CONSIGLIO NAZIONALE DELLE RICERCHE

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE

Contratto di collaborazione NES s.r.l. – ISTI/CNR

TECNOLOGIE DEI SISTEMI DI HOME AUTOMATION
-Caso di studio: Progetto del controllo remoto basato su WAP

Luca Tarrini, Rolando Bianchi Bandinelli, Graziano Bertini

Nota Tecnica B4 - Dicembre 2002

Contratto di collaborazione NES s.r.l. – ISTI/CNR

TECNOLOGIE DEI SISTEMI DI HOME AUTOMATION
Caso di studio: Progetto del controllo remoto basato su WAP

Luca Tarrini*, Rolando Bianchi Bandinelli, Graziano Bertini

*Associato alla ricerca

Sommario

Nell'ambito di un contratto conto/terzi dell'ISTI-CNR con la soc. NES Srl Pisa, una parte del lavoro ha riguardato una panoramica dello stato attuale delle tecnologie dei sistemi per l'home automation ed uno studio sulla possibilità di progettare delle funzioni aggiuntive di controllo remoto tramite cellulari o palmari per l'esecuzione di comandi e richieste sullo stato di dispositivi nell'ambiente domestico. Uno degli scopi può essere quello di tenere sotto osservazione in qualsiasi momento persone disabili e anziani che abitano da soli o in apposite residenze ed in generale poter operare anche in ambito commerciale sulle gestioni di magazzini ecc. essendo l'utente in movimento.

Dapprima si da un cenno alle possibilità offerte in generale dall'home networking, mettendo in risalto il concetto di *pervasive computing* dove ognuno può accedere all'informazione sia dall'interno che dall'esterno degli ambienti, ed in particolare viene sviluppata una proposta basata su protocollo WAP disponibile su telefoni cellulari.

Per completezza dell'esposizione sono stati anche descritti i principali protocolli che meglio si adattano alla realizzazione di una rete domotica, presentando le loro principali caratteristiche.

Successivamente viene quindi descritto il protocollo WAP, utilizzato sui telefonini mobili per connettersi ad Internet, individuando le principali problematiche che si devono affrontare, (CPU poco potenti, poca memoria, piccoli display e restrizioni di consumo di batteria). Tra i vari ambienti di sviluppo per applicazioni WAP messi a disposizione dai principali operatori presenti nel panorama delle telefonia mobile, ne abbiamo scelto uno (il Nokia) sul quale è stata sviluppata la nostra applicazione. In generale questi toolkit rappresentano un browser capace di interpretare il codice WML in modo da realizzare simulazioni il più possibile vicino alla realtà.

Sono inoltre state descritte le modalità di comunicazione sull'interfaccia fra l'ambiente da controllare e l'utente remoto descrivendo la tecnologia .NET che verrà utilizzata per conseguire i vari obbiettivi e come integrarla nell'ambiente domotico.

Relativamente alla rete locale domestica è descritto come sono controllati gli attuatori e sensori nel caso di impiego del protocollo X10.

1. Introduzione

Con il termine *Home Automation* (dall'inglese oppure termine francese *Domotique*), si intende l'insieme delle tecnologie e delle tecniche di progettazione che sono alla base dell'automazione domestica applicabile però a qualsiasi altro ambiente, sia di tipo residenziale che commerciale o industriale.

In effetti il termine "Intelligente" è già divenuto di uso abbastanza comune per indicare un ambiente opportunamente attrezzato al fine di rendere più agevoli le attività al suo interno (1). Nel caso di un'abitazione si possono citare l'apertura di tapparelle, gestione riscaldamento, accensione luci, attivazione elettrodomestici, ecc.), di aumentare la sicurezza di chi vi abita (controllo fughe di gas, incendi, allagamenti, anti-intrusione, ecc.) e di consentire la connessione a distanza con innovativi servizi di assistenza (quali tele-soccorso, tele-monitoraggio, tele-medicina, ecc.) e con altri servizi di utilità generale (tele-shopping, *home-banking*, servizi informativi, ecc...)..

Occorre rilevare inoltre che negli ultimi anni si è verificato un forte sviluppo delle nuove tecnologie digitali che ha prodotto un notevole incremento delle loro capacità in svariati settori ed una continua riduzione dei costi, consentendo così una loro diffusione anche in ambienti di tipo domestico: molti sono i prodotti ormai largamente diffusi all'interno delle abitazioni che utilizzano queste tecnologie nell'ambito della comunicazione, dell'intrattenimento, della formazione, del lavoro. Fra questi si distinguono i PC (sia di tipo desktop che portatili o palmari, di dimensioni ridotte e con diversi livelli di funzionalità), i prodotti "tradizionali dell'elettronica di intrattenimento (televisori, videoregistratori, videocamere, HI-FI, lettori DVD, console per videogiochi), gli apparecchi per la comunicazione (telefoni normali e cellulari, *smart phones*), i dispositivi di controllo (impianti di allarme e di video sorveglianza, impianti di riscaldamento e condizionamento, sistemi di controllo dell'illuminazione).

Attualmente la casa si presenta come un ambiente eterogeneo di terminali connessi a reti indipendenti, ovvero non intercomunicanti, come ad esempio quella elettrica, quella telefonica, quella della TV via cavo, quella dei telefoni *cordless*; ognuna di queste possiede una differente interfaccia, un differente protocollo, supporta un diverso mezzo trasmissivo, diverse velocità, e differenti applicazioni. La maggior parte dei dispositivi funzionano indipendentemente l'uno dall'altro, anche se sul mercato cominciano ad apparire una serie di dispositivi definiti di "convergenza": in un unico dispositivo confluiscono aspetti e funzioni differenti con l'obiettivo finale di rendere più agevole e immediata l'interazione e lo scambio di dati e di servizi, come ad esempio telefoni cellulari che possono diventare terminali Internet o frigoriferi che possono essere utilizzati anche per inviare video-messaggi ai membri della famiglia.

A completare lo scenario domestico, l'utilizzo dei diversi servizi resi disponibili da Internet, che vanno dal "*web browsing*" alla posta elettronica e alla *chat*, dal trasferimento di file (anche con contenuto multimediale) all'acquisto di beni e servizi per via telematica, ha introdotto nuove esigenze di comunicazione accanto a quelle tradizionali.

Poiché essere connessi alla Rete diventa quindi un elemento di forte interesse per il consumatore, accanto ai dispositivi citati, si prospetta la diffusione di tutta una nuova categoria di prodotti che permettano la fruizione dell'informazione in maniera semplificata o l'utilizzo di nuovi servizi. Si tratta delle cosiddette *"information appliance"*, dispositivi dedicati all'esecuzione di compiti specifici in maniera ottimizzata e disegnati per accedere all'informazione su Internet. Esempi di prodotti

appartenenti a questa categoria sono i lettori MP3, i *Personal Digital Assistant* (PDA), i *Web phones*. Tali dispositivi hanno la velocità e la potenza di un personal computer, ma sono più facili da utilizzare e da gestire, hanno una affidabilità più alta e costano di meno. L'ottica è quella di spostare le funzionalità computazionali di cui ha bisogno l'utente domestico, dai classici PC, disegnati come strumenti di supporto alla produttività, verso dispositivi orientati all'esecuzione dei loro compiti specifici in maniera semplice, versatile e piacevole per l'utente.

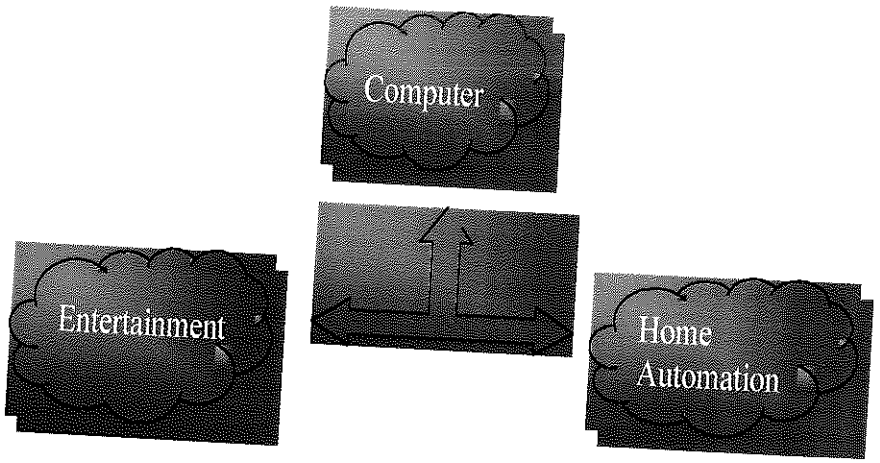
Le nuove potenzialità di comunicazione offerte da Internet e dalle moderne reti di trasporto fisse e mobili, consentono lo sviluppo di tutta una serie di prodotti *intelligenti* per la casa, nei quali funzionalità di tipo tradizionale vengono estese ed integrate con capacità di comunicazione e di elaborazione avanzate, al fine di estendere i servizi tradizionali offerti dai dispositivi stessi, "mascherando" però all'utente ogni complicazione tecnologica che possa derivare dall'installazione, dall'utilizzo o dalla manutenzione di tale sistema

2. Possibilità offerte dall'Home Networking

Il concetto di *home networking* si riferisce all'interconnessione di diversi dispositivi elettronici utilizzati in ambito domestico (2), in modo da realizzare una integrazione delle loro funzionalità attraverso una rete domestica che consenta agli apparati di interagire l'uno con l'altro e con le reti esterne (ad esempio Internet).

Il fine è quello di integrare tutti i servizi e le applicazioni presenti nell'universo domestico, rendendo quindi interoperabili, tendenzialmente, tre categorie diverse di dispositivi ognuna caratterizzata da esigenze computazionali e trasmissive diverse, quali:

- i dispositivi come PC, stampanti, scanner, fax, smart phones etc. *(Categoria Computer)*
- i dispositivi utilizzati per il gioco e l'intrattenimento in generale, quali TV, HI-FI videoregistratori, lettori DVD, lettori CD, console per videogiochi etc. *(Categoria Entertainment)*
- i dispositivi utilizzati nella gestione dell'ambiente domestico quali sistemi di allarme, luci, sistemi di condizionamento, elettrodomestici etc. *(Categoria Home Automation)*.



3. Le reti domotiche

Esistono attualmente numerosi (troppi!) standard utilizzati nel campo dell'automazione domestica ed uno dei fattori che ha contribuito a rallentarne la crescita è proprio l' incertezza su quale di essi prevarrà e diventerà lo standard di mercato. Diamo qui un cenno alle caratteristiche dei principali standard presenti sul mercato, rimandando ai rispettivi manuali di riferimento per i particolari di ciascuna tipologia.

Architettura di base

Gli elementi che costituiscono l'infrastruttura di trasporto dell'informazione sono essenzialmente tre (vedi Figura 1):

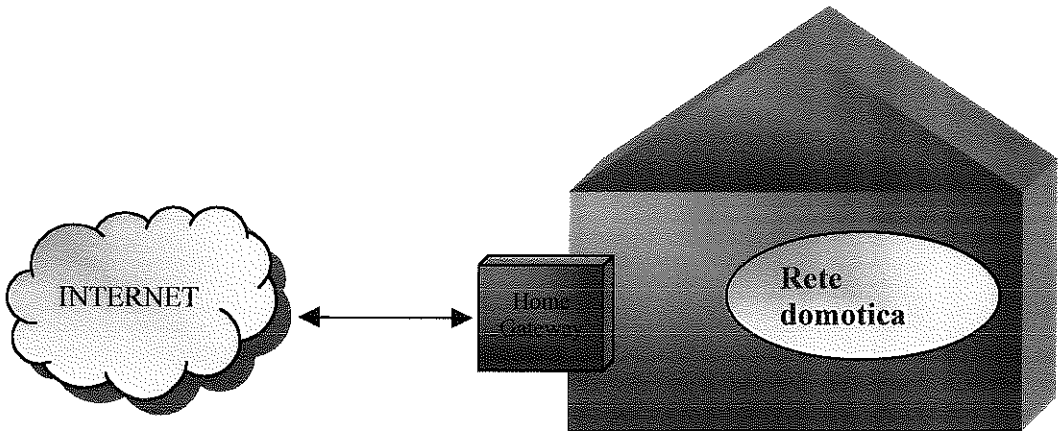


Figura .1 Architettura di base

a) la connessione a banda larga che interconnette direttamente la casa con i possibili fornitori di servizi;

b) il *residential gateway* che rappresenta l'interfaccia fra la rete esterna (WAN) e quella interna alla casa (LAN);

c) la rete domotica a cui sono direttamente connessi tutti i dispositivi presenti nella casa.

a) Accesso a Internet

Oggi l'accesso a Internet più comune è realizzato prevalentemente attraverso modem analogici che trasmettono su PSTN attraverso le linee POTS; la capacità trasmissiva disponibile in questo caso (56 Kbps) risulta quindi limitata e per lo più insufficiente a garantire le necessità delle applicazioni in uno scenario di home networking.

La connessione deve quindi essere realizzata sfruttando tecnologie a banda larga in grado di supportare il traffico voce, dati, video e di controllo proveniente e diretto verso la casa, quali ad esempio quella satellitare, ISDN (*Integrated Service Digital Network*), xDSL (*Data Subscriber Line*), quella basata su linee elettriche DPL (*Digital Powerline*), quella dei "cable modem" e un domani eventualmente quella wireless di III generazione (GPRS, UMTS).

Tecnologie più promettenti per l'home networking sembrano invece essere quelle definite "no-new-wires": queste possono essere distinte in base alla rete che supporta il trasporto dell'informazione per ognuna di esse e sono costituite dalla soluzione phoneline (che utilizza la struttura di cavi telefonici), da quella powerline (che utilizza i cavi per il trasporto dell'elettricità) e da quella wireless (che utilizza il mezzo radio). Nel seguito vengono presentate le caratteristiche di ognuna di queste e viene realizzato un confronto fra di esse.

La tecnologia "phoneline"

Questa tecnologia si basa sulla trasmissione dell'informazione attraverso la rete costituita dai normali doppini telefonici presenti all'interno della casa utilizzata per il trasporto del traffico voce. La trasmissione dell'informazione è però realizzata utilizzando una banda di frequenze superiore rispetto a quella riservata al servizio telefonico o ai servizi dati e permettendo in questo modo la realizzazione sullo stesso circuito di più trasmissioni contemporanee dedicate a servizi diversi, senza interferenze fra queste e senza degradazione della qualità delle trasmissioni stesse.

Le linee telefoniche attuali, costruite essenzialmente per il segnale vocale e ottimizzate per tale scopo, non sono adatte per trasmissione dati ad alta velocità. Le loro caratteristiche di impedenza e attenuazione non possono essere ben controllate. La prima compagnia che ha cercato di risolvere questi problemi è la Tut Systems Inc. di Pleasant Hill, California.

Essa impiega un multiplexing a divisione di frequenza FDM (vedi Figura 2.2) per creare tre canali differenziati, attraverso l'utilizzo di filtri selettivi, ognuno dei quali con un differente scopo: servizio telefonico normale (dc – 3400 Hz), segnale ADSL per il collegamento ad Internet (da 25 KHz a 1.1 MHz), networking della casa (5.5 – 9.5 MHz).

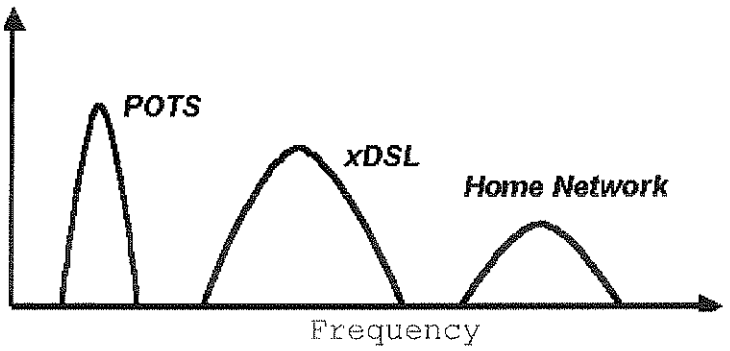


Fig. 2: multiplexing a divisione di frequenza FDM

Relativamente a questa tecnologia, l'attività di standardizzazione è condotta dalla *Home Phoneline Networking Alliance (HomePNA)* [WEB03] che ha pubblicato due specifiche di standard aperti per la comunicazione su linee telefoniche rispettivamente denominate 1.0 e 2.0.

La tecnologia "wireless"

Questa tecnologia [WEB13] si basa sulla trasmissione dell'informazione attraverso onde elettromagnetiche (radio o infrarossi) nell'etere. I dispositivi sono in questo modo

Le due varianti dell'xDSL più promettenti per l'home networking sono l'ADSL ed una sua variante a più bassa velocità denominata G.lite (1,5 Mbps *downstream*, 512 Kbps *upstream*). Quest'ultima tecnologia elimina la necessità di installare presso l'abitazione dell'utente l'elemento che provveda alla separazione del traffico dati e presenta costi di installazione ridotti rispetto all'ADSL.

b) Residential Gateway

Per quanto riguarda invece il *residential gateway*, chiamato anche Home Network Service Point *(HNSP)*, questo rappresenta l'interfaccia di accesso centralizzato fra le reti esterne e quelle interne alla casa, rappresentando quindi un importante anello di collegamento.

Il *gateway* rappresenta la piattaforma attraverso la quale i fornitori di servizi possono implementare da remoto, attraverso Internet, i loro servizi direttamente all'interno della casa, utilizzandolo inoltre per realizzare la gestione e le funzionalità di amministrazione dei servizi stessi.

Da un altro punto di vista il *gateway* è l'elemento che consente ai diversi dispositivi di usufruire simultaneamente di una connessione a larga banda verso Internet. Esso inoltre rappresenta l'elemento che realizza l'integrazione di eventuali reti e protocolli differenti funzionanti all'interno della casa, assicurando la loro interoperabilità.

Può inoltre implementare funzionalità di controllo degli accessi alla rete domestica, assicurando la sicurezza e la riservatezza del sistema domestico e dei servizi forniti, ed impedendo l'utilizzo non autorizzato delle risorse. In alcuni casi il gateway realizza anche funzionalità di gestione del traffico interno alla rete domestica, permettendo in questo modo ai dispositivi connessi di lavorare alla loro massima efficienza.

c) Rete domotica

Relativamente alla rete domestica utilizzata per interconnettere i dispositivi, questa deve soddisfarediversi requisiti generali di cui si parlerà nel seguito.

Una grande varietà di tecnologie risulta oggi a disposizione o in fase di realizzazione per soddisfare le esigenze di comunicazione all'interno della casa: solo alcune di esse sembrano invece possedere requisiti di versatilità tali da poter rappresentare una soluzione generale al problema dell'interconnessione.

3.1 Le tecnologie di trasporto

Nell'insieme delle possibilità offerte per il trasporto dell'informazione all'interno della casa, una prima distinzione può essere fatta fra le soluzioni che richiedono la realizzazione di una nuova infrastruttura di cavi (tecnologie definite "new-wires") che raggiunga ogni angolo della casa, e quelle che invece non prevedono interventi all'interno dell'abitazione (tecnologie definite "no-new-wires") ma utilizzino le reti già presenti, come ad esempio quella elettrica o quella telefonica, o in alternativa reti wireless.

Alla prima categoria appartengono le tecnologie USB, 10base-T/Cat5 Ethernet e IEEE 1394 "Firewire". Benché questi sistemi rappresentino un ambiente di comunicazione sperimentato, robusto, affidabile e ad alta velocità (fino a 100 Mbps), non sembrano possedere le caratteristiche per una loro larga adozione sul mercato implicando una significativa complicazione tecnologica e richiedendo interventi strutturali per il cablaggio della casa, con un conseguente innalzamento dei costi.

interconnessi a formare una wireless LAN *(WLAN)* senza l'utilizzo di cavi o di altre connessioni fisiche.

Una prima possibile topologia di rete *(centralized mode)* di una *WLAN* prevede la presenza di "access point" (AP) collegati ad una rete fissa di trasporto e che funzionano come trasmettitori e ricevitori per i terminabili mobili serviti. Tipicamente ogni AP gestisce un numero variabile di utenti (fino a 50, dipendentemente dalla tecnologia utilizzata) entro un raggio d'azione definito (fino a 300 metri, dipendentemente anche in questo caso dalla tecnologia utilizzata).

La copertura radio di zone più ampie è realizzata attraverso la sovrapposizione delle microcelle (in maniera simile a quanto realizzato su area geografica da un sistema radiomobile cellulare) utilizzando più AP e garantendo la possibilità per i dispositivi di migrare da una cella ad un'altra adiacente attraverso l'implementazione di funzionalità di roaming. Al fine di stabilire la connessione con i dispositivi presenti all'interno della casa, questi sono dotati di opportuni "WLAN adapter" realizzano l'interfaccia fra il *Network Operating System (NOS)* e l'etere, permettendo al dispositivo di trasmettere e ricevere i dati sul canale radio attraverso un'antenna. La comunicazione fra i dispositivi wireless è in questo caso mediata dagli AP.

Una differente topologia *(direct mode)* prevede invece la possibilità per i dispositivi dotati di "WLAN adapter" di realizzare direttamente fra di loro una comunicazione *peer to peer* senza la mediazione di AP, o eventualmente relegando a questi il solo compito di assegnare le frequenze da utilizzare per la trasmissione radio.

L'utilizzo di WLAN per garantire la connettività dei dispositivi presenti nella casa al fine di realizzare uno scenario integrato di home networking, sembra la più adatta a rispondere ai requisiti per applicazioni di questo tipo. I vantaggi principali risiedono nella possibilità di poter posizionare in ogni angolo della casa i dispositivi da interconnettere, in maniera non vincolata a cablaggi ed eventualmente potendo spostare i dispositivi stessi nei punti dove si renda opportuno di volta in volta il loro utilizzo.

Le velocità di trasmissione realizzate (fino a 54 Mbps) rendono la tecnologia *wireless* confrontabile in termini di throughput rispetto a soluzioni tradizionali di tipo *wired* (ad esempio Ethernet). Tuttavia, al fine di permettere un funzionamento efficiente della tecnologia wireless e quindi una sua larga adozione da parte del mercato si dovrà garantire la risoluzione del problema dei cammini multipli (che si rende particolarmente sensibile in un ambiente domestico) che riduce sensibilmente la banda disponibile e le possibili problematiche di interferenze dovute all'utilizzo della banda non sottoposta a licenza.

3.2 Protocolli Home Automation

Dopo aver visto i possibili mezzi fisici da utilizzare vediamo i vari protocolli sul mercato della domotica.

Echelon

La compagnia Echelon di Polo Alto, in California, è stata costituita nel 1988 e produce gli strumenti necessari per creare una rete chiamata LON (Local Operatine Network), fornita di apposito protocollo chiamato Lon Talk. Il network con il suo supporto di hardware e software si chiama LONWORKS.

LonWorks è un protocollo di comunicazione peer-to-peer. Come per i protocolli Internet e Ethernet, esso segue le linea guida della architettura OSI.

Il principio di funzionamento del LON consiste in un' unica interfaccia dove tutti i collegamenti sono legati al network. L'interfaccia si chiama Neuron Chip circuito integrato costituito secondo le specifiche Echelon, e commercializzata da Motorola e Toshiba. Il network Lon Works è usato molto nell'automazione di edifici industriali, aeroporti ecc. e ultimamente è proposto anche per le case e costruzioni civili in generale.

I network Lon Talk forniscono una vasta gamma di mezzi per la trasmissione. I nodi, come elettrodomestici, interruttori e sensori, dotati di un proprio hardware o firmware possono essere collegati ad ogni mezzo di trasmissione (energia elettrica, frequenze radio e cavi twisted-pair).

I segnali tra i vari nodi viaggiano ad una velocità che varia da circa 4kbps per linee che trasmettono energia elettrica a 1,25 Mbps per doppino di lunghezza limitata. Ci sono ditte che offrono trasduttori per cavi coassiali e fibre ottiche. Lon Talk fornisce network separati a seconda che il mezzo di trasmissione sia energia elettrica o radio. Nel Lon Talk ogni network logico è chiamato domain (sfera o ramo). Lon Talk provvede alla radiodiffusione indirizzata a ciascun nodo presente in un domain o in una rete subordinata. È ammessa anche una trasmissione di gruppo. Un nodo può essere parte di gruppi multipli.

Ciascun nodo di un network LON è programmato in modo da comunicare specifici dati interni ad uno o più nodi. Questi rapporti sono emanati come risultato di un cambiamento di stato o di eventi programmati. Gli indirizzi dei destinatari di un cambio di stato in Echelon sono codificati in un quadro interno del Chip Neuron dallo strumento di configurazione.

Un esempio tipico è il termostato che segnala all'unità di controllo della caldaia quando la temperatura diminuisce più del limite. I dati riportati sono stati formattati nelle variabili definite nel network Echelon, chiamato *Standard Network Variable Types* (SNVTs).

Echelon ha sviluppato una serie di componenti per la creazione di un network LonWorks. Tra questi componenti c'è una collezione di utensili chiamata LonBuilder

Una LonBuilder Development Station (una stazione di sviluppo LonBuilder) è pensata per la creazione di apparecchi interfacciati, che su ordinazione, comunicheranno usando il protocollo LonTalk per mezzo del network LON. Questo permette ad ogni utente di selezionare un mezzo di comunicazione e di programmare i Neurons per specifiche applicazioni, di verificare e rimuovere errori dagli apparecchi, di configurare un network, e di analizzarne il funzionamento.

Un software su PC provvede a programmare tutti i moduli inseriti nel network. Dopo che l'applicazione di un network è stata progettata sul LonBuilder Development Station, gli effettivi nodi possono essere programmati usando un programmatore LonBuilder connesso ad un PC.in uno Neuron Chip usando un Neuron C, un comune linguaggio di programmazione C. In ogni Neuron Chip può essere contenuto un software in aggiunta al protocollo LonTalk. Le applicazioni troppo grandi per un Neuron Chip possono essere programmate in un processore esterno che permette al Neuron firmware di alternarsi con un LonBuilder Microprocessor Interface Program (programma di

interfacciamento del microprocessore LonBuilder). Queste opzioni sono illustrate in figura.

