Università degli Studi di Napoli
"Federico II"

Facoltà di Scienze MM.FF.NN.

Corso di Laurea quinquennale in Informatica

Tesi di Laurea

PerfectFit06:
Sistema di personalizzazione dei contenuti web per gli scavi archeologici di Ercolano

Relatore:

Prof. Ernesto Burattini

Candidato:

Vincenzo Scognamiglio

Matricola: 050/181

Anno Accademico 2005/2006

Indice

Introduzione 5

Capitolo I: Sistemi di personalizzazione e agenti Web

1.1 Introduzione 11
1.2 Definizione di agente 13
1.2.1 Tipologie di agenti 14
1.3 I sistemi di personalizzazione Web 15
1.4 I sistemi di raccomandazione 17
1.5 Il modello utente 18
1.5.1 Le caratteristiche dell'utente 18
1.5.2 Tipi di modelli utente 20
1.5.3 Rappresentazione del modello utente 22
1.5.4 Tipi di sistemi di personalizzazione in base al modello utente 23
1.6 Aspetti adattabili nel Web 24
1.7 Fasi del processo di modellizzazione 26
1.8 Tecniche per la personalizzazione 28
1.8.1 I metodi per l'identificazione e il tracking dell'utente 28
1.9 Il filtraggio delle informazioni 29
1.9.1 Il filtraggio collaborativo e il filtraggio cognitivo 29
1.9.2 Tecniche per la rappresentazione dei significati 31
1.9.3 Semantic Web 32
1.10 I sistemi basati su regole 33
1.11 Capacità di adattamento all'ambiente dell'utente 34

Capitolo II: Architettura di PerfectFit06

2.1 Evoluzione dei sistemi ipermediali di Ercolano 36
2.2 Web e scavi di Ercolano, scopi di PerfectFit06 38
2.3 Adattività di PerfectFit06 40
2.4 Introduzione all'architettura di PerfectFit06 42
2.5 Grafo della conoscenza 45
2.6 Adattamento del grafo al Web 47
2.7 Rappresentazione del grafo nel sistema web 51
2.8 Le "risorse" 53
2.9 Il modello risorsa 56
2.11 Il modello utente di PerfectFit06 60
2.12 Struttura del modello utente di PerfectFit06 61
2.13 Adattatività strutturale di PerfectFit06 rispetto al modello 62
2.14 Inizializzazione del modello utente 64
2.15 Regole di associazione e regole di estensione 65
2.15.1 Le regole di associazione 66
2.15.2 Le regole di estensione 68
2.15.3 Modello risorsa ideale: creazione e utilizzo per il calcolo delle risorse interessanti 69
2.15.4 Calcolo automatico delle regole di estensione 72
2.16 Riepilogo attività di PerfectFit06 75
2.17 Tecniche per velocizzare l'adattività: modello utente normalizzato 77

Capitolo III: Utilizzo di PerfectFit06

3.1 PerfectFit06 in pratica 80
3.2 L'interfaccia di PerfectFit06 82
3.2.1 La barra superiore 84
3.2.2 La colonna "Risorse accessibili" 86
3.2.3 Colonna risorse 88
3.3 La raccolta degli appunti 90
3.4 Le regole di estensione 93
3.4.1 Tabella delle regole di estensione 94
3.4.2 Applicazione delle regole di estensione 97
3.5 L'adattatività di PerfectFit06 100
3.6 Esempio pratico di funzionamento 102
3.6.1 Test di funzionamento: utente turista con interessi storici 103
3.6.2 Test di funzionamento: utente archeologo con interessi antiquari 107
3.7 Scelte progettuali in PerfectFit06 111
3.7.1 Scelta dei linguaggi di programmazione 112
3.7.2 Scelta del database 115
3.8 Evoluzione del sistema 116
3.8.1 PerfectFit06 come Agente Web 117
3.8.2 Modulo esperto 120
3.8.3 Miglioramenti generici del sistema 121

Appendici

Appendice A – Regole di associazione 123
Appendice B – Regole di estensione calcolate da PerfectFit06 124
Appendice C – Modelli risorsa 125
Appendice D – Pagina web di PerfectFit06 133
Appendice E – Libreria di funzioni di PerfectFit06 139
Appendice F – Codice delle pagine per la visualizzazione dello stato di PerfectFit06 152
Appendice G – Codice MySQL 155

Glossario

Glossario specifico di perfectFit06 158

Bibliografia

Bibliografia consultata 159
Bibliografia di riferimento 162
Siti web consultati 165

PerfectFit06:

Sistema di personalizzazione dei contenuti per gli scavi archeologici di Ercolano

Introduzione

Il World Wide Web si è imposto, in maniera sempre più determinante, come un potentissimo strumento di diffusione delle informazioni su scala internazionale.
La sua duttilità, la semplicità d'utilizzo da parte di utenti inesperti e la facilità con cui si possono realizzare siti web, sono diventati gli elementi che hanno maggiormente incentivato il crescente interesse nei suoi confronti sia da parte di esperti del settore che non.
La nascita di un numero incredibile di siti web ha portato con se la possibilità di trovare praticamente qualsiasi tipo di informazione all'interno del Web.
E' naturale che una disponibilità così ampia di materiale, evolutasi senza regole di base, porti dei problemi con se, primo fra tutti il disorientamento dell'utente.
Questo problema del disorientamento dell'utente non è riscontrabile solo sull'intero contenuto informativo del Web, ma può presentarsi anche durante la navigazione di un singolo sito web il cui carico cognitivo è eccessivo o semplicemente mal gestito.
Il sistema PerfectFit06 si inserisce proprio all'interno di questo contesto dove il disorientamento dell'utente, all'interno di un sito web con alto contenuto informativo, è un fattore di rischio elevato.
PerfectFit06 non è un sistema a parte rispetto ad un sito web, ma ne è parte integrante, si può dire senza sbagliare che PerfectFit06 è un sito web con una architettura molto più articolata.
Lo scopo principale di PerfectFit06 è riuscire a catturare, in maniera non intrusiva, gli interessi dell'utente durante la navigazione, così da potergli suggerire o mostrare solamente i contenuti informativi per lui significativi.
Durante la progettazione di PerfectFit06 si è mirato a realizzare una architettura quanto più possibile flessibile, in modo da rendere perfettamente riutilizzabili sia idee che caratteristiche implementate.
Si sono usati come caso di studio gli scavi archeologici di Ercolano, particolarmente adatti a mostrare i problemi legati a un grosso quantitativo di contenuti estremamente eterogenei sia per tipologia che grado di approfondimento.
Dal punto di vista dell'utente, PerfectFit06 si presenta inizialmente come un qualsiasi sito web, che egli comincerà a esplorare nel modo classico.
Durante la navigazione l'utente farà, per forza di cose, delle scelte esplicitate nei link che gli interessano e che andrà a cliccare, PerfectFit06 utilizzerà questa fase inevitabile della navigazione per "studiarne" il comportamento e modellarlo.

Dopo un numero basso di passi di navigazione, PerfectFit06, avrà raggiunto una sufficiente consistenza del suo modello dell'utente e comincerà a sfruttare questa sua rappresentazione per "intuire" gli interessi dell'utente e cominciare a proporgli in maniera automatica i contenuti supposti significativi e a nascondergli, lasciandoli comunque accessibili, i contenuti ritenuti meno rilevanti.
E' opportuno sottolineare che PerfectFit06 non chiede mai esplicitamente gusti e preferenze all'utente, ma li ricava studiandone il comportamento durante la navigazione.
Dal punto di vista di PerfectFit06, l'utente interagisce con l'ambiente web esprimendo determinate preferenze rispetto a certi contenuti piuttosto che altri.
Il sistema vede l'intero insieme di contenuti suddiviso in nuclei informativi atomici definiti "risorse".
Ogni pagina web viene definita come un insieme finito di risorse.
Ad ogni risorsa sarà associato un relativo modello risorsa la cui struttura va specificata in base al contesto semantico dell'intero insieme informativo.
Associato ad ogni utente attivo sul sistema ci sarà un modello utente di cui ogni attributo rappresenterà una classe "ideale" di utente, mentre il valore specifico indicherà il grado di appartenenza dell'utente in questione alla classe "ideale".
Il modello utente, di un utente reale, sarà dunque una serie di pesi che specificano in base a classi ideali, definite classi "pure", il suo profilo comportamentale.
La struttura del modello utente non ha motivo di essere legata al contesto di applicazione, si può dunque scegliere un modello assolutamente generico.
Ogni volta che l'utente accede a una risorsa, chiedendo al sistema di visualizzarla, sta esplicitando il suo interesse riguardo quella determinata risorsa.
Associate a ogni risorsa ci sono una serie di regole, dette regole di estensione, che indicano come variare ogni campo del modello utente in base al comportamento espresso dalla scelta della risorsa in questione.
Il modello utente in un dato istante sarà il risultato delle regole di estensione utilizzate ad ogni selezione di risorsa effettuata dall'utente.
Ogni volta che l'utente chiede al sistema una nuova pagina web, PerfectFit06 utilizza il modello utente attuale per creare una risorsa ideale.
La risorsa ideale è creata tramite delle regole di associazione che permettono di trasformare il modello utente, ipoteticamente acontestuale, in un modello risorsa, legato al contesto.

Il modello così calcolato rappresenterà la risorsa ideale per l'utente rispetto al suo modello attivo in quel momento.
Sulla scorta del modello risorsa ideale il sistema andrà a calcolare la distanza delle risorse reali, contenute nella pagina web richiesta, rispetto a quella ideale.
Le risorse che avranno una distanza ritenuta sufficientemente bassa saranno visualizzate direttamente, poiché interessanti, mentre le altre verranno occultate, rimanendo comunque accessibili.
Bisogna sottolineare che gli unici legami tra il modello utente e le risorse sono le regole di estensione e di associazione.
Tramite questa doppia applicazione di regole è possibile gestire un interessante livello di astrazione tra il modello utente, generico, e le risorse, specifiche.
PerfectFit06 ha interessanti peculiarità anche dal punto di vista architetturale.
La caratteristica sicuramente più interessante strutturalmente è la capacità del sistema di generare, all'occorrenza, anche dei suoi moduli e utilizzarli nell'immediato, estendendo dunque la sua adattività anche a livello strutturale.
PerfectFit06 è formato da una sola pagina web fisica che, dinamicamente, costruisce i suoi contenuti a seconda dello stato di navigazione dell'utente.
La struttura del sito web è realizzata mediante un albero della conoscenza, opportunamente memorizzata in una struttura dati che rappresenta un generico albero.
Ogni nodo del grafo della conoscenza usato è una risorsa.
Il sistema si adatta dinamicamente a questo albero durante la navigazione gestendo i nodi-risorsa in modo gerarchico.
Praticamente, basta cambiare l'albero rappresentato all'interno della struttura dati e il sistema si adatterà dinamicamente alla nuova struttura senza modificare una sola riga di codice del sistema.
Ad ogni risorsa sono associati dei modelli risorsa e delle regole di estensione, se un modello risorsa o una regola di associazione viene cambiata bisogna ricreare l'insieme delle regole di estensione.
PerfectFit06 prevede, già nella versione attuale, un modulo capace di generare automaticamente e su richiesta tutto l'insieme di regole di estensione basandosi sulle regole di associazione e i modelli risorsa.
L'aspetto del sistema più delicato è sicuramente la definizione delle classi pure, ovvero degli attributi del modello utente.

E’ estremamente probabile trovarsi davanti alla necessità di aggiungere nuove classi pure o modificarne di preesistenti.
Proprio in virtù della generalità del modello utente, a cui prima si è fatto riferimento, il sistema deve avere la capacità di gestire un modello estremamente flessibile.
La struttura del modello utente è memorizzata all’interno di una struttura dati in un database.
Nel momento in cui si varia questa struttura dati il sistema è capace di ricreare tutti i suoi moduli che fanno uso del modello utente e utilizzarli all’interno della stessa sessione.
In questo modo basta cambiare il contenuto, per giunta banale, di una tabella, e il sistema è capace di gestire un modello utente nuovo senza necessità di modifiche umane al suo codice.
Poiché, come si può intuire, l’architettura del sistema è piuttosto complessa, in questa breve introduzione non si è entrati nel merito tecnico.
Si può, però, già farsi una idea di come il sistema, a dispetto della sua articolata architettura, si ponga in maniera del tutto trasparente rispetto al suo utilizzo da parte dell’utente, permettendogli di avere facile accesso alle risorse che più gli interessano ed evitandogli un potenziale sovraccarico cognitivo.
Da sottolineare che le risorse ritenute non interessanti per l’utente rimangono comunque accessibili.
Questa proprietà è estremamente importante per permettere all’utente di sopperire ad un eventuale errore del sistema consentendo a quest’ultimo di recuperare.
La capacità di rimediare ad una modellizzazione iniziale dell’utente sbagliata oppure far fronte a un suo eventuale cambio di interesse durante la navigazione è una proprietà molto importante per i sistemi come PerfectFit06.
Per rendere il sistema più veloce nella modellizzazione dell’utente e consentirgli un facile recupero di errori di modellizzazione o cambio piano dell’utente, è stato introdotto nel sistema il concetto di modello utente normalizzato.
Il sistema è stato implementato utilizzando i linguaggi di scripting Javascitpt/CSS e PHP, come database è stato utilizzato MySQL.
La versione di PerfectFit06 attuale implementa tutte le tecniche presentate nell’introduzione e nella tesi.

La tesi è organizzata nel seguente modo:

- **Primo capitolo:** in questo capitolo vengono presentati gli aspetti teorici alla base dell'architettura di PerfectFit06. Si proporrà una introduzione al concetto di agente e si effettuerà una panoramica sui sistemi di personalizzazione dei contenuti e le relative tecniche, con particolare enfasi sulla modellizzazione dell'utente e l'adattatività.

- **Secondo capitolo:** si presenterà in dettaglio l'architettura di PerfectFit06 mostrandone tutte le caratteristiche salienti. Verranno presentati tutti i concetti innovativi introdotti, sia dal punto di vista teorico che applicativo, dando ampio spazio al funzionamento logico delle varie componenti. Verranno definite tutte le metriche usate nonché verranno descritti i vari metodi di modellazione utilizzati.

- **Terzo capitolo:** verrà presentata una sperimentazione di PerfectFit06, mostrando delle interazioni col sistema da parte dell'utente, ed esibendo il funzionamento run-time dei meccanismi sottostanti presentati nel secondo capitolo. Verranno mostrati e commentati alcuni test d'uso realizzati sul sistema attualmente implementato. Saranno presentate alcune delle possibili future evoluzioni del sistema.

CAPITOLO I

SISTEMI DI PERSONALIZZAZIONE E AGENTI WEB

Author: Vincenzo Scognamiglio

Version: 2.6 – printed: 23/06/2006

Keywords: agenti web, sistemi adattativi, sistemi di personalizzazione, sistemi di raccomandazione, modello utente.

Capitolo I

Sistemi di personalizzazione e Agenti Web

1.1 Introduzione

Con l'incessante incremento di interesse da parte degli utenti e degli esperti di settore verso Internet e il World Wide Web è cominciata l'ascesa al successo dei sistemi di personalizzazione dei contenuti Web.
Un sistema di personalizzazione dei contenuti Web è un sistema software in grado di selezionare, preparare e inviare in modo personalizzato dei contenuti ad un determinato utente, tenendo conto delle sue specifiche esigenze.[Tasso-1]
Il campo degli agenti intelligenti, separatamente e in parallelo al successo dei servizi Web, si è concentrato sull'offerta di un supporto all'incapsulamento concettuale dei comportamenti autonomi e focalizzati a uno scopo (goal-driven) per un'ampia varietà di applicazioni e soluzioni software.[Greenwood-1]
Un agente è un'entità computazionale, quale un software o un robot, con capacità percettive e attuative rispetto al suo ambiente ed è potenzialmente autonomo nei suoi comportamenti[Weiss-1].
Come si evince, la definizione di agente è molto generica e quindi facilmente adeguabile a qualsiasi contesto. Infatti, questo concetto, è diventato particolarmente importante sia nell'ambito dell'Intelligenza Artificiale (IA) sia in tutta la computer science[Rossi-1].
Un agente Web è dunque un agente che ha come suo ambiente il World Wide Web.

Poiché sia gli agenti che i sistemi di personalizzazione si basano sulla centralità dell'utente nel sistema, il loro incontro è un evento del tutto naturale se si pone come scopo principale dell'agente Web proprio la personalizzazione dei contenuti.
In questo primo capitolo verrà fatta una panoramica, non esaustiva, dei principali aspetti riguardanti gli agenti e in particolare i sistemi di personalizzazione dei contenuti.
Ci si soffermerà maggiormente sui concetti utilizzati e implementati nell'agente Web del sistema informativo sugli scavi di Ercolano, ovvero PerfectFit06, presentato in dettaglio nel secondo e terzo capitolo.

1.2 Definizione di agente

Abbiamo definito un agente come "una entità computazionale, quale un software o un robot, con capacità percettive e attuative rispetto al suo ambiente e potenzialmente autonomo nei suoi comportamenti"[Weiss-1].
In figura 1.2.1 è mostrata una semplificazione dell'attività di un agente.
In questa definizione già si ritrovano le parole chiave per la caratterizzazione degli agenti ma è bene approfondire la questione.

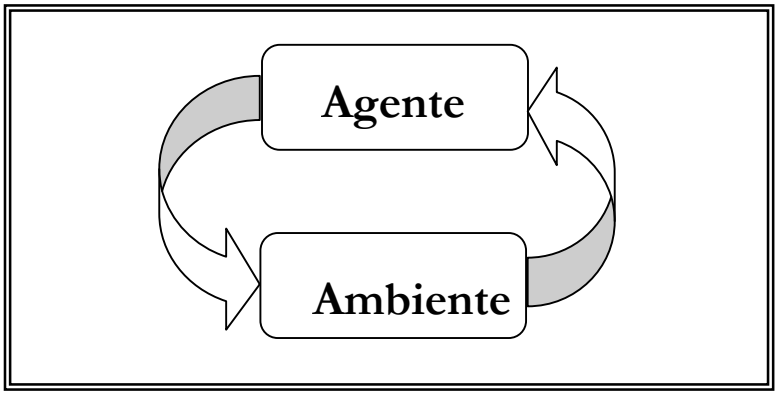


Fig.1.2.1 – Un agente e il suo ambiente. L'agente riceve, tramite dei sensori, input dal suo ambiente e produce come output delle azioni.

Nel senso più generale, il termine Agente è usato per indicare un sistema che gode di alcune particolari proprietà:

- **Autonomia:** un agente può operare "indipendentemente", senza l'intervento diretto di un essere umano o d'altro e inoltre possiede un certo controllo delle proprie azioni. [Weiss-1, Wooldridge-1]
- **Reattività:** essere reattivi significa saper rispondere ai cambiamenti dell'ambiente. La reattività può aversi sia come una risposta immediata ad uno stimolo, sia come una deliberazione sulle azioni da compiere a seguito di uno stimolo o di un cambiamento. [Weiss-1, Wooldridge-1]
- **Pro-attività:** un agente può non solo reagire agli stimoli che gli provengono dall'ambiente esterno, ma può avere un comportamento indirizzato al perseguimento di uno scopo (*Goal-directed behaviour*).[Weiss-1, Wooldridge-1]
- **Abilità sociale:** un agente può interagire (tramite negoziazione e cooperazione) con altri agenti mediante un sistema e/o un linguaggio di comunicazione. [Weiss-1, Wooldridge-1]

Nell'evoluzione dei sistemi ad agenti molte proprietà sono state aggiunte a queste, spesso in base al contesto specifico di applicazione, ma le caratteristiche sopra elencate rimangono comunque il punto di partenza imprescindibile per qualsiasi trattazione del tema.

1.2.1 Tipologie di agenti

Gli agenti posso essere classificati in base a numerose caratteristiche architetturali. Non avendo l'obbiettivo di fornire una trattazione esaustiva riguardo le tipologie di agenti, si andranno a presentare le architetture ritenute più rilevanti:

- **Agenti su base logica:** la realizzazione di decisioni da parte dell'agente è effettuata attraverso la deduzione logica applicata a sistemi di regole. [Weiss-1]
- **Agenti reattivi:** l'agente prende autonomamente le decisioni attraverso una mappatura diretta della situazione attuale in una corrispondente azione.[Weiss-1]
- **Agenti BDI (belief-desire-intention):** le decisioni sono prese in base alla manipolazione di strutture dati rappresentanti le opinioni, i desideri e le intenzioni dell'utente.[Weiss-1]
- **Architettura stratificata:** le decisioni dell'agente sono prese attraverso diversi strati software, ognuno dei quali realizza ragionamenti, più o meno espliciti rispetto all'ambiente, a differenti livelli di astrazione.[Weiss-1]

In ognuno dei casi presentati si fa comunque riferimento a una vista astratta dell'agente senza entrare nel merito di quale effettivamente sia la sua struttura interna.

1.3 I sistemi di personalizzazione Web

Un sistema di personalizzazione Web è un sistema software in grado di selezionare, preparare e inviare, in modo personalizzato, dei contenuti ad un determinato utente, tenendo conto delle sue specifiche esigenze[Tasso-1].
Personalizzare significa capire cosa offrire all'utente, farlo nel modo migliore e nei tempi più adatti.
Nella figura 1.3.1 è mostrato uno schema di organizzazione generale di un sistema di personalizzazione.

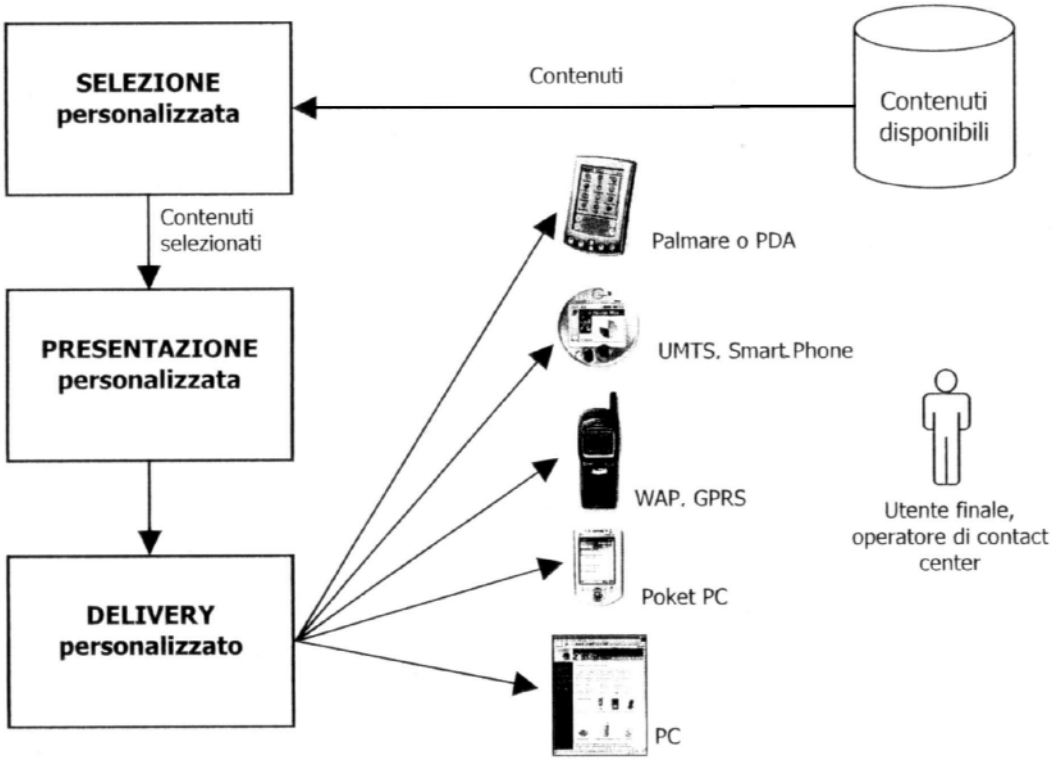


fig.1.3.1 – organizzazione generale di un sistema di personalizzazione

I sistemi di personalizzazione generalmente hanno come obiettivo principale una selezione dei contenuti fatta in base a opportuni filtri, in secondo luogo si interessano di adattare l'interfaccia individuando i gusti e le esigenze dell'utente.
La centralità dell'utente, rispetto a qualsiasi sistema di personalizzazione si faccia riferimento, porta alla necessità di rappresentarne i gusti e le preferenze attraverso un adeguato modello utente.

