

Capitolo 3

Adattamento dell'Interazione all' Utente e al Contesto

- 3.1. Introduzione
- 3.2. Metodi per la modellizzazione di utenti
 - 3.2.1. Acquisizione delle informazioni
 - 3.2.2. Modellizzazione per stereotipi
 - 3.2.3. Aggiornamento dei profili
- 3.3. Trattamento dell'incertezza nei Modelli
 - 3.3.1. Cos'è una Rete Causale Probabilistica (BN)
 - 3.3.2. I BN nei modelli di Utente

3.1. Introduzione

Nel Capitolo 2, abbiamo visto come la flessibilità sia una delle caratteristiche che influenzano l'usabilità delle interfacce e come la sua importanza sia proporzionale alla variabilità nelle caratteristiche dell'utente. Se l'applicazione è progettata per una sola categoria di utenti, tutti con caratteristiche presumibilmente simili e stabili, non sarà necessario personalizzare l'interazione. Altrimenti, occorrerà costruire un'architettura simile a quella indicata nella figura 3.1.

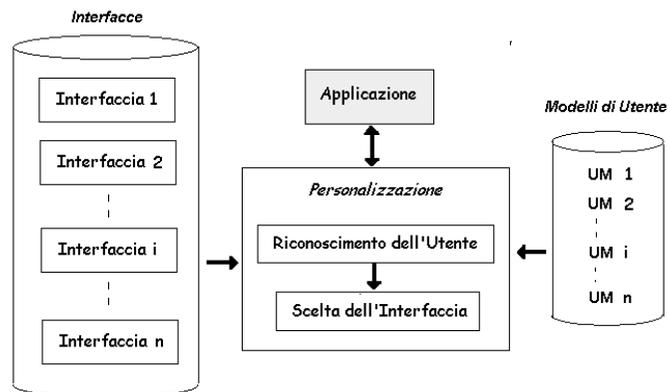


Figura 3.1: Architettura dei Sistemi ad Interazione Personalizzata

In questa architettura, la separazione fra applicazione ed interfaccia viene utilizzata costruendo diverse interfacce, una per ciascuna delle classi di utenti potenziali previste. Un database dei *modelli di utente* conterrà una descrizione di queste classi, che potrà essere più o meno dettagliata, a seconda delle esigenze dell'applicazione. Quando un utente accede all'applicazione, un modulo di *personalizzazione* si occuperà di riconoscerlo, attribuirlo ad una delle classi conosciute e quindi scegliere l'interfaccia più opportuna.

Tuttavia, le caratteristiche di uno specifico utente non sono, in genere, identiche a quelle della classe a cui appartiene: esistono, ad esempio, molti utenti inesperti nell'uso di calcolatori che hanno invece esperienza nell'uso di altre tecnologie, come i telefoni cellulari: questo rende le loro esigenze d'interazione con un sistema informativo diverse da quelle di utenti che non utilizzano nessun tipo di tecnologia.

Esempio sul Caso di Studio 3.1

Una prima classificazione degli utenti di SIMB può essere fatta sulla base dell'età: i medici giovani saranno presumibilmente più esperti nell'uso di tecnologie rispetto ai loro colleghi più anziani. Tuttavia, è possibile che esistano medici che hanno utilizzato, per ragioni professionali o non, un motore di ricerca. Questi utenti non apparterranno né alla classe degli 'esperti' (perché le loro conoscenze si limitano ad alcune applicazioni limitate) né a quella degli inesperti; nessuno dei due modelli darà dunque una descrizione adeguata delle loro caratteristiche.

Le caratteristiche di uno specifico utente possono, inoltre, variare nel tempo: tipicamente, un utente frequente potrà essere inizialmente attribuito alla categoria degli 'inesperti' ma diverrà 'esperto' dopo aver utilizzato l'applicazione per un tempo più o meno lungo. Conviene, per queste ragioni, distinguere fra *modelli di utenti* (che rappresentano, come abbiamo detto, le caratteristiche della classe) e *profili di utenti* (che rappresentano invece un singolo individuo). In questo caso, il modulo di *riconoscimento dell'utente* verrà raffinato come descritto in figura 3.2, l'interfaccia sarà definita sulla base del profilo anziché del modello e il profilo di ogni utente verrà aggiornato dinamicamente nel corso dell'interazione.



Figura 3.2: Riconoscimento dell'utente nei sistemi personalizzati

Notiamo che gli m elementi nell'insieme dei profili di utente non sono in relazione biunivoca con gli n elementi dell'insieme dei modelli: un singolo modello, applicato ad utenti diversi, darà luogo ad altrettanti profili. Il numero m dei profili coincide quindi con il numero degli utenti che hanno utilizzato il sistema almeno una volta. Qual' è allora il ruolo dei due database, e come si comporta questo sistema di personalizzazione?

- La prima volta che un utente U_j accede al sistema, viene effettuato un primo riconoscimento delle sue caratteristiche: l'utente viene attribuito ad una delle classi predefinite e quindi viene attivato un *modello* UM_i che descrive (in modo approssimativo, per default), le sue caratteristiche presunte;
- viene subito fatta una copia del modello UM_i in un *profilo* UP_j ; questo viene utilizzato per scegliere l'interazione più opportuna;
- ad ogni interazione successiva, il profilo UP_j viene aggiornato con le nuove informazioni via via acquisite sull'utente; l'aggiornamento può consistere nella *modifica dei valori* di alcune caratteristiche oppure nell'*aggiunta* di nuove caratteristiche.

Esempio 3.1

Progettiamo un sistema informativo sulla didattica, orientato a dare informazioni e suggerimenti personalizzati agli studenti della laurea in ICD. Il sistema potrà assumere che uno studente iscritto al 3 anno abbia superato buona parte degli esami del primo biennio e sia interessato soprattutto a ricevere informazioni sui complementari e sui possibili argomenti di tesi. Il suo modello conterrà quindi questa immagine per default. Ma il suo profilo potrà essere aggiornato, avendo osservato nel corso dell'interazione che si tratta di uno studente lavoratore che ha avuto difficoltà con gli esami di matematica ma che ha già un'idea chiara dell'argomento che sceglierà per la tesi. Se lo stesso studente accederà, in seguito, al sistema informativo (dopo aver superato altri esami), il suo profilo verrà aggiornato con i nuovi dati.

In alcune applicazioni, l'insieme dei profili generati può essere molto diversificato. Questo rende inadeguato un metodo di personalizzazione basato sulla scelta di un'interfaccia all'interno di un insieme predefinito. Occorre allora realizzare un sistema che sia capace di *generare dinamicamente l'interfaccia* adatta ad ogni profilo, utilizzando una base di conoscenza sui metodi di generazione, come indicato in figura 3.3.

Commentiamo la differenza fra questa figura e le precedenti. Rispetto alla figura 3.1, qui:

- il sistema prevede una 'base di conoscenza' sui metodi di generazione delle interfacce,
- il modulo di 'scelta dell'interfaccia' è sostituito da un modulo di 'generazione' dell'interfaccia stessa e, quindi,
- l'interfaccia da utilizzare in ogni contesto non viene prelevata da un DB di alternative predefinite ma viene generata 'on the fly'.

Il modulo di riconoscimento dell'utente in figura 3.3 viene sviluppato come indicato nella figura 3.2.

Questo metodo di personalizzazione è (come è già evidente, e vedremo meglio in seguito) più complesso da realizzare ma permette di ottenere la massima flessibilità evitando di memorizzare in un database di interfacce un numero molto elevato di alternative.

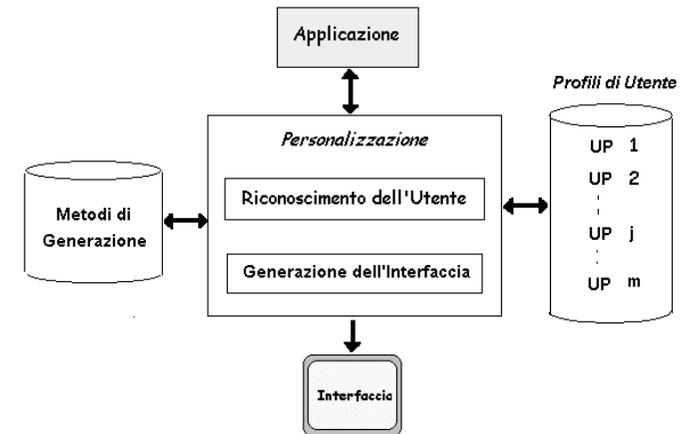


Figura 3.3: Generazione dinamica di Interfacce

Esempio 3.2

Consideriamo il caso di una Enciclopedia Elettronica che contenga la descrizione e spiegazione di un numero x molto elevato di concetti e immaginiamo che y diverse categorie di soggetti possano accedere a questa Enciclopedia, con esigenze conoscitive diverse. Occorrerà allora prevedere la capacità, per l'enciclopedia, di fornire $x*y$ diverse spiegazioni. Se si tiene conto però del fatto che ogni utente chiede, in genere, spiegazioni su un sottoinsieme molto limitato delle y voci, si capirà che memorizzarle tutte, in tutte le possibili versioni, è estremamente costoso, e che conviene pensare invece ad un metodo di generazione automatica della spiegazione, che viene attivato ad ogni richiesta.

Ai metodi di personalizzazione che abbiamo descritto vengono dati nomi diversi.

