UNIVERSITA' DEGLI STUDI DI TRENTO
Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea triennale in Informatica

Elaborato finale

BlueOrganizer: Scheduler di impegni per cellulari e PDA

Relatore:
Ing. Paolo Giorgini

Laureando:
Thomas Ferrari

Anno Accademico 2007 - 2008

Indice generale

Introduzione

Lo sviluppo tecnologico ha accompagnato – in modo non lineare eppure costante – l'intera storia dell'umanità. La cultura, la cui principale espressione risiede nella creazione e nell'utilizzo di strumenti, ha determinato brusche accelerazioni e discontinuità anche nella moderna età industriale.

Le recenti conquiste tecnologiche specialmente nell'ambito delle comunicazioni e degli "universi virtuali" informatici sembrano svincolarsi dal mero asservimento alle esigenze economiche del complesso sociale spingendosi sull'inedito terreno della cura del tempo libero, delle relazioni interpersonali, della gestione dei ritmi e delle scadenze quotidiane.

Negli ultimi anni abbiamo assistito ad un grande sviluppo nel campo dei dispositivi mobili e del wirless. La comunicazione tra persone risulta essere in tal modo agevolata. Abbiamo, infatti, la possibilità di disporre di dispositivi mobili come PDA e cellulari di terza generazione. Gli utenti dispongono di terminali multimediali interattivi, capaci di collegarsi ad una rete globale che può trasportare voce, immagini, dati, permettendo l'accesso ad una quantità enorme di informazioni e servizi. Tali terminali hanno capacità di calcolo, di visualizzazione e acquisizione di immagini, avvicinandosi sempre più a veri e propri computer tascabili. Ciò che ha migliorato queste operazioni è l'avvento di sistemi di comunicazione come Wi-Fi e Bluetooth. In attesa di un maggior sviluppo di Wi-Fi su dispositivi mobili si sfruttano appieno le capacità di Bluetooth, che permette la comunicazione senza fili con basso consumo energetico. Blutooth è ormai diffuso sulla maggior parte dei dispositivi di media qualità.

La presente tesi si pone come obiettivo la creazione di un software agenda che permette l'organizzazione automatizzata degli impegni, sfruttando le potenzialità offerte dai summenzionati dispositivi mobili. Tale applicazione permette all'utente di impegnarsi con un'altra persona senza interagire direttamente con essa, abbreviando quindi i tempi. E' l'agenda stessa a verificare la possibilità di fissare l'incontro colloquiando con quella della persona selezionata. La comunicazione tra le due agende avviene in modo totalmente trasparente agli utenti, attraverso lo scambio di informazioni via Bluetooth o SMS.

La tesi si articola in tre capitoli. Il primo capitolo riguarda le motivazioni del progetto svolto, con cenni sulla tecnologia utilizzata.

Il secondo capitolo verte la progettazione dell'applicazione creata e la motivazione delle scelte strutturali.

Nel terzo capitolo, infine, viene presentata la parte implementativa del software e se ne descrive l'utilizzo.

Capitolo 1

Agende e tecnologie di supporto

Nella prima parte di questo capitolo verrà offerta una panoramica sulle agende attualmente esistenti. Nell'ultima fase è prevista una breve descrizione delle tecnologie utilizzate.

1.1 Agende cartacee e elettroniche

Il termine agenda deriva dal latino *ago -is egi actum -ere* [1] ed è tradotto in lingua italiana con il verbo “agire”. L'uomo, nel corso della propria giornata, compie azioni di varia natura e genere. Con il passare del tempo, alcuni soggetti hanno ideato dei promemoria. L'esigenza di maggiore coordinamento ed efficienza nel campo professionale e del lavoro in generale hanno sviluppato ed organizzato questi promemoria (bigliettini, appunti, ecc.), in un'unica fonte: l'agenda.

Oggi, per agenda, si intende un libro o un software che da la possibilità di annotare i propri impegni organizzati per ore, giorni, settimane, mesi.

Ne sono disponibili varie tipologie. Ogni tipo di agenda mira a conquistare una diversa fascia di mercato, che varia dall'età del consumatore ai gusti allo stile di vita e modo di utilizzo del prodotto stesso.

Inizialmente, le agende erano presenti nel solo formato cartaceo, ma, con l'avanzamento della tecnologia, sono state create agende elettroniche utilizzabili

su dispositivi dedicati o sul proprio personal computer. Da quando l'utilizzo del telefono cellulare è entrato a far parte della vita quotidiana sono state create apposite agende, diffuse anche con il termine inglese organizer, utilizzabili direttamente dal dispositivo.

Girando in città ed osservando le persone passeggiare è praticamente normale trovare solo alcune categorie di individui con borse, valigette, zaini…, probabilmente studenti, impiegati, rappresentanti, persone, che in questi spazi contengono il loro lavoro, lo studio e quasi certamente anche un'agenda, dove segnare gli impegni da affrontare giorno per giorno.

Chi possiede un'agenda cartacea si lamenta per il continuo disordine, appuntamenti presi, poi cancellati, poi sostituiti e risostituiti, fogliettini volanti, che escono ovunque e si perdono. Aumentando così il tempo perso a cercare di trovarli ed a cercare alcune volte di capire ciò che magari si è scritto di fretta ed è ormai illeggibile. Quindi scoppia il caos, se una persona non è molto ordinata, almeno un appuntamento rischia sempre di saltare!

La maggior parte delle persone, non possiede un'agenda per ragioni di ingombro. Tanti uomini non potrebbero sicuramente metterla nella tasca posteriore dei pantaloni e nemmeno in quella interna della giacca. Le donne, poi, dovrebbero usare solo borsette extra large visto che contengono tutto l'indispensabile per intervenire su ogni baffo del make- up. Inoltre, non sempre si ha una penna od una matita a portata di mano. Per non pensare, nel caso di lasciare l’agenda

momentaneamente incustodita, tutti vi possono accedere e leggere ogni vostro impegno, anche se intimo. Quindi addio alla privacy.

Con l'introduzione dell'agenda elettronica (inizio anni '90), alcuni problemi che si ponevano con la versione cartacea sono stati risolti. Gli appuntamenti risultano più ordinati, senza cancellature. Viene rispettata la privacy e si può portare ovunque con un gran risparmio di spazio. Da questo tipo di agenda vengono offerti altri servizi, quali rubrica, calcolatrice, convertitore di valuta ed orario. Nonostante queste buone qualità, con la diffusione di PDA e di telefoni cellulari, questi dispositivi possono essere considerati superati. Il principale motivo è che quasi tutta la popolazione è in possesso di un telefono cellulare, contenente i servizi offerti dall'agenda elettronica, vanificandone il suo acquisto.

Un servizio di agenda è fornito anche dal web. Il web offre, infatti, la possibilità di utilizzare delle agende condivise, dove un utente dà il permesso ai navigatori di visualizzare i propri impegni. La persona, che desidera incontrare il proprietario dell'agenda, può proporre un incontro seguendo il calendario online. Un esempio di questa tipologia d'agenda è Google Calendar[2].

Google Calendar è un servizio web gratuito, di cui attualmente è disponibile la versione beta. E' strettamente legato con il servizio di posta web Gmail. Google Calendar consente di tenere traccia di tutti gli eventi della propria vita, comparandoli con quelli di amici e parenti: impostare reminders, inviare inviti e

condividere i propri calendari con tutti, o decidere di mantenere i propri dati privati. L'applicazione include un servizio di notifica degli eventi tramite SMS, feeds di integrazione con iCal e altre applicazioni desktop calendars; offre inoltre anche funzioni di ricerca e sharing avanzate.

Google Calendar prevede la gestione di calendari multipli. Offre la possibilità di creare calendari personali, professionali, privati, pubblici, etc..., di vederli insieme o ciascuno separato dall'altro, facilitandone il riconoscimento con la scelta di diversi colori;

I calendari possono essere condivisi con altre persone, in base alla scelta di renderli pubblici o meno, e soprattutto ognuno è libero di scegliere a quali persone concedere l'accesso alla propria agenda. I calendari possono essere pubblicati come pagine web o via RSS, così che i lettori, per essere aggiornati, non devono necessariamente loggarsi nella piattaforma del calendario. Si possono importare eventi da altri programmi, come Yahoo Calendar o Microsoft Outlook.

Si deve ammettere che il servizio offerto è pressoché completo. Però presenta alcune limitazioni. Per poter sfruttare questo importante servizio, si necessita di un PC da tavolo o di un portatile, oltre che di una connessione internet. Strumenti non comunemente trasportabili. Spesso utilizzati quotidianamente per questioni di lavoro o studio. Si potrebbe anche recriminare che dal PDA si può comunque raggiungere il web, ma non mi sento di sostenere tali ipotesi poiché se il telefono cellulare ha pienamente soddisfatto la clientela, mentre non è ancora così per il PDA. Perciò l'utente, per poter aggiornare la propria agenda, deve essere seduto davanti ad un PC. Inoltre, nel caso della condivisione dell'agenda, la privacy non è

tutelata.

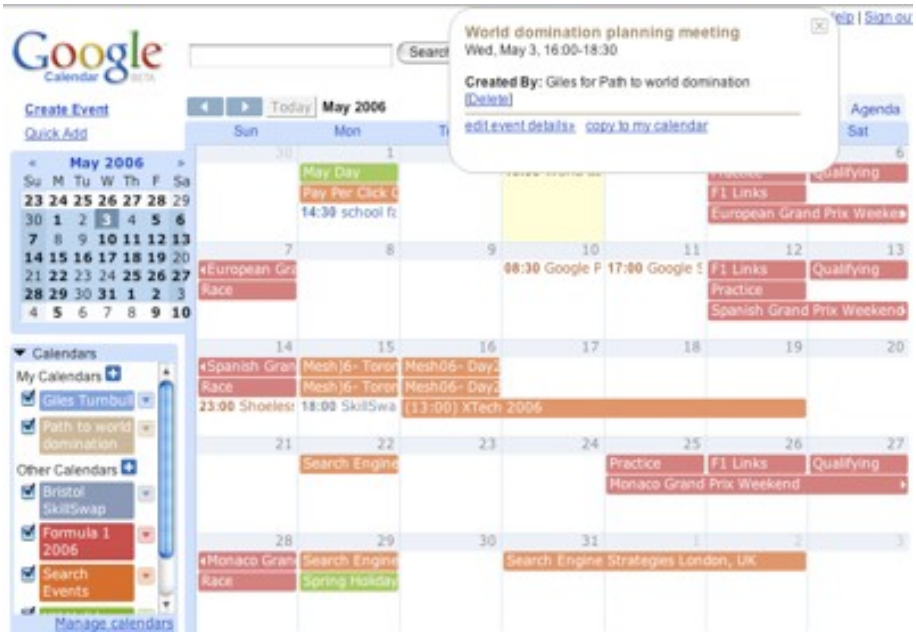


Figura 1: Google Calendar

1.2 Agenda a contrattazione automatica dell'impegno

I problemi esposti in precedenza, rispetto ai diversi tipi di agenda, sono stati in parte risolti con la creazione di software di agenda per telefoni cellulari e PDA rimane comunque irrisolto un grosso problema: il tempo.

Nella vita quotidiana, per poter avere un appuntamento con un'altra persona si devono affrontare ore al telefono, invio di e-mail, SMS continui, con una immancabile perdita di tempo e dei costi, alcune volte esosi.

Questo problema potrebbe essere risolto, introducendo sui software di agenda già presenti nei telefoni cellulari e PDA, l'automatizzazione della contrattazione dell'orario di un impegno fra due soggetti. Con questa funzione si avrebbero

enormi vantaggi: si sommerebbero i benefic,i che le agende hanno dato, ad un grosso aumento del guadagno, passando per una quantità maggiore di tempo risparmiato. Basti pensare che, all'interno di un'azienda, la grossa riduzione del tempo, data dall'immediata contrattazione degli impegni, aumenta la produttività individuale. Consentendo la partecipazione a più eventi.

Alcuni tipi di agenda di vecchia generazione non rispettavano appieno la privacy; adesso con l'innovazione che si vuole apportare, la privacy verrebbe tutelata, ovvero al di là dell'impegno in contrattazione le parti non conoscerebbero nessun 'altro degli impegni che le persone hanno in agenda. Visto che i telefoni cellulari sono già in possesso di una grossa fetta della popolazione, visto che da questi si esige continue migliorie, per rispondere ad ogni più disparata esigenza, la funzione innovativa che si vorrebbe introdurre, sarebbe disponibile e a portata di mano alla maggior parte delle persone.

L'obbiettivo di questa tesi è di creare un prototipo di scheduler per impegni, capace di organizzare in modo automatico l'incontro tra due persone. L'applicazione, che si vuole creare, dovrà essere accessibile in ogni situazione agli utenti.

1.3 Strumenti e tecnologie utilizzate

In questo paragrafo vengono elencate le tecnologie utilizzate nello sviluppo dell'applicazione.

1.3.1 Java MicroEdition (J2ME)

Ora verrà analizzato la natura della Java MicroEdition[3][4].

La massiccia diffusione di cellulari e PDA ha “costretto” la Sun a esportare Java anche su questi tipi di dispositivi, creando un'apposita versione chiamata Java MicroEdition. Si è dovuto definire un set di API che capaci di rispondere alle potenzialità dei dispositivi mobili. Questa tipologia di dispositivo presenta una limitata capacità di elaborazione, poca memoria e schermi di ridotte dimensioni. Esistono, comunque, vari tipi di dispositivi mobili che presentano caratteristiche diverse.

Non tutti i dispositivi mobili supportano J2ME.

Sun, per ovviare a questo sistema eterogeneo di dispositivi, ha introdotto il concetto di Profile e di Configuration. La Sun, con l'utilizzo di Profile e Configuration, vuole stabilire delle librerie base, comuni a tutti i dispositivi, lasciando, poi, ai costruttori di uno specifico dispositivo il compito di aggiungere nuove funzionalità sotto forma di librerie.