La personalizzazione dei contenuti può essere di due tipi:

- ***Esplicita*** o customizzazione, ossia quel tipo di personalizzazione nella quale i contenuti che l'utente desidera ricevere vengono indicati da lui esplicitamente al sistema; il modello utente viene creato con queste indicazioni e il sistema provvede a selezionare e fornire i contenuti in base allo schema impartito. [Tasso-1]

- ***Implicita*** o personalizzazione adattiva (adattativa), si riferisce al caso in cui il sistema non possiede informazioni esplicite sulle preferenze dell'utente ma è capace di costruire autonomamente un modello utente osservando e valutando il comportamento di quest'ultimo durante la navigazione del sito. [Tasso-1]

La personalizzazione implicita muove un forte interesse nelle comunità di ricerca e sarà trattata, all'interno di questo capitolo, con maggiore dettaglio rispetto all'esplicita anche perché è la forma di personalizzazione implementata nel sistema che si andrà a presentare.

1.4 I sistemi di raccomandazione

I sistemi di personalizzazione sono particolarmente utili quando la quantità di dati è enorme e l'utente tende quindi a smarrirsi.
Una classe particolare di sistemi di personalizzazione sono i sistemi di raccomandazione.
Un sistema di raccomandazione consiglia informazioni apparentemente utili all'utente o suggerisce strategie che esso può applicare per giungere al suo presunto scopo.
Una raccomandazione può essere generata in base a un dato evento, come un errore, o basarsi sulla supervisione di comportamenti dell'utente[usability-1].
Esempi abbastanza semplici di sistemi di raccomandazione sono i motori di ricerca come Google [Google-1] che suggeriscono una nuova stringa di ricerca se quella inserita dall'utente non ha ottenuto risultati (figura 1.4.1).



fig.1.4.1 – suggerimento automatico della stringa di ricerca di Google

I sistemi di raccomandazione e i sistemi di personalizzazione sono dunque intimamente legati[Terveen-1].
Ciò si evince chiaramente nei casi, tutt'altro che rari in Internet, in cui l'utente si trova a navigare tra un numero enorme di informazioni ed è improponibile esaminare tutto. In questi casi, dunque, la personalizzazione dei contenuti si lega indissolubilmente con il filtraggio - raccomandazione degli stessi.
Un famoso sistema di raccomandazione per i libri è Amazon [Amazon-1] mentre tra i numerosi sistemi disponibili per il cinema c'è MovieFinder[Moviefinder-1].

1.5 Il modello utente

L'elemento immancabile per qualsiasi sistema di personalizzazione è il modello utente. Un modello utente è una struttura dati, permanente e continuamente aggiornata dal sistema, usata per guidare l'utente attraverso le sole informazioni che lo interessano [DeBra-3].
La definizione di un modello utente è un processo piuttosto complesso soprattutto per la necessità di classificare caratteristiche "umane" tramite delle proprietà. Tali proprietà non potranno mai essere un rappresentazione esaustiva e saranno sempre soggette in qualche modo al contesto e all'interpretazione del progettista.
E' inevitabile che molte problematiche del modello utente abbiano un loro corrispettivo nell'AI (Artificial Intelligence) e che questi due settori trovino sbocchi comuni in più casi.
E' ovvio però che l'AI si muove su più ampio raggio e che i sistemi di personalizzazione Web a volte siano alquanto semplicistici anche per questioni tecnologiche.
Gli aspetti rappresentabili di una persona/utente sono praticamente infiniti, semplicemente non sarà mai possibile descrivere *completamente* l'utente, ciò però non vuol dire che non lo si riesca a rappresentare *sufficientemente*, almeno per alcune specifiche classi di interesse.

1.5.1 Le caratteristiche dell'utente

In un loro famoso film, Totò e Peppino dissero: *"Per andare dove dobbiamo andare, per dove dobbiamo andare?"* [Totò-1].
La simpatica frase, per giunta detta a un "agente" di polizia, è forse quella che più si avvicina all'atteggiamento usuale di un utente nello spazio Web.
Infatti, molto spesso, l'utente ha idee molto vaghe su quali siano le cose che realmente lo interessano e riporta la sua indecisione nelle ricerche Internet sperando che il sistema, o l'agente, lo capisca a prescindere dalla sua manifesta incertezza.
Risultato di questa osservazione è che la navigazione Web viene spesso effettuata per tentativi o comunque senza un piano di navigazione.
Il concetto di cercare tutto "*perché tanto qualcosa sempre si trova*", è molto diffuso in Internet e nel World Wide Web. Questo atteggiamento ha però come conseguenza un

grosso dispendio di risorse temporali e mentali che non necessariamente porta a dei risultati.
Se da un lato il grande interesse verso Internet è legato alla immensa mole di dati che contiene, dall'altro, la sua debolezza, è proprio l'intrattabilità di questa grosse mole di dati nata ed evoluta senza regole.
Poiché i sistemi Web sono diventati sempre più popolari come strumento di guida per l'utente che accede alle informazioni[De Bra-3], diventa essenziale progettare un sistema capace di assistere e guidare l'utente deducendone le preferenze in maniera quanto più trasparente possibile.
Vista la centralità dell'utente nel sistema, una sua modellizzazione efficace e formale è un punto di partenza imprescindibile ma senza dubbio di difficile trattazione.
Un problema non trascurabile sta nel fatto che, qualsiasi modello si decida di adottare, deve essere rappresentabile finitamente.
Nel corso di anni di ricerca si sono comunque definiti i principali aspetti dell'utente suscettibili all' adattamento:

- ***Conoscenza dell'utente:*** quali e quanto profonde sono le conoscenze dell'utente rispetto ai contenuti Web del sistema. Questa caratteristica è considerata spesso la più importante e senza dubbio la più delicata poiché porta come immediata conseguenza la capacità di riconoscere ciò che può essere ritenuto interessante o meno per l'utente[Brusilovsky-1].
- ***Scopo dell'utente:*** questa caratteristica è strettamente legata al contesto di applicazione molto più di quanto lo sia individualmente con l'utente[Brusilovsky-1].Questo aspetto è importante anche per gli agenti (par. 1.2).
- ***Background ed esperienza:*** queste caratteristiche si riferiscono alla conoscenza e all'esperienza dell'utente rispetto all'ambiente in cui si muove il sistema[Brusilovsky-1]. In realtà la semplicità e l'immediatezza dell'ambiente Web permette di tralasciare la modellizzazione di questi aspetti[Brusilovsky-1].
- ***Preferenze:*** questa caratteristica indica le preferenze di un utente di un nodo rispetto ad un altro. Queste preferenze possono essere assolute o relative, cioè, dipendono dal nodo corrente, dagli scopi dell'utente e dal contesto generale corrente[Brusilovsky-1].

- ***Interessi dell'utente:*** questa caratteristica rappresenta gli interessi dell'utente ed è a più ampio raggio rispetto allo scopo immediato in quanto tenta di inquadrare gli obiettivi dell'utente a lungo termine[Brusilovsky-1].
- ***Tratti individuali:*** in realtà si tratta di un gruppo di caratteristiche che, insieme, rappresentano l'utente come individuo. Sono un esempio: le caratteristiche comportamentali (estroverso/introverso), i fattori cognitivi, gli stili di apprendimento etc[Brusilovsky-1].

Resta il fatto che la decisione di quali importanti strutture o processi vadano inseriti nei sistemi di modellizzazione dell'utente è, per la maggior parte, frutto dell'intuito e dell'esperienza del progettista del sistema[Kobsa-1].
Nella figura 1.5.1 è mostrato un ciclo classico di adattatività basata su modello utente.

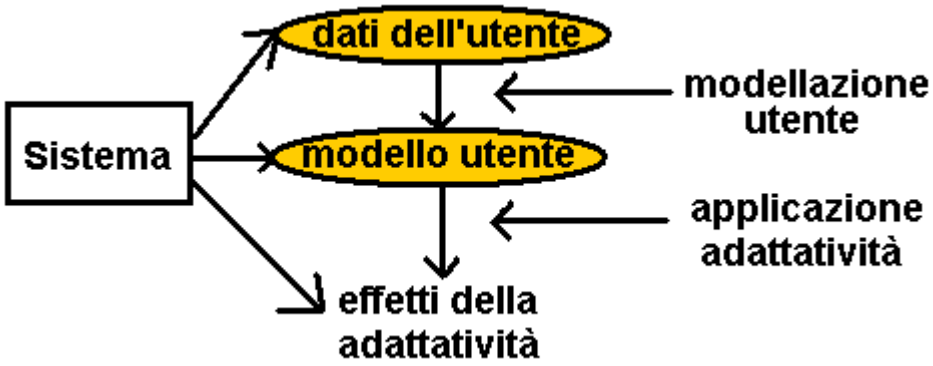


fig.1.5.1 – Ciclo classico di adattatività basata sulla modellizzazione dell'utente

1.5.2 Tipi di modelli utente

Al fine di avere una conoscenza sufficiente alla comprensione delle possibili caratterizzazioni dello UM (user modeling) verranno presentate ora alcune 'dimensioni'[Rich-1] utili a classificare i vari sistemi interattivi relativamente alla loro capacità di rappresentazione del profilo utente.

Modello Stereotipato – Modello Individuale: Un modello di caratteristiche dell'utente è detto **stereotipato** (stereotype model) se ha lo scopo di rappresentare le caratteristiche comuni di una intera popolazione di utenti, focalizzandosi sulle caratteristiche più frequenti e condivise dai membri della popolazione, trascurando invece le peculiarità individuali. [Tasso-1]

Un modello è detto **individuale** (overlay model) se rappresenta caratteristiche di ogni singolo individuo della popolazione, incluse sia le caratteristiche comuni agli altri che quelle peculiari di ciascuno. [Tasso-1]

Modello dato – modello utente inferito: Un modello di caratteristiche dell'utente si definisce **dato** se è fornito da un attore esterno al sistema come ad esempio quando l'utente stesso esplicita le sue preferenze. Un modello è invece detto **inferito** se viene costruito automaticamente dal sistema durante il suo funzionamento attraverso un eventuale modulo adibito a questo scopo. [Tasso-1]

Modello implicito – modello esplicito: un modello è **implicito** quando è non riconoscibile all'interno del codice, ovvero il sistema non ha una rappresentazione distinta di esso in una struttura dati adibita allo scopo o in una parte specifica del codice. Un modello utente è detto **esplicito** se è rappresentato direttamente all'interno del sistema, in una componente separata del codice, accessibile durante l'interazione con l'utente e contenente parametri utilizzati per configurare specifici aspetti dell'interazione. [Tasso-1]

Modello statico – modello dinamico: un modello utente è detto **statico** se rimane invariato durante tutto il suo ciclo di vita, **dinamico** se è esteso, raffinato o modificato durante il funzionamento del sistema al fine di adeguarsi con tempestività alle esigenze sull'utente. [Tasso-1]

Modello a breve termine – a lungo termine: un modello si dice a **breve termine** se il suo ciclo di vita non supera quello della sessione. Un modello è detto a **lungo termine** se viene utilizzato in più sessioni successive. Nel primo caso si usa anche dire che è una **modellizzazione effimera**, nel secondo una **modellizzazione persistente**. [Tasso-1]

Le 'dimensioni' introdotte non necessariamente sono esclusive e si possono dunque presentare anche in concomitanza.

1.5.3 Rappresentazione del modello utente

Un modello utente consiste in una entità con nome per la quale sono memorizzate un insieme di coppie attributo-valore. Per ogni entità ci possono essere diversi attributi [De Bra-2].

Le coppie attributo-valore possono essere tra loro:

- ***Dipendenti***: il valore di una coppia è funzione di un'altra o comunque ci sono dei vincoli di comportamento tra loro, ad esempio la somma di due valori di attributo non deve superare un certo valore.
- ***Indipendenti:*** non c'è alcun tipo di vincolo che lega una coppia a qualsiasi altra.

Il valore assunto da un attributo può essere:

- ***Binario:*** l'attributo ha solo due stati (ad es. buono/cattivo)
- ***Qualitativo:*** esprime il valore tramite un aggettivo (es. buono/ottimo/pessimo/poco/molto)
- ***Quantitativo:*** il valore dell'attributo è rappresentato da un numero (ad es. percentuali)

1.5.4 Tipi di sistemi di personalizzazione in base al modello utente

Nel caso dei modelli individuali un sistema di personalizzazione dei contenuti Web può essere di due tipi:

Adattabile: ossia un sistema con modello esplicito e dato, in particolare fornito dall'utente che quindi realizza una forma di personalizzazione esplicita.

Adattativo (o adattivo): ossia un modello esplicito inferito autonomamente dal sistema (fig. 1.5.2) che permette la realizzazione di un sistema di personalizzazione implicita anche detta personalizzazione adattiva[Tasso-1].

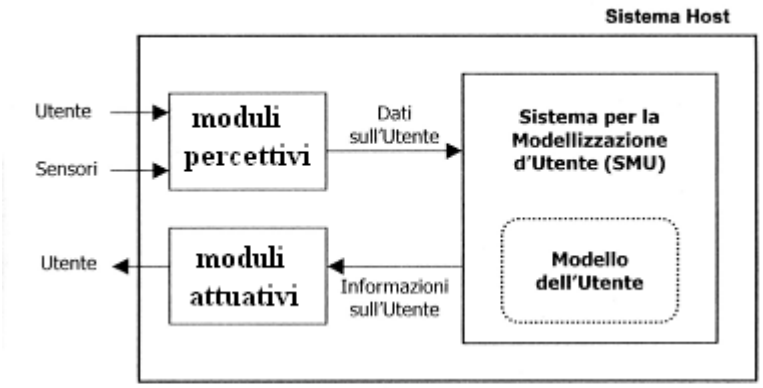


fig. 1.5.2 Organizzazione generale di un sistema basato su modellizzazione dell'utente adattativa con modello esplicito

1.6 Aspetti adattabili nel Web

Nel momento in cui si vuole realizzare un sistema di personalizzazione sono molti i quesiti a cui bisogna rispondere. Infatti, prima di passare al nodo cruciale della definizione del modello utente, bisogna fare una ponderata scelta delle caratteristiche dell'ambiente Web che si vogliono rendere personalizzabili.
La scelta non è per nulla semplice anche se gli aspetti che generalmente riscuotono maggior attenzione sono l'adattività di presentazione e quella di navigazione [Brusilovsky-1].
Segue una tassonomia delle tecnologie dei sistemi ipermediali adattativi:

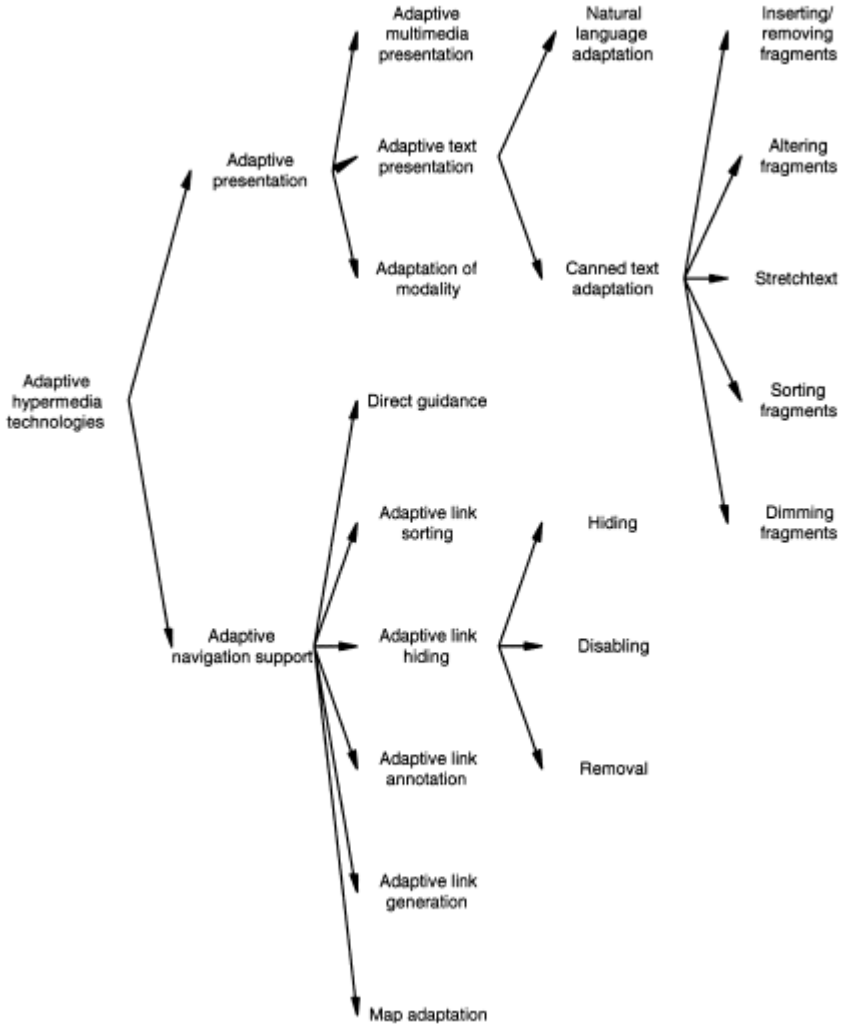


fig1.6.1. tassonomia delle tecnologie dei sistemi ipermediali adattivi[Brusilovsky-1]

L'idea più studiata dagli esperti, sulle varie tecniche di adattività di presentazione, è di adattare il contenuto delle pagine in base al particolare utente che vi sta accedendo[Brusilovsky-1].
Verrà realizzata una panoramica dei termini introdotti con la tassonomia di figura 1.6.1.

L'**adattività di presentazione o adaptive presentation** si divide principalmente in:

- ***Adattività testuale o adaptive text presentation:*** il contenuto della pagina viene riformulato nella maniera più adatta all'utente che vi sta accedendo (ad esempio in un linguaggio naturale se è un non vedente).[Brusilovsky-1]
- ***Adattività multimediale o adaptive multimedia presentation:*** il contenuto della pagina viene presentato in maniera diversa in base al tipo di multimedia tramite cui l'utente effettua l'accesso. [Brusilovsky-1]

L'**adattività di navigazione o adaptive navigation support**, invece, si dirama nei seguenti sottorami:

- ***Guida diretta o direct guidance:*** è la tecnica più semplice di adattività di navigazione. Per il sistema si tratta di scegliere autonomamente il nodo considerato migliore per l'utente. [Brusilovsky-1]
- ***Ordinamento dei link o adaptive link sorting:*** realizza un ordinamento dei link della pagina corrente in base a criteri ricavabilibili dal modello utente.
- ***Offuscamento dei link o adaptive link hiding:*** sugli stessi principi dell'ordinamento si sceglie di rendere dei link ritenuti poco interessanti invisibili, disabilitati o sconsigliati (eventualmente cambiandone il colore). [Brusilovsky-1]
- ***Commento ai link o adaptive link annotation:*** ad ogni link viene associato un commento, in una qualsiasi forma, che offre ulteriori informazioni all'utente. Sostanzialmente è quello che succede staticamente con i tool type text nella maggioranza delle interfacce grafiche oppure con il cambiamento di colore dei link nei browser dipende anche dal fatto che le relative pagine siano state visitate o meno . [Brusilovsky-1]
- ***Generazione di link o adaptive link generation:*** Vengono generati automaticamente solo i link ritenuti rilevanti, praticamente quello che succede in modo banale nei motori di ricerca. [Brusilovsky-1]

- ***Mappa o map adaptation:*** l'adattività di mappa in realtà consiste in varie tecniche di adattività di navigazione (offuscamento, guida diretta etc.) che cambiano sostanzialmente i percorsi che l'utente può intraprendere. [Brusilovsky-1]

1.7 Fasi del processo di modellizzazione

Modellizzare l'utente significa fondamentalmente costruire e gestire il modello utente. Il processo di modellizzazione può essere articolato in tre fasi distinte:

1. ***Acquisizione*** dei dati dai moduli di osservazione sfruttando l'interazione con l'utente e il contesto. Questa fase può essere realizzata in maniera implicita o esplicita a seconda che i dati dell'utente siano espressamene richiesti o vengano acquisiti in maniera trasparente. [Tasso-1]
2. ***Estensione*** dei contenuti del modello utente tramite attività inferenziali basate su specifiche conoscenze e su processi di ragionamento che emulano le attività di modellizzazione umana. [Tasso-1]
3. ***Mantenimento*** della consistenza del modello. La necessità di questa fase nasce dal fatto che è possibile ottenere molte informazioni sull'utente tramite successive attività di inferenza che però, in un dato momento, possono essere incoerenti tra loro.

L'identificazione delle tre fasi permette di definire un modello generale dell'architettura di un Sistema di Modellizzazione d'Utente.

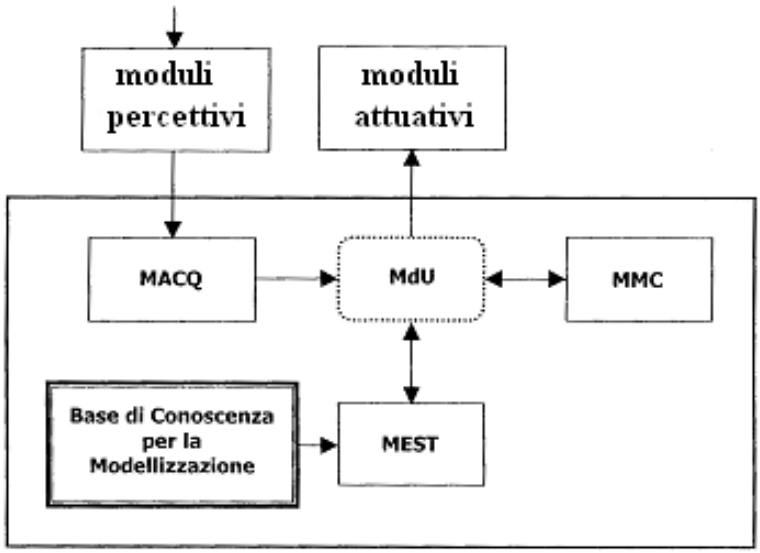


fig. 1.7.1 Architettura generale di un Sistema di Modellizzazione d'Utente

In particolare definendo:

- ***MdU:*** Modello dell'utente
- ***MACQ:*** Modulo di acquisizione
- ***MEST:*** modulo di estensione
- ***MMC:*** Modulo per il mantenimento della consistenza

[Tasso-1]

Si può riassumere questo aspetto del sistema come in figura 1.7.1.

1.8 Tecniche per la personalizzazione

Per personalizzare i contenuti di un sito Web si devono risolvere fondamentalmente tre problemi: identificare l'utente, gestirne il modello e sfruttare il modello per la selezione dei contenuti[Tasso-1].
La gestione e l'utilizzo del modello utente sono scelte riguardanti il particolare sistema mentre per quanto riguarda i metodi di identificazione si può effettuare una prima individuazione.