Definizione 3.1

Un sistema si dice

adattabile all'utente se, per ogni task compreso nell'applicazione, vengono definiti modi diversi di eseguirlo e si lascia all'utente la scelta della modalità da utilizzare di volta in volta;

adattato all'utente se contiene diverse interfacce per la stessa applicazione (una per ciascuna delle classi di utenti potenziali), se ogni versione del sistema viene 'settata' inizialmente in relazione all'utente che lo utilizzerà e se questa scelta può essere cambiata nel corso dell'interazione soltanto con l'intervento di un esperto esterno;

adattivo se l'adattamento viene effettuato automaticamente e in modo dinamico, sulla base di una osservazione permanente delle caratteristiche dell'utente.

La figura 3.1. descrive l'architettura di un sistema adattato all'utente, la 3.3 quella di un sistema adattivo; un sistema 'adattabile' non avrà, invece, bisogno di modelli di utente. Anche se non esistono regole rigide su quale soluzione sia preferibile adottare in ogni situazione, si può dire che, se le caratteristiche degli utenti sono stabili anche se diverse nelle diverse categorie, sarà preferibile realizzare un sistema adattato all'utente. Se invece è probabile che cambino nel tempo, sarà preferibile una soluzione adattiva, anche se più complessa da realizzare. Da notare, infine, che non tutti i task di una applicazione devono essere necessariamente realizzati con una interazione flessibile: è anzi comune identificare, fra i diversi task, quelli per i quali l'adattamento è realmente necessario. Questo è uno dei tipici casi in cui, prima di decidere se e come adattare, vale la pena di realizzare uno studio di valutazione del tipo 'Wizard of Oz'.

Esempio sul Caso di Studio 3.2

Un modo semplice per introdurre flessibilità nell'interazione con SIMB consiste nel realizzare interfacce diverse per le diverse categorie di utenti individuate nella fase di user analysis: ad esempio, ambulatori organizzati con visite per appuntamento o per ordine di arrivo; sistema per il medico o per la segretaria. Il problema di una soluzione di questo tipo è che, se il medico cambia, ad esempio, il modo di organizzare il suo ambulatorio, il sistema dovrà essere re-installato e l'utente dovrà imparare ad usare la nuova versione.

Esempio 3.3

L'interazione per manipolazione di oggetti grafici o mediante comandi da tastiera, nelle applicazioni in Windows, è un esempio tipico di sistema adattabile. Uno dei primi esempi di sistema adattivo, ancora in Windows (non molto apprezzato dagli utenti, per la verità!) di sistema adattato all'utente, ancora in Windows, è l'aiuto fornito da Assistant. Qui, la funzione di assistenza (realizzata mediante un antenato degli Agenti Animati che vedremo nel Capitolo 6) viene attivata automaticamente quando il sistema 'intuisce' (attraverso l'osservazione delle azioni compiute dall'utente) che l'utente stesso si trova in difficoltà. Il generatore di pagine web personalizzate che vedremo nel Capitolo 6 è un altro esempio di adattività.

Esercizio 3.1

Considera le seguenti applicazioni, e discuti in quali casi ritieni che sia desiderabile introdurre una qualche forma di flessibilità nell'interazione, e quale: sistema informativo per gli sportellisti di pagamenti bancari e postali; sistema informativo sui Corsi di Laurea in informatica; sistema per l'affitto di video; sistema per la scelta e la presentazione di programmi di TV digitale.

Nota che una delle caratteristiche che influenzano le scelte di adattamento è il supporto tecnologico utilizzato (ad esempio, pc o palmare).

Esempio 3.4

Considera un semplice sistema di accesso a siti web: vista la diversa dimensione dello schermo e le diverse modalità di input (che abbiamo discusso nel Capitolo 2), l'adattamento al supporto tecnologico è essenziale, se si vuole ottenere la visibilità dell'informazione associata al sito con supporti diversi. Puoi verificare questa esigenza connettendoti con un palmare ad un sito web che conosci bene. Riprenderemo, nel Capitolo 6, questo problema, descrivendo per grandi linee come possa essere realizzato un sistema che contiene questo tipo di adattamento.

Dopo questa premessa generale, possiamo passare ad esaminare i dettagli. In particolare: che cos'è un modello di utente? Qual è la sua struttura, il suo contenuto? Come si costruisce? Come si riconosce quale modello si adatta meglio ad un particolare utente? Come si riconosce che un profilo è errato e va cambiato? Come si aggiorna un profilo? Daremo, nelle prossime Sezioni, qualche risposta di metodo a questi quesiti.

3.2. Metodi per la modellizzazione di utenti

Un sistema 'personalizzato' deve dimostrare di essere preferibile ad uno 'standardizzato': non deve quindi richiedere all'utente uno sforzo eccessivo (neanche nella fase di attivazione), ma anzi deve facilitare il suo lavoro; non deve dargli l'impressione di violare il suo diritto alla privacy o di fare su di lui ipotesi errate che l'utente non può modificare. Queste esigenze comportano diverse conseguenze sul modo di costruire ed aggiornare un modello di utente.

3.2.1 Acquisizione delle informazioni

I dati da inserire nel modello possono essere acquisiti in modo implicito o esplicito:

- l'acquisizione *esplicita* implica che venga richiesto all'utente di rispondere ad una serie di domande orientate a raccogliere le informazioni necessarie per la personalizzazione;
- quella *implicita*, invece, limita al minimo le domande, attiva un primo modello approssimativo e quindi lo aggiorna attraverso l'osservazione del comportamento dell'utente.

Ci sono vantaggi e svantaggi in ognuno dei due metodi. Il vantaggio principale dell'acquisizione implicita è, ovviamente, nella riduzione dell'impegno chiesto all'utente per costruire il suo modello, mentre il suo principale svantaggio è nel rischio di errori dovuti ad assunzioni errate. D'altra parte, l'acquisizione *esplicita*, oltre a richiedere un livello di collaborazione elevato da parte dell'utente (che deve rispondere a numerose domande prima ancora di iniziare ad interagire col sistema) non garantisce di raccogliere informazioni precise: gli utenti possono (a seconda della loro personalità) tendere a sottovalutare o sopravvalutare le loro conoscenze e capacità e possono non avere le idee chiare sulle loro preferenze. A meno di modelli molto semplici, quindi, l'acquisizione implicita viene in genere preferita.

Un buon modello deve però essere aggiornato, oltreché costruito. Questo significa che occorre tenere l'utente sotto continua osservazione, per verificare se le informazioni di default assunte inizialmente sono

corrette o devono essere riviste. Questo richiede che l'interfaccia comprenda un modulo capace di 'osservare' quello che l'utente fa, 'interpretarlo' alla luce delle informazioni contenute nel profilo e modificare una o più di queste informazioni, se necessario. Vedremo nella prossima Sezione come questo possa essere fatto.

Esempio 3.5

Un esempio di acquisizione implicita: in Assistano, il comportamento dell'utente viene osservato (azioni fatte, errori, tempi di esecuzione) e viene interpretato per inferire quali sono i suoi obiettivi e se sa o no eseguire il task corrente; il suo modello viene aggiornato di conseguenza; se la conclusione è che l'utente non sa eseguire il task, viene attivato il sistema di aiuto che corrisponde all'obiettivo presunto.

Esempio 3.6

Nell'Enciclopedia Elettronica, il modello dell'utente può essere acquisito sia in modo esplicito che in modo implicito. Nel primo caso, all'inizio dell'interazione viene chiesto all'utente di compilare un questionario nel quale si chiede a quale parte dell'Enciclopedia è interessato e qual'è il suo grado di conoscenza dei concetti inclusi in quella parte. Questa informazione può venire aggiornata o no, nel corso dell'interazione. Nel secondo caso, l'utente entra direttamente nell'applicazione; la richiesta d'informazione su un concetto viene assunta come indicatore d'interesse per quel concetto e per altri eventualmente correlati. Il tipo di 'visita' (più o meno approfondito) effettuata nell'ipertesto che spiega un concetto può essere assunto come indicatore del grado di conoscenza di quel concetto, ecc.

3.2.2 Modellizzazione per stereotipi

Il metodo comunemente adottato per conciliare le due esigenze contrastanti di ridurre al minimo le informazioni acquisite in modo esplicito costruendo però una immagine dell'utente che contenga le informazioni necessarie per l'adattamento consiste nell'accettare l'idea di costruire *modelli approssimati e incompleti* applicando a questo dominio il concetto di *stereotipo*. Il Dizionario Zingarelli (1999) definisce così questo termine:

“Lo stereotipo è una percezione o un concetto relativamente rigido ed eccessivamente semplificato o distorto della realtà, in particolare di persone o di gruppi sociali. Si dicono ‘stereotipi linguistici’ quelle espressioni proverbiali o singole parole nelle quali si riflettono pregiudizi e opinioni, spesso negative, su gruppi sociali, professionali, etnici.”

Malgrado questa connotazione negativa, il metodo è utilizzato nella vita comune per evocare, in modo veloce, approssimativo (e spesso discutibile), caratteristiche di gruppi di soggetti: *“i milanesi sono legaioli, mangiano risotto e tengono ai soldi”*; *“le donne sono intuitive e sensibili ma hanno una limitata capacità logica”*... eccetera. In questi due stereotipi, basta una sola proprietà (essere milanesi, o essere donne) per evocare molte altre, non necessariamente vere ma probabili, plausibili. E' possibile, inoltre combinare

diverse proprietà per costruire stereotipi più complessi. Così: *“le donne milanesi sono legaiole, mangiano risotto, tengono ai soldi, sono intuitive e sensibili ed hanno una limitata capacità logica”*.

Elaine Rich, nel 1984, ha introdotto per la prima volta questo concetto nel campo dell'adattamento dell'interazione, con la seguente definizione:

Definizione 3.2

uno stereotipo è *“a collection of frequently occurring characteristics of users”*.

