettura, i suoi componenti per giungere infine ad un discorso generale che tende a considerare gli aspetti positivi e negativi dell'applicazione ed il concetto di comunità virtuale mobile. **Per contro il secondo e il terzo capitolo** saranno dedicati in primo luogo all'illustrazione delle componenti tecnologiche adottate per implementare il nostro progetto e successivamente ad una breve spiegazione, attraverso un esempio pratico, del sistema che, come descritto brevemente nell'introduzione, consiste nella predisposizione di un'applicazione in grado di creare una rete di comunicazione che metta in contatto tra loro i vari utenti di un servizio di car-pooling co-localizzati.

1.1 Caratteristiche generali

1.1.1 Risolvere il problema dell'inquinamento

Negli ultimi tempi il problema dell'inquinamento sembra essersi fatto più importante: sempre maggiore è infatti il numero di conferenze internazionali, leggi comunitarie e nazionali e trattati internazionali che cercano di costruire un mondo, se non migliore, certamente più pulito. Meritano a tal proposito di essere ricordate due conferenze sopra tutte: l'una tenutasi a Rio de Janeiro nel 1992

[16] e l'altra a Kyoto nel 1996 [20], con le quali la comunità internazionale ha definitivamente e compiutamente sancito l'importanza della questione dell'inquinamento ambientale e della relativa necessità di promuovere azioni concrete per far rientrare nell'arco di un paio di anni il tasso di inquinamento globale entro valori soglia di contenimento. Nello specifico, la prima delle due conferenze si poneva il preciso scopo di regolamentare le emissioni di gas ad effetto serra a livelli che impedissero l'interferenza antropogenica sul sistema climatico, mentre la seconda, in questo senso maggiormente rilevante ai fini dell'argomento che di seguito verremo a trattare, impegna i Paesi industrializzati a ridurre, entro il periodo compreso tra il 2008 e il 2012, le proprie emissioni complessive di CO_2, CH_4, N_2O, HFC, PFC e SF_6, del 5,2% rispetto ai livelli del 1990. In ogni caso allo Stato attuale 169 sono gli Stati aderenti; tra i Paesi firmatari spicca tra tutti la non presenza degli Stati Uniti d'America, assenza motivata dal fatto che essi stessi producono circa il 36,1% del totale delle emissioni (dato aggiornato al marzo 2001).
Occorre in ogni caso precisare che la mancata ratifica del protocollo di Kyoto da parte degli Stati Uniti d'America non significa che essi abbiano fin'ora trascurato il problema dell'inquinamento ambientale: i primi interventi americani in campo ambientale risalgono già infatti al lontano 1962 con l'emanazione del Clean Air Act [12], successivamente emendato nel 1966, 1970, 1977 e nel 1990, tramite il quale il Congresso prevedeva un ulteriore abbassamento dei limiti di emissioni di autoveicoli privati e industriali e di nuovi per le industrie. Da segnalare a tal proposito anche l'emanazione nel 1970 da parte dell'EPA (Environmental Protection Agency) di valori soglia standard per le emissioni al fine di proteggere la salute pubblica [9].
Intanto nel 1989 in tutti gli stati d'Europa eccetto Germania, Danimarca e Olanda si iniziava a vendere la benzina "verde " senza piombo con 95 ottani. Solo a partire dall'ottobre del 1993 vennero introdotte nuove restrizioni: in particolar modo si ricorda la direttiva europea 441/91 che diede il via alla fase Euro1, ai sensi della quale i veicoli di ogni cilindrata potevano circolare solo se muniti di marmitta catalitica; in tal modo si è riusciti a ridurre drasticamente le emissioni degli autoveicoli a benzina con importanti e positive ripercussioni su determinanti inquinanti (in particolare benzene, monossido di carbonio ed ossidi di azoto). Di fatto, si era dato inizio a un viaggio che avrebbe portato le vetture europee a essere fra le meno inquinanti nel mondo: con il decorso del tempo le normative si inasprirono sempre di più fino ad introdurre le classi Euro2, Euro3 ed Euro4 [3]. Nota negativa della normativa europea è rappresentata dai limiti previsti per i veicoli diesel(che rappresentano quasi il 50% delle autovetture in circolazione tra i quali camion e bus) ai quali è d'obbligo il filtro antiparticolato [1] ed i veicoli a due ruote (ed in particolare quelli funzionanti a miscela), che non sembrano in grado di ridurre in modo sufficiente le emissioni di polveri fini ed ultrafini. Un altro intervento importante consiste nella predisposizione di incentivi per l'utilizzo di combustibili alternativi, tra i quali spicca in primo luogo il gas metano, che però ha il limite di una rete di distribuzione inadeguata, il GPL, ed infine il biodiesel (oli di origine vegetale). Una vera e propria rivoluzione è costituita dai veicoli ad idrogeno, che non emettono alcun inquinante (solo acqua),ma la tecnologia di settore non sembra essere in grado di poter offrire in tempi brevi tali veicoli per il grande pubblico [10]. Gli interventi fin'ora descritti sono tutti mirati alla creazione di vetture sempre più pulite in quanto si è visto che la gran parte degli inquinanti sono emessi nelle aree urbane per lo

più generati dal traffico veicolare che diviene con il passare degli anni una realtà sempre più avvertita non solo dai governi e dalle organizzazioni ambientaliste, ma anche dalla stessa popolazione; proprio quest'ultima si preoccupa dell'inquinamento, della vivibilità e funzionalità della città, sottolineando la mancanza nelle aree urbane di apposite strutture per pedoni e ciclisti che sono stati privati della loro sicurezza e dei loro spazi per favorire la motorizzazione di massa. Da questo momento si inizia ad utilizzare il termine "*Mobilità sostenibile*" per riferirsi all'esigenza di avere un sistema di mobilità urbana che, pur consentendo a ciascuno l'esercizio del proprio diritto alla mobilità, sia tale da non gravare eccessivamente sul sistema sociale in termini di costi esterni, quali inquinamento atmosferico ed emissioni di gas serra, inquinamento acustico, congestione di traffico stradale ed infine il fattore incidentalità [21]. La mobilità sostenibile mira quindi a sviluppare una strategia efficiente che assicuri una mobilità delle persone e un trasporto di merci, rispettosi tanto delle caratteristiche ambientali e sociali dei singoli luoghi, quanto della questione del risparmio energetico. A tal fine si propone di limitare il numero di autovetture circolanti in favore di mezzi di locomozione alternativi, migliorando così l'accessibilità al centro urbano e diminuendo il grado di concentrazione di sostanze inquinanti.
Provvedimenti finalizzati ad incentivare una mobilità sostenibile iniziarono ad affermarsi solo dagli inizi degli anni '90 negli Stati Uniti e in alcuni paesi Europei come Gran Bretagna, Svizzera, Belgio, Olanda ed infine in Italia, dove un primo passo è stato compiuto con il *decreto del 3 agosto del 1998*. Con esso si è sottolineata l'importanza della promozione di azioni di sostegno e incentivazione agli enti locali affinché questi attuino interventi volti a migliorare la qualità della vita urbana a favore dell'infanzia, cercando di attivare un maggiore e capillare coinvolgimento della popolazione [14]. Sempre con tale direttiva sono stati previsti, per il personale dipendente, vari premi per incentivare la scelta di mezzi di trasporto pubblici, piuttosto che quelli privati, per lo spostamento casa-lavoro. All'interno delle Pubbliche Amministrazioni e delle grandi Aziende a partire dal 5 aprile del 2001 sono infine state istituite delle nuove figure di personale chiamate Mobility Manager [7] , alle quali è devoluto il compito di occuparsi dei problemi di mobilità urbana dei propri dipendenti. Il loro ruolo si concentra essenzialmente su iniziative di informazione del problema, studio delle abitudini, proposte ecologiche e presentazione di nuove politiche come Park Pricing e Road Pricing o zone a traffico limitato.

1.1.2 Rideshare in Trentino Alto-Adige

A questo punto appare opportuno passare da un discorso generale ed astratto ad un'analisi più concreta applicata alla realtà che ci circonda, in quanto è in essa che il presente progetto si è inserito; pertanto ci si soffermerà sulla situazione nella nostra regione(Trentino AltoAdige) e cercando di vedere in luce le caratteristiche salienti del nostro territorio.
I dati derivanti dal rapporto annuale dell'Agenzia Provinciale per la Protezione dell'Ambiente, a causa della conformazione morfologica, la situazione del Trentino Alto Adige risulta essere un'emergenza per quanto riguarda il traffico automobilistico e il derivante abbattimento delle polveri sottili all'interno dei centri urbani. Le misure adottate negli ultimi anni per combattere tale problema sono state le seguenti:

- riduzione del trasporto merci su gomma e incremento del trasporto su rotaia lungo la direttrice del Brennero;
- sviluppo di trasporto elettrico o ibrido (elettrico + metano o GPL) urbano;
- riduzione del trasporto passeggeri su strada mediante l'incremento delle piste ciclabili;
- disincentivazione dell'uso del mezzo privato nei principali centri urbani (tramite estensione delle zone di sosta a pagamento ...);
- limitazione alla circolazione di veicoli euro 0 e diesel euro 1;
- riduzione del trasporto passeggeri su strada incentivando la condivisione dei mezzi privati tra più passeggeri (interventi di car pooling), con l'obiettivo di raggiungere un tasso medio di occupazione delle auto in circolazione pari a due persone per veicolo.

In particolare quest'ultimo appare essere un vero punto di forza per la creazione di un piano di trasporti sostenibile, tant'è che gli stessi interventi locali tendono a favorire il rideshare [13] al punto da consentire la circolazione di autovetture anche in deroga alle normali restrizioni posto che trasportino più di tre persone a bordo (e questo al fine di incentivare una riduzione del quantitativo di vetture circolanti, ottenibile appunto incentivando gli utenti a condividere un unico mezzo in caso di comunanza di partenza e destinazione).
In tal caso compete al cittadino l'onere di trovare alcuni partners con cui condividere il viaggio, ed è proprio per tale motivo che si è cercato di costruire un sistema basato su dispositivi mobili, oltre a quelli già esistenti in internet che fossero in grado di far incontrare domanda e offerta. In altre parole mancava un sistema che fosse in grado di mettere in contatto in maniera automatica e capillare coloro che disponessero di posti liberi in auto e coloro che al contrario fossero alla ricerca di un passaggio; infatti veniva offerto un servizio di rideshare su Web [11] naufragato perchè poco conosciuto e scomodo da utilizzare a causa dell'impossibilità di disporre di un collegamento ad internet quasi costante, oltre alle difficoltà tra cui l'accesso al sistema ed in ultimo(ma non per importanza)la delicata questione della privacy. La creazione del sistema Andiamo ha risolto tanti problemi come ad esempio quella di utilizzare applicazioni mobili per sviluppare il Car Pooling, sfruttando l'ampia diffusione di cellulari all'interno della popolazione per offrire un servizio al quale si può accedere ovunque ed in ogni momento. La realizzazione del sistema Andiamo dovrebbe aver agevolato non poco l'operatività del modello del rideshare in quanto in tal modo i singoli autisti che devono riempire la propria autovettura non dovrebbero più far affidamento solo sulle proprie conoscenze personali ma potrebbero avvalersi di un bacino di utenza assai più distribuito.

1.1.3 Architettura

Dopo aver analizzato il contesto in cui si è inserito Andiamo, è giunto ora il momento di descrivere le principali caratteristiche del sistema in questione [19]; consiste in un numero di piattaforme multi-agente accessibili da dispositivi mobili che supportano comunicazioni Bluetooth. L'utente può accedere al sistema

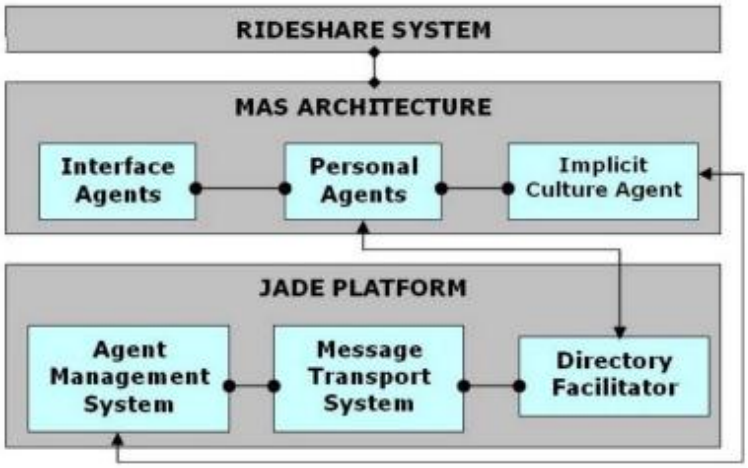


Figura 1.1: Framework rideshare a tre livelli

attraverso una rete di access-points Bluetooth direttamente collegati alle piattaforme multi-agente sulle quali è stato implementato un sistema dove i personal agents (PA) dei ride-offerer e ride-seeker interagiscono e contrattano eventuali passaggi automobilistici. La struttura del sistema di rideshare è stata sviluppata con un approccio su più livelli.