Il profilo che Java propone è il Mobile Information Device Profile(MIDP), giunto ormai alla versione stabile 2.0. MIDP è formato da una serie di librerie, usate per interfacciarsi al terminale, dà così la possibilità di gestire le operazioni di input, creare un'interfaccia grafica, di gestire la memoria persistente ed altre funzionalità.

Le configurazioni presenti sono principalmente due Connected Device Configuration(CDC)[3], e Connected Limited Device Configuration(CLDC)[3]. Le due configurazioni sono giunte alla versione 1.1, ma sono in costante evoluzione.

La configurazione CDC è presente in dispositivi che presentano almeno 512 kB di

memoria RAM disponibile e di una memoria a runtime di almeno 256 kB. Questi dispositivi sono solitamente dei palmari.

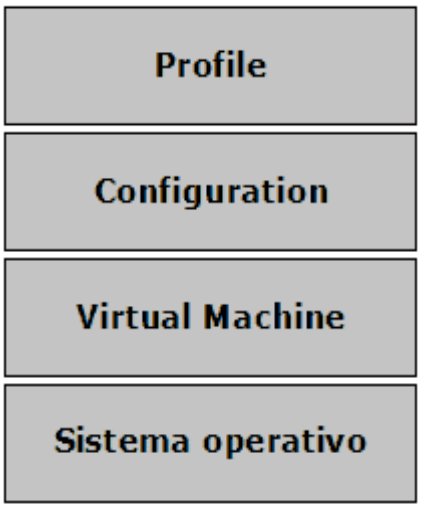


Figura 2: Architettura J2ME

CLDC è presente in terminali con proprietà meno performanti rispetto a quelle elencate in precedenza. Sono necessari almeno 128 kB di memoria RAM e almeno 16 kB di memoria a runtime.

I terminali, che supportano J2ME hanno installata una Java Virtual Machine chiamata KVM, caratterizzata da una bassa necessità di memoria.

I varii dispositivi possono supportare delle API aggiuntive, che seguono uno standard Sun, tramite delle specifiche JSR (Java Specification Request).

- Mobile Media API (JSR 135) permettono l'accesso alle funzionalità multimediali.
- Wirless Messaging API (JSR 120) utilizzate per l'invio di SMS e MMS.
- Web Service API (JSR 172) consentono l'accesso ai Web Services.
- Platform-Securety and Trust Services API (JSR 177), libreria che consente

un accesso a servizi in maniera sicura e autenticata.

- PDA Optional Package (JSR 75) permette di accedere alle informazioni personali dell'utente, come ad esempio la rubrica.

Bisogna nuovamente sottolineare che queste API addizionali non sono presenti su tutti i . Ogni produttore effettua le proprie scelte su quali API supportare. Inoltre alcune case produttrici si sono create delle API proprietarie, disponibili al programmatore su alcuni terminali.

Punti di forza di J2Me sono rappresentati dalla possibilità di creare un semplice interlacciamento con piattaforme web di tipo Java. L'installazione dell'applicazione sul terminale è molto semplice ed immediata. Viene la possibilità di utilizzare l'applicazione anche in modalità off-line.

Nonostante l'obbiettivo della Sun fosse quello di creare uno standard, esistono problemi di compatibilità tra i vari dispositivi. Questi nascono perchè ogni dispositivo presenta una propria architettura. Inoltre, l'utilizzo delle API proprietarie permettono operazioni solo su specifici dispositivi mobili. La mancanza di licenze e di alcune librerie non permettono al programmatore di avere il totale controllo delle potenzialità del dispositivo. Ad esempio, l'invio di un messaggio da parte di un'applicazione necessita la conferma da parte dell'utente. Questo è causato dal tipo di profilo presente sul terminale utilizzato.

1.3.2 Bluetooth

La scelta di utilizzare la trasmissione di messaggi tramite connessione Bluetooth[8] è nata perché questa tecnologia offre alcuni vantaggi:

-È diffuso sulla maggior parte di dispositivi mobili;

-Ha costi nulli di connessione

Il Bluetooth oltre alle caratteristiche, che hanno portato alla scelta dell'utilizzo nell'applicazione, presenta altre proprietà: ha bassi costi di consumo e una motivazione essenziale, per poter risiedere su dispositivi di dimensioni ridotte.

I dispositivi dotati di Bluetooth presentano la possibilità di connettersi tra loro, anche se presentano natura diversa, basti pensare un auricolare bluetooth che comunica con un cellulare. Possono inoltre connettersi ad una LAN tramite appositi Access Point, ma anche alla rete di telefonia mobile e alla rete pubblica PSTN(Public Switched Telephone Network).

Le piconet sono delle reti a basso raggio d'azione, formate da dispositivi provvisti di bluetooth. Sono formate da otto nodi, dove uno è il nodo master e sette sono slave. Uno stesso nodo può far parte di più piconet. Questo tipo di rete è definita a configurazione automatica, in quanto un nodo può entrare e uscire dal piconet senza la necessità di intervento umano. Un nodo viene connesso alla rete, se entra nel raggio d'azione del nodo, che svolge il ruolo di master e sconnesso, se esce dal campo d'azione. All'interno di una rete il dispositivo ha la possibilità di cercare servizi offerti oppure offrire servizi. Naturalmente, il dispositivo utilizzerà solo i servizi con cui ha la possibilità di comunicare. Più piconet formano una scatternet.

Esistono tre classi di dispositivi, riportate nella tabella sottostante, che si differenziano tra loro, per il raggio d'azione e potenza.

Classe Potenza	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 1: Classi di potenza del Bluetooth

Specifiche Bluetooth

Il SIG (Special Interest Group), nato nel 1998, è il gruppo formato dalle più importanti compagnie di telecomunicazioni, che collabora per definire le specifiche Bluetooth. Oggi, questo gruppo comprende più di 2000 compagnie.

Bluetooth 1.0 e 1.0B. Queste due versioni sono state rilasciate a pochi mesi di distanza l'una dall'altra. Bluetooth 1.0 esce nel luglio del 1999 e la 1.0B esce a dicembre dello stesso anno. Sono entrambe afflitte da numerosi problemi. I dispositivi Bluetooth prodotti da società diverse avevano grosse difficoltà nella comunicazione. Nella versione 1.0 non era permesso rimanere anonimi durante una comunicazione, permettendo così ad un utente di intercettare eventuali informazioni confidenziali. La versione 1.0B risolse questo problema.

Bluetooth 1.1 Le specifiche di questa versione sono state rilasciate nel febbraio 2001. Bluetooth 1.1 permette la comunicazione su canali non cifrati e risolve i problemi di interoperabilità.

Bluetooth 1.2 Questa versione è stata rilasciata a novembre 2003 ed è compatibile con la precedente versione. Viene introdotto l'Adaptive Frequency Hopping

(AFH), una tecnica che permette di evitare i canali soggetti a forti interferenze. Le specifiche 1.2 presentano miglioramenti della velocità di connessione tra periferiche. Vengono in tal caso utilizzati algoritmi avanzati, che rendono più rapida la ricerca tra dispositivi.

Bluetooth 2.0 E' compatibile con le versioni precedenti. Prevede l'utilizzo di onde radio di minore potenza, generando maggiore risparmio energetico e tempi di risposta notevolmente ridotti. L'utilizzo di Enhaced Date Rate (EDR) porta ad una velocità di trasmissione di 10 Mbit/sec ad una distanza di 10 metri. Bluetooth 2.0 utilizza la crittografia per garantire l'anonimato.

Bluetooth 2.1 E' stato annunciato il rilascio delle nuove specifiche Bluetooth nel marzo 2007 ed è compatibile con le versioni precedenti. Viene migliorata l'associazione tra dispositivi. Near Field Communication (NFC) permette una maggiore velocità di trasmissione dati tra due dispositivi che si trovano vicini. La tecnica, chiamata Sniff Subrating, promette un notevole risparmio di energia.

1.3.3 Short Message Service(SMS)

L'espressione SMS[11] viene utilizzata nel linguaggio comune per indicare un breve messaggio di testo inviabile da un dispositivo (cellulare, pc, telefono) ad un altro. Inizialmente l'sms era utilizzato solo dalle reti GSM. Attualmente è prevista la possibilità di usare reti di tipologia diversa come la UMTS.

La dimensione fissa del messaggio è di 140 byte ed è prevista la possibilità di

inviare più sms concatenati.

Sono presenti due diverse tipologie di messaggi. Point-to-Point(SMS/PP) utilizzati per la comunicazione da terminale a terminale e Cell Broadcast(SMS/CB), utilizzati per la diffusione in broadcast.

Ora saranno brevemente esposte le due tecnologie.

Point-to-Point(SMS/PP)

Questo tipo di servizio viene utilizzato per inviare un messaggio tra due terminali. Il messaggio, prima di giungere al destinatario, effettua il percorso proposto nell'immagine riportata sotto.

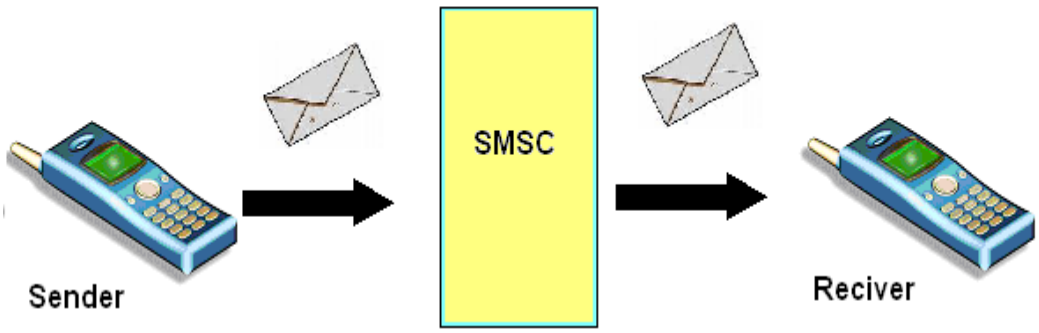


Figura 3: Funzionamento SMS/PP

Il messaggio viene spedito ad un centro servizi (SMSC, Short Message Services Centre), il quale svolge l'azione di store-and- forward. Ciò significa, che immagazzina una copia del messaggio e ne invia una al destinatario. L'operazione viene eseguita anche per poter recapitare il messaggio in un secondo momento,

nell'eventualità che il destinatario non sia momentaneamente raggiungibile.

La connessione usata tra terminale e il Centro Servizi è di tipo connectionless, quindi non è possibile avere garanzia della ricezione del messaggio e neppure del corretto invio.

Cell Broadcast(SMS/CB)

Il servizio di SMS, normalmente operante da utente ad utente, esiste anche in una variante broadcast. Ogni cella ha la possibilità di inviare brevi messaggi a tutti i terminali sotto la sua copertura, su uno specifico *canale* numerato, ciascuno dedicato ad un tipo particolare di informazione, quali emergenze, informazioni sul traffico, eccetera. Una lista non esaustiva di canali definiti dallo standard è la seguente:

- 000 indice
- 010 notizie flash
- 020 ospedali
- 022 dottori
- 024 farmacie
- 030 viabilità strade a lunga percorrenza
- 032 viabilità strade locali
- 034 taxi

- 040 condizioni meteorologiche
- 050 distretto
- 052 informazioni sulla rete
- 054 servizi tramite operatore
- 056 elenco abbonati nazionale
- 057 elenco abbonati internazionale
- 058 servizio clienti nazionale
- 059 servizio clienti internazionale

Attualmente, in Italia, il solo servizio abilitato è il nr. 50: se il cellulare ricevente ha abilitata la ricezione dei messaggi locali su tale canale, ogni 10 secondi circa viene mostrata sullo schermo dell'utente la provincia in cui è posta la cella a cui l'utente è registrato.

L'impiego originario di tale canale era connesso ad un tipo di tariffazione basata sulla posizione dell'utente: nel caso specifico, le chiamate effettuate dal proprio distretto di residenza erano tariffate ad un costo minore.

Capitolo 2

Analisi di BlueOrganizer

L'applicazione BlueOrganizer intende superare alcuni limiti posti dalle agende attualmente disponibili. Si vuole così proporre un software accessibile da cellulari e PDA. La scelta cade sui dispositivi mobili con la principale motivazione di utilizzare un tipo di dispositivo accessibile abitualmente in ogni momento. L'attuale diffusione di dispositivi mobili da la possibilità a quasi la totalità dei soggetti di poter disporre del prodotto.

Il problema principale, trattato da BlueOrganizer, è l'automatizzazione della contrattazione dell'orario di un impegno tra due soggetti. Con questo processo innovativo si vuole annullare la quantità di tempo, che i soggetti interessati usano, per fissare un appuntamento. Ciò implica l'eliminare telefonate, invio di email o qualsiasi tipo di accordo diretto tra i due protagonisti, ed un conseguente notevole guadagno di tempo per entrambi gli utenti.

Con il concetto di contrattazione automatica, viene superato l'utilizzo di agenda condivisa, se adoperata per scopi strettamente personali e non pubblicitari. L'utente non è più obbligato a rendere pubblici, una parte o la totalità, dei propri impegni personali; sarà l'agenda stessa ad amministrarlo.

2.1 Analisi dei requisiti

In tabella 2 è riportato il glossario contenente la terminologia utilizzata all'interno di questa tesi.

GLOSSARIO	
Termine	**Significato**
Tempo libero	Arco temporale dove l'utente non ha previsto nessun impegno;
Contrattazione	Processo che porta a fissare un impegno tra due persone tramite la ricerca del tempo libero comune;
Impegno Concordato	Impegno tra due persone con data di svolgimento concordata tramite una contrattazione da parte del software BlueOrganizer;
Impegno Singolo	Impegno che presenta scadenza singola e non necessita di contrattazione;
Impegno Routine	Impegno che si presenta in modo costante all'interno di uno spazio temporale e non necessita di contrattazione;
Classe delle attività	Categorie utili a classificare l'impegno (Lavoro, Famiglia, Att. Sportive, ecc)
Attività correlate	Classi delle attività di un impegno. Determinano quali attività in caso di contrattazione automatica possono eliminare l'impegno
Utente "A"	Utente che invia la richiesta di contrattazione;
Utente "B"	Utente che riceve la richiesta di contrattazione.