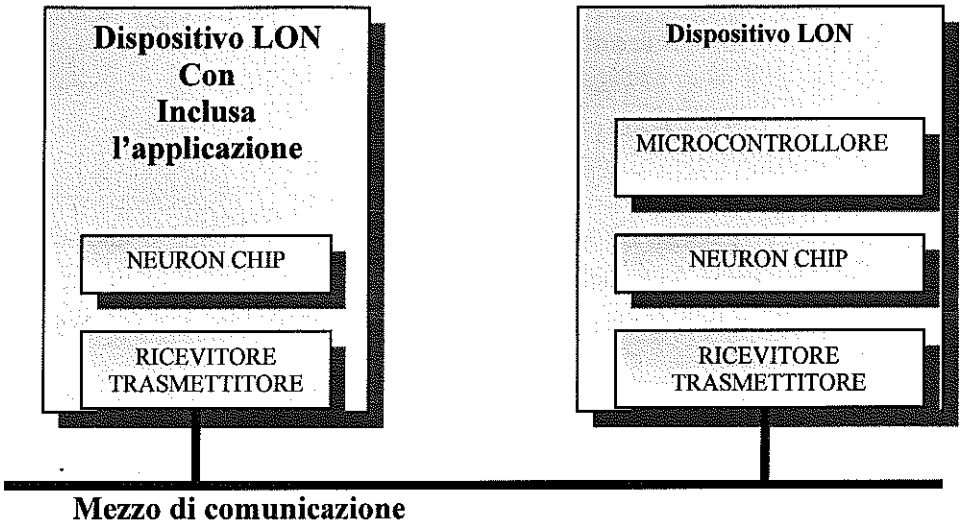


Fig.3: Neuron con funzioni supplementari

Una volta che il Neuron e qualsiasi altro microprocessore o PC associato, sono stati programmati, il network è configurato con gli strumenti del LonManager. LonManager offre una vasta gamma di programmi applicativi per PC con sistema DOS e WINDOWS.

Il sistema X-10

La tecnologia X10, che ha rappresentato in qualche modo l'avanguardia dei sistemi di trasmissione per l'home networking essendo sul mercato da più di 20 anni, è largamente usato negli Stati Uniti. Le caratteristiche principali sono la sua semplicità e i bassi costi (per altri dettagli vedi anche (3, 6) nota tecnica ISTI-CNR B4 21, dic. 2002).

La produzione è effettuata prevalentemente dall'azienda X-10, ma esistono altre aziende che sviluppano prodotti con questo standard.

X-10 è un protocollo di comunicazione per il controllo di apparecchiature elettriche e il mezzo di comunicazione è la linea elettrica. Permette la trasmissione di dati binari elementari a bassa velocità utilizzando una tecnica di modulazione AM. X10 permette il controllo di luci e virtualmente di ogni altro dispositivo elettrico all'interno della casa. Ciò è realizzato attraverso un modulo inserito in una comune presa elettrica al quale viene assegnato uno dei 256 indirizzi possibili; i dispositivi che si vogliono controllare sono quindi inseriti nelle prese di questi moduli.

Una unità di controllo e trasmissione, eventualmente collegata con un PC, è quindi inserita all'interno della stessa rete elettrica e può controllare, inviare semplici comandi di accensione/spegnimento o di interrogazione ai moduli inseriti nella rete, realizzando in questo modo un controllo intelligente dei dispositivi direttamente connessi a questi moduli.

L'inconveniente di un sistema di questo tipo è che ci sono problemi se il bit rate aumenta oltre i pochi bits per secondo a causa di rumori, interferenze, attenuazione, variazioni di impedenza, riflessioni per impedenza.

Le limitate capacità trasmissive di questa tecnologia la rendono non adatta ad un contesto evoluto di home networking (ad es. non è previsto all'interno del protocollo il meccanismo dell'aknowledgment del comando eseguito)

Il sistema CEBus

CBUS è un protocollo di comunicazione sviluppato dall'associazione EIA (Electronic Industries Association). CEBus (Consumer Electronics Bus) è uno standard, dal 1992 per sistemi di Home Automation.

Il protocollo non è stato progettato solo per Home Automation, ma anche nei settori commerciali, industriali e nei trasporti.

Molte industrie hanno partecipato ai metting organizzati da CEBus con l'obbiettivo di creare prodotti e applicazioni per i vari settori (elettrico, gas e telefono). Alcune caratteristiche di questo protocollo sono:

1. Applica le automazioni alle case già esistenti;
2. Permette di aggiungere e togliere applicazioni e componenti nel sistema senza interruzioni e con una minima variazione della configurazione della rete, cioè il Plug and Play;
3. Consente un facile metodo di accesso ai componenti
4. Supporta la distribuzione audio a larga banda e i servizi video oltre a una varietà di segnali analogici e digitali
5. Il protocollo usa un metodo di comunicazione non centralizzato; l'unità di controllo è distribuita tra le varie applicazioni.
6. È un sistema flessibile e aperto a tutti i costruttori di apparecchiature elettroniche

Architettura CEBus

CEBus supporta una topologia flessibile. Un dispositivo può essere messo dovunque e viene collegato alla rete per mezzo di interfacce CEBus. I messaggi possono essere trasmessi, nel sistema, attraverso circuiti elettronici chiamati *router*. Il router non è necessariamente una unità separata, può essere contenuto negli apparecchi.

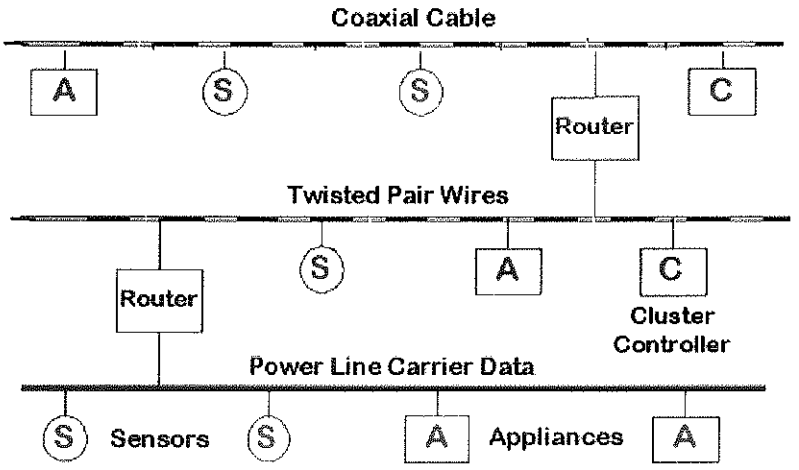
In aggiunta ai messaggi individuali, tutti i componenti o specifici gruppi di componenti possono essere raggiunti con un singolo messaggio contenente un unico *broadcast address*. Tutti i componenti CEBus devono rispondere al *broadcast address.*

Il singolo componente può essere inserito in uno o più gruppi. Questo permette ad un messaggio di essere spedito a tutti gli allarmi o alla portineria di un edificio. I costruttori di componenti scelgono i componenti che devono creare gli indirizzi e come devono essere sostenuti. I membri di un gruppo ricevono i messaggi contenenti l'indirizzo di gruppo (*group address*).

CEBus non usa una unità di controllo centralizzata per controllare i messaggi inviati. Il controllo è distribuito attraverso le applicazioni CEBus e i routers. Lo standard CEBus non specifica una particolare topologia.

Tutte le applicazioni collegate ai punti della rete sono considerate logicamente come se fossero sul bus. Questo significa che tutte le applicazioni su un particolare mezzo avvertono pacchetti di dati quasi nello stesso momento. Tutte le applicazioni leggono gli indirizzi contenuti in un messaggio. Alcune delle applicazioni con un determinato indirizzo leggono, agiscono e rispondono di conseguenza.

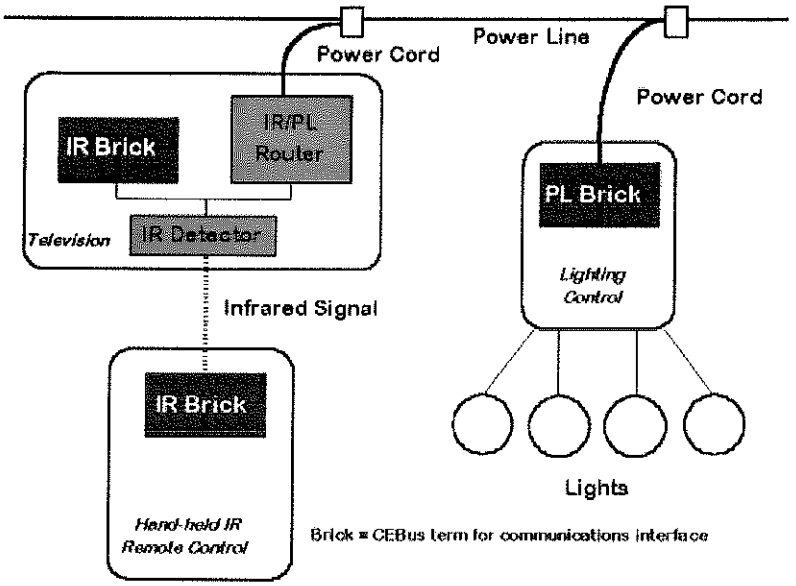
La figura illustra una tipica rete CEBus con collegamenti tramite router. Le applicazioni e i sensori sono collegati alla rete CEBus dove è più conveniente. Il controllore, illustrato in figura, è responsabile dell'organizzazione, dell'illuminazione e della gestione dell'energia.



In figura viene illustrata la configurazione delle applicazioni che operano in una rete CEBus . Con il telecomando della televisione possono essere accese le luci digitando un opportuno codice. La televisione riceve il segnale IR da un modulo di interfaccia chiamato brick. Il televisore interpreta il segnale e riconosce che non è indirizzato ad esso; quindi passa il segnale al router incorporato nella TV; il router invia alla rete elettrica un segnale contenente il comando da indirizzare all'unità di controllo delle luci. Il controllore delle luci riceve il messaggio e le accende.

Il sottolivello MAC del livello 2 del CEBus implementa un CSMA/CDCR. Questo permette ad ogni dispositivo sulla rete di accedere al media ad ogni istante; tuttavia un nodo che vuol mandare un pacchetto dati, deve prima ascoltare che non ci siano altri pacchetti in quel momento sulla linea e solo quando la linea è libera, si può mandare il pacchetto.

CEBus specifica anche un linguaggio CAL (Common Language Application) che permette alle varie unità di comunicare tra loro. Ogni unità è definita come un contesto in CAL. Ogni contesto è suddiviso in oggetti. I segnali di comando possono identificare il contesto e attivare funzioni sugli oggetti. I ricettori attaccati a quest'unità ricevono questi comandi e li attuano.



Il sistema Home Electronic System (HES)

La Home Electronic System (HES) è uno standard che sta per essere sviluppato da un gruppo di lavoro costituito dall'ISO (Organizzazione Internazionale per la Standardizzazione) e dalla IEC (Commissione Elettrotecnica Internazionale) di Ginevra, Svizzera. La scrittura dello standard è eseguita da esperti dei paesi membri. Esperti tecnici dei seguenti paesi si incontrano due volte l'anno per formulare lo standard HES: Canada, Francia, Italia, Giappone, Olanda, Norvegia, Svezia, Regno Unito e Stati Uniti.

Un primo scopo dell'HES è definire un hardware e un software in modo che un costruttore possa offrire prodotti in grado di operare in una rete comune all'interno della casa automatica.

Per completare questo il gruppo di lavoro sta definendo i seguenti componenti per l'HES:

Universal Interface: un modulo da incorporare in un apparecchio per comunicare su varie reti di home automation;

Command Language: un linguaggio per la comunicazione tra gli apparecchi indipendente dalla rete che trasporta i messaggi;

Home Gate: un accesso residenziale per collegare la rete domestica con la rete di servizio degli enti fornitori.

Il gruppo di lavoro HES ha anche il compito di studiare le applicazioni delle reti di comando, di controllo e di comunicazione in edifici commerciali o misti (edifici con appartamenti in affitto, negozi e uffici).

Modelli di applicazione HES

La costruzione di nuovi dispositivi per la home automation deve rispondere alle seguenti caratteristiche: osservabilità e controllabilità.

Allo scopo di far interagire i dispositivi, queste caratteristiche devono essere conformi tra tutti i vari componenti. Un modello di applicazione descrive come un dispositivo può essere letto, scritto e eseguito da una rete per l'automazione della casa.

Nessun protocollo include modelli completi per il funzionamento di più sottosistemi. Esaminando gli oggetti che servono a soddisfare determinate richieste si può costruire un modello. La scelta degli oggetti, dei metodi e delle variabili è basata su una conoscenza del dispositivo da parte del costruttore di interfacce. Questa informazione dovrebbe essere descritta esplicitamente in un modello di applicazione.

I componenti del sistema HES

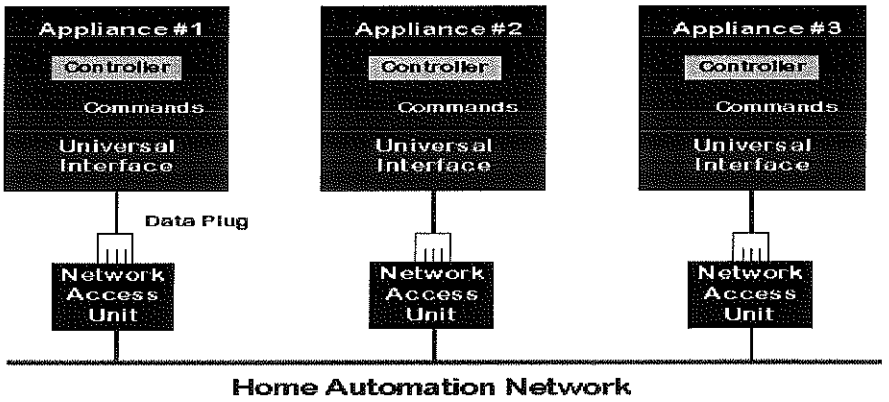
L'interfaccia universale

Il fine primario dello standard internazionale HES è di permettere ad un apparecchio di comunicare su qualunque rete di comunicazione della home automation.

Il dispositivo è dotato di una interfaccia universale (UI) composta da una spina standard. Un linguaggio di comunicazione standard è in via di sviluppo per tutti i comandi e messaggi di applicazione.

Ogni punto di collegamento alla rete contiene una unità di accesso (Network Access Unit) (NAU) per convertire i segnali e i messaggi di un dispositivo in un particolare protocollo di comunicazione della home automation.

HES definisce il protocollo di comunicazione tra UI e la NAU. Il collegamento degli apparecchi all'HES è illustrato nella figura che segue.



Il linguaggio dell'applicazione HES

Il gruppo di lavoro dell'HES sta sintetizzando un nuovo linguaggio di applicazione. La UI dovrebbe funzionare con tutte le possibili reti di home automation.

Il linguaggio HES deve soddisfare una vasta gamma di comandi per le varie reti.

Il legame UI-NAU tra l'apparecchio e la rete non ottimizza le attività del sistema di automazione della casa; ciò nonostante, riduce i costi. Così, la sfida del gruppo di lavoro dell'HES è di definire un legame tra UI-NAU che diminuisca i costi delle interfacce dei dispositivi senza compromettere le prestazioni della rete.

Home gateway

Il concetto di accesso residenziale sta ottenendo molta attenzione nell'industria dei componenti e nei fornitori di servizi. È un mezzo essenziale per raggiungere i clienti con nuovi servizi; ciò nonostante la forma di tale accesso è ancora incerta.

Alla conferenza degli accessi residenziali del maggio 1997 sono state presentate molte opzioni (quanti accessi, dove sono collocati e chi sono i proprietari).

La funzione base di un accesso di rete è il trasferimento di un protocollo di rete locale (LAN) ad un protocollo di rete più ampia (WAN).

Inoltre, l'accesso può contenere caratteristiche di sbarramento (firewall) che limitano la circolazione dei messaggi dentro e fuori la rete. Un firewall è un termine molto usato in internet. Molte reti sono collegate all'internet pubblico attraverso un processore che esamina i messaggi circolanti per impedire l'accesso alla rete da parte di fonti non autorizzate. In tal modo, una caratteristica firewall in un accesso, permetterà all'utente di esercitare il controllo sui messaggi esterni che entrano.

L'utente e i fornitori di servizi si accorderanno su specifici diritti di accesso per distribuire determinati servizi. Il gruppo di lavoro dell'HES sta scrivendo delle disposizioni sul firewall in una descrizione particolareggiata di accesso residenziale. La gestione della privacy sta ottenendo l'attenzione internazionale. Per esempio, il Canada e alcuni paesi europei hanno leggi affidate ad un mandatario di protezione della privacy del cliente che controlla l'accesso ai suoi messaggi personali dal cliente.

Avanzamento degli standard internazionali

Anche se l'avanzamento degli standard internazionali è relativamente lento, la partecipazione è importante per lo scambio di idee tra tutti gli interessati all'automazione della casa. La descrizione particolareggiata dei modelli di applicazione può favorire l'interoperabilità del prodotto. Il forum sui protocolli sta incoraggiando la competizione dei protocolli stessi. Tale competizione però comporta uno spreco di risorse, confondendo i potenziali clienti e ritardando l'industria dell'automazione della casa.

L'obbiettivo primario delle industrie di prodotti e delle compagnie fornitrici di servizi dovrebbe essere quello di stimolare l'interesse dell'utente, in modo da permettere lo sviluppo dell'industria del settore.

Associazione EHSA

EHSA, acronimo di European Home System Association, è un'organizzazione aperta. Il suo scopo è sostenere e favorire l'industria europea nel campo della Home Automation. Tra i suoi membri ci sono le maggiori industrie europee che producono sistemi per la Home Automation.

EHSA include anche rappresentanti del settore edile, impiantistico, architetti, produttori di energia e fornitori di sistemi di telecomunicazione. Ognuno di questi ha il proprio campo di conoscenze e tutti danno il loro contributo per ottenere uno standard comune.

Sostenendo e promuovendo l'European Home System (EHS), EHSA riflette gli interessi e gli scopi dei suoi associati aiutandoli a sviluppare nuove conoscenze e nuovi affari.

Il sistema EHS determina il modo in cui gli apparecchi elettronici ed elettrici di una casa devono essere collegati e come questi possono comunicare per completare le loro funzioni.

EHS garantisce un controllo ripartito, un linguaggio e un modo di comunicazione comune per tutte le apparecchiature, elettriche ed elettroniche, della casa.

EHS intercollega e completa tutti i vari componenti presenti in una casa, tutti questi lavoreranno insieme formando così un sistema di Home Automation. EHS non è un'altra rete o un sistema bus per l'edilizia, ma è una rete aperta a tutto. EHS è applicato su media e larga scala in diversi paesi europei, coinvolgendo varie industrie di prodotti, tutte queste partecipano allo scopo di sviluppare uno standard aperto per l'Europa ed il resto del mondo.

Per controllare e far interagire gli apparecchi elettronici ed elettrici fuori e dentro la casa, indipendentemente dal costruttore, è stato delineato un protocollo comunemente concordato per fornire un mezzo di comunicazione e una più ampia applicazione: European Home System (EHS).

Multiple media

I segnali di controllo di un tale sistema di automazione domestico possono essere trasportati su impianti già esistenti: onde convogliate (, cavo coassiale della TV e il doppino telefonico (TP).

EHS copre anche il controllo a distanza con infrarossi (e radio frequenze (RF). Il modello standard di riferimento usato per l'EHS (modello OSI) è modulare a condizione che i nuovi mezzi (come la fibra ottica) possano essere facilmente accresciuti in futuro.

Plug and play

È la caratteristica più potente di tutto il sistema EHS. Tutti i prodotti possono essere semplicemente inseriti o collegati alla rete. Non è richiesta nessuna installazione professionale, eccetto per il collegamento coi cavi e la sistemazione delle connessioni.

L'autoconfigurazione del sistema all'accensione è prevista, come la riconfigurazione del sistema, quando gli apparecchi sono mossi dalla rete. L'intelligenza del sistema garantisce che EHS è facile da usare.

Controllo distribuito

Il sistema non è dipendente da una singola unità di controllo in una posizione centrale. Attraverso il concetto degli apparecchi logici, le unità di controllo possono essere inserite in opportune posizioni.

L'intelligenza disponibile in microprocessori può essere ripartita tra i vari elementi del sistema.

Il protocollo EHS può essere implementato da qualunque costruttore dello standard su un microcontrollore a 8 bit, il quale assicura la disponibilità di funzioni multiple per i singoli componenti e la possibilità di inserire componenti che non sono EHS.

Applicazioni multiple

L'EHS è applicabile universalmente ed è indipendente da specifiche esigenze di applicazione; un vasto set di comandi è determinato per fornire le funzioni di controllo e di comando dei vari apparecchi inseriti nelle varie zone della casa.

Caratteristiche della rete EHS

La specifica dell'EHS - EHS 1.3 – descrive vari modelli di mezzi per il trasporto dei segnali di controllo, dell'energia e delle informazioni.

Per il momento i tipi di mezzi sostenuti sono i conduttori di potenza (2,4 Kbps a topologia libera) e il cavo doppino intrecciato (48 Kbps a topologia libera TP).

EHS costituisce una rete di controllo pienamente integrata, costituita da una o più sezioni di rete, ognuna usa un mezzo singolo, legati insieme da dispositivi di instradamento detti *routers*.

tipo	TP1	TP2	CX	PL	RF	IR
Bit	9,6 kbps	64 kbps	9,6 kbps	2,4 kbps	1,2 kbps	1,1 kbps
Subnet di	56	255	255	255	255	255
canali	-	14	Molti	-	40	-
Bit	-	64 kbps	analogico	-	32 kbps	-
topologia	libera	bus	bus	libera	libera	libera
distanza	500 m	300 m	150 m	3 km	50-200 m	stanza

Caratteristiche degli indirizzi EHS

I livelli di destinazione dei segnali di comando sono rappresentati da indirizzi che collegano gli attuatori con le applicazioni. La gestione della rete usa un unico codice di destinazione. Ogni sezione della rete permette di indirizzare fino a 256 terminali. Le sezioni della rete possono essere facilmente intercollegate attraverso un instradatore. Un sistema può gestire milioni di indirizzi (oltre 10^{12}).

Sicurezza della comunicazione

La funzione Medium Access Controll (MAC) decide quale unità può trasmettere in caso di competizione per il mezzo. La decisione è ottenuta attraverso la comparazione di segnali spediti e ricevuti. Dopo la decisione, una unità assicura l'accesso al mezzo, il controllo di accesso è ottimizzato per le caratteristiche del mezzo; per esempio, il PL aderisce alla normativa europea per la segnalazione su linee principali.

I pacchetti trasmessi sui mezzi sono ricevuti dal sistema ricevente qualora arrivino senza errori. La tecnica di codificazione dipende dal mezzo su cui viaggiano i segnali; per esempio PL usa una codificazione di correzione ed individuazione dell'errore. TP1, TP2 e CX usano solo l'individuazione dell'errore.

Gestione della rete

I servizi di gestione del sistema sono responsabili della configurazione della rete. Essi gestiscono l'inizializzazione della rete, i conflitti di destinazione e il corretto utilizzo delle comunicazioni. I servizi di gestione delle applicazioni controllano la corretta cooperazione tra le subunità della rete.

I servizi sostengono l'associazione automatica tra gli oggetti e risolvono i conflitti che possono nascere in tali associazioni.

È definito un set di funzioni nel quale ogni **unità** fornisce una funzione specifica nel sistema.

Le unità del sistema sono utilizzate per l'integrazione e gestione della rete. Le unità del sistema si possono dividere in Device Coordinator (coordinatore dell'apparecchio), Medium Controller (regolatore del mezzo) e Router (instradatore).

Le unità composte rappresentano una o più subunità, ognuna rappresenta un processo di applicazione indirizzato individualmente. Una subunità è una raccolta di oggetti che rappresentano gli elementi base delle applicazioni. Le subunità sono i regolatori e gli apparecchi semplici o complessi.

Certificazioni e test di conformità

Dovuto alla natura aperta dell'EHS, chiunque può perfezionare le specifiche dell'EHS o parti di essa. I servizi di test di conformità saranno garantiti da test fatti in case indipendenti in diversi paesi. Le specifiche dell'EHS è il riferimento per i test di conformità. L'EHSA e i laboratori indipendenti per i test hanno definito le specifiche per i test di conformità. L'EHSA nominerà i laboratori dei test e coordinerà l'omologazione delle specifiche dei test, le procedure e gli strumenti da usare.

Associazione EIBA

EIBA è l'acronimo di "European Installation Bus Association". Raccoglie in Europa oltre 70 aziende tutte appartenenti al mondo dell'installazione elettrica. L'associazione è stata fondata in Belgio nel 1990 dalle principali aziende europee operanti nel settore dell'impiantistica elettrica ed ha sede a Bruxelles. L'obiettivo che EIBA si è data è quello di promuovere un sistema unico per l'installazione elettrica. I prodotti marchiati con il simbolo "EIB" sono garantiti come compatibili ed interoperabili fra di loro e quindi possono coesistere nel sistema anche se provenienti da costruttori differenti. Anche in Italia, come negli altri paesi europei, esiste una associazione "EIBA Italia"

Soluzioni possibili con il sistema EIBA

Nell'impianto realizzato con il sistema EIB tutti i componenti possono "colloquiare" fra di loro utilizzando un unico conduttore bipolare denominato appunto "linea bus".

I comandi, le segnalazioni, i dati necessari per la supervisione e tutti i parametri dell'impianto come corrente, tensioni, informazioni sui consumi hanno come unico mezzo di trasmissione un "cavo bipolare" il cavo bus appunto. Questa circostanza implica una drastica riduzione dei tempi di posa dei conduttori e di tutto ciò che necessita per l'installazione degli stessi, come canaline, tracce ecc.