Come mostrato in Figura 1.1, il modello è composto da tre livelli e partendo dal più basso è costituito da:

1. una piattaforma ad agenti (JADE)
2. un'architettura multi-agente con il modulo di Cultura Implicita
3. all'estremità superiore,un sistema vero e proprio di rideshare

1.1.4 La piattaforma ad agenti

La piattaforma ad agenti è implementata in JADE (java Agent Development Framework) [5], conforme alle specifiche FIPA [4] per lo sviluppo di sistemi multi-agente. JADE inoltre fornisce un supporto completo per l'interazione e la contrattazione tra i personal agents; essa implementa:

1. un sistema di gestione degli agenti (AMS- Agent Management System) che permette di avere contenitori di agenti in differenti hosts (una piattaforma distribuita),

2. un Directory Facilitator (DF) che fornisce il servizio di pagine gialle

3. un sistema di trasporto dei messaggi (MTS-Message Transport System) che supporta la comunicazione tra gli agenti.

Nel sistema ogni utente viene rappresentato da un dispositivo mobile e inserito all'interno di una piattaforma attraverso un solo personal agent: l'indirizzo Bluetooth del corrispondente dispositivo mobile fa subito risalire quindi al fruitore; quest'ultimo comunque può avere altri agenti distribuiti su differenti piattaforme. Ogni personal agents scambia informazioni e interagisce con altri agenti in modo da trovare dei partners, per soddisfare i desideri del proprio soggetto, e resta attivo finchè rimane la richiesta da soddisfare.

1.1.5 Il modello di Cultura Implicita

La Cultura Implicita è la relazione che ha lo scopo di permettere ai nuovi membri della comunità di comportarsi in accordo con la cultura della comunità stessa e dell'ambiente in cui si trovano; tale concetto è basato su tecniche di Data Mining ed estende tecniche di Collaborative Filtering; per esempio, lo studente X potrebbe voler conoscere i punti di ritrovo più utilizzati nelle vicinanze dell'università e il modulo della Cultura Implicita,in questo caso viene adottato per supportare l'utente nel compiere delle scelte. Infatti l'idea di fondo è fare in modo che il sistema suggerisca i punti di incontro che sono stati utilizzati con maggior frequenza in passato da altri studenti universitari (cioè, dai membri della comunità). In questo caso, il sistema potrebbe suggerire allo studente X di muoversi verso il parcheggio delle automobili vicino all'università supponendo sia il luogo più gettonato fino a quel momento.
Grazie all'inserimento e utilizzo del framework di Cultura Implicita nel sistema [15], per ogni piattaforma multi-agente, si è potuto costituire una comunità di utenti, guidando così un nuovo utente ad accedere al sistema in maniera da comportarsi secondo la cultura della comunità stessa e ricavarne i massimi benefici.

1.1.6 Esempio

Andiamo è un sistema di carpooling basato sugli agenti usufruibile da dispositivi mobili dotati di tecnologia Bluetooth.
Ciascun utente è rappresentato da un agente personale che interagisce con gli agenti personali degli altri utenti della comunità virtuale mobile per trovare oppure offrire un servizio di condivisione di passaggi attraverso l'utilizzo di autovetture.

Per comprendere meglio il meccanismo di Rideshare vale la pena spiegare il sistema attraverso l'ausilio di un esempio come mostrato in Figura 1.2. Supponiamo ci siano tre attori A,B e C e siano P_1, P_2, P_3 tre differenti luoghi (o punti di incontro) tra loro connessi e facilmente raggiungibili dalle automobili. Si supponga inoltre che i passeggeri e i proprietari d'auto siano distribuiti casualmente tra queste località e che entrambe le tipologie di utente condividano i medesimi interessi e necessità (es. destinazione desiderata, momento e punto di

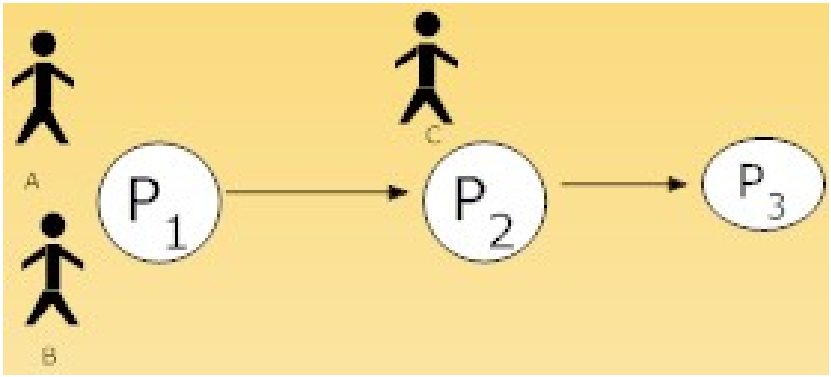


Figura 1.2: Interazione

partenza). Consideriamo il punto P_1 come la stazione dei treni, P_2 la fermata del bus e P_3 l'università. Inoltre, A e B due differenti persone che si trovano in P_1 mentre C sia un'altra persona che si trova presso P_2. Queste persone hanno un unico obiettivo finale che consiste nel raggiungimento della stessa destinazione, che assumiamo possa essere in questo caso P_3.
Supponiamo che A sia un ride-offerer, mentre B e C due ride-seeker; appare evidente che se queste tre persone si mettessero d'accordo potrebbero percorrere assieme il viaggio fino alla meta P_3 mediante l'utilizzo dell'automobile di A.
L'utente B potrebbe formulare la richiesta nella seguente maniera:"Cerco un passaggio verso il luogo P_3 " senza specificare esattamente il punto di partenza, producendo così una richiesta incompleta. A questo punto il Personal Agents relativo all'utente B rende la richiesta più chiara usando le informazioni riferite ai punti di incontro che altre persone hanno recentemente e in passato indicato per la medesima località. Un altro scenario interessante che potrebbe presentarsi è il seguente: l'utente A potrebbe richiedere: "Io voglio sulla mia auto solo passeggeri con un valore di feedback minimo pari a 100 " e allo stesso tempo non vi è un utente nel sistema che abbia un valore di feedback uguale a 100. In questo caso, l'azione che il Personal Agents, dopo aver ottenuto il nullaosta dall'utente contattato tramite l'invio di un messaggio al suo dispositivo mobile, sarà ridurre il valore di feedback indicato dall'utente. Tutto ciò è possibile soltanto se vi è un sistema che fa incontrare queste persone (non fisicamente, ma virtualmente) e magari facilita loro il raggiungimento di un accordo che tenga conto delle loro preferenze, come il sistema di rideshare Andiamo.

1.2 Le comunità virtuali mobili

Con lo sviluppo delle tecnologie per la comunicazione mobile, intesa non solo come comunicazione vocale ma anche come scambio di informazioni (sms, mms, foto e in generale ogni tipo di file) è nata l'idea di formare dei gruppi entro i quali le persone potessero interagire, confrontarsi, comunicare. A questi gruppi di persone è stato dato il nome di *comunità virtuali* [17] poichè i membri che le compongono possono anche non conoscersi e non avere rapporti di alcun tipo

al di là di quelli virtuali appunto. La particolarità di queste comunità sta nel fatto che hanno un carattere mobile perchè utilizzano dispositivi come cellulari, PDAs, notebook, smartphone, ecc. per comunicare e non sono legate al luogo in cui operano, come l' ufficio o il luogo di lavoro: esse vengono quindi chiamate comunità virtuali mobili. Le comunità virtuali in genere (Virtual Community - VC) hanno la caratteristica di privilegiare, nel rapporto tra due o più individui che vivono anche in paesi diversi, la condivisione di interessi comuni; ciò avviene in virtù dell'abbattimento delle barriere socio culturali che nelle comunità reali possono creare distanze comunicative. La comunicazione virtuale permette inoltre il gioco di identità e lo scambio dei ruoli; come ciò possa influire sulla psicologia di un individuo rimane una riflessione ancora poco esplorata in campo psico-sociologico; comunque, l'abilità di mascherare la propria identità può produrre effetti diversi, negativi e positivi ed è una cosa da tenere in considerazione quando si vuole offrire un servizio come il rideshare.
Le comunità virtuali forniscono informazioni a cui è possibile accedere ovunque; tali informazioni tuttavia, per essere accessibili ed usabili in ogni luogo e in qualunque momento, necessitano di una tecnologia di comunicazione appropriata che diviene l'elemento cruciale del sistema. Dalla combinazione delle tradizionali tecnologie internet con le nuove capacità delle reti mobili, si ottiene un approccio per soddisfare la richiesta di accesso e connessione in ogni momento e ovunque ci si trovi. Le comunità virtuali mobili hanno così l'enorme potenziale di poter servire in qualunque momento i desideri degli utenti.
Howard Rheingold, conosciuto nella comunità scientifica per aver inventato il termine comunità virtuale, ha recentemente ideato anche l'espressione Smart Mobs: con tale termine ha voluto riferirsi ad un gruppo di persone facente parte delle più moderne comunità high-tech intelligenti in cui le persone utilizzano le molteplici tecnologie oggi disponibili per organizzarsi e coordinarsi in azioni collettive di vario genere anche se non si conoscono tra di loro [18]. Una comunità virtuale può tuttavia essere costituita in un qualsiasi ambito e con qualsiasi tecnologia mobile: basti pensare ad un'università, in cui gli studenti con interessi sostanzialmente comuni potrebbero formare dei gruppi di discussione per argomenti specifici, per lo scambio di informazioni e appunti, per l' organizzazione di eventi culturali e sportivi. Utilizzando i dispositivi mobili, oltre all'uso tradizionale di internet, e-mail, blog, forum, potrebbero rimanere sempre in contatto e dialogare: tutto questo senza l'obbligo di conoscersi personalmente ed incontrarsi. Le prospettive aperte per il futuro dalle comunità virtuali mobili porteranno ad avere nuovi modi di comunicare, agire, fare affari come si sta già verificando oggi con i cellulari che usano la rete di terza generazione. Diviene quindi chiara la necessità di supportare queste comunità al fine di agevolare la comunicazione interna dei membri che la compongono.
Esistono numerose proposte in letteratura di sistemi a supporto di comunità virtuali mobili in molteplici contesti; il cuore di questi sistemi, al di là della tecnologia di comunicazione adottata, solitamente si trova nella piattaforma che gestisce gli utenti e che mira al soddisfacimento dei loro desideri, sia che essi ricerchino semplici informazioni statiche, sia che necessitino di complicate interazioni e contrattazioni per ottenere il risultato finale. Un esempio di progetto di ricerca nel campo del supporto informativo ai turisti e di utenti mobili è dato da Portable-Cicero [8] : esso utilizza dei raggi IR posti all'entrata di ogni sala di un museo per individuare il posto esatto in cui si trova l'utente per poi poter fornire ad esso tutte le informazioni di cui necessita. L'obiettivo di questo lavoro

è fornire uno strumento che faccia da cicerone al turista all'interno del museo. Infatti grazie a delle mappe pre-caricate sul PDA in possesso dell'utente, il sistema può mostrare una piantina della stanza con le opere presenti; a questo punto il turista può selezionare l'opera a cui è interessato e ottenere informazioni su di essa tramite un'interfaccia grafica. PortableCicero presenta un solo grande inconveniente: richiede una buona dimestichezza con i computer palmari e ciò purtroppo limita il numero di utenti che possono usufruire del servizio. In talune proposte l'utente si aspetta che il proprio software esegua del lavoro per lui in modo autonomo, proattivo e/o propositivo contrattando, collaborando o entrando in competizione con software di altri utenti. A situazioni come la precedente bene si addice un sistema ottenuto con la programmazione ad agenti; questo paradigma prevede infatti che sulla piattaforma multi-agente vivano ed operino delle entità software, dette appunto agenti, che agiscono in maniera del tutto trasparente per il bene dell'utente.

