Università degli Studi di Roma "La Sapienza"
Facoltà di Ingegneria – Corso di Laurea in Ingegneria Gestionale
Corso di Progettazione del Software
Proff. Toni Mancini e Monica Scannapieco

Progetto **PC.20060626**

versione del 12 marzo 2007

La *domotica* (detta anche *home automation*) è una disciplina che si occupa dell'integrazione delle tecnologie che consentono di automatizzare molte azioni che di solito si svolgono in casa. Una cosiddetta "casa intelligente" di solito prevede un computer (una sorta di *maggiordomo informatico*), e un insieme di sistemi software che assistono gli ospiti in una serie di azioni frequenti, ad esempio, automatizzando l'accensione e lo spegnimento delle luci all'ingresso/uscita delle varie persone in/da una stanza, o dell'impianto di climatizzazione a seconda della stagione e della presenza/assenza degli ospiti).

Tuttavia, un maggiordomo che si possa davvero definire "intelligente" dovrebbe essere in grado di gestire alcuni compiti in modo completamente autonomo, e cercare di assistere gli ospiti in modo pro-attivo, prevedendo le loro necessità e dispensando opportuni suggerimenti.

L'oggetto di questo progetto è una applicazione di domotica avanzata che permetta al maggiordomo, tenendo sotto controllo la dispensa degli alimenti, di fare automaticamente la spesa (via Internet, con consegna a domicilio)[1] evitando che gli ospiti della casa si ritrovino con il "frigo vuoto", e suggerendo i piatti e i menu che possono essere preparati con gli alimenti di cui si dispone, imparando le combinazioni di pietanze sgradite.

L'applicazione complessiva è composta da tre sistemi interagenti: il *FridgeMonitor*, lo *Shopper* e lo *ChefAdvisor*. Il FridgeMonitor si occupa di monitorare il contenuto del frigo, ed estrarre (mediante opportuni sensori) le informazioni circa la quantità disponibile dei singoli alimenti; lo Shopper ha il compito di fare la spesa via Internet quando la quantità di alcuni alimenti scarseggia; infine lo ChefAdvisor si occupa di suggerire all'utente affamato (e quindi poco propenso alla creatività) i piatti che possono essere preparati in breve tempo con gli alimenti di cui dispone.

In questa sede ci occuperemo *esclusivamente* del sistema ChefAdvisor.

Si richiede di effettuare le fasi di Analisi, Progetto, e Realizzazione del sistema in Java, utilizzando la metodologia illustrata nel corso.

[1] Ad es. su `www.prontospesa.it`

Requisiti

ChefAdvisor deve conoscere i vari piatti che l'utente sa cucinare, insieme con i relativi ingredienti (e rispettive quantità) necessari alla loro preparazione. I piatti vengono classificati in antipasti, primi, secondi, contorni, e dessert. Tali categorie sono tutte disgiunte, ad eccezione degli antipasti e dei secondi che possono avere elementi in comune (ovvero, vanno previsti piatti che possono considerarsi sia antipasti che secondi). Di ogni piatto interessa inoltre conoscere il nome, il tempo di preparazione (in minuti) e le relative kilo-calorie (kcal) a porzione. Queste sono pari alla somma delle kcal fornite dai singoli ingredienti.

Degli alimenti interessa conoscere il nome e la loro disponibilità, ovvero la quantità presente nella dispensa (i valori di tale attributo verranno aggiornati dal sistema FridgeMonitor). Le quantità dei singoli alimenti possono essere espresse in grammi, in litri, oppure in unità, a seconda dei casi. Infine degli alimenti interessa la tipologia (ad es., "pasta&riso", "carne", "pesce&fruttiDiMare", "verdura", etc.).