La funzione dell'impianto viene poi determinata con l'impiego di un pacchetto software, denominato ETS (EIBA Tool Software) con il quale i singoli componenti collegati tutti in parallelo all'unica linea bifilare necessaria acquistano una loro "individualità" cioè possono essere riconosciuti singolarmente, mediante un indirizzo, che altro non è se non il numero di identificazione del dispositivo stesso ed una "coscienza" dei loro compiti cioè il programma di funzionamento personalizzato per il tipo di impianto in cui sono installati.

Immediate le conseguenze di quanto realizzato:

- i componenti sono di impiego generale e vengono personalizzati mediante un adeguato software (ETS);
- la messa a punto delle funzioni secondo le esigenze dell'utilizzatore può essere effettuata ad installazione ultimata;
- le modifiche successive all'installazione possono essere effettuate da qualunque punto dell'impianto ed in qualunque momento senza interrompere la funzionalità dello stesso.

Un altro vantaggio che non appare immediatamente ma che va considerato nella giusta prospettiva è la possibilità di realizzare un considerevole risparmio energetico distribuendo l'energia dove serve e quando serve sospendendola nei momenti opportuni ed erogandola secondo un apposito programma temporale.

Architettura generale

Tecnologia: Il sistema EIB è un sistema essenzialmente "decentralizzato" che comporta l'assenza di una centrale che contiene tutti i dati dell'impianto; ogni dispositivo è costituito da una parte "intelligente" (un microprocessore) che contiene le istruzioni per il proprio funzionamento, quindi:

come si chiama (indirizzo fisico);

cosa deve fare (funzioni implementate con ETS);

con chi lo deve fare (sempre mediante ETS).

Nella sua configurazione minima un sistema EIB può essere costituito come segue:

1. alimentatore (può alimentare fino a 64 componenti);
2. terminale di uscita (per l'attivazione delle utenze);
3. terminale di ingresso (per l'invio di comandi).

Questa configurazione minima è in grado di funzionare perfettamente e, cosa più importante, può essere adeguata per adattarsi perfettamente ad eventuali esigenze di ampliamento; la struttura del sistema può adattarsi fino a gestire oltre 10.000 componenti.

Alimentazione: l'alimentatore è provvisto di circuiti di regolazione di corrente e di tensione ed è quindi protetto contro eventuali corto circuiti; inoltre brevi eventuali interruzioni di rete inferiori a 100 ms non hanno conseguenze sul funzionamento dell'impianto. Tutti i dispositivi possono funzionare correttamente con una tensione minima di 21 Vcc ed assorbono una potenza inferiore a 150-200 mW ciascuno.

In caso di impianti con una concentrazione di più di 30 dispositivi entro una distanza di 10 m (si parla di lunghezza del cavo) l'alimentatore deve essere situato nelle immediate vicinanze; nel caso sia necessario un secondo alimentatore questo deve essere posto ad una distanza non inferiore a 200 m; in ciascuna linea non possono essere inseriti più di due alimentatori.

Essendo i dispositivi interconnessi attraverso una sola linea bifilare l'alimentazione ed i segnali condividono lo stesso supporto quindi sia l'alimentatore ed i dispositivi devono essere adatti a separare il segnale dall'alimentazione.

Trasmissione dei segnali

L'informazione viene trasmessa su una coppia di conduttori in modo simmetrico (un conduttore trasmette il segnale, l'altro conduttore trasmette lo stesso segnale invertito di 180°), in tal modo l'eventuale disturbo essendo presente in modo identico (cioè con la stessa polarità) su entrambi i conduttori viene ad essere eliminato dal circuito di ingresso del dispositivo.

Il cavo BUS dispone di isolamento a 4 KV tra i conduttori interni e la guaina esterna.

Per questi motivi il cavo BUS può essere posato anche nelle canaline dei conduttori di potenza; può essere inoltre interrotto e connesso in parallelo per effettuare connessioni con qualunque geometria possibile: -a bus propriamente detto; - ad anello; - a stella; - ad albero; - con topologia mista.

Questa circostanza facilita in modo incredibile l'installazione, ma soprattutto le eventuali modifiche ed ampliamenti essendo possibile interrompere in qualunque punto il cavo bus ed effettuare una derivazione per aggiungere altri dispositivi; non sono inoltre necessarie resistenze terminali.

In conclusione dopo questa breve panoramica di protocolli domotici , possiamo renderci conto come la disciplina sia molto vasta. La speranza è la convergenza di queste tecnologie verso un unico standard che raccolga le varie esperienza dei vari produttori. Anche se un primo passo è stato fatto con Konnex (ottenuto dalla convergenza di EHS, EIBA, BatiBus) riteniamo che la strada sia ancora lontana.

Architettura di base

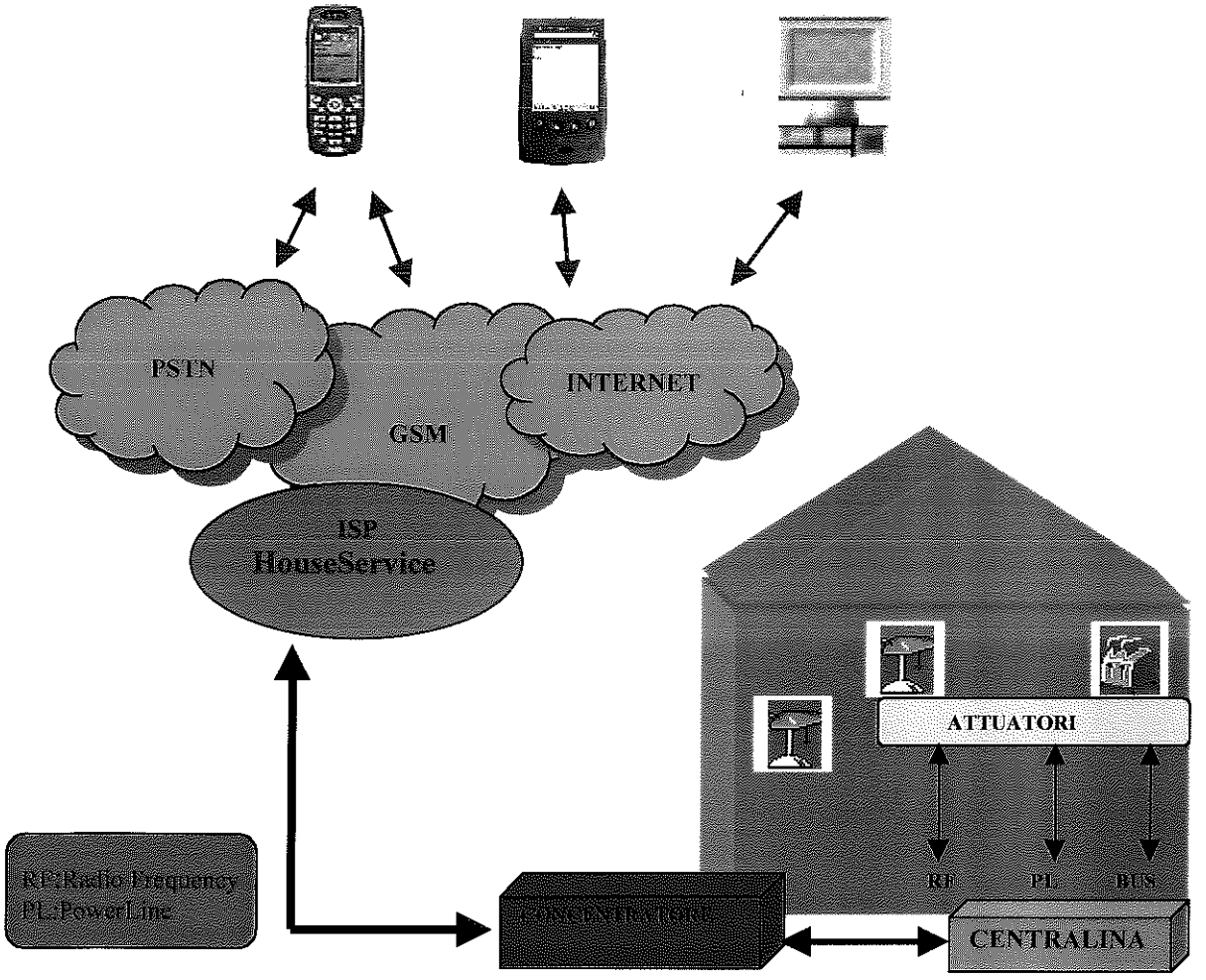
Fino ad ora si è descritto che cosa rappresenta l'*home automation* e i protocolli che caratterizzano questa area. Come abbiamo visto la *domotica* è un campo molto vasto che abbraccia molte tecnologie ben conosciute ma che non riescono ancora ad avere il loro spazio nell'ambiente domestico.

Nel seguito si descrive il sistema con cui abbiamo lavorato, presentando le sue caratteristiche e le scelte progettuali che sono state fatte per lo sviluppo della tecnica WAP.

La fornitura di servizi di home networking all'interno della abitazione richiede, in base all'architettura descritta precedentemente la realizzazione di un sistema a più livelli cioè:

fornitore del servizio, *concentratore*, *centralina*, *attuatori*.

Il *concentratore* è l'unico elemento architetturale che è in comunicazione diretta, attraverso l'interfaccia domestica, fra la casa e la rete esterna.



La sua progettazione risponde all'esigenza di disporre localmente di un sistema che esegua i comandi provenienti dall'esterno e le passi alla centralina..Quest'ultima realizza le

funzionalità di trasmissione, all'interno della rete domestica, delle informazioni di stato e degli eventi di sistema (ad esempio un evento di allarme o di malfunzionamento) relativi ai dispositivi stessi. Gli attuatori sono i componenti a cui sono inviati i comandi e normalmente sono individuati da un indirizzo fisico

Lo scenario che il sistema finale dovrà implementare è quello di un *ISP*, che offre ai suoi utenti, attraverso una interfaccia multicanale, funzionalità per il controllo, la gestione e il monitoraggio da remoto di una serie di dispositivi "intelligenti" presenti all'interno della propria abitazioni.

I canali di comunicazione che deve mettere a disposizione dell'utente finale per la sua interazione con il sistema sono diversi e comprendono le interfacce *Web, Wap, SMS, E-mail, Fax* e *Voce*. Uno o più di questi canali viene utilizzato per la realizzazione di ognuna delle funzionalità previste, sulla base dei requisiti specifici di ogni interazione e delle preferenze di utilizzo espresse dall'utente in fase di configurazione dei servizi.

In generale, *HouseService* offrirà un servizio altamente personalizzabile, consentendo ai propri utenti la possibilità di configurare tutti i parametri relativi alla fornitura dei servizi stessi, in termini di scelta dei canali di comunicazione, di impostazione della frequenza di notifica di eventi o messaggi generati dal sistema verso l'utente, di configurazione di tutte le variabili operative relative ai dispositivi controllati.

Il sistema dovrà implementare tutte le funzionalità interne al fornitore dei servizi e relative all'organizzazione del cliente. In particolare dovrà gestire:

- L'insieme di attività per soddisfare il servizio richiesto dal cliente. Comprende i processi di vendita dei servizi, di Order Handling cioè accettare e gestire ordini di attivazione, verificare la correttezza degli stessi, notificare al cliente finale l'attivazione completa dei servizi sottoscritti, di Service Configuration, per gestire l'installazione e la configurazione dei dispositivi presso il cliente e il testing dei servizi.
- Dare fiducia al cliente attraverso funzionalità di gestione delle segnalazioni di problemi/disservizi/lamentele da parte del cliente finale, e nell'intercettazione e la gestione degli allarmi (ad esempio malfunzionamenti nei dispositivi) e per la realizzazione di funzionalità di telemanutenzione.
- La pubblicità, la realizzazione di funzionalità appartenenti ai processi di consultazione on line della propria fattura completa dei dati di consumo, la preparazione e l'invio di bollette, la gestione dei pagamenti attraverso diverse modalità, l'interruzione del servizio a seguito dell'esaurimento del credito o di mancati pagamenti

4. Interfaccia verso l'utente

In questo lavoro di tesi facciamo a meno del fornitore di servizi *HouseService* poichè il sistema è ancora in fase di progettazione e pertanto l'utente interagisce direttamente con il *concentratore* senza passare attraverso un livello intermedio. I canali di comunicazione che l'utente utilizza sono i dispositivi mobili.

La creazione di interfacce è diviso in due fasi. Inizialmente siamo partiti dal considerare il semplice telefonino WAP e la programmazione di pagine WML da visualizzare sul display. Successivamente abbiamo abbandonato tale strada per far posto alla nuova piattaforma della Microsoft .NET, che ha permesso lo sviluppo di interfacce per qualunque dispositivo mobile, cioè gli smart devices.

Prima Fase

Il primo dispositivo a cui ci siamo rivolti è stato il telefonino cellulare che colloquia con il concentratore di casa mediante pagine scritte in *WML*. Il *WML* è un linguaggio di markup basato sull'*XML (Extensible Markup Language)*.

Un dispositivo Wap al contrario del Web, non apre una pagina, bensì un deck formato da card. Le card specificano una o più unità di interazione con l'utente (ad esempio un menù di possibili scelte, una schermata di testo, un campo di immissione dati). Dal punto di vista logico, un utente naviga attraverso una serie di card *WML*, ne scorre il contenuto, inserisce le informazioni richieste, fa delle scelte e si sposta verso un'altra card.

WML include il supporto per gestire esplicitamente la navigazione tra card e deck e la gestione degli eventi all'interno del dispositivo (utilizzabili per navigare o per eseguire degli script). Inoltre esso include un supporto per la presentazione di testo e immagini, inclusa una varietà di comandi per la formattazione e il layout. Per esempio può essere specificato un testo in grassetto.

Ogni deck *WML* ha inizio con lo stesso header *XML*:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
```

La prima riga di codice indica semplicemente che ciò che segue è un documento *XML* e mostra il numero di versione usata. La riga seguente seleziona il tipo di documento e indica l'URL del DTD. Questo DTD segnala l'intera definizione XML di WML.

Nella figura possiamo vedere un piccolo tratto di codice *WML* e ciò che viene visualizzato sul display del telefonino. Si tratta di un menù molto ridotto che elenca alcuni dispositivi presenti dentro l'abitazione.

```
<card id="card3" title="Configura">
  <p>
    Menù
    <select name ="dispositivo">
<option value="Persiana" onpick="#card12">Persiana</option>
<option value="Luce" onpick="#card12">Luce</option>
    </select>
</card>
```

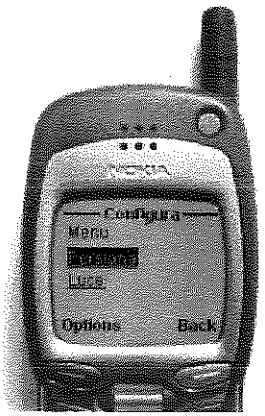


Figura 4. Card di un telefonino

Cerchiamo di indicare le linee guida generali che ci hanno condotto a realizzare le interfacce per i dispositivi mobili.

Applicazioni Intuitive

L'applicazione è stata resa il più possibile intuitiva affinché risulti semplice da eseguire da parte dell'utente. A causa dei display piuttosto ridotti dei dispositivi portatili, abbiamo cercato una interazione uomo-macchina, ripetitiva e prevedibile all'utente. Allorché il numero dei dispositivi da controllare comincia a crescere dobbiamo cercare un compromesso tra flessibilità e efficienza.

Applicazioni efficienti

L'utente è generalmente impaziente e perciò l'applicazione deve essere rapida da utilizzare. Per tale motivo si riduce al minimo la quantità di input richiesta all'utente.

Facilità di memorizzazione
Se l'utente utilizza raramente il mezzo cellulare per accedere al *concentratore*, deve essere in grado di ricordare facilmente l'intero processo.
Tolleranza
L'applicazione *WAP* deve essere il più possibile tollerante ai malfunzionamenti. Occorre richiedere una conferma prima di eseguire una azione costruttiva o distruttiva. Se ad esempio decidesse di accendere o spengere la propria lampada è consigliabile inserire una consenso.

Una possibile via è di organizzare le card dell'applicazione in una struttura gerarchica ordinata alla successione di operazioni che l'utente deve effettuare nel caso in cui si trovasse davanti al portone di casa. Qualunque operazione che l'utente decide di eseguire deve avvenire con il minor numero di clic. Un esempio di gerarchia dell'applicazione potrebbe essere quella mostrata in *figura 5* che visualizza un esempio banale di una gerarchia di card.

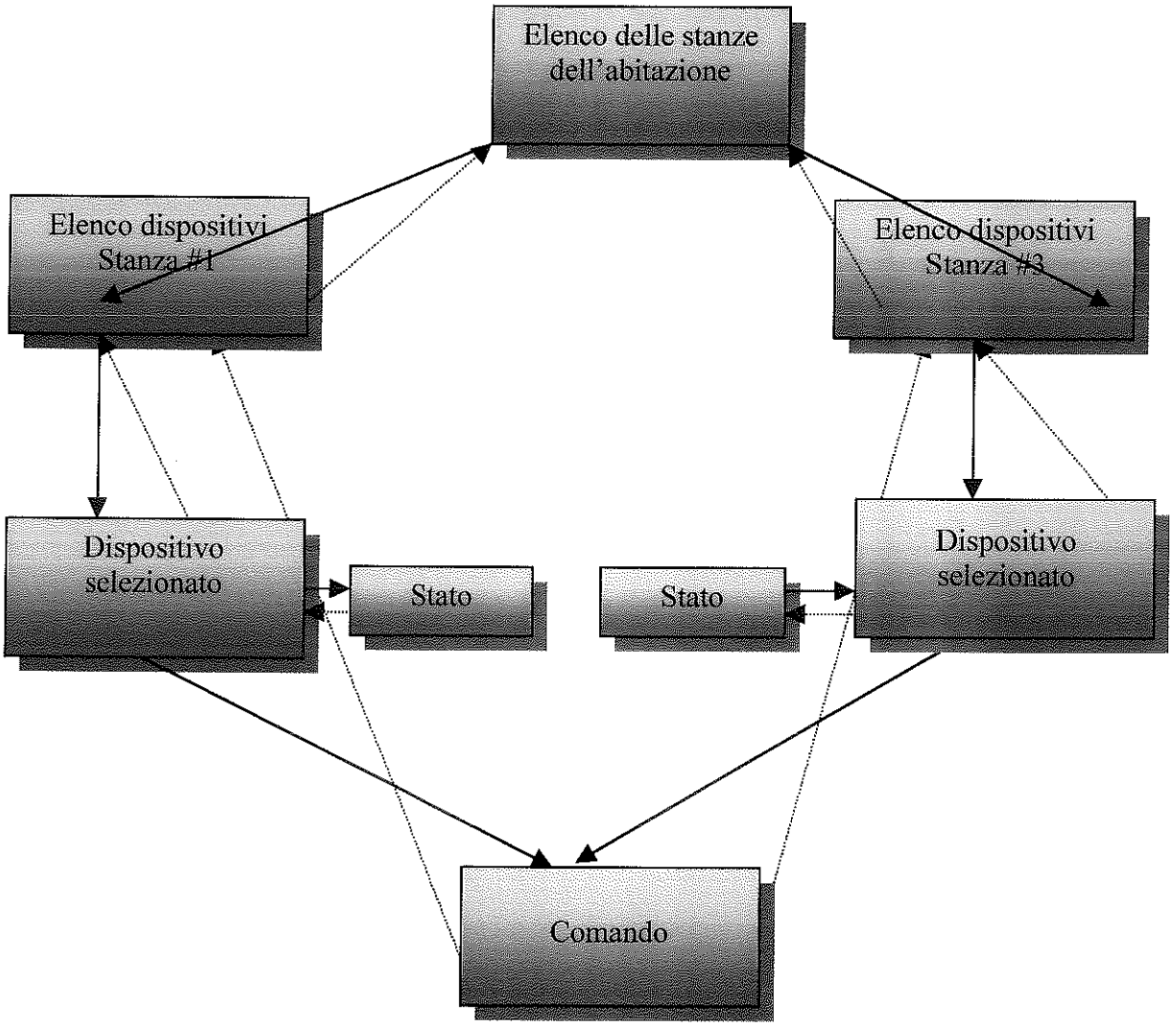


Figura 5 . Esempio di undeck di gestione della abitazione

La gestione del sistema HouseSystem non può essere controllato con un semplice deck come quello in figura perché non soddisfa alcuni requisiti come:
- non può esistere un'unica card **comando** per tutti i dispositivi dislocati all'interno dell'abitazione; alcuni potrebbero richiedere operazioni differenziate.

- le stanze possono essere unite da elementi comuni e quindi è necessario utilizzare un link per passare da una stanza all'altra senza dover ritornare alla card principale;
- l'interazione con un database risulta necessario nel caso in cui l'utente volesse consultare informazioni sullo stato dei dispositivi;

Inoltre la struttura gerarchica funziona in modo particolarmente efficace in connessione con un tipo di esplorazione a ritroso (vedi le linee tratteggiate in figura), in modo da permettere all'utente di spostarsi molto rapidamente anche tornando indietro verso la card principale. Potrebbe essere una buona idea anche fornire agli utenti un meccanismo per ritornare verso la card principale (home page) con un semplice clic.

Nello sviluppo dell'applicazione *WAP* si cerca di eseguire qualsiasi operazione riducendo la quantità di dati immessi. Infatti i dati di testo da inserire costituiscono un tipo di input fastidioso da parte dell'utente in quanto risulta un'operazione incline agli errori.

Purtroppo nella nostra applicazione la immissione di dati risulta necessaria in certe operazioni (come l'immissione dell'indirizzo del dispositivo da controllare); per evitare che l'utente possa commettere errori catastrofici, ogni inserimento di dati implica la successiva convalida dell'input da parte dell'utente. Occorre segnalare a quest'ultimo ciò che ha inserito per decidere se proseguire.

Prima di concludere il paragrafo sulla creazione di pagine per dispositivi cellulari è necessario fare una precisazione. Abbiamo indicato con pagine *WML* il contenuto che il concentratore spedisce al cellulare. In realtà ciò che viene spedito all'utente sono pagine *ASP (Active Server Pages)* che permettono di rendere dinamiche le pagine *WML*. Infatti *ASP* è la tecnologia della Microsoft per la generazione dinamica di contenuti *Web* e *WAP*. Per questo convertiamo le pagine *WML* in *ASP* cambiando l'estensione del file in .asp e aggiungendo in cima la seguente riga di codice:

```
<% Response.ContentType="text/vnd.wap.wml"%>
```

Gestire un sistema complesso come ***HouseSystem*** con un semplice telefonino può risultare difficile; l'utente dovrebbe gestire un numero assai ampio di card con il pericolo di perdersi. Il supporto di un database e di una tabella di instradamento (quest'ultima verrà illustrata nel capitolo successivo), può rendere l'applicazione snella e non ridondante, eliminando le card in eccesso

. Il mezzo di comunicazione mobile da noi preso inizialmente in esame è il telefonino *WAP*, ma attualmente è possibile realizzare una applicazione valida per qualunque altro dispositivo mobile; perciò abbiamo deciso di seguire un'altra strada che porta a considerare l'ultima piattaforma della Microsoft, :*NET*, che verrà illustrata nel paragrafo successivo.

Seconda Fase

Durante lo svolgimento del lavoro è uscita l'ultima release della Microsoft, cioè *.NET Framework Software Development Kit (SDK),* che si è dimostrata interessante e utile per i nostri scopi. Prima di cominciare a descrivere i benefici apportati da .NET al nostro lavoro , è necessario descrivere, seppur brevemente, l'architettura di *.NET Framework*, e in particolare di *ASP.NET.*

.NET Framework

Il nome .NET tende a sottolineare da parte di Microsoft che le applicazioni distribuite, cioè che ripartiscono l'elaborazione tra client e server, saranno sempre più diffuse. Cerchiamo di comprendere meglio il significato del nome .NET.

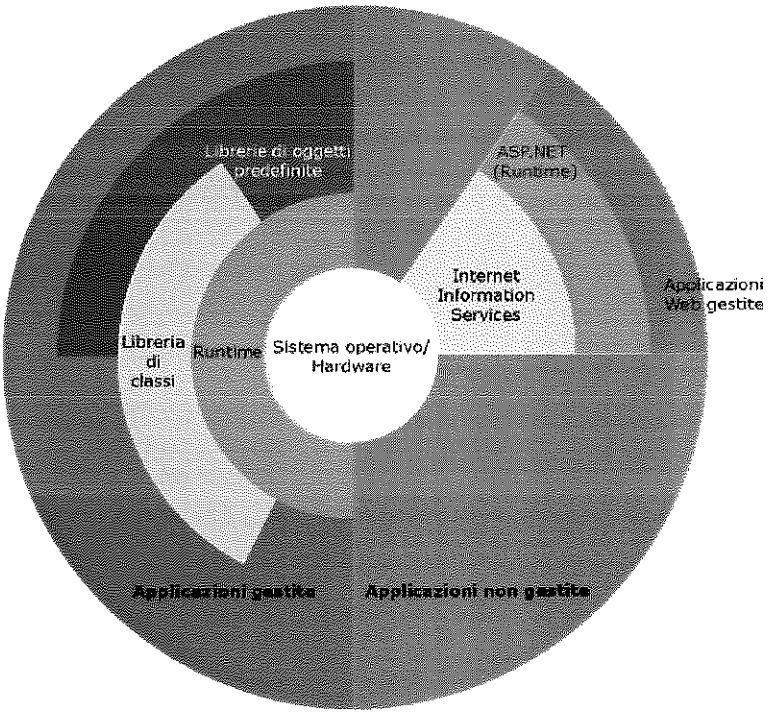
Tutti i sistemi operativi per gli home PC si basano sulla stessa API di Windows. Con l'introduzione delle nuove versioni, alle API sono state aggiunte nuove funzioni, ma si è trattato essenzialmente di un processo di evoluzione ed estensione, anziché di sostituzione.

Ad esempio il modello COM (Component Object Model) era stato concepito come tecnologia OLE (Object Linking and Embedding) e costituiva semplicemente un metodo per collegare fra loro vari tipi di documenti di Office. Dalla sua creazione, questo modello si è evoluto da COM a DCOM (Distribuited COM) e infine a COM+.

È evidente che l'adattamento alle nuove esigenze operative degli stessi strumenti di sviluppo finiva per renderli sempre più complessi e questo processo non poteva essere esteso all'infinito per poter supportare nuovi dispositivi hardware, mantenendo la compatibilità con ciò che è stato creato agli inizi degli anni Novanta. .NET è stato la risposta della Microsoft all'esigenza di cambiamento. In pratica .NET è un nuovo framework ossia una nuova API per la programmazione in Windows.

.NET non è solo una libreria di classi, ma fornisce anche l'ambiente in cui il programma costruito dall'utente viene eseguito e prende il nome *Common Language Runtime*, o *CLR*. Occorre sottolineare che .NET non è un sistema operativo. Il sistema operativo è sempre Windows, almeno fino a che esso non sarà più disponibile per altre piattaforme.

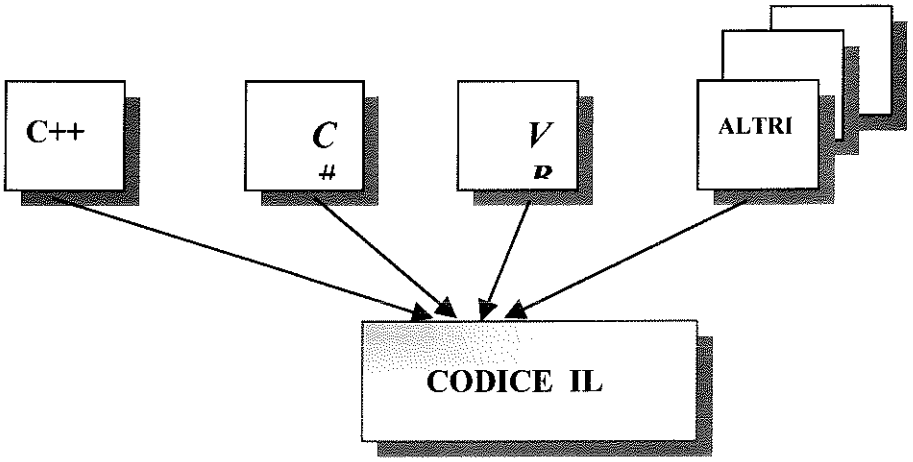
Architettura di .NET



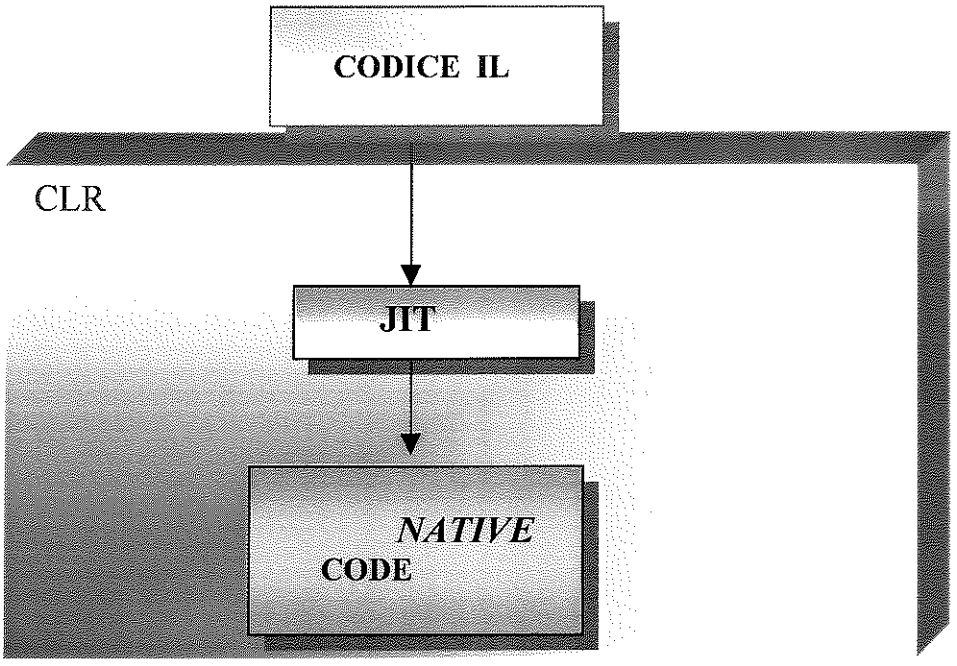
.NET Framework presenta due componenti principali: *Common Language Runtime* (CLR) e *la libreria di classi.* Il CLR rappresenta la base di .NET Framework ed è un ambiente che gestisce l'esecuzione di codice come ad esempio il modo di caricare il

programma, eseguirlo e fornirgli tutti i servizi necessari. Inoltre esso gestisce la memoria, l'esecuzione di thread, l'esecuzione del codice, la verifica della protezione del codice, la compilazione e altri servizi di sistema. Il codice destinato al runtime è definito codice gestito, mentre quello non destinato al runtime è definito codice non gestito. Il *codice gestito* prima di poter essere eseguito deve essere compilato e il codice compilato non contiene istruzioni in linguaggio assembly, bensì in MSIL (*Micosoft Intermediate Language*) o IL.

Il CLR può essere comparato con la *virtual machine* di Java che è un programma che fa dà interfaccia tra il programma vero e proprio ed il sistema operativo sottostante. In questo modo (almeno in teoria) è possibile cambiare il sistema operativo sul quale gira il programma semplicemente utilizzando la versione del framework per l'altro sistema e senza dover ricompilare l'applicazione.



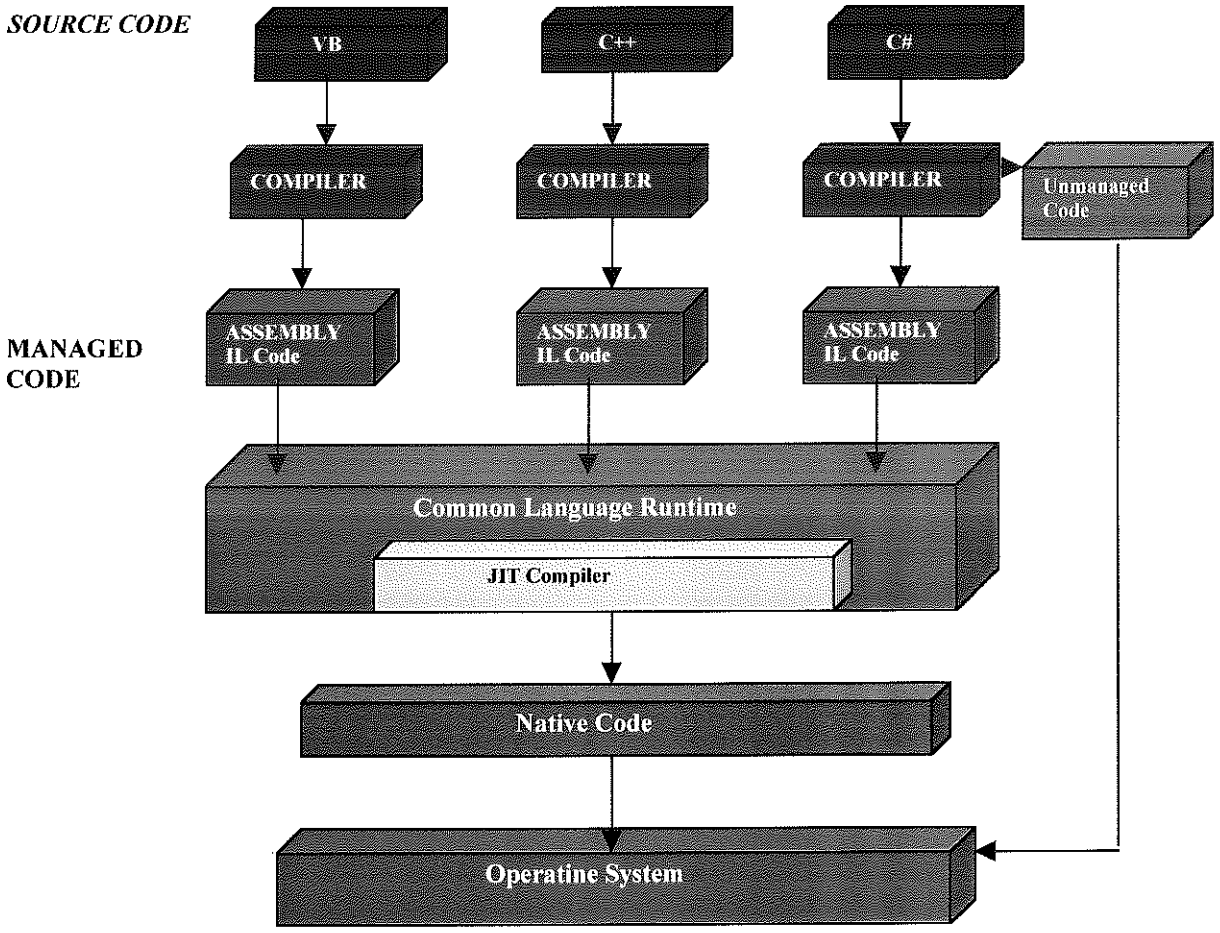
Questo tipo di compilazione produce i *metadati* che rappresentano informazioni descrittive circa l'applicazione, indicando ciò che essa può fare e a chi appartiene. Ovvero i *metadati* descrivono un programma e vengono memorizzate in un file chiamato Portable Executable (PE,).



In fase di esecuzione il linguaggio intermedio viene compilato in codice nativo dalla piattaforma sulla quale .NET è in esecuzione grazie ad un JITter (*Just in Time Compiler*).

Lo scopo per cui IL è compilato *just-in-time* è semplicemente l'indipendenza dal linguaggio in un ambiente orientato agli oggetti. In sostanza si compila in codice intermedio da uno di una serie di linguaggi e il codice compilato può essere in grado di interoperare con quello che è stato compilato da altri linguaggi. Assicurare l'interoperabilità dei linguaggi è sempre stato un obiettivo ambizioso. Gli sviluppatori possono quindi utilizzare una gran varietà di tecnologie e strumenti, ciascuno dei quali può supportare funzionalità e tipi diversi. I compilatori dei linguaggi e gli strumenti che si avvalgono di Common Language Runtime beneficiano comunque del supporto che il runtime incorpora per l'interoperabilità dei linguaggi. CLR fornisce le necessarie basi per l'interoperabilità dei linguaggi specificando e applicando un sistema di tipi comune e fornendo metadati.

Poiché tutti i linguaggi che si avvalgono del runtime osservano le regole del *sistema di tipi comune* per la definizione e l'uso dei tipi, i diversi linguaggi utilizzano i tipi nello stesso modo. I metadati contribuiscono all'interoperabilità dei linguaggi definendo un meccanismo unico per l'archiviazione e il recupero delle informazioni sui tipi. I compilatori archiviano le informazioni sui tipi come metadati e Common Language Runtime utilizza tali informazioni per fornire servizi durante l'esecuzione.



Il runtime è in grado di gestire l'esecuzione di applicazioni multilinguaggio perché tutte le informazioni sui tipi vengono archiviate e recuperate allo stesso modo, indipendentemente dal linguaggio in cui è scritto il codice.
Per essere certi che il proprio codice gestito sia accessibile ad altri sviluppatori indipendentemente dal linguaggio di programmazione da essi adottato, .NET Framework fornisce le specifiche *Commn Language Specification* (CLS) che descrivono un set di funzionalità fondamentali dei linguaggi e definiscono le regole per l'utilizzo di tali funzionalità. Il CLS coopera con CTS (*Common Type System*) per garantire l'interoperabilità.

Il CTS è una specifica in cui sono definite le modalità di dichiarazione, utilizzo e gestione dei tipi nel runtime e che rappresenta una parte importante del supporto runtime nell'integrazione di più linguaggi, cioè una variabile di tipo stringa in C++ non poteva essere passata direttamente ad un programma scritto in VB, ma doveva essere opportunamente convertita perché i due linguaggi non memorizzano le stringhe nello stesso modo. Ma i linguaggi che rispettano CTS, utilizzano le stringhe e gli altri tipi di dati nello stesso modo e quindi possono scambiarsi direttamente i dati. Qualsiasi linguaggio supporti lo standard CLS potrà interoperare con i linguaggi .NET.

Il secondo componente principale è la libreria della classe di base di .NET. Essa rappresenta una dei vantaggi principali nello scrivere codice gestito (cioè qualunque codice progettato per operare all'interno dell'ambiente .NET). Queste classi, scritte da Microsoft, sostituiscono le API di Windows. Si tratta di una libreria di classi,presente nel .Net Framework,ed utilizzabile dai linguaggi come VB.NET, C#, C++.

Un secondo vantaggio nell'uso delle librerie è l'importanza nel fornire l'interoperabilità tra i linguaggi, poiché consentono di usare una singola interfaccia di programmazione per tutte le funzionalità esposte dal CLR. Concludiamo qui la breve panoramica sulla piattaforma .NET e vediamo come è stata utilizzata per i nostri fini.

Il motivo per cui viene abbandonata la precedente strada è la possibilità di allargare le vie di comunicazione che l'utente può utilizzare per comunicare con l'HouseService o il concentratore di casa. Non sarà più il solo telefonino cellulare l'oggetto adoperato dall'utente, ma qualunque dispositivo mobile cioè gli smart devices, grazie alle pagine ASP.NET mobile.

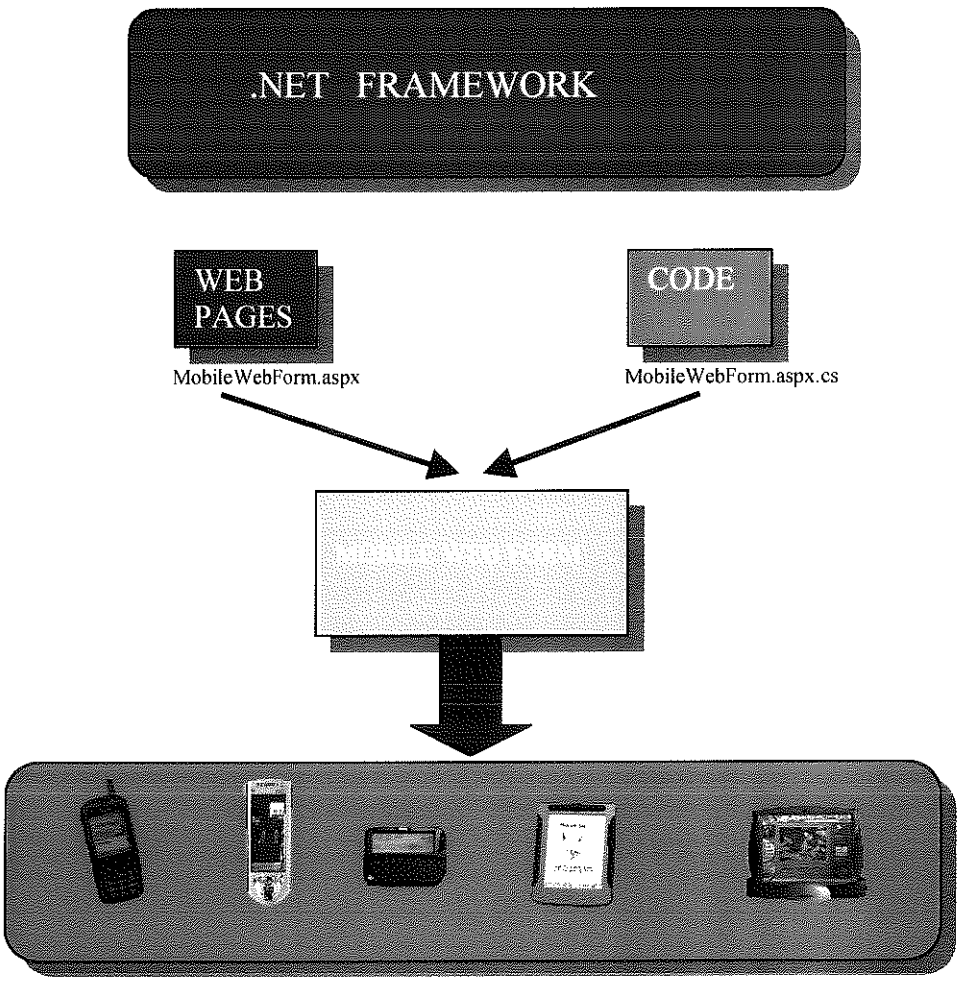
ASP.NET è ampiamente compatibile con ASP a livello di sintassi e fornisce anche un nuovo modello di programmazione e una nuova infrastruttura per la creazione di applicazioni più sicure, scalabili e affidabili. ASP.NET è un ambiente compilato basato su .NET Framework e consente di sfruttare appieno le funzionalità di Common Language Runtime, quali indipendenza dai tipi, ereditarietà, interoperabilità tra i linguaggi e controllo delle versioni.

È possibile creare applicazioni, i web form, in un qualsiasi linguaggio compatibile con questa piattaforma, inclusi Visual Basic .NET, C# e JScript .NET.

In particolare sfrutteremo un componente mobile di ASP.NET che ci permette di creare applicazioni per i dispositivi mobili. La programmazione di form web mobile è simile alla programmazione delle classiche pagine ASP.NET.

Nelle pagine Web Form, come da figura sotttostante, la programmazione dell'interfaccia utente è modulare, cioè viene suddivisa in due elementi distinti: *il componente visivo* e *la logica.*

L'elemento visivo viene definito come form Web, dove sono presenti i controlli Web Forms (che hanno proprietà, metodi ed eventi) del lato server. I controlli Web Forms ci aiutano nella realizzazione di applicazioni Web wireless, come ad esempio telefonini e PDA, e sono in grado di adattare il rendering dell'output ai diversi dispositivi, poiché generano il linguaggio markup più appropriato e supportano i linguaggi WLM (Wireless Markup Language) versione 1.1, HTML versione 3.2 e cHTML (HTML compatto).



La logica della pagina Web Form è composta dal codice che viene creato per interagire con il form. La logica della programmazione può risiedere in un file distinto dal file dell'interfaccia utente. Questo file viene definito file di "codice sottostante" e presenta un'estensione "ASPX.VB" o "ASPX.CS" a seconda del linguaggio di programmazione usato. Questo codice risponde agli "eventi" dei controlli del server.

L'applicazione che costruiamo è *personalizzabile* ed *estensibile*. Infatti è possibile sia personalizzare i controlli per particolari tipi di dispositivi (si pensi ad un PDA che può visualizzare i checkbox a diferenza di un telefonino WAP), sia è possibile creare controlli specializzati estendendo o combinando quelli esistenti, sia è possibile creare controlli Web Forms completamente nuovi. In più, il modello di estensibilità consente di supportare i nuovi dispositivi futuri. Ciò garantisce che le applicazioni Web Forms

per dispositivi portatili scritte oggi potranno essere utilizzate anche con le prossime generazioni di dispositivi intelligenti.

Quando scriviamo la nostra applicazione si creano pagine e controlli Web Forms per dispositivi portatili indipendenti dal dispositivo. Quando uno dei dispositivi supportati esegue una richiesta per una pagina Web Forms, la pagina e i controlli identificano automaticamente il tipo di dispositivo e generano un rendering adatto alle funzionalità del dispositivo. Ad esempio, alcuni dispositivi sono in grado di visualizzare più righe di testo rispetto ad altre, alcune sono in grado di visualizzare immagini e altre no, oppure possono effettuare chiamate telefoniche mentre altri dispositivi non dispongono di questa funzionalità.

Ogni pagina Web Forms mobile deve includere le seguenti direttive standard relative all'intestazione, che ne consentono l'identificazione come pagina per dispositivi portatili. L'attributo Language varia in base al linguaggio scelto per lo sviluppo. Attualmente si può utilizzare C# o Visual Basic. Noi utilizzeremo C#.

```
<%@ Page Inherits="System.Web.UI.MobileControls.MobilePage" Language="C#" %>
<%@ Register TagPrefix="mobile" Namespace="System.Web.UI.MobileControls"
Assembly="System.Web.Mobile"
```

Alla prima riga mediante l'attributo **Inherits** si specifica la classe da cui questa pagina dovrebbe ereditare. Tutte le pagine mobili devono ereditare da questa classe in modo che ASP.NET sappia che state sviluppando una pagina particolare.

La seconda riga assomiglia alla registrazione di un controllo utente. In pratica lo spazio dei nomi System.Web.UI.MobileControls contiene tutti i controlli mobili del server che poi useremo nelle nostre pagine e che si trova nel file System.Web.Mobile.dll.

I controlli che useremo nell'applicazione sono racchiusi in tag che devono includere l'attributo runat="server":

```
<mobile:Form runat="server">
</mobile:Form>
```

Descrivendo la prima fase abbiamo parlato di deck/card, cioè il concentratore spedisce una deck composta da più card. Ora con .NET il deck è sostituito dalla ***MobilePage*** mentre la card dalla ***MobileForm***. Infatti sul piccolo display del mio cellulare si visualizza una pagina mobile che contiene al suo interno diversi form. Ogni form è una spcie di contenitore di controlli e contiene le informazioni e comandi per navigare attraverso gli altri form.

L'output su dispositivi specifici

Anche se ASP.NET gestisce in modo automatico il rendering di ogni controllo mobile a seconda del dispositivo in uso, possiamo volere la visualizzazione di un

controllo su un particolare dispositivo come ad esempio un cellulare e un controllo differente su un altro dispositivo come il PDA. Infatti mentre al PDA spedisco alcuni immagini a colori (come la lampadina della luce accesa o spenta), queste non verrebbero ben fruite attraverso il display monocromatico e piccolo del cellulare. Allora introduco **<DeviceSpecific>** e gli elementi **<choice>,** che permettono di rendere gli articoli differenti.

```
<mobile:Form runat=server id="Form5">
<DeviceSpecific>
<Choice Filter="IsHtml32" ImageURL="Luce.gif"
<AlternateText="Luce" />
<Choice Filter="IsWML" mageURL="Luce.wbmp"/>
</DeviceSpecific>
```

Come si vede abbiamo selezionato due tipi di immagini che devono essere visualizzate sui dispositivi. La prima **choice** esamina se il dispositivo è dotato delle compatibilità IsHtml32, cioè la versione HTML 3.2, e se questa non la supporta allora visualizza l'immagine con estensione wbmp (Wireless BitMap).

Posso scegliere tanti elementi **choice** quanti mi pare, uno per ogni dispositivo. Soltanto uno di questi elementi verrà renderizzato; quello che corrisponde meglio alle prestazioni del dispositivo. Se pensiamo di aver finito, ci sbagliamo perché dobbiamo informare ASP.NET che cosa è il filtro IsHtml32 e IsWML. In particolare questa informazione va comunicato al file **web.config**, un file XML basato su testo, aggiungendo il seguente codice:

```
<system.web>
<deviceFilters>
<filter name="IsHtml32" compare="PreferredrenderingType"
< a r g u m e n t = " h t m l "                    / >
<filter name="IsWML" compare=" PreferredrenderingType"
< a r g u m e n t = " w m l "                    / >
</deviceFilters>
</system.web>
```

Mobile Internet Controls Runtime usa questi **deviceFilters** per eseguire l'opportuno rendering. Abbiamo introdotto il file **web.config** che ha un ruolo importante nel controllare l'elaborazione e la configurazione dell'applicazione. Il file è memorizzato in una directory del concentratore, nella stessa in cui si trova la pagina ASP.NET mobile. Il file di configurazione contiene una gerarchia nidificata di tag e sottotag XML con attributi che specificano le impostazioni di configurazione.

Una volta che è avvenuta l'opportuna configurazione del file web.config, passiamo a chiarire un altro tratto dell'applicazione. Lavorare con una mobilepage non risolve i

problemi illustrati nella prima fase, come ad esempio l'inserimento da parte dell'utente di dati che possono essere inesatti. ASP.NET ha un robusto meccanismo di convalida che posso usare per controllare facilmente gli input dell'utente per vedere se sono stati commessi errori e in caso necessario visualizzare messaggi di *warning* all'utente che possa correggerli. Vediamo il codice sottostante.

Attraverso il controllo **CompareValidator** posso controllare se quello che l'utente sta inserendo è un valore inferiore di 16. Oltre al controllo CompareValidator sono presenti altri due controlli: **Textbox** e **Command**. Il controllo Command genera un evento andando a richiamare la funzione scritta in C# Invia.