1.3 Considerazioni generali

Come detto in precedenza il sistema Andiamo nasce per sostituire tutte le soluzioni che, in passato, sono state sviluppate per fornire agli utenti un servizio di rideshare basato su web, che tuttavia non avevano riscosso successo poichè vi erano problemi legati alla limitata accessibilità ed alla mancanza di un'infrastruttura di supporto per riuscire a soddisfare la grande mobilità cittadina. Questa volta si è cercato di sviluppare un'applicazione che consentisse sia di sfruttare la diffusione capillare della tecnologia per inserirvi il servizio, sia di avere gli strumenti per coordinare la parte relativa alla gestione delle richieste. Per ottenere tale risultato si e scelto di sfruttare la grande diffusione dei dispositivi mobili, in particolare dei cellulari, e delle loro sempre maggiori funzioni avanzate (oltre che alle semplici chiamate), per consentire agli utenti che intendano avvalersi dei sistema di rideshare, di comunicare tra di loro e costituire,in tal modo, una comunità virtuale mobile. Nello specifico si è scelto di adoperare i dispositivi Bluetooth, collegandoli tra loro tramite server multi-agente al fine di mettere in comunicazione, coordinazione, collaborazione e competizione i vari fruitori virtuali. Applicando su larga scala tale procedimento si ottiene una grande comunità virtuale sul territorio, in cui i fruitori possono liberamente scambiarsi informazioni.
Ogni utente che vuole utilizzare il servizio, invia al server una di richiesta, con le proprie preferenze e desideri relativi al servizio, creato un agente che lavora in modo attivo ed autonomo sulla piattaforma multi-agente installata sul server; quest'ultimo, dopo aver dialogato, collaborato e negoziato con gli altri agenti presenti sulla piattaforma, in base ai criteri forniti ricerca un passaggio oppure un passeggero ed infine compone la richiesta che verrà inoltrata al proprio utente.
L'estensione territoriale di tale comunità è abbastanza vasta in quanto non comprende solo persone di un piccolo sobborgo, ma tutti gli utenti di Trento e zone limitrofe. Per quanto tali fruitori siano tutti accomunati tra loro dalla volontà di far parte di un programma di rideshare, è innegabile che tra loro sussistano pur sempre notevoli differenze legate al preciso punto della città in cui il soggetto radica i propri centri di interesse: un lavoratore di Mattarello che deve recarsi a

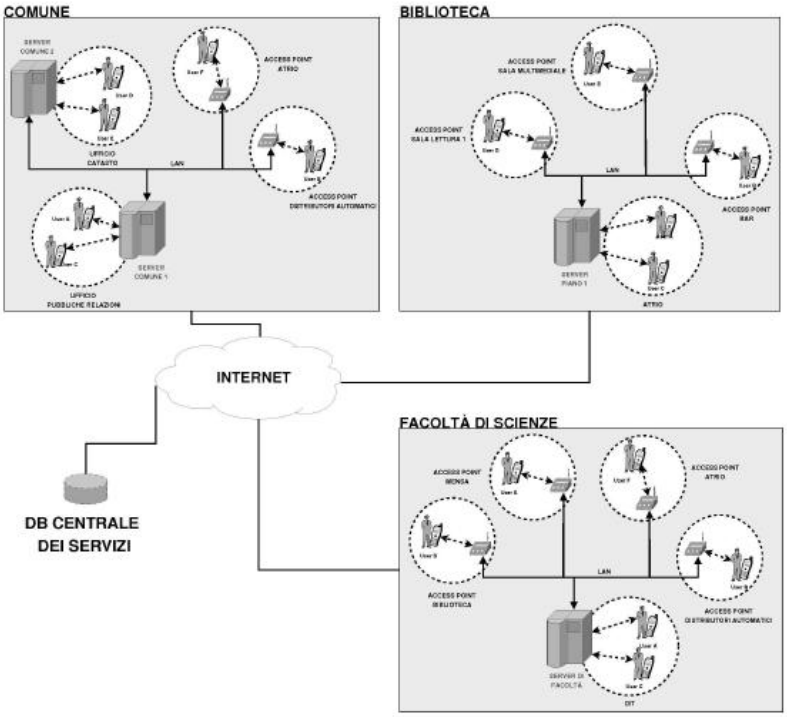


Figura 1.3: Possibile scenario di Andiamo

Trento centro avrà sicuramente esigenze differenti rispetto a quelle di uno studente di Trento che deve recarsi a Povo per seguire le lezioni universitarie. Le richieste di tali utenti difficilmente saranno tra loro compatibili, questo poichè essi appartengono a due sotto-comunità mobili differenti.
In ragione di tale rilievo si comprende l'importanza che all'interno del sistema Andiamo potrebbero rivestire le comunità virtuali mobili. Allo stato attuale infatti, nel momento in cui il lavoratore si connette al servizio per visualizzare i passaggi disponibili, riceve anche l'offerta dello studente, che per ragioni pratiche non può essergli di interesse. Al contrario se si suddividesse il territorio in sotto-comunità si eviterebbe una siffatta conseguenza, nella misura in cui il lavoratore riceverebbe sul proprio cellulare solo le offerte relative alla propria zona.
Lo scenario mostrato in Figura 1.3 è composto da tante piccole comunità virtuali eterogenee dal punto di vista degli utenti che ne fanno parte, ma omogenee per quanto riguarda l'utilizzo del sistema. Vi è un unico database centrale per memorizzare le informazioni relative al sistema e agli utenti ed attorno molteplici realtà collegate tra loro attraverso Internet; come accennato in precedenza i punti di accesso al sistema possono essere o un access-point Bluetooth che fa da tramite tra utente e server oppure direttamente un server, che accetta le richieste degli utenti, le elabora ed infine le inoltra al dispositivo targhet. Due sono le tipologie di messaggio che possono giungere:

- offerta passaggio: viene interrogato il server più vicino ed in base ai para-

metri offerti consulta gli altri agenti e infine spedisce il risultato al mittente dopo averlo memorizzato nel database centrale;

- ricerca passaggio: analogamente nel caso in cui una persona cercasse un passaggio in una determinata zona, è necessario accedere nuovamente al database per reperire i dati, mandarli al server in questione ed in ultimo spedirli al mittente.

Come si può notare in entrambi i casi indipendentemente dal tipo di richiesta, il meccanismo di reperimento dati risulta essere una procedura piuttosto lunga. Si pensi al caso di un generico utente che sia interessato alla ricerca di un passaggio e che quindi utilizzando il cellulare, si connette al server, invia le coordinate del tragitto ed attende che il pc riunisca tutte le richieste di ricerca fino ad ora attive nel database centrale e le rispedisca al mittente; queste operazioni richiedono tempi di collegamento abbastanza lunghi e poichè il cellulare è l'oggetto maggiormente coinvolto accusa un veloce abbassamento del livello della carica della batteria.L'aspetto negativo di questa procedura è legato ai tempi di attesa nel caso in cui il sistema fosse composto da molte realtà distribuite su un vasto territorio;infatti il notevole traffico di informazioni-richieste potrebbe creare lunghi ed "inutili " tempi di attese ed il fruitore riceverebbe dati non richiesti e superflui. Una soluzione a questo problema potrebbe essere svolgere il maggior numero di azioni sul posto; ovviamente tali considerazioni hanno solo una valenza teorica, in quanto Andiamo non è ancora attivo in Trentino A.A. quindi non si possono ancora osservare le prestazioni del sistema progettato. Alla luce dei rilievi appena svolti, ho scelto di estendere Andiamo, per cercare di risolvere il problema in ultimo evidenziato, concentrando la mia attenzione sul tema della localizzazione dell'utente al fine di portare miglioramenti significativi nell'efficienza di gestione dell'intero sistema e della qualità del servizio offerto. In particolare si è tentato di perseguire i seguenti obbiettivi:

- diminuzione del tempo di attesa dell'utente;
- evitare lunghi tempi di connessione da parte del dispositivo mobile nel caso in cui si fosse interessati solo a cercare un passaggio, senza dover aspettare che il sistema fornisca integralmente tutte le informazioni simili;
- diminuzione del lavoro del sistema e conseguente miglioramento delle prestazioni.

Capitolo 2

Estensione di Andiamo

In questo capitolo verrà illustrata l'architettura estesa di ANDIAMO: si presenteranno le estensioni e le modifiche apportate al sistema in questione, analizzando nuovamente i requisiti e le varie interazioni che avvengono tra componenti in grado di comunicare con utenti mobili tramite la tecnologia Bluetooth e di scambiare informazioni che rispettino gli interessi e le preferenze dei differenti fruitori.

2.1 Requisiti del sistema

2.1.1 Requisiti funzionali

Si analizzano ora le caratteristiche che dovrà soddisfare il sistema:

- Il sistema deve essere semplice ed accessibile a tutti, soprattutto per favorire coloro che non hanno tanta dimestichezza con i dispositivi mobili riguardo alla configurazione e attivazione dei servizi.
- Il sistema deve fornire l'accesso al servizio di rideshare quando il dispositivo mobile e l'access point Bluetooth sono co-localizzati; ovvero un generico utente deve poter essere individuato all'interno del sistema al fine di poter offrire un servizio che risulta molto più efficente ed utile rispetto alla sua posizione relativa.
- Per accedere al servizio,come mostrato in Figura 2.1 bisogna collegarsi al sito internet dove l'utente dovrà registrarsi fornendo i propri dati personali e quelli del dispositivo mobile che utilizzerà; se tutto viene eseguito correttamente dal sito è possibile scaricare il software da installare sul proprio dispositivo mobile necessario per poter usufruire del servizio;
- Il sistema deve garantire l'accesso ai servizi richiesti quando il dispositivo mobile si trova nelle immediate vicinanze del server come mostrato in Figura 2.2. Ovviamente tale distanza dipende molto dalle caratteristiche del telefono cellulare o del computer palmare dell'utente: lo scambio di

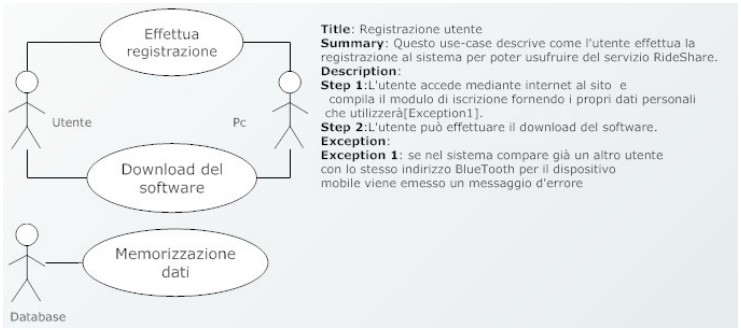


Figura 2.1: Registrazione nuovo utente

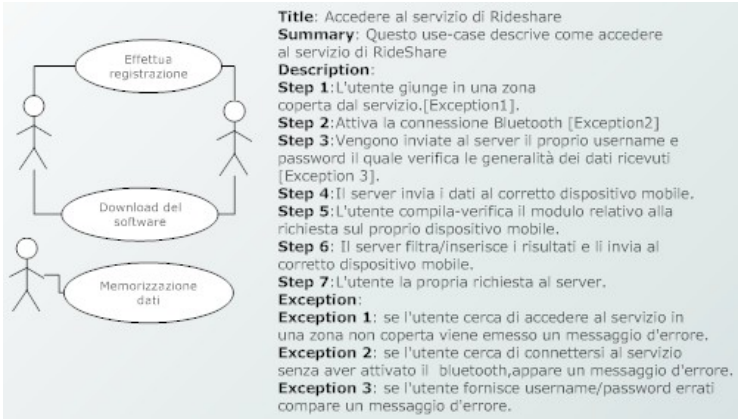


Figura 2.2: Richiesta di accesso al Rideshare

informazioni fra i dispositivi è automatico e istantaneo. Il sistema deve continuamente ricercare le preferenze dell'utente per poi inoltrarle sul dispositivo interessato.

- Il sistema dovrà permettere agli utenti di esprimere i loro interessi nella formulazione delle richieste di servizio. Tramite l'applicazione mobile l'utente compilerà le richieste di servizio rideshare fornendo diverse informazioni a seconda che egli stia cercando oppure offrendo un passaggio:
  - se dispone di un'autovettura dovrà specificare il luogo di partenza, di arrivo e l'orario.
  - se al contrario ricerca un passaggio, nel momento in cui effettua il login nel sistema riceve sul proprio dispositivo mobile i dati dei vari viaggi; trovato quello che maggiormente lo soddisfa lo prenota e invia la richiesta al server.
- l'utente avrà quindi la possibilità di accedere al database contattando via Bluetooth il server che si trova nelle immediate vicinanze e quindi recuperare le informazioni presenti sugli altri server; tutto questo avverrà ancora una volta in maniera completamente automatica.

2.1.2 Requisiti non funzionali

- Il sistema deve essere semplice da utilizzare per qualsiasi tipo di utente e non dovrà richiedere approfondite conoscenze riguardo l'utilizzo dei dispositivi mobili o computer.
- Il sistema deve essere utilizzabile su ogni device che abbia un sistema operativo Symbian e una connessione Bluetooth disponibile.