Gli stereotipi secondo cui gli utenti possono essere classificati sono rappresentabili in un grafo orientato i cui archi corrispondono a relazioni di *specializzazione*. Ad ogni nodo nella gerarchia sono associati uno o più *attributi*; la proprietà di specializzazione implica che, scendendo nella gerarchia, si aggiungano nuovi attributi e che ogni nodo *erediti* gli attributi dei nodi-genitori.

Nell'idea di Elaine Rich (applicata a personalizzare al sistema GRUNDY di ricerca di libri in una biblioteca), un individuo è caratterizzato dall'appartenenza ad una o più sottocategorie e ne eredita tutti gli attributi (Vedi Scheda 3.1).

Esempio relativo al Caso di Studio 3.3

Se, anziché età e sesso, i fattori che contano nella modellizzazione delle caratteristiche del medico sono la sua attività e il suo rapporto di lavoro, il grafo degli stereotipi potrebbe avere la struttura delineata nella figura 3.4. Un medico può esercitare soltanto come medico generico (con lunga o breve esperienza), o anche come specialista o come ospedaliero, in regime privato o di convenzione con il Servizio Sanitario Nazionale. I medici di base con esperienza breve conoscono le patologie e i farmaci più comuni, mentre quelli con esperienza più lunga conoscono anche le patologie più complesse e i farmaci meno usati, eccetera.

Definizione 3.3

Chiamiamo, rispettivamente, *trigger* e *body* le parti dello stereotipo che contengono le informazioni iniziali e quelle dedotte: le informazioni nel *trigger* attivano lo stereotipo, consentendo di dedurre le informazioni contenute nel *body*.

Nel corso dell'interazione, le informazioni nel *body* dello User Profile verranno progressivamente aggiornate mentre quelle nel *trigger* resteranno (almeno a breve termine) invariate.

Qual'è la differenza tra le variabili rappresentate nella parte *trigger* del modello e quelle rappresentate nel suo *body*? Poiché le variabili nel *trigger* hanno, come abbiamo detto, la funzione di attivare lo stereotipo che meglio descrive la categoria a cui un determinato utente appartiene, queste saranno *in numero limitato e semplici da ricavare*: corrisponderanno, in sostanza, alle *variabili indipendenti* che descrivono la categoria. Le variabili nel *body*, invece, sono quelle che verranno utilizzate per scegliere o generare l'interazione personalizzata: sono le *variabili dipendenti* e potranno essere in numero molto maggiore.

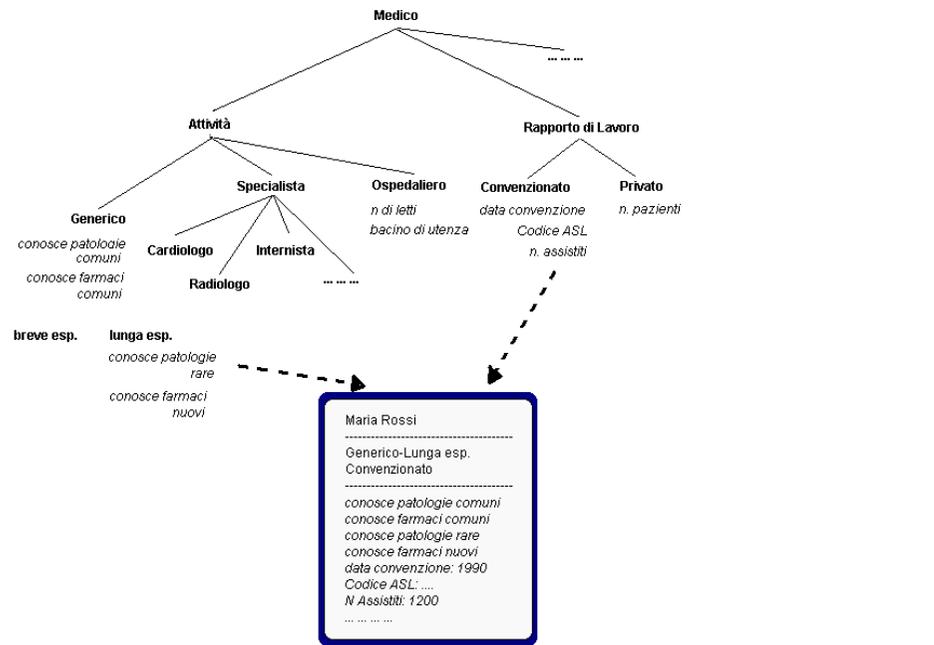


Figura 3.4: stereotipi nel dominio medico

Esempio 3.7
 Variabili indipendenti nello stereotipo "Matricole del Corso di ICD": anno d'iscrizione = 1° e luogo di residenza ≠ Bari. Variabili indipendenti: conosce il Pascal, il C; conosce Windows ma non Linux; non conosce l'ingegneria del software, l'HCI, Ha interesse a trovare casa e amici a Bari; desidera tornare a casa nel finesettimana, ...

Esercizio 3.2
 Prova a descrivere la gerarchia degli stereotipi e le variabili ad essi associate, in un sistema informativo sulla didattica per gli studenti di ICD. Sviluppando l'esempio 3.6, descrivi qualche categoria di utenti il cui stereotipo è definito dalla combinazione di sottocategorie diverse.

Scheda 3.1: Stereotipi in E Rich

Negli stereotipi utilizzati in Grundy, i 'values' (pesi associati ad ogni 'facet', cioè caratteristica) variano nell'intervallo (-5, 5) mentre i 'ratings' variano nell'intervallo (0, 1000) e rappresentano il livello di incertezza associato alla coppia (caratteristica, valore). Come si nota, entrambi gli stereotipi in questo esempio appartengono alla categoria, 'ANY-PERSON', ma le caratteristiche descritte nei loro body sono completamente diverse. Uno sportivo è (molto probabilmente) interessato allo sport, ha un'elevata 'forza fisica', 'tolleranza per la violenza', ecc. mentre può essere poco romantico e non necessariamente istruito.

SPORTS-PERSON		
FACET	VALUE	RATING
Activated-by	Athletic-w-trig	
Genl	ANY-PERSON	
Motivations		
Excite	3	600
Interests		
Sports	4	800
Thrill	5	700
Tolerate-violence	4	600
Romance	-5	500
Education	-2	500
Tolerate-suffering	4	600
Strengths		
Physical-strength	4	900
Perseverance	3	600

Una femminista, invece, è politicamente 'liberal', aperta nei confronti delle questioni politiche femminili, interessata ai conflitti fra sessi, ecc.

FEMINIST		
Activated-by	Feminist-w-trig	
Genl	ANY-PERSON	
Genres		
Woman	3	700
Politics	Liberal	700
Sex-open	5	900
Piety	-5	800
Political-causes		
Women	5	1000
Conflicts		
Sex-roles	4	900
Upbringing	3	800
Tolerate-sex	5	700
Strength		
Perseverance	3	600
Independence	3	600
Triggers	Fem-woman-trig	

Perché le informazioni rappresentate nel modello di utente siano utilizzabili per l'adattamento dell'interazione, occorre che siano 'formalizzate', cioè rappresentate con un formalismo che ne consenta l'elaborazione.

Esaminiamo di nuovo i due esempi di stereotipo di Elaine Rich, nella Scheda 3.1. Qui, ogni attributo (o 'facet') ha un nome (ad esempio 'sports') e può appartenere ad una categoria (ad es, 'Interests'); ad esso è associato un valore numerico, che rappresenta il peso dell'attributo, ed un fattore di certezza. Values e

ratings sono assegnati ad ogni attributo dal progettista, in modo puramente soggettivo, e sono aggiornati mediante un metodo 'ad hoc' (che non segue, cioè, nessuna teoria ben definita per il trattamento dell'incertezza). Il metodo di Rich (interessante per ragioni storiche) è stato in seguito sostituito da una rappresentazione e un trattamento logico o probabilistico della conoscenza.

3.2.3 Rappresentazione della conoscenza nei modelli

La conoscenza su una classe di utenti oppure su uno specifico utente deve essere rappresentata in modo non ambiguo. Il formalismo utilizzato per questa rappresentazione è funzione da un lato del dominio, dall'altro del metodo che s'intende adottare per 'ragionare' su questa conoscenza. Elenchiamo, innanzitutto, le categorie di variabili che possono essere rilevanti nei processi di personalizzazione e i modi per rappresentarle nel body di un Modello o un di Profilo di Utente.

Definizione 3.4

Linguaggio per la descrizione dei modelli di utente.

Dato un utente u , e dati

una variabile x , che denota un *oggetto* (concreto o astratto) e

una variabile a che denota una *azione*

una variabile booleana y che denota un *fatto* (vero o falso)¹

introduciamo le seguenti *formule elementari 'non istanziate'*:²

KnowAbout (u, x) per denotare l'ipotesi: " u conosce x ";

Prefer (u, x) per denotare l'ipotesi: " u preferisce x ";

Like (u, x) per denotare l'ipotesi: " u piace x ";

Is-Interested-In (u, x) per denotare l'ipotesi: " u è interessato a x ";

KnowHow (u, a) per denotare l'ipotesi: " u è capace di eseguire a "

CanDo (u, a) per denotare l'ipotesi: " u è abilitato a eseguire a "

Know (u, y) per denotare l'ipotesi: " u sa che y è vero"

Goal (u, y) per denotare l'ipotesi: " u desidera che y sia vero".

Il body del Modello o del Profilo di uno specifico utente U sarà composto da un insieme coerente di formule elementari 'istanziate', nelle quali la costante U sarà sostituita al posto di u e le variabili x, a, y saranno sostituite con costanti che denotano oggetti, azioni o fatti relativi al dominio. L'insieme delle formule definite in 3.4 può essere esteso per rappresentare altre caratteristiche rilevanti dell'utente.