```
protected void Invia(Object sender, EventArgs e)
{
   if (Page.IsValid)
   {
      ActiveForm = Form2;
      Result.Text = String.Format("Il numero selezionato:
{0}", NumberEdit.Text);
   }
}


<mobile:CompareValidator          runat=server
ControlToValidate="NumberEdit"  Type="Integer"
ValueToCompare="16" Operator="LessThanEqual"> Il
numero è sbagliato. </mobile:CompareValidator>

<mobile:TextBox  id="Numero"  Numeric="true"
runat=server>
      *
      *
      *
<mobile:Command       OnClick="Invia"
runat=server>Invia</mobile:Command>


<mobile:Form id="Form2" runat=server>
  <mobile:Label id="Warning" runat=server/>
  <mobile:Link Text="Back" NavigateURL="#Form1"
```

5 La tecnologia WAP

In questo paragrafo verrà analizzato il protocollo WAP utilizzato per la realizzazione del nostro progetto di *Home Automation.* Verrà fornita anche una panoramica su alcuni toolkits che simulano un brownser WAP oltre al WAP Gateway adoperato.

Le informazioni reperibili attraverso Internet sono ormai innumerevoli ed accessibili, ad es. da un cellulare o da un altro strumento portatile tramite un protocollo (WAP: Wireless Application Protocol), per aumentare l'interattività, la comunicazione tra gli utenti e i servizi, e terminali in grado di gestirlo.

Standardizzazione WAP

L'idea di far navigare in Internet i cellulari è nata già alcuni anni fa da Unwired Placet (oggi Phone.com), che aveva sviluppato un protocollo *proprietario*, noto con il nome di HDML (Handheld Device Markup Language). Questo protocollo, seppur funzionante, non ha avuto seguito proprio perché *proprietario*. Infatti tutte le aziende avrebbero dovuto, per utilizzare l'HDML, pagare le royalty ad UP.

È nata allora l'idea di creare un gruppo super-partes che studiasse un protocollo unico, utilizzabile da tutti, e operante su qualunque piattaforma radio-mobile (GSM, CDMA, UMTS,...). Questo gruppo super-partes (noto come WapForum) oggi esiste: è una associazione no-profit nata nel Giugno 1997 dalla collaborazione tra Nokia, Motorola, Ericsson e Unwired lacet e conta oggi più di un centinaio di aziende. Il suo scopo è di definire un insieme di specifiche per lo sviluppo di applicazioni per dispositivi wireless, indipendenti dal particolare vendor.

Quindi WAP è il tentativo di creare uno **standard aperto** per i protocolli wireless con l'obiettivo non solo di creare un microbrowser in grado di garantire ottime prestazioni sul display dei cellulari, ma di realizzare una completa architettura a strati, simile a quella dell'ISO-OSI, oppure a quella del TCP-IP, che permetta, a chi realizza un servizio ad alto livello, di disinteressarsi completamente di come le cose vengano poi effettivamente strutturate a livelli sottostanti, di come vengano gestiti gli errori, le connessioni, le trasmissioni, le sessioni, le differenze tra fornitori del collegamento mobile, ecc.

Il WAP si posiziona dunque come punto di convergenza tra due tecnologie in rapida espansione: *i dati wireless* ed *Internet*. Si è rivelato necessario standardizzare un nuovo protocollo (il WAP appunto) per far sì che esso sia in grado di superare una serie di problematiche legate al trasporto sulle reti mobili di Internet e dei suoi servizi.

Quali sono i problemi principali che il WAP deve affrontare (e ovviamente risolvere)?

La maggior parte delle tecnologie Internet sono state sviluppate per poter funzionare su computer potenti e aventi a disposizione reti con un'ampia banda.

I dispositivi portatili (cellulari, PDA,...) sono caratterizzati invece da: CPU poco potenti, con poca memoria (ROM e RAM) e severe restrizioni sul consumo di potenza (batterie limitate), piccoli (se non piccolissimi) display; dispositivi di Input per l'utente differenti rispetto al Pc (si pensi al tastierino di un cellulare).

Inoltre le reti wireless tendono ad avere rispetto quelle wired: minore ampiezza di banda, maggiore latenza, stabilità di connessione minore;

Alcune di queste limitazioni sono riferibili sopratutto ai telefoni cellulari. Altri dispositivi mobili si comportano meglio. La dimensione dello schermo di un dispositivo palmare (vedi il *Palm* o lo *Psion*), ad esempio, è tale da poter eseguire elementari funzioni di text processing. Comunque, per forza di cose, qualsiasi dispositivo mobile è sottoposto in qualche misura alle limitazioni elencate sopra.

Infine per poter andare incontro alle richieste degli operatori delle reti mobili, la soluzione proposta deve avere le seguenti caratteristiche:

- *interoperabilità:* terminali di diverse industrie devono poter comunicare con le reti mobili attraverso questo servizio;
- *scalabilità:* gli operatori delle reti mobili devono essere in grado di poter fornire servizi personalizzati al cliente finale;
- *efficienza:* fornire servizi di qualità adattabili al comportamento ed alle caratteristiche delle reti mobili;
- *affidabilità:* fornire una piattaforma consistente e resistente per la distribuzione di servizi;
- *sicurezza:* proteggere i dispositivi ed i servizi da tutti i problemi che riguardano la sicurezza (quali ad esempio l'integrità dei dati, l'autenticazione)

Il WapForum ha effettivamente sviluppato una architettura che soddisfa i seguenti requisiti:

utilizza dove è possibile tutti gli standard già esistenti;
si basa su una architettura a livelli, scalare ed estendibile;
supporta la maggior parte delle reti mobili;
ottimizza i bearer (che rappresentano i possibili livelli di trasporto del protocollo WAP) aventi bande ristrette e alte latenze;
ottimizza le risorse dei dispositivi per un uso efficiente (piccole memorie, utilizzo della CPU, consumo di potenza);
fornisce supporti per applicazioni e comunicazioni sicure; abilita la creazione di interfacce MMI (Man Machine Interface) lasciando massimo controllo e flessibilità al venditore;

5.1 L'architettura WAP

Il modello di programmazione WAP è molto simile a quello per il World-Wide-Web (WWW). Ovviamente il modello WWW è stato ottimizzato ed esteso per soddisfare le caratteristiche delle reti mobili.

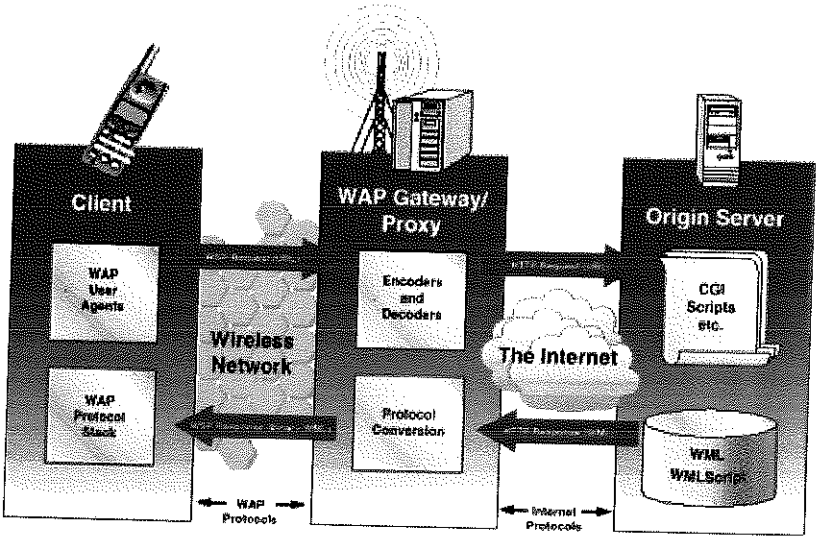


Figura 6 Il modello WAP

Il contenuto è trasportato usando un insieme di protocolli di comunicazione standard concettualmente simili ai protocolli di comunicazione WWW. Un microbrowser nel terminale wireless coordina l'interfaccia utente, risultando così analogo ad un Web browser.

Il protocollo definisce un insieme di componenti standard che abilitano la comunicazione tra terminali mobili e server di rete, tra cui:

- modello di naming standardizzato: per identificare il contenuto WAP sui server di origine sono usati i medesimi URL del WWW;

- tutti i dati WAP hanno uno specifico tipo che è compatibile con quello WWW. Ciò permette all'interfaccia utente (user agent) WAP di visualizzare ed elaborare correttamente i dati basati su questo tipo; i tipi di contenuto WAP e i protocolli sono stati ottimizzati per il mercato di massa dei dispositivi mobili, sfruttando la tecnologia dei Gateway per connettere il dominio wireless con il WWW e spostando su essa molto del peso computazionale necessario per l'ottimizzazione.

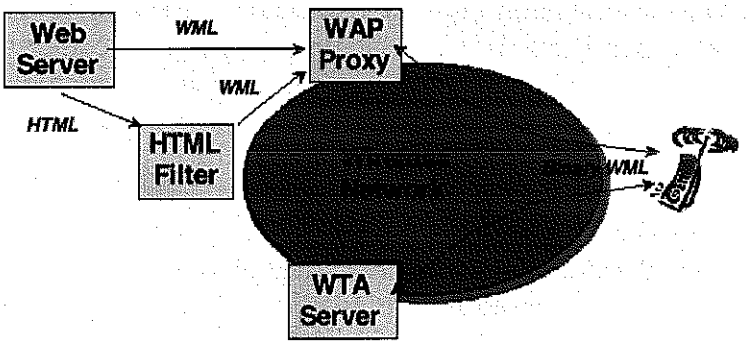
- i protocolli di comunicazione WAP consentono al browser di effettuare richieste verso i web server di rete.

Mentre l'uso classico di WAP prevede un Web server, un Gateway e un client WAP, l'architettura WAP può anche supportare altre configurazioni. Ad esempio è possibile creare un server di origine che includa le funzionalità del Gateway. Un tale server potrebbe essere usato per facilitare la realizzazione di soluzioni di sicurezza end-to-end o applicazioni che richiedono un miglior controllo degli accessi.

Nell'esempio di Figura 6 il client WAP comunica con due server della rete mobile: uno di essi è il WAP Proxy, che traduce le richieste WAP in richieste WWW, permettendo quindi al client WAP di dialogare con un qualsiasi Web server, e codificando le risposte provenienti dal Web server nel formato binario compatto comprensibile dal client. Se il Web server fornisce un contenuto WAP (ad esempio WML), il proxy lo recupera direttamente dal Web server, mentre invece se il Web server fornisce contenuto WWW (come HTML), viene utilizzato un filtro per tradurre il

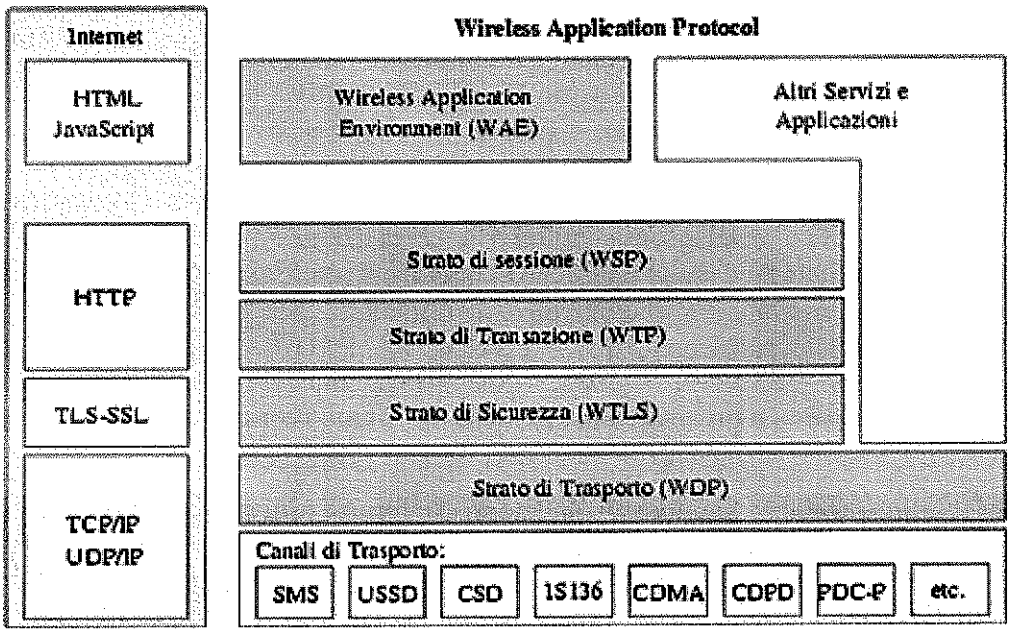
contenuto WWW in contenuto WAP: per esempio un filtro HTML può trasformare HTML in WML.

L'altro server WTA (Wireless Telephony Application), invece, è un esempio di server di origine o di Gateway che risponde direttamente a richieste provenienti dal WAP client: viene usato generalmente per fornire un accesso WAP alle infrastrutture di telecomunicazione del provider della rete wireless.



Esempio di rete WAP

L'architettura WAP fornisce un ambiente scalare ed estendibile.. Ogni livello dell'architettura è accessibile dai livelli sovrastanti, così come da ogni altra applicazione o servizi (vedi figura).



L'architettura WAP

L'architettura a livelli WAP abilita i servizi e le applicazioni ad utilizzare le caratteristiche del WAP stack attraverso un insieme ben definito di interfacce. Le applicazioni esterne possono accedere direttamente attraverso i livelli di sessione, transazione, sicurezza e di trasporto. Viene qui presentata una breve spiegazione delle funzionalità dei vari livelli dell'architettura.

WAE (Wireless Application Environment)

Il WAE è un ambiente basato sulla combinazione di tecnologie World Wide Web e Mobile Telephony. L'obiettivo primario di WAE è quello di stabilire un ambiente di interoperabilità che permetta agli operatori e ai service provider di costruire applicazioni e servizi che possano raggiungere una larga varietà di differenti dispositivi wireless in modo efficiente e semplice. WAE include un microbrowser che fornisce le seguenti funzionalità

- Wireless Markup Language (WML): linguaggio a marcatori leggero, simile a HTML, ma ottimizzato per l'uso in terminali mobili per non congestionare il canale di comunicazione, normalmente così ristretto, tra un client mobile e un server. Il protocollo WAP prevede la codifica di WML in una forma bytecode binaria a token, che può portare ad una consistente diminuzione della dimensione delle informazioni trasmesse al dispositivo client. Il Codificatore WML esegue la conversione del formato WML nel formato compresso chiamato WAP Binary XML (WBXML). Questa conversione in WBXML può essere realizzata al volo sul server, ma più spesso viene fatta off- line, se il contenuto non ha la necessità di essere cambiato dinamicamente durante una richiesta da parte di uno User Agent WAP.

– WMLScript: linguaggio di scripting leggero, simile a JavaScript™, anche se tra i due ci sono tante differenze che, come al solito, servono per ottimizzare la comunicazione su un canale di trasmissione wireless. La differenza maggiore consiste nel fatto che nelle specifiche di WMLScript sono stati definiti sia un formato bytecode sia il relativo interprete. WMLScript viene trasformato in bytecode e poi trasmesso al client mobile, mentre dal lato client c'è un interprete in grado di eseguire il bytecode stesso. WMLScript è un linguaggio tipato dinamicamente nel senso che è in grado di eseguire una conversione del tipo delle variabili, è case sensitive, permette l'uso delle funzioni, contiene vari controlli di flusso.

- Wireless Telephony Application (WTA) e Wireless Telephony Application Interface (WTAI): servizi di telefonia ed interfacce di programmazione; questa estensione fornisce una interfaccia alle classiche applicazioni telefoniche dei telefoni mobili, come ad esempio metodi per il controllo di chiamata, per l'invio di messaggi, per la manipolazione di rubriche telefoniche, ecc.

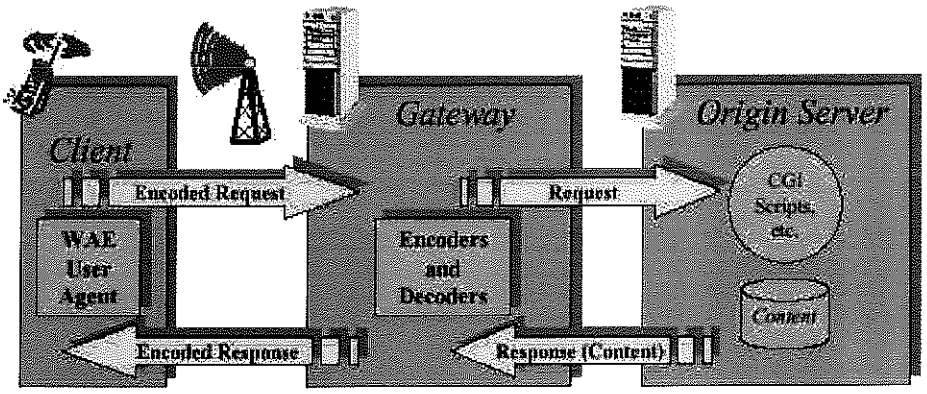
– Supporto per ben precisi formati di dati: immagini, rubriche telefoniche e calendario.

Il WAE è stato costruito senza fare alcuna assunzione sul tipo di modello MMI (Man Machine Interface) supportato, in modo da fornire agli sviluppatori solo un metodo generico per progettare le applicazioni WAP. La mappatura di una applicazione WAP in un particolare MMI diventa un compito specifico dello User Agent, che si occupa dell'interfaccia utente sul microbrowser. L'architettura WAE è rivolta principalmente agli aspetti client side dell'architettura WAP, cioè tutto ciò che riguarda l'interfaccia utente: è definita principalmente in termini di schemi di rete, formato dei dati, linguaggio di programmazione e servizi condivisi.

IL Modello WAE

WAE adotta un modello molto simile a quello WWW: tutti i dati sono specificati in formati simili; per quanto riguarda il trasporto degli stessi si utilizza un protocollo standard nel dominio WWW ed un protocollo HTTP-like nel dominio wireless.

WAE migliora alcuni standard WWW in modo da preservare le caratteristiche dei terminali e della rete mobile. WAE assume l'esistenza di un Gateway con funzionalità di codifica e decodifica dei dati verso e da il client mobile, come si può vedere in Figura: lo scopo della codifica dei dati è quello di minimizzarne la dimensione in modo da ridurre la banda occupata e l'energia richiesta in fase di computazione dei dati dal lato client.

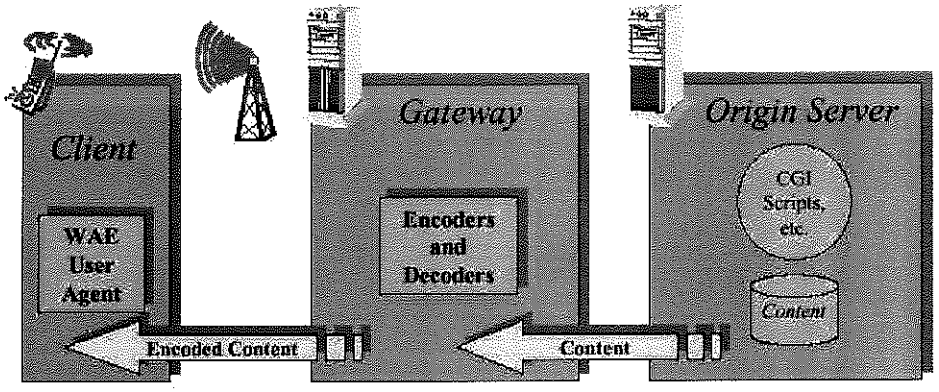


I principali elementi del modello WAE sono:

WAE User Agent (interfaccia utente): software client-side che permette la visualizzazione dei dati sul dispositivo dell'utente finale. L'interfaccia utente è integrata nell'architettura WAP ed è a tutti gli effetti come un browser: serve per la visualizzazione di WML codificato e WMLScript compilato;

generatore di contenuto: applicazione o servizio di un server di origine (come ad esempio un CGI script) che produce dati in formati standard, in risposta ad una richiesta proveniente dallo User Agent del terminale mobile

decodificatore standard dei dati: un insieme ben definito di codifiche dei dati permette allo User Agent (browser) di interpretare correttamente i dati. Lo standard di codifica include la codifica compressa del WML, la codifica dei bytecode per WMLScript ed un formato standard per le immagini ed altri tipi di dati.



Non è sempre vero, però, che tutti i dati ricevuti dal terminale arrivino in risposta a particolari richieste dell'utente. Il WTA, per esempio, è in grado di generare un meccanismo che permette al server di origine di distribuire dati al terminale senza che questi ne abbia fatto specifica richiesta: si tratta del modello basato sul Push illustrato in Figura.

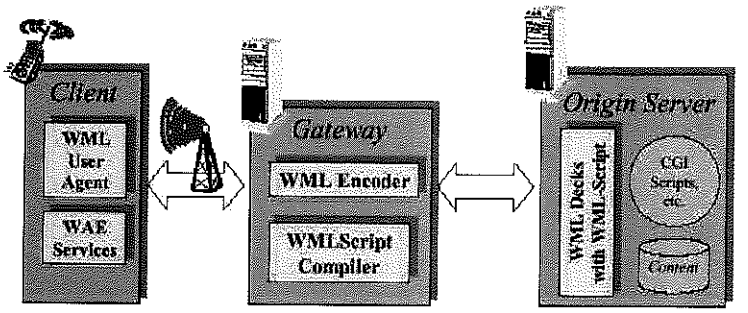
In alcuni casi, inoltre, ciò che il server di origine invia al terminale dipende dalle caratteristiche del dispositivo stesso: queste sono comunicate al server di origine attraverso un meccanismo standard di negoziazione che permette alle applicazioni di determinare le qualità peculiari dello User Agent (come ad esempio la versione WML/WMLScript supportata, il formato delle immagini supportato).

La situazione in cui un browser deve connettersi attraverso un proxy per raggiungere un server di origine è molto simile al caso dei dispositivi wireless in cui l'accesso al server avviene attraverso il Gateway. La maggior parte delle comunicazioni tra il browser ed il Gateway avviene usando WSP, disinteressandosi del protocollo di comunicazione del server di destinazione. L'URL, usato per trovare il contenuto desiderato, specifica sempre il protocollo di comunicazione usato dal server di origine ed è indipendente da quello usato tra il browser del dispositivo mobile ed il Gateway. Il Gateway è il responsabile delle conversioni di protocollo: in fase di richiesta traduce da WSP in un altro protocollo (quello del server) ed in fase di risposta dal protocollo usato dal server a WSP.

Per esempio pensiamo ad un utente con un telefono WAP che effettua una richiesta usando uno specifico URL. Il browser del terminale si connette al Gateway attraverso il protocollo WSP ed invia una richiesta GET per quell'URL specifico. Il Gateway individua, attraverso l'URL, l'indirizzo dell'host e crea con questo una sessione HTTP. Il server HTTP contattato esegue la richiesta ed invia una risposta al Gateway che la codifica e la restituisce al browser, tramite la sessione precedentemente aperta sul protocollo WSP.

L'esistenza del Gateway è obbligatoria, ma è possibile immaginare un server di origine che inglobi al proprio interno il WAP Gateway, con il relativo Codificatore WML e il relativo compilatore WMLScript.

Inoltre è possibile memorizzare staticamente (o in cache) servizi del server direttamente sottoforma di token WML e/o sottoforma di bytecode WMLScript per evitare la conversione al volo del deck. L'interazione tra l'utente ed il server d'origine avviene per mezzo di WML: quando un utente desidera accedere ad un particolare servizio del server di origine, gli invia una richiesta attraverso lo User Agent usando uno schema URL (ad esempio utilizzando il metodo HTTP GET). Il server risponde inviando un singolo deck (normalmente in formato testuale). Il codificatore WML (Encoder in Figura) residente nel Gateway (il Gateway in questo esempio è inglobato all'interno del server di origine) converte ogni deck WML nel corrispettivo formato binario che viene restituito al client, in modo che questi possa interpretarlo e visualizzarlo. Inoltre il Gateway può eseguire alcune ottimizzazioni, basandosi sulle caratteristiche del client (ciò avviene solo se vi è stata una effettiva negoziazione tra il client ed il Gateway).



Nel caso di WMLScript il concetto è del tutto analogo, con la sola differenza che il Gateway ha al suo interno un compilatore che compila la risposta del server in bytecode da inviare al client (WMLScript Compiler, in Figura).

WSP(Wireless Session Protocol)

Il WSP fornisce al livello di applicazione una interfaccia consistente per due differenti servizi di sessione : il primo è un servizio connection oriented che opera sopra il protocollo di livello transazionale (WTP), mentre il secondo è un servizio connectionless che opera sopra il servizio datagram (WDP).

La differenza essenziale tra le due classi di servizio è che il servizio connection oriented usa il protocollo WTP per realizzare un servizio affidabile, in cui non vengono persi i dati, attraverso un paradigma richiesta/risposta con ritrasmissione di dati perduti, ritrasmissioni selettive, segmentazione e riassemblamento di pacchetti voluminosi, indirizzamento con numero di porta, metodi per il controllo di flusso, connessioni di lunga durata con scambio di dati bidirezionale. Il servizio connectionless, invece, lascia viaggiare le informazioni in maniera completamente indipendente, disordinata, in cui i dati possono essere persi o duplicati. Entrambi i servizi possono operare su un canale sicuro (WTLS). Il WSP consiste di servizi adatti per applicazioni di browsing (WSP/B

L'implementazione del WSP segue fedelmente l'HTTP/1.1 per quanto riguarda la negoziazione delle modalità di trasmissione: di conseguenza tutti i suoi metodi sono supportati e inoltre le richieste spedite dal client al server e le corrispondenti risposte includono sia un'intestazione (header) sia i dati.

È prevista una codifica binaria compatta per tutti gli header generici (cioè gli header usati nella maggioranza dei casi) per ridurre l'overhead del protocollo. Una delle caratteristiche principali del WSP è quella che riguarda il tempo di vita di una sessione: questo tempo non è infatti legato in alcun modo al livello di trasporto sottostante. Inoltre una sessione può essere ripristinata su una rete di un portatore diverso da quello che era attivo prima della sospensione della stessa.

WTP(Wireless Transaction Protocol)

Il WTP svolge le sue funzioni al di sopra del servizio datagram: si tratta di un protocollo leggero ed orientato alla transazione, adeguato per una implementazione su client poco potenti. WTP opera in modo efficiente su reti datagram wireless sicure e non sicure. A differenza del TCP, che trasmette una grande quantità di informazioni per ciascuna transazione richiesta/risposta, il WTP non mantiene traccia dell'instradamento, perché esiste una unica possibile strada tra il WAP Gateway e il microbrowser wireless.

Il protocollo di transazione è definito per fornire i servizi necessari al browsing interattivo (richiesta/risposta). Durante una sessione, il client richiede informazioni ad un server, fisso o mobile, che risponde con i dati richiesti. WTP si appoggia direttamente al livello datagram (nel caso di connessioni non sicure), oppure al livello sicurezza WTLS (nel caso di connessioni sicure). I benefici derivanti dall'utilizzo del WTP sono:

- miglioramento dell'affidabilità su servizi datagram. WTP solleva i livelli più alti dal compito delle ritrasmissioni e degli acknowledgement che sono necessari quando si opera su servizi datagram.

- miglioramento dell'efficienza su servizi connection oriented.

Il WTP è "message oriented" ed è progettato per servizi orientati alla transazione, quali ad esempio il browsing; le due entità fondamentali che, dialogando tra loro, permettono la comunicazione su tale protocollo sono

- Initiator: provider WTP che dà inizio effettivamente una transazione;
- Responder: provider WTP che risponde ad una richiesta di inizio transazione.

WTLS(Wireless Transaction Layer Session)

WTLS è un protocollo di sicurezza che trova il suo fondamento nel protocollo industriale standard Transport Layer Security (TLS), formalmente noto come Secure Sockets Layer (SSL). WTLS viene proposto insieme ai protocolli di trasporto WAP, dopo essere stato ottimizzato per l'uso su canali di comunicazione con banda ristretta. Le applicazioni possono abilitare/disabilitare WTLS a seconda dei requisiti di sicurezza e delle caratteristiche della rete sottostante (ad esempio la privacy può essere disabilitata su reti che già garantiscano questo servizio ad un livello più basso..

Modello di sicurezza WAP

L'architettura WAP però non è completamene adattabile al modello di sicurezza del Web per i seguenti due motivi:

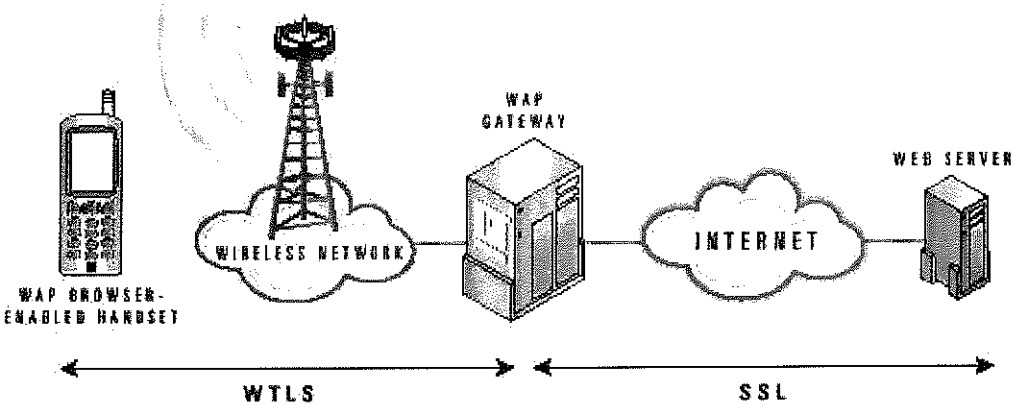
1 il protocollo SSL è stato progettato per comunicazioni di tipo wired (ampia banda a disposizione e basse latenze) che vedono coinvolti desktop con elevate capacità di calcolo e memorizzazione: una transazione SSL con un terminale WAP comporterebbe notevoli ritardi nella comunicazione andando a incidere enormemente sia sulle prestazioni sia sui costi della comunicazione);

2 il mondo WAP non prevede una comunicazione diretta tra il client ed il Web Server: tra loro vi è sempre il WAP Gateway che fa da tramite tra il terminale mobile ed il Web Server.

Il WTLS è stato appositamente progettato per i terminali mobili, tenendo in considerazione i loro limiti fisici. Tale protocollo garantisce comunque un buon livello di sicurezza, riducendo enormemente gli overhead del protocollo SSL.

Il problema legato alla presenza del WAP Gateway è stato invece risolto facendo utilizzare a quest'ultimo il protocollo SSL per comunicare in modo sicuro con il Web Server, mentre per comunicare con il WAP browser utilizza il protocollo WTLS.

La necessità di passare da WTLS a SSL (e viceversa) è resa obbligatoria dalla natura delle comunicazioni wireless, che sono caratterizzate da una banda di trasmissione ristretta ed una elevata latenza. Il modello di sicurezza WAP è mostrato in Figura



Si noti che tale soluzione non porta alla realizzazione di un unico canale sicuro tra il client ed il Web Server (come avviene nel mondo WWW quando si utilizza il protocollo SSL), ma si creano due canali sicuri separati: il primo, tramite il protocollo WTLS, collega il WAP browser al WAP Gateway, mentre il secondo, tramite il protocollo SSL, collega il WAP Gateway al Web Server.

Il punto critico di tale modello è proprio il WAP Gateway. Il WAP Gateway, infatti, è il punto di collegamento tra i due canali ed è il responsabile delle traduzioni WTLS/SSL (e viceversa): qui i dati, seppur per brevi istanti, passano in chiaro, perché per tradurre, ad esempio, il WTLS in SSL occorre prima decifrare i dati in WTLS portandoli in chiaro e poi cifrarli in SSL.

Tali traduzioni, necessarie al fine di permettere una connessione sicura e virtuale tra i due protocolli, richiedono tempo e necessitano di una memorizzazione (seppur temporanea) nella memoria del WAP Gateway.