2.2 Architettura

Dopo una descrizione generale del vecchio sistema è giunta l'ora di analizzare l'architettura del nuovo: si illustreranno quindi le varie sotto-componenti entrando nel dettaglio delle loro interazioni e interconnessioni. Ciò che si vuole realizzare è un sistema semplice che permetta di usufruire del servizio di Rideshare nei luoghi in cui si può effettuare una connessione Bluetooth tramite un dispositivo mobile. In altre parole si intende creare un sistema di comunicazione e prenotazione passaggi tra utenti che si trovano entro un'area coperta dal servizio e sia in grado di offrire loro una prestazione che più si avvicina alla località del fruitore.
Il principale problema connesso all'utilizzo di dispositivi Bluetooth consiste nella ristrettezza del loro raggio di azione (che appunto consente la comunicazione solo se i dispositivi sono nella vicinanza dell'access point), problema questo astrattamente risolvibile attraverso l'impiego di access-point collegati via Ethernet.
Ogni utente interfacciandosi al sistema con il proprio dispositivo invia al server la richiesta con le preferenze relative al servizio ed in seguito viene eseguita un'interrogazione al server centrale al fine di scoprire se esiste un passaggio o un passeggero per le coordinate inserite: il responso viene poi inoltrato al dispositivo mobile del richiedente.
A differenza del precedente sistema Andiamo, si è scelto di sviluppare un'applicazione che non utilizzi piattaforme multi-agente, quindi è venuto meno la necessità di inserire personal agents per la contrattazione dell'orario, punto di incontro e arrivo del passaggio.

2.2.1 Componenti del sistema

Il sistema include quattro componenti principali come in Figura 2.3: i dispositivi mobili, un computer collegato a internet, access-point Bluetooth ed infine il server contente il database.

Si analizza ora il ruolo specifico di ogni singolo componente all'interno della struttura generale indicandone le loro principali caratteristiche,l'architettura e le interazioni tra di essi.

- *Dispositivi mobili* vengono utilizzati per permettere all'utente di inviare e ricevere le proprie richieste.
- *Server* contente database centrale, accessibile via web viene utilizzato per reperire e memorizzare le informazioni degli utenti registrati e le loro preferenze per accedere al servizio.

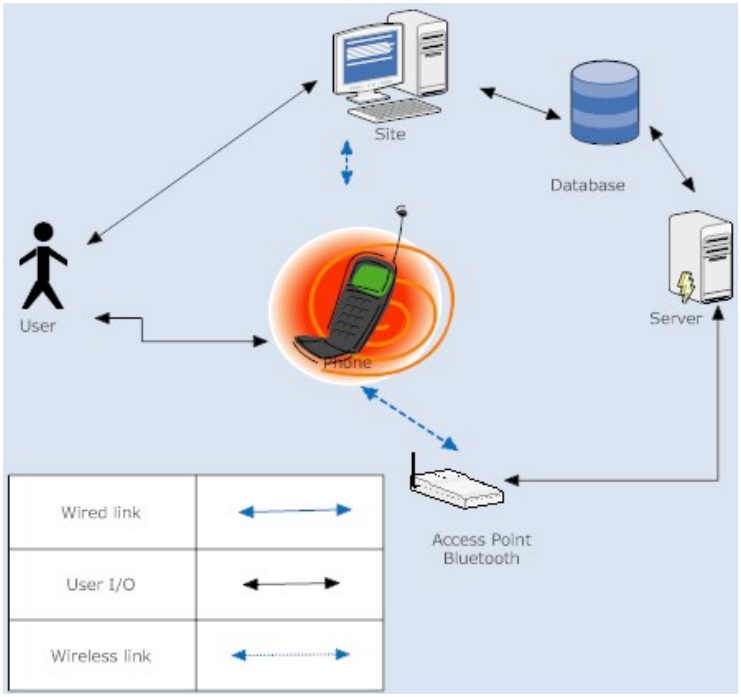


Figura 2.3: Interazione tra i vari componenti del sistema

- *Personal computer* permette all'utente di registrarsi al sistema attraverso l'accesso alla pagina web del servizio. Dopo aver compilato un modulo di registrazione dei dati,questi ultimi vengono memorizzati sul database centrale ed in seguito è possibile eseguire un download del programma che sarà installato sul proprio dispositivo mobile.
- *Access Point Bluetooth* permettono all'utente di accedere al servizio ed essere localizzati all'interno del sistema.

2.3 Implementazione

In questa sezione verranno analizzate le singole componenti al fine di comprendere e capire come interagiscono all' interno del sistema. Si mostreranno le scelte di implementazione adottate.

2.3.1 Server

Ogni server che supporta il servizio rideshare può comunicare con i circostanti dispositivi mobili (cellulari o PDAs) attraverso una connessione wireless Bluetooth, e attraverso una connessione Ethernet con gli altri apparecchi disposti nella zona. Inoltre è dotato di un database nel quale vengono selezionati e memorizzati i risultati che al momento necessario saranno inoltrati attraverso una comunicazione ai dispositivi mobili, registrati al servizio.

Ip	Città	Codice Città	Indirizzo Bluetooth	Nome Posizione	Codice Posizione
192.168.0.90	Trento	1	00119f73dc85	Piazza Fiera	4/1
192.168.2.122	Rovereto	2	00119b73cd21	Museo Mart	3/1
⋮	⋮	⋮	⋮	⋮	⋮

Tabella 2.1: Tabella che raccoglie i dati relativi agli access-point Bluetooth

Il database è composto da tre tabelle indicanti le peculiarità dei server/access point Bluetooth,gli utenti registrati al sistema ed infine tutte le coordinate riguardanti i viaggi fino ad ora inseriti.
E'giunto ora il momento di scendere nei dettagli delle loro principali caratteristiche.

- Server
  I dati rigurdanti il server possono essere riassunti nella Tabella 2.1:

  Come mostrato in Tabella 2.1 viene memorizzato per ogni access-point Bluetooth o personal computer il suo indirizzo ip,quello di Bluetooth,il nome della città in cui risiede,il codice associato alla locazione di appartenenza,la posizione all'interno del luogo e il nome del posto in cui è collocato fornendo inoltre dei possibili valori da inserire.
  L'ultima colonna ha una notevole rilevanza:di fatto permette di localizzare l'utente e di offrirgli un servizio che tenga conto della sua posizione fisica relativa all'interno del sistema. Per semplicità si assume che i punti di incontro tra le varie persone siano i medesimi in cui risiede un punto di accesso alla rete tramite Bluetooth;questi vengono memorizzati nel database in una maniera che consenta di posizionarli geograficamente ed enumerarli in maniera sequenziale; andando nel concreto si è diviso il territorio in punti cardinali assegnando

  - Al Nord → 1;
  - Al Sud → 2;
  - Ad Est → 3;
  - Ad Ovest → 4;
  - Al Centro →5;
  - Alla Periferia → 6;

  In seguito per ogni fascia si sono rintracciati i vari punti di incontro ed elencati in maniera sequenziale come ad esempio:

  - Ferrovie dello Stato : P.zza Dante 5/1
  - Parcheggio Economia : via Zanella 5/2
  - P.zza Santa Maria Maggiore 5/3
  - Piazzale S. Severino 5/4
  - P.zza Fiera : largo Luigi Pirandello 5/5

  Da sottolineare ancora una volta il significato della numerazione: ad esempio "5/3" viene valutato nella seguente maniera: il "3" è l'identificatore di

```
public double dammiDistanza(Punto p)
{
int ascissa=(x-p.x)*(x-p.x);
//quadrato della differenza della x dei due punti
int ordinata=(y-p.y)*(y-p.y);
//quadrato della differenza della y dei due punti
return Math.sqrt(ascissa+ordinata);
}
```

Figura 2.4: Codice

P.zza Santa Maria Maggiore e il "5" della zona di appartenenza; tutto ciò è stato fatto per permettere la localizzazione all'interno del sistema; ovvero, un generico utente che ricerca un passaggio, ottiene le informazioni relative ai punti di incontro, ordinati per distanza. Questa caratteristica è stata ottenuta attraverso le seguenti righe di codice: Come mostrato in Figura 2.4 quando l'utente si collega al servizio tramite dispositivo mobile, il server riceve anche indicazione della sua posizione in modo da poter calcolare al meglio i punti di incontro più vicini.
Ad esempio supponiamo che l'utente A si trovi presso

Parco Gocciadoro : via Gocciadoro 2/9

e cerchi un passaggio che abbia come punto di incontro il luogo più vicino. Supponiamo che le partenze inserite siano quelle mostrate precedentemente(Trento centro) e le seguenti:

- Ospedale S. Chiara : Largo Medaglie d'Oro 2/10
- Piazzale Ex-Zuffo : Uscita Trento Centro 3/1
- Parcheggio Ex Sit : via Canestrini 3/6
- Sede distaccata sociologia : via Grazioli 4/8

Applicando l'algoritmo visto in precedenza otteniamo il seguente ordinamento

Punto di Incontro	**Codice**	**Distanza**
Ospedale S. Chiara - Largo Medaglie d'Oro	2/10	1
Sede distaccata sociologia - via Grazioli	4/8	2.2
Parcheggio Ex Sit - via Canestrini	3/6	3.2
P.zza Fiera - largo Luigi Pirandello	5/5	5
Piazzale Ex-Zuffo - Uscita Trento Centro	3/1	8.1

Come mostrato in Tabella l'utente A in base alle sue preferenze cercherà-sceglierà il passaggio che ha come partenza l'Ospedale S.Chiara in quanto risulta essere il più vicino rispetto al luogo in cui si trova.

- User

Gli utenti registrati al sistema devono inserire i seguenti dati:

nome	cognome	userName	password	indirizzoMail
nomeAuto	statoGuida	sicurezza	statoAutomobile	pulizia

Gli ultimi quattro dati appartenenti alla seconda riga vengono assegnati in un secondo momento quando l'utente in questione offrirà un passaggio e verrà giudicato dal passeggero secondo le voci indicate. Ho scelto di offrire un meccanismo di feedback solo per utenti che offrono passaggi, in quanto il principale obbiettivo del lavoro è quello di offrire un servizio di condivisione posti auto.

- Trip
  Come mostrato in tabella,vengono memorizzati le coordinate principali di un trasporto.

tipo	ideCity	locPart	partenza	giorno
arrivo	userName	ideArrivo	locArrivo	arrivo
posti	cellDriver	cognomeDriver	userNamePrenot	orario

L'utente una volta identificato al sistema, riceve sul proprio device i passaggi disponibili in quel momento ottenuti attraverso una semplice query sulla tabella. In seguito può effettuare la propria richiesta di servizio e in un secondo momento riceverà sul proprio cellulare la notifica dell'esito dell'operazione.

2.3.2 Comunicazione Bluetooth

Per comunicare tra loro due device devono stabilire una connessione. Se si conosce l'indirizzo Bluetooth dell'altro dispositivo e l'identificativo del servizio, è possibile inoltrare una richiesta di connessione. I dispositivi si connetteranno se la loro distanza non è superiore alla massima consentita dal modulo installato e se i parametri di sicurezza lo consentono; questo implica chiaramente che il dispositivo a cui ci si connette deve possedere il servizio richiesto. Le operazioni da svolgere sono le seguenti: ricerca di dispositivi(inquiry),gestione dei servizi offerti e infine connessione con i dispositivi.

Inquiry()

La prima cosa che deve fare un'applicazione di rete Bluetooth è ricercare gli altri dispositivi abilitati presenti nel proprio range di copertura dell'antenna. Per dispositivi abilitati si intendono tutti quei dispositivi bluetooth che sono in modalità discoverable ossia in inquiry scan. Il dispositivo che inizia questa procedura spedisce in modalità broadcast un messaggio di inquiry e attende che i dispositivi presenti nella zona rispondano. Per stabilire quando un dispositivo deve rispondere sono state definite tre modalità di discoverable (general, limited e not discoverable) e due diversi tipi di inquiry (general e limited). Quando un dispositivo esegue una general inquiry tutti i dispositivi in modalità general e limited discoverable devono rispondere al messaggio. Nel caso in cui invece si

esegua una limited inquiry solo i dispositivi in modalità limited discoverable risponderanno. I dispositivi che sono in modalità not discoverable non rispondono in nessun caso a messaggi di inquiry ed è come se fossero invisibili agli altri.