1.8.1 I metodi per l'identificazione e il tracking dell'utente

Verranno presentati brevemente i metodi principali per l'identificazione dell'utente:

- ***Cookie:*** file testuali di piccole dimensioni residenti sul client in cui il server può salvare informazioni. Il server può scrivere solo in questi file "speciali" che hanno dimensioni massime prestabilite per ragioni di sicurezza.
- ***User-id e password:*** il metodo più utilizzato in Internet anche per i servizi come la posta elettronica, consiste in un nome univoco nel sistema, assegnato all'utente, in coppia con una chiave scelta generalmente da quest'ultimo o dal sistema. Il principio consiste nel far corrispondere ad ogni singola coppia user id - password un unico utente.
- ***Agenti software:*** tramite uno specifico programma è possibile ottenere un accesso personalizzato e il programma stesso si occuperà del metodo di identificazione dell'utente (come ad esempio MSN della Microsoft per la messaggistica istantanea).
- ***Indirizzo IP:*** si sfrutta l'indirizzo IP associato all'utente per identificarlo univocamente, in realtà il metodo non funziona sempre poiché alcuni firewall o altri programmi speciali possono nascondere l'indirizzo IP (come Get Anonymus) o, ancora, l'utilizzo di un proxy può invalidare l'ipotetico rapporto uno-a-uno tra IP e utente .

L'identificazione dell'utente è il primo passo per realizzarne il tracking, ovvero il tracciamento delle attività all'interno del sistema.
Ci sono numerosi problemi pratici che si pongono a seconda del metodo scelto.

Una trattazione specifica dei criteri e delle soluzioni tecniche implementate sarà discussa nei capitoli successivi

1.9 Il filtraggio delle informazioni

Un sistema di filtraggio delle informazioni si interpone tra una *sorgente di informazioni* e un utente ed è:

1. Progettato per trattare documenti semistrutturati. [Belkin-1]
2. In grado di costruire e aggiornare un profilo dell'utente. [Belkin-1]
3. In grado di selezionare, in base al contenuto del profilo, le informazioni potenzialmente interessanti tra quelle prodotte dalla sorgente. [Belkin-1]

Un'analisi della definizione data fa apparire chiaro come il filtraggio delle informazioni sia un caso particolare di Information Retrievial (IR).

1.9.1 Il filtraggio collaborativo e il filtraggio cognitivo.

Il filtraggio collaborativo[Malone-1] è un metodo per filtrare i contenuti che si basa sull'istinto naturale dell'uomo di avvalersi delle esperienze, dei consigli e delle preferenze altrui[Tasso-1].
In sintesi, il filtraggio collaborativo, si avvale di algoritmi per calcolare le preferenze dell'utente, basandosi su quelle precedentemente espresse dagli altri utenti, con tecniche più o meno raffinate a seconda dei casi.

Il filtraggio cognitivo[Malone-1] è un metodo per filtrare informazioni basandosi sull'analisi della loro semantica[Tasso-1].
In entrambi i metodi di filtraggio lo scopo è evidentemente quello di individuare le sole informazioni importanti per l'utente.
Non ci si soffermerà sul filtraggio collaborativo poiché non è stato implementato nel sistema che verrà presentato.
Il termine "filtraggio cognitivo" è descritto molto bene dal suo equivalente in inglese: content-based filtering, ovvero filtraggio basato sul contenuto.
Il filtraggio cognitivo è sicuramente più complesso del collaborativo, le diverse modalità con cui vengono realizzati gli algoritmi in grado di analizzare il significato dei

documenti semi-strutturati caratterizzano le diverse tecniche di filtraggio cognitivo[Tasso-1].

Una organizzazione comune di tutti i sistemi di filtraggio cognitivo è illustrata in figura 1.9.1.

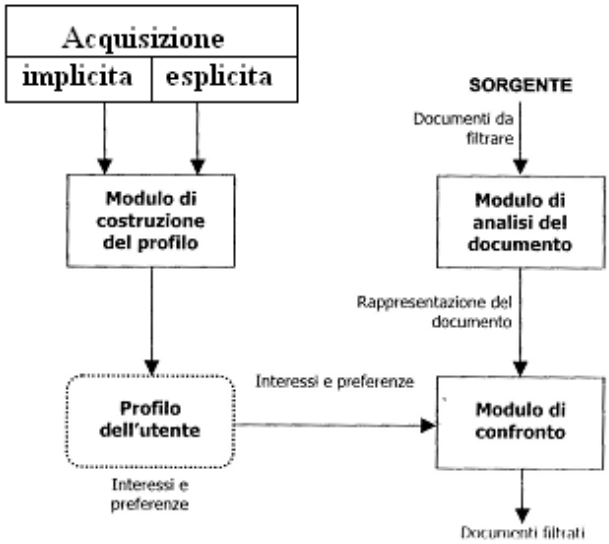


fig. 1.9.1 organizzazione comune di tutti i sistemici filtraggio cognitivo

In esso si distinguono:

- Modulo per la costruzione (e l'aggiornamento) del profilo utente.
- Modulo per la rappresentazione del contenuto concettuale dei documenti provenienti dalla sorgente di informazioni.
- Modulo per il confronto del profilo con la rappresentazione dei documenti.

Per ogni sistema si scelgono le tecniche e le tecnologie che si ritengono più adeguate per i vari moduli ma il nodo cruciale resta comunque la rappresentazione dei significati dei contenuti da filtrare.

1.9.2 Tecniche per la rappresentazione dei significati

Ci sono numerose tecniche per la rappresentazione dei significati, tra cui:

- ***Parole chiave:*** l'argomento di interesse e il contenuto del documento sono rappresentate mediante un insieme di termini (keywords) [Folz-1]. E' il metodo classico utilizzato anche per i motori di ricerca basati su query. I problemi principali che riguardano questa tecnica sono legati alla possibile polisemia e alla sinonimia delle parole chiave.
- ***Vettori pesati su parole:*** consiste nell'associare a delle parole chiave un indice di importanza quantitativo o qualitativo oppure un valore di frequenza come ad esempio il numero di volte in cui il termine compare in un testo.
- ***Coppie attributo-valore:*** si utilizzano strutture simili a record che contengono campi utilizzati per rappresentare attributi ben precisi che assumono valori in range predefiniti[Shoshana-1].
- ***Reti Bayesiane:*** una rete Bayesiana è un modello probabilistico per la rappresentazione della conoscenza[Pearl-1]. Generalmente un rete Bayesiana è costituita da nodi, che rappresentano variabili proposizionali, e archi che collegano i nodi.

  Agli archi sono associati matrici di probabilità che specificano la probabilità di un evento in base all'avvento di un altro[Tasso-1].
- ***Reti semantiche:*** il metodo consiste nell'utilizzare il formalismo di rappresentazione della conoscenza delle reti semantiche[Tasso-2] e si adatta bene a rappresentare conoscenze concettuali e, in particolare, concetti e connesse relazioni[Tasso-1].

1.9.3 Semantic Web

A questo punto è bene aprire una breve parentesi chiarificatrice sulle ontologie e il Semantic Web in modo da non creare confusione in chi già conosce questi argomenti.
Le ontologie sono delle strutture dati associate ai contenuti Web, generalmente molto estese, atte a descriverne la semantica.
Le ontologie sono generalmente scritte in linguaggi (come l'RDF) nati come estensioni dell'XML.
Il Semantic Web nasce dal sogno di TimBerners-Lee, descritto in un articolo edito su Scientific American nel Maggio del 2001[Lee-1], in cui si parla di un Web dove i sistemi autonomi sono in grado di reperire le informazioni di interesse per l'utente in maniera automatica.
Il primo passo per la realizzazione del sogno di Tim Berners-Lee è rappresentato dal Semantic Web, un Web in cui a ogni contenuto è associata almeno un'ontologia.
Le similitudini tra SemanticWeb e i sistemi di personalizzazione sono dunque fortissime ma la differenza sostanziale sta nel contesto in cui si muovono.
Mentre il Semantic Web ha come ambito l'intero Web, un sistema di personalizzazione si muove, invece, solo nel contesto esplicitamente definito dal sistemista.
Nulla toglie, però, che un sistema di personalizzazione possa basarsi su ontologie descritte come richiesto per il Semantic Web e che, quest'ultimo, possa incominciare la sua vera fase evolutiva proprio da una standardizzazione dei sistemi di personalizzazione.

1.10 I sistemi basati su regole

I sistemi di personalizzazione basati su regole utilizzano regole di produzione e ragionamento deduttivo[Tasso-1].
Le regole di produzione sono un formalismo utilizzato in intelligenza artificiale per rappresentare la conoscenza[Newell-1].
Una regola di **produzione**, d'ora in poi semplicemente regola, è strutturata nel seguente modo

IF <condizione> **THEN** <azione>

Questo tipo di ragionamento è chiamato ragionamento deduttivo[Tasso-1].
Si definisca ora come transazione un sottoinsieme T dell'insieme C definito come l'unione di tutti i contenuti.
Presi due insiemi disgiunti X e Y, una regola **associativa** è del tipo X → Y e indica una correlazione tra X e Y cioè, se dei contenuti dell'insieme X appartengono a una transazione T, allora è molto alta la probabilità che anche contenuti dell'insieme Y appartengano a T[Tasso-1].
Quindi le regole associative rappresentano una significativa correlazione tra due o più contenuti del sito[Tasso-1].
Le regole associative sono utilissime nei sistemi di personalizzazione dei contenuti, infatti, la possibilità di relazionare in tal modo i contenuti, permette una agevole selezione di informazioni in
base a quelle verso cui ha già mostrato interesse l'utente nella sua interazione.
Un sistema di raccomandazione basato su regole disporrà di una serie di regole (base di regole) che ne influenzeranno il comportamento e che possono essere fornite manualmente o acquisite in maniera automatica[Tasso-1].
Nel caso di una acquisizione manuale, per la definizione delle regole, c'è bisogno almeno di un esperto del settore di cui trattano i contenuti oltre, ovviamente, di qualcuno in grado di elaborare le informazioni ottenute dall'esperto per trasformarle in regole.
Nel caso di una acquisizione automatica delle regole ciò avverrà tramite un modulo osservatore che studierà l'interazione dell'utente col sistema e ne estrarrà delle regole da sottoporre all'esperto o da inserire direttamente all'interno della base di regole.

1.11 Capacità di adattamento all'ambiente dell'utente

Bisogna sottolineare che quando si parla di adattare i contenuti non bisogna fermarsi al concetto di un utente che sta utilizzando il classico personal computer come mezzo di accesso al Web.
Si deve inserire nella discussione un altro fattore: l'ambiente dell'utente.
Le tecnologie Web, oramai, sono ampiamente diffuse e fruibili da dispositivi di tutti i tipi, basti pensare all'evoluzione dei telefonini e del Wap.
Adattarsi all'utente significa non solo riconoscerne le preferenze ma anche adeguarsi agli strumenti che sta sfruttando per l'accesso.
L'adeguamento all'ambiente dell'utente è un nuovo tipo di adattamento che sta prendendo corpo nei sistemi Web[Brusilovsky-1]. Ad esempio, sapere che l'utente sta utilizzando un telefonino potrebbe portare il sistema a scegliere di non visualizzare immagini che risulterebbero illeggibili su uno schermo piccolo. Esempi ancora più complessi possono riguardare il linguaggio naturale, le visite guidate "elettroniche" ai musei etc.
Senza dubbio l'adeguamento all'ambiente utente è uno sforzo molto significativo da parte del progettista, ma una buona architettura dovrebbe rendere minimo il numero di accorgimenti per il rinnovamento del sistema a nuove tecnologie Web-based.
Inoltre, numerosi studi sull'incrocio tra Mobile Web e Web adattivo, hanno dimostrato che questa ubiquità di elaborazione e i modelli utente possono beneficiare molto l'uno dell'altro. Da un lato, una capacità di adattarsi può significativamente migliorare l'usabilità della applicazioni mobili, dall'altro, l'ubiquità di elaborazione permette di estendere notevolmente i limiti entro cui effettuare l'adattamento[Ziegler-1].

CAPITOLO II

ARCHITETTURA DI PERFECTFIT06

Author: Vincenzo Scognamiglio

Version: 3.4 – printed: 23/06/2006

Keywords: PerfectFit06, grafo della conoscenza, modello risorsa, modello utente, classi pure, regole di associazione, regole di estensione, modelli ideali.

Capitolo II

Architettura di PerfectFit06

2.1 Evoluzione dei sistemi ipermediali di Ercolano

La creazione di un sistema multimediale per gli scavi archeologici di Ercolano è un progetto iniziato ormai da alcuni lustri e che è cresciuto in maniera progressiva rispetto alla tecnologia disponibile.
Già in passato sono stati realizzati, con la collaborazione del CNR, dei sistemi ipermediali efficienti che differivano tra loro per le tecniche e le tecnologia utilizzate.
Il primo progetto in assoluto è stato un ipertesto, scritto in Toolbook, grazie al quale si è iniziata una corposa raccolta di documenti e la relativa organizzazione, sintetizzata nel grafo descritto nel par. 2.5, e ancora oggi utilizzata[Ercolano-1] .
Il secondo progetto è stato un agente web intelligente, scritto in linguaggio Java [Java-1], capace di modellare l'utente e di personalizzare alcuni aspetti della navigazione[Rossi-1].
Il terzo progetto, quello presentato in questo documento col nome di **PerfectFit06**, rispetto agli altri due ha il vantaggio di nascere a diversi anni di distanza e di poter quindi fare affidamento su basi teoriche e pratiche molto più evolute, nonché sull'essenziale esperienza specifica maturata grazie ai due sistemi ipermediali antecedenti.
Il sistema presentato mantiene fermi gli obiettivi dei suoi predecessori spingendosi però molto più avanti con le tecniche di modellizzazione dell'utente e conseguente personalizzazione dei contenuti.
L'inarrestabile progresso tecnologico ha dunque aperto nuove strade per questi sistemi, evidenziando la necessità di un completo rinnovamento dell'architettura e degli stessi obiettivi del sistema.

Durante l’evoluzione dei progetti legati agli scavi archeologici di Ercolano, anche i contenuti digitalizzati sono andati man mano aumentando in quantità, qualità e tipo, permettendo di rendere sempre più ipermediale il sistema.
Un così vasto e variegato insieme di contenuti informativi ha richiesto un grosso lavoro di raccolta, organizzazione e gestione sia dal punto di vista fisico che informatico rendendo estremamente interessanti i suddetti progetti.

2.2 Web e scavi di Ercolano, scopi di PerfectFit06

Il primo problema che si presenta per la definizione di un sistema ipermediale come quello di Ercolano, è il modo in cui rendere accessibili all'utente le informazioni disponibili.
Fin dalla prima implementazione in Toolbook è stato scelto l'ipertesto su PC come mezzo di diffusione, o meglio di organizzazione, di questi contenuti informativi.
Bisogna sottolineare che la scelta dell'ipertesto è stata molto poco banale in quanto effettuata in tempi in cui ancora il World Wide Web e l'ipertesto non erano diventati, neanche lontanamente, diffusi come lo sono attualmente.
La disponibilità in qualsiasi momento di una enorme mole di informazioni con la possibilità di ricerche su scala planetaria, la semplicità dell'interfaccia e la navigazione intuitiva hanno permesso l'accesso a questa tecnologia a milioni e milioni di utenti, rendendo il Web l'ambiente più appetibile per qualsiasi tipo di divulgazione e proponendolo, dunque, come un nuovo tipo di spazio per le informazioni[Rossi-1].
La mancanza in esso di un meccanismo di controllo centrale porta a nuove interessanti caratteristiche, ma, allo stesso tempo, presenta il pericolo potenziale che l'utente possa trovare molte informazioni irrilevanti o sentirsi totalmente smarrito.
I due principali problemi con i quali l'utente si deve confrontare sono quindi quelli del sovraccarico di informazioni e del disorientamento nello "spazio".[Rossi-1]
Nel primo caso ci sono più informazioni di quante un utente può maneggiare (sovraccarico cognitivo o effetto sirena): anche se, come lettore, egli ha chiaro in mente il tipo di informazione da recuperare, spesso si trova innanzi una tale abbondanza di materiale che il piacere del perdersi nei testi riesce a disorientare la ricerca stessa.[Rossi-1]
Nel secondo caso, l'utente, può trovarsi disorientato tra i nodi della rete smarrendo la cognizione del proprio percorso (effetto labirinto). Spesso il navigatore percepisce sulla rete un certo senso di disorientamento[Rossi-1] a cui reagisce solitamente abbandonando lo scopo che si era prefisso o perdendo tempo prezioso a scartare informazioni inutili.
E' dunque ovvio l'interesse di creare una architettura web per gli scavi di Ercolano in grado di integrare la semplicità e l'efficienza della navigazione web classica con le nuove tecniche e tecnologie capaci di assistere l'utente nella sua navigazione.

L’enorme mole di dati disponibili per gli scavi archeologici di Ercolano rappresenta un ottimo caso di studio per i sistemi di personalizzazione dei contenuti presentati nel primo capitolo, ed è in questo contesto che si inserisce PerfectFit06.
Il cuore del funzionamento di PerfectFit06, come sarà evidente in seguito, è l’interesse dell’utente; con “interesse dell’utente” si intende la base comportamentale secondo cui un utilizzatore del sistema sceglie, o è propenso a scegliere, determinati contenuti piuttosto che altri.
Come si può facilmente arguire, non è un compito banale tentare di “comprendere” l’interesse dell’utente e trasformare questa “comprensione” in effetti reali sulla qualità della navigazione.
L’architettura del sistema PerfectFit06 verte su moduli adibiti alla creazione e alla gestione di modelli atti a descrivere opportunamente l’interesse dell’utente col fine di semplificargli il reperimento delle informazioni per lui interessanti.
Mentre in un sistema web classico, ogni scelta all’interno dell’ambiente Web, è esplicita e i percorsi ipertestuali che portano a una data informazione sono stabiliti a priori, nel sistema web di PerfectFit06, l’intuizione degli interessi dell’utente porta, come effetto più evidente e immediato, un più facile raggiungimento di queste informazioni.
Nel sistema è stato inserito anche un modulo di raccolta appunti che permette all’utente, in qualsiasi momento, di selezionare dei contenuti, visualizzare la raccolta effettuata in una pagina a parte e salvarla sulla propria macchina.

2.3 Adattività di PerfectFit06

PerfectFit06 è un sistema di personalizzazione dei contenuti adattativo basato su modello utente (par 1.3, 1.5).
Lo scopo principale del sistema è quello di assistere la navigazione dell'utente in siti Web in cui la quantità di materiale informativo a disposizione è considerevole, com'è il caso degli Scavi di Ercolano.
PerfectFit06 tenta di individuare gli interessi dell'utente rispetto al contesto specifico effettuando un tracking (par 1.8) della sua interazione all'interno dell'ambiente Web (fig.2.3.1).

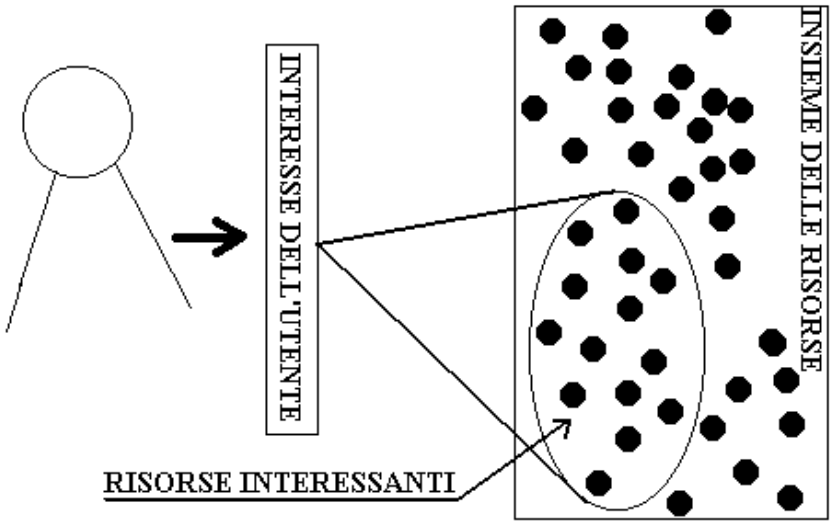


fig 2.3.1 schematizzazione del filtraggio effettuato da PF06

Per rappresentare gli interessi dell'utente si fa uso di un modello utente che viene aggiornato continuamente, durante la sessione di utilizzo del sistema, in base alle scelte che esso di volta in volta effettua.
Il sistema effettuerà dei calcoli, in funzione dello stato corrente del modello utente, per stabilire il grado di interesse rispetto a ogni informazione presente nella pagina.
Le informazioni ritenute più rilevanti per gli interessi dell'utente saranno rese visibili mentre quelle non ritenute sufficientemente rilevanti saranno parzialmente occultate.
All'utente saranno quindi sempre accessibili tutte le informazioni in modo che nessun percorso all'interno del sito gli sia precluso, semplicemente, i dati ritenuti più rilevanti, saranno messi in evidenza.

In PerfectFit06 tutte le risorse di una pagina, rilevanti o meno, rimangono comunque sempre accessibili, in modo che nessun percorso di navigazione sia precluso a priori dalle scelte del sistema.
Le operazioni effettuate dall'utente per visualizzare e nascondere delle informazioni sono ritenute altamente indicative delle sue preferenze e vengono sfruttare dal sistema per aggiornare il modello mediante tecniche avanzate (par 2.15.3).
Essendo PerfectFit06 un sistema adattativo, maggiore sarà il numero di interazioni e maggiore sarà il grado di precisione con cui si effettueranno le selezioni di risorse interessanti o meno.
Nella pratica, la capacità di adattatività e di tracking del sistema, non si limita a quanto descritto, ma, si è ritenuto ragionevole, fare una veloce presentazione delle caratteristiche salienti di PerfectFit06 per rendere più chiaro il prosieguo del capitolo.
La gestione delle informazioni sarà presentata nei par 2.5 – 2.9, la struttura del modello utente sarà trattata nei par 2.10-2.11, la gestione del suddetto modello nel par 2.15.

2.4 Introduzione all'architettura di PerfectFit06

La progettazione dell'architettura di PerfectFit06 è stata realizzata applicando la modularizzazione delle componenti per evitare di creare un sistema chiuso alle espansioni.
L'architettura del sistema è composta da moduli indipendenti funzionalmente che cooperano tra loro all'interno dell'ambiente di sistema.
Ogni modulo ha un compito più o meno specializzato a seconda dei casi.
L'organizzazione dell'architettura è stata pensata in modo tale da distinguere opportunamente i moduli con compiti specifici.
Un'applicazione così spinta della modularità permette di aggiornare e aggiungere componenti al sistema lasciando la più ampia apertura a miglioramenti e nuove idee.
Modularizzare un sistema web spesso significa rinunciare a parte dell'efficienza a causa, soprattutto, di accessi ripetitivi alle basi di dati.
Il sistema è stato progettato tenendo conto della esigenza di ridurre il sovraccarico dovuto a questo aspetto.
Entrando maggiormente nel dettaglio tecnico e richiamando i concetti presentati nel primo capitolo, PerfectFit06 è un sistema ipermediale adattativo di personalizzazione e raccomandazione dei contenuti basato su regole con modello utente individuale inferito.