Tabella 2: Glossario

2.1.1 Requisiti funzionali

In tabella 3 sono riportati i requisiti funzionali[1] del sistema.

L'utente per poter usufruire del servizio d'agenda di BlueOrganizer deve scaricare il software dall'apposito sito internet. Si tratta di un'applicazione installabile sul proprio dispositivo mobile. L'operazione non richiede nessun tipo di registrazione

1 **Requisiti funzionali**: descrivono le funzionalità ed i servizi offerti dal sistema

e non è prevista nessuna sorta di pagamento.

Per poter usufruire di un servizio di agenda completo, si è pensato di dare la possibilità all'utente di inserire diverse tipologie di impegno: occasionali o sistematici a scadenza fissa. Quindi, si vuole che l'agenda supporti impegni con scadenza singola, per impegni occasionali, ma soprattutto di agevolare l'inserimento di impegni ripetitivi, senza obbligare l'utente ad aggiungerli uno alla volta. Per facilitare l'inserimento di questi impegni di routine, l'utente dovrà semplicemente inserire un unico impegno specificando la data che determina l'inizio e la fine, in cui l'impegno è valido e in che giorni della settimana, con i rispettivi orari in cui viene svolto.

Oltre a questi due tipi di impegni l'applicazione, deve permettere all'utente di poter impegnarsi con un altro senza dover accordasi in maniera diretta. Ma delegare all'applicazione il compito di trovare del tempo libero comune ai due utenti e fissare l'appuntamento. L'agenda deve permettere l'inserimento di un terzo tipo di impegno che specifichi le generalità della persona con cui ci si vuole incontrare e in che periodo. Le agende provvederanno a fissare l'impegno tramite uno scambio di informazioni.

Si vuole in tal modo creare un apposito scheduler per impegni, adattabile a stili di vita di soggetti con esigenze ed abitudini diverse.

Lo scheduler non deve permettere ad un impegno di sovrastare un'altra attività, salvo specifiche dell'utente.

Lo scheduler deve disporre di categorie in grado di classificare l'impegno. Per ogni categoria deve essere prevista la possibilità di rendere attiva la contrattazione

in determinate ore e giorni della settimana, dando così all'utente la facoltà di decidere quali attività svolgere durante ogni giornata.

Inoltre lo scheduler deve prevedere la possibilità di eliminare un impegno con priorità inferiore.

N°	REQUISITO FUNZIONALE
1	L'utente per poter usufruire di BlueOrganizer deve scaricare l'applicazione dall'apposito sito internet;
2	Il software deve prevedere la possibilità di inserire Impegni Concordati;
3	La contrattazione dell'Impegno Concordato deve risultare trasparente agli utenti;
4	L'utente deve poter scegliere in che giorno della settimana e in che orario un'attività può essere soggetta a contrattazione;
5	Se l'utente decide di modificare o eliminare un Impegno Concordato l'applicazione deve inviare una notifica all'utente che partecipava all'impegno;
6	Il software deve prevedere l'inserimento di Impegni Singoli e di Impegni Routine;
7	Non ci possono essere impegni sovrapposti;
8	Impegni con priorità superiore devono avere la possibilità di eliminare impegni con priorità inferiore.

Tabella 3: Requisiti funzionali

2.1.2 Requisiti non funzionali

In tabella 4 sono riportati i requisiti non funzionali[2] del sistema.

Il programma non è indirizzato ad una specifica fascia d'utenza, ma è ideato per

2 **Requisiti non funzionali** non sono collegati direttamente con le funzioni implementate dal sistema, ma definiscono vincoli sullo sviluppo del sistema

essere accessibile a tutti i possessori di un cellulare o PDA. Il dispositivo mobile deve supportare la piattaforma Java, visto che l'applicazione utilizza questo linguaggio di programmazione. La scelta di J2ME[3] viene analizzata successivamente.

Il sistema dovrà essere di facile utilizzo da parte di chi non ha grande dimestichezza con i dispositivi mobili. Il sistema dovrà essere aperto a tutti, proprio per questo motivo non dovranno essere richieste particolari abilità all'utente per la configurazione e l'inserimento dei propri impegni.

Per poter accordare l'impegno concordato l'applicazione dovrà comunicare con l'agenda interessata. Ci dovrà essere uno scambio di informazioni. tra le agende interessate contenenti orari disponibili per eventuali appuntamenti attraverso messaggi, tramite connessione Bluetooth se ciò risulta possibile, altrimenti mediante messaggi SMS.

N°	REQUISITO NON FUNZIONALE
1	L'applicazione deve essere sviluppata con un ambiente di sviluppo J2ME;
2	Il software deve essere accessibile a dispositivi mobili (cellulari e PDA), che supportano la piattaforma Java;
3	La comunicazione tra le applicazioni deve avvenire tramite lo scambio di messaggi SMS o connessione Bluetooth. I dispositivi privi di quest'ultima tecnologia potranno avvalersi della sola comunicazione tramite lo scambio di messaggi SMS;
4	L'applicazione dovrà essere semplice da utilizzare e non dovrà richiedere particolare dimestichezza con i dispositivi mobili.

Tabella 4: Requisiti non funzionali

3 Acronimo della versione Java per dispositivi mobili. J2ME o chiamata anche Java MicroEdition è la versione Java illustrata nel primo capitolo.

2.2 Scelta delle tecnologiche utilizzate

Prima di sviluppare il software, si sono dovute fare alcune scelte sia sul tipo di linguaggio di programmazione da utilizzare, sia sulle modalità in cui le agende potranno comunicare fra loro.

La scelta di utilizzare Java come linguaggio di programmazione è nata dalla forte portabilità che da questa viene offerta. Infatti, come si è visto nel precedente capitolo, Java è supportato da tutti i sistemi che implementano una Java Virtual Machine. I linguaggi alternativi, presi in considerazione, sono c++ e Python. Entrambi offrono maggiori prestazioni rispetto Java, ma hanno come grossa limitazione il vincolo a sistemi operativi sviluppati in Symbian. Si è preferito, così, adottare un linguaggio meno performante, ma che massimizza la possibilità di diffusione dell'applicazione.

Le agende per poter organizzare in maniera automatica l'impegno tra due persone, devono scambiarsi delle informazioni. Per effettuare tale operazione sono state effettuate due scelte. La comunicazione mediante connessione Bluetooth è la principale. Questa tecnologia wirless è la più diffusa su dispositivi mobili, grazie anche ai bassi costi dell'hardware richiesto. L'applicazione è stata denominata BlueOrganizer, proprio per evidenziare l'utilizzo di questa tecnologia. Essa presenta come limitazione un basso raggio d'azione. E' stata comunque utilizzata in attesa di una futura diffusione del WiFi su dispositivi mobili. Visto che la piattaforma J2Me utilizza le medesime API di comunicazione per i protocolli Bluetooth e WiFi, in futuro basterebbe specificare il nuovo protocollo all'interno del codice dell'applicazione e la comunicazione tramite WiFi sarebbe disponibile.

Nel frattempo per ovviare ai problemi di località posti dal Bluetooth, si è deciso di utilizzare come mezzo di trasporto delle informazioni i messaggi SMS. Ciò consente il trasporto di una quantità limitata di informazioni e ogni invio presenta un costo per l'utente.

2.3 Creazione degli impegni

Le procedure descritte in questo paragrafo sono state inserite con l'obbiettivo di evidenziare le differenze tra le tipologie d' impegno utilizzabili in BlueOrganizer.

Impegno Concordato

L'activity diagram (Figura 4), mostra le fasi di creazione di un Impegno Concordato. I primi passi prevedono l'inserimento dei dati, che identificano l'impegno (nome, classe, attività correlate), e le caratteristiche della contrattazione (data, ora d'inzio e fine, modalità). La scelta dell'utente “B” stabilisce se la contrattazione avviene tramite connessione Bluetooth o scambio di messaggi SMS. Inseriti questi dati sarà l'applicazione, attraverso uno scambio di informazioni con l'agenda dell'utente selezionato, a determinare l'orario in cui verrà svolto l'impegno dai due soggetti.

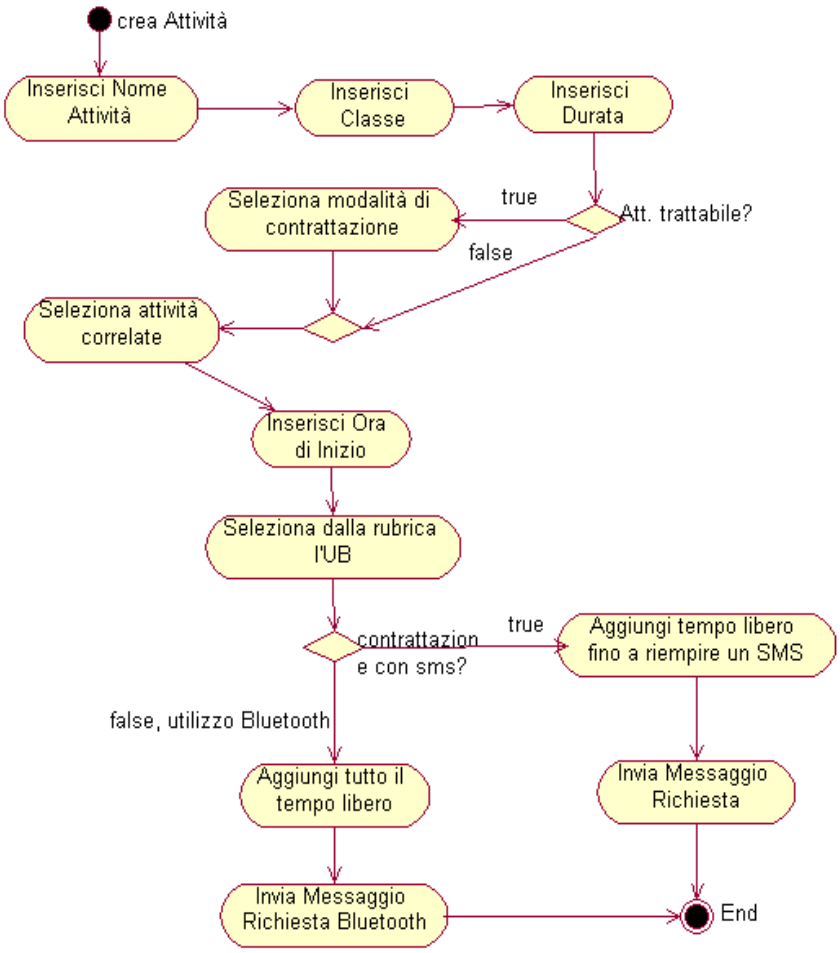


Figura 4: Activity Diagram Impegno Concordato

Impegno Singolo

La procedura di inserimento di Impegno Singolo(Figura 5), presenta il settaggio dei seguenti dati che caratterizzano l'impegno: nome, classe, data, ora d'inizio, ora fine e attività correlate. Non è previsto nessun tipo di contrattazione. Questa modalità è utile all'utente ad inserire un impegno personale che non necessita l'incontro con un'altra persona.

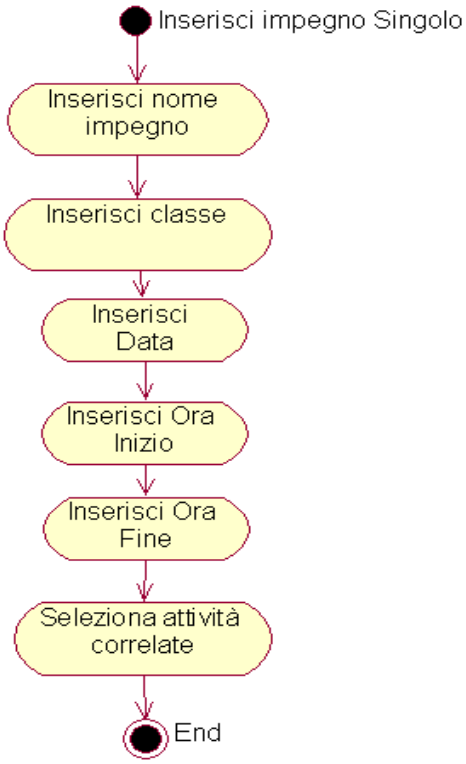


Figura 5: Activity Diagram Impegno Singolo

Impegno Routine

I passaggi utili all'inserimento di un Impegno Routine sono schematizzati nell'activity diagram presente nella figura 6. I dati da inserire per la creazione di questa tipologia di impegno sono: nome, classe, data da cui l'impegno è attivo, data di scadenza dell'impegno. Vista la natura dell'impegno, l'utente deve selezionare i giorni della settimana in cui si verifica l'impegno e inserire i relativi orari.

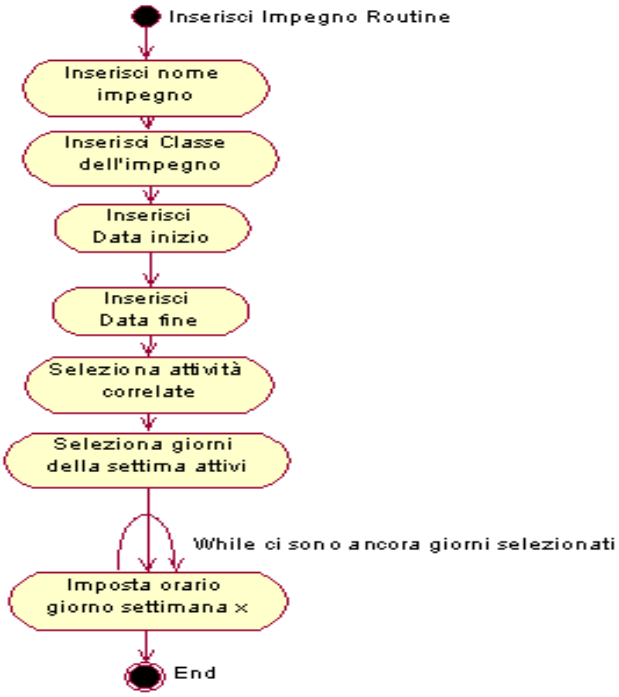


Figura 6: Activity Diagram Impegno Routine

2.4 Architettura e gestione dei messaggi di contrattazione

La contrattazione dell'Impegno Concordato prevede lo scambio di messaggi tramite connessione Bluetooth e SMS. In questo paragrafo vengono analizzate le architetture dei messaggi e l'interpretazione che ricevono.