Per permettere di settare la modalità di discoverable le API Java JSR- 82 [6] definiscono il metodo setDiscoverable() della classe LocalDevice. L'argomento passato a setDiscoverable() rappresenta la modalità di discoverable che il dispositivo dovrà usare per rispondere a messaggi di tipo inquiry. Per eseguire la ricerca vera e propria dei dispositivi si utilizzano i metodi della classe DiscoveryAgent che, come vedremo, si occupa anche della ricerca dei servizi. A tale scopo sono forniti due metodi: startInquiry() e retrieveDevices(). Come si intuisce facilmente il metodo startInquiry() dà inizio al procedimento di ricerca e prende come argomento due parametri che rappresentano il tipo di inquiry da eseguire (general o limited) ed un DiscoveryListener. L'applicazione deve infatti creare una classe che implementi l'interfaccia DiscoveryListener e ne implementi i metodi, che nel caso di inquiry sono deviceDiscovered() e inquiryCompleted(). Il metodo startInquiry() è una chiamata non bloccante e questo significa che ritorna prima del termine della procedura. È per questo motivo che si rende necessario il DiscoveryListener: al termine dell'inquiry infatti l'implementazione delle JABWT (Java APIs for Bluetooth Wireless Technology) prevede che venga richiamato il metodo inquiryCompleted() della classe DiscoveryListener per mettere a conoscenza l'applicazione della conclusione della ricerca. Il metodo deviceDiscovered() viene invece richiamato automaticamente ogni volta che viene scoperto un nuovo dispositivo e si riceve la sua risposta. Secondo questo metodo viene passato come argomento un'istanza della classe RemoteDevice che appunto rappresenta il dispositivo remoto appena conosciuto e permette attraverso una serie di metodi di conoscere alcune informazioni come l'indirizzo Bluetooth o il nome alfanumerico che lo identifica (user-friendly name). L'altro metodo messo a disposizione dalla classe DiscoveryAgent per l'inquiry è il retrieveDevices() che in realtà non esegue un'inquiry vera e propria ma si limita a ritornare quei dispositivi che erano già stati scoperti da inquiry precedenti e quindi non fornisce nessun tipo di garanzia sulla loro reale presenza. Con questo metodo è possibile anche inserire nella lista dei device ritornati, una serie di dispositivi di default che possono tornare utili nel caso in cui la tipologia della rete sia piuttosto stabile e non soggetta a grossi cambiamenti.

Service Discovery()

Una volta che si conoscono i dispositivi presenti nell'area di operabilità, il passo successivo è quello di determinare quali servizi questi mettono a disposizione; per fare questo si utilizza il Service Discovery Protocol definito dalle specifiche Bluetooth. Un'applicazione che vuole fornire dei servizi, e che per questo funge da server, deve permettere agli altri dispositivi di trovare questi servizi: li pubblica così tra i suoi service records i quali, attraverso una serie di attributi, li descrivono. Questi attributi specificano il nome del servizio, il modo con il quale connettersi ad esso, una breve descrizione del suo funzionamento ed altre informazioni utili al suo corretto utilizzo.

La procedura con la quale un server rende visibili i propri service records ai potenziali client prende il nome di service registration. Questa inizia con la chiamata a Connector.open() che restituisce uno StreamConnectionNotifier e crea un apposito service record per il nuovo servizio, con una serie di attributi

di default già settati in base ad una stringa che viene passata come argomento al metodo Connetor.open() la quale verrà descritta in dettaglio nel paragrafo successivo. Per ora basta sapere che in questa stringa è contenuto anche l'identificatore globalmente univoco del servizi (UUID Universally Unique Identifier). Una volta creato il service record è possibile aggiungerci dei propri attributi con il metodo setAttributeValue() della classe ServiceRecord che prende come argomenti un identificatore e il valore dell'attributo sotto forma di istanza della classe DataElement. Arrivati a questo punto il servizio non è ancora visibile agli altri dispositivi; per renderlo visibile occorre andare a modificare il contenuto del SDDB usando il metodo acceptAndOpen() sull'istanza della classe StreamConnectionNotifier ritornata da Connector.open().
Ora il servizio è correttamente registrato e il server è in attesa di client che si connettano. Vediamo ora quali sono le operazioni, definite dal Service Discovery Application Profile, che devono compiere i client per trovare e connettersi ai servizi. Supponiamo che sia già stata eseguita la procedura di inquiry e che abbia restituito come risultato alcuni dispositivi. Si vuole ora verificare se qualcuno di questi dispone di un particolare servizio al quale si è interessati e del quale già si conosce il suo UUID. Per fare questo si deve richiamare, su ognuno dei dispositivi trovati, il metodo searchServices() della classe DiscoveryAgent che prende come argomento una lista degli UUID che si stanno cercando e una lista dei service attributes che devono essere ritornati per ogni service record trovato e identificato con uno degli UUID ricercati. Come nel caso della ricerca dei dispositivi anche in questo caso occorre implementare l'interfaccia DiscoveryListener (passando il Listener alla funzione searchServices()) e definirne i metodi servicesDiscovered() e serviceSearchCompleted().
La chiamata a searchServices() come startInquiry() non è bloccante e l'implementazione delle JSR-82 si preoccupa di richiamare il metodo servicesDiscovered() ogni volta che si ricevono dei service records di risposta. Generalmente quando riceve la prima risposta, il client sospende la ricerca invocando il metodo cancelServiceSearch(). Nulla impedisce comunque di portare a termine la ricerca e anzi può essere utile farlo perchè, nel caso in cui il servizio non sia più raggiungibile o funzionante, si potrebbero già avere delle altre possibili fonti sulle quali provare. In ogni caso al termine della ricerca viene automaticamente richiamato il metodo serviceSearchCompleted() che riporta lo stato d'uscita dell'operazione (ricerca completata correttamente, ricerca terminata dall'utente, errori ...). Se la ricerca è andata a buon fine il cliente è ora in grado, con le informazioni contenute nel/i service record/s, di creare delle connessioni con altri dispositivi che mettono a disposizione il servizio ricercato.

Codice

```
public class ServerCommunicationBluetooth implements Runnable{
private LocalDevice localDevice;

private Thread accepterThread;
```

```
/** Identificativo univoco del nostro service */
private static UUID MY_SERVICE_UUID = new UUID("074304105205694F04F04C84904E8470", fa
public void run() {
System.out.println("Nuovo thread lanciato\n");

try {

// crea/ottiene la referenza al LocalDevice
localDevice = LocalDevice.getLocalDevice();
// settiamo le impostazioni in modo da essere discoverable
localDevice.setDiscoverable(DiscoveryAgent.GIAC);
} catch (BluetoothStateException e) {
System.out.println("Errore nel settare la modalità discoverable\n");
}
//Metodo che gestisce le connessioni
private void processConnection(StreamConnection conn) {

//apro I/O stream;
Input in,output out;
in=open();
out=open();
//leggo dati inviati dal dispositivo
if (identificationString == "ERROR"){
closeConnection();
}
else{
// attendo la stringa che inizia con username e password

userName = getUserName(identificationString);
password = getPass(identificationString);
DatabaseConnection dbc = null;
// Verifico password/userName ricevuti accedendo al databas

if (checkUserPass(userName,password))) {
/Sto x inviare i viaggi memorizzati
SendTrip sendTrip = new SendTrip();
while (there's connection){
```

```
//leggo la stringa inviata
String receive= read();
if(isAnOffer){
inserisco dati db;
}
if(isASeeker){
decremento i posti;
}
if (configurationString == "ERROR"){
closeConnection();
}

}
}

else{//pw sbagliata
System.out.println("Password non corretta o utente non registrato");
}
}
}
}
private class SendTrip extends Thread {
String trip;
public void run(){
Database dbc;
while (conn != null && out != null){
//mi connetto al database ed estraggo i passaggi disponibili

if (dbc.verifyConnection() != null) {
trip = dbc.getAnswer(SELECT* FROM trip);

}
else System.out.println("Nessuna risposta");
}
// ho tutti i viaggi. Ora le invio al device mobile
if (!send(trip)){
closeConnection();
System.out.println("Problemi nel spedire dati");
```

```
return;
}
else
"Fallimento";
}
}
}

}
```

2.3.3 Lato Device Mobile

In questa sezione verrà trattata superficialmente l'applicazione relativa al device mobile poichè nel capitolo 3 saranno illustrate in maniera precisa tutte le sue caratteristiche e il funzionamento; per ora sottolineamo che dispositivi mobili vengono utilizzati come ponte fra i PC e i server,ovvero servono all'utente per inviare-ricevere le richieste. L'applicazione installata consta di semplici ed intuitivi menu per favorire la fetta di fruitori non tanto pratica di cellulari.

2.3.4 Lato Personal Computer

Attraverso il computer collegato a internet si può accedere alla pagina del servizio esteso-modificato di Andiamo; l'utente se sta effettuando il primo accesso deve visitare alcune pagine scritte in linguaggio php che servono per effettuare una registrazione; quest'ultima se compilata in maniera corretta permette di memorizzazione i dati nel database e di poter scaricare il programma che verrà installato sul dispositivo mobile.

2.4 Scenari d'utilizzo

Il sistema ANDIAMO (e la relativa estensione) può essere replicato ed applicato in diversi contesti. Per esempio lo si può inserire all'interno di un ateneo, in biblioteca, nei bar, oppure nelle stazioni ferroviarie, ovvero in qualsiasi ambiente in cui si voglia offrire agli utenti/clienti un particolare servizio; ampliando i luoghi di attuazione viene formato uno scenario più esteso ed eterogeneo. Essendo il sistema principalmente composto da device mobili, si deve sicuramente tener conto delle problematiche relative allo spostamento degli utenti da un luogo ad un altro e contemporaneamente del carattere al contempo locale e distribuito; infatti non è da dimenticare, sia la modalità limitata con la quale i cellulari ed i palmari possono comunicare attraverso l'uso del Bluetooth (ha un raggio di azione piuttosto ristretto), sia il carattere distribuito del sistema al fine di fornire all'utente più punti d'accesso a seconda dei suoi spostamenti. Un possibile scenario di utilizzo dell'applicazione potrebbe essere una struttura universitaria nella quale vi sono utenti che, per il loro ruolo, possono essere interessati a

svariati servizi inerenti i loro studi o i loro hobby. Un altro scenario potrebbe essere una stazione ferroviaria, che potrebbe mettere a disposizione dei passeggeri servizi di informazione sui treni o sulle città di destinazione, avvisi di ritardi e/o soppressioni, ma anche servizi di intrattenimento e svago. La località dove il sistema potrebbe riscuotere il maggior successo è inserirlo in una facoltà universitaria dove il servizio di rideshare potrebbe risultare molto utile dato che la comunità è contraddistinta da persone che quotidianamente si muovono per raggiungere il luogo di lavoro e che quindi potrebbero condividere il viaggio di andata o ritorno. In un ambiente come questo dove esiste già una rete wireless che copre l'intero edificio, si può pensare di predisporre un unico server centrale e costruire una rete tramite il Wi-Fi di access-point Bluetooth ad esso collegati con un unico database centrale.

Capitolo 3

Esempio pratico

Si fornirà ora un'applicazione concreta del sistema precedentemente descritto: passo dopo passo verranno discusse le scelte d'implementazione adottate riguardanti la registrazione on-line dell'utente nonché il corretto utilizzo del cellulare e dei suoi servizi.

3.0.1 Accedere al servizio

L'utente deve usufruire di una connessione internet per poter collegarsi al portale; dopo aver digitato l'indirizzo si apre la pagina come mostrato in Figura 3.1: Per prima cosa bisogna effettuare la registrazione e precisamente basta accedere al menù di sinistra e selezionare login; al passo successivo,come mostrato in Figura 3.2 se non è il primo accesso, si immettono login e password e si accede ai servizi offerti dalla pagina web oppure in caso contrario si passa al paragrafo successivo.

3.0.2 Registrazione on line

La registrazione al sistema è il primo passo che ogni utente deve compiere per poter usufruire del servizio. Essa viene eseguita collegandosi alla rete Internet ed utilizzando un browser qualsiasi. L'utente deve compilare i campi della pagina web inserendo,i propri dati personali e quelli che intende utilizzare per autenticarsi al sistema. Ogni qual volta dovesse accedere al servizio, via internet oppure dispositivo mobile, gli verranno richieste username e password. Le generalità richieste sono quelle mostrate in Figura 3.3.

Se la registrazione avviene con successo tutte le informazioni fornite dagli utenti vengono memorizzate nel database; dopo di che l'utente può scaricare il file contenente il software per il cellulare/PDA (file di formato jar).