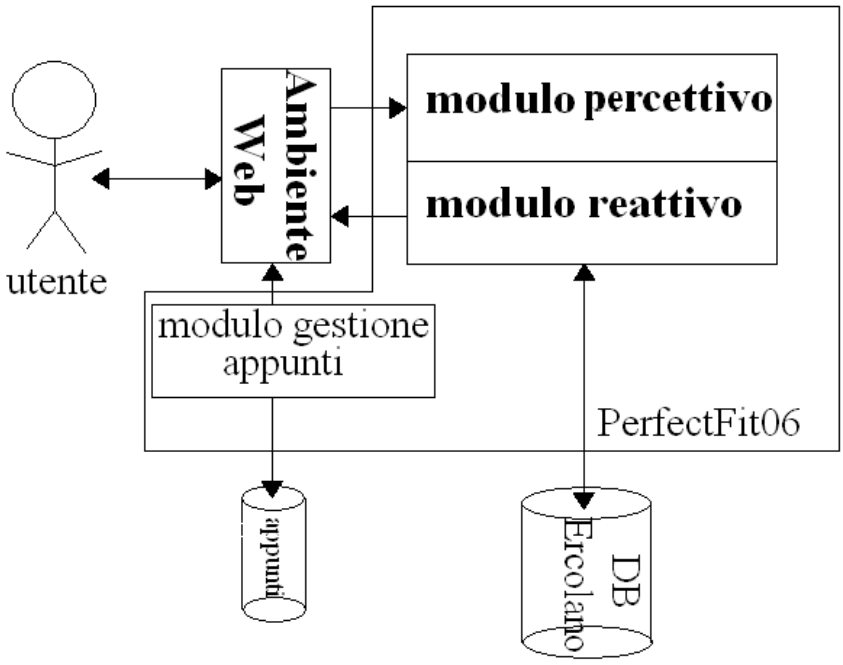


fig. 2.4.1 Architettura di PerfectFit06 al livello più astratto

L'architettura di PerfectFit06 è piuttosto complessa.
Presentarne le componenti e la loro organizzazione, senza esibire le motivazioni alla base della loro esistenza, vuol dire rinunciare in partenza alla chiarezza.
Verrà fornita quindi una introduzione graduale all'architettura del sistema per mezzo di livelli di astrazione progressivamente più dettagliati.
Le prime componenti da distinguere sono quelle relative all'interazione dell'utente col sistema e sono riconoscibili nella seguente figura 2.4.1
L'utente effettua tutta la sua interazione col sistema all'interno dell'ambiente Web.
Per ambiente Web, in questo contesto, si fa sinteticamente riferimento all'insieme di programmi, protocolli e meccanismi che nel loro complesso permettono all'utente di accedere alle pagine web del sistema e di interagire, nel modo a lui più congeniale, tramite l'interfaccia di navigazione messa a disposizione dal browser.
L'utente opererà completamente nell'ambiente Web effettuando, al suo interno, tutte le scelte come la visualizzazione di determinate informazioni e lo scarto di altre.
Il primo passo di PerfectFit06 è catturare queste interazioni effettuando il tracking dell'utente (par 1.8).
La cattura delle operazioni effettuate dall'utente riguarderà un modulo percettivo capace di raccogliere, con tecniche specificate nei paragrafi successivi, le scelte discriminate dall'operato dell'utente; le informazioni raccolte saranno elaborate e poi sfruttate dal sistema per personalizzare la navigazione dell'utente
Rispetto agli altri moduli, quello percettivo, sarà fortemente dipendente dal contesto in quanto, necessariamente, deve essere immerso completamente nell'ambiente Web specifico.
Come intuibile dalla fig. 2.4.1, il modulo percettivo avrà il compito di raccogliere informazioni sull'interazione dell'utente "all'esterno" di PerfectFit06, ovvero nell'ambiente Web, e riportarle all'interno del sistema in modo che quest'ultimo le possa analizzare.
Le informazioni raccolte dal modulo percettivo verranno opportunamente elaborate dal sistema che le trasformerà, tramite adeguate regole, in una serie di altre informazioni utili a definire gli interessi dell'utente.
I dati raccolti e inferiti, riguardanti l'interesse dell'utente, verranno sfruttati dal modulo reattivo per mostrare, in maniera personalizzata, le informazioni nell'ambiente Web.

Sono visibili nello schema in fig. 2.4.1 due interazioni con altrettanti database; anche se nella pratica il database a cui accede il sistema è fisicamente unico, a questo livello concettuale, è giusto vederlo scomposto in due entità distinte.
L'accesso dell'utente agli appunti avviene tramite il modulo intermedio di gestione degli appunti
Questo modulo ha il ruolo di recepire gli ordini impartiti dall'utente nell'ambiente Web relativamente alla raccolta di appunti.
Il modulo degli appunti avrà principalmente due ruoli, quello di stipare nel database i contenuti informativi indicati esplicitamente dall'utente e quello di prelevarli e visualizzarli quando l'utente stesso ne fa richiesta.
L'altra interazione avviene tra l'intero sistema PerfectFit06 e il database di Ercolano; questa interazione riguarda tutti i contenuti informatici e le strutture dati principali del sistema.
Gran parte di questo capitolo chiarirà quali funzionalità e quali meccanismi sono collegati a questa parte del sistema.

2.5 Grafo della conoscenza

I primi problemi affrontati per la realizzazione del sistema ipermediale di Ercolano sono stati quelli della definizione e organizzazione dei contenuti a un livello di astrazione alto.

Fin dalla prima edizione del sistema ipermediale di Ercolano è stato realizzato un grafo della conoscenza relativo alle risorse informative degli scavi in questione.

Per risorse informative si intendono nuclei informativi che possono riguardare qualsiasi argomento, dalla mera informazione turistica fino a approfondimenti di aspetti tecnici di archeologia, architettura etc.

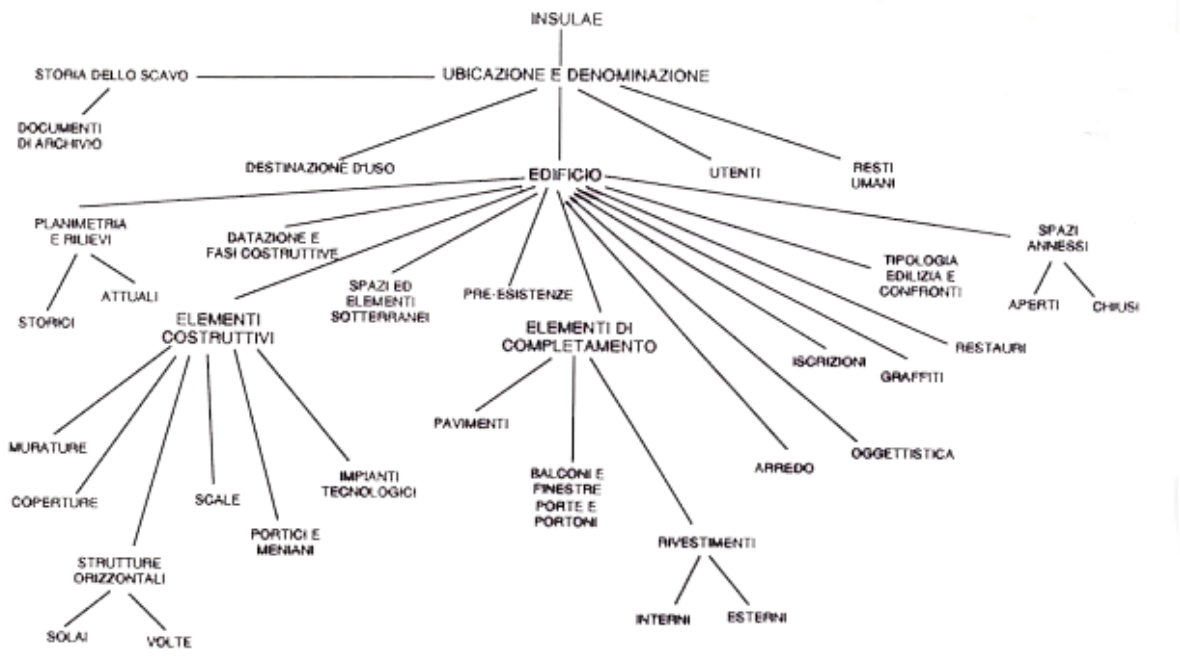


fig. 2.5.1 Grafo della conoscenza per gli scavi archeologici di Ercolano

Il grafo è il risultato di un grosso sforzo di definizione, raccolta e organizzazione di dati e metadati realizzato grazie alla stretta collaborazione di un gruppo multidisciplinare di esperti del settore archeologico, architettonico, storico e informatico.

Il grafo della conoscenza riportato in figura 2.5.1 stabilisce un valido punto di partenza per qualsiasi struttura ipermediale definendo, in maniera inequivocabile, le risorse informative che si troverà a gestire il sistema e la relativa nomenclatura da utilizzare.

Un altro aspetto di fondamentale importanza, presentato nel grafo della conoscenza, è la definizione delle relazioni intercorrenti tra le varie risorse che soddisfa l'indubbia

necessità, da parte di un non-esperto, di avere una chiara indicazione del modo in cui le informazioni si correlano tra loro.
Anche se durante gli anni non ha subito grosse rivoluzioni, la struttura del grafo della conoscenza, è stata ampliata e perfezionata raccogliendo e definendo sempre con maggior dettaglio e precisione le informazioni disponibili.
Ciò che rende questo albero estremamente interessante e funzionale è la sua capacità di definire adeguatamente anche informazioni trasversali rispetto a vari campi di interesse inerenti al contesto.
Come si può dedurre dalla figura 2.5.1, i nodi del grafo definiscono risorse informative di carattere notevolmente differenziato che va da indicazioni generiche fino a dettagli storici, architettonici, archeologici e così via.

2.6 Adattamento del grafo al Web

L'architettura e le strutture dati di ogni versione del sistema ipermediale di Ercolano sono state concepite basandosi in maniera determinante sul grafo della conoscenza descritto nel paragrafo precedente.
L'utilizzo principale che si è fatto del grafo riguarda la strutturazione dei contenuti all'interno del sistema.
In maniera abbastanza esplicita è stata realizzata una trasposizione dei percorsi tracciabili sul grafo in percorsi realizzabili dall'utente all'interno dell'ipertesto.
La trasposizione a cui si è fatto cenno è stata realizzata anche per l'attuale sistema web di Ercolano.
Rispetto al passato sono stati però effettuati degli accorgimenti per rendere maggiormente usabile il sito finale.
Questi accorgimenti hanno riguardato principalmente l'altezza del grafo, che è decisamente troppo elevata per essere trasposta brutalmente in un grafo di collegamenti ipertestuali.
Per rendersene conto basta pensare, osservando la figura 2.5.1, che tra la radice 'insulae' e la foglia 'solai' bisogna attraversare altri quattro nodi, per un totale di sei pagine visitate dall'utente per raggiungere una informazione.
Si è dunque pensato di raggruppare, con criteri ad hoc, alcuni nodi e di realizzare una classificazione che permettesse la riduzione delle pagine web da visitare per raggiungere le foglie.
Il criterio di raggruppamento è stato definito per ogni nodo, in base al numero di nodi-foglia: i nodi con non più di due figli e caratterizzati come foglie sono stati considerati come un unico nodo.
Il risultato dell'applicazione del criterio di raggruppamento è mostrato in figura 2.6.1.

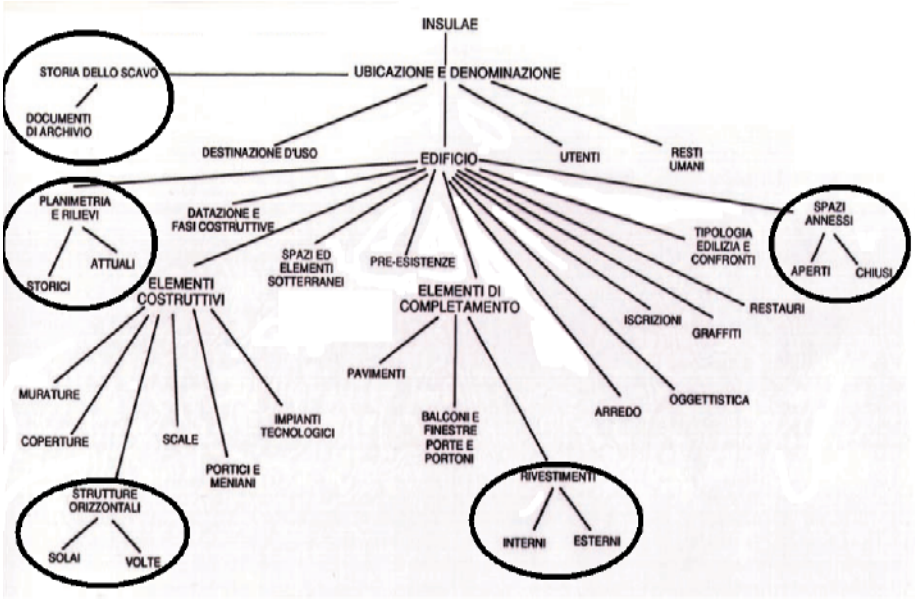


fig. 2.6.1 Primo livello di adattamento del grafo

I sottoalberi cerchiati sono dunque considerati nodi unici.

Anche se già significativa, la riduzione del numero di nodi effettuata, non è comunque sufficiente da sola al raggiungimento dello scopo prefissato.

Un altro passo importante, nella riorganizzazione per il web del grafo della conoscenza, è stato l'individuazione di nodi principali e nodi secondari.

Per nodo principale si intende un nodo con elevato contenuto informativo.

Per nodo secondario si intende un nodo con ridotto contenuto informativo.

Si precisa che con i termini "elevato contenuto informativo" e "ridotto contenuto informativo" si fa riferimento a indicazioni di carattere quantitativo e non qualitativo, questo per sottolineare che sono valutazioni fatte a priori sul grafo e non sui dati reali.

Verranno considerati nodi principali tutti quelli aventi almeno tre figli, secondari tutti gli altri; l'operazione verrà effettuata sul grafo ottenuto in precedenza.

La conseguenza di questa classificazione è mostrato nell'albero in figura 2.6.2

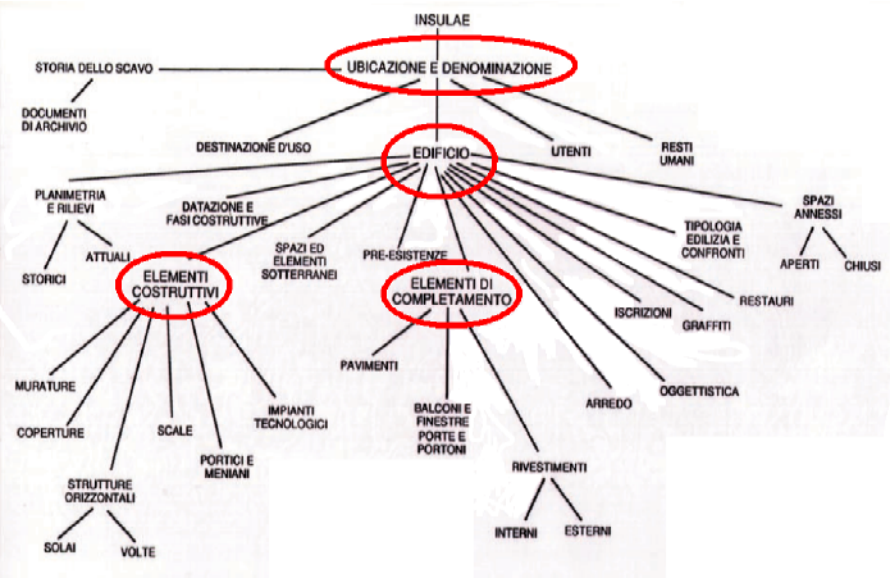


fig. 2.6.2 - secondo livello di adattamento del grafo

Questa operazione influenza nel seguente modo l'interno del sito web:

- I nodi principali verranno mostrati all'utente come pagine web a se stanti.
- I nodi principali figli di un nodo principale verranno mostrati come un link a un'altra pagina web.

Nel caso specifico del grafo della conoscenza di Ercolano si avrà che:

- I nodi secondari, in questo grafo, saranno sempre figli di un nodo principale.
- I nodi secondari verranno presentati come testo a espansione all'interno della pagine web del nodo padre.

Per testo ad espansione si intende una sottosezione della pagina web collegata a un tasto capace di far apparire o sparire il contenuto del nodo (fig. 2.6.3).

Bisogna sottolineare che c'è un punto critico, nell'utilizzo del grafo presentato, legato al ruolo dei nodi "Insulae" e "Ubicazione e denominazione".

Infatti, mentre le istanziazioni di tutti gli altri nodi sono uniche, i due nodi citati possono avere istanziazioni multiple.



fig. 2.6.3 applicazione del tasto mostra/nascondi in PerfectFit06

Maggior chiarezza può essere fatta con l'esempio reale, per il nodo 'Insulae' esistono, come istanze reali, "Insulae VI" e "Insulae III" così come per il nodo "Ubicazione e denominazione" esistono "Terme del foro", "Collegio degli Augustali" e tanti altri.
Il sottoalbero di "Ubicazione e denominazione" è invece formato da tutti nodi con una unica istanza.
Quindi si ha che esistono più istanze di 'insulae', per ogni 'insulae' esistono più istanze di 'ubicazione e denominazione' mentre per ogni sottoalbero di "ubicazione e denominazione" esiste una sola istanza per nodo.

2.7 Rappresentazione del grafo nel sistema web

PerfectFit06 non si pone come sistema adattativo solo rispetto all'utente ma anche rispetto alle strutture dati principali.
Come si è già accennato, il grafo della conoscenza non è definitivo poiché soggetto a variazioni di raffinamento o ampliamento.
Progettare un'architettura qualsiasi che si basi in maniera statica e definitiva su una versione del grafo vuol dire legare indissolubilmente il suo ciclo di vita con quello di quella data versione del grafo.
Il primo passo realizzato nella progettazione dell'architettura di PerfectFit06 è stato la definizione di una struttura dati capace di rappresentare adeguatamente il grafo permettendo la massima flessibilità riguardo a modifiche dello stesso, PerfectFit06 non è solo capace di adattarsi agli aggiornamenti del grafo presentato ma, addirittura, a qualsiasi grafo rappresentabile con la struttura presentata nella figura 2.7.1.

Campo	Tipo	Attributi	Null	Predefinito	Extra
id_struttura_albero	bigint(20)		No		auto_increment
padre	bigint(20)		Si	*NULL*	
nome_nodo	varchar(255)		No		

fig 2.7.1 descrizione del grafo mediante la tabella "struttura_albero":

I campi hanno il seguente significato:

id_struttura_albero **:** è l'identificativo univoco della tabella.

padre **:** contiene l'identificativo univoco nella tabella struttura_albero del nodo che è padre nel grafo del nodo corrente.

nome_nodo **:** è il nome mnemonico dato al nodo.

In parole povere viene rappresentato, in ogni riga della tabella, un singolo nodo del grafo indicando solo il suo nome e il suo nodo-padre. Il sistema sarà in grado di ricostruire l'intero albero, o l'intera foresta, partendo dai nodi "senza padre" e recuperando, passo dopo passo, i figli in base ai valori del campo 'padre' contenuti nella tabella.

Questa tabella permette in qualsiasi momento di ridefinire, anche completamente, il grafo; in realtà, con questa strutturazione, è possibile rappresentare un qualsiasi albero o foresta.
Se si vuole aggiungere un nodo, basta creare una nuova riga e inserire il nome del nodo e l'identificativo del padre, al resto penserà PerfectFit06.
PerfectFit06 è capace di adattarsi, in maniera del tutto autonoma e in qualsiasi momento, alla struttura di albero rappresentata nella tabella, l'unica eccezione è per i nodi 'insulae' e 'ubicazione e denominazione' che per motivi già citati sono strutturalmente diversi dagli altri e sono stati trattati come elementi speciali.
I valori inseriti all'interno della tabella per la descrizione del grafo si basano sulla riduzione effettuata nel par. 2.6.
PerfectFit06 è anche capace, grazie a questa descrizione del grafo, di discriminare del tutto automaticamente i nodi principali dai nodi secondari in base ai criteri descritti nel paragrafo 2.6.

2.8 Le “risorse”

Per PerfectFit06 si è progettata un’architettura quanto più indipendente possibile dal contesto specifico degli scavi archeologici in modo tale da poter rendere le sue componenti maggiormente riutilizzabili.
Un passo fondamentale, nel processo di astrazione realizzato, ha portato a rendere indipendenti i moduli di livello logico più alto del sistema PerfectFit06, dalle informazioni specifiche che esso tratta.
Il sistema non conosce direttamente i dati ma è capace di gestire solo una loro astrazione, definita nell’insieme delle “risorse”.
Una risorsa è un singolo nucleo informativo ed è l’elemento atomico gestibile dal sistema per quel che concerne i dati del contesto.
L’insieme delle risorse è il risultato di un partizionamento della totalità dei dati secondo criteri semantici.
Nel momento in cui il sistema deve comporre una serie di informazioni, ad esempio per creare una pagina web, gli basta raccogliere opportunamente le risorse in cui il progettista ha scomposto l’informazione iniziale.
E’ evidente che bisogna stabilire dei criteri per definire correttamente le risorse
Seguire i seguenti passi è sicuramente un punto di partenza:

1. Stabilire la granularità della scomposizione delle informazioni in risorse.
2. Applicare un metodo di scomposizione coerente per l’intero sistema.
3. Classificare opportunamente le risorse in base a caratteristiche comuni e utili ai fini del sistema.
4. Stabilire le relazioni intercorrenti tra le varie risorse, ad esempio dipendenze e legami semantici.

Bisogna creare, in pratica, un grafo della conoscenza.
E’ necessario distinguere il concetto di “classe risorsa” da quello di “istanza di risorsa”.
Una “classe risorsa” individua un insieme di contenuti informativi caratterizzati da proprietà simili rispetto ai criteri di scomposizione adottati dal progettista.
Una “istanza di risorsa” è invece un elemento di una data classe risorsa che ha in se il reale contenuto informativo.

In PerfectFit06, per la classificazione delle risorse, si è usato l'albero adattato dal grafo della conoscenza presentato nel par. 2.6 dove ogni nodo del grafo rappresenta un "classe risorsa".
Il fatto che sia stato usato l'albero della conoscenza di Ercolano per la definizione delle risorse, permette al sistema di sfruttare la stessa struttura dati utilizzata per memorizzare l'albero al fine di riconoscere dinamicamente le classi risorsa.
Il tipo di classificazione utilizzata per realizzare il grafo usato dal sistema definisce in maniera esplicita anche le risorse del sistema e, dunque, i contenuti che verranno personalizzati.
La risorsa è stata definita anche come componente informativo atomico, di conseguenza, maggiore sarà il dettaglio con cui sarà realizzato il grafo, maggiore sarà la precisione del sistema nel realizzare la personalizzazione dei contenuti.

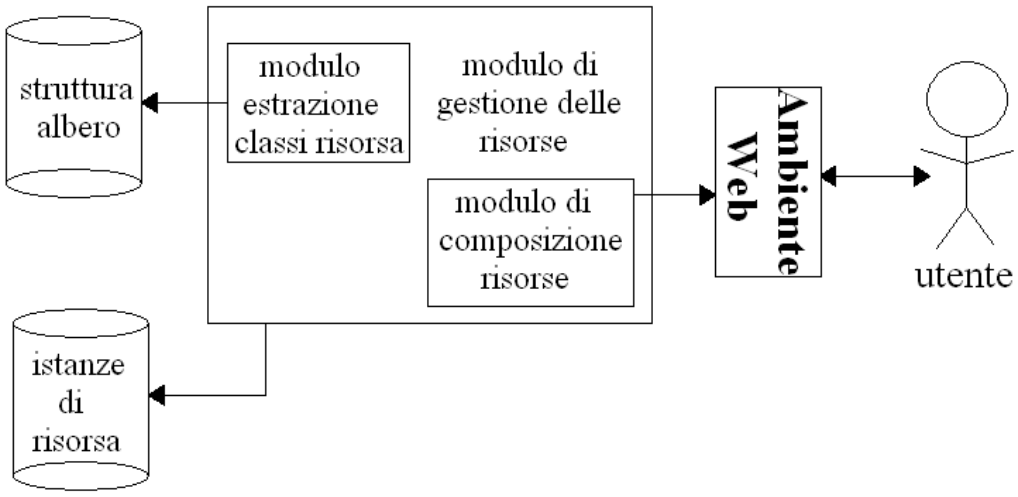


fig 2.8.1 Modello concettuale della gestione delle risorse

Poiché la creazione di un grafo della conoscenza, o qualcosa di simile funzionalmente, è un primo passo per tutti i sistemi adattativi o intelligenti, avere un sistema che rispetti un modello concettuale come quello riportato in fig 2.8.1 permette un assoluto riutilizzo dei moduli di alto livello e soprattutto delle idee, nonché rende facile l'inserimento di nuovi moduli per aggiungere o migliorare la parte "intelligente" del sistema.
Si noti inoltre che nella definizione di risorsa non viene data nessuna specifica sul tipo di media.
Praticamente la risorsa può essere indipendentemente video, testo, immagini o altro.