ChefAdvisor deve anche essere in grado di mantenere informazioni circa i gusti dell'utente.[2] In particolare, deve essere in grado di decidere, dati due piatti, se questi sono tra loro incompatibili (ovvero, non sono graditi dall'utente durante lo stesso pasto –ad es. "linguine alle vongole" e "bistecca"). Il meccanismo che si desidera utilizzare per tali decisioni si basa su dichiarazioni fatte dall'utente circa incompatibilità tra singoli ingredienti o tipologie di ingredienti, ed è il seguente: per ogni piatto, alcuni dei suoi ingredienti (da uno a tre) vengono considerati "caratterizzanti" (ad es., il piatto "linguine alle vongole" è caratterizzato dagli ingredienti "linguine" –di tipologia "pasta&riso"– e "vongole" –di tipologia "pesce&fruttiDiMare", anche se la sua preparazione include altri ingredienti, ad es., "olio"). Il sistema deve essere in grado di distinguere gli ingredienti caratterizzanti dei piatti, perché due piatti si definiscono incompatibili se alcuni dei loro ingredienti caratterizzanti sono tra loro incompatibili.
Infine, due alimenti (possibili ingredienti di piatti) sono considerati incompatibili secondo un meccanismo a due livelli:

- Sono "implicitamente incompatibili", perché di tipologie incompatibili (ad es. l'utente potrebbe istruire il sistema dichiarando che "carne" e "pesce&fruttiDiMare" sono tipologie incompatibili: da questo deriva che ogni coppia di alimenti di cui il primo di tipologia "carne" e il secondo di tipologia "pesce&fruttiDiMare" –come "linguine alle vongole" e "bistecca"– saranno considerati incompatibili);
- Sebbene siano di tipologie compatibili tra loro, sono stati esplicitamente dichiarati dall'utente incompatibili (ad es., l'utente dichiara di non gradire "riso" e "patate" nello stesso menu, nonostante le rispettive tipologie ("pasta&riso" e "verdura") non siano in generale incompatibili).

[2] Si assuma per semplicità che l'utente sia unico.

Come detto, l'obiettivo del sistema è quello di suggerire all'utente un menu per un pasto **gradevole**, **realizzabile** con gli alimenti disponibili al momento, e sufficientemente **veloce** da preparare. Un menu può essere considerato come una lista di più piatti: un antipasto, un primo, un secondo, un contorno, un dessert. Tuttavia, per ovvie ragioni dietetiche, le componenti di un menu sono tutte opzionali (ovvero può mancare l'antipasto, il primo, . . . , il dessert). Dei menu è importante conoscere le kcal totali.

In particolare, dato un insieme di alimenti con relative quantità disponibili (la "dispensa"), un intero positivo `npersone`, due valori `minKCal` e `maxKCal`, un valore `tempoMax` e 5 valori booleani (`antip`, `primo`, `secondo`, `contorno`, `dessert`), si vuole suggerire l'insieme dei possibili menu (composti da un antipasto –se `antip=true`–, un primo –se `primo=true`–, . . . , un dessert –se `dessert=true`) che è possibile realizzare (per `npersone`) con la dispensa a disposizione (**realizzabilità**), le cui kcal totali siano nell'intervallo indicato (**gradevolezza**), e tale da potersi preparare nel tempo `tempoMax` (**velocità**), assumendo che i singoli piatti siano preparati uno alla volta.

Inoltre, è richiesto che il FridgeMonitor, dato un alimento, sia in grado di aggiornare il valore della sua disponibilità, e che lo ChefAdvisor offra un opportuno servizio all'utente, che gli permetta di dichiarare come incompatibili coppie di alimenti, o coppie di tipologie di alimenti.

Suggerimento. Si osservi che i menu sono *identificati* dall'insieme dei loro piatti. In altre parole, due menu sono considerati **uguali** se e solo se coincidono nell'insieme dei relativi piatti. Questa uguaglianza "profonda" suggerisce che i singoli menu non possono essere modellati come oggetti di una classe, ma devono essere modellati come istanze di un tipo di dato. Inoltre, per il sistema non è di interesse mantenere i menu che suggerisce: questi vengono "calcolati" di volta in volta a partire dall'insieme dei piatti realizzabili.