3.0.3 Dispositivo Mobile

L'applicazione del dispositivo mobile è stata realizzata con JavaMe e utilizza le JSR-82 che, come riferito nell'Appendice sono le API Bluetooth per Java; gli unici requisiti per renderla operativa è che il device supporti la tecnologia Java

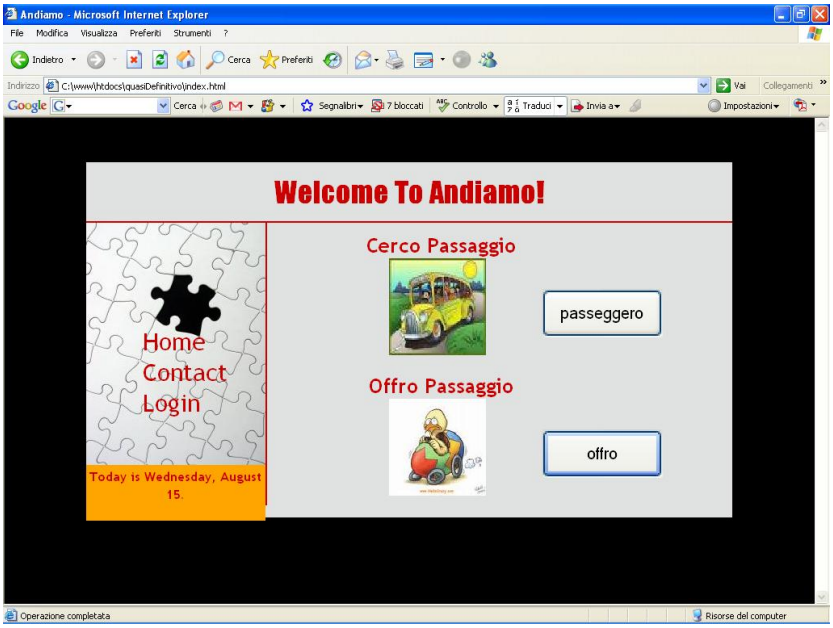


Figura 3.1: Pagina web iniziale del portale Andiamo

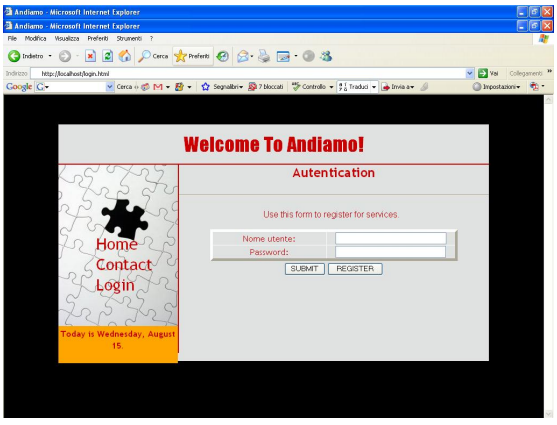
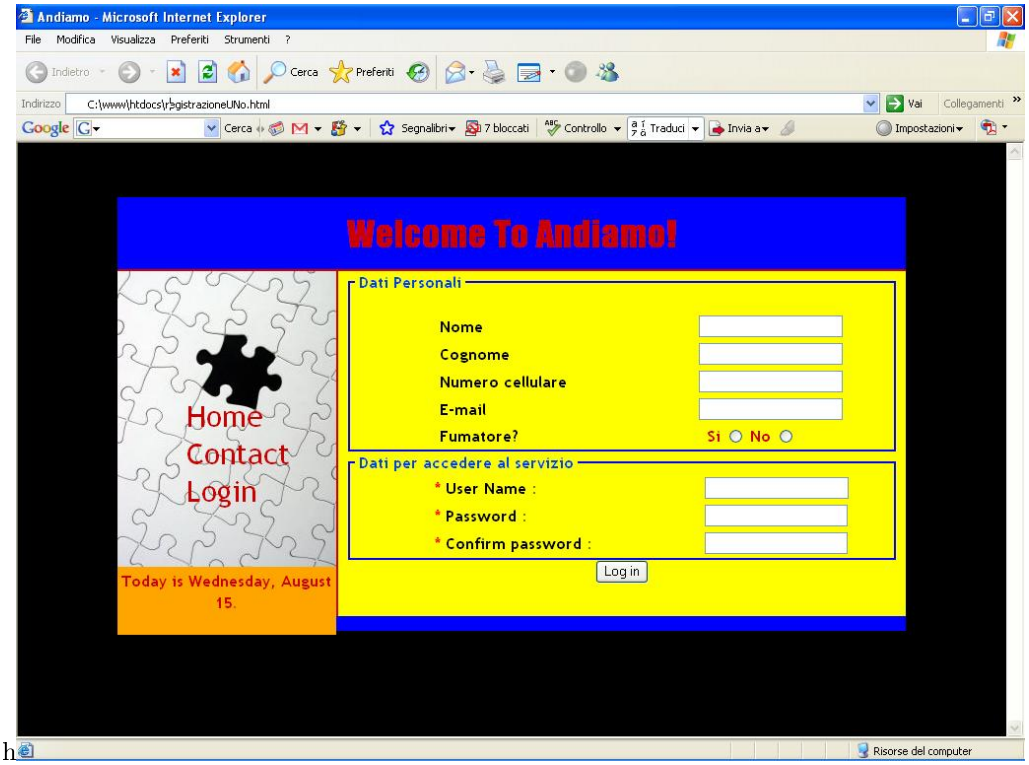


Figura 3.2: Login del portale Andiamo



widthwidth

Figura 3.3: Registrazione per accedere al servizio del portale di Andiamo

Figura 3.4: Accedere al servizio

e, ovviamente, la comunicazione Bluetooth.
Come accennato in precedenza il programma viene scaricato dal sito internet sul proprio Pc; è un file di tipo jar che viene trasferito via Bluetooth oppure anche via IR o USB e/o utilizzando programmi proprietari come Nokia PC suite sul proprio cellulare/PDA.
Per accedere al servizio l'utente deve svolgere le seguenti azioni:

1. attivare la comunicazione Bluetooth selezionando quindi la relativa opzione nel menù del dispositivo;
2. inizia a ricercare un access-point Bluetooth;
3. effettuare la richiesta;

Analizzando in dettaglio i passi da eseguire, l'applicazione avvia una ricerca continua di dispositivi Bluetooth abilitati nei dintorni e quando trova il servizio di Rideshare, stabilisce una connessione (passo 2); se l'utente offre un passaggio ed ha già compilato la richiesta, allora quest'ultima viene inoltrata via messaggio al server(passo 5), altrimenti il dispositivo riceve i trasporti fino ad ora disponibili (passo 3)avendo come punto di origine la data zona; lo scambio di informazioni tra dispositivi e server avviene sottoforma di stringhe XML. Le richieste sono poi elaborate dal server e i risultati rispediti all'utente (passi 6 - 4). Un esempio di richiesta è visibile in Figura 3.4

Il sequence diagram mostrato in Figura 3.5, descrive le interazioni fra il server e il dispositivo mobile dell'utente. Non appena viene individuato un access-point Bluetooth, il dispositivo mobile stabilisce con esso una connessione e invia i propri dati (username e password) per identificarsi univocamente al sistema; se l'autenticazione avviene con successo, il server risponde immediatamente inviando i passaggi disponibili fino a quel momento, altrimenti interrompe la comunicazione e invia un annuncio di errore; giunto a questo punto, se sono presenti richieste inoltrate ma non ancora inviate, vengono spedite al server che ha il compito di memorizzarle in apposite tabelle, altrimenti rimane in attesa. Intanto l'utente sceglie il servizio che più lo soddisfa: può compilare una nuova richiesta di passaggio oppure decidere di essere trasportato; in quest'ultimo caso,dopo aver ottenuto la lista di trasporti disponibili,sceglie il punto di incontro

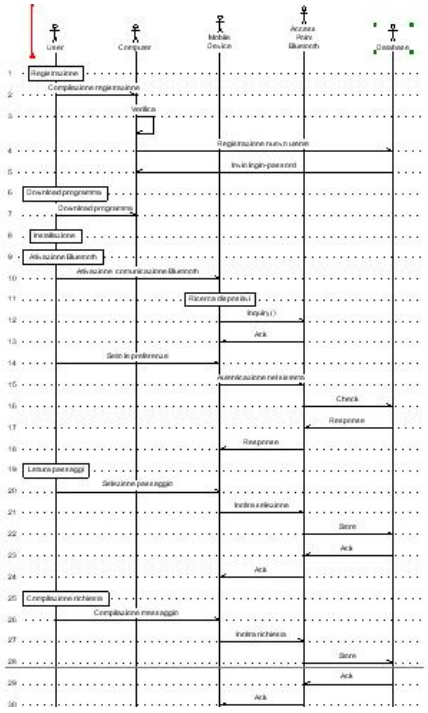


Figura 3.5: Azioni da svolgere per accedere al servizio

e orario preferito,interrompe la ricerca ed inoltra la richiesta al server, il quale provvederà a memorizzarla, oppure visita gli altri punti di incontro.

Installazione del file

Dopo aver effettuato la registrazione si può scaricare il software per accedere al servizio mediante dispositivi mobili e lo si installa sul cellulare/PDA; se ciò è andato a buon fine, l'utente è in grado di visualizzare le varie interfacce, ognuna destinata a svolgere compiti diversi. L'interfaccia principale dell'applicazione mobile è visibile in Figura 3.6;

Le richieste come mostrato in Figura 3.7 che posso essere di vario tipo:

- "Inviate" si accede alla lista di richieste di servizio già attivate presso il sistema ANDIAMO;
- "Archivio" si accede all'archivio di tutte le richieste di servizio precedentemente salvate nell'applicazione mobile;
- "Nuova richiesta" si avvia la configurazione di una nuova richiesta di servizio;

Quest'ultima opzione permette all'utente di accedere alla schermata successiva,come mostrato in Figura 3.8 e di scegliere il tipo di servizio desiderato:offrire o ricevere un passaggio.

Figura 3.6: Login del portale Andiamo

Figura 3.7: Schermata delle richieste

Figura 3.8: Schermata relativa al tipo di servizio desiderato

Figura 3.9: Schermata relativa all'offerta di viaggio

Figura 3.10: Schermata relativa al tipo di servizio desiderato

Offrire passaggio

L'utente interessato a questa sezione,deve fornire le principali coordinate dal passaggio che offre: il giorno, l'orario, il numero di posti disponibili ed infine selezionare da una lista la citta di partenza-arrivo e i rispettivi punti di incontro; quest'ultimo passaggio è molto importante poichè se il fruitore inserisse dei luoghi a suo piacimento, il server che gestisce le richieste sarebbe in difficoltà come l'utilizzatore. Compilato il modulo può salvare la richiesta o inviarla direttamente al server.

Cercare passaggio

Se si è interessati a questa opzione,una volta attivata la comunicazione Bluetooth ed inviati i propri dati di identificazione al sistema, il dispositivo riceve dal sever i passaggi,in "ordine di distanza" , giunti fino ad ora. L'utente a questo punto consulta la lista finchè non trova il punto di incontro, orario ed

Figura 3.11: Schermata relativa al tipo di servizio desiderato

Figura 3.12: Schermata relativa al tipo di servizio desiderato

Figura 3.13: Schermata relativa all'inserimento delle coordinate del viaggio desiderato

Figura 3.14: Schermata relativa al tipo di servizio desiderato

Figura 3.15: Schermata relativa al tipo di servizio desiderato

luogo di arrivo desiderati, sia per esempio il Parcheggio di Piazza Fiera come mostrato in Figura 3.14. L'interfaccia mostrata in Figura 3.15 mostra dunque maggiori dettagli relativi al trasporto ed infine il fruitore invia la sua richiesta di prenotazione in qualità di passeggero.

3.0.4 Settings

Il menù Settings dell'applicazione permette di configurare gli ulteriori parametri necessari per una richiesta di servizio. Viene mostrato l'intero menù,necessario a configurare/settare i parametri necessari per accedere al servizio; per prima cosa si richiede di inserire il proprio username e password fondamentali per essere identificati dal sistema ed infine il metodo di comunicazione di notifica messaggi che si vuole adottare; una volta compiute le precedenti azioni, le proprie preferenze vengono salvate come mostrato in Figura 3.16.

Figura 3.16: Schermata relativa alla memorizzazione delle preferenze dell'utente

3.1 Risultati ottenuti

Il sistema sviluppato è stato testato usando telefoni cellulari Nokia 6630 e PC/Server dotati con dispositivi Bluetooth generici di classe 2. L'applicazione ha dimostrato di funzionare correttamente, ma sono sorti problemi riguardo alla qualità del dispositivo da noi adoperato: infatti sul Nokia 6630 quando il server invia i dati, si sono riscontrati molti problemi nella corretta visualizzazione del messaggio a tutto schermo. Per quanto riguarda il Bluetooth, in ambienti aperti come in quelli chiusi, cioè con pareti ed ostacoli, il raggio d'azione è pari a quello fissato dallo standard, cioè circa 10 metri. Anche se la tecnologia wireless Bluetooth è molto diffusa per i suoi ridotti consumi energetici, (alcuni telefoni)il cellulare utilizzato ha mostrato un consumo eccessivo della batteria quando l'applicazione era in esecuzione e il modulo Bluetooth abilitato;per risolvere questo problema basterebbe permettere all'utente di configurare l'applicazione mobile in modo da avere un risparmio energetico più o meno efficiente; infatti la continua ricerca di dispositivi è la causa principale dell'eccessivo consumo della batteria. Purtroppo manca una vera e propria verifica del funzionamento dell'applicazione mobile su dispositivi non Nokia e con sistema operativo diverso da Symbian; si è notato comunque che per una compatibilità a largo spettro nel mondo delle piattaforme Java per dispositivi mobili (sono state rilasciate fino ad oggi già alcune versione di J2ME) è conveniente produrre l'applicazione mobile tenendo come riferimento le prime versioni di J2ME. Non è stato tuttavia possibile, a causa del ridotto numero di dispositivi mobili in nostro possesso, un testing più completo e di più ampio raggio che includesse un gran numero di comunicazioni.