Per quanto riguarda le istanze di risorsa, sicuramente c'è bisogno di almeno una tabella per contenerle, in PerfectFit06 si è fatto uso della seguente struttura dati "istanze_risorsa":

Campo	Tipo	Attributi	Null	Predefinito	Extra
id_istanze_risorsa	bigint(20)		No		auto_increment
nome_istanza	varchar(255)		No		
id_struttura_albero	bigint(20)		No	0	
id_istanza_padre	bigint(20)		Si	*NULL*	
contenuto	varchar(255)		Si	blank.htm	

fig 2.8.2 Struttura dati "istanze_risorsa"

i campi hanno il seguente significato:

id_istanze_risorse: identificativo univoco dell'istanza risorsa.
nome_istanza: è il nome specifico dell'istanza, ad esempio "insulae VI", "terme del foro".
id_struttura_albero: fa riferimento all'identificativo della tabella "struttura_albero" presentata nel paragrafo precedente, permette di identificare la classe d'appartenenza dell'istanza.
id_istanza_padre: è l'identificativo di una istanza risorsa, serve per rappresentare, tra due istanze, il legame dalla relazione padre-figlio definito nell'albero della conoscenza.
contenuto: indica il percorso di una pagina web esterna, scritta in HTML o una sua evoluzione, formata dal solo contenuto informativo specifico dell'istanza di risorsa.

La scelta di tenere esterni al sistema i documenti col codice HTML permette di modificarli a proprio piacimento senza aver bisogno di conoscere o di accedere al codice interno di PerfectFit06.
Grazie alle relazioni definite nella tabella "struttura_albero" e "istanze_risorsa", il sistema sarà in grado di provvedere autonomamente alla composizione dei pezzi di codici HTML delle risorse in pagine Web secondo i criteri specifici che esso adotterà.
I criteri di cui sopra, sono quelli che distinguono la tipologia del sistema, cioè sono al livello logico che permette di definire se esso è adattativo, intelligente o una qualsiasi combinazione dei due.

Il fatto che i contenuti informativi delle risorse siano memorizzate sotto forma di pagine web esterne permette anche di disinteressarsi, a livello di sistema, del tipo di media utilizzato per la rappresentazione della risorsa.

2.9 Il modello risorsa

Aver astratto i dati in risorse è senza dubbio utile e apporta molta flessibilità al sistema. Nel paragrafo precedente si è già discusso della necessità di classificare le risorse secondo criteri coerenti e pertinenti al contesto semantico specifico.
Effettuare una classificazione significa individuare delle proprietà caratteristiche del contesto e suddividere i vari elementi in insiemi omogenei basandosi sui valori di queste proprietà.
Anche se gran parte del lavoro di organizzazione dei dati in risorse è stato facilitato dall'esistenza del grafo, ci si è trovati comunque a dover definire le proprietà caratteristiche, ovvero un modello, per le risorse.
Si è stabilito che, ad ogni classe risorsa, venisse associato un modello con determinati valori; conseguenza di questo fatto è che tutte le risorse, istanza della stessa classe, sono associate a un stesso modello.
Con la collaborazione di un gruppo formato da architetti e archeologi è stata effettuata una scelta delle proprietà ritenute più specifiche per le risorse di Ercolano, in particolare si è tentato di modellare le risorse in base agli interessi che esse possono suscitare nel contesto.
Ogni campo del modello risorsa rappresenta, in percentuale, il presentarsi di quel caratteristico interesse per la classe risorsa che si sta modellando.
Segue una descrizione dei campi del modello risorsa:

Azione conservativa: indica quanto la classe risorsa descriva le azioni di restauro o mantenimento effettuate per oggetti o costruzioni.
Decorativa: indica quanto la classe risorsa descriva le decorazioni di un oggetto o di una costruzione.
Destinazione d'uso: indica quanto la classe risorsa descriva l'uso che si faceva di un oggetto o di una costruzione.
Documentaria: indica quanto la classe risorsa descriva documenti di ritrovamenti, scavi, etc.

Esplorativa: indica quanto la classe risorsa descriva opere o risultati emersi da azioni di scavo
Iconografica: indica quanto la classe risorsa descriva o rappresenti l'iconografia di un oggetto o una costruzione.
Informativa: indica quanto la classe risorsa descriva o rappresenti nozioni di carattere generico, ovvero di nessun interesse specifico.
Materica: indica quanto la classe risorsa descriva i materiali di cui è composto un oggetto o una costruzione.
Planimetrica: indica quanto la classe risorsa descriva o rappresenti contenuti di tipo planimetrico.
Storica: indica quanto la classe risorsa descriva contenuti di tipo storico, nel senso di narrazione di eventi storici.
Tecnica: indica quanto la classe risorsa descriva le tecniche utilizzate per la creazione e l'utilizzo di oggetti o di costruzioni.
Tecnologica: indica quanto la classe risorsa descriva le tecnologie adoperate o presenti in oggetti e costruzioni.
Tipologica: indica quanto la classe risorsa descriva o rappresenti contenuti riguardanti la tipologia.
Topografica: indica quanto la classe risorsa descriva o rappresenti contenuti di tipo topografico.

I valori dei campi del modello risorsa sono tra loro indipendenti.
Una volta presentati i campi del modello risorsa è il momento di descrivere come effettivamente avviene l'associazione tra una risorsa e il suo modello in PerfectFit06.
A ogni classe è associato un insieme specifico di valori del modello, come detto nel paragrafo 2.8 le classi sono identificate dai nodi definiti nella tabella "struttura_albero".
Ovviamente, ogni modello creato per una classe risorsa, è stato anch'esso associato alla tabella "struttura _albero" .
La struttura dati usata per la memorizzazione dei modelli risorsa istanziati per PerfectFit06 è quella in figura 2.9.1.

Campo	Tipo	Attributi	Null	Predefinito	Extra
id_profilo_risorse	bigint(20)		No		auto_increment
id_struttura_albero	bigint(20)		No	0	
topografica	smallint(6)		No	0	
tipologica	smallint(6)		No	0	
planimetrica	smallint(6)		No	0	
informativa	smallint(6)		No	0	
storica	smallint(6)		No	0	
documentaria	smallint(6)		No	0	
materica	smallint(6)		No	0	
decorativa	smallint(6)		No	0	
iconografica	smallint(6)		No	0	
destinazione_d_uso	smallint(6)		No	0	
esplorativa	smallint(6)		No	0	
azione_conservativa	smallint(6)		No	0	
tecnica	smallint(6)		No	0	
tecnologica	smallint(6)		No	0	

fig 2.9.1 Struttura dati "Struttura_albero"

Non c'è bisogno di ripetere la descrizione dei campi prettamente legati al modello risorsa, seguirà quindi una breve descrizione solo dei campi utili ai fini del database:

id_profilo_risorse: identificativo univoco della tabella
id_struttura_albero: l'identificativo della riga della tabella "struttura_albero" che permette di identificare la classe risorsa a cui è collegata la tupla di valori della riga corrente.

E' necessario che per ogni nodo dell'albero, quindi per ogni classe risorsa, esista un modello associato, altrimenti il sistema rischia di entrare in uno stato di incoerenza per quel che concerne la rappresentazione delle risorse.

2.10 Tipo di modello utente di PerfectFit06

In accordo con la teoria presentata nel primo capitolo, è chiaro che il cuore di un sistema adattativo come PerfectFit06 è il modello utente.
Scegliere quali aspetti modellare di un utente, come realizzare e come gestire il modello, sono decisioni molto difficili, soggettive del progettista ed estremamente determinanti.
Sbagliare una sola scelta durante la fase di modellizzazione dell'utente può pregiudicare il funzionamento dell'intero sistema.
Per definire la tipologia di modello utente secondo le definizioni presentate nel paragrafo 1.5.2 si è realizzata una analisi dei requisiti funzionali del sistema.
Lo scopo di PerfectFit06 è di offrire all'utente una personalizzazione dei contenuti basata unicamente sull'analisi del suo comportamento all'interno dell'ambiente del sistema e senza domande esplicite sulle sue preferenze, questo porta alla scelta di un modello di tipo inferito.
Per poter gestire un modello quanto più possibile capace di sagomarsi sui comportamenti dell'utente, si è utilizzato un modello individuale, a scapito di un modello prototipato che non avrebbe offerto la stessa precisione potenziale.
Il sistema deve essere in grado di raffinare la sua rappresentazione dell'utente durante la navigazione, quindi il modello deve essere dinamico.
PerfectFit06 prevede dei moduli dedicati alla gestione del modello utente che deve, dunque, essere esplicito.
Il sistema PerfecFit06 fa quindi uso di un tipo di modello utente individuale, dinamico, inferito, esplicito.

2.11 Il modello utente di PerfectFit06

Definito il tipo di modello usato, si può andare ora a vedere quali caratteristiche dell'utente si vanno a modellare.
Per realizzare la modellizzazione dell'utente si è partiti dalla definizione di classi di utente ideali definite come classi '*pure*'.
Una classe utente *'pura'* è la rappresentazione di un insieme di utenti con comportamenti e interessi estremamente definiti e caratterizzanti, una classe prototipata a tutti gli effetti.
E' di fondamentale importanza che gli utenti 'puri', cioè appartenenti a una classe *pura*, non presentino nessun comportamento che si possa far risalire direttamente a quelli di un'altra classe *pura*.
Un utente puro non è un utente reale, ma solo l'astrazione di uno modello comportamentale.
Il discorso relativo a un utente reale è sicuramente più ampio e frammentato, difficilmente un prototipo può rappresentare bene o completamente un utente reale, d'altronde questo è il limite più evidente dei sistemi adattabili.
Ogni utente reale si presenterà con comportamenti considerabili estremamente personali e poco classificabili.
Una ipotesi basilare di PerfectFit06 è che un utente reale appartenga contemporaneamente a più classi *pure* e che questa sua appartenenza sia quantificabile, seppure con le dovute approssimazioni.
Considerare un utente come appartenente a più classi è ragionevole, una persona può naturalmente appartenere alla classe degli sciatori e a quella dei cantanti.
Ovviamente le classi *pure,* a cui appartiene contemporaneamente un utente reale, non devono essere in contrapposizione o sovrapposizione comportamentale tra loro per questioni di coerenza.
Un utente, ad esempio, non può essere sia 'onesto' che 'corrotto'.
Il mantenimento della coerenza è la ragione per cui nelle righe precedenti si è sottolineata l'importanza di definire classi *pure* con comportamenti unici e non direttamente ridefinibili in funzione delle altre classi, richiamando l'esempio precedente, la classe 'onesto' sarà la semplice negazione della classe 'corrotto'.
Con la definizione delle classi *pure* si va a definire a tutti gli effetti una base canonica comportamentale del modello utente.

E’ necessario sottolineare che la classificazione va realizzata comunque in base a un contesto di riferimento che stabilisce anche il livello di astrazione del modello, è ovvio che realizzando un modello utente per un sito di articoli sportivi si partirà dal presupposto che tutti gli utenti abbiano in comune un interesse per lo sport.

2.12 Struttura del modello utente di PerfectFit06

La struttura del modello utente di PerfectFit06 è formata da un numero di coppie attributo-valore pari al numero di classi *pure* individuate.
Ad ogni attributo sarà associato un valore che indica il grado di appartenenza dell’utente alla classe *pura* rappresentata.
In PerfectFit06, il valore che indica il grado di appartenenza di un utente a una classe *pura*, è espresso in percentuale.
L’appartenenza a una data classe si può considerare, come detto, indipendente dall’appartenenza alle altre, questa indipendenza è ereditata da ogni campo del modello utente che non ha vincoli fissati in funzione degli altri campi.
Un esempio possibile di modello utente a quattro campi per modellare delle capacità pratiche è il seguente:

Falegname:	80%
Elettricista:	0%
Idraulico:	10%
Carpentiere:	50%

Ogni coppia attributo-valore del modello rappresenta l’appartenenza dell’utente alla classe *pura* indicata nel campo.
Quindi l’utente modellato sarà un discreto falegname, totalmente incapace come elettricista, piuttosto incapace come idraulico e un discreto carpentiere.
E’ ovvio che con questo tipo di modello si può scegliere di modellare un qualsiasi aspetto dell’utente, ad esempio le sue capacità pratiche, i suoi interessi, le sue capacità intellettive etc.

Bisogna comunque definire opportunamente le classi *pure*, in maniera coerente alla caratteristica dell'utente che si sta modellando, usando le accortezze già presentate nel paragrafo precedente.
Va sottolineato che per descrivere più aspetti di un utente è consigliabile utilizzare altrettanti modelli da gestire separatamente tra loro.

2.13 Adattatività strutturale di PerfectFit06 rispetto al modello

Una colonna portante dell'architettura di PerfectFit06 è la sua indipendenza dal contesto di applicazione grazie alla gestione di oggetti astratti quali le risorse (par 2.8).
Pensare a tutta questa flessibilità nella definizione delle risorse senza offrire la possibilità di variare con altrettanta facilità la struttura del modello utente significherebbe lasciare incompleta questa grande potenzialità del sistema.
PerfectFit06 realizza la gestione di un modello utente esplicito (paragrafo 1.5.2).
Il fatto che nel codice sia rappresentata una struttura dati per il modello utente implica che una modifica alla struttura del modello è subordinata alla modifica del codice.
L'applicazione della modularità nell'architettura di sistema in questo caso diventa una caratteristica fondamentale per sormontare questo ostacolo.

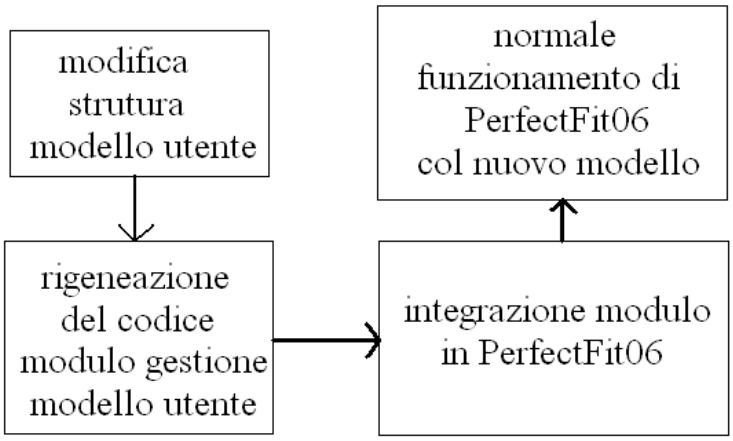


fig. 2.13.1 Schema di rigenerazione del codice di perfectFito06

PerfectFit06 è capace di rigenerare in maniera totalmente automatica il codice del modulo di gestione del modello utente e, di conseguenza, di adattarsi alle modifiche della sua struttura (fig. 2.13.1).

La capacità di modificare e rigenerare dinamicamente dei suoi moduli è una delle potenzialità più forti di questo sistema.
I campi del modello utente, rappresentati dalle classi *pure* individuate, sono inseriti all'interno di una tabella del database da cui il sistema è capace di caricare le informazioni al fine di rigenerare il codice.
La tabella usata è "base_regole_associazione" (fig. 2.13.2) che trova il suo utilizzo principale nella definizione delle regole di associazione descritte più avanti (par 2.15.1).



profilo_utente_riferimento
archeologo
architetto
storico
antiquario
turista
artista
restauratore

fig. 2.13.2 tabella parziale della base canonica

Le classi *pure* presenti nella fig. 2.13.2 sono quelle individuate per PefectFit06.
Se si modifica o si aggiunge una riga a questa tabella, l'intero sistema PerfectFit06 si adatterà alla gestione di un modello utente col nuovo attributo – classe *pura*.

2.14 Inizializzazione del modello utente

Una volta stabilita la struttura del modello utente ci sono da definire i criteri di gestione e aggiornamento delle istanze di questo modello.
Le scelte che deve effettuare un sistemista in questa fase sono numerose ed estremamente delicate.
I sistemi adattativi hanno bisogno sempre che l'utente faccia un minimo di interazione prima di riuscire a definire un modello sufficientemente identificativo.
La prima scelta riguarda i valori del modello utente all'avvio di una sessione.
Qualsiasi scelta si faccia porta ad ottenere dei vantaggi nella velocità di adattamento se l'utente è vicino a questi valori, dei ritardi altrimenti.
Stabilire dei valori iniziali significa fare ipotesi sul futuro comportamento dell'utente.
In PerfectFit06 si è scelto di assegnare inizialmente zero a tutti i campi del modello e di affidarsi alla tecnica di normalizzazione di quest'ultimo (par. 2.17) per velocizzare l'adattività del sistema.
Quindi PerfectFit06 parte senza fare alcuna assunzione sulle caratteristiche dell'utente che accede al sistema.
La navigazione dell'utente all'interno dell'ambiente di PerfectFit06 è realizzata nell'insieme delle risorse presentato nel paragrafo 2.17.
Gli elementi più importanti per modellare l'utente sono proprio le scelte che esso esplicitamente effettua nella navigazione.

2.15 Regole di associazione e regole di estensione

Finora si sono descritti i due livelli di rappresentazione fondamentali del sistema, il modello risorsa e il modello utente, come concetti separati.
Nell'utilizzo fatto da PerfectFit06 di questi due modelli c'è una sorta di gioco ad incastro.
Il modello utente viene modificato dalle scelte compiute dall'utente nell'insieme delle risorse, la selezione delle risorse preferite dall'utente, a sua volta, dipende dallo stato del modello utente attuale.
Per comprendere il funzionamento del sistema c'è bisogno di esplicitare il legame tra questi due modelli e di andarne ad analizzare dettagliatamente la natura.
Esaminando il funzionamento di PerfectFit06 si parte dal presupposto di avere un modello utente definito.
Il sistema sfrutterà questo modello utente per generare una istanza di modello risorsa considerata "ideale" per l'utente modellato (par. 2.15.3).
La trasformazione di una istanza di un modello utente nell'istanza "ideale" di un modello risorsa avverrà tramite delle regole dette "regole di associazione" (par. 2.15.1).
In base alla risorsa ideale generata verranno poi selezionate le risorse reali ritenute interessanti per l'utente secondo i criteri che verranno presentati nel par 2.15.3.
Viceversa, quando l'utente effettuerà una operazione su una determinata risorsa, questa sua azione rappresenterà una manifestazione esplicita del suo comportamento e dovrà avere delle opportune conseguenze sullo stato del modello utente attuale.
Praticamente il modello utente dovrà essere modificato.
Le regole che stabiliscono quanto devono essere modificati i campi del modello utente, al presentarsi di un dato evento, si chiamano regole di estensione (par 2.15.2).
Le regole di associazione e le regole di estensione sono meccanismi essenziali del funzionamento di PerfectFit06 e assolutamente determinanti per l'efficacia del sistema.
Basti pensare che sono alla base delle due fasi fondamentali di tracking (regole di estensione) e di scelta delle risorse interessanti (regole di associazione).

2.15.1 Le regole di associazione

Le regole di associazione servono per rappresentare la regolarità di comportamento da parte di una classe di utenti[Meo-1].
Esse sono, in un certo senso, delle implicazioni[Lanzi-1].
Una regola di associazione è una misura statistica che lega insieme due variabili casuali X e Y e che risponde alla domanda "Dato X, in che quantità implica Y?"[Meo-1].
Le regole di associazione sono solitamente create sulla base di statistiche su dati reali[Lanzi-1].
In PerfectFit06 le due classi che vengono messe in gioco, per definire una regola di associazione, sono la classe utente *pura* e una specifica tipologia di interesse rispetto a una risorsa.
Una regola di associazione in PerfectFit06 stabilisce, data una classe *pura*, con quale percentuale una sua istanza sarà interessata a una data proprietà della risorsa.
Il valore di una regola di associazione può essere letto come doppia chiave di lettura. Da un lato misura il grado di interessamento dell'utente e dall'altro la probabilità che esso sia interessato a una data proprietà della risorsa.
Per come vengono usate le regole in PerfectFit06 le due interpretazioni coincidono.

	archeologo	architetto	storico	antiquario	turista	artista	restauratore
topografica	90	0	0	0	70	0	0
tipologica	100	0	0	100	0	0	0
planimetrica	0	80	0	0	0	0	0
informativa	0	0	0	0	100	0	0
storica	90	0	100	60	0	0	0
documentaria	0	0	100	0	0	0	0
materica	0	0	0	100	0	0	100
tecnica	100	100	0	0	0	0	0
tecnologica	0	90	0	0	0	0	0

fig 2.15.1 Estratto della griglia delle regole di associazione in PerfectFit06

Nella pratica, in base alle caratteristiche dei modelli di PerfectFit06, ogni classe *pura* di utenti deve essere associata, tramite una opportuna regola di associazione, a ogni proprietà del modello risorsa.
L'insieme delle regole di associazione sarà dunque una griglia di probabilità come mostrato in figura 2.15.1
La suddetta figura è anche il contenuto parziale della tabella "base_regole_associazione" del database di PerfectFit06 contenente le suddette regole.

Campo	Tipo	Attributi	Null	Predefinito	Extra
id_base_regole_associazione	bigint(20)		No		auto_increment
topografica	smallint(6)		No	0	
tipologica	smallint(6)		No	0	
planimetrica	smallint(6)		No	0	
informativa	smallint(6)		No	0	
storica	smallint(6)		No	0	
documentaria	smallint(6)		No	0	
materica	smallint(6)		No	0	
decorativa	smallint(6)		No	0	
iconografica	smallint(6)		No	0	
destinazione_d_uso	smallint(6)		No	0	
esplorativa	smallint(6)		No	0	
azione_conservativa	smallint(6)		No	0	
tecnica	smallint(6)		No	0	
tecnologica	smallint(6)		No	0	
profilo_utente_riferimento	varchar(255)		No		

fig 2.15.2 struttura della tabella "base_regole_associazione"

La struttura della tabella (fig. 2.15.2) è fondamentalmente formata da un modello risorsa associato a una classe *pura* del modello utente individuata nel campo "profilo_utente_riferimento" (par. 2.11).
In PerfectFit06, i valori delle regole di associazione sono stati stabiliti da un team composto da archeologi e architetti.
Gestire tramite queste strutture dati le regole di associazione permette di renderle estremamente semplici da variare.

2.15.2 Le regole di estensione

Estendere un modello significa principalmente aggiungervi dettaglio, raffinare la definizione/modellazione effettuata tramite le sue coppie attributo-valore.
Le regole di estensione hanno lo scopo di "perfezionare" una istanza di modello utente. Esse associano a un dato evento X, nell'ambiente di sistema, una rispettiva modifica dell'istanza del modello utente.
Concettualmente, le regole di estensione sono molto simili alle regole di associazione in quanto dato X ciò implica in qualche misura Y.
La prima differenza sostanziale è che, mentre le regole di associazione stabiliscono una percentuale, le regole di estensione stabiliscono quantitativamente le modifiche da apportare al modello utente.
La seconda differenza è che le regole di estensione vanno dal dominio degli eventi nell'ambiente a quello delle modifiche del modello utente.
In PerfectFit06, l'ambiente Web è formato da risorse.
Tutti gli eventi all'interno di quest'ambiente sono catturati nella fase di tracking dell'utente e, a ogni evento, viene associata una specifica modifica del modello utente (fig 2.15.3).