¹ Indicheremo le variabili con lettere minuscole e le costanti con lettere maiuscole.

² Una formula elementare del tipo $\text{Pred}(t_1, \dots, t_n)$ applica un *predicato* Pred ad un numero n di *termini* t_1, \dots, t_n , ed assume valore di verità (T/F). Se t_1, \dots, t_n sono variabili, la formula si dice 'non istanziata'. In una formula 'non completamente istanziata', alcuni dei termini possono essere costanti.

Esempio 3.8

Nell'esempio 3.6, il body dello stereotipo di una matricola di ICD conterrà formule elementari del tipo: KnowAbout ($u, C++$), KnowAbout ($u, PASCAL$), \neg KnowAbout ($u, SOFT_ENGIN$), \neg KnowAbout (u, HCI), KnowAbout ($u, WINDOWS$), \neg KnowAbout ($u, LINUX$), Is_Interested_In ($u, FINDING_HOUSE$), Is_Interested_In ($u, FINDING_FRIENDS$), Goal ($u, GO_HOME_IN_WEEKEND$).

Notare che la variabile u indica un generico utente: quindi, l'insieme delle formule si applica ad un modello di utente, mentre sarà sostituito dall'identificatore di un particolare utente quando viene costruito il suo profilo.

Esempio 3.9

Un esempio di insieme coerente di formule istanziate nel profilo dell'utente 'Fiorella':

KnowAbout (FIORELLA, HCI), Like (FIORELLA, ENGLISH_LANGUAGE),

\neg KnowHow (FIORELLA, PROGRAMMING_IN_JAVA)

Un esempio di insieme incoerente:

KnowAbout (FIORELLA, HCI), \neg KnowAbout (FIORELLA, HCI)

Esempio sul Caso di Studio 3.3

Maria Rossi è un medico generico convenzionato, con lunga esperienza: in quanto tale, conosce patologie comuni, farmaci comuni e nuovi, mentre non conosce le malattie rare. Il Body del suo profilo avrà le seguenti componenti (oltre ai dati amministrativi):

KnowAbout (MARIA_ROSSI, PATOLOGIE_COMUNI)

KnowAbout (MARIA_ROSSI, FARMACI_COMUNI),

KnowAbout (MARIA_ROSSI, FARMACI_NUOVI)

\neg KnowAbout (MARIA_ROSSI, PATOLOGIE_RARE)

Inoltre, coerentemente con lo stereotipo a cui questo medico appartiene, il suo Body potrebbe contenere altri dati, del tipo:

KnowHow (MARIA_ROSSI, CURARE_INFLUENZA)

KnowHow (MARIA_ROSSI, CURARE_IPERTENSIONE)

CanDo (MARIA_ROSSI, FARE_PRESCRIZIONI)

... eccetera

Qui, PATOLOGIE_COMUNI, PATOLOGIE_RARE, FARMACI_COMUNI e FARMACI_NUOVI sono oggetti nel dominio medico, mentre CURARE_INFLUENZA, CURARE_IPERTENSIONE e FARE_PRESCRIZIONI sono azioni nello stesso dominio.

Esercizio 3.3

Prova a formalizzare gli stereotipi che hai costruito nell'Esercizio 3.2, con il linguaggio descritto nella Definizione 3.4.

3.2.4 Verifiche di coerenza e Aggiornamento dei Profili

Formalizzare le informazioni nei modelli e nei profili di utente come indicato nella Definizione 3.4 è necessario non soltanto per costruire modelli non ambigui, ma anche per memorizzare l'informazione in modo più *compatto*. Questo richiede la capacità di 'ragionare' sulle informazioni memorizzate, per verificarne la coerenza e per 'derivare' nuovi dati. Se infatti si può assumere che uno stereotipo sia stato definito in modo da garantire che il suo body non contenga informazioni incoerenti, occorre anche garantire che questa informazione conservi, dopo ogni fase di aggiornamento, la sua coerenza. Nell'esempio 3.8, abbiamo visto come non sia (ad esempio) accettabile che convivano informazioni del tipo:

KnowAbout (u,X) e \neg KnowAbout (u,X) (che chiameremo *contraddizioni*).

Questo richiede che, ad ogni aggiornamento, sia fatta una *verifica di coerenza* delle informazioni contenute nel body del profilo dell'utente. Vediamo una definizione di coerenza e quindi un esempio.

Definizione 3.5

Dati:

un particolare utente U ed un suo profilo $UP(U)$,

un insieme di formule elementari istanziate, contenute nel Body di $UP(U)$ e rappresentate con il linguaggio descritto nella Definizione 3.5 (che chiameremo *conoscenza specifica sull'utente U*);

un insieme di relazioni tra formule elementari non istanziate, valide per qualsiasi categoria di utenti, che chiameremo *conoscenza generale* sul dominio a cui gli stereotipi UP si applicano:

$UP(U)$ si dice *coerente* se l'applicazione della conoscenza generale alla conoscenza specifica su U non produce contraddizioni.

La logica definisce un metodo per ragionare su insiemi di formule per verificarne la coerenza o per derivare nuove formule. Rimandiamo a testi elementari di logica lo studio delle teorie e dei metodi di ragionamento. Qui, introduciamo innanzitutto una definizione più precisa di conoscenza generale:

Definizione 3.6

Dati:

un linguaggio per la rappresentazione di formule elementari non istanziate (come definito nella 3.4), il quantificatore universale \forall (che leggiamo 'per ogni') e i connettivi \neg (per 'negazione') e \rightarrow (per 'implicazione')

chiamiamo *conoscenza generale* su un dominio un insieme di formule 'ben formate', ottenute combinando (opportunamente) formule elementari non completamente istanziate con i connettivi e il quantificatore.

Esempio 3.10

Un esempio di conoscenza generale sul dominio dell'interazione uomo-macchina, che corrisponde alla 'conoscenza di senso comune', secondo cui tutti quelli che conoscono l'Interazione Uomo-Macchina conoscono anche la Task Analysis: KnowAbout (u, HCI) \rightarrow KnowAbout (u, TASK_ANALYSIS)

Cosa significa 'applicare la conoscenza generale alla conoscenza specifica? In logica, significa utilizzare una regola generale, che si chiama *modus ponens*, per inferire nuove formule da un insieme dato di formule, istanziate o non:

Definizione 3.7

Ragionamento per modus ponens: Dati:

una formula elementare (parzialmente o totalmente istanziate) $Pred1(t_1, \dots, A, \dots, t_n)$, (con A costante e t_1, \dots, t_n variabili) e

una formula non elementare del tipo: $Pred1(t_1, \dots, t_j, \dots, t_n) \rightarrow Pred2(t_1, \dots, t_j, \dots, t_n)$

la regola di *modus ponens* permette di inferire la nuova formula elementare parzialmente o totalmente istanziate: $Pred2(t_1, \dots, A, \dots, t_n)$.

Il ragionamento logico è, in realtà, molto più complesso di quanto abbiamo detto in queste Dispense: le nostre definizioni hanno puro valore esemplificativo (sono una banalizzazione della teoria, ai limiti della volgarizzazione) e gli studenti avranno modo di riprendere in maniera più formalizzata questi concetti, nel corso degli studi. Quanto abbiamo detto ci permette comunque di vedere alcuni esempi di come sia possibile 'ragionare' sui modelli di utente.

Esempio 3.11

Da KnowAbout (u, HCI) \rightarrow KnowAbout (u, TASK_ANALYSIS) e KnowAbout (FIORELLA, HCI)

si inferisce, per modus ponens: KnowAbout (FIORELLA, TASK_ANALYSIS)

Esempio 3.12

Assumiamo di sapere (conoscenza generale nel dominio della didattica dell'informatica) che:
 uno studente che conosce un linguaggio di programmazione è capace di scrivere programmi semplici in quel linguaggio: $\text{KnowAbout}(u, x) \rightarrow \text{CanWriteProgramsIn}(u, x)$;
 uno studente capace di scrivere programmi in un linguaggio sa anche testarli:
 $\text{CanWriteProgramsIn}(u, x) \rightarrow \text{CanTestSimpleProgramsIn}(u, x)$.
 Assumiamo che il profilo di NICOLA contenga la seguente conoscenza specifica su di lui:
 $\text{KnowAbout}(\text{NICOLA}, C) \rightarrow \text{CanTestSimpleProgramsIn}(\text{NICOLA}, C)$.

Secondo la Definizione 3.5 di coerenza, questo profilo non è coerente: infatti, applicando due volte la regola del modus ponens, si può inferire che $\text{CanTestSimpleProgramsIn}(\text{NICOLA}, C)$, e questa formula è incoerente con quanto scritto nella conoscenza specifica su Nicola:
 $\neg \text{CanTestSimpleProgramsIn}(\text{NICOLA}, C)$.

Notare che, in questo esempio, abbiamo specificato il predicato generale di 'CanDo' introdotto nella Definizione 3.4.

Esercizio 3.4

Prova a verificare la coerenza di questo profilo:

"Mario sa nuotare e gli piace il mare di scoglio. Conosce bene le coste attorno a Bari. Vuole andare a Zaponeta a fare il bagno."

Formalizza questa conoscenza specifica, aggiungi la conoscenza generale 'di senso comune' e verifica se il profilo contiene o no contraddizioni.

I metodi che consentono di fare verifiche di coerenza consentono anche di *aggiornare un profilo tenendo conto non soltanto della particolare informazione raccolta, ma anche delle sue conseguenze*. Questo permette di partire da un body limitato, con una conoscenza espressa in modo più *compatto*, e di estenderlo nel corso dell'interazione con le informazioni inferite successivamente.