Affinché un certo livello di sicurezza sia garantito, diventa allora assolutamente indispensabile che:

- i WAP Gateway non memorizzino mai informazioni decrittografate su mezzi di memorizzazione secondari;

– il processo di traduzione sia il più veloce possibile in modo da eliminare velocemente i dati residenti nella memoria volatile interna del WAP Gateway;

– si garantisca una sicurezza fisica del WAP Gateway, in modo che vi possano accedere solo amministratori autorizzati;

– siano limitati gli accessi remoti al WAP Gateway per effettuare operazioni di manutenzione.

Per evitare i problemi legati alle traduzioni WTLS/SSL la tendenza attuale da parte dei fornitori di servizio è quella di inglobare il WAP Gateway all'interno del proprio Web Server: tale soluzione, se da una parte aumenta il livello di sicurezza, dall'altra limita le prestazioni della navigazione Internet.

Prendiamo infatti in considerazione il Nokia 7110 (il primo cellulare entrato in commercio con supporto WAP). Per effettuare la navigazione Internet occorre settare, prima di cominciare il collegamento, l'indirizzo IP del WAP Gateway attraverso il quale le richieste saranno trasferite in HTTP al Web Server specificato. Una volta settato l'indirizzo IP del WAP Gateway si può effettuare il classico collegamento Internet specificando il numero del fornitore di servizio, con il quale abbiamo l'accesso alla rete, e la classica coppia username e password, attraverso le quali ci autentichiamo con il provider stesso.

A questo punto se non si è interessati ad una comunicazione sicura ed il WAP Gateway specificato prima del collegamento ha una visone globale della rete, la navigazione Internet avviene in modo del tutto analogo a come la conosciamo.

Supponiamo invece di voler comprare un libro e di voler effettuare una operazione di trading on- line: non possiamo fare le due operazioni in un unico collegamento, se si desidera che queste siano realizzate in modo sicuro e che i due Web server interessati incorporino anche due diversi WAP Gateway. Infatti prima settiamo il 7110 con l'indirizzo IP del Gateway gestito dal server che vende i libri, ci colleghiamo ed acquistiamo il libro. Dopodiché siamo costretti a disconnetterci, cambiare l'indirizzo IP del WAP Gateway inserendo quello gestito dalla banca con cui fare l'operazione di trading on- line, e infine ci colleghiamo ed eseguiamo l'operazione.

Oltre al discorso della sicurezza va notato che un particolare ISP che gestisce un Gateway ne può anche limitare la visibilità nei confronti della rete. Un portale, ad esempio, che gestisce anche un WAP Gateway può avere particolari interessi economici a limitare la visibilità della rete solo nei confronti di alcuni siti/servizi con i quali ha stipulato particolari contratti di sponsorizzazione.

L'alternativa rimane quella della gestione completa e accurata del WAP Gateway e delle traduzioni che esso deve effettuare rispettando i vincoli sopra descritti. Tale soluzione ha sicuramente il grande vantaggio di disaccoppiare il Gateway da un particolare fornitore di servizi (evitando i problemi di prestazioni esistenti per la soluzione che prevede l'accoppiamento Gateway/Web Server) ma richiede una amministrazione più difficoltosa.

A prescindere comunque dalla gestione dei WAP Gateway, rimane immutata la fiducia che il client deve avere nei loro confronti. Ovviamente il client, prima di riporre la propria fiducia (e magari il proprio denaro) verso un particolare WAP Gateway, deve accertarsi della sua identità analizzandone accuratamente il certificato. È bene chiarire inoltre che l'unico certificato che il client riceve è proprio quello del WAP Gateway con cui comunica e non quello del Web Server che restituisce i servizi richiesti.

Anche l'autenticazione infatti è spezzata dalla presenza del WAP Gateway: il client sceglie il WAP Gateway e ne verifica l'attendibilità, mentre il WAP Gateway a sua volta verificherà l'attendibilità dei siti con cui il client vorrà comunicare. Supponiamo, per chiarire la situazione, che un client debba acquistare on-line un libro su Amazon appoggiandosi al WAP Gateway di O.T.Consulting: il client non può verificare direttamente che il sito cui destina il proprio denaro sia proprio Amazon e non uno shadow server; tale verifica, infatti, la può svolgere solo il WAP Gateway. È proprio ora che entra in gioco il rapporto di fiducia client/WAP Gateway: il client, una volta che ha verificato che il WAP Gateway con cui comunica è proprio quello di O.T.Consulting, non si pone il problema della veridicità dei dati che questi gli fornisce.

Il WTLS fornisce tutta una serie di meccanismi che permettono la mutua autenticazione tra il client ed il WAP Gateway (anche quest'ultimo, infatti, può richiedere il certificato del client per controllarne l'identità). L'autenticazione del client da parte del WAP Gateway è però ancora poco utilizzata per via dell'impossibilità da parte dei dispositivi wireless attualmente in commercio di memorizzare certificati. In futuro tale problema sarà superato integrando dispositivi di memorizzazione simili alle Smartcard nei terminali mobili.

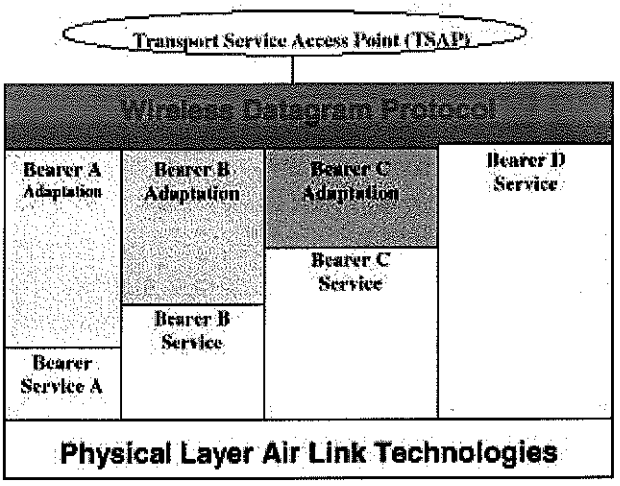
Le problematiche relative al modello di sicurezza WAP attuale sono allo studio del WAP Forum e l'obiettivo prefissato è quello di realizzare un meccanismo di sicurezza punto-punto tra WAP Browser e Web Server. Fino a che non verrà raggiunto tale obiettivo il WTLS fornirà una connessione sicura solo tra il WAP Gateway ed il WAP Browser garantendo privacy, integrità e autenticazione tra se stesso ed il client WAP. Inoltre i fornitori di WAP Gateway possono implementare dei meccanismi per permettere l'utilizzo della certificazione del client (se il dispositivo lo consente) e dell'uso di firma digitale per garantire anche il concetto di non ripudio.

5.2 WDP (Wireless Datagram Protocol)

Il protocollo a livello di trasporto nell'architettura WAP è il Wireless Datagram Protocol (WDP) che opera a stretto contatto con i servizi di trasporto dei portatori supportati dalle diverse tecnologie di rete wireless. Configurandosi come servizio di trasporto a carattere generale, WDP offre un servizio consistente agli strati superiori del protocollo, comunicando, in modo trasparente, con uno dei tanti portatori (bearer) disponibili. Grazie al fatto che il protocollo WDP fornisce una interfaccia

comune ai protocolli di livello più alto, i protocolli di sicurezza, sessione e applicazione sono in grado di funzionare in modo indipendente dalla rete wireless sottostante: questo risultato viene raggiunto adattando il livello di trasporto alle specifiche caratteristiche del portatore sottostante.

Conservando consistente l'interfaccia del livello di trasporto e le sue caratteristiche basilari, pur cambiando l'implementazione interna (in un'ottica simile a quella ad oggetti), si può raggiungere l'obiettivo di una interoperabilità globale. Nella Figura è mostrato un modello di architettura per il WDP.



Il protocollo WAP è progettato per operare su una varietà di differenti servizi forniti dai portatori (GSM, CDPD, Mobitex, CDMA), inclusi quelli per gli Short Message (SMS), per i dati a commutazione di circuito e per i dati a pacchetto (GPRS). I portatori offrono differenti livelli di qualità di servizio nel rispetto di parametri quali lo throughput, la frequenza di errore e i ritardi: i protocolli WAP sono progettati proprio per compensare o tollerare questa variabilità nel livello del servizio.

Dal momento che il livello WDP porta alla convergenza tra il servizio del portatore e il resto dello stack WAP, le specifiche WDP devono fornire la lista dei portatori supportati e le tecniche usate per permettere ai protocolli WAP di funzionare sopra qualsiasi portatore. Naturalmente la lista dei portatori supportati cambia nel tempo, con l'aggiunta di nuovi portatori, man mano che il mercato wireless si evolve.

L'architettura a livelli di WAP permette anche ad altri servizi ed applicazioni di utilizzare le caratteristiche dello stack WAP attraverso un insieme di interfacce ben definite.

Le applicazioni esterne possono accedere ai livelli di sessione, transazione, sicurezza e trasporto direttamente: questo permette allo stack WAP di essere usato per applicazioni e servizi non correntemente specificati da WAP, ma giudicati validi per il mercato wireless. Per esempio, le applicazioni come la posta elettronica, i calendari, le rubriche telefoniche e il commercio elettronico, o i servizi, come le pagine gialle, possono essere sviluppati per l'uso sul protocollo WAP.

Questo è in linea con i principi enunciati dal WAP Forum, secondo i quali la tecnologia WAP dovrà essere utile per applicazioni e servizi anche al di là di quelli specificati nelle specifiche. Il livello di trasporto nell'architettura WAP consiste nei livelli WTP (Wireless Transaction Protocol) e WDP (Wireless Datagram Protocol).

In Figura sono evidenziati le differenti funzioni (diverse altezze nella Figura) offerte dai portatori e quindi i diversi protocolli WDP necessari per garantire lo stesso livello di servizi.

WDP supporta diverse istanze simultanee di comunicazione da un livello superiore ad un singolo servizio portatore sottostante. Il numero di porta identifica l'entità del livello superiore sopra il WDP: questa entità può essere un altro protocollo (come il WTP o il WDP) o una applicazione (come ad esempio la posta elettronica). Il livello Adaptation è il livello del WDP che mappa le funzioni del protocollo direttamente sopra le caratteristiche specifiche del portatore; il livello Adaptation inoltre è diverso per ogni tipo di portatore.

Esiste una entità di gestione del WDP che è usata come interfaccia tra il livello WDP e l'ambiente del dispositivo: questa entità fornisce informazioni al livello WDP riguardo eventuali cambiamenti avvenuti nell'ambiente del dispositivo che potrebbero influire sul corretto funzionamento del WDP stesso.

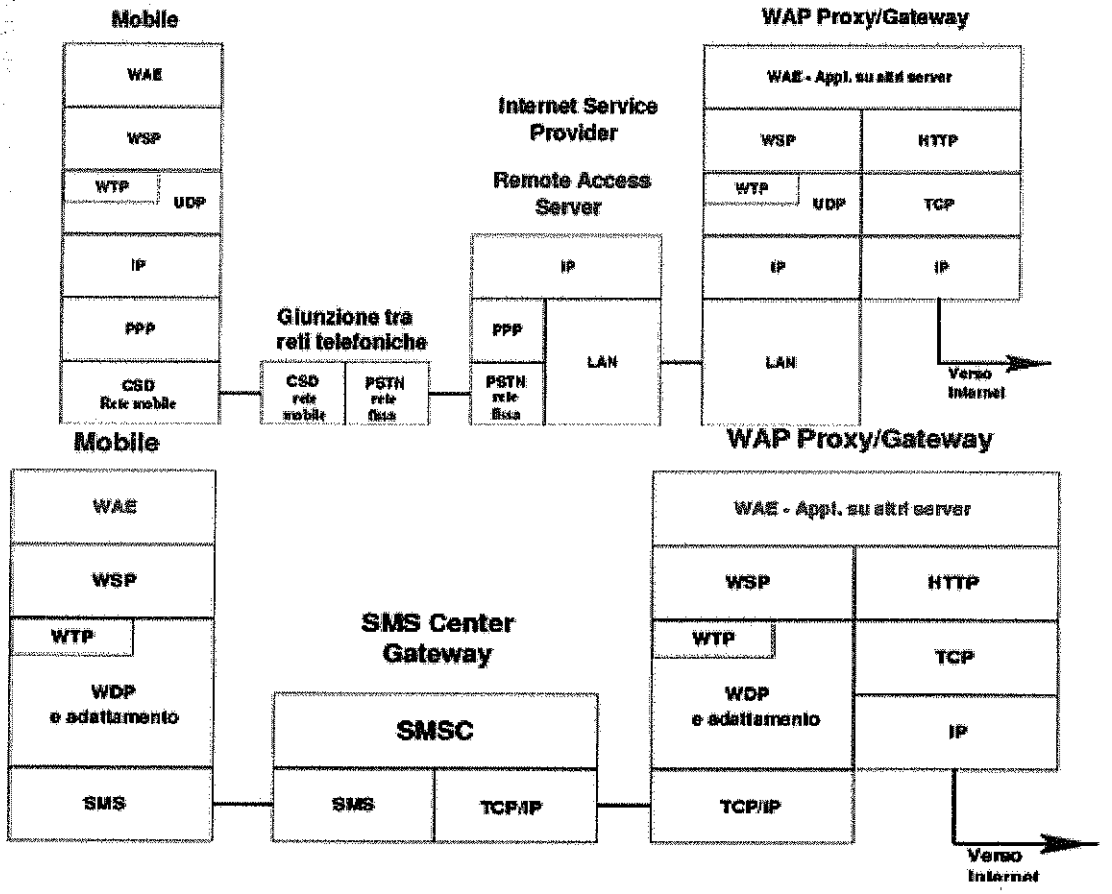
L'entità di gestione del WDP svolge le stesse funzioni ed ha le stesse caratteristiche dell'entità di gestione del WTP. Il WDP inoltre utilizza il Wireless Control Message Protocol (WCMP) per la gestione ed il trattamento degli errori in modo da migliorare le prestazioni del protocollo WAP e delle applicazioni.

I profili del WDP a seconda del tipo di portatore sottostante

WDP è il livello di trasporto nello stack dei protocolli WAP. È l'analogo di IP (o UDP/IP) e opera sopra i canali di trasporto pre-esistenti supportati dai vari tipi di rete. Essendo un servizio di trasporto generale, WDP isola i livelli superiori dalle specificità della rete sottostante, offrendo un modello comodo e consistente per lo sviluppo delle applicazioni indipendentemente dal tipo di rete radiomobile utilizzata. Questo risultato è ottenuto adattando i parametri del protocollo (ad esempio i tempi di timeout) alle specificità del mezzo.

WDP è pensato per funzionare con una gran varietà di canali di trasporto dei dati, compresi SMS, reti a commutazione di circuito e reti a pacchetto. I diversi canali offrono diversi livelli di qualità del servizio in riferimento a throughput, probabilità di errore, ritardi e diversi livelli di astrazione in termini di funzioni e procedure di handshake. WDP cerca di compensare e tollerare queste differenze aggiugendo ad ogni canale di trasporto le funzionalità di cui è privo. L'interoperabilità tra reti che adottano canali di trasporto diversi (es. tra operatori di telefonia diversi) si può facilmente ottenere per mezzo di gateway intermediari.

La specifica WDP elenca i canali che sono supportati e le tecniche per permetterne l'uso. Tale lista cambierà nel tempo mano a mano che nuovi canali saranno aggiunti. Nelle figure sottostanti vediamo per esempio la realizzazione di connessioni WAP sui canali CSD e SMS.



6 I toolkit di sviluppo e i WAP Gateway

I principali collaboratori del WAP Forum mettono a disposizione diversi ambienti di sviluppo per applicazioni WAP. I toolkit analizzati sono quelli proposti dalla Nokia, dall'Ericsson, dalla Motorola, dalla Microsoft e ultimamente il Kannel che, dopo una valutazione comparativa, useremo. Gli SDK simulano un generico cellulare WAP. In pratica si tratta di un browser in grado di interpretare il codice WML e WMLScript o addirittura, nel caso dei programmi della Nokia e dell'Ericsson, di interpretare il bytecode cioè la codifica in codice binario compatto particolarmente adatto per reti a banda ristretta.

Tali ambienti consentono quindi di visualizzare deck/card WML e verificare che le applicazioni progettate siano funzionanti. È utile, al fine di realizzare delle simulazioni il più possibile vicino alla realtà procurarsi un web server per accedere direttamente tramite HTTP alle applicazioni WAP (ad esempio nel toolkit dell'Ericsson viene fornito il web server per Windows 95/98/NT "Xitami v2.4d3). Qualunque sia il web server utilizzato è sempre necessario configurare i tipi MIME come si può vedere in tabella.

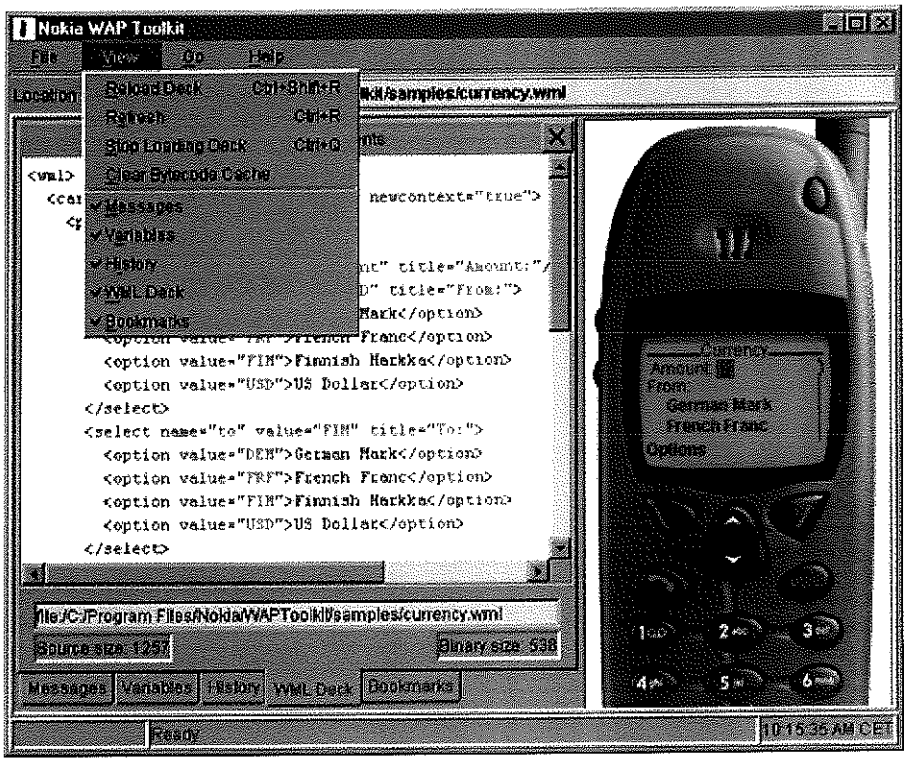
Contenuto	Tipo MIME	Estensione
Sorgente WML	text/vnd.wap.wml	wml
Wireless bitmap	image/vnd.wap.wbmp	Wbmp
WML Bytecode	application/vnd.wap.wmlc	Wmlc
Sorgente WMLScript	text/vnd.wap.wmlscript	Wmls
WMLScript Bytecode	application/vnd.wap.wmlscriptc	wmlsc

6.1 Il toolkit della NOKIA

Il WAP SDK della Nokia per Windows 95/98/NT può essere scaricato gratuitamente dal sito della Nokia previa registrazione. E' un software scritto interamente in Java e richiede il JRE 1.2.2 (Java Runtime Environment). Al momento dell'inizio di questo lavoro la versione più recente di Nokia Wap Toolkit è la 1.2. Ora è presente la versione 3.0 ma noi faremo riferimento alla versione 1.2 con cui abbiamo lavorato.

Il toolkit è costituito dai seguenti componenti:

- Browser WML, in grado di interpretare codice WML e WMLScript
- WAP phone user interface (Nokia 6110, 6150, 7110)
- Compilatore WML e WMLScript
- Editore WML, WMLScript, e WBMP
- Moduli di accesso per protocollo WAP, HTTP
- Informazioni di Debug



Il toolkit permette di visualizzare sei finestre:

- *Messagges*: visualizzazione delle operazioni di request e respond che vengono effettuate; vengono inoltre segnalati gli errori (utile per la fase di debug).
- *Variables*: visualizza le variabili, e i relativi valori, che sono utilizzate nel corrente codice WML o WMLScript; in questa finestra tali variabili possono anche essere settate per facilitare l'input dei dati al posto di utilizzare il tastierino del simulatore di cellulare.
- *History*: mostra l'"history stack " cio`e tutte le card visitate nella corrente sessione di lavoro.
- *WML Deck*: mostra il codice e le dimensioni del corrente WML o WMLScript. E' possibile inoltre visualizzare il bytecode (e la sua dimensione).
- *Bookmark*: contiene i bookmark del browser.
- *Session*: contiene tre colonne che mostrano la lista degli URL caricati, il tipo MIME dell'URL caricato e le dimensioni in byte dell'URL.

Vi `e inoltre la finestra di editor con la quale si possono creare e modificare sorgenti WML, WMLScript e immagini bitmap (WBMP). Quando si carica, nell'editor, una pagina WML o WMLScript `e possibile farne la compilazione per verificare se vi sono errori di sintassi e per generare il relativo bytecode.

La modifica delle opzioni (preferences)

E' possibile modificare le opzioni accedendo a tre tabs:

1. Comunication
2. Encoding
3. General

Con la modifica dei parametri di comunicazione (punto 1) si può scegliere se utilizzare l'HTTP diretto per accedere agli URL attraverso la rete oppure utilizzare il protocollo WSP per comunicare con un WAP Gateway che ha il compito di prelevare gli URL dalla rete per conto suo e trasferire il relativo contenuto, già codificato in bytecode, al toolkit. Il primo metodo è semplice e diretto mentre il secondo simula più fedelmente il comportamento dell'attuale percorso di trasmissione per i dispositivi che supportano il protocollo WAP.

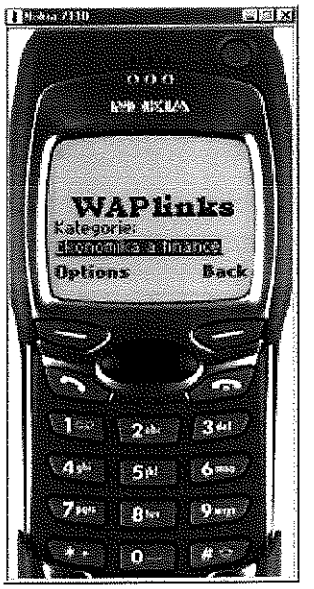
Se si sceglie di utilizzare HTTP, il toolkit della Nokia adatta direttamente le richieste HTTP. Vi sono alcune opzioni che si possono abilitare o disabilitare:

- Utilizzo di cookies
- Utilizzo dell'HTTP basic authentication protocol quando si riceve un errore di non autenticazione.
- Se si deve passare attraverso un proxy server `e necessario inserire l'indirizzo nell'apposito campo.

Se si comunica con un WAP Gateway,invece, tutte le richieste sono codificate secondo il protocollo WAP e inviate WAP Gateway. È necessario quindi specificare l'indirizzo del Gateway, il tipo di connessione (Connection-oriented o Connectionless) e la porta sulla quale il Gateway si trova in ascolto (di default 9200 per modo Connectionless, 9201 per modo Connection-oriented).

La modifica dei parametri di codifica (punto 2) consente di scegliere il tipo di codifica da effettuare. Se si abilita la codifica rapida potranno essere visualizzati solo un numero limitato di errori; se si desidera vedere tutti i messaggi di errore non si deve abilitare tale opzione.

Inoltre in questa tab si può scegliere il "Character Set " che il toolkit utilizzerà di default se questo non è specificato nell'header HTTP. Nella sezione "General " si pu`o scegliere se abilitare l'utilizzo dell' "element access " del WML che specifica il controllo di accesso ad altre deck: utilizzare questa opzione può causare difficoltà in caso di testing..



Se si sceglie di utilizzare il modello 7110 bisogna tener presente che:

Infine possiamo indicare il numero di entry che possono essere visualizzate nella "History View " e il tipo di cellulare da usare per le simulazioni. Si può scegliere di visualizzare tre diversi tipi di cellulari: il 6110 e il 6150 che non sono dei prodotti reali, e il 7110 che invece è già in vendit. Alcune funzioni del toolkit sono disabilitate (ad esempio le viste di WML Deck e Session).. Si può accedere a deck e card solo attraverso un WAP Gateway.

Il toolkit mette a disposizione, inoltre, un efficiente tool per la gestione delle immagini bitmap definite nello standard WAP: con tale tool è possibile creare e modificare immagini che possono manipolate per essere poi visualizzate su dispositivi mobili

6.2 Il WAP Gateway Open Source

L'architettura WAP ha come suo elemento fondamentale come detto nei precedenti discorsi nel WAP Gateway che ha il compito di fare da tramite tra il client ed i dati ospitati sui web server. Vista l'importanza di tale elemento nell'architettura WAP le grandi società coinvolte nello sviluppo di applicazioni e servizi wireless (Nokia su tutte) non hanno perso tempo nello sviluppare WAP Gateway commerciali.

La nascita del progetto Kannel (Giugno 1999) ad opera di una ditta di servizi wireless finlandese (la WapIT Ltd.) prevede la realizzazione di un WAP Gateway Open Source perfettamente funzionante e stabile servendosi della collaborazione di tutti gli interessati a livello mondiale. Dalla prima versione del WAP Gateway si sono susseguite ad una velocità impressionante gli aggiornamenti e le nuove versioni.

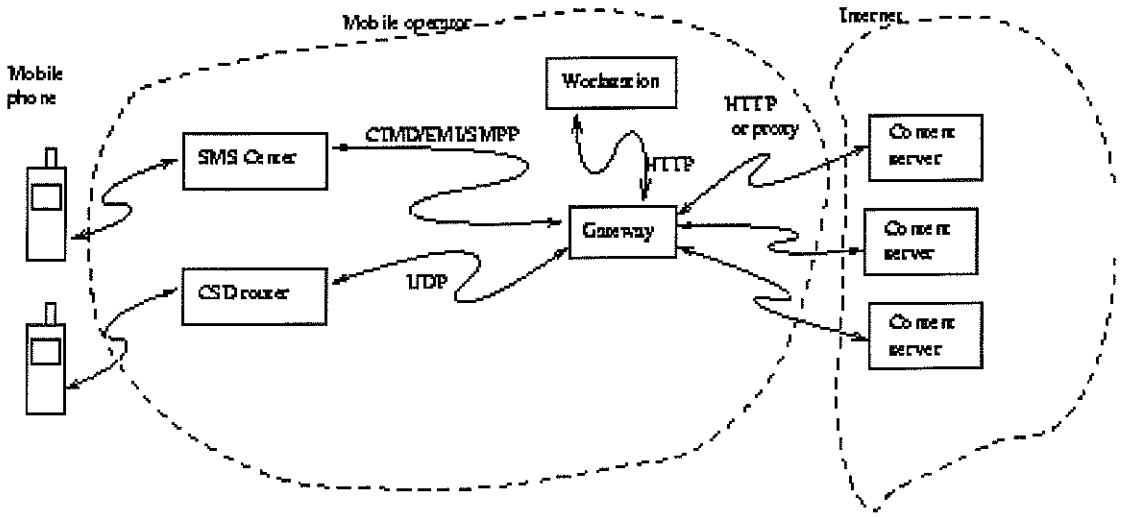
Il Kannel, inoltre, non è solo WAP Gateway ma funziona anche come SMS Gateway. Il gateway gestisce solo connessioni di tipo connection-oriented e quindi si pone in ascolto della porta 9201. Il Kannel, inoltre, non forniva all'inizio di questo lavoro il supporto di sicurezza del protocollo WAP (WTLS). Solo da pochi mesi è presente in Kannel il livello di sicurezza.

L'architettura del Kannel

Requisiti del Gateway

Il Gateway deve essere in grado di servire centinaia di utenti concorrenti è quindi necessario un Pc dotato di almeno un processore di 400Mhz, una Ram di almeno 128 Megabyte ed una interfaccia di rete di 10Mbit/sec. Il Gateway è nato per funzionare su piattaforma Linux (caldamente consigliate le verioni Red Hat 6.x) anche se oggi è presente una versione per Windows.

Interfaccia esterna del Gateway



Il Gateway in linea di principio può comunicare con diverse entità (figura): centri SMS, router CSD, una workstation per configurare e monitorare il funzionamento del gateway (questa può anche essere considerata come parte del gateway ma in realtà non lo è) content server.

Le comunicazioni con i cellulari attraverso i router CSD utilizzano pacchetti di tipo UDP, mentre lo scambio di informazioni con la workstation e con i content server avviene via-HTTP. Il Gateway si comporta rispettivamente come server o come client a seconda dell'entità con cui comunica.

Componenti interne al Gateway: Hosts, Moduli e Threads

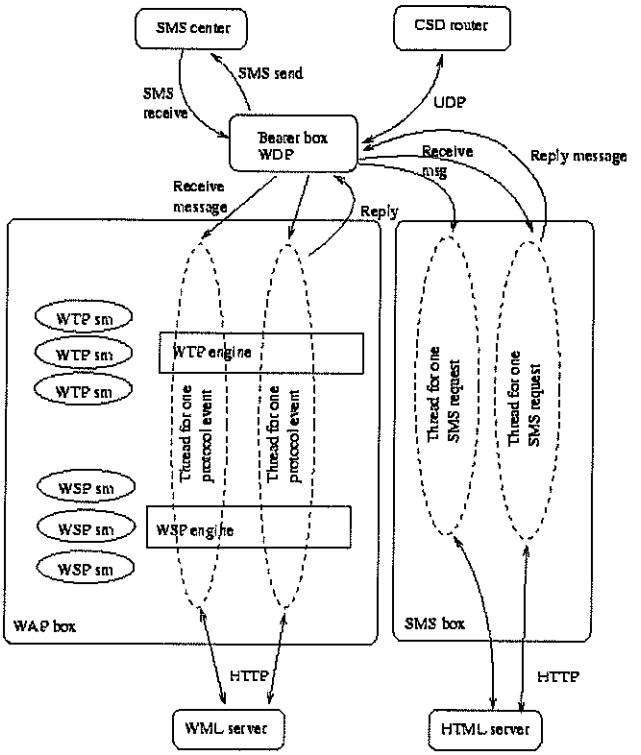
Il Gateway divide il carico computazionale su vari host che possono essere di tre differenti tipi (figura):

- Bearer Box: connette i centri SMS ed i CSD router e fornisce un'interfaccia unica agli altri box. Implementa il livello WDP del WAP stack;
- WAP Box: esegue i livelli superiori al WDP del WAP stack (WTP e WSP, non gestisce il livello WTLS). Ogni sessione e tutte le transazioni che appartengono a quella sessione sono gestite dallo stesso WAP Box. Le sessioni e le transazioni non possono migrare tra i vari WAP box;
- SMS Box: esegue il Gateway SMS.

Il bearer box riceve i messaggi SMS da un centro SMS, li controlla per verificare se si trattano di pacchetti WAP o messaggi SMS e quindi li instrada al box appropriato. Il bearer box tiene traccia di tutti i box, WAP e SMS, che sono correntemente attivi e manda allo stesso box tutti i messaggi che arrivano da un particolare terminale. In maniera simile riceve i pacchetti UDP da un CSD router e li instrada al box associato al

sender. Il bearer box inoltre invia anche i messaggi SMS ed i pacchetti UDP che sono generati dagli altri box.

Questa architettura prevede, quindi, la presenza di un unico bearer box e molteplici WAP box che possono risiedere anche su macchine diverse.



Il Wap box implementa i livelli WTP(transaction) e WSP(session). Questi due livelli ricevono le richieste dal dispositivo mobile, le traducono in richieste HTTP e le inviano ai content server. In ricezione invece traducono le risposte HTTP dei content server, le comprimono e le trasmettono ai dispositivi mobili.

Il livello WTLS non è implementato.

Inter-Component Communication

Di seguito sono descritte le comunicazioni tra i vari componenti del gateway. La connessione tra il bearer box ed i WAP box è una connesione di tipo TCP stream. Il bearer box si comporta come un server ed è dunque in attesa che i WAP box si connettino a lui. Per tenere traccia dei WAP box che si sono connessi al bearer box questo utilizza una lista dinamica. Questa architettura rende inoltre semplice aggiungere nuovi WAP box, al sistema, on the fly: `e sufficiente installarne uno nuovo e farlo connettere al bearer box.

Se lo stream TCP tra il bearer box ed il WAP box si "rompe ", il bearer box deve togliere dalla lista dei client il WAP box con cui si è persa la connessione. Se un WAP box in seguito ad un crash è stato rimosso dalla lista dei client, i pachetti provenienti dal dispositivo mobile e che sono a lui destinati vengono trattati come pacchetti per un nuovo client, cioè un altro WAP box attivo nel sistema.

Un box può anche entrare in uno stato catatonico in cui, pur non essendo andato in crash, non è più in grado di rispondere. Per accorgersi di questa situazione i WAP box,

all'interno del gateway (cioè connessi ad un bearer box), inviano dei pacchetti denominati di heartbeat. In pratica ad intervalli regolari di tempo i WAP box inviano al bearer box dei pacchetti per comunicargli il loro stato.

Se il bearer box non riceve questi messaggi assume che il WAP box in questione sia andato in crash anche se non si è rotta la connessione TCP e quindi chiude il collegamento e rimuove il WAP box dalla lista di quelli attivi. I messaggi heartbeat contengono inoltre informazioni sul carico di lavoro che, al momento dell'invio, il sender (WAP box) sta gestendo; il bearer box utilizza tali informazioni per bilanciare l'utilizzo delle risorse.

Struttura dei messaggi ed implementazione La struttura dei messaggi che si scambiano i vari componenti all'interno del gateway `e sempre la stessa. In particolare questi iniziano sempre con un campo tipo seguito da altre strutture che costituiscono i campi del messaggio. La struttura, chiamata Msg, è la seguente:

```
struct {
    enum msg_type type;
    struct heartbeat
   {
     long load;
   } heartbeat;
   struct smart_sms
   {
   OpaqueVector *sender;
         OpaqueVector *receiver;
   long flag_8bit;
   long flag_udh;
         OpaqueVector *udhdata;
   OpaqueVector *msgdata;
   long time;
   } smart_sms;

   struct wdp_datagram
   {
     OpaqueVector *source_address;
     int source_port;
     OpaqueVector destination_address;
     int destination_port;
     OpaqueVector *user_data
   } wdp_datagram;
 } Msg;
```

Msg, tante struct quanti sono i tipi di messaggi che si possono passare i vari componenti del gateway; il campo type indica, quindi, a quale delle strutture interne fare riferimento.

Messaggi Heartbeat

I box SMS e WAP inviano messaggi heartbeat al bearerbox per informarlo che sono "vivi" e comunicargli il loro carico. Il campo load indica il numero di richieste HTTP che il sending box (WAP o SMS) ha aperte nel momento in cui invia il messaggio heartbeat.

Messaggi SMS tra il bearer box e il SMS box

Il bearer box invia messaggi SMS al box SMS e questi gli risponde con nuovi messaggi SMSI:

Messggi WDP tra il bearer box e il WAP box

Il bearer box invia messaggi WDP al WAP box e questi gli risponde con nuovi messaggi WDP. I messaggi WDP sono costituiti dalle seguenti parti di informazione:

- Source address
- Source port
- Destination address
- Destination port
- Dati nel messaggio

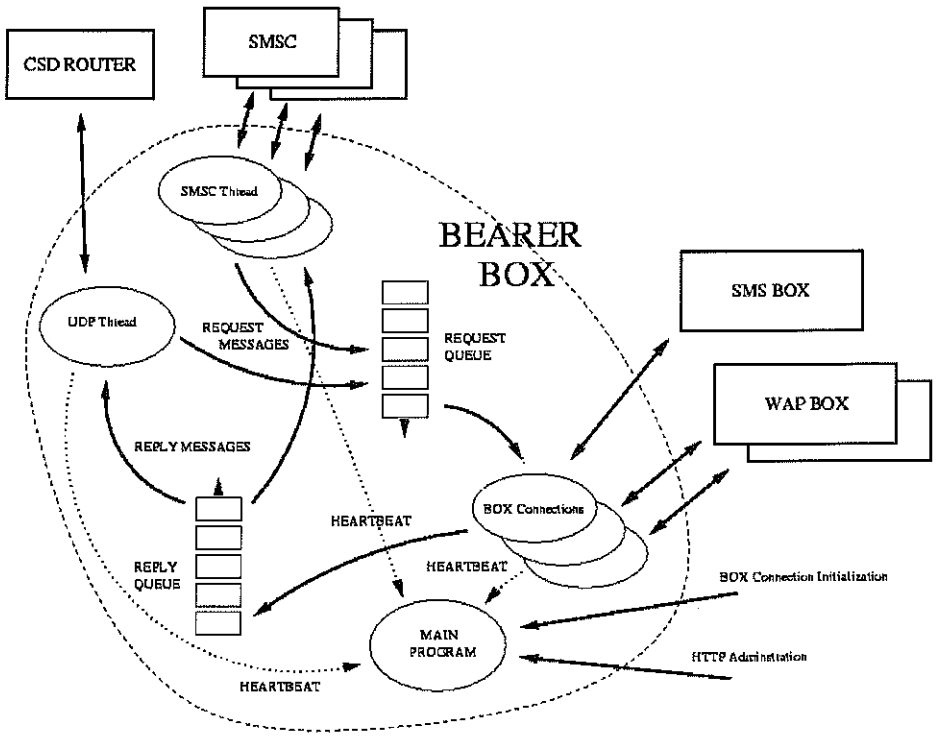
Il WAP box non si preoccupa degli indirizzi, li usa solo per identificare la state machine associata al WAP client.

Il Bearer Box

Il bearer box comunica con i centri SMS ed i router CSD, accetta pacchetti UDP e verifica se essi sono pacchetti WDP. Il bearer box inoltre invia i messaggi ai WAP box appropriati e gestisce il load balancing. Infine inserisce in una coda apposita i messaggi non inviati in caso di traffico e tiene traccia dei vari WAP box connessi.

Design

Il bearer box è implementato con diversi thread per semplifiare il modello di programmazione e per incrementare l'efficienza. In figura i componenti esterni al sistema sono descritti per mezzo di rettangoli, i thread del bearer box sono rappresentati da cerchi e le connessioni di tipo data stream sono indicate per mezzo delle frecce.



I threads SMSC gestiscono la connessione con i centri SMS (un thread per ogni centro) e comunicano con loro per mezzo di appositi protocolli. Un thread riceve i messaggi da un SMS Center e li deposita in una Request Queue; in maniera del tutto analoga preleva i messaggi da inoltrare al centro SMS (cui `e collegato) da una Reply Queue. I thread CDSR (UDP thread) sono simili ai thread SMSC con la differenza che essi ricevono WDP datagram da router CSD: si utilizzano tanti threads CDSR quanti sono i client WAP connessi al gateway.

I thread box connection gestiscono il trasferimento dei dati con i box WAP e SMS. Tali thread prelevano i messaggi da una Request Queue e li inviano ai box. Analogamente ricevono i messaggi dai box e li depositano in una Reply Queue dalla quale gli SMSC o CDSR thread li leggeranno.

Il Main program gestisce tutti i principali controlli del bearer box. In particolare è sempre in ascolto di eventuali nuove richieste di connessione da parte di WAP o SMS box e crea nuovi thread per loro.

Inoltre è in ascolto sulla porta HTTP-service per i comandi amministrativi e mantiene informazioni associate a tutti i box collegati (per fare questo rileva i messaggi heartbeat e compie azioni correttive quando si ferma il flusso di tali messaggi).

Box Connection

Ogni SMS/WAP box si connette con il bearer box per mezzo di un socket TCP/IP: un nuovo thread Box Connection è quindi generato per gestire questa nuova connessione. Non appena si apre una nuova connessione, il bearer box invia le informazioni di configuarazione ai box SMS e Wap.

HTTP Monitoring

Il bearer box si può monitorare e configurare per mezzo di una interfaccia HTTP: il main program gestisce le HTTP-requests.

Load Balancing

Il bearer box utlizza un metodo semplice per bilanciare il carico tra i diversi box WAP e SMS. Ogni Box Connection ha un livello di carico che si riferisce al box a cui `e collegato. Se il livello di carico è più grande del più piccolo livello di carico di un box connection dello stesso tipo (SMS o WAP), il box connection non preleva nessuna richiesta dalla coda fino a che il suo livello di carico non `e diminuito (o il livello di carico dei box connection dello stesso tipo non lo hanno raggiunto).

Request Queue

Questa coda gestisce tutti i messaggi in arrivo dai centri SMS e dai routers CSD. Tutti i messaggi in arrivo sono aggiunti alla fine della coda mentre i messaggi in uscita sono prelevati a partire dalla testa della coda.

Un thread non prende il primo messaggio della coda ma preleva il primo messaggio associato al tipo di thread (ad esempio un thread connection box, che collega il bearer box ad un WAP box, preleverà solo messaggi provenienti da router CSD.

Solo i box connection leggono i messaggi provenienti da una coda. L'attuale reader è determinato dal tipo di messaggio (SMS o UDP) e dal criterio di load balancing. Infine solo i threads CSDR e SMSC possono aggiungere messaggi in tale coda.

Reply Queue

Il comportamento della Reply queue è molto simile a quello della Request Queue. I nuovi messaggi sono appesi alla coda, I messaggi sono aggiunti nella Reply Queue dai box connection e sono letti dai thread CSDR e SMSC.

6.3 Il WAP Box

Viene di seguito brevemente descritto come sono state implementate le macchine a stati dei vari layer del WAP stack, come sono gestiti i thread e le comunicazioni con i WML content server, come il WML ed il WMLScript sono convertiti in una forma binaria compatta e come sono gestiti i timeout.

Ogni transazione ed ogni sessione è rappresentata da una struttura dati che implementa la macchina a stati per quel livello. Ogni nuovo pacchetto WDP è gestito da un nuovo thread del WAP box che cerca la corretta WTP state machine o ne crea una nuova (in pratica un pacchetto WDP è un evento per la macchina a stati WTP). Se questa macchina a stati necessita di trattare con una sessione chiama il WSP engine code per cercare o creare una nuova macchina a stati del livello WSP e diventa a sua volta un evento per la WSP states table. Tutto questo è gestito sempre dallo stesso thread che era stato originato per gestire l'evento generato dalla ricezione del pacchetto WDP.

Similmente se la WSP states machine necessita di fornire un evento alla WTP state machine, la gestione è affidata sempre allo stesso thread che cesserà di esistere solo quando tutti gli eventi generati da quel pacchetto WDP sono stati elaborati.

È possibile che arrivi una raffica di pacchetti WDP e che non sia possibile elaborarli tutti. In tal caso un blocco delle strutture dati, gestito per mezzo di un semaforo,

assicura che si elabori un solo pacchetto alla volta mentre i restanti sono messi in una coda.

Infatti se la creazione di un nuovo thread per la gestione di un nuovo pacchetto WDP non pu`o essere processato, perché la WTP states table è ancora bloccata dal precedente pacchetto allora l'evento è inserito in una apposita coda della WTP states machine ed il thread è terminato.

Quando poi il thread che aveva bloccato la WTP states table ha terminato di processare il pacchetto che lo aveva generato, va a prelevare dalla coda degli eventi pendenti il prossimo pacchetto da gestire.

In tal modo non è più necessario creare un nuovo thread apposito: questa caratteristica è molto importante in caso di traffico dei pacchetti sostenuto perchè così non si corre il rischio di riempire la thread table del sistema operativo.

La struttura della macchina a stati del livello WTP contiene le informazioni necessarie a gestire la state table del WTP layer e in particolare:

- stato corrente;
- identificatore di transazione (TID);
- source address, port;
- destination address, port;
- classe della transazione;
- informazioni sugli acknowledge inviati e attesi;
- timer per la gestione dei timeout;
- coda degli eventi;
- evento WSP necessario per la comunicazione con il livello WSP.

La struttura necessaria, invece, per la gestione della WSP state machine, oltre alle informazioni associate allo stato, agli eventi e ai timeout deve conservare i dati da utilizzare, durante la sessione, per la connessione ai WML content server (web server).

La conversione dei codici sorgenti WML e WMLScript in formato binario compatto avviene grazie all'utlizzo della libreria Gnome-xml (almeno la versione libxml 1.8.6) che consente un facile parsing di documenti di tipo XML (si ricorda che il WML `e definito come DTD di XML).

7 Esempio di applicazione WAP per Casa Domotica.

In questo paragrafo viene descritto il livello del sistema **HouseSystem** relativamente alla comunicazione sia fra il *concentratore* e la *centralina* che fra la centralina e gli *attuatori* all'interno della *rete domotica*. La centralina su cui è stato svolto il lavoro non è quella definitiva che farà parte del sistema finale, ma solo un dispositivo con prestazioni ridotte (CM11) anche se sufficiente per gli scopi demo di questo lavoro (vedi anche (7)).

Gli attuatori sono semplici moduli X10 chiamati LM12 (cioè lamp module), vedi Figura 7. X10 è il protocollo che governa questi moduli e implementa la capacità di comunicazione verso la rete locale domestica a cui questi dispositivi sono connessi.

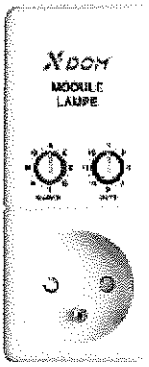


Figura 7: Lamp Module

Gli attuatori sono deputati essenzialmente a svolgere puri compiti operativi e non implementano particolari funzionalità di servizio; l'intelligenza che governa (configurazioni, attività temporizzate, eventi etc.) i dispositivi stessi, sono spostate sulla centralina. Invece nella applicazione l'intelligenza che governa gli attuatori si trova direttamente sul concentratore che si interfaccia con la rete domotica mediante un altro dispositivo X10, chiamato CM11, vedi Figura 8.

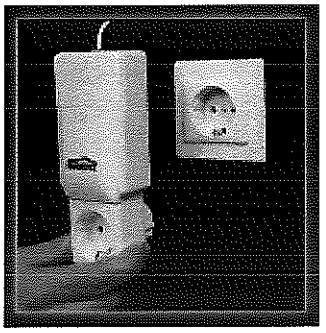
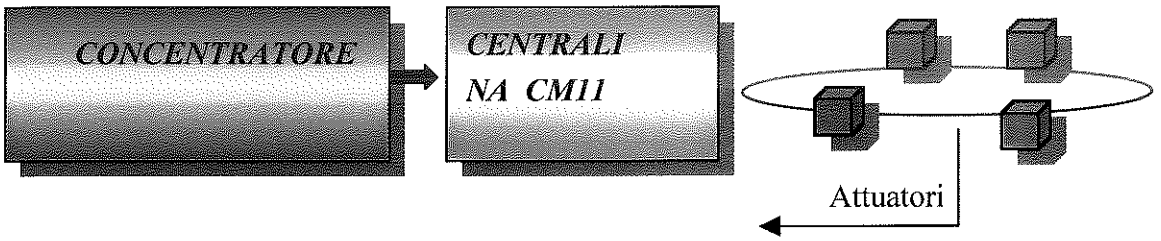


Figura 6: CM11

Quindi a seconda di quello che prendiamo in considerazione, cambia chi governa gli attuatori, vedi Figura



La centralina è il sistema completo, capace di veicolare informazione verso tutti gli attuatori della casa (allarmi, sensori, termovalvole ecc..) e si interfaccia con vari tipi di rete domestiche (PL, RF, BUS), e sarà uno degli obiettivi del gruppo lavoro renderlo integrato con il concentratore. La soluzione ideale sarebbe rendere il tutto un unico blocco. Invece il CM11 è solo una interfaccia X10 tra il concentratore e la rete domestica PL e si preoccupa di spedire o ricevere comandi lungo tale rete.

In realtà gli attuatori possono essere non solo dispositivi "stupidi" ma anche moduli software destinati ad essere integrati all'interno degli elettrodomestici e tali da realizzare la

logica di elaborazione e di controllo degli stessi. In generale ogni attuatore, presente e futuro, deve essere in grado:

- configurare automaticamente il sistema nel momento della connessione alla rete domestica;
- garantire l'impostazione, la lettura e la supervisione delle variabili operative di funzionamento (valori impostati, valori rilevati dai sensori, valori di stato) del dispositivo, provvedendo a verificare attraverso queste il corretto funzionamento dello stesso;
- rilevare malfunzionamenti o comportamenti anormali al loro interno;
- realizzare semplici funzioni di test;
- gestire la notifica verso il concentratore delle risposte alle sue interrogazioni (ad esempio circa le variabili di stato) o di eventi (allarmi, cambiamenti di stato, malfunzionamenti, etc.) generati all'interno del dispositivo stesso.

Quindi la centralina è un dispositivo essenziale, in quanto supervisiona il funzionamento di tutti gli elettrodomestici presenti nella abitazione, attraverso lo scambio di informazione e di controllo realizzato da una parte con gli attuatori connessi nella rete locale domotica, dall'altra con il concentratore che si occuperà di interagire con il back-end del sistema ServiceHouse, attraverso la connessione con la rete esterna Internet.

Dal punto di vista funzionale il concentratore risponde ad una doppia esigenza: la prima è quella di fornire un "ponte" di comunicazione fra il Service Provider e la centralina, fornendo le funzionalità tipiche di un gateway e provvedendo a filtrare e tradurre opportunamente le informazioni scambiate fra la casa e il mondo esterno, e la seconda, valida solo nel caso del CM11, quella di implementare la logica di gestione dei dispositivi

In dettaglio il concentratore è in grado di:

- ricevere e interpretare comandi o richieste riguardanti gli attuatori provenienti dal backend della centralina;
- realizzare la logica di gestione degli attuatori controllati (CM11);
- trasmettere informazioni riguardanti gli attuatori o correlate al loro funzionamento verso il backend del sistema;
- trasmettere comandi o richieste verso gli attuatori (CM11);
- memorizzare i parametri di funzionamento degli attuatori controllati (CM11);
- realizzare le funzionalità di programmazione delle attività per quei dispositivi che prevedono anche un controllo temporizzato (CM11);

Dal punto di vista architetturale, il concentratore ha due interfacce distinte sui differenti canali (bidirezionali) con cui è in comunicazione: quella verso la Rete, sulla quale realizza le funzionalità di Web o Wap Server e utilizza le tecnologie standard di Internet, e quella verso o gli attuatori, sulla quale si presenta come un dispositivo e utilizza il protocollo della rete locale domestica, o centralina. Queste vengono gestite in maniera diversa, essendo diverse le esigenze di comunicazione e le informazioni trasmesse sui due differenti canali.

Interfaccia verso la Rete

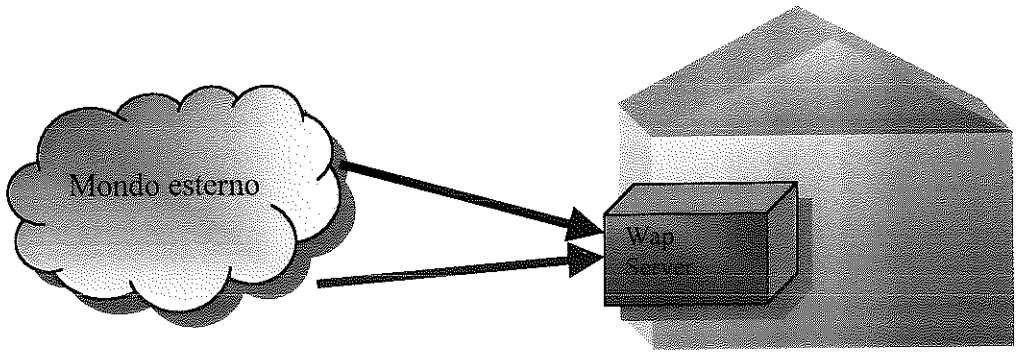
Come detto sopra, un interfaccia è quella rivolta al mondo esterno, cioè il concentratore prevede la realizzazione di una particolare connessione fra la casa e il fornitore del servizio,

che permetta la trasmissione di tutte le informazioni necessarie al controllo dei dispositivi domestici e in generale per la realizzazione di tutti i servizi previsti dalla piattaforma.

La comunicazione, descritta nel precedente capitolo, avviene secondo le seguenti modalità: non appena una delle due parti ha l'esigenza di trasmettere informazione (come ad esempio il comando di un cambiamento di stato per uno dei dispositivi controllati) verso l'altra, invia il messaggio; questo viene elaborato dinamicamente dalla parte ricevente attraverso la realizzazione di tutte le attività previste per il corretto completamento dell'operazione (come scrittura su database, accesso ai dispositivi ecc..): infine la parte ricevente provvede a notificare indietro il risultato dell'operazione.

Nel caso in cui le elaborazioni richieste non vadano a buon fine, viene riportato alla parte trasmittente un codice di errore che identifica il problema riscontrato. Solo in caso di notifica positiva la parte trasmittente assume realizzata correttamente l'operazione richiesta. In questo modo viene garantita la transazionalità degli scambi informativi realizzati tra back-end e concentratore.

Ci siamo limitati a configurare un Wap server, ottenuto dalla unione di un Wap Gateway e un server, con una connessione di tipo non permanente. La trasmissione è monodirezionale cioè dal mondo esterno al concentratore.



Il server adoperato è quello fornito dalla Microsoft IIS 5.0, in cui risiedono i Web Form mobile che sono inviati ai dispositivi mobili in risposta alla connessione dell'utente alla sua abiatazione. Il Gateway, installato sullo stesso computer del server, è quello della Kannel descritto nel capitolo 3. Il file in cui risiede la configurazione si chiama **kannel.conf**. Ci siamo solo interessati del wapbox, tralasciando smsbox, non necessario per l'applicazione. Alcune istruzioni di configurazione sono riportati qui sotto:

```
group = core
admin-port = 13000
wapbox-port = 13002
wdp-interface-name = "*"
box-deny-ip = "*.*.*.*"
box-allow-ip = "127.0.0.1"
```

Concludiamo qui la descrizione di questa interfaccia e analizziamo in maggior dettaglio la seconda.

Interfaccia verso gli attuatori

In accordo con il titolo, tratteremo il CM11, tralasciando la centralina. Nella casa il concentratore e i dispositivi sono interconnessi all'interno della rete locale domestica e su questa si scambiano le informazioni e i comandi per la gestione dei dispositivi

Relativamente al mezzo di comunicazione utilizzato nella abitazione è opportuno fare una distinzione fra le capacità di trasporto offerte da una rete dati, disegnata per supportare la trasmissione di contenuti informativi, e quelle offerte da una rete di controllo, specifica per la trasmissione di segnali di controllo.

Una rete dati utilizza pacchetti informativi di grandi dimensioni con una relativamente bassa frequenza di emissione. Una rete di controllo è invece caratterizzata dal trasporto di pacchetti di dimensioni contenute ma con frequenza di emissione maggiore e con requisiti di real-time. La dimensione dei pacchetti X10 arriva a pochi byte.

La rete elettrica si presta bene per il trasporto di dati del secondo tipo; esso deve trasportare semplici comandi di attivazione o di notifica, e l'uso di reti ad alta velocità risultano inopportuni e costosi. Un'altra possibile soluzione è data anche dalla trasmissione wireless che sarebbe una soluzione ottimale per il supporto ai comandi di controllo.

Dal punto di vista funzionale, la comunicazione può essere iniziata sia dal concentratore che dall'attuatore. Il concentratore infatti può invocare sul dispositivo i comandi o le richieste (di impostazione dello stato o dei parametri di funzionamento) ricevuti direttamente dal back-end del sistema ISP, ovvero temporizzati in base ai programmi di attivazione impostati dall'utente e residenti sul concentratore stesso. L'attuatore, per parte sua, notifica al concentratore ogni cambiamento di stato determinato da un intervento diretto manuale da parte dell'utente su di esso, così come anche eventi generati all'interno del dispositivo stesso (come ad esempio malfunzionamenti o allarmi) o risposte ad interrogazioni fatte dal concentratore (ad esempio della temperatura rilevata dai sensori del climatizzatore).

Il sistema deve poi prevedere che la variazione di uno qualsiasi dei parametri di funzionamento degli attuatori venga notificato al concentratore e da questo direttamente al back-end dell'ISP. In questo modo viene garantito sempre l'allineamento dei valori attuali di funzionamento dei dispositivi all'interno delle varie sezioni di sistema e sull'interfaccia utente viene presentata la situazione dei dispositivi presenti nella casa aggiornata in tempo reale. La notifica in tempo reale al back-end può avvenire solo se la connessione tra il concentratore e l'ISP è di tipo permanente.

Meccanismi di *polling* diretto dei dispositivi, cioè interrogazione sullo stato di funzionamento del dispositivo, possono essere indicati dall'utente in base a particolari necessità, come ad esempio il dispositivo antifurto (ritenendo di dover fornire un grado assoluto di certezza per questa informazione in relazione alla criticità del dispositivo controllato) e di rilevazione a intervalli regolari della temperatura e dell'umidità rilevate dai sensori associati al dispositivo climatizzatore.

Quindi le caratteristiche della comunicazione fra i dispositivi e il concentratore suggeriscono proprio l'utilizzo di un protocollo per la rete domestica semplice ed efficiente, e che si adatti alle esigenze dell'applicazione di home automation. Informazioni come audio, video devono per ora usufruire di una rete diversa.

Si ritiene opportuno che per la realizzazione di un sistema domotico adottare un protocollo che permetta ai dispositivi connessi di *auto-configurarsi* all'atto dell'installazione senza costringere l'utente a effettuare interventi manuali, caratteristica non è presente nel protocollo X10. Non è presente un meccanismo di identificazione dei servizi attraverso il quale i dispositivi acquisiscono informazioni circa la rete a cui sono connessi per individuare quali servizi sono resi disponibili dagli altri elementi presenti.

La funzionalità di un protocollo di comunicazione con la caratteristica di autonfigurarsi, è resa necessaria considerando la possibilità che gli elettrodomestici all'interno della casa possano essere aggiunti o sostituiti con una certa frequenza. All'utente non è richiesto in

questo modo altro che la connessione o la disconnessione del dispositivo dalla rete domestica, garantendo in ogni momento il funzionamento del sistema che, in maniera "intelligente", acquisisce le informazioni necessarie al suo funzionamento in rete.

Diversi sono i protocolli che implementano un simile meccanismo di scoperta dei servizi. Fra questi possono essere citati SLIP (Salutation, Service Location Protocol) che propone un approccio al problema applicando gli esistenti standard Internet e che è disegnato per essere un protocollo leggero e con requisiti di amministrazione minimi, Universal Plug & Play (UpnP, basato sul protocollo IP e che è disegnato per lo scambio di grandi quantità di dati fra PC e altri dispositivi come telefoni, televisioni, stampanti, console per giochi), Home Audio/Video Interoperability (HAVi), Bluetooth Service Discovery e JINI.

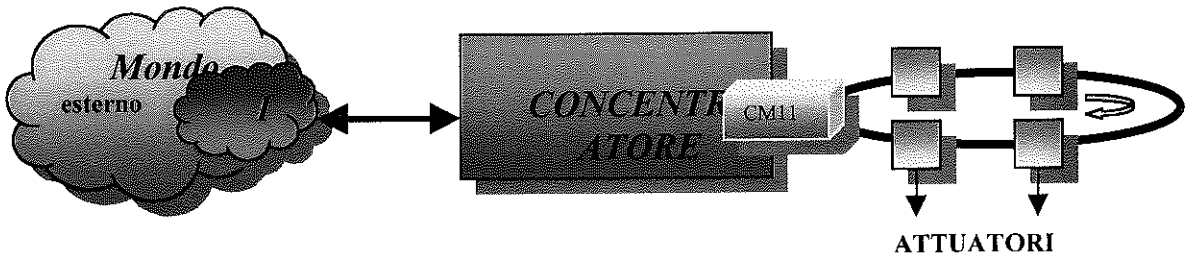
Il concentratore

Il concentratore è l'elemento che contiene la logica di gestione dei dispositivi presenti nella casa. Esso trasferisce le richieste provenienti dal mondo esterno verso gli attuatori e viceversa. All'interno del concentratore è presente l'applicazione di gestione, che realizza la logica di supervisione della sezione domestica del sistema, provvedendo a interpretare le informazioni ricevute su entrambe le interfacce, elaborare e trasmettere i comandi.

Si fa distinzione tra il *command-based* e lo *status-based* a seconda della direzione della comunicazione. Il command-based rappresenta la notifica proveniente dal mondo e catturata dal concentratore che la elabora in maniera opportuna e la inoltra atrraverso la rete domestica. Infatti la richiesta del dispositivo mobile (ad esempio accendere la lampada posizionata nella casa dell'abitazione che ha l'indirizzo fisso A1) perviene al Web server che la elabora e la comunica all'interfaccia con la rete elettrica, cioè il CM11, che la inoltra alla lampada A1 che è attaccata al modulo X10.

Invece lo status-based rappresenta l'interrogazione da parte del concentratore dei dispositivi connessi alla rete.Il concentratore registra tutte le notifiche emesse dai diversi attuatori; lo scatenarsi di un evento interno al dispositivo (ad esempio un evento di allarme a seguito della rilevazione da parte di un sensore dell'antifurto) genererà la segnalazione sulla rete di una notifica contenente i parametri necessari alla sua identificazione (tipo di dispositivo emittente, tipologia della notifica) e una serie di attributi opzionali (ad esempio l'indicazione del sensore che ha rilevato l'intrusione nel caso in cui si tratti di una notifica di allarme emessa dal dispositivo antifurto).

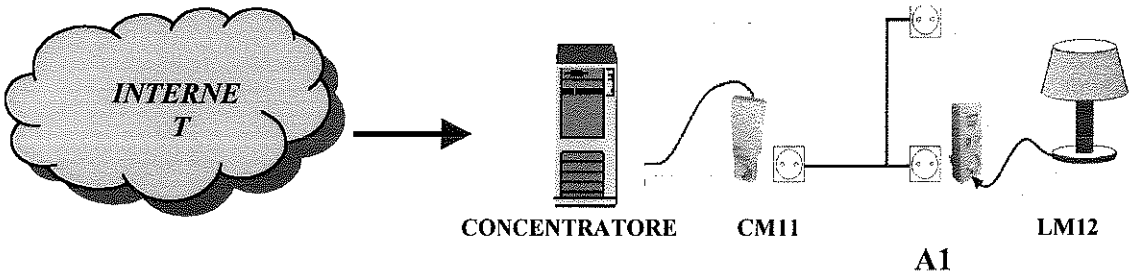
Questa notifica sarà quindi catturata dal concentratore che la elaborerà in maniera opportuna e provvederà a darne comunicazione al backend per la segnalazione di questa all'utente, vedi figura.



Il programma di gestione della rete domestica è stato effettuata mediante un programma scritto in Visual Basic.NET. Se consideriamo che la gestione dell'interfaccia verso il mondo esterno è programmata in Visual C#, abbiamo dato esempio di quella interoperabilità tra linguaggi descritta nel capitolo precedente.

Command-based

L'architettura della struttura del tipo command-based è mostrata in figura



Come dispone il noto protocollo X10, tutti i dispositivi all'interno della abitazione sono individuati da un indirizzo fisico composto da una lettera (A..P) e da un numero (1..16). Il comando da impartire al modulo si chiama funzione (on, off....). Allorché l'elettrodomestico viene collegato al LM12, esso acquisisce l'indirizzo fisico del modulo e tale informazione viene indicata al concentratore manualmente. Modulo e dispositivo diventano un'unica entità. Se l'elettrodomestico viene spostato dalla sua posizione, esso porta con sé il modulo. In questo modo ogni oggetto connesso alla rete ha un indirizzo statico; se dovessi scambiare i moduli, il concentratore deve essere subito informato.

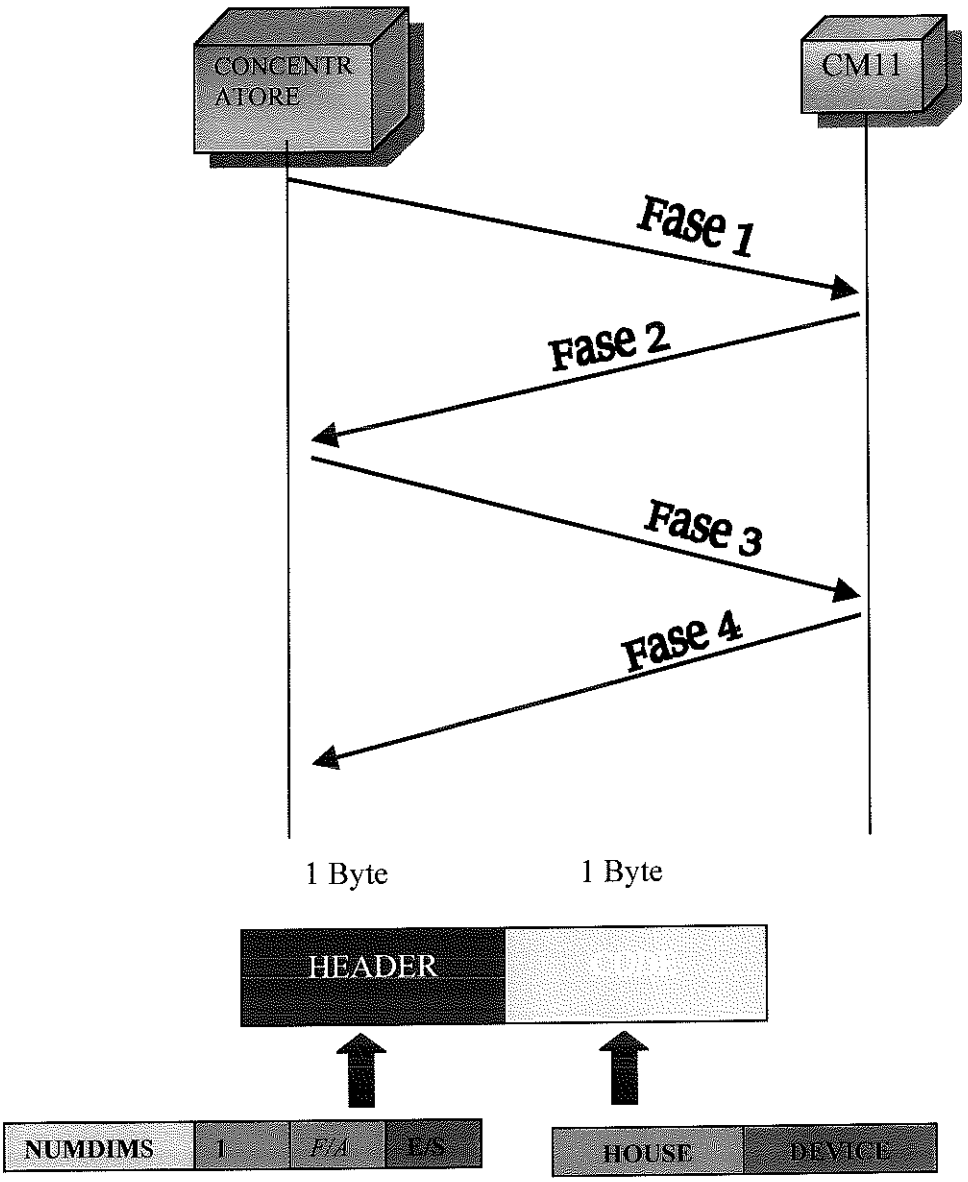
In pratica abbiamo creato una *tabella di instradamento,* collocata all'interno del concentratore, che sarà aggiornata tutte le volte che avviene una modifica. La tabella è utile sia per l'utente di casa, ma in particolar modo all'utente che da remoto aziona un dispositivo ubicato dentro l'abitazione. Infatti non mi devo preoccupare quale sia il suo indirizzo. L'utente accende la lampada di camera e non la lampada con indirizzo x. In questo modo scompare il problema degli indirizzi fisici dei dispositivi. Questa tecnica è in fase di sviluppo e non ancora implementata.

Finito di elaborare la richiesta proveniente dal mondo esterno, il concentratore apre immediatamente una connessione con la sua interfaccia, collegata alla propria porta seriale.

La trasmissione tra il concentratore e CM11 può essere schematizzata in quattro fasi come mostrato in figura:

FASE 1:

Il concentratore invia al CM11 un primo pacchetto di 2 byte formato dall'Header + Indirizzo del dispositivo da comandare



NUMDIMS: rappresenta il valore di luminosità da trasmettere e il suo numero varia tra 0-22. Occupa all'interno del pacchetto i primi 5 bits.
Bit 1: serve a mantenere il sincronismo dell'interfaccia. Occupa il bit posizione 2.
F/A: indica se il pacchetto è una funzione o un indirizzo. Se è una funzione il bit vale 1 altrimenti 0. Occupa il bit di posizione 1.
E/S: indica se il seguente pacchetto è una trasmissione di tipo extended (bit = 1)o standard (bit = 0). Occupa il bit di posizione 0.
Quindi a seconda del bit F/A, il pacchetto è un indirizzo oppure un comando.

FASE 2:

Una volta che l'interfaccia riceve la trasmissione dal concentratore, effettua una somma dei bytes, e spedisce un byte di controllo, cioè il **checksum.** Il controllo è dato da

checksum = (header + code) & (0xff)

FASE 3:

Se il controllo fatto dal CM11 risulta corretto, allora il concentratore spedisce un byte pari a **0x00** di conferma. Se il controllo è errato, allora si ritorna alla fase 1.

FASE 4

In questo momento l'interfaccia è pronta a ricevere i comandi dal concentratore.

Abbiamo mostrata una semplice procedura in cui il concentratore apre una connessione con la sua interfaccia e invia lungo la rete elettrica i comandi per gestire gli attuatori. La medesima operazione può essere eseguita sulla centralina permettendo di interagire con un numero maggiore di reti domestiche e controllare una vasta gamma di attuatori.

8 Conclusioni e sviluppi futuri

In questa nota abbiamo descritto una metodologia che permette di sviluppare una reale applicazione di Home Automation e di individuare una piattaforma tecnologica che abilita il controllo dei dispositivi posti all'interno di un ambiente tramite protocollo WAP.

Il sistema da noi realizzato utilizza tecnologie aggiornate (.NET, C#, Microsoft Mobile Internet Toolkit), affrontando le problematiche delle interfacce da presentare ai dispositivi mobili.

La tecnologia .NET ha richiesto una fase iniziale di studio di base e apprendimento pratico non irrilevante ma che ha permesso di ottenere in modo più elegante, ottimi risultati. Il più importante è quello della possibilità di avere il rendering automatico di ogni controllo a seconda del dispositivo mobile in uso, permettendo di ottimizzare la visualizzazione delle pagine.

Sono stati anche descritti i protocolli che meglio si adattano alla realizzazione di una rete domotica, confrontando le loro caratteristiche e utilizzando il protocollo X10, le cui caratteristiche principali sono i bassi costi e la semplicità d'uso.

In pratica si può ritenere che la connessione dei dispositivi attraverso una rete domestica e soprattutto la possibilità di rendere disponibile un loro accesso alla Rete, aprono la possibilità a tutta una nuova categoria di servizi per la casa. Verosimilmente il loro sviluppo non sarà contemporaneo, ma una crescente offerta di applicazioni sarà disponibile non appena verrà riconosciuto il potenziale commerciale di una architettura capace di portare dei miglioramenti in molte situazioni.

In altre parole è probabile che non tutte le applicazioni avranno una penetrazione immediata nel mercato, così come è prevedibile una evoluzione attraverso fasi differenti che porteranno segmenti diversi del mercato ad adottare progressivamente le tecnologie disponibili.

Un aspetto importante, per favorirne appunto lo sviluppo, soprattutto per l'aiuto ai disabili e alle persone anziane, è quello di costruire nuove interfacce utente (UI) per i dispositivi domestici. Ad esempio, per persone che hanno difficoltà con l'uso della tastiera si potrebbero integrare UI che riconoscano il parlato, con i necessari attributi di affidabilità.

Un innovativo approccio potrebbe essere quella di progettare una struttura completamente adattabile alle specifiche esigenze dell'utente, contrariamente a quanto

succede spesso negli attuali sistemi commerciali, dove sono le esigenze dell'utente che si devono adattare alle funzionalità messe a disposizione dal sistema.

Bibliografia

1 Bianchi Bandinelli, R. Fusco, G. Rossi, R Saba, A.: "IB (Intelligent Building)" Proc. of the I° TIDE Congress, Of Rehabilitation Technology, Strategies for the EU, Brussels, (1993).

2 Dutta-Roy, A.: Networks for Homes, IEEE Spectrum (1999) 27-33

3 Krishnan, C.N. Ramakrishna, P.V. Prasad, T.V. Karthikeyan, S. : Power-Line As Access Medium-A Survey", Int. COMMSPHERE 2000, Indian Institute of Technology Madras, Chennay, India. (2000).

4 Manuali e depliant Sistema Casa 2000x ed. Sistema Casa Srl , Milano

5 Vari manuali e libri di testo Microsoft sui temi trattati.

6 Rocchi A., Bianchi Bandinelli R, Bertini G. "Progetto dei sistemi di Home Automation. Introduzione e un caso di studio: Sistema Casa". Nota Tecnica ISTI-CNR B4-21 , dic. 2002.

7 L. Tarrini, R. Bandinelli, V. Miori, G. Bertini. : "Remote Control of Home Automation Systems with Mobile Devices", Proc. 4° Int. Symp. Mobile HCI2002, Settembre 2002, Pisa , Springer Verlag.

Indice