3.2 Strumenti e tools

- Il sistema è stato sviluppato su sistema operativo Microsoft Windows XP Prof. SP2 poiché, programmando in linguaggio Java, non vi è la possibilità di usare la tecnologia Bluetooth al momento su altri sistemi operativi causa la mancanza di adeguate librerie gratuite. Inoltre è necessario ave-

re obbligatoriamente il Service Pack 2 perché implementa il supporto al Bluetooth e lo stack Bluetooth Microsoft, senza la necessità di driver proprietari (vedremo in seguito perchè questa cosa è importante), altrimenti tutte le versioni precedenti del sistema non lo supportavano, a meno di utilizzare software ad hoc.

- La comunicazione Bluetooth per le applicazioni Java è stata realizzata con l'utilizzo delle librerie Blue Cove, prodotto del progetto Blue Cove; esse sono un'implementazione Open Source delle APIs Bluetooth JSR-82 per Java di Sun ma prodotte per la piattaforma desktop (J2SE) e sviluppate inizialmente da Intel. Al momento solo un sottoinsieme delle APIs JSR-82 è implementato all'interno di Blue Cove e viene supportato soltanto lo stack Bluetooth Microsoft per Windows XP SP2; questo implica una serie di limitazioni e vincoli alla programmazione e utilizzo pesanti, in primo luogo a riguardo del sistema operativo che è possibile usare e poi relative ai dispositivi Bluetooth che si possono adoperare. Infatti lo stack Bluetooth Microsoft non riconosce tutti i dispositivi presenti sul mercato ma solo un piccolo sottoinsieme (esiste una lista on-line presso http://support.microsoft.com/kb/841803) e neanche così completo. La maggioranza dei dispositivi in commercio è legata a driver e software proprietari, come WIDCOMM o BlueSoleil per citare i due più conosciuti, e non si trovano informazioni a riguardo del supporto o meno dello stack Bluetooth Microsoft nelle loro specifiche. L'unica operazione che è possibile fare è provarli effettivamente su una macchina e sperare di non rimetterci troppo denaro. Sono state scelte le librerie Blue Cove per integrare il nostro sistema principalmente perchè sono gratuite e Open Source. La versione delle librerie è datata 14.05.2005. Per la traduzione dei comandi fra Java e C++ vengono usate le API Windows Socket e JNI (Java Native Interface).

- Per lo sviluppo di MIDP e applicazioni Personal Profile lato mobile Java è stato utilizzato il tool Nokia Carbide.j 1.0; esso può essere integrato nei più importanti tool di sviluppo Java moderni come NetBeans, JBuilder e Eclipse. Carbide.j permette di sviluppare applicazioni per ogni tipo di dispositivo che supporti Java ed esegua una J2ME; il tool è corredato da numerose funzionalità e da un emulatore per dispositivi mobili.

- Come dongle USB Bluetooth è stato utilizzato un dispositivo generico di classe 2, compatibile con le specifiche standard 1.2. Con questo dispositivo si è abilitato il computer alla comunicazione Bluetooth fino a distanze di circa 10 m [2].

- La piattaforma Java utilizzata per lo sviluppo della piattaforma multiagente lato server è la versione 1.5.0 06 della J2SE e il tool di sviluppo adottato è stato Eclipse SDK versione 3.2. Eclipse è un'applicazione Open Source, dietro cui esiste una vera e propria comunità di sviluppatori, destinata allo sviluppo di software in diversi linguaggi e su piattaforme diverse.

- I dispositivi mobili con cui abbiamo testato il sistema è stato il Nokia 6630 (Figura). Per dettagli e per la caratteristiche del cellulare è possibi-

le consultare i sito della Nokia agli indirizzi http://www.nokia.it oppure http://www.nokia.com. Figura 3.21: Nokia 6630

- Le basi di dati usate nel sistema Carpooling sono ottenute usando MySQL Database System versione Server 5.0. MySQL è il più popolare database SQL Open Source al mondo ed è sviluppato, distribuito e supportato da MySQL AB, compagnia commerciale fondata dagli sviluppatori di MySQL. Per lo sviluppo del database è stato usato MySQL Query Browser versione 1.1.20,prodotto Open Source anche questo di MySQL AB. Il database è stato realizzato con MySQL Database Server versione 5.0 e per lo sviluppo ho usato MySQL query browser.

Capitolo 4

Conclusioni

Il rideshare, ovvero la condivisione del mezzo di trasporto tra due o più persone, è una delle soluzioni che si possono adottare al fine di ridurre il quantitativo di autovetture in circolazione in ambito cittadino. Andiamo é stato creato al fine di agevolare tale condivisione, consentendo ai possibili utenti di disporre di un mezzo in grado di metterli in comunicazione con una molteplicità di persone, ovviando così al principale problema connesso al rideshare: la difficoltà di trovare tra la cerchia dei propri conoscenti persone interessate al servizio. Gli utenti di Andiamo formamo all'interno del territorio delle comunità virtuali, all'interno delle quali possono comunicare e scambiarsi informazioni pur non conoscendosi personalmente.
Sfruttando tale concetto,l'obiettivo del presente lavoro è stato quello di estendere il precedente progetto di Andiamo, introducendo e sviluppando l'idea della comunità virtuale mobile: il fruitore a seconda della sua posizione all'interno del sistema deve ricevere un servizio differente e mirato in base alla propria locazione e alle proprie preferenze.
Prima di poter essere applicato al mondo reale, il lavoro prodotto necessita ancora di ulteriori processi di verifica in merito alla scalabilità di sistema e alle sue prestazioni in ambienti con un numero di utenti piuttosto alto.
Il lavoro aveva come requisiti quello di essere facilmente accessibile ed utilizzare tecnologie ampiamente diffuse; questi due obiettivi sono stati raggiunti puntando su dispositivi mobili che supportano la tecnologia Bluetooth e programmabili in Java.
I vantaggi derivanti dall'utilizzo del sistema ANDIAMO per offrire il servizio di rideshare riguardano principalmente la facilità con cui l'utente può usufruire di questo servizio tramite il proprio dispositivo mobile e gli aspetti ambientali legati ad un servizio di questo tipo. Allo stato attuale il limite del sistema è costituito dall'assenza di un sistema in grado di gestire le informazioni geografiche utilizzate per abbinare gli utenti in base alla loro collocazione spaziale: infatti al momento sono ritenuti tra loro compatibili soltanto utenti che desiderano partire e arrivare nelle medesime località.
L'estensione del sistema Andiamo è stata testata usando telefoni cellulari Nokia 6630 e PC/Server dotati con dispositivi Bluetooth generici di classe 2 e access-point Bluetooth. Il sistema sviluppato ha dimostrato di funzionare correttamen-

te senza evidenziare anomalie rispetto al dispositivo da noi adoperato; tuttavia in fase di invio dei dati, si sono riscontrati problemi nella corretta visualizzazione del messaggio a tutto schermo. Durante il periodo di prova abbiamo notato che, anche se la tecnologia wireless Bluetooth è molto diffusa per i suoi ridotti consumi energetici su ogni dispositivo, quello da me utilizzato ha mostrato un consumo eccessivo della batteria nel momento in cui l'applicazione è in esecuzione e il modulo Bluetooth attivato; per risolvere tale problema bisognerebbe trovare il modo di consentire all'utente di decidere il periodo di connessione al servizio, oppure permettergli di eseguire periodicamente delle richieste al server e poi, appena giunta la risposta interrompere l'operazione. Purtroppo è mancata una vera e propria verifica del funzionamento dell'applicazione mobile su dispositivi diversi da Nokia e con sistema operativo diverso da Symbian; si è notato comunque che per una compatibilità a largo spettro nel mondo delle piattaforme Java per dispositivi mobili (sono state rilasciate fino ad oggi già alcune versioni di J2ME) è conveniente produrre l'applicazione mobile tenendo come riferimento le prime versioni di J2ME.
Non è stato tuttavia possibile, a causa del ridotto numero di dispositivi mobili in nostro possesso, un testing più completo e di più ampio raggio che includesse un gran numero di comunicazioni Bluetooth verso access-point Bluetooth e testare se il sistema si fosse comportato correttamente.

Appendice A

Uno Sguardo generale alla comunicazione Bluetooth

Il *Bluetooth* è una tecnologia di interconnessione wireless low-power(mWatt), in grado di far "comunicare" fino a 8 dispositivi elettronici come telefoni, stereo, notebook, computer e Pda, attraverso onde radio a basso raggio emesse da alcuni trasmettitori presenti all'interno di apparecchi. Il Bluetooth consente di gestire sia una trasmissione a pacchetto per i dati sia una modalità connection-oriented per la comunicazione vocale.

Cenni storici

Le prime ricerche sulla tecnologia Bluetooth vengono avviate nel 1994, quando la Ericsson Mobile Communication intraprende una serie di studi allo scopo di trovare delle alternative wireless per il collegamento tra telefoni e accessori (es. auricolari). Lo studio riguarda un collegamento radio che, non essendo direzionale, non necessita della cosiddetta line of sight, cioè della visibilità diretta, ed ha quindi degli evidenti vantaggi rispetto alle tecnologie infrarossi precedentemente utilizzate per collegare tra loro cellulari e dispositivi vari.
Nel febbraio 1998 nasce il SIG (Special Interest Group), un gruppo di compagnie leader del settore delle telecomunicazioni e dell'elettronica (di cui inizialmente facevano parte solo Ericsson, Nokia, Intel, IBM e Toshiba), le quali a partire da tale data iniziano a lavorare assieme per progettare dei dispositivi di comunicazione via radio a corto raggio caratterizzati da potenze di trasmissione ridotte e bassi costi di produzione. L'estensione e l'importanza del gruppo aumentano nei successivi anni, durante i quali si verifica un notevole incremento del numero di compagnie che richiedono di divenirne parte: si pensi che allo stato attuale il SIG comprende più di 2000, tra cui anche Microsoft, Lucent, 3Com e Motorola.
La tecnologia Bluetooth nasce appunto nel 1998 proprio su iniziativa del SIG. Lo strano nome con il quale venne battezzata questa tecnologia (letteralmente significa dente blu) ha un'origine antica: Bluetooth è infatti una traduzione approssimata di Blatand, il nome di un re Vichingo del X secolo che unificò Danimarca e Norvegia. Proprio come Blatand riuscì nell'intento ad unire due

popoli geograficamente vicini ma culturalmente distanti, così questa tecnologia si propone di far comunicare dispositivi diversi che si trovano per esempio nella stessa stanza.
Nel luglio 1999 viene pubblicata la versione 1.0 delle specifiche Bluetooth, che viene seguita nel dicembre 1999 dalla versione 1.0b. Nel febbraio 2001 esce la versione 1.1, mentre a novembre del 2003 viene rilasciata la versione 1.2.
La più grossa novità introdotta dalla Bluetooth Core Specification Version 1.2 è rappresentata dall'Adaptive Frequency Hopping (AFH), una tecnica pensata per ridurre al minimo le interferenze fra le varie tecnologie wireless che condividono la gamma di frequenze dei 2,4 GHz. L'AFH si preoccupa di analizzare dinamicamente lo spettro in cerca delle frequenze già occupate da altri dispositivi, come i telefoni cordless e i device Wi-Fi (802.11b/g), e utilizzare solo quelle disponibili: secondo gli ingegneri che l'hanno sviluppato questo può tradursi, in ambienti con molte interferenze, in un sensibile incremento delle prestazioni.
L'altra miglioria promessa dalle nuove specifiche 1.2 riguarda la velocità di instaurazione della connessione fra periferiche, un aspetto che nel 2002 il Bluetooth SIG si ripromise di affinare con il varo del progetto Five Minute Ready (pronto in cinque minuti). La nuova versione di Bluetooth include diversi nuovi metodi per la ricerca dei device che, attraverso l'adozione di algoritmi ottimizzati, riducono i tempi necessari affinché due o più dispositivi si vedano e siano pronti a scambiarsi dati: i progettisti hanno fatto in modo che i benefici apportati da queste nuove tecniche possano essere parzialmente sfruttati anche nel caso in cui solo il dispositivo ricevente sia aggiornato alla versione 1.2 dello standard.
Attualmente, è ancora in fase di definizione la versione 2.0; tuttavia un primo aggiornamento di questa tecnologia si è avuto all'inizio del 2005, con l'introduzione sul mercato dei primi dispositivi rispondenti alle specifiche Bluetooth 2.0 con EDR (Enhanced Data Rate).