Esempio 3.13

Nell'esempio 3.11, se il body dell'utente Nicola contiene la formula $\text{KnowAbout}(\text{NICOLA}, C)$, potrà essere esteso in modo automatico con le formule:

$\text{CanWriteProgramsIn}(\text{NICOLA}, C)$ e $\text{CanTestSimpleProgramsIn}(\text{NICOLA}, C)$.

Esempio sul Caso di Studio 3.4

Vediamo allora come potremmo organizzare il profilo di Maria Rossi, descritto nell'esempio 3.4, in modo che sia compatto ma, allo stesso tempo, elaborabile per verifiche di coerenza e per derivare nuove informazioni.

Il Body potrebbe contenere la seguente *conoscenza generale*:

$\text{KnowAbout}(u, \text{PATOLOGIE_COMUNI}) \rightarrow \text{KnowHow}(u, \text{CURARE_INFLUENZA})$
 $\text{KnowAbout}(u, \text{PATOLOGIE_COMUNI}) \rightarrow \text{KnowHow}(u, \text{DIAGNOSTICARE_INFLUENZA})$
 $\text{KnowAbout}(u, \text{PATOLOGIE_COMUNI}) \rightarrow \text{KnowHow}(u, \text{CURARE_IPERTENSIONE})$
 $\text{KnowAbout}(u, \text{PATOLOGIE_COMUNI}) \rightarrow \text{KnowHow}(u, \text{DIAGNOSTICARE_IPERTENSIONE})$

 $\text{KnowAbout}(U, \text{FARMACI_COMUNI}) \rightarrow \text{CanDo}(U, \text{FARE_PRESCRIZIONI_COMUNI})$
 $\neg \text{KnowAbout}(U, \text{PATOLOGIE_COMPLESSE}) \rightarrow \neg \text{KnowHow}(u, \text{CURARE_AIDS})$

E la seguente *conoscenza specifica*, che viene attivata dalle informazioni (nel trigger): 'è un medico di base' e 'ha una lunga esperienza':

$\text{KnowAbout}(\text{MARIA_ROSSI}, \text{PATOLOGIE_COMUNI})$
 $\text{KnowAbout}(\text{MARIA_ROSSI}, \text{FARMACI_COMUNI})$,
 $\text{KnowAbout}(\text{MARIA_ROSSI}, \text{FARMACI_NUOVI})$
 $\neg \text{KnowAbout}(\text{MARIA_ROSSI}, \text{PATOLOGIE_COMPLESSE})$

Questo permetterebbe di *derivare informazioni nuove, non rappresentate in modo esplicito nel profilo*, come:

$\text{KnowHow}(\text{MARIA_ROSSI}, \text{CURARE_INFLUENZA})$
 $\text{KnowHow}(\text{MARIA_ROSSI}, \text{DIAGNOSTICARE_INFLUENZA})$
 $\text{KnowHow}(\text{MARIA_ROSSI}, \text{CURARE_IPERTENSIONE})$
 $\text{KnowHow}(\text{MARIA_ROSSI}, \text{DIAGNOSTICARE_IPERTENSIONE})$
 $\text{CanDo}(\text{MARIA_ROSSI}, \text{FARE_PRESCRIZIONI_COMUNI})$
 $\neg \text{KnowHow}(\text{MARIA_ROSSI}, \text{CURARE_AIDS})$

... eccetera.

Queste Informazioni potrebbero essere utilizzate, ad esempio, per adattare la funzione di 'aiuto alla diagnosi e alla terapia' alle caratteristiche di Maria Rossi: non verranno dati suggerimenti diagnostici o terapeutici per malattie come l'influenza o l'ipertensione, mentre verranno dati suggerimenti per malattie più complesse, come l'aids.

3.3. Ragionamento Incerto nei Modelli di Utente

Per verificare la coerenza di profili e aggiornarli, è possibile applicare, come abbiamo visto, i metodi della logica. Le verifiche di coerenza si riducono così alla verifica che l'applicazione della conoscenza generale sul dominio al body del Profilo non generi formule elementari contraddittorie.

Tuttavia, gli esempi che abbiamo visto nelle Sezioni precedenti mostrano che, in realtà, modellizzare conoscenze, preferenze e capacità degli utenti in modo logico e trarre conclusioni logiche da questo insieme di dati significa, nella maggior parte dei casi, irrigidire in modo inaccettabile la conoscenza, sia specifica che generale. Ad esempio, si può essere più o meno sicuri del fatto che Nicola conosca il C (può aver seguito il corso ma non ancora superato l'esame) e anche del fatto che chi conosce un linguaggio sappia come testare correttamente un programma semplice in questo linguaggio. Analogamente, si può essere più o meno sicuri del fatto che a Mario piaccia il mare di scoglio, e che chi ama questo tipo di mare non desideri mai andare alla spiaggia (ad esempio, la ragazza di Mario potrebbe non saper nuotare).

Questo insieme di ragioni giustifica la scelta, crescente negli ultimi anni, di rappresentare modelli e profili di utente che tengano conto di questi fattori d'incertezza e la trattino in modo appropriato. Sebbene i metodi proposti siano diversi, descriveremo quello più diffuso e anche (non solo a nostro avviso) più corretto: le Reti Causali Probabilistiche.

3.3.1. Cos'è una rete Causale Probabilistica (BN)

Definizione 3.8

Le *Reti Causali Probabilistiche* (o 'belief network', che denoteremo con BN) sono grafi orientati e senza cicli, i cui nodi n_i rappresentano variabili a due o più valori. Il nodo n_i si chiama *genitore* del nodo n_j se esiste un arco orientato che va da n_i a n_j ; n_i si dice *figlio* di n_j .

Gli archi rappresentano *relazioni d'ordine* fra le variabili associate ai nodi.

Un nodo si chiama *radice del BN* se non ha genitori, *foglia* se non ha figli. Ad ogni nodo-radice di un BN è associata una distribuzione di probabilità. Ad ogni altro nodo n_i è associata una *tabella di probabilità condizionate* che misura la dipendenza fra i valori assunti dalla variabile associata a n_i e quelli assunti dalle variabili associate ai suoi genitori.

Le variabili associate alle radici si dicono *nascoste*, mentre quelle associate alle foglie si dicono *osservabili*. Le relazioni fra variabili hanno una *forza* che è misurata dalle tabelle di probabilità condizionate.

Esempio 3.14

Un BN con tre nodi (A,B,C) e con gli archi $A \rightarrow B$, $B \rightarrow C$ è aciclico; se aggiungo l'arco $C \rightarrow A$, introduco un ciclo. E aciclico anche il BN che contiene gli archi $A \rightarrow C$, $A \rightarrow B$, $B \rightarrow C$.

Esempio 3.15

La figura 3.5 mostra un esempio di BN con quattro nodi, a ciascuno dei quali è associata una variabile booleana. D e B sono nodi radice: ad essi sono associate le distribuzioni di probabilità $P(D)$ e $P(B)$.

Al nodo A, figlio di D, è associata la tabella a due valori di probabilità condizionate: $P(A|D)$ e $P(A|\neg D)$.

C è un nodo-foglia con due genitori, A e B: a C è associata una tabella di probabilità a quattro valori:

$P(C|A,B)$, $P(C|A,\neg B)$, $P(C|\neg A, B)$, $P(C|\neg A, \neg B)$.

Il BN rappresenta il fatto che la variabile D 'influenza' la variabile A e che A e B influenzano C. La relazione di influenza può essere una relazione causa-effetto (D è causa di A; A e B sono causa di C) o una qualsiasi altra *relazione d'ordine* (D implica A 'con un certo grado di incertezza'; A e B implicano C 'con un certo grado di incertezza').

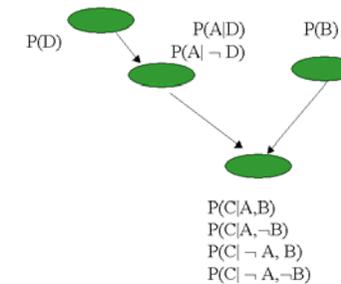


Figura 3.4: Una semplice Rete Causale Probabilistica

Notiamo che, nel caso più semplice di un nodo con un solo genitore (come il nodo A, nell'esempio), la rappresentazione della relazione nell'arco sostituisce, all'implicazione logica, una implicazione 'incerta'. Rappresentata in modo logico, la relazione: $A \rightarrow B$ va interpretata come: "se A è vero, allora anche B è vero; se A è falso, non si può dire nulla su B":

Rappresentata invece come un arco in un BN³, la stessa relazione, che denoteremo con $A \rightarrow^? B$ potrà avere diverse interpretazioni, a seconda dei valori di probabilità associati ad A e a B. Ad esempio:

assumiamo che le probabilità condizionate associate a B siano: $P(B|A) = .75$, $P(B|\neg A) = .25$;

allora: se A è vero, avremo che $P(B) = .75$ (B è molto probabilmente vero); se A è falso, avremo che $P(B) = .2$ (B è molto probabilmente falso). Ma potremo anche trattare situazioni incerte per A, in cui la variabile associata a questo nodo ha, a sua volta, un valore di probabilità.

Ad esempio, se $P(A) = .65$ (A è probabilmente vero), avremo che $P(B) = .575$ (B è molto incerto, con tendenza ad essere vero).

³ E trattata secondo il Teorema di Bayes, teorema fondamentale della Teoria della Probabilità

In modo analogo si possono rappresentare implicazioni incerte in cui la premessa include più di una variabile (come, nell'esempio 3.13, per C).