2.41 Messaggi tramite connessione Bluetooth

Architettura dei messaggi utilizzati

Sono presenti due tipi di messaggio. MessaggioRichiestaBLTH e MessaggioRispostaBLTH (Figura 7). MessaggioRichiestaBLTH viene inviato dall'utente "A". MessaggioRispostaBLTH è può contenere quattro modalità di

risposta :

- “Fine Connessione” e Orario non più disponibile” sono inviate dall'utente “A”;
- “Conferma” e “Impossibile fissare l'impegno” sono inviate dall'utente “B”.

Figura 7: Architettura messaggi Bluetooth

Contrattazione Bluetooth

Nei sequence diagram riportati sotto (figure 8, 9, 10)**,** viene illustrata la contrattazione tra due applicazioni BlueOrganizer utilizzando lo scambio di messaggi tramite connessione bluetooth.

In tutti i casi riportati, la contrattazione inizia con l'invio da parte di “A” di un messaggio di richiesta di una nuova contrattazione. Tale messaggio contiene i dati identificativi dell'impegno e la totalità del tempo libero che “A” dispone. L'applicazione “B”, appena riceve la richiesta, confronta il proprio tempo libero con quello contenuto nel messaggio ricevuto. Il confronto permette di scartare il tempo libero non comune a entrambi gli utenti.

Nel caso (Figura 8), in cui “B” non trova nessun tempo libero comune, invia ad “A” un messaggio con il quale notifica l'impossibilità di fissare l'impegno. La connessione tra le due applicazioni termina.

In caso contrario, l'applicazione “B” fissa l'impegno sulla propria agenda ed invia ad “A” l'orario selezionato.

Giunti a questo punto si possono verificare due casi.

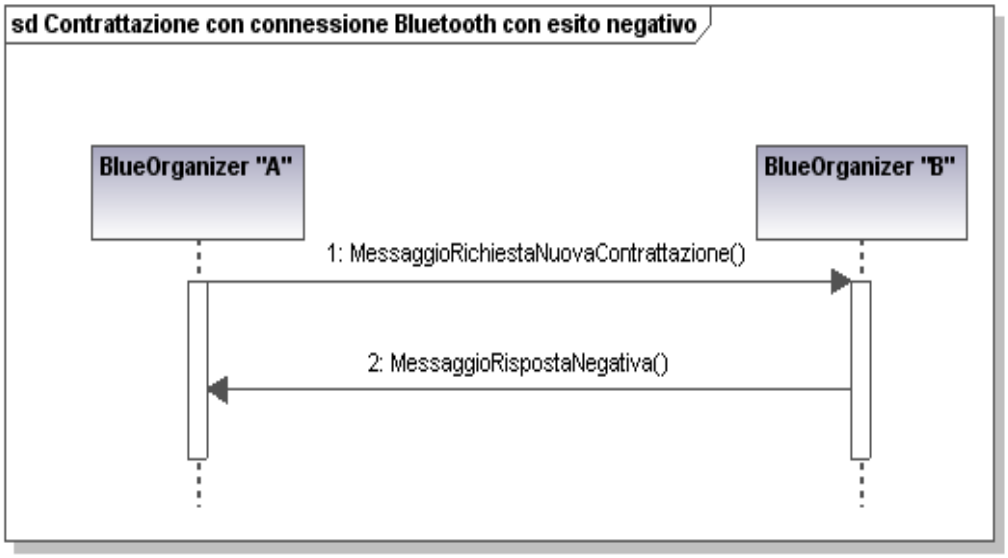


Figura 8: Sequence Diagram Contrattazione Bluetooth con esito negativo

La figura 9 mostra la situazione in cui “A” dispone ancora del tempo libero che precedentemente aveva inviato. Quindi anche “A” fissa l'impegno ed invia un messaggio a “B” con il quale comunica che la contrattazione si è conclusa con esito positivo. La connessione viene interrotta.

Nel secondo caso (Figura 10), “A” riceve la proposta di orario da “B”. Ma l'orario

scelto non risulta più disponibile (es. durante le fasi di contrattazione l'utente inserisce un impegno che occupa la fascia oraria precedentemente libera). “A” invia a “B” un messaggio che riferisce l'impossibilità di fissare l'impegno nell'orario proposto. “B” toglie dalla propria agenda l'impegno precedentemente inserito, seleziona il prossimo tempo libero comune, fissa l'impegno ed invia la nuova proposta ad “A”.

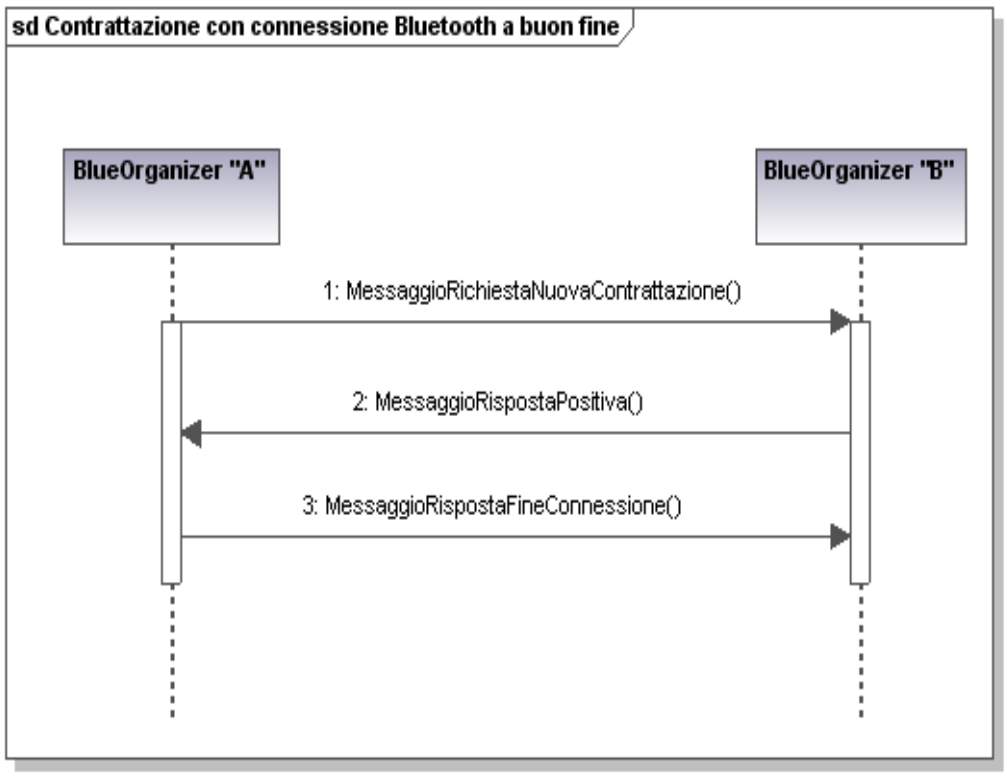


Figura 9: Sequence Diagram Contrattazione Bluetooth con esito positivo

Questa fase (Figura 10 messaggio 3-4), si ripete fino che “A” accetta l'orario proposto e da confermare con il messaggio di fine connessione a “B”. Nel caso in cui, “B” non disponga di altri orari da proporre, e invia ad “A” il messaggio di

notifica dell'impossibilità di fissare l'impegno.

In tutti i casi analizzati, la connessione, e il relativo scambio di messaggi termina solo quando la contrattazione ha ottenuto un esito, che può essere positivo nel caso in cui l'impegno venga fissato o negativo nel caso in cui non esista del tempo libero comune da destinare all'impegno.

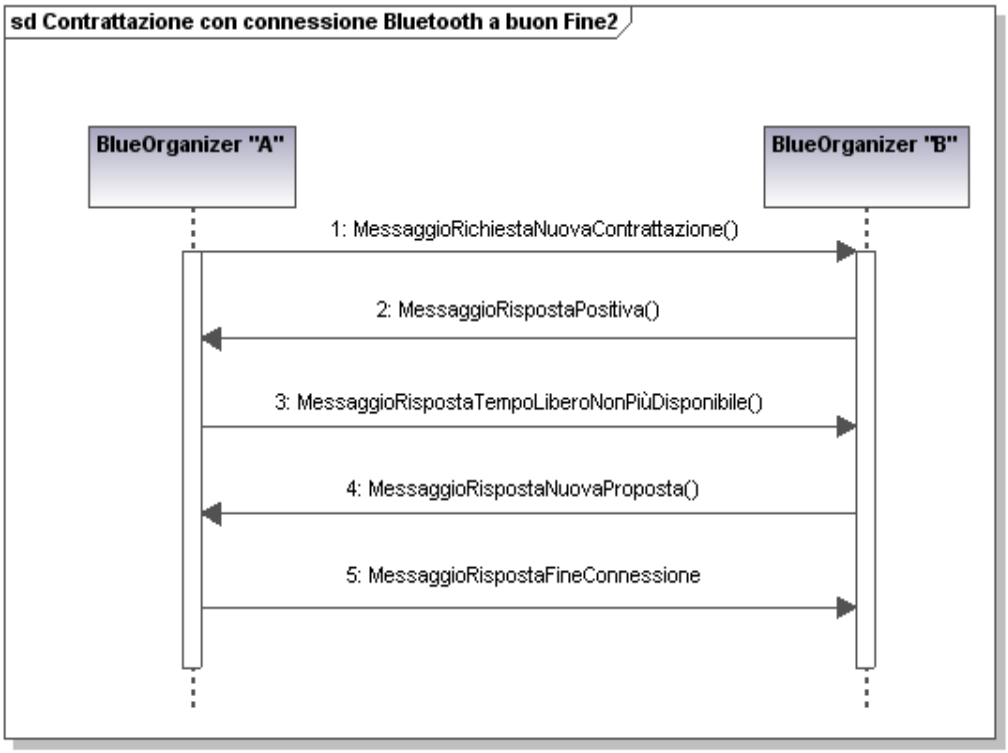


Figura 10: Sequence Diagram Contrattazione Bluetooth con esito positivo 2

Interpretazione dei messaggi

L'activity diagram, (Figura 11), mostra come vengono interpretati i messaggi ricevuti durante la connessione Bluetooth.

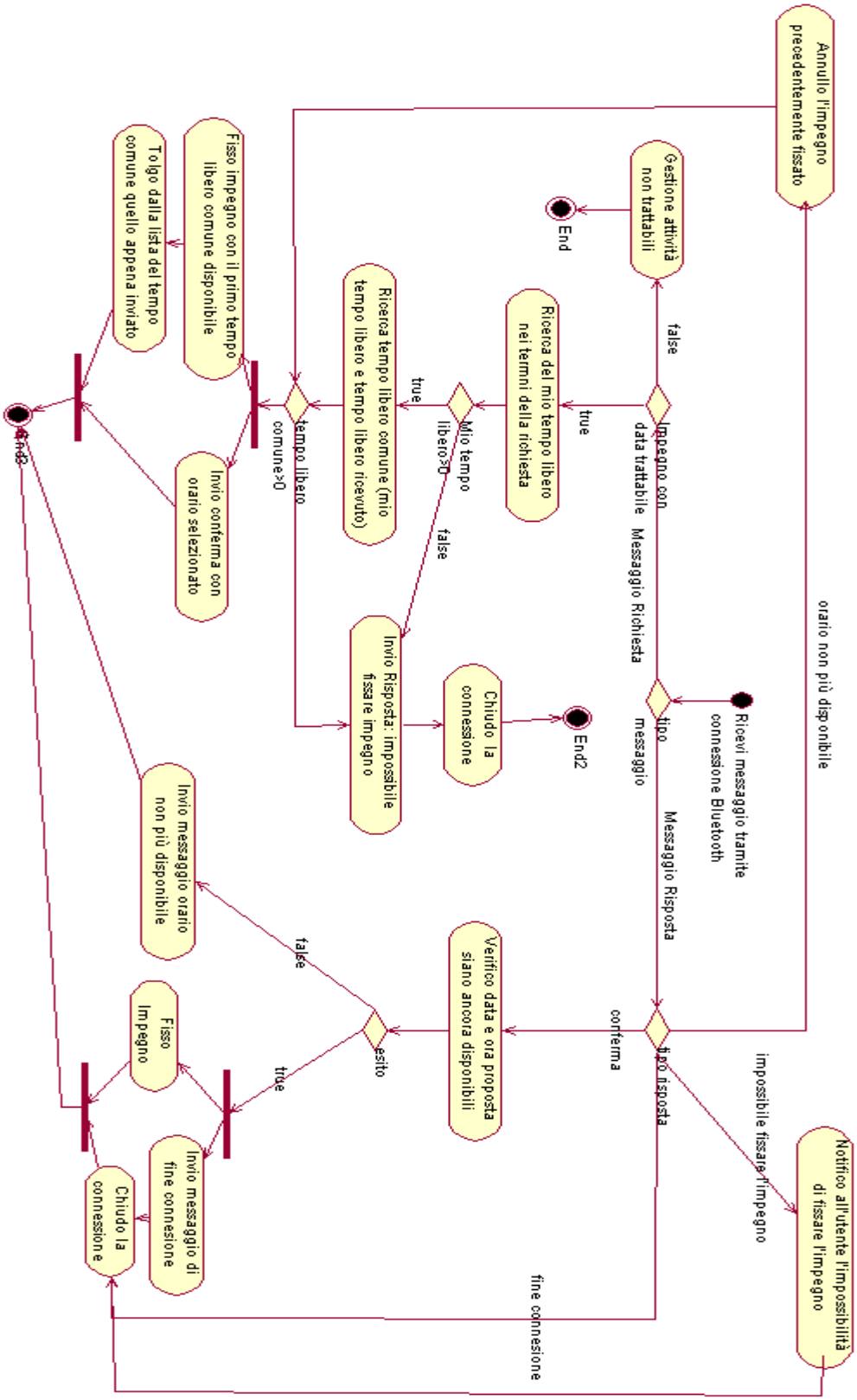


Figura 11: Activity Diagram Gestione messaggi Bluetooth

2.4.2 Messaggi tramite SMS

Architettura dei messaggi utilizzati

L'applicazione non prevede l'invio di più SMS concatenati, quindi la dimensione massima di ogni messaggio è di 140 byte.