Il successo del Bluetooth

Attualmente il Bluetooth viene utilizzato specialmente da dispositivi mobili quali telefoni cellulari, laptop e palmari, tuttavia comincia ad affermarsi anche per le comunicazioni tre stampanti, fotocamere, tastiere e dispositivi di puntamento.
Esistono essenzialmente due tipologie di adattatori Bluetooth per pc: le cosiddette penne usb (dongle) e le schede PCMcia. I laptop più recenti e in particolar modo di fascia più alta hanno un chip bluetooth integrato sulla motherboard e non è quindi necessario ricorrere all'acquisto di adattatori esterni. I fattori determinanti per il successo del Bluetooth a discapito di altre tecnologie già in commercio come ad esempio IR,Wap,Gprs,Wi-Fi sono i seguenti:

- il vantaggio di poter mettere in comunicazione dispositivi tra i quali siano presenti ostacoli, cosa che non è possibile nelle comunicazioni ad infrarossi, dimostrando di essere il candidato ideale per sostituire le inaffidabili interfacce IrDa (Infrared Data Association);
- la possibilità di minimizzare il consumo di energia usando potenze di trasmissione molto basse, che superano molto raramente i 2,5 mW;
- il suo carattere tipicamente localizzato, essendo il suo raggio di azione limitato;

Classe	Potenza output (max)	Distanza(max)
1	100 mW (20 dBm)	100 m
2	2.5 mW (4 dBm)	10 m
3	1 mW (0 dBm)	1 m

Tabella A.1: Classi di potenza

- costi di connessione nulli, a differenza del WAP per cui si deve pagare un prezzo che varia a seconda della compagnia telefonica e della quantità dati che vengono scaricati;

- la grande diffusione (a bassi costi) che il Bluetooth sta avendo in questi ultimissimi anni, a differenza del GPS che al contrario non è ancora integrato nei cellulari e lo è solo in pochissimi PDA;

Classi di potenza

Le specifiche Bluetooth,definiscono tre classi di potenza e le relative distanze che si possono raggiungere in trasmissione. Come mostrato dalla Tabella A.1, i dispositivi di Classe 1 hanno d'obbligo il requisito del controllo di potenza, il quale è opzionale nelle classi 2 e 3. In ogni caso, per un minimo consumo di potenza, è preferibile avere tale controllo. Il controllo di potenza agisce tramite un ricevitore che esamina un segnale di monitorazione della potenza, l'RSSI (Receiver Signal Strength Indication); in caso di segnale troppo debole viene mandato un segnale di controllo LMP incr pow (a livello Link Manager), che richiede di aumentare la potenza per avere un canale efficiente, mentre in caso di segnale più forte del necessario, si richiede una diminuzione della potenza tramite un segnale di controllo LMP decr pow req.

Informazioni generali

Nonostante le basse potenze di trasmissione, i dispositivi Bluetooth sono in grado di garantire un collegamento affidabile e quasi del tutto immune alle interferenze elettromagnetiche: le trasmissioni vengono effettuate in spread spectrum, una tecnica nata in ambito militare, durante la seconda guerra mondiale, che consente di trasmettere un segnale distribuendo la sua energia su una banda molto più ampia: dato che le interferenze colpiscono solo una parte minima dello spettro, quando si va a ricostruire il segnale originale da quello "spread" l'interferenza subisce il processo opposto, ovvero la sua energia viene distribuita su un intervallo molto ampio, diventando trascurabile. In particolare la tecnica utilizzata dal Bluetooth per fare lo Spreading è detta FHSS (Frequency Hopping Spread Spectrum) e consiste nel saltare di frequenza in frequenza all'interno di una banda molto ampia per 1600 volte al secondo seguendo una particolare sequenza generata da un algoritmo noto al trasmettitore e ricevitore. Questo sistema di comunicazione funziona secondo uno schema Time Division Multiplexing (TDM), dove l'unità base di operazione è uno slot di durata pari a 625

*mu*s; quando i dispositivi sono connessi, negli slot pari è abilitato a trasmettere il master, mentre negli slot dispari sono abilitati a rispondere gli slave, uno alla volta; in particolare può rispondere lo slave che aveva ricevuto dati dal master nello slot precedente. L'utilizzo della tecnica FHSS comporta che ogni time slot sia caratterizzato da una frequenza di trasmissione differente, secondo una sequenza di valori di frequenze decisi dal master seguendo un particolare algoritmo di calcolo del frequency hopping. Il bitrate massimo raggiungibile è pari a 1 Mbps. Al contrario di altre tecnologie come 802.11b, il bluetooth non permette di raggiungere velocità di trasmissione elevate e neppure consente di coprire aree con un raggio superiore ai 100 m, nemmeno a potenza elevata (ci si riferisce in tal caso ai dispositivi di classe 1): queste limitazioni sono più che legittime poiché questa tecnologia nasce per collegare piccoli dispositivi in modo facile ed affidabile.
Le trasmissioni vengono effettuate sulla banda dei 2,4 GHz e in condizioni tipiche coprono un'area con un'estensione di poco inferiore ai 15 m, si parla quindi di Personal Area Network(PAN). Attualmente tutte le versioni in commercio di Bluetooth sono retro compatibili con Bluetooth 1.1; la versione 1.0 non conobbe mai una adeguata diffusione, mentre la versione 1.2 al momento è quella più diffusa e introduce alcune migliorie rispetto alla 1.1 quali un Advenced Frequency Hopping (AFH) simile al FhSS; tale versione riduce al minimo le interferenze fra le varie tecnologie wireless che condividono la gamma di frequenze dei 2,4 GHz. L'AFH si preoccupa di analizzare dinamicamente lo spettro per cercare delle frequenze già occupate da altri dispositivi, come i telefoni cordless e i device Wi-Fi (802.11b/g), e utilizza solo quelle disponibili: secondo gli ingegneri che l'hanno sviluppato questo può trovare una larga applicazione in ambienti caratterizzati da interferenze facendo riscontrare un sensibile incremento delle prestazioni. La versione 2.0, introdotta recentemente porta la velocità massima teorica a 3 Mbps. Un collegamento Bluetooth può trasportare sia voce che dati; le trasmissioni voce usano link di tipo Sco (Synchronous Connection Oriented)mentre per quelle dati si usano quelle del tipo ACL (Asynchronous Connection Less).

Appendice B

JavaMe e Bluetooth

Per permettere l'implementazione del Bluetooth sul maggior numero di device possibili, agevolandone la diffusione, è stata progettata e rilasciata una specifica aperta e senza royalty dal Bluetooth SIG che ne definisce sia i livelli hardware, sia quelli software. Un comportamento ben diverso è invece stato adottato dai produttori dei moduli Bluetooth che forniscono delle apposite SDK per interfacciarsi con i propri moduli, vincolando in questo modo gli sviluppatori all'uso di API proprietarie per i loro rispettivi chip-sets. Una diretta conseguenza 'e che le applicazioni Bluetooth costruite utilizzando queste API non sono portabili su device e piattaforme diverse, nonostante la tecnologia sia basata su delle specifiche standardizzate. In realtà però queste specifiche sono di tipo funzionale nel senso che descrivono come i dispositivi Bluetooth si comportano e come interagiscono tra di loro attraverso i profiles, tralasciando il modo con cui questi devono essere programmati o come un'applicazione deve utilizzare le funzionalità di questi dispositivi ed i canali di comunicazione. Per risolvere questi problemi di compatibilità, che ostacolavano la creazione di applicazioni, si è resa necessaria la creazione di un set di API standardizzate che permettessero lo sviluppo di software Bluetooth indipendente dalla piattaforma. Quando si decise di creare questo set di API standardizzate per la tecnologia wireless Bluetooth, la scelta del linguaggio di programmazione da utilizzare ricadde su Java. Oltre ai benefici della portabilità, che permettono alle API Java di essere eseguite su hardware, sistemi operativi e classi di dispositivi diversi, Java permette inoltre un rapido sviluppo delle applicazioni, grazie al suo paradigma di programmazione orientato agli oggetti che ne garantisce un'astrazione ad alto livello, ed una comunità di sviluppatori già molto estesa. Il Java Community Process introdusse le prime specifiche di API standardizzate per il Bluetooth nell'anno 2000. Queste specifiche conosciute come JSR-82 (Java Specification Request 82) o JABWT (Java Api for Bluetooth Wireless Technology) definiscono le basi per lo sviluppo di applicazioni Bluetooth e sono state pensate e progettate per poter essere utilizzate anche da sistemi di tipo CLDC (Connected Limited Device Configuration) che hanno limitate capacità di calcolo, di memoria e batteria. Gli sviluppatori possono così creare delle applicazioni Bluetooth indipendenti dall'hardware utilizzato e che quindi possono essere implementate anche su device diversi purché questi supportino JSR-82. Per poter utilizza-

re le JABWT un dispositivo deve disporre di uno stack che sia qualificato per almeno il Generic Access Profile, il Service Discovery Application Profile ed il Serial Port Profile e che permetta l'accesso al Service Discovery Protocol ed ai suoi livelli RFCOMM e L2CAP. Molta attenzione è stata posta anche alla scalabilità, in modo da permettere a questo tipo di applicazioni di poter essere eseguite su qualsiasi tipo di piattaforma Java 2 (J2ME, J2SE, J2EE) dotata del Generic Connection Framework (GCF). Proprio per motivi di scalabilità queste nuove API non implementano tutti i livelli ed i profiles indicati dalle specifiche Bluetooth ma si limitano a ricoprirne solo una parte. Per esempio sono supportati:

- la comunicazione dati e non quella vocale;
- l'utilizzo dei protocolli;
- L2CAP (solo nella versione orientata alla connessione);
- RFCOMM;
- SDP;
- OBEX (OBject EXchange protocol)

I profiles supportati sono:

- Generic Access Profile (GAP);
- Server Discovery Access Profile (SDAP);
- Serial Port Profile (SPP);
- Generic Object Exchange Profile (GOEP)

Le funzionalità che mettono quindi a disposizione le JABWT sono:

- la ricerca di dispositivi e servizi;
- la registrazione dei servizi;
- la possibilità di stabilire connessioni di tipo RFCOMM, L2CAP e OBEX;
- la possibilità di eseguire queste operazioni in una modalità sicura.

Bibliografia

[1] Autovetture euro. `www.liberiamolaria.it/download/fap.pdf`.

[2] Bluetooth device. http://www.tecomproduct.com/bt3030.htm.

[3] Caratteristiche inquinamento autovetture. `www.aci.it/fileadmin/documenti/bassihome/TabellaEuro.pdf`.

[4] Fipa: Foundation for intelligent physical agents. http://www.fipa.org/.

[5] Jade: Java agent development framework website. http://jade.tilab.com/.

[6] Jsr-82: Java apis for bluetooth. http://www.jcp.org/en/jsr/detailsid=82.

[7] Mobility manager. `www.euromobility.org`.

[8] Portable cicero. http://giove.cnuce.cnr.it/Cicero.html.

[9] Protezione dell'ambiente. www.epa.gov.

[10] Trasporti e inquinamento atmosferico nella provincia di bolzano. http://www.provinz.bz.it/umweltagentur/2902/FAP/trasporti.htm.

[11] Web site promosso dalla provincia di trento. http://www.abertech.it/carpooling/default.asp.

[12] *Clean Air Act*, volume 2. Queen's Printer of Acts of Parliament, 1990. http://www.epa.gov/air/caa/caa.txt.

[13] Deroghe per mezzi nuovi e car pooling. *Adige*, page 23, 17 novembre 2006.

[14] S.I Ministero Ambientale. Decreto 27 marzo: Mobilità sostenibile nelle aree urbane. *Gazzetta ufficiale della Repubblica Italiana*, (179), 3 agosto 1998.

[15] Aliaksandr Birukou, Enrico Blanzieri, and Paolo. 4rd international joint conference on autonomous agents and multi-agent systems. In *An agent-based recommendation system for web*, pages 618–624. ACM Press, 2005. http://doi.acm.org/10.1145/1082473.1082567.

[16] UNCED:United Conference Environment and Developement. La conferenza internazionale su ambiente e sviluppo. In *Conferenza di Rio*, 14 giugno 1992.

[17] A. Rakotonirainy, Seng Wai Loke, and Arkady. *Towards Multi-Agent Support for Open Mobile Virtual.* aprile 2000. http://citeseer.ist.psu.edu/306410.html.

[18] Howard Rheingold. Smart mobs: the next social revolution. *Inf. Res*, 8(3), 2003. http://informationr.net/reviews/revs086.html.

[19] Francesco Sottini. *Progettazione e sviluppo di un sistema multi-agente per servizi di Carpooling accessibili da dispositivi mobili.* PhD thesis, Laurea triennale Informatica, 2005-2006.

[20] UNCED. Il protocollo di kyoto della convenzione sui cambiamenti climatici. In *Conferenza di Kyoto*, 1992.

[21] Wikipedia. Mobilita sostenibile. `it.wikipedia.org/wiki/Mobilità_Sostenibile`.