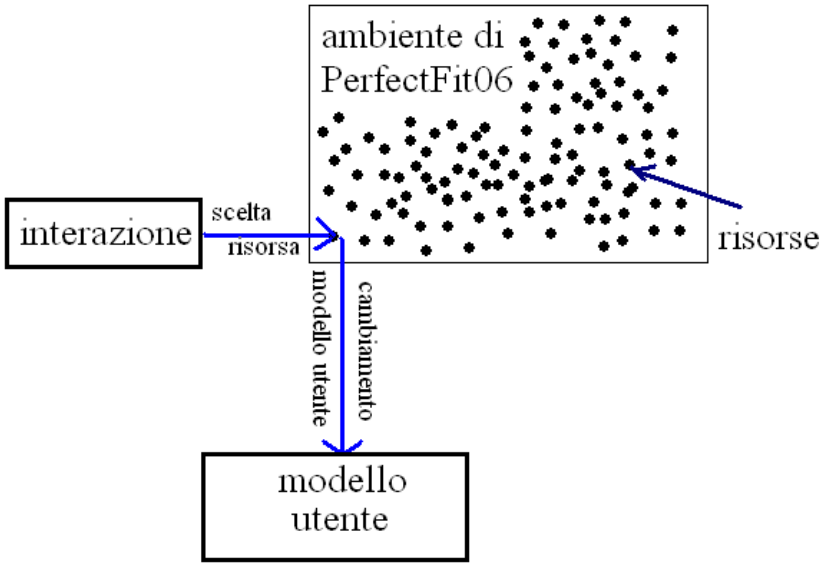


fig 2.15.3 Esemplificazione del meccanismo di applicazione delle regole di estensione

Ogni interazione dell'utente avrà quindi una relazione diretta con qualche risorsa, di conseguenza, le regole di estensione nel sistema trasformeranno ogni scelta dell'utente, rispetto a un'istanza di risorsa, in una modifica dello stato del suo modello.

Il legame fortissimo tra regole di estensione e risorse reali in PerfectFit06 è assolutamente determinante per il funzionamento del sistema.
Il peso di una modifica apportata al modello utente, usando una regola di estensione, deve essere ben ponderato.
Nel caso di PerfectFit06, per esempio, una modifica sostanziale si ha quando l'utente inserisce una risorsa nei suoi appunti multimediali (par. 3.3).

2.15.3 Modello risorsa ideale: creazione e utilizzo per il calcolo delle risorse interessanti

La funzione principale di un sistema di personalizzazione dei contenuti è quella di realizzare una cernita dei contenuti che possono essere interessanti o meno per l'utente[Tasso-1].
PerfectFit06 realizza la scelta delle risorse interessanti in funzione dello stato corrente del modello utente.
Il sistema provvede alla creazione di un modello risorsa ritenuto "ideale" rispetto al modello utente attuale e lo usa come punto di riferimento per la scelta delle risorse interessanti.
In base al modello risorsa "ideale" viene calcolata una distanza, secondo parametri che verranno presentati tra poco, rispetto ai modelli delle risorse reali su cui deve effettuare l'indagine.
Utilizzando come parametro di giudizio la distanza tra un modello risorsa reale e quello ideale, verrà stabilito il potenziale grado di interesse dell'utente per una data risorsa reale.
Le regole di associazione sono quelle che più di tutte marcano il legame tra modello utente e modello risorsa poiché, tramite queste regole, è possibile la trasformazione di un modello utente in un modello risorsa ideale.
Nella figura 2.15.2 del paragrafo 2.15.1 si è vista la struttura in cui vengono memorizzate le regole di associazione.
E' evidente che a ogni classe *pura* è associata una risorsa ideale, come descritto nel par 2.12.
Al modello utente sarà associato, durante l'interazione col sistema, un valore per il grado di appartenenza a ogni classe *pura.*
La risorsa ideale per una classe *pura* sarà ottenuta da una proporzione tra la risorsa ideale della classe e il grado di appartenenza dell'utente alla classe stessa.

La risorsa ideale così calcolata sarà "*parziale*" rispetto all'intero modello utente, poiché fa riferimento solo a un suo campo.
Le risorse ideali *parziali* verranno calcolate secondo la formula:

$$RIM = RI * MU / 100 \qquad (1)$$

Dove:
RIM è la risorsa ideale parziale generata da un campo del modello utente attuale.
RI è la risorsa ideale rispetto alla classe *pura* specificata nelle regole di associazione.
MU è il valore dell'attributo del modello utente, che varierà tra 1 e 100 (par 2.12), usato per la classe *pura* in questione.

L'operazione andrà effettuata per ogni classe *pura*, quindi per ogni coppia attributo-valore del modello utente attuale.
Si otterrà così una risorsa ideale *parziale* per ogni campo del modello utente attuale.
La risorsa ideale finale sarà ottenuta da una funzione applicata alle risorse ideali *parziali*.
Per definire questa funzione, in PerfectFit06, si è partiti dall'ipotesi che, ogni campo di queste risorse ideali *parziali*, rappresenti l'interesse dell'utente per la caratteristica della risorsa rappresentata dal campo in questione.
Un campo di una risorsa reale *"soddisfa"* lo stesso campo della risorsa ideale se gli è maggiore o uguale in valore, ovvero se offre un interesse che "copre" quello definito idealmente.
La risorsa ideale finale sarà dunque calcolata assegnando a ogni suo campo il valore massimo tra tutti quelli degli stessi campi nelle risorse ideali *parziali* calcolate finora.
In questo modo, per una ovvietà matematica, ogni campo di una risorsa reale che soddisfa il suo rispettivo ideale, soddisferà anche tutti i suoi rispettivi nelle risorse ideali *parziali*.
Partendo ora da questa risorsa ideale si calcolerà la "distanza", secondo criteri specifici del sistema tra poco presentati, tra essa e le varie risorse ideali.
Il valore ottenuto sarà determinante per valutare se la risorsa è troppo "distante" dagli interessi dell'utente o se invece può colpire la sua attenzione.
Per stabilire la distanza, in PerefectFit06, si procederà come segue.
Inizialmente, la distanza è nulla.

Vengono conteggiati i campi della risorsa reale che non soddisfano, secondo la definizione data poco sopra, i campi della risorsa ideale.
La differenza finale sarà data dalla seguente formula:

$$\mathbf{DF = round(100 - ((diff/tot)*100))} \qquad \textit{(2)}$$

Dove:
DF è la distanza finale tra la risorsa reale e quella ideale.
Round() è la funzione di arrotondamento.
diff è il numero di campi della risorsa che non soddisfano il modello ideale.
tot è il numero totale di campi del modello risorsa.

2.15.4 Calcolo automatico delle regole di estensione

Nel sistema Web di Ercolano, le regole di estensione non sono scritte esplicitamente ma sono calcolate automaticamente tramite opportune trasformazioni delle regole di associazione sulle classi *pure*.
La creazione di una singola regola di estensione avviene secondo una proporzione tra le regole di associazione di una classe *pura* e una risorsa reale.
Richiamando il fatto che ogni classe *pura* è anche un campo del modello utente (par. 2.11), viene calcolata una distanza tra la risorsa ideale di una classe *pura*, indicata esplicitamente in una serie di regole di associazione (una colonna della tabella parzialmente riportata in fig. 2.15.1), e la risorsa reale.
La distanza calcolata, in maniera diversa dal paragrafo precedente, verrà trasformata in un valore numerico, con criteri scelti dal sistemista, che rappresenterà la regola di estensione.
Praticamente, quando verrà effettuata una operazione su quella risorsa nell'ambiente di sistema, il campo del modello utente facente riferimento alla classe *pura* verrà incrementato/decrementato del valore calcolato (fig. 2.15.4).

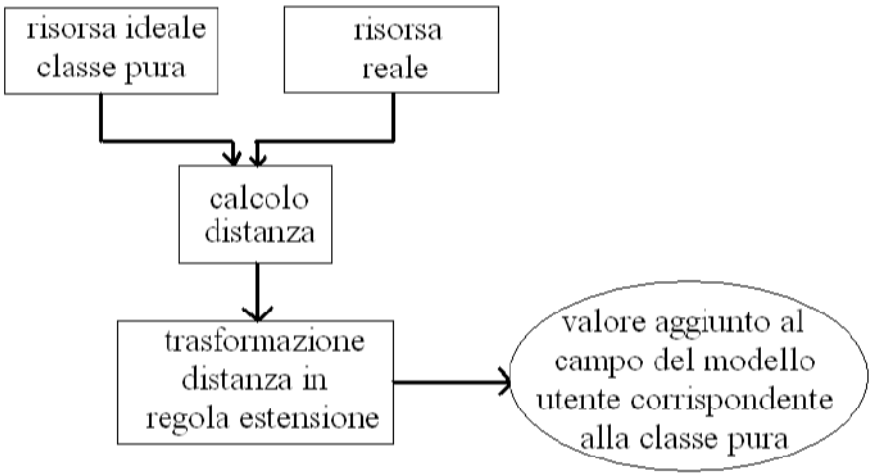


fig 2.15.4 Meccanismo di creazione delle regole di estensione

Facendo un esempio, se una risorsa reale "dista" 30 unità dalla risorsa ideale della classe *pura* "archeologo", si può stabilire, secondo parametri arbitrari, che alla selezione dell'utente della risorsa reale in questione, il valore del campo "archeologo" del modello utente attuale verrà incremento di 30/10=3 unità.
Ripetendo questa operazione, per ogni classe *pura* e per ogni risorsa reale, si riesce a creare l'intero insieme delle regole di estensione.

In PerfectFit06 il calcolo delle regole di estensione segue il concetto di proporzione presentato.
Ovviamente, per tarare al meglio il sistema, sono state fatte anche altre ipotesi che ora verranno illustrate.
Si consideri di calcolare le regole di estensione per una singola risorsa che, ovviamente, saranno pari al numero di campi del modello utente.
Ogni regola di estensione farà riferimento a una specifica classe *pura*, ovvero a un campo del modello utente(par. 2.11).
Per ogni campo del modello utente bisogna stabilire quanto questo verrà modificato dalla regola di estensione.
Si consideri l'insieme di regole di associazione individuate per una specifica classe *pura*.
Ogni regola di associazione farà riferimento a un campo del modello risorsa (par. 2.15.1), quindi l'insieme delle regole di associazione di una data classe pura sarà strutturalmente simile a un modello risorsa.
Per calcolare una regola di estensione di una risorsa rispetto a una classe *pura*, si consideri il modello della risorsa in questione.
Si andrà ora a stabilire una differenza tra il modello ideale della classe pura, descritto tramite le regole di associazione, e il modello della risorsa di riferimento.
Inizialmente la differenza sarà, ovviamente, nulla.
Si confrontino a uno a uno i campi del modello risorsa con i rispettivi campi delle regole di associazione.
Ogni qualvolta che il valore di un campo del modello risorsa è inferiore a quello dello stesso campo del modello associato alla classe pura, vuol dire che la risorsa ideale non soddisfa, per quel dato campo, gli interessi dell'utente.
La differenza finale sarà la somma di tutte le differenze, in modulo, dei campi che rientrano nelle condizioni descritte sopra.
Nel momento in cui si saranno effettuati tutti i confronti si saprà anche il numero di campi che non hanno soddisfatto l'utente puro.
Verrà poi calcolata la regola finale secondo la seguente formula:

$$\mathbf{RE = round(100-(diff/nc)/10)-3} \qquad \textit{(3)}$$

Dove:

- **RE** è la regola di estensione risultante.
- **Round()** è la funzione di arrotondamento.
- **Diff** è la differenza calcolata secondo il metodo su descritto.

Ripetendo lo stesso procedimento per ogni classe *pura* si ottengono tutte le regole di estensione per la risorsa.
Ripetendo il procedimento per ogni risorsa si calcola l'intero insieme delle regole di estensione.

2.16 Riepilogo attività di PerfectFit06

Al fine di avere il quadro completo della situazione è bene fare un riepilogo generale delle funzionalità di PerfectFit06 ora che si sono visti tutti gli elementi necessari.
Lo scopo principale di PerfectFit06 è quello di realizzare una selezione dei contenuti Web al fine di rendere agevole la navigazione per l'utente.
Per ottenere la capacità di gestire qualsiasi tipo di contenuto Web e poter adattare l'architettura del sistema a qualsiasi contesto, nel par 2.8 sono state introdotte le risorse che sono una astrazione dei contenuti web e rappresentano il nucleo atomico informativo per il sistema.
L'individuazione delle risorse è stata realizzata utilizzando un grafo della conoscenza (par. 2.5) opportunamente adattato alle esigenze dell'architettura di un sistema Web (par. 2.6).
Per descrivere semanticamente le risorse a disposizione del sistema si è fatto uso di un modello risorsa, associandone una istanza a ogni risorsa esistente (par 2.9).
Il modello risorsa è un insieme di coppie attributo-valore dove l'attributo rappresenta un interesse specifico rispetto al tipo di risorsa che si sta modellando e il valore stabilisce quanto è manifestato questo interesse.
L'intero ambiente Web in cui si muove l'utente è dunque composto dalle sole risorse. Ogni sua interazione col sistema avrà a che fare direttamente o indirettamente con una operazione su esse.
Per riuscire a gestire gli interessi dell'utente e, dunque, poter stabilire quanto una risorsa gli possa interessare o meno, è stato creato un modello utente(par 2.10).
Questo modello utente è formato da coppie attributo-valore dove ogni attributo rappresenta una classe *pura* (par 2.11) e ogni valore è un peso che stabilisce "quanto" l'utente appartiene a quella classe.
Quando si deve stabilire se una risorsa ideale può essere interessante o meno per l'utente viene creata una risorsa ideale per esso (par 2.15.3) in base a delle regole di associazione (par 2.15.1).
Si effettua quindi una stima della distanza tra la risorsa ideale calcolata e la risorsa reale. Il risultato di questa operazione di stima sarà utilizzato per stabilire se l'utente è interessato o meno alla risorsa reale.
Rimane da definire come viene gestito il modello utente che, ovviamente, deve essere modificato durante l'interazione col sito per modellarne i comportamenti

Un modello utente statico non avrebbe alcun senso.
Il sistema PerfectFit06 effettua il tracking dell'utente catturandone tutte le operazioni all'interno dell'ambiente Web.
Come detto, ogni interazione dell'utente avrà a che fare con una risorsa.
In PerfectFit06 a ogni azione su una risorsa viene associata una regola di estensione(par 2.15.3).
Le regole di estensione stabiliscono quanto, per ogni azione dell'utente, deve essere modificato il suo modello.
Ogni volta che viene modificato il modello utente, a sua volta varierà la corrispettiva risorsa ideale e al passo successivo di interazione i parametri per il calcolo delle risorse interessanti saranno aggiornati.

2.17 Tecniche per velocizzare l'adattività: modello utente normalizzato

Il problema principale dei sistemi adattativi è la velocità con cui realizzano la personalizzazione.
Per definire un modello utente sufficientemente descrittivo hanno, per forza di cose, bisogno di un certo numero di interazioni.
In PerfectFit06, per velocizzare l'adattività, si fa uso di una tecnica di normalizzazione del modello utente.
Praticamente, viene preso il campo del modello utente col valore più alto, indipendentemente dal valore, e viene usato come massimo relativo di riferimento.
Tutti gli altri campi vengono proporzionalmente normalizzati rispetto al nuovo massimo relativo.
Consideriamo questo esempio:

modello utente effettivo

Falegname:	10%
Elettricista:	0%
Idraulico:	25%
Carpentiere:	50%

Il massimo relativo di riferimento sarà il 50 del campo carpentiere, il modello normalizzato che si otterrà sarà:

Modello utente normalizzato

Falegname:	20% (10/50 * 100)
Elettricista:	0% (0/50 * 100)
Idraulico:	50% (25/50 * 100)
Carpentiere:	100% (50/50 * 100)

Tramite la normalizzazione del modello utente si riesce ad ottenere più velocemente un modello sufficientemente definito per realizzare l'adattività.
In PerfectFit06 il modello utente normalizzato viene creato, usato per i calcoli e distrutto, questo vuol dire che le regole di estensione saranno sempre e comunque applicate al modello utente attuale.

Così facendo, non si varierà il comportamento di base del sistema adattativo che, dopo un certo numero di operazioni, si comporterà a prescindere dalla normalizzazione.
Tramite le regole di estensione è possibile cambiare i valori del modello utente (par 2.15.3).
Queste regole sono associate ad azioni dell'utente sia che esso scelga una risorsa sia che esso la occulti, mostrando, in quest'ultimo caso, disinteresse per la suddetta risorsa.
Con l'utilizzo del modello utente normalizzato, se il modello di partenza non è sufficientemente definito, si rischia anche di incorrere più velocemente in un errore.
Il fatto che però le regole di estensione vengano apportate direttamente al modello utente, e non a quello normalizzato, permette di recuperare velocemente l'errore.
Infatti, poiché l'utente è, in qualche modo, spinto a interagire maggiormente con l'ambiente nella fase iniziale per occultare dei contenuti mostrati erroneamente, e quindi a delineare maggiormente il suo comportamento, aumenta considerevolmente la velocità con cui si arriva a un modello utente effettivo definito.
Come si vedrà nell'esempio di funzionamento pratico del paragrafo 3.6 sono sufficienti anche tre o quattro interazione per ottenere già dei buoni risultati.
Oltretutto, grazie alla normalizzazione, si riesce ad avere una maggiore velocità di adattatività anche nel caso che l'utente cominci a cambiare obiettivi, e quindi il suo comportamento, nell'ambiente Web.

CAPITOLO III

UTILIZZO DI PERFECTFIT06

Author: Vincenzo Scognamiglio

Version: 2.1 – printed: 23/06/2006

Keywords: PerfectFit06, scavi archeologici di Ercolano, appunti multimediali, adattatività, agenti web, gestione modello utente.

Capitolo III

Utilizzo di PerfectFit06

3.1 PerfectFit06 in pratica

Nel primo capitolo sono stati introdotti i concetti teorici che sono alla base dell'architettura di PerfectFit06.
Il secondo capitolo è stato dedicato alla descrizione dell'architettura di PerfectFit06, degli scopi, delle tecniche e delle innovazioni introdotte da questo sistema.
Questo capitolo invece tratterà il funzionamento pratico di PerfectFit06 in modo da mostrare la sua efficacia.
Il sistema PerfectFit06 è nato per essere utilizzato come sistema informativo su grandi quantità di dati.
L'applicazione immediata che si è trovata per questo sistema è sugli scavi archeologici di Ercolano.
Riguardo alla parte implementata, PerfectFit06 è un prototipo realizzato con lo scopo di verificare, anche nella pratica, l'efficacia delle idee presentate nel secondo capitolo.
PerfectFit06, pur essendo un prototipo, non è un sistema incompleto; semplicemente le funzionalità presentate nei capitoli precedenti sono state applicate su un insieme di informazioni ridotto, rispetto alla mole immensa di quelle disponibili per gli scavi di Ercolano, e sono state fatte alcune semplificazioni durante il settaggio dei valori dei modelli.
Tutte le tecniche e i meccanismi presentati nel secondo capitolo sono stati dunque implementati e testati dimostrando di funzionare, come verrà mostrato nel par 3.6, entro i limiti di parzialità imposti dalle esigenze.
La scelta delle classi '*pure*' per il modello utente (par 2.12) è stata realizzata in collaborazione con architetti e archeologi.
Non si ha la presunzione di ritenere questa scelta definitiva o completa ma semplicemente sufficiente agli scopi di sperimentazione del sistema.

Stesso discorso vale per i modelli risorsa (par 2.8) e per le regole di associazione (par. 2.15.1) che sono state comunque stabilite con attenzione e dopo aver analizzato fino in fondo il problema.
L'intero lavoro di elaborazione e rielaborazione dei contenuti, analisi comportamentali dell'utente e definizione delle relative regole, avrebbe richiesto un team di lavoro composto da numerose e varie figure professionali nonché un tempo di lavoro sicuramente sproporzionato per gli scopi immediati del progetto.
D'altra parte, con PerfectFit06, si ha l'ambizione di creare un sistema web piuttosto innovativo sia dal punto di vista della sua architettura che dal punto di vista delle tecnologie adattative utilizzate.
Nell'ottica di future evoluzioni del sistema, è stata incentivata la progettazione di un sistema estremamente adattabile anche dal punto di vista strutturale, tant'è vero che cambiando il contenuto della tabella contenente il grafo della conoscenza o quella contenente le definizioni delle classi *pure* si può dire tranquillamente che l'intero sistema si adatta alle modifiche, rigenerando addirittura delle sue componenti (par 2.13).
In base alle osservazioni appena fatte, un primo risultato del sistema è quello di poter considerare la sua architettura indipendente dal contesto di applicazione.

3.2 L'interfaccia di PerfectFit06

L'attuale interfaccia utente di PerfectFit06 è stata realizzata in maniera tale che sia chiaro l'effettivo funzionamento del sistema, saranno dunque presenti sia voci destinate all'utente finale, che altre destinate invece a esibire lo stato del sistema.
Da una pagina web tipo di PerfectFit06 (fig. 3.2.1) è possibile accedere a un vero e proprio pannello di comando che fornisce informazioni sullo stato del modello utente attuale, delle regole di associazione, del modello utente normalizzato e della risorsa ideale calcolata rispetto al modello utente normalizzato (paragrafi 2.10-2.15).
Queste informazioni saranno ovviamente trasparenti all'utente nella versione definitiva dell'interfaccia ma sono state ritenute indispensabili per esibire l'effettivo funzionamento del sistema in questo capitolo.
La trattazione continuerà, nell'immediato, illustrando l'interfaccia di una pagina tipo di PerfectFit06 (fig. 3.2.1).

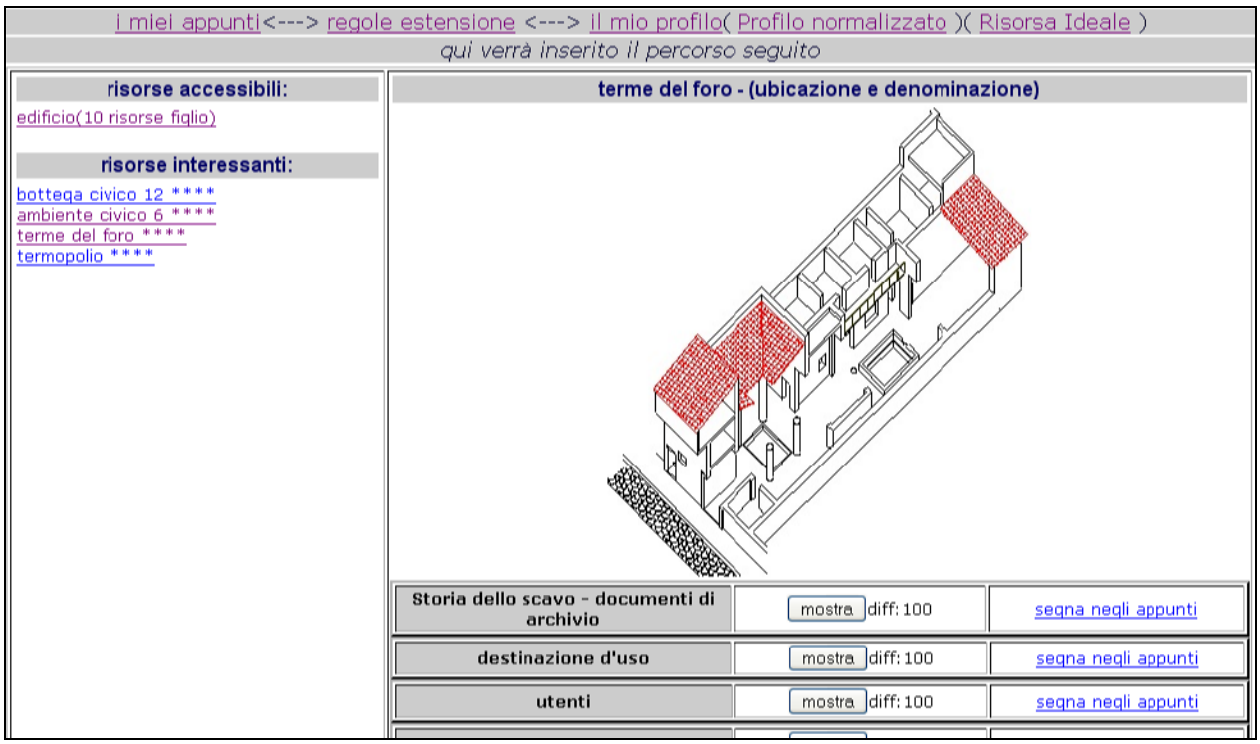


fig 3.2.1 – pagina web – tipo di PerfectFit06

Una pagina tipo di PerfectFit06 fa riferimento a una specifica risorsa che, nel grafo adattato descritto nel par 2.6, è un nodo principale secondo i criteri descritti nello stesso paragrafo.