Sono previste tre architetture di messaggio: MessaggioRichiestaSMS, MessaggioRispostaSMS e MessaggioModificaSMS.

I messaggi vengono incapsulati all'interno di un MessaggioGenerale. L'applicazione tramite l'id di MessaggioGenerale stabilisce il tipo di messaggio trasportato.

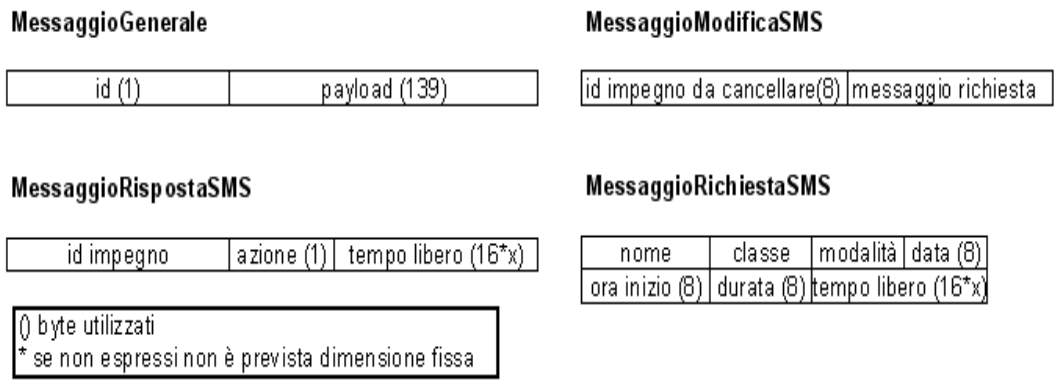


Figura 12: Architettura messaggi SMS

Contrattazione tramite SMS

I sequence diagram **(**Figure 13, 14, 15**),** illustrano lo scambio di informazioni tramite messaggi SMS. Come precedentemente citato, la dimensione massima di un SMS è di 140 byte. Si è quindi stabilito di inviare, in ogni messaggio,

informazioni che non superino tale dimensione. La contrattazione prevede la massimizzazione dell'utilizzo dei byte, di cui l'sms dispone, inserendo sempre del tempo libero relativo all'utente che invia il messaggio, in modo da ottimizzare i costi. L'applicazione può ricevere contemporaneamente messaggi relativi a contrattazioni di diversi impegni, visto che (come analizzato nel paragrafo 1.3.3), il servizio di Short Message Service presenta una connessione connectionless.

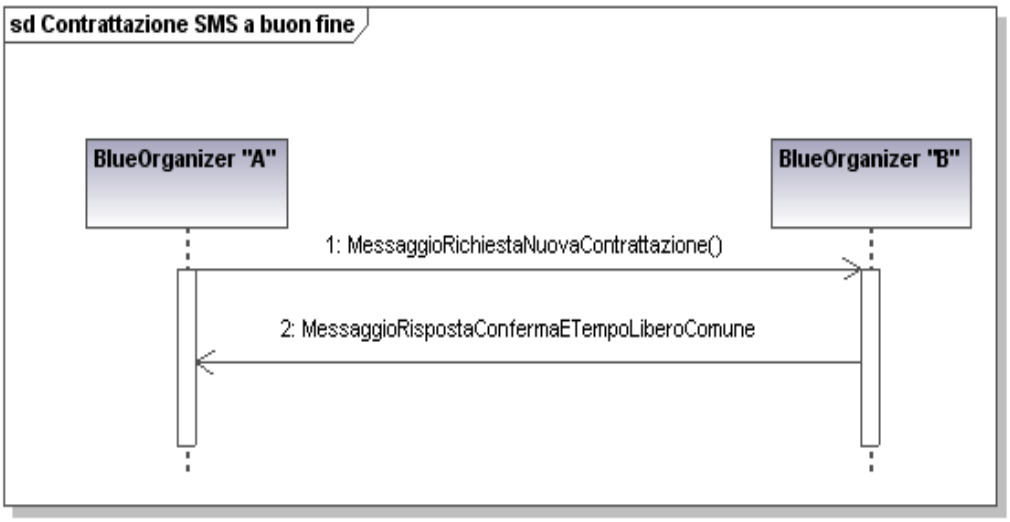


Figura 13: Sequence Diagram Contrattazione SMS con esito positivo

In tutti i casi che verranno analizzati il primo messaggio viene inviato da “A” a “B” costituisce una richiesta di nuova contrattazione. Il messaggio si compone di da dati che identificano l'impegno e il tempo libero a disposizione di “A”. “B” deve memorizzare i dati relativi all'impegno fino a contrattazione finita. L'operazione sarà utile ad associare i futuri messaggi relativi all'impegno.

L'applicazione “B” verifica se, tra il tempo libero presente nel messaggio ricevuto

esiste del tempo libero comune. “B” fissa l'impegno sulla propria agenda ed invia ad “A” l'orario scelto, il rimanente tempo libero comune (se presente) e il proprio tempo libero, con data e ora superiore all'ultimo tempo libero contenuto nel messaggio ricevuto precedentemente. “A” verifica che l'orario proposto sia ancora disponibile e, in caso affermativo, fissa l'impegno (Figura 13).

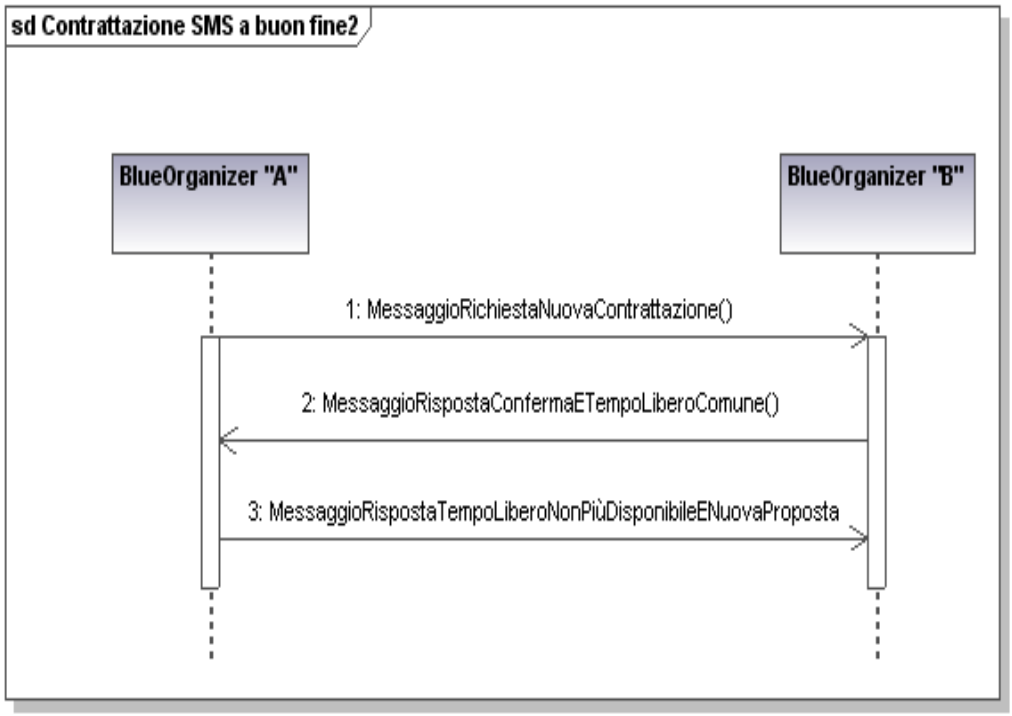


Figura 14: Sequence Diagram Contrattazione SMS con esito positivo 2

Se l'orario proposto risultasse occupato (Figura 14), l'applicazione confronta il tempo libero presente nel messaggio ricevuto, con il proprio. Se esiste tempo libero comune viene inviato all'utente “B” un messaggio contenente una notifica di eliminazione dell'impegno inserito in precedenza, una nuova proposta di orario e se presente del tempo libero.

Le fasi appena analizzate possono ripetersi finché “A” o “B” accetta l'orario proposto, o non esiste tempo libero comune (figura 15) entro la data stabilita ad inizio contrattazione.

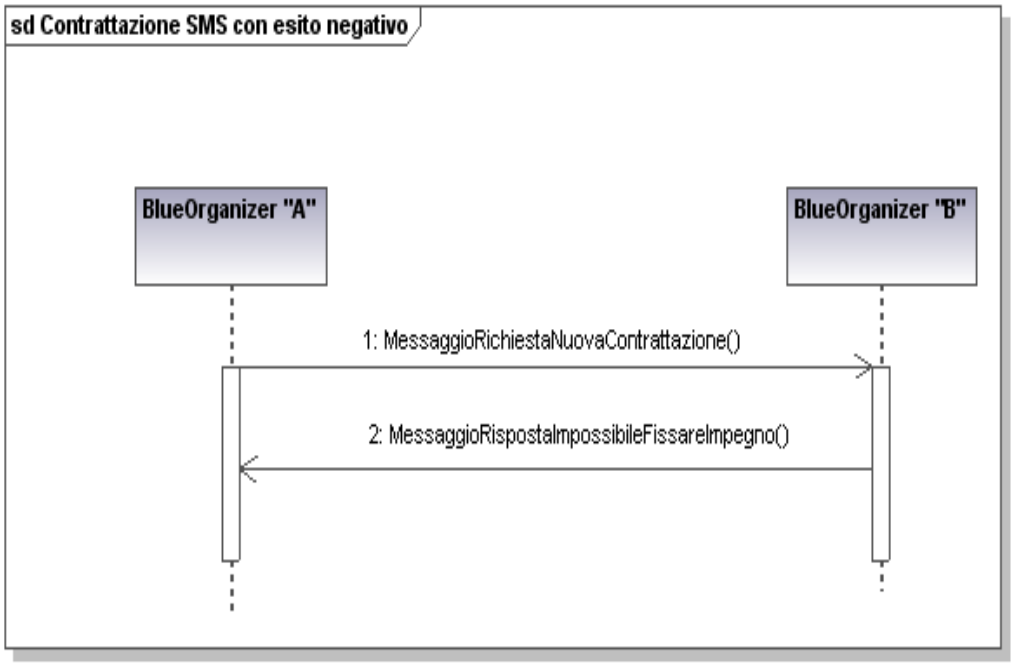


Illustration 15: Sequence Diagram Contrattazione SMS con esito negativo

Interpretazione dei messaggi

Nell'activity diagram (Figura 16), viene analizzato come l'applicazione interpreta i messaggi SMS ricevuti.