Esempio 3.16

Nel BN in figura 3.4, potremmo associare ai quattro nodi le seguenti variabili booleane:

- D: "piove"
 A: "fa freddo"
 B: "x teme il freddo"
 C: "x mette l'impermeabile"

Associando al nodo A la tabella $P(A|D) = .8$ e $P(A|\neg D) = .5$, si rappresenta l'informazione che, se piove, molto probabilmente fa freddo mentre, se non piove, non si sa se fa freddo o no.

La seguente tabella associata al nodo C: $P(C|A,B) = .9$, $P(C|A,\neg B) = .3$, $P(C|\neg A, B) = .4$, $P(C|\neg A, \neg B) = .1$ rappresenta l'informazione che C è legato ad A e B da una relazione di 'quasi-AND': x metterà quasi certamente l'impermeabile se fa freddo e piove, quasi sicuramente non lo metterà se non fa freddo e non piove, mentre avrà un comportamento incerto se una sola delle due condizioni è vera (con B che 'pesa' più di A).

Infine, assegnare ai due nodi radice valori di probabilità $P(D) = .5$ e $P(B) = .5$ equivale ad affermare che nulla si sa sul valore di verità di queste due variabili (cioè che non si sa se piove o no e se x teme o no il freddo).

La struttura di un BN rappresenta, implicitamente, un insieme di ipotesi sulla *dipendenza fra variabili*.

Definizione 3.9

Due variabili X e Y sono *condizionalmente indipendenti*, dato Z, se $P(X | Y, Z) = P(X | Z)$.

Questo equivale a dire che $P(X, Y | Z) = P(X | Z) * P(Y | Z)$.

In un belief network, si assume che "la variabile associata ad un nodo è condizionalmente indipendente dalle variabili che **non** sono associate ai suoi figli, dati i suoi genitori".

La definizione di indipendenza condizionale in un BN semplifica, ancora una volta, il problema: rimandiamo, ancora una volta, all'articolo di Spiegelhalter per approfondimenti, mentre ci limitiamo qui a dare alcuni esempi di indipendenza condizionale in BN.

Esempio 3.17

Nel BN 3.14, ad esempio:

$P(C|A,B,D) = P(C|A,B)$: cioè "la probabilità che x metta l'impermeabile se fa freddo e x teme il freddo è indipendente dal fatto che piova", oppure:

$P(B|A) = P(B)$: cioè "la probabilità che x tema (in generale) il freddo non dipende dal fatto che faccia (in un giorno particolare) freddo"

Notare che, se voglio eliminare la seconda ipotesi e rappresentare, ad esempio, il fatto che x è un tipo influenzabile, il cui atteggiamento generale rispetto al freddo dipende dalla temperatura di un giorno particolare, dovrò inserire un arco che va da A a B. In questo caso, non assocerò più a B un valore di probabilità ma una coppia di valori di probabilità condizionate: $P(B|A)$ e $P(B|\neg A)$.

Una volta costruito il BN e associati ai suoi nodi i parametri, la rete può essere utilizzata per *ragionare in condizioni di incertezza* sul dominio rappresentato. Il primo passo consiste nell'applicare un algoritmo che calcola le *probabilità a priori* dei nodi non-radice nel grafo. Non descriveremo qui questo algoritmo: una descrizione dettagliata si può trovare nel lavoro di (Spiegelhalter, ...). Sugeriamo invece a chi voglia esercitarsi nella costruzione di BN, di farlo utilizzando uno dei tanti tool disponibili su rete: ad esempio, Hugin Lite, vedi Scheda 3.2. Gli esempi che faremo d'ora in poi sono stati costruiti con questo tool.

Esempio 3.18

Nell'esempio di figura 3.4, $P(A) = .65$ e $P(C) = .48$. Questo significa che, con i parametri introdotti nel modello, se nulla si sa sul tempo e sulle abitudini di x, non si può prevedere se x metterà o no l'impermeabile.

Esercizio 3.5

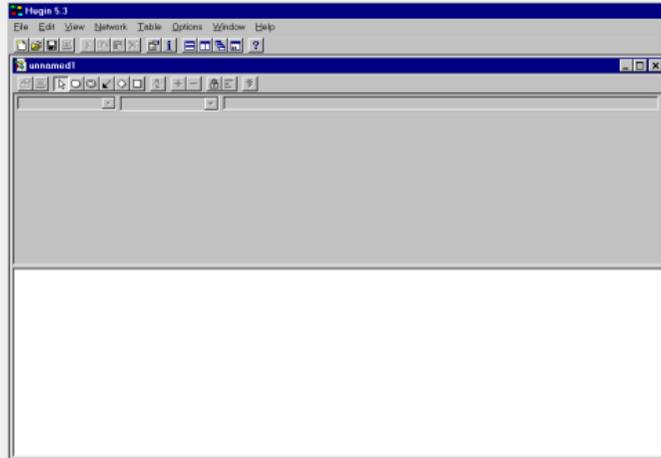
Prova (con Hugin) a lasciare la struttura della figura 3.4 invariata, cambiando i parametri. Verifica come variano le probabilità a priori dei nodi intermedi e foglia, e ragiona su queste variazioni.

Esempio 3.19

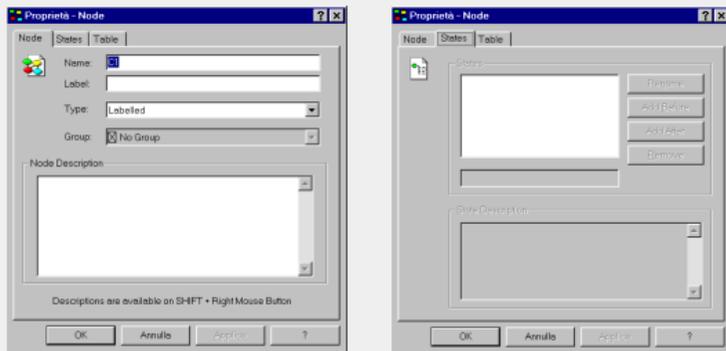
L'idea di consistenza nell'Esercizio 3.4 può essere vista in modo 'soft' ragionando in condizioni d'incertezza e rappresentando la conoscenza con un BN: se una persona sa nuotare, ama il mare di scoglio e conosce le coste attorno a Bari, probabilmente non vorrà andare a Zapponeta a fare il bagno. Ma se va al mare con amici che sanno nuotare poco, probabilmente andrà a Zapponeta anche se non è il suo posto ideale.

Scheda 3.2: Hugin: www-hugin.com

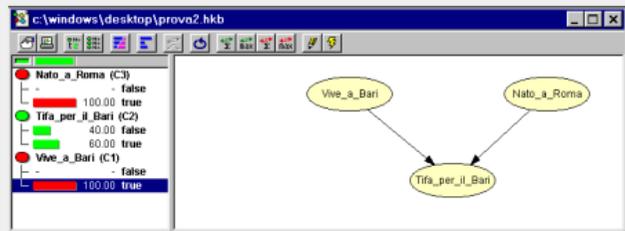
Esistono diversi software che permettono di costruire e gestire Belief Network, molti dei quali sono free. Uno dei migliori (Hugin) è disponibile in due versioni: una versione 'Lite', free ed una 'Professional', con API. Nella versione Lite, si possono costruire BN con un numero limitato di nodi, sufficiente tuttavia per imparare il metodo. La finestra principale di Hugin contiene una toolbar per la *creazione interattiva di network*:



con pop-up window per inserire le proprietà dei nodi (tipo di variabili e probabilità).



La finestra di 'run' permette di propagare evidenza nel BN o di tornare ad aggiornare il BN (struttura e parametri)



Vediamo ora come la rete può essere utilizzata per ragionare su un caso specifico.

Un caso specifico è definito dall'assegnazione di un valore ad una o più delle variabili associate ai nodi del BN. Ad esempio, nel BN di figura 3.4 interpretato secondo l'esempio 3.14:

C è vero ("x ha l'impermeabile").

Ragionare su questo caso significa *inserire evidenza* sui nodi implicati nel caso e *propagarla sull'intera rete* (di nuovo, con un algoritmo descritto, ad esempio, in Spiegelhalter). Inserire evidenza significa assegnare ai nodi della rete i valori definiti; propagare questa evidenza significa calcolare come questa nuova conoscenza cambia la distribuzione di probabilità dei valori assunti dagli altri nodi.

Esempio 3.20

Continuiamo l'esempio 3.14.

Inseriamo l'evidenza: $P(C) = 1$, cioè C è vero ("x ha l'impermeabile").

La conseguenza di questa nuova conoscenza sarà che $P(D) = .55$ ("non si può dire se piove o no"), mentre $P(A) = .82$ ("quasi certamente fa freddo") e $P(B) = .76$ ("molto probabilmente x teme il freddo"): questo è logico, visto che A e B sono in relazione di 'quasi-AND' con C.

Ma l'evidenza può essere introdotta su un qualsiasi nodo del BN. Ad esempio, ritraiamo l'evidenza precedente e inseriamo, invece, l'evidenza: $P(A) = 0$ ("non fa freddo").

La conseguenza di questa nuova evidenza è che $P(D) = .29$ ("è poco probabile che piova") e $P(C) = .25$ ("è poco probabile che x metta l'impermeabile"). Invece B non viene influenzato da questa evidenza ($P(B)=.5$): questo è logico, visto che il fatto che faccia o no freddo non influenza il fatto che x sia o no freddoloso.

Notare che è possibile inserire, contemporaneamente, evidenza su più nodi ("piove e x non è freddoloso", ...). Notare anche che i BN permettono di ragionare 'a partire dalle variabili osservabili', 'a partire dalle variabili nascoste' oppure a partire da una qualsiasi combinazione delle diverse variabili rappresentate.