Si sottolinea che PerfectFit06 genera dinamicamente queste pagine in base al contenuto del grafo, una variazione di quest'ultimo porta all'immediata ridefinizione dei nodi web del sistema.

All'interno di una pagina tipo di PerfectFit06, saranno presenti:

- una barra superiore (par 3.2.1)
- una colonna con le risorse accessibili (par 3.2.2)
- una colonna delle risorse (par 3.2.3)

3.2.1 La barra superiore

Nella barra superiore (fig 3.2.2) sono accessibili le seguenti voci:

- I miei appunti
- Regole estensione
- Il mio profilo
- Profilo normalizzato
- Risorsa ideale



fig. 3.2.2 barra degli strumenti di PefectFit06

La voce "I miei appunti" sarà è principalmente destinata all'utente, mentre le altre voci sono utili per l'analisi del sistema.

La voce "I miei appunti " permette all'utente di accedere alla pagina contenente gli appunti raccolti durante la sua interazione, questa funzionalità verrà trattata in maniera più approfondita nel par 3.3.

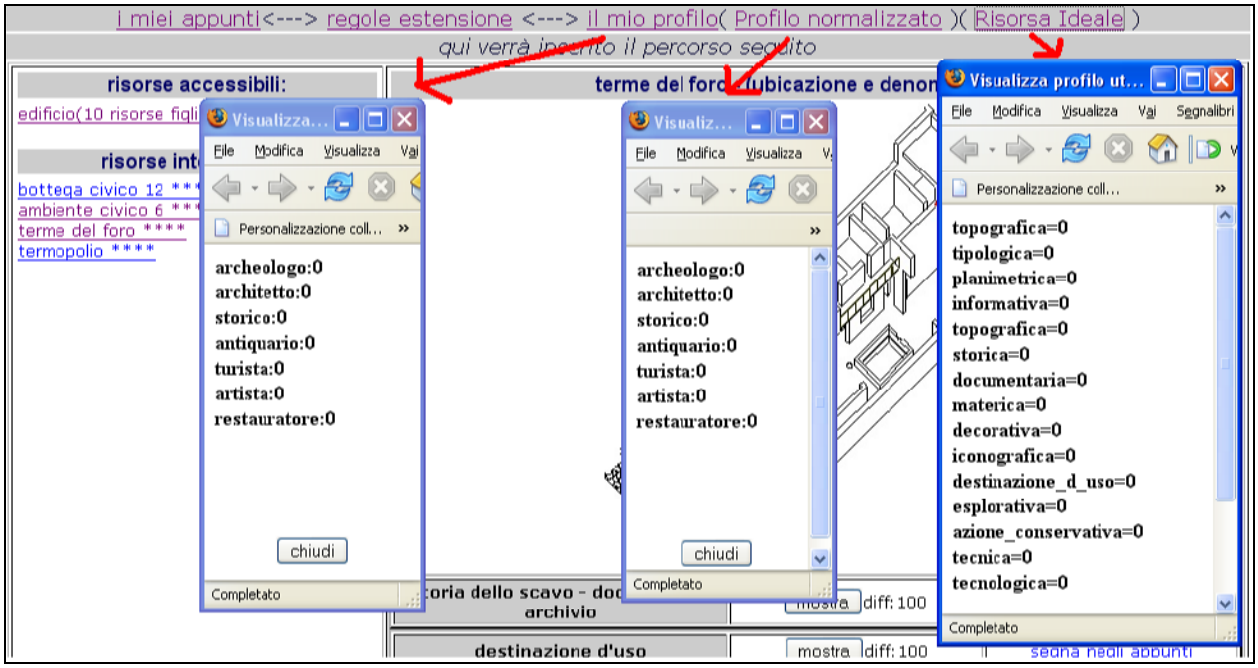


fig. 3.2.3 – visualizzazione modello utente attuale, modello utente attuale normalizzato e risorsa ideale

La voce "regole di estensione" apre una nuova pagina web dove vengono mostrate le regole di estensione calcolate dinamicamente come descritto nel par 2.15.2, nel par 3.6 verrà invece illustrato come utilizzare il contenuto di questa pagina per verificare il funzionamento del sistema.

La voce “Il mio profilo” carica una nuova pagina contenente il modello utente attuale con i valori aggiornati.
La voce “Profilo normalizzato” mostrerà il modello utente calcolato secondo il par 2.17.
La voce “Risorsa Ideale” aprirà una pagina con il modello della risorsa ideale rispetto al modello utente attuale, calcolata in base ai parametri descritti nel par 2.15.3.
In figura 3.2.3 è possibile vedere il risultato dell’apertura dei link “il mio profilo”, “profilo normalizzato” e “risorsa ideale”; la pagina presentata può essere considerato il modo più efficace di valutare lo stato del sistema in funzione dei suoi modelli.

3.2.2 La colonna “Risorse accessibili”

Ogni pagina di PerfectFit06 fa riferimento a una specifica risorsa, il suo contenuto informativo sarà costituito dalle informazioni legate alla risorsa di pagina e dalle rispettive risorse figlio.



fig 3.2.4 – la colonna delle risorse accessibili

Nel par 2.6 si è illustrato come il sistema realizza, in funzione del numero di risorse figlio, una divisione dei nodi del grafo in nodi principali e nodi secondari per il web.



fig. 3.2.5 – apertura di un nodo principale dalla colonna delle risorse accessibili

Un nodo principale avrà almeno tre risorse figlio, ai fini dell'usabilità della pagina il contenuto informativo di un nodo principale non sarà mostrato all'interno della stessa pagina per non creare un sovraccarico cognitivo per l'utente[Visciola-1].
L'accesso ai contenuti informativi della risorsa-nodo principale sarà reso disponibile mediante un link a una nuova pagina creata ad hoc e dinamicamente dal sistema.
La colonna delle "Risorse accessibili" (fig. 3.2.4) è appunto la lista dei link alle pagine dei nodi principali, figli della risorsa di pagina.
Se si clicca su un link principale il contenuto della pagina corrente diventerà quello della risorsa cui fa riferimento il link al nodo principale (fig. 3.2.5).

3.2.3 Colonna risorse

La colonna destra della pagina web di PerfectFit06 è quella composta effettivamente dalle risorse associate al nodo web che si sta esplorando.
Sempre visibile nella pagina sarà il contenuto della risorsa principale di pagina (fig. 3.2.6).

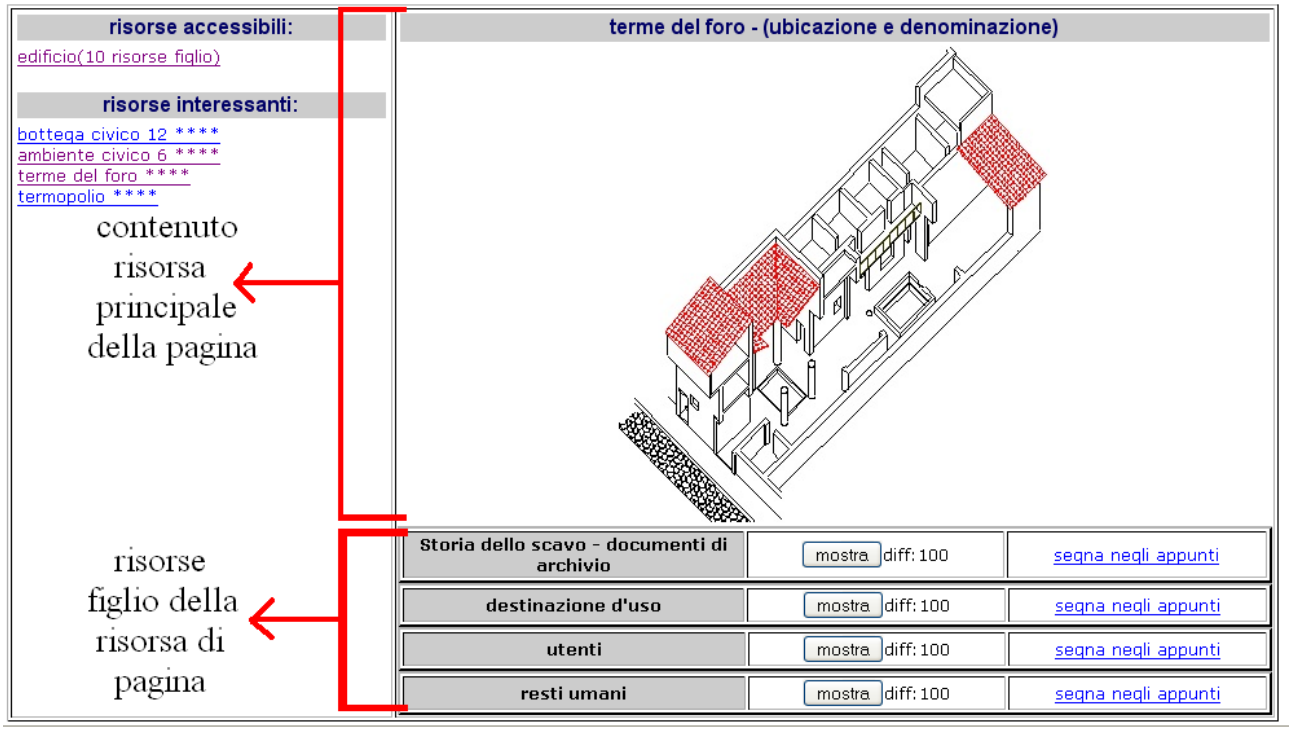


fig 3.2.6 – contenuto di una pagina web di PerfectFit06

Dopo il contenuto della risorsa principale ci sarà un elenco delle risorse, figlie della risorsa principale, che sono state classificate come nodi secondari (par 2.6).
Per ognuna di queste risorse è accessibile, al suo fianco, una barra degli strumenti (fig 3.2.7).



fig 3.2.7 – barra degli strumenti della risorsa

L'elemento più importante per ogni risorsa secondaria è il tasto "mostra" a cui, come verrà mostrato nel par 3.4.2, è abbinata l'applicazione delle regole di estensione.
La voce "diff:" indica la distanza tra quella particolare risorsa e la risorsa ideale al momento del caricamento della pagina, calcolata secondo la formula (2) e i criteri presentati nel par 2.15.3.
D'ora in avanti si riferirà alla distanza tra risorsa ideale e risorsa reale semplicemente col termine 'distanza'.
La voce "segna negli appunti" permette di inserire la risorsa specifica negli appunti, funzionalità descritta nel par 3.3.
La pressione del tasto "mostra" permette di visualizzare il contenuto associato alla risorsa (fig 3.2.8).
Alla sua attivazione si trasforma poi nel tasto "nascondi" che ha, ovviamente, la funzione inversa.

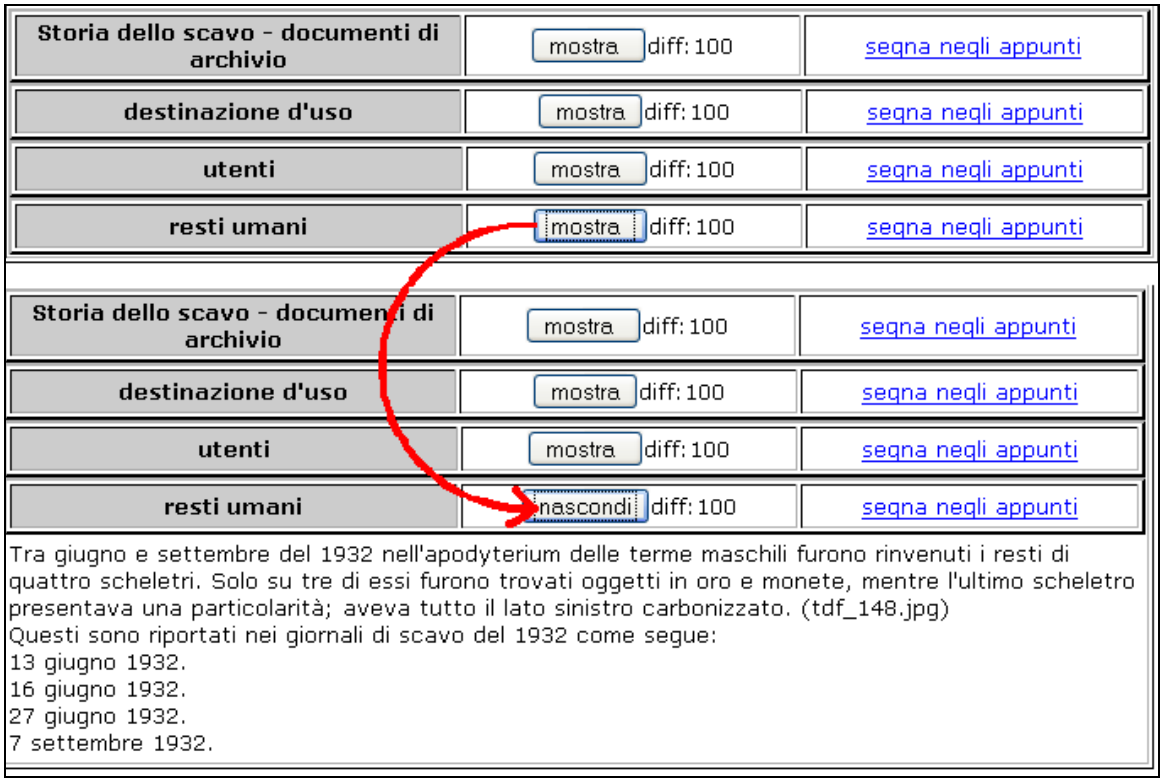


fig 3.2.8 – effetto del tasto "mostra/nascondi"

3.3 La raccolta degli appunti

L'architettura di PerfectFit06 è pensata per essere utilizzata con un sistema informativo composto da numerosi dati.
Qualunque utente acceda a un sistema di questo tipo, con alta probabilità, sentirà l'esigenza di raccogliere degli appunti.
Questa funzionalità di raccolta degli appunti, anche se piuttosto importante, viene spesso tralasciata dai sistemi web.
PerfectFit06 mette a disposizione dell'utente un semplice ed efficace meccanismo per la creazione di appunti personalizzati.

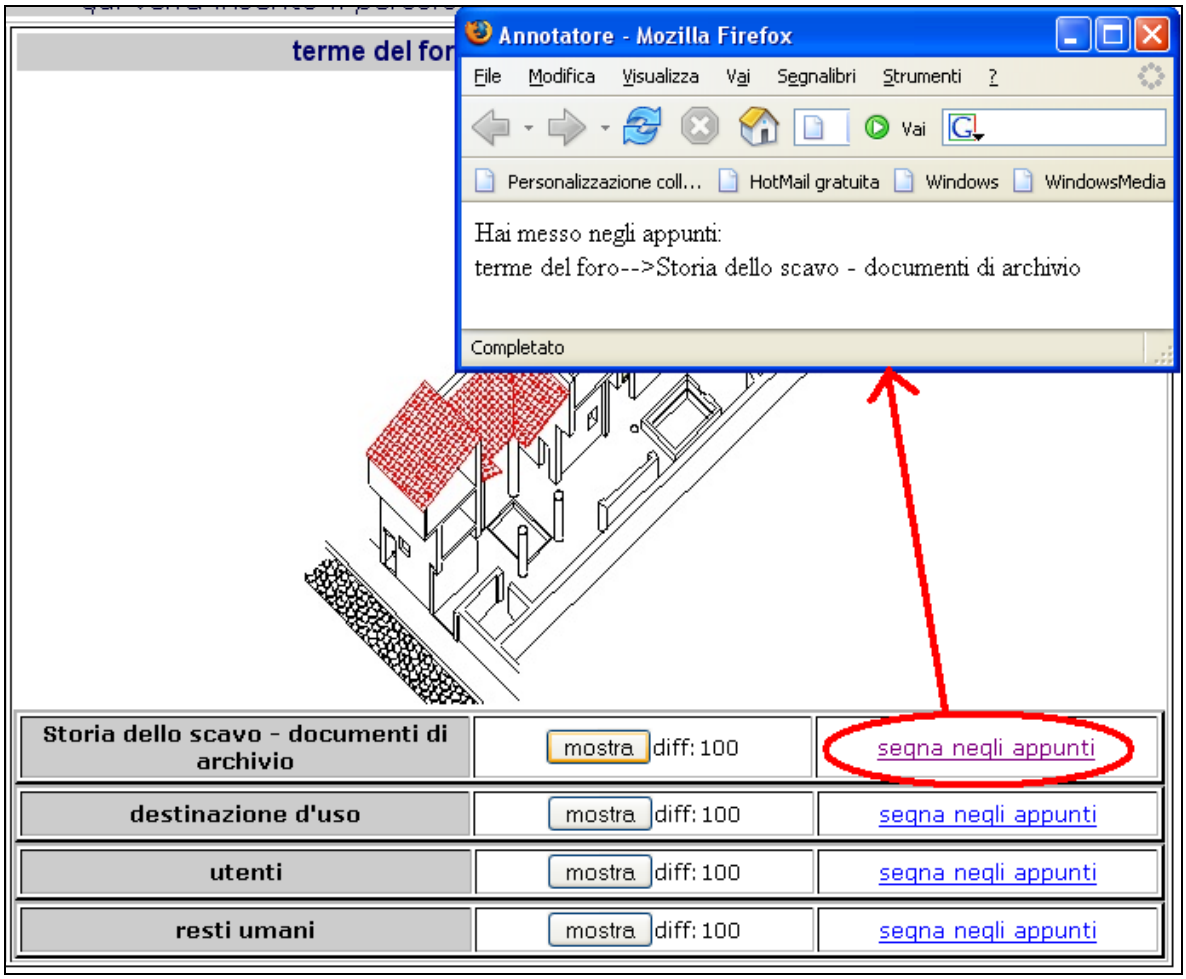


fig 3.3.1 – inserimento di una risorsa negli appunti e messaggio dell'annotatore

Come descritto nel paragrafo 2.8, l'informazione atomica gestita da PerfectFit06 è la singola risorsa, il sistema considera gli appunti come un insieme di risorse composto in funzione di selezioni esplicite dell'utente.

Accanto a ogni risorsa in una data pagina verrà inserita la voce “segna negli appunti” che permetterà all’utente di inserire quella risorsa nei suoi appunti personali.
Una volta inserita questa risorsa negli appunti, verrà mostrato un messaggio di notifica per avvertire dell’avvenuto inserimento(fig 3.3.1).
La stessa operazione si può fare per tutte le risorse per cui è disponibile la voce “segna negli appunti” (fig 3.3.2).

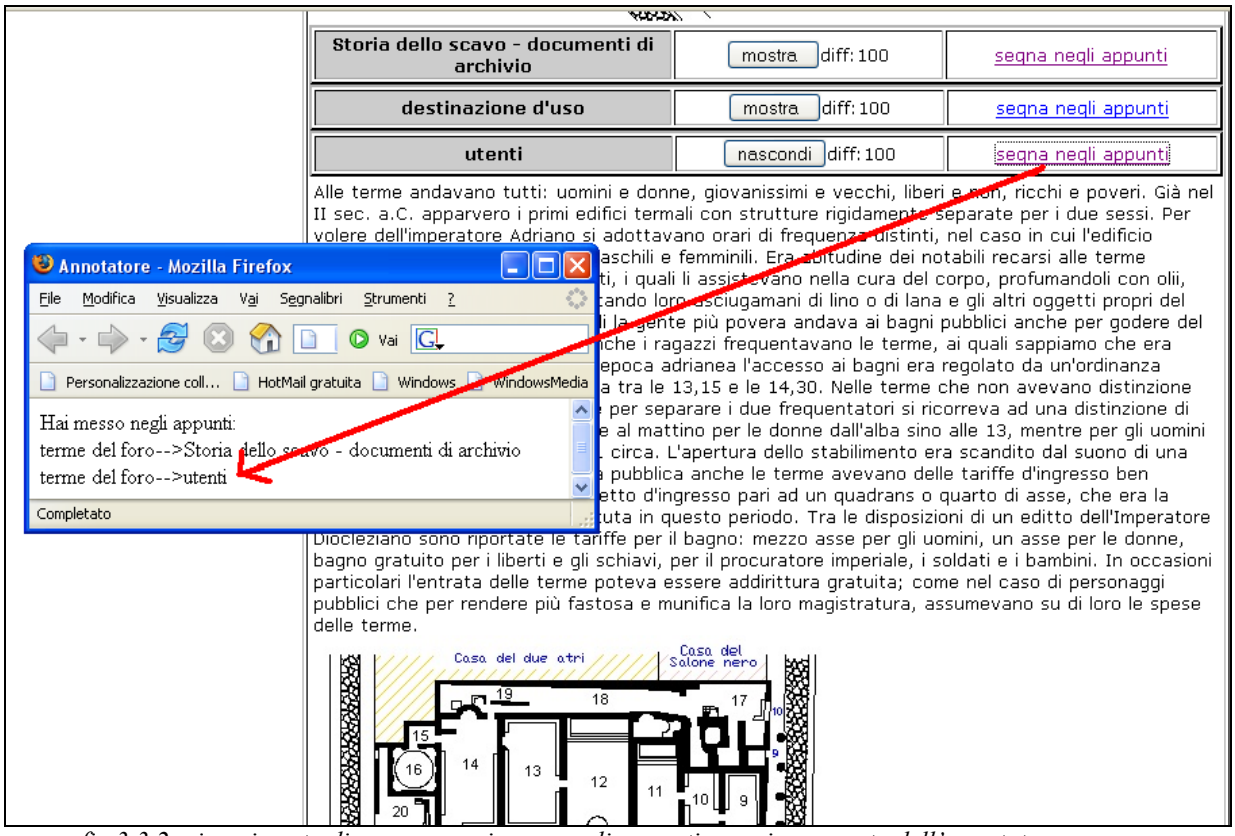


fig 3.3.2 – inserimento di una nuova risorsa negli appunti e aggiornamento dell’annotatore

Oltre alla raccolta appunti viene messa a disposizione dell’utente, in qualsiasi momento, anche la possibilità di visionare la selezione effettuata.
L’insieme delle risorse inserite dall’utente negli appunti viene utilizzata da PerfectFito06 per generare dinamicamente una pagina web che rappresenterà così l’insieme degli appunti dell’utente.
Per accedere ai suoi appunti basterà che l’utente selezioni la voce “i miei appunti” nella barra superiore (fig 3.3.3).