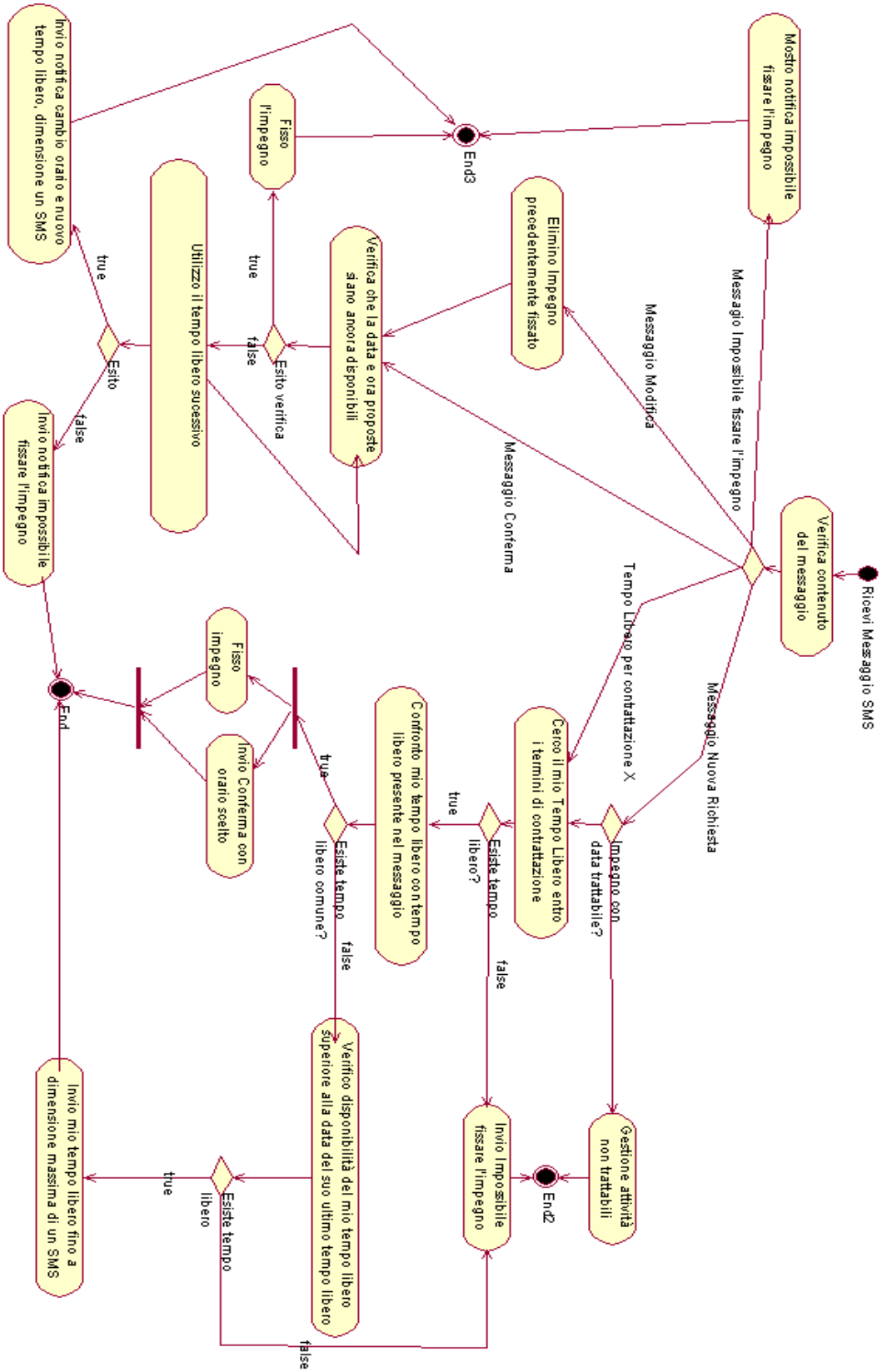


Figura 16: Gestione Messagi SMS

2.4.3 Gestione attività non trattabili

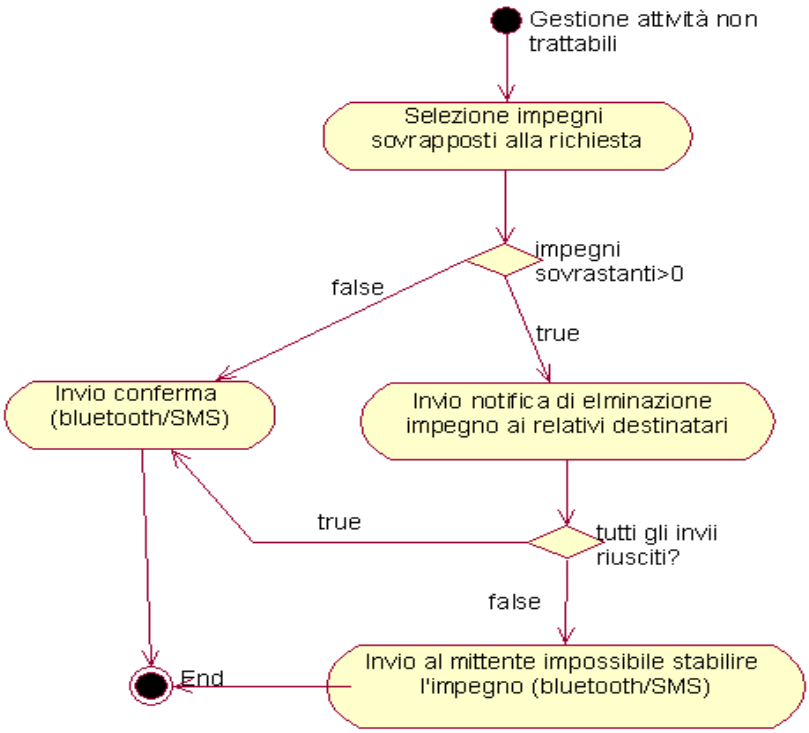


Figura 17: Gestione attività non trattabili

Nella figura 17 è riportato l'activity diagram che rappresenta la gestione delle attività non trattabili, ovvero quando gli Impegni Concordati non prevedono un margine di tempo utile alla contrattazione, ma assumono validità nel solo caso in cui l'impegno è fissato nell'orario stabilito dall'utente "A".

2.4.4 Ricerca del tempo libero

L'activity diagram presente nella figura 19 analizza il funzionamento dell'algoritmo utile alla ricerca del tempo libero che un utente ha a disposizione.

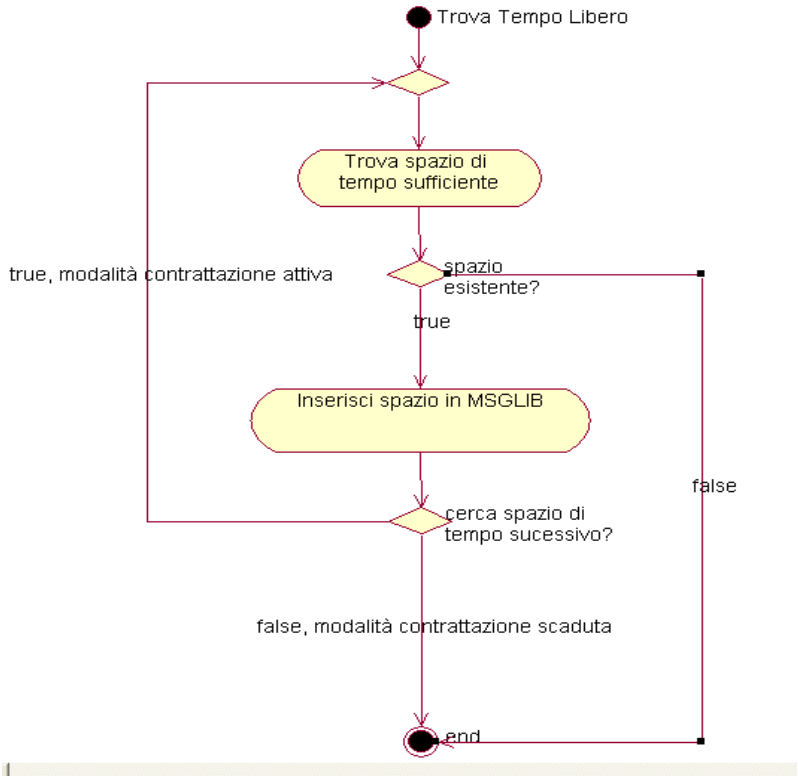


Figura 18: Ricerca del tempo libero

2.5 Class Diagram di BlueOrganizer

La figura 19 rappresenta il class diagram a livello concettuale dell'applicazione BlueOrganizer. Sono state riportate solo le classi principali per rendere il diagramma leggibile.

Il software è suddiviso in sei package principali. Si possono identificare le classi di un determinato package dal colore con cui sono visualizzate nel class diagram.

Package recordstore

L'unica classe contenuta da recordstore è RecordStoreManager il quale contiene i

metodi necessari alle operazioni di caricamento e salvataggio dei dati persistenti.

Package screen

In questo package sono contenute le classi che implementano l'interfaccia grafica. Risulta superfluo elencarle tutte visto che presentano caratteristiche molto simili. Nel class diagram si nota la presenza di VistaAgenda, GestioneImpegnoConcordato e ScreenGenerali.

VistaAgenda contiene i metodi utili alla visualizzazione dell'interfaccia principale di BlueOrganizer.

GestioneImpegnoConcordato è la classe che gestisce la creazione, la modifica e l'eliminazione di un Impegno Concordato. A differenza degli altri screen questa classe contiene l'istanza delle classi utili alla comunicazione tra BlueOrganizer.

ScreenGenerali è una classe fittizia che rappresenta tutte le rimanenti classi del package, utili al seeting dell'applicazione o a creare, visualizzare, modificare, eliminare le varie tipologie di impegni e contatti Come si può vedere dal class diagram nessun screen richiama un altro screen. Lo switch tra una schermata e l'altra è gestito dalla classe main che verrà esposta successivamente.

Package utility

In utility sono presenti le classi Utility e Sort.

Utility contiene funzioni che svolgono operazioni di selezione, controlli e confronto tra dati.

Sort contiene metodi statici utili a svolgere l'ordinamento di impegni.

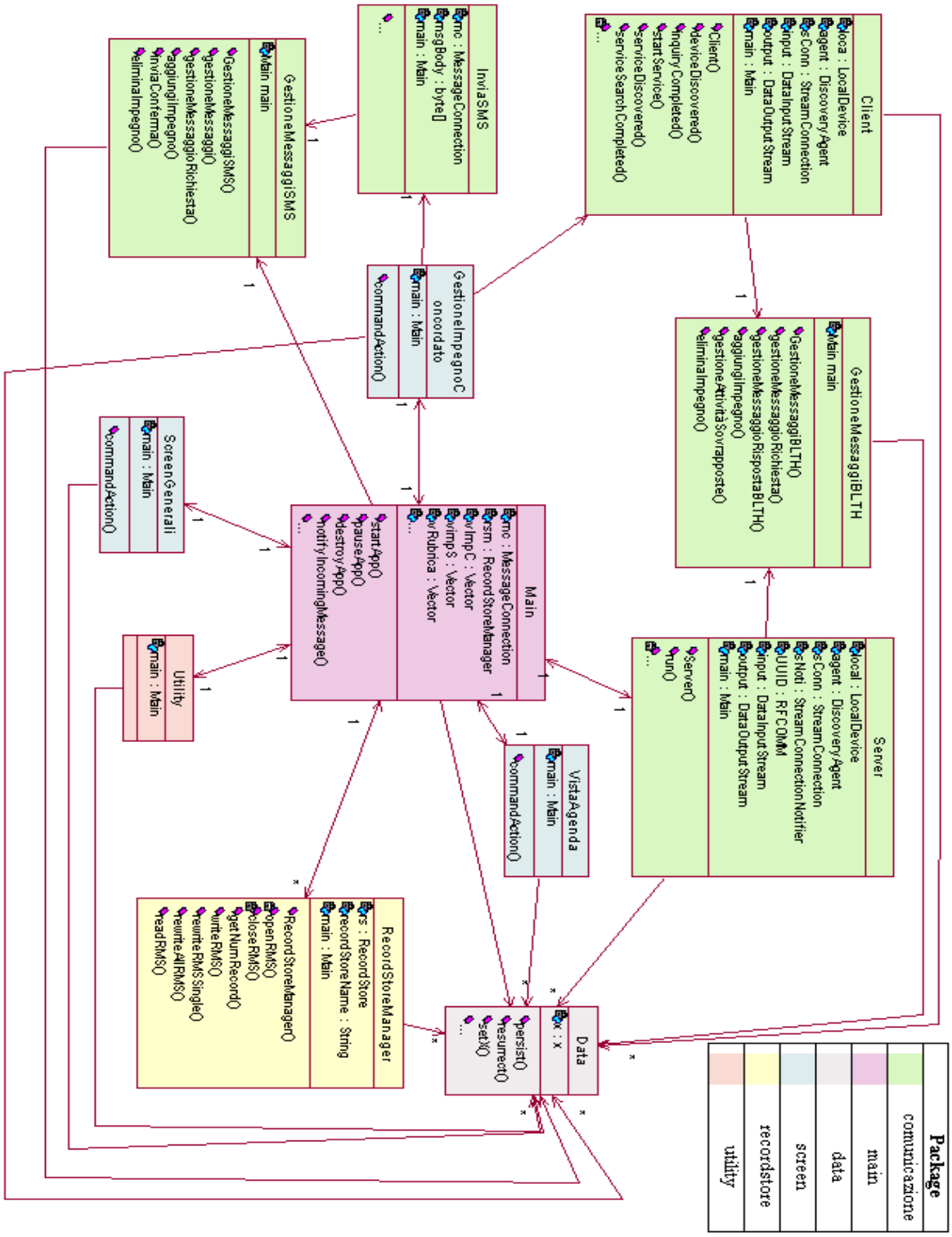


Figura 19: Class Diagram di BlueOrganizer

Package data

In data sono presenti le classi degli oggetti (Messaggi, Impegni), utilizzati dall'applicazione.

Package comunicazione

Le classi di questo package vengono utilizzate per la comunicazione tra BlueOrganizer.

Server e Client sono dei Thread relativi alla connessione Bluetooth.

InviaSMS è un Thread che ha la funzione di inviare messaggi binari. Per la ricezione di messaggi SMS non è necessario creare una classe dedicata. La midlet dell'applicazione implementa un'apposita interfaccia che gestisce la ricezione di SMS.

GestioneMessaggiSMS e GestioneMessaggiBLTH contengono i metodi che interpretano i messaggi ricevuti, provvedendo alla formulazione di un messaggio di risposta.

Package Main

L'unica classe presente in questo package è la midlet Main. E' il cuore dell'applicazione. Gestisce tutte le classi precedentemente discusse.

Capitolo 3

Sviluppo di BlueOrganizer

Questo capitolo tratta lo sviluppo di BlueOrganizer, senza mostrare direttamente il codice prodotto.

3.1 Comunicazione tra dispositivi

BlueOrganizer utilizza due diverse modalità di comunicazione. La principale utilizza uno scambio di messaggi, mediante una connessione Bluetooth. La seconda forma di comunicazione avviene tramite lo scambio di messaggi SMS.

3.1.1 Comunicazione tra dispositivi utilizzando le API Java JSR-82

Per permettere la comunicazione tramite bluetooth dei BlueOrganizer, sono state create due apposite classi utilizzando le API Java JSR-82[5][9][10], una che compie la funzione di Server e una di Client.

Server Bluetooth di BlueOrganizer

Il server viene utilizzato per permettere al dispositivo in cui è installato BlueOrganizer di condividere il servizio con altri dispositivi. Per poter effettuare questa operazione, la classe Server ha bisogno di rendere visibile il servizio agli altri . La classe LocalDevice delle API Java contiene il metodo setDiscoverable(), che permette di settare il grado di visibilità del dispositivo. Nel caso di BlueOrganizer è stato settato un grado di visibilità totale. Per poter accettare la

comunicazione e rendere visibili i servizi offerti il server deve contenere l'istanza della classe StreamConnectionNotifier e della classe StreamConnection. L'inizio della comunicazione avviene con la chiamata a Connector.open() che restituisce un oggetto di tipo StreamConnectionNotifier. Il parametro che viene passato al metodo Connector.open() è un URL che contiene tutti gli identificativi del servizio che verranno analizzate successivamente. Il servizio non risulta ancora accessibile agli altri dispositivi finché non avviene la chiamata del metodo acceptAndOpen(), contenuto nell'istanza di StreamConnectionNotifier. Ora i servizi offerti da BlueOrganizer risultano disponibili agli altri . Appena giunge una richiesta di connessione da parte di un client, viene notificata a StremConnection che consente di creare dei canali di comunicazione. Con la chiamata di openDataInputStream(), viene creato il flusso di dati in entrata e openDataOutputStream(), per il flusso di dati in uscita. Ovviamente la connessione viene accettata solo nel caso in cui il client ricerca il servizio offerto dal server di BlueOrganizer. Ora il server è in grado di soddisfare le richieste del client.