Esercizio 3.6

Prova a inserire evidenza su altri nodi, nel BN rappresentato in figura 3.4, e ragiona sui risultati.

Dopo questa premessa di metodo, vediamo ora come il metodo descritto possa essere applicato per ragionare sui modelli di utente.

3.3.2. I BN nei modelli di Utente

Per mostrare come i BN siano utilizzabili come formalismo per costruire ed aggiornare modelli e profili di utente, partiamo da un esempio nel dominio dei sistemi intelligenti per l'insegnamento (gli *Intelligent Tutoring Systems*). Qui, gli utenti sono studenti e quindi parleremo di 'modelli di studenti'.

Esempio 3.21

Consideriamo un sistema 'intelligente' per l'insegnamento della matematica, che chiameremo *Gaetana*⁴. *Gaetana* ha un modello degli studenti della sua classe che comprende una parte generale (relativa alla classe) ed una specifica, relativa ad ogni studente. Nella *parte generale* è rappresentata la conoscenza sull'apprendimento della matematica.

Ad esempio:

Le ragazze non capiscono, in genere, la matematica.

Chi non capisce la matematica e non è particolarmente intelligente ha, in genere, difficoltà a capire la trigonometria.

Chi è intelligente ha dato, in genere, prove precedenti di bravura.

La *parte specifica* contiene, invece, una serie di fatti relativi ad uno specifico studente. Ad esempio, nella classe c'è una ragazza, Maria, che ha sbagliato l'ultimo compito: questo fatto sarà memorizzato nel profilo di Maria.

Assumiamo ora che *Gaetana* abbia appena fatto una lezione di trigonometria e voglia verificare se Maria ha capito quello che lei ha spiegato, proponendole un esercizio adeguato al suo livello di preparazione. Come può usare la conoscenza generale e quella specifica per prendere questa decisione?

Introduciamo, prima di tutto, le seguenti notazioni:

$F(x)$: "x è una ragazza"

$Und(x, y)$: "x capisce y"

$Smart(x)$: "x è intelligente"

$Good(x, LASTP)$: "x ha dato prove precedenti di bravura".

Ora, formalizziamo la *conoscenza generale*⁵:

$F(x) \rightarrow ? \neg Und(x, MATH)$

$\neg Und(x, MATH) \wedge \neg Smart(x) \rightarrow ? \neg Und(x, TRIG)$

$\neg Und(x, MATH) \wedge Smart(x) \rightarrow ? Und(x, TRIG)$

$Smart(x) \rightarrow ? Good(x, LASTP)$

Rappresentiamo questa conoscenza nel BN nella parte destra della figura 3.5, con i seguenti valori di probabilità associati ai nodi:

$P(F(x)) = .5$ (nella classe, c'è la stessa proporzione di ragazze e di ragazzi)

$P(Smart(x)) = .5$ (nella classe, c'è la stessa proporzione di studenti intelligenti e non intelligenti)

$P(Good(x, LASTP)|Smart(x)) = .9$ (se uno studente è intelligente, molto probabilmente ha dato prove precedenti di bravura)

$P(Good(x, LASTP)|\neg Smart(x)) = .4$ (se uno studente non è intelligente, è poco probabile che abbia dato prove precedenti di bravura)

$P(Und(x, Math)|F(x)) = .3$ (una ragazza, è poco probabile che capisca la matematica)

⁴ Dal nome di Gaetana Agnesi, matematica del 1700.

⁵ Con il simbolo $\rightarrow ?$ Denotiamo una *relazione incerta fra due variabili*.

$P(Und(x, Math)|\neg F(x)) = .6$ (un ragazzo, è più probabile che capisca la matematica)

$P(Und(x, Trig)|Und(x, Math), Smart(x)) = .95$ (se uno studente capisce la matematica ed è intelligente, quasi certamente capisce anche la trigonometria)

$P(Und(x, Trig)|\neg Und(x, Math), Smart(x)) = .5$ (se uno studente non capisce la matematica ma è intelligente, non si può prevedere se capisce o no la trigonometria)

$P(Und(x, Trig)|Und(x, Math), \neg Smart(x)) = .6$ (se uno studente capisce la matematica ma non è intelligente, potrebbe anche capire la trigonometria)

$P(Und(x, Trig)|\neg Und(x, Math), \neg Smart(x)) = .1$ (gli studenti che non capiscono la matematica e non sono intelligenti, molto probabilmente non capiscono neanche la trigonometria).

Dopo aver rappresentato questo modello generale in un belief network, possiamo verificare che le probabilità iniziali di ciascuno dei nodi sono quelle indicate nella parte in alto a sinistra della figura 3.5: in particolare, la probabilità che un qualsiasi studente x nella classe di *Gaetana* capisca la trigonometria è = .5138.

Introduciamo ora i dati noti a *Gaetana*, con la sostituzione di Maria al posto di x:

$P(F(Maria)) = 1$

$P(Good(Maria, LASTP)) = 0$, cioè $\neg Good(Maria, LASTP)$.

Propaghiamo questa 'evidenza' e ci chiediamo qual è la probabilità del nodo "Und(Maria,TRIG)".

I valori di probabilità nella parte sinistra in basso della figura mostrano che $P(Und(Maria,TRIG)) = .305$.

Quindi, *Gaetana* conclude che non si sa se Maria capisce o no la trigonometria e sceglie per lei, fra quelli possibili, un esercizio facile.

Esercizio 3.7

Prova a cambiare l'esempio precedente considerando il caso seguente:

La maggior parte dei ragazzi e delle ragazze, nella classe di Gaetana, capiscono la matematica.

Chi capisce la matematica e segue con regolarità le lezioni ha, in genere, un buon livello di preparazione in questa materia.

Chi ama i sistemi d'insegnamento automatizzati segue le lezioni in modo regolare.

Gli studenti pendolari tendono ad amare i sistemi d'insegnamento automatizzati.

Costruisci il modello con un tool di gestione di belief network e rispondi al quesito seguente:

Nicola è un pendolare. Qual è la probabilità che abbia un buon livello di preparazione in matematica?

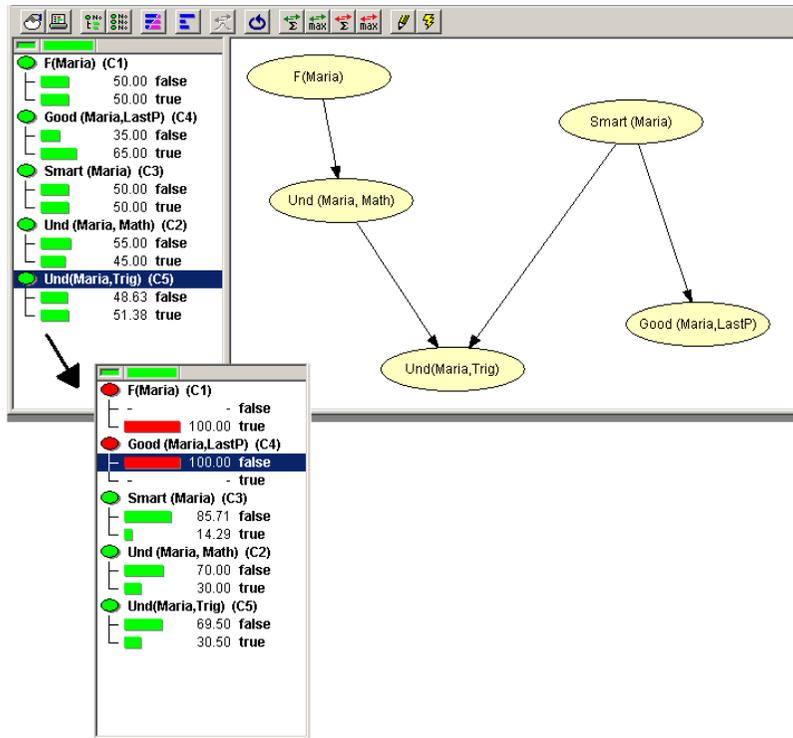


Figura 3.5: Propagazione di evidenza in un BN, con Hugin

L'esempio precedente mostra che Hugin non permette di distinguere fra 'conoscenza generale' e 'conoscenza specifica'. Occorre quindi, quando si ragiona su uno specifico utente, creare una copia del BN che rappresenta la conoscenza generale e propagare in questa rete l'evidenza relativa all'utente considerato.

Esempio sul Caso di Studio 3.5

Introduciamo l'incertezza nell'esempio 3.5 relativo al Caso di Studio e rivediamo allora come potremmo organizzare il profilo di Maria Rossi.