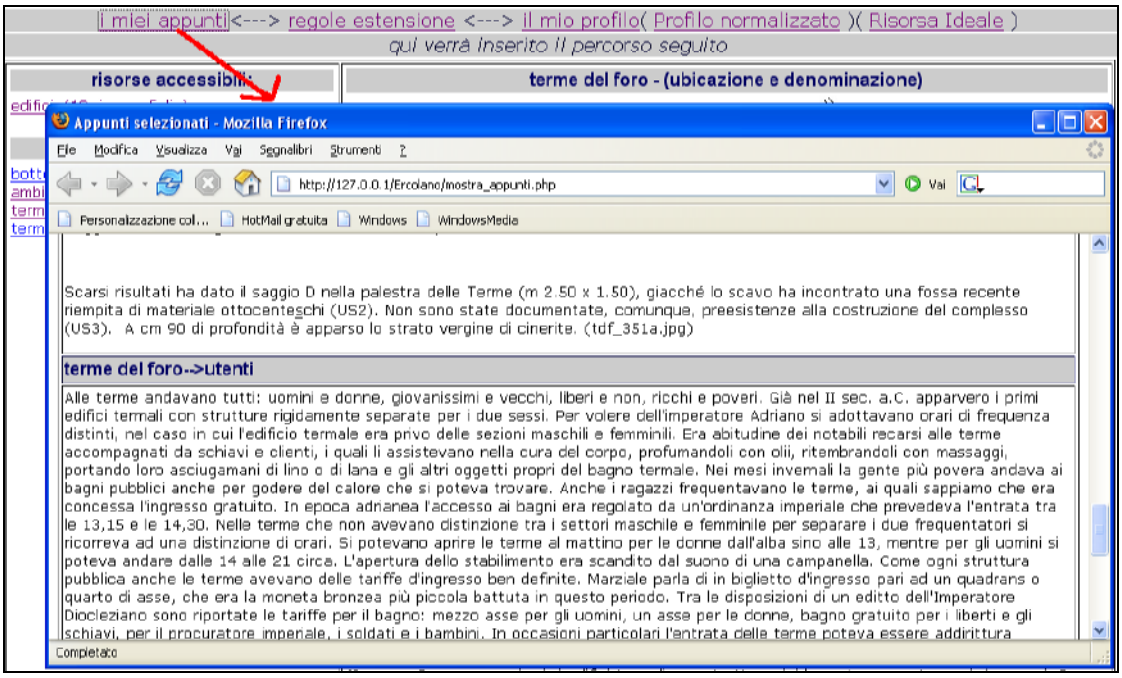


fig 3.3.3 – visualizzazione degli appunti raccolti

A questo punto, l'utente, potrà scegliere di chiudere la pagina degli appunti o salvarla (fig 3.3.4) sul proprio computer.

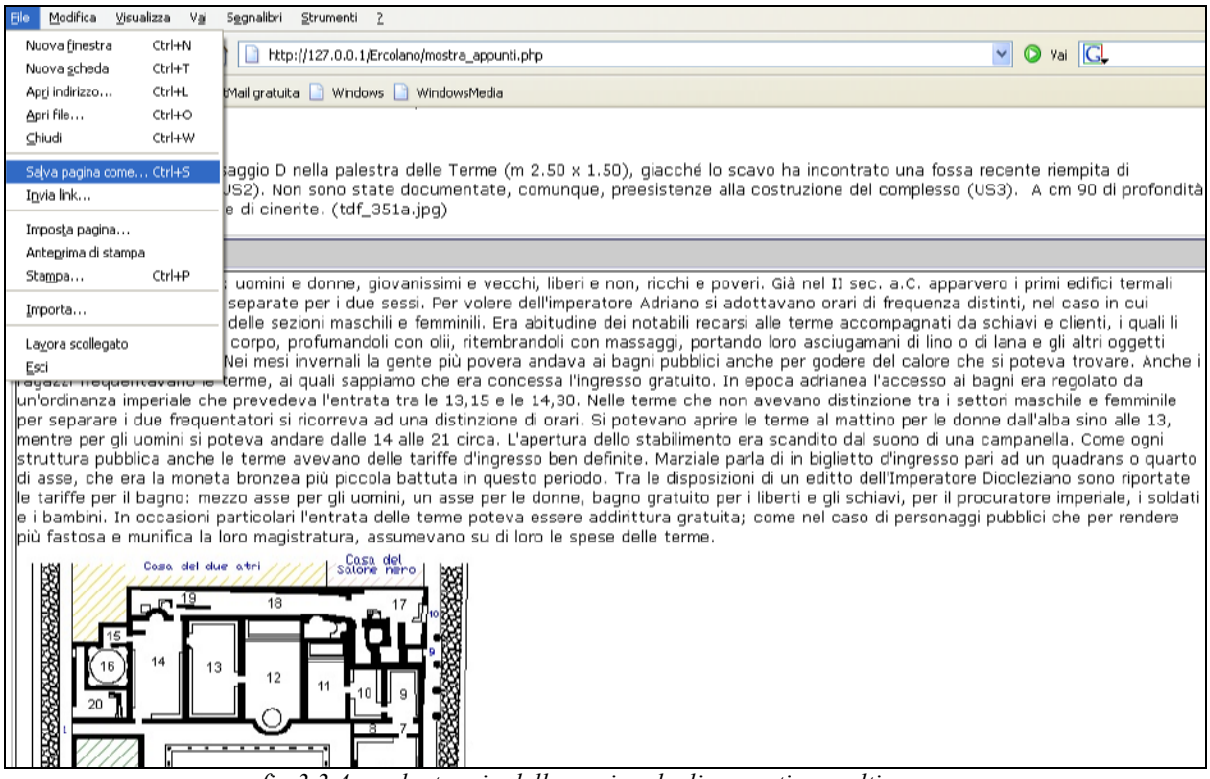


fig 3.3.4 – salvataggio della pagina degli appunti raccolti

La generazione della pagina degli appunti è dinamica quindi, se l'utente la chiude e continua la navigazione del sito, aggiungendo altre risorse negli appunti personali, ogni volta che aprirà nuovamente la pagina delle risorse raccolte, tramite "i miei appunti", verrà mostrato l'insieme aggiornato delle risorse selezionate.

3.4 Le regole di estensione

Le regole di estensione (par 2.15.2) servono per stabilire quantitativamente le modifiche da apportare al modello utente quando viene selezionata una specifica risorsa.
Per come è stata progettata l'interfaccia di PerfectFit06, l'utente può scegliere le risorse principalmente in due modi: cioè tramite il tasto mostra e tramite l'aggiunta della risorsa negli appunti.
E' naturale partire dalla considerazione che l'aggiunta di una risorsa negli appunti deve avere un
peso maggiore, a livello di regole di estensione, rispetto alla semplice visualizzazione della stessa.
In PerfectFit06 le regole di estensione vengono calcolate per essere utilizzate con il tasto "mostra/nascondi", per quanto riguarda la memorizzazione degli appunti, la stessa regola utilizzata per il "mostra/nascondi" di quella risorsa, viene moltiplicata per un fattore positivo che, nell'attuale implementazione di PerfectFit06, è 10.

3.4.1 Tabella delle regole di estensione

Le regole di estensione vengono generate dinamicamente ogni volta che vengono variate le regole di associazione o il grafo delle risorse (par 2.15.2, 2.15.4).
In figura 3.4.1 è mostrato un estratto delle regole di estensione attualmente calcolate da PerfectFit06 in base al contenuto delle sue tabelle, l'intera tabella è in appendice.
La lettura e l'utilizzo di questa tabella è particolarmente importante per verificare il comportamento del sistema.
La prima cosa da precisare è che le quantità indicate nella tabella sono i valori che vengono aggiunti (risp. sottratti) al modello utente attuale quando il navigatore sceglie di mostrare (risp. nascondere) una risorsa.
Particolarmente interessante è confrontare i valori ottenuti per le regole di estensione di una risorsa con i valori del suo modello e quelli delle regole di associazione.

nome risorsa	archeologo	architetto	storico	antiquario	turista	artista	restauratore
insulae	0	0	0	0	0	0	0
ubicazione e denominazione	0	0	0	0	0	0	0
Storia dello scavo - documenti di archivio	1	0	5	0	0	0	0
destinazione d'uso	0	0	0	0	6	0	0
edificio	0	0	0	0	1	0	0
utenti	0	0	0	0	3	0	0
resti umani	0	0	2	0	0	0	0
planimetria - storici - attuali	0	0	0	0	0	0	0
datazione e fasi costruttive	0	0	4	0	2	0	0
elementi costruttivi	0	0	0	2	1	0	0

fig 3.4.1 – estratto delle regole di estensione

Si prenda in considerazione la risorsa "datazione e fasi costruttive" che, come è evidenziato in figura 3.4.1, modifica di quattro unità il campo "storico" e di due quello "turista" nel modello utente attuale.
Praticamente questa risorsa interesserà di un fattore due un turista e di un fattore quattro uno storico.

Campo	Valore
topografica	0
tipologica	0
planimetrica	0
informativa	50
storica	100
documentaria	100
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	40
esplorativa	50
azione_conservativa	0
tecnica	0
tecnologica	0

fig 3.4.2 – modello risorsa di "datazione e fasi costruttive"

Si guardi ora il modello associato alla risorsa in questione (fig 3.4.2) e successivamente alle risorse ideali, ovvero alle regole di estensione, della stessa risorsa (fig 3.4.3).

topografica	tipologica	planimetrica	informativa	storica	documentaria	materica	decorativa	iconografica	profilo_utente_riferimento
90	100	0	0	90	0	0	0	0	archeologo
0	0	80	0	0	0	0	0	0	architetto
0	0	0	0	100	100	0	0	0	storico
0	100	0	0	60	0	100	0	0	antiquario
70	0	0	100	0	0	0	0	0	turista
0	0	0	0	0	0	0	100	100	artista
0	0	0	0	0	0	100	80	0	restauratore

destinazione_d_uso	esplorativa	azione_conservativa	tecnica	tecnologica	profilo_utente_riferimento
0	0	0	100	0	archeologo
0	0	100	100	90	architetto
0	80	0	0	0	storico
100	0	0	0	0	antiquario
80	0	0	0	0	turista
0	0	0	0	0	artista
0	0	100	0	0	restauratore

fig. 3.4.3 – regole di associazione

Si riesce a vedere come, in base al metodo descritto nel paragrafo 2.15.4, la regola di estensione calcolata individua la tipologia di comportamento descritto tramite le regole di associazione e il modello della risorsa.
Infatti confrontando le regole di associazione, ovvero il modello ideale, dello storico col modello della risorsa “datazioni e fasi costruttive” e ripetendo l’operazione per il turista, ci si accorge anche senza fare i calcoli della “somiglianza” tra i modelli.
E’ ovvio che l’interesse principale dell’architettura di PerfectFit06 è riuscire a generare correttamente queste regole, così come l’esempio dimostra che avviene.
L’essere d’accordo o meno coi valori specifici della rappresentazione è tutt’altro discorso.
Nel caso ci si accorga di aver realizzato male la modellizzazione o aver stabilito erroneamente delle regole di associazione, basta cambiarle e il sistema si adatterà dinamicamente a esse.

3.4.2 Applicazione delle regole di estensione

Le regole di estensione, parzialmente presentate nella figura 3.4.1, vengono applicate alla singola risorsa mediante il tasto “mostra/nascondi”.
Questo vuol dire che i valori delle regole di estensione rappresenteranno esattamente la modifica subita dal modello utente se viene premuto il tasto “mostra” associato alla risorsa.
Si consideri la risorsa “datazione e fasi costruttive” la cui relativa regola di estensione è evidenziata in figura 3.4.1.
Si parta dalla situazione di base del sistema (fig 3.4.4) , col modello utente inizializzato come descritto nel par 2.11, ovvero con tutti i valori azzerati.

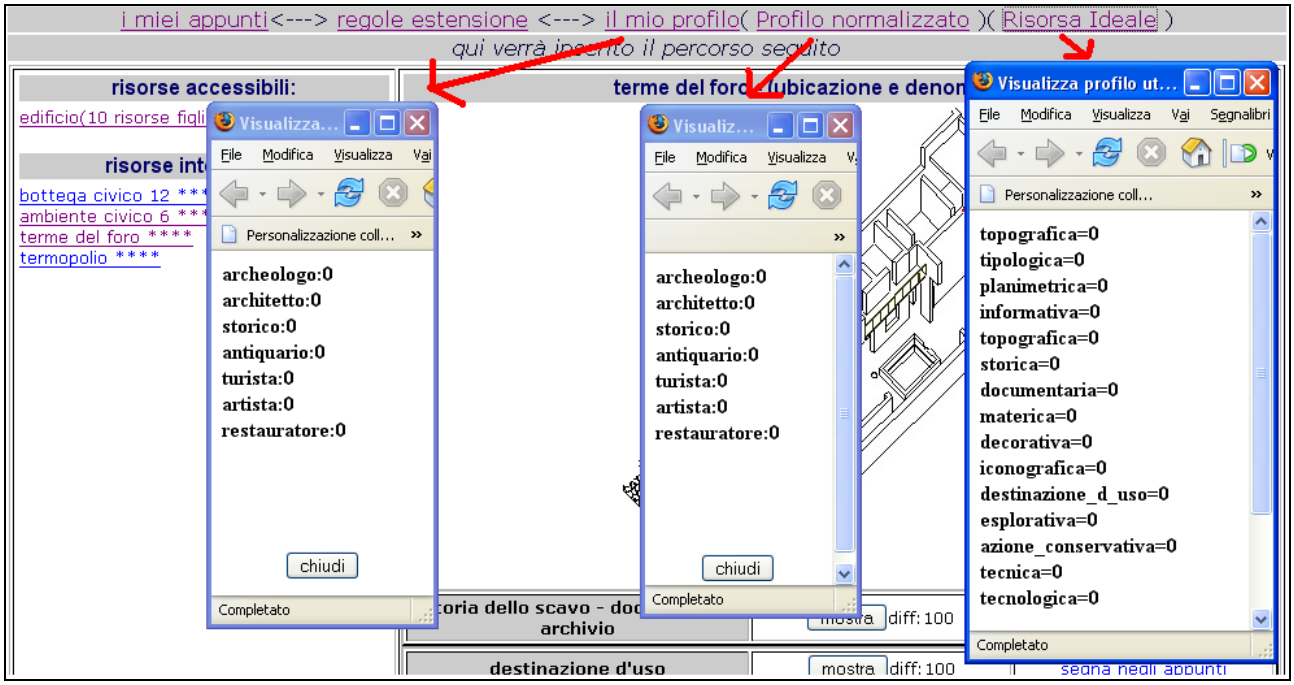


fig 3.4.4 – stato iniziale dei modelli di PerfectFit06

Alla pressione del tasto “mostra” associato alla risorsa "datazione e fasi costruttive”, oltre a comparire il contenuto della risorsa in modo simile a quello in figura 3.2.8 verrà modificato il modello utente secondo la regola associata alla risorsa (fig. 3.4.1).

In figura 3.4.5 è mostrato il nuovo stato delle cose dopo la visualizzazione del contenuto della risorsa.

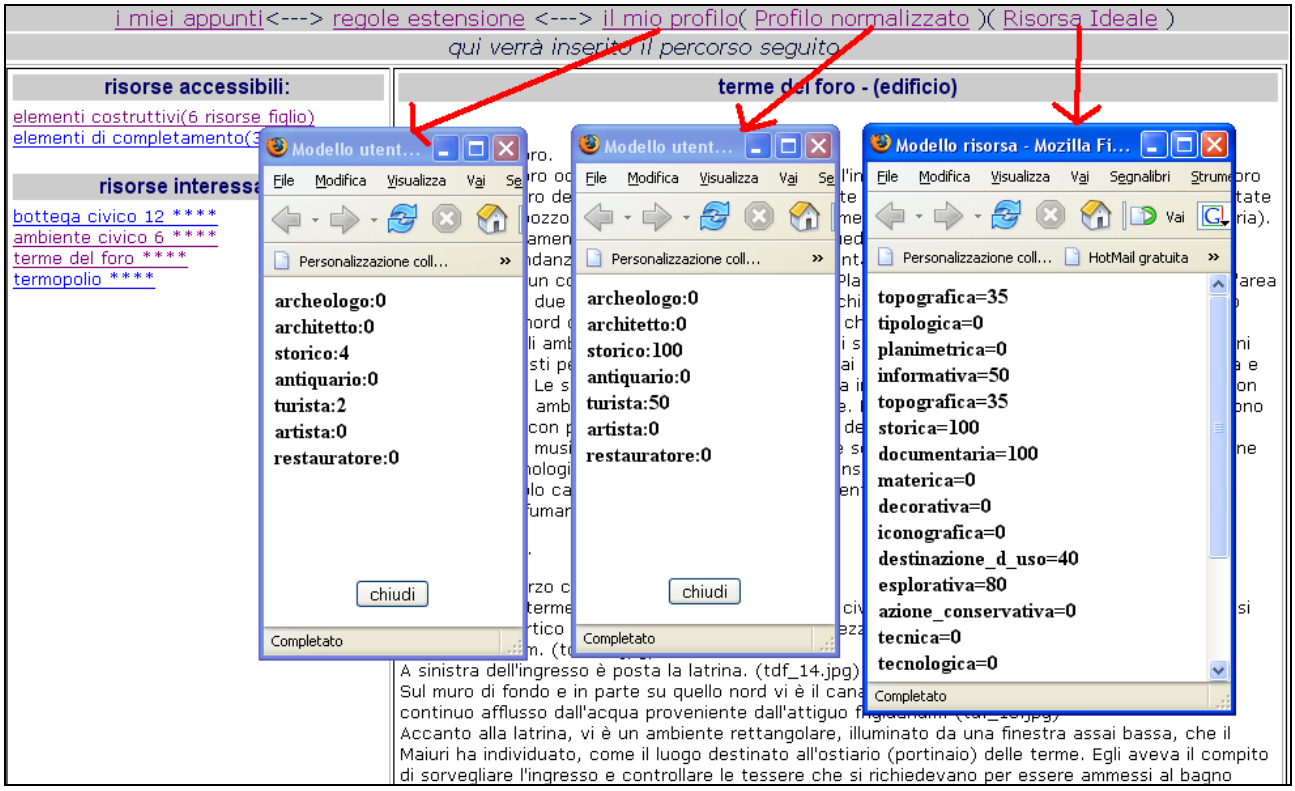


fig 3.4.5 – stato di PerfectFit06 dopo la scelta della risorsa "datazione e fasi costruttive"

Dall'esempio, analizzando le tabelle di regole e i modelli utente delle classi *pure*, si incomincia a delineare piuttosto nettamente il comportamento del sistema e già si inizia a chiudere il cerchio della modellizzazione dell'utente attuata.

Come già detto, una volta cliccato, il tasto "mostra" permette di visualizzare il contenuto collegato a una data risorsa e si trasforma nel tasto "nascondi" che ha esattamente la funzione opposta.

Se viene cliccato il tasto "nascondi" e, quindi, si annulla sostanzialmente l'azione precedente di visualizzazione, viene applicata la stessa regola di estensione del tasto "mostra" ma a segni invertiti.

Se non ci sono state altre interazioni premendo prima "mostra" e poi "nascondi" si ritorna allo stato di partenza di fig 3.4.4.

Altro tipo di selezione della risorsa possibile è quello realizzabile tramite "segna negli appunti" di cui si è parlato nel paragrafo 3.3.

L'aggiunta della risorsa negli appunti attiva la stessa regola di estensione del comando "mostra" moltiplicata per un fattore positivo, nell'attuale implementazione questo fattore è 10.

Se la risorsa "datazione e fasi costruttive" viene segnata negli appunti, partendo dallo stato iniziale, lo stato del sistema diventa quello mostrato in figura 3.4.6.

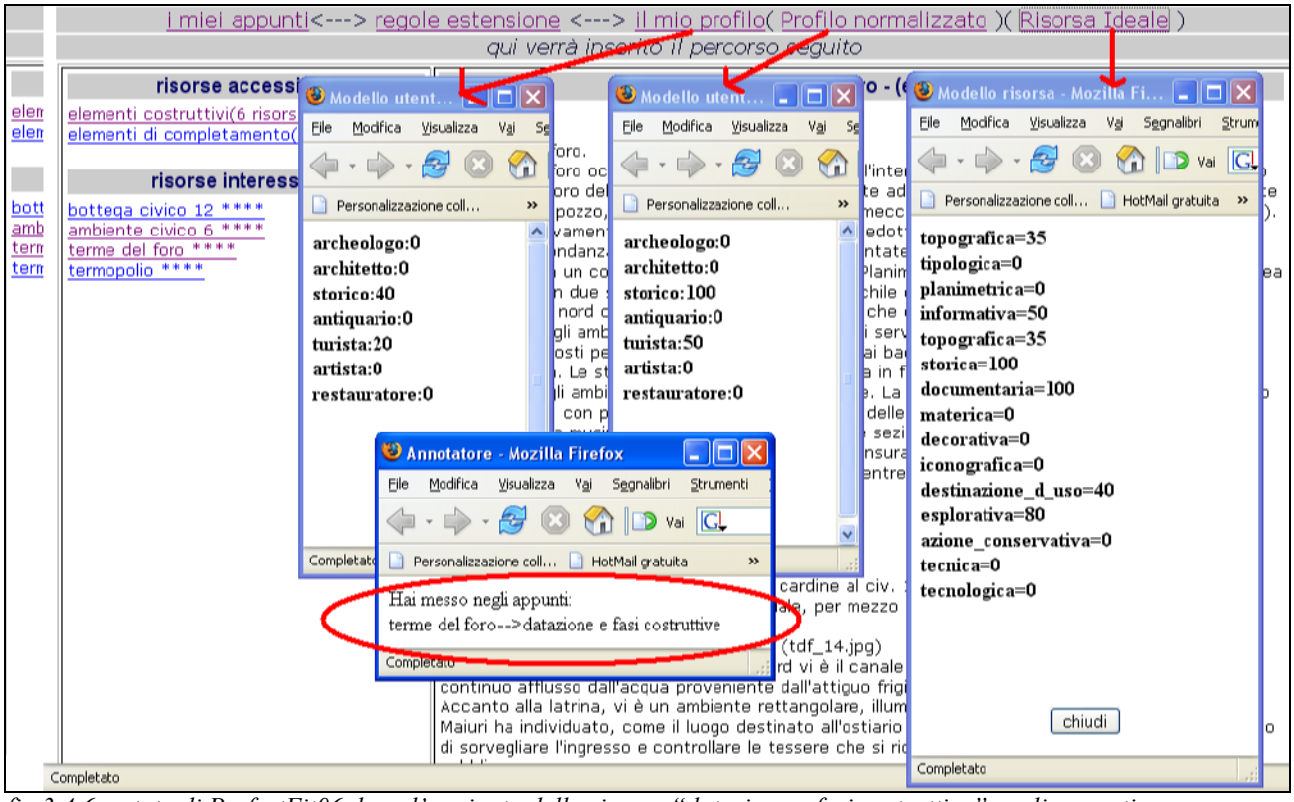


fig 3.4.6 – stato di PerfectFit06 dopo l'aggiunta della risorsa "datazione e fasi costruttive" negli appunti

Si noti come, per effetto della normalizzazione, ciò non comporta alcuna modifica sul modello utente normalizzato e sulla risorsa ideale rispetto alla semplice pressione del tasto "mostra" (fig 3.4.5).

Questo comportamento del sistema è osservabile se si eseguono tutte operazioni solo di "mostra" o solo di "segna negli appunti" poiché la normalizzazione, unicamente