Client Bluetooth di BlueOrganizer

Il client ha il compito di verificare, nei dispositivi master presenti, la disponibilità del servizio richiesto e portare a termine la contrattazione dell'impegno.

La classe Client per poter svolgere l'operazione di ricerca di dispositivi e servizi, deve implementare DiscoveryListener e i relativi metodi. I metodi startInquiry(),

deviceDiscovery(), inquiryCompleted() e retrieve() sono relativi alla scoperta dei dispositivi presenti. Invece servicesDiscovered() e serviceSearchCompleted() riguardano la ricerca dei servizi.

L'azione che svolge la ricerca dei dispositivi è denominata inquiry.

StartInquiry() inizia la ricerca dei dispositivi. Ogni volta ne rileva uno, avviene la chiamata automatica del metodo deviceDiscovered(), che ha il compito di aggiungerli alla lista dei scoperti. I dispositivi sono aggiunti come istanze della classe Remote (contenente metodi che restituiscono l'indirizzo bluetooth o il nome alfanumerico del dispositivo aggiunto). Retrieve() restituisce i dispositivi che erano stati trovati in ricerche precedenti. Al termine di inquiry JABWT(Java APIs for Bluetooth Wirless Technology) prevedono la chiamata della funzione inquiryCompleted().

La ricerca effettiva dei servizi offerti dal RemoteDevice è attuata da serviceDiscovered(). Terminata l'operazione, viene invocato il metodo serviceSearchCompleted() il quale scarta le istanze di Remote, che non offrono il servizio di BlueOrganizer. All'istanze rimaste viene invece associata un URL(univoca per ogni Remote Device), relativa al servizio. Giunti a questo punto il Client attende istruzioni dall'utente su quale dispositivo connettersi. Anche il server, per stabilire la connessione, necessita della chiamata Connection.open(), che restituisce un'istanza di StreamConnection(), la quale consente di creare i canali di comunicazione nella medesima modalità del server. In questo caso, il parametro che viene passato a Connection.open() è l'URL relativa al device selezionato in precedenza. Ora client e server possono comunicare.

Comunicazione mediante il protocollo RFCOMM

Per iniziare un qualsiasi tipo di connessione bisogna effettuare la chiamata Connector.Open(). Questo metodo necessita di un parametro costituito da una stringa rappresentante l'identificativo del protocollo utilizzato dal dispositivo, con cui si desidera stabilire la connessione. Il protocollo utilizzato dai dispositivi bluetooth prende il nome di RFCOMM. Nelle API Java JSR-82 non è prevista nessuna interfaccia nè classe specifica, per utilizzare RFCOMM, ma riutilizza le interfacce di GFC(Generic Connection Framework). GFC specifica il seguente schema per la definizione dell'URL:

schema://target;parametri

Nel caso di BlueOrganizer la stringa assume la seguente forma:

- Server:

  btspp://localhost:N4I5IERT62CD49IN4I5IETRT62CD49I2;name=BlueOrganizer;authorize=true

- Client:

  btspp://09HF55YXO11E

L'URL utilizzata dal server è definita staticamente durante l'implementazione ed è riconoscibile per la presenza di “localhost”. La stringa utilizzata dal client è costruita dinamicamente in fase d'esecuzione del programma. L'URL del client è costruita, recuperando le informazioni necessarie sui servizi offerti dai device trovati nella fase di inquiry.

La chiamata Connection.open(), nel caso in cui riceve un parametro simile a

quello del server, restituisce un'istanza dell'oggetto di tipo StreamConnectionNotifier. La classe restituita rende disponibile il metodo bloccante accepAndOpen(), che attende una richiesta di connessione da parte del client.

Se Connection.open() riceve un parametro di tipo Client, restituisce un'istanza della StreamConnection, dove sono accessibili i metodi per la creazione dei canali di comunicazione.

3.1.2 Comunicazione tramite scambio di SMS(Short Message Service), utilizzando le Wireless Messaging API(WMA).

Le Wireless Messaging API (WMA)[7] consentono alle applicazioni J2ME l'utilizzo di Short Message Service. La versione che è utilizzata è la la 1.1 identificata dalla JSR120.

Per inviare e ricevere sms, è necessario stabilire una connessione utilizzando la chiamata Connection.open(), la quale restituisce un'istanza di MessageConnection. Il parametro utilizzato dalla funzione è come nel caso del bluetooth, una stringa che rispetta le specifiche di Generic Connection Framework(GFC). Nel caso degli sms, l'url riceve la seguente forma:

- Ricezione: sms://:7877
- Invio: sms://3488927999:7877

Dove sms notifica l'utilizzo di Short Message Service e 7877 identifica la porta utilizzata dalla connessione. Nel caso di invio messaggio, bisogna specificare anche il numero del destinatario.

L'applicazione deve utilizzare la medesima porta per ricevere e inviare messaggi.

Le specifiche WMA prevedono la possibilità di servirsi di messaggi di testo (TextMessage), o messaggi binari (BinaryMessagge).

TextMessage permette l'invio di semplici Stringhe. Invece BinaryMessage invia i messaggi, sotto forma di byte. Bisogna quindi trasformare l'oggetto che si desidera inviare in un array di byte.

BlueOrganizer utilizza solo BinaryMessage, in quanto vengono spediti degli oggetti composti da elementi diversi da stringhe.

Per inviare sms, dopo aver istanziato il tipo di messaggio da spedire, basta utilizzare il metodo della classe MessageConnectio send() che, necessita come parametro, il messaggio stesso.

In BlueOrganizer la parte relativa all'invio del messaggio è posizionata all'interno di un Thread. L'applicazione riesce così a svolgere altre operazioni durante l'invio di messaggi.

Per ricevere sms, l'applicazione implementa l'interfaccia MessageListener. Alla ricezione di un sms la piattaforma Java chiama il metodo notifyIncomingMessage, la quale si occupa della gestione del messaggio.

3.1.3 Confronto tra le modalità di comunicazione

Si possono notare differenze sostanziali tra i due tipi di comunicazione.

La comunicazione tramite bluetooth risulta essere gratuita, a differenza della trasmissione messaggi tramite sms, che ha un costo non quantificabile. Questa

spesa varia dal numero di messaggi scambiati dai due dispositivi e dalle compagnie telefoniche utilizzate dagli utenti. Il costo dello scambio dei messaggi grava sia sull'utente, che ha iniziato la comunicazione, sia sul destinatario.

Un'unica connessione bluetooth riesce ad attuare uno scambio di informazioni sufficiente per soddisfare la richiesta del client. La comunicazione sms costringe l'invio minimo di un messaggio da parte del client e uno da parte del server, fino a raggiungere una quantità massima stabilita dall'utente.

La comunicazione sms presenta come vantaggio la mancanza di vincoli di distanza, presenti, invece in una una connessione bluetooth.

3.2 Serializzazione degli oggetti

Per poter inviare messaggi di byte, utilizzati durante la comunicazione tra le applicazioni BlueOrganizer, è stato necessario servirsi della serializzazione degli oggetti interessati. La Java MicroEdition non supporta l'implementazione dell'interfaccia Serializable. E' stato così necessario creare due metodi persit() e resurect(). Il metodo persist(), avvalendosi delle classi ByteArrayOutputStream e DataOutputStream, trasforma l'oggetto in un array di byte. Al contrario, resurect() trasforma l'array di byte (parametro del metodo), nell'oggetto, utilizzando le classi parallele ByteArrayInputStream e DataInputStream.

3.3 Utilizzo di BlueOrganizer

In questa fase, viene analizzata la formazione e l'utilizzo, da parte dell'utente delle principali finestre grafiche, di BlueOrganizer.

3.3.1 Oggetti grafici J2ME

J2ME mette a disposizione degli oggetti grafici che permettono di costruire velocemente delle interfacce molto semplici e veloci. Ogni screen presente in BlueOrganizer, è rappresentato da una classe che estende l'oggetto Form() per interfacce, composte da vari elementi o con l'estensione dell'oggetto List(), per rappresentare una semplice lista di elementi.

All'interno di un Form e di una List, si possono inserire dei comandi (Command), che appaiono nella parte inferiore del display del dispositivo. Essi ricevono alcuni parametri, che determinano la modalità di visualizzazione sul display, come posizione, priorità ed etichetta. Nonostante queste specifiche, ogni dispositivo visualizza in modo differente la posizione dei comandi. Il programmatore è, così, obbligato a modificare queste impostazioni in base al dispositivo, che si intende adoperare. Per rendere utilizzabili i comandi, la classe deve implementare l'interfaccia CommandListener e il relativo metodo commandAction(), impiegato per gestire l'azione che il comando dovrà svolgere.

Il Form può essere composto da oggetti utilizzati per le azioni di input che l'utente dovrà svolgere. Esistono vari tipi di oggetti, di cui ci si può servire per questa operazione e variano in base alla tipologia di dato che l'utente deve inserire.

ChoiceGroup è utilizzato per visualizzare opzioni, a scelta. Questo oggetto permette all'utente di selezionare più scelte (check button), oppure di rendere attivo un solo campo (radio button).

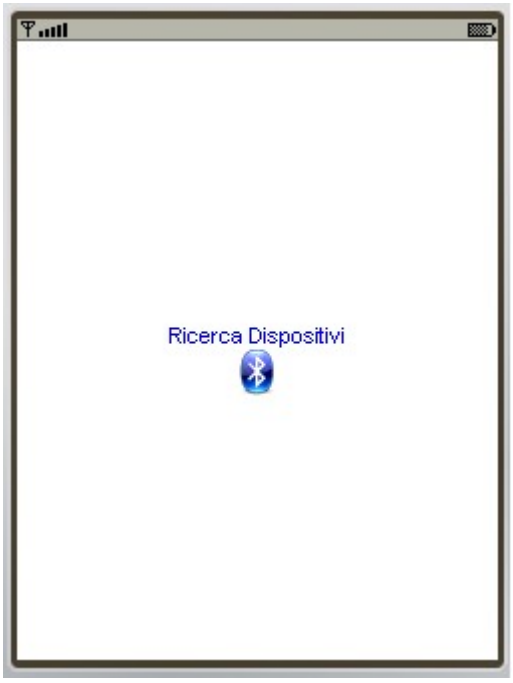
DateField permette la creazione di un'interfaccia utile all'inserimento di data e ore. Alcuni cellulari mostrano l'oggetto come un vero e proprio calendario, altri come

semplice formato di data gg/mm/aaaa.

Un'istanza di TextField crea un oggetto adatto all'inserimento di testo. E' previsto un filtro sui dati in entrata. Basta settare un apposito parametro e l'istanza dell'oggetto riceve in input, solo una specifica gamma di dati. Ad esempio solo numeri, password, url, numeri telofonici o nessun tipo di filtro.

La classe Gauge permette, invece, di visualizzare un grafico, su cui si può impostare il numero di colonne piene.

Le Canvas sono oggetti che consentono di creare elementi grafici di basso livello, dando la possibilità al programmatore di evadere dagli oggetti visti finora, creando, attraverso delle primitive, degli oggetti grafici personalizzati.



Screen 1: Canvas utilizzata in BlueOrganizer

Gli oggetti, finora descritti, sono stati utilizzati per la creazione degli screen di BlueOrganizer, con l'eccezione dell'oggetto Gauge. La classe Canvas è stata utilizzata solo per notificare all'utente, che il terminale è impegnato nella ricerca di dispositivi Bluetooth.

3.3.2 Finestra principale di BlueOraganizer

Screen 2: Vista Principale

La finestra principale di BlueOrganizer è formata da un campo DateField, rappresentante la data degli impegni, che l'utente desidera visualizzare. Gli impegni presenti vengono mostrati in una ChoiceGroup. Da questa finestra, è possibile accedere al menù, che offre alcune possibili azioni.

Accedendo al menù "Impostazioni", l'utente decide come organizzare la propria giornata, impostando l'ora d'inizio, fine e i giorni della settimana, in cui l'utente risulta disponibile per uno specifico impegno.

Il menù “Gestione Impegni” dà accesso al managing delle attività che non comportano l'accordo con persone esterne, tramite l'utilizzo di una contrattazione della data dell'impegno. Da qui, l'utente può aggiungere, modificare, eliminare impegni definiti di routine. Sono stati chiamati così, proprio per la caratteristica di presentarsi a intervalli costanti, durante un arco temporale, o aggiungere impegni a scadenza singola.

Selezionando il menù “Nuovo Impegno”, si passa alla schermata utilizzata per iniziare una nuova contrattazione di attività con un utente esterno. Quest'operazione verrà analizzata in dettaglio nel paragrafo successivo.

La voce “Rubrica” porta l'utente alle schermate utili alla gestione dei contatti.

Screen 3: Rubrica durante la selezione del partecipante all'impegno concordato

Azionando il comando “Visualizza”, viene mostrato sul display l'impegno selezionato nel ChoiceGroup, con la totalità dei propri attributi.

Visualizza Imp. Concordato **Visualizza Imp. Singolo o Imp. Routine**

Screen 4: Visualizzazione Impegni

Il comando “Attività annullate” viene mostrato all'utente, solo se l'applicazione riceve una notifica di un impegno concordato annullato. Questo menù permette di visualizzare, eliminare le attività che sono state annullate, oppure proporre un nuovo orario.

3.3.3 Invio di una richiesta di un impegno concordato

Lo screen utilizzato per iniziare la contrattazione dell'orario, comporta all'utente il settaggio di alcuni campi.

“Classe dell'attività” obbliga l'utente, attraverso un ChoiceGroup a scelta singola, a selezionare una categoria di cui fa parte l'impegno, che si vuole organizzare.