La *conoscenza generale* relativa ad un medico di base esperto può essere rappresentata in uno stereotipo 'medici di base esperti', come nel BN in figura 3.6. In questa figura, il nodo radice contiene le informazioni trigger e i nomi dei nodi sono stati semplificati per ragioni di spazio. Gli archi rappresentano le relazioni fra i diversi elementi nel modello, con un verso che va 'dalle conoscenze alle capacità' (nell'ipotesi che le conoscenze, ad esempio sulle malattie, siano condizione necessaria per la capacità di diagnosticarle e

trattarle) e 'dal generale al particolare' (nell'ipotesi che la conoscenza di una classe di concetti sia indice di conoscenza degli elementi che a questa classe appartengono):

```
KnowAbout (u, PATOLOGIE_COMUNI) →? KnowHow (u, CURARE_INFLUENZA)
KnowAbout (u, PATOLOGIE_COMUNI) →? KnowHow (u, DIAGNOSTICARE_INFLUENZA)
KnowAbout (u, PATOLOGIE_COMUNI) →? KnowHow (u, CURARE_IPERTENSIONE)
KnowAbout (u, PATOLOGIE_COMUNI) →? KnowHow (u, DIAGNOSTICARE_IPERTENSIONE)
....
KnowAbout (U, FARMACI_COMUNI) →? CanDo (U, FARE_PRESCRIZIONI_COMUNI)
→ KnowAbout (U, PATOLOGIE_COMPLESSE) →? → KnowHow (u, CURARE_AIDS)
```

I parametri da inserire della rete possono essere definiti in modo 'puramente soggettivo' oppure a partire da una elaborazione statistica di una base di dati sul dominio. In particolare, nel nostro esempio possiamo introdurre i parametri seguenti:

```
P(KnowAbout (u, PATOLOGIE_COMUNI) | MedicoEsperto(u)) = .85
P(KnowAbout (u, PATOLOGIE_COMUNI) | ¬ MedicoEsperto(u)) = .3
P(KnowHow (u, CURARE_INFLUENZA) | KnowAbout (u, PATOLOGIE_COMUNI)) = .9
P(KnowHow (u, CURARE_INFLUENZA) | ¬ KnowAbout (u, PATOLOGIE_COMUNI)) = .35
... e così via....
```

Per ragionare su Maria (un particolare medico di base esperto), creiamo una copia del BN (in cui la variabile u è sostituito dalla costante MARIA_ROSSI) e inizializziamo così il suo profilo.

Introduciamo ora nel profilo la *conoscenza specifica* relativa a Maria. Notiamo che si può introdurre, sui nodi-radice, una evidenza 'incerta', mentre sugli altri nodi è possibile soltanto inserire evidenza 'certa': se le variabili sono booleane, come nel nostro esempio, la variabile associata ad un nodo può essere stabilita come 'vera' o 'falsa'.

```
P(ExpertGP (MARIA_ROSSI)) = .8 "Maria Rossi è un medico di base, molto probabilmente esperto"
P(KnowAbout (MARIA_ROSSI, PATOLOGIE_COMUNI)) = 1
P(KnowAbout (MARIA_ROSSI, PATOLOGIE_COMPLESSE)) = 0
```

Propaghiamo questa evidenza nel BN, e deriviamo *informazioni nuove e incerte*, come:

```
P(KnowHow (MARIA_ROSSI, CURARE_INFLUENZA)) = ?
P(KnowHow (MARIA_ROSSI, DIANOSTICARE_INFLUENZA)) = ?
P(KnowHow (MARIA_ROSSI, CURARE_IPERTENSIONE)) = ?
P(KnowHow (MARIA_ROSSI, DIAGNOSTICARE_IPERTENSIONE)) = ?
P(CanDo (MARIA_ROSSI, FARE_PRESCRIZIONI_COMUNI)) = ?
P(KnowHow (MARIA_ROSSI, CURARE_AIDS)) = ?
```

... eccetera.

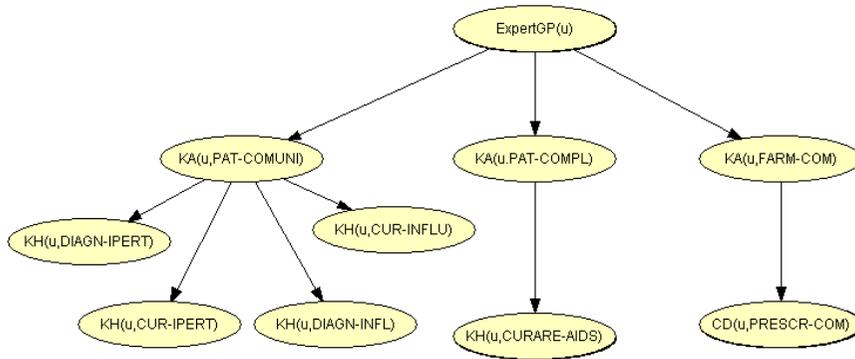


Figura 3.6: Rappresentazione di conoscenza generale in un BN

Esercizio 3.8

Prova a completare l'esempio precedente inserendo tutti i parametri e propagando l'evidenza come nell'esempio. Ragiona sui risultati. Verifica come cambiano i valori delle 'informazioni nuove' al variare di alcuni parametri.

Costruire un modello e un profilo dell'utente è il primo passo del processo di adattamento dell'interazione. Come abbiamo visto dagli esempi in questo Capitolo, il contenuto dei due modelli dipende fortemente dal dominio a cui l'applicazione si riferisce: ad esempio, il contenuto del 'modello per il medico di base' sarà diverso dal 'modello di studente' utilizzato da Gaetana. Indipendentemente dal metodo che si sceglie di applicare per acquisire l'informazione e costruire ed aggiornare il modello, una regola fondamentale da seguire è quella del *realismo*. Occorre inserire, nel modello, *soltanto le informazioni che sono necessarie per adattare l'interazione*. Quali informazioni e quante, dipende quindi da cosa è possibile e si è ritenuto utile adattare.

Facciamo due esempi 'estremi':

Caso 1: si vogliono costruire due versioni di una applicazione: una, per gli utenti 'inesperti', in cui l'interazione avviene con il mouse. E un'altra, indirizzata agli utenti 'esperti', in cui l'interazione avviene mediante comandi da tastiera. In questo caso, non sarà necessario avere nessun modello di utente, ma basterà la semplice informazione sul suo 'livello di esperienza'.

Caso 2: si vuole costruire una Enciclopedia Elettronica in cui il modo di spiegare ciascuno dei concetti compresi nell'Enciclopedia sia adattato al livello di conoscenza che l'utente ha *per quel concetto*. In questo caso, se l'Enciclopedia contiene n concetti C_i ($i = 1, \dots, n$), avremo bisogno di inserire, nel profilo dell'utente U , n elementi del tipo: KnowAbout (u, C_i). Potremo ridurre l'ampiezza del Profilo creando una conoscenza generale che stabilisce le relazioni fra le conoscenze di alcuni concetti, nella forma:

$$\text{KnowAbout}(u, C_i) \wedge \text{KnowAbout}(u, C_j) \wedge \dots \wedge \text{KnowAbout}(u, C_k) \rightarrow^? \text{KnowAbout}(u, C_h)$$

E cioè "la conoscenza dei concetti C_i, C_j, \dots e C_k è (presumibilmente, probabilmente, ...) prerequisito per la conoscenza del concetto C_h .

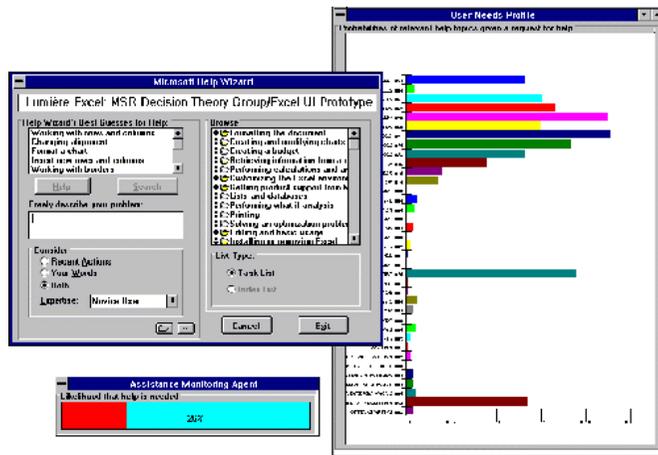
Potremmo rappresentare questa conoscenza generale sottoforma di un BN e poi aggiornare la conoscenza specifica sui concetti conosciuti da uno specifico utente U che sta interagendo con il sistema, nel modo che abbiamo visto in questo Capitolo.

La Scheda 3.3 descrive sinteticamente come i BN siano stati applicati nel Progetto Lumière della Microsoft. Maggiori dettagli su questo progetto si possono trovare nelle pubblicazioni di Eric Horvitz (responsabile del gruppo di ricerca)

Vedremo, nei Capitoli 5 e 6, quali aspetti dell'interfaccia possano essere personalizzati (linguaggio grafico d'interazione, linguaggio naturale o comportamento di Agenti Animati) e come.

Scheda 3.3: Il Progetto Lumiere

Questo Progetto è stato avviato dalla Microsoft nel 1993 con l'obiettivo di ragionare sugli obiettivi e le necessità di aiuto degli utenti durante la loro interazione con le applicazioni. Il cuore del Progetto è costituito dai modelli bayesiani, che rappresentano le relazioni fra obiettivi ed esigenze dell'utente ('variabili nascoste') e stato del programma, sequenza di azioni eseguite nel tempo e formulazione delle query ('osservabili'). In una prima fase, sono state investite circa 25 mila ore in studi di usabilità, nei quali veniva analizzato il modo di lavorare degli utenti e i problemi incontrati nell'uso del software (Office 97). Questi studi erano basati su analisi di videotape, trascritti di 'pensieri a voce alta' fatti dagli utenti e studi basati su Wizard of Oz. E' stato poi definito un metodo per interpretare le query dell'utente basato su reti bayesiane i cui nodi rappresentano il background dell'utente, le sue azioni, lo stato del programma e l'interpretazione possibile delle parole. Inizialmente, la variabile 'necessità di aiuto' ha una distribuzione di probabilità che è costruita sulla base degli studi di fattibilità prima menzionati: vedi figura:



Quando l'utente compie una azione o formula una query in un particolare contesto, questa evidenza viene propagata nella rete e la distribuzione di probabilità viene rivista, come indicato nella figura seguente: vengono selezionate le forme di aiuto con probabilità più elevate e viene mostrato il messaggio di help corrispondente.