Il secondo parametro da inserire è il “Nome dell'attività”, e se l'utente desidera una breve descrizione dell'impegno. La stringa inserita non presenta nessuna restrizione (salvo la lunghezza del testo), nè in termini di univocità all'interno dell'applicazione, nè per tipo di dati inseriti.

Il campo “Data prevista” rappresenta il giorno in cui BlueOrganizer inizia la ricerca del tempo libero presente. “Ora prevista” è il campo associato a “Data prevista”, che stabilisce l'ora in cui la contrattazione è abilitata il primo giorno della ricerca. Nella contrattazione bluetooth l'applicazione cerca di collocare l'impegno in orari vicini a quello indicato.

“Modalità di contrattazione” stabilisce i termini in cui la contrattazione risulta valida, è presente un caso particolare in cui la modalità assume il nome di “Nessuna” che verrà analizzata più tardi.

“Durata attività” indica la durata dell'impegno. Campo che assume importanza è “Classe dell'attività correlate” (campo presente in ogni tipologia di impegno). E' composto da un ChoiceGroup con la possibilità di scelta multipla. Questo campo non viene inviato al destinatario. Infatti rappresenta le attività che in futuro potranno fare annullare automaticamente l'impegno (solo usando la modalità riportata sotto). Se l'utente non seleziona nessuna attività di questo campo, significa che l'impegno non è rinviabile, salvo dall'utente stesso. Il destinatario a impegno fissato può settare le proprie attività correlate.

L'applicazione elimina in maniera automatica gli impegni, solo se avviene una contrattazione svolta, come nell'esempio esposto nelle righe successive.

Screen 5: Crea Impegno Concordato

Esempio:

Chiamiamo“A” l'utente che inizierà la contrattazione, e “B” il destinatario.

A definisce un impegno da concordare con “Modalità di contrattazione” impostata uguale a “Nessuna”. “A” con questa impostazione vuole rendere valido l'impegno solo per la data e l'ora specificata. Il BlueOrganizer di “B” verifica se la richiesta può essere evasa. Nel caso in cui uno o più impegni risultano essere in conflitto con la richiesta, viene controllato se questi presentano tra le loro “attività correlate” la classe della richiesta ricevuta. In caso affermativo, viene convalidata la richiesta di “A” e gli impegni in conflitto spostati nelle “Contrattazioni

Annullate". Naturalmente se tra gli impegni in conflitto esistono altri impegni concordati viene inviata una notifica ai rispettivi destinatari.

3.3.4 Modifica di un impegno concordato

La modifica di un impegno concordato può comportare alcune difficoltà nel caso in cui la contrattazione sia stata effettuata tramite connessione Bluetooth. Nel caso in cui non fosse possibile comunicare con il dispositivo del destinatario, è stato deciso di impossibilitare sia la modifica dell'impegno, che la cancellazione.

Se la contrattazione è stata svolta tramite scambio di sms, il problema non sussiste, visto che la ricezione del messaggio di modifica o cancellazione dell'impegno può essere ricevuto in maniera asincrona.

Le considerazioni tratte da questo paragrafo sono da considerarsi valide anche nell'esempio analizzato nel paragrafo "3.3.3 Invio di una richiesta di un impegno concordato".

3.4 Limiti dello scheduler per agende e sviluppi futuri

L'applicazione BlueOrganizer, come ora strutturata presenta alcune limitazioni. Lo scheduler per impegni creato si pone al pubblico, solo come prototipo iniziale.

La società attuale impone ad ogni persona grande flessibilità e ritmi di vita sfrenati. Le relazioni tra gli individui, soprattutto in ambito professionale, richiedono continui spostamenti. Per effettuare tali movimenti, il soggetto adopera una quantità di tempo variabile, in relazione alla distanza del trasferimento e al mezzo utilizzato per compierlo.

Lo scheduler per agende, allo stato attuale, non è in grado di calcolare la quantità di tempo necessaria all'utente per spostarsi sul luogo dove avviene un impegno. Di conseguenza non riesce neppure a determinare la quantità di tempo utile al trasferimento dal luogo di un impegno ad un altro.

Lo scheduler è programmato in modo da evitare l'accavallamento degli impegni, inserendo una quantità di tempo tra un impegno e l'altro decisa dall'utente.

Apportando alcune modifiche è possibile raggirare l'ostacolo.

Tramite la creazione di una tabella contenente il tempo necessario per lo spostamento da un luogo A ad un luogo B, si potrebbe risolvere il problema. Lo scheduler, prima di fissare l'impegno, verifica nella tabella se tutti e due gli utenti interessati dispongono del tempo minimo per raggiungere il luogo destinato all'incontro.

I dispositivi mobili, per cui è stata creata l'applicazione, presentano limitazioni che cambiano in base al modello utilizzato. Tra queste restrizioni è presente anche la dimensione, espressa in kilobyte, delle applicazioni che il device ha la capacità di installare.

La tabella avrebbe dimensioni non standardizzabili introducendo un problema di portabilità del software. Inoltre la tabella non riuscirebbe mai a ricoprire tutti gli spostamenti che un individuo può compiere. Nonostante ciò, l'introduzione di questa tabella potrebbe essere utile per l'uso di BlueOrganizer in piccole comunità . Verrà proposto nel paragrafo successivo un possibile scenario di utilizzo.

Un'altra possibile soluzione prevede l'utilizzo di una tecnologia presente sulla maggior parte di cellulari e PDA che prende il nome di GPRS. La tecnologia

GPRS (General Packet Radio Service) permette di inviare e ricevere informazioni attraverso un network telefonico. In questo caso non si parla piu' di commutazione di circuito (tipico della trasmissione voce), ma di commutazione di pacchetto (stessa logica utilizzata per la trasmissione dati su Internet). Il GPRS facilita le connessioni istantanee, perchè l'informazione può essere mandata o ricevuta immediatamente appena se ne ha bisogno; infatti i terminali GPRS vengono identificati come sempre on line. Le caratteristiche principali che contraddistinguono il GPRS sono :

- una discreta velocità di trasmissione dati ;
- trasmissione dati basata sulla commutazione a pacchetto;
- connettività always on (sempre aperta);
- possibilità di accesso ai servizi internet;
- tariffazione costi per quantità dati.

L'applicazione avrebbe la possibilità di richiedere ad un servizio web il tempo impiegato per il trasferimento dell'utente da una zona all'altra. Il servizio utilizzato potrebbe essere creato ad hoc per BlueOrganizer, o avvalersi di un servizio già esistente. Questo tipo di soluzione introdurrebbe nuovi costi per l'utente, ma risulta essere ottimale per eliminare il problema di calcolo del tempo di spostamento.

3.5 Possibili scenari di utilizzo di BlueOrganizer

In questo paragrafo si vuole mostrare possibili situazioni dove è applicabile l'utilizzo di BlueOrganizer. Verranno esposti esempi riguardanti lo stato attuale

dello scheduler per agende, che richiedono le modifiche esposte nel paragrafo precedente.

3.5.1 Scenario di utilizzo allo stato attuale di BlueOrganizer

Il sistema, allo stato attuale, può essere contestualizzato in diversi ambiti. Nonostante il problema, analizzato nel paragrafo precedente, del calcolo della quantità di tempo necessario all'utente per il trasferimento dal luogo dell'impegno A al luogo dell'impegno B, BlueOrgaizer potrebbe già essere utilizzata nei casi dove esiste uno scenario statico. In tale situazione gli utenti, o almeno colui che assumerebbe la figura principale di utilizzatore del software, rimane in un ambiente limitato. Per esempio, lo si potrebbe usare in uno studio dentistico, o in un qualsiasi luogo dove viene offerta una prestazione. I clienti potrebbero utilizzare BlueOrganizer per prendere appuntamento con lo studio preso in considerazione.

3.5.2 Futuri scenari di utilizzo di BlueOrganizer

Questa sezione è composta dall'illustrazione di due esempi, che utilizzano le soluzioni proposte nel paragrafo 3.4 al problema del calcolo del tempo necessario, al trasferimento di un individuo da un luogo ad un altro.

Soluzione che utilizza la tabella delle distanze

Adottando la soluzione, che proponeva l'uso della tabella delle distanze, descritta nel paragrafo precedente, BlueOrganizer potrebbe essere utilizzato in piccole comunità, dove gli spostamenti risultano essere ridotti e ripetitivi. Di conseguenza la tabella assumerebbe dimensioni supportabili dalla maggior parte di cellulari e

PDA.

Esempio di comunità potrebbe essere un'università. Un docente avrebbe la possibilità di organizzare le proprie riunioni con docenti, studenti o personale vario, in modo automatizzato. Si verrebbe a creare una situazione vantaggiosa per entrambi i soggetti.

Nella figura N viene riportata una possibile tabella delle distanze adattata ad un eventuale utilizzo di BlueOrganizer nell'Università di Scienze Naturali Matematiche e Fisiche di Trento.

Luogo A	Luogo B	Tempo Necessario (min)
I e II padiglione IRST	Sede Principale	5
I e II padiglione IRST	III Padiglione IRST	2
III padiglione IRST	Sede Principale	5

Tabella 5: Tabella dei Tempi - Università Scienze Naturali, Matematiche e fisiche

Soluzione che utilizza un Servizio Web

La soluzione impiega una connessione GPRS (General Packet Radio Service), per accedere ad un Web Service capace di calcolare il tempo impiegato per il trasferimento dell'utente da una zona all'altra. La soluzione permetterebbe all'applicazione di essere adattata ad un qualsiasi scenario. Sarebbe l'utente a scegliere, in quale situazione sia valido usufruire del servizio di agenda offerto da BlueOrganizer.

Conclusioni

Il lavoro svolto in questa tesi ha come risultato finale un'applicazione denominata BlueOrganizer. Il software prodotto è un agenda con una particolare innovazione. BlueOrganizer dà la possibilità di accordare in maniera automatica la data e l'ora di un impegno tra due persone entro un limite di tempo prestabilito. Il software è stato pensato per essere utilizzato da dispositivi mobili (cellulari e PDA), che supportano la piattaforma Java. La scelta è da ricondursi alla loro elevata diffusione. Le ridotte dimensioni dei dispositivi in questione permettono all'utente di avere l'agenda sempre con sé. Il linguaggio di programmazione è stato utilizzato per la sua portabilità su dispositivi con architetture diverse.

L'applicazione, per poter organizzare gli impegni utilizza un apposito scheduler per agende ed un sistema di comunicazione, basato sullo scambio di messaggi, tramite connessione Bluetooth o in alternativa SMS. Lo scheduler per agende, è stato pensato durante la fase di analisi, in modo da soddisfare le esigenze di diversi stili di vita. BlueOrganizer divide gli impegni in differenti attività che variano in base alla natura dell'impegno (Lavoro, Attività sportiva...). L'utente stabilisce in quali ore e giorni della settimana un'attività può essere sottoposta a contrattazione. L'applicazione prevede l'inserimento di tre tipologie di impegno. Una che utilizza la fase di contrattazione, le rimanenti sono utilizzate per annotare impegni a carattere personale, con scadenza singola o che si ripetono a intervalli costanti all'interno di uno spazio temporale.

L'applicazione non permette la presenza di impegni sovrapposti. Lo scheduler è programmato in modo da consentire, durante la contrattazione dell'orario di

svolgimento dell'impegno, di eliminare impegni con priorità inferiore. In caso di modifica o cancellazione, da parte dell'utente o dell'applicazione stessa, di un impegno che prevede la contrattazione viene garantita la sincronizzazione dei dati su entrambi i dispositivi interessati.

BlueOrganizer presenta una limitazione. Lo scheduler non calcola il tempo che l'utente impiega a spostarsi dal luogo dell'impegno precedente a quello successivo. L'applicazione allo stato attuale permette di settare una quantità di tempo a scelta dell'utente da inserire tra i due impegni. All'interno di questa tesi sono state proposte alcune soluzioni, per ovviare a questo problema.

La prima prevede l'utilizzo di una tabella contenente il tempo necessario, che l'utente impiega a raggiungere la località interessata. La soluzione risulta attuabile solo nel caso di utilizzo dell'applicazione in piccole comunità, dove i movimenti risultano essere limitati e ripetitivi. L'altra soluzione proposta si avvale di un web service, capace di calcolare i tempi necessari a compiere le distanze richieste. Per accedere al servizio web i dispositivi devono essere dotati di GPRS. La connessione GPRS introdurrebbe nuovi costi, che gli utenti dovrebbero sostenere, ma renderebbe BlueOrganizer adatto a qualsiasi scenario di utilizzo.

Bibliografia

[1] Definizione di agenda di Wikipedia http://it.wikipedia.org/wiki/Agenda

[2] Mondo Google, non solo motore di ricerca – Jackson Libri

[3] J2ME - Guida pratica alla programmazione di dispositivi wireless John W. Muchow The McGraw-Hill Companies, S.r.l.

[4] Qusay H. Mahmoud [2002] Learning Wireless Java

[5] Sito degli sviluppatori Sun dedicato alle specifiche delle JSR

http://java.sun.com/javame/reference/apis.jsp

[6] Guida a JavaMicroEdition offerta dal sito javastaff.com http://j2me.javastaff.com/

[7] C.Erique Ortiz [Ottobre 2005] The Wireless Messaging API 2.0

http://developers.sun.com/mobility/midp/articles/wma2/

[8] Sito ufficiale della tecnologia Bluetooth http://www.bluetooth.com

[9] Qusay H.Mahmoud [Aprile 2003] The Java APIs for Bluetooth Wireless Technology http://developers.sun.com/mobility/midp/articles/bluetooth2/

[10] C.Erique Ortiz [Dicembre 2004] Using the Java APIs for Bluetooth Wireless Technology
http://developers.sun.com/mobility/apis/articles/bluetoothintro/index.html

[11] Sito che offre un'ampia panoramica su Short Message Service (SMS) http://www.developershome.com/sms/

[12] Forum ufficiale Nokia: Tools and SDK

http://wiki.forum.nokia.com/index.php/Portal:Tools_and_SDK

[13] Forum ufficiale Nokia: Java Security Domains

http://wiki.forum.nokia.com/index.php/Java_Security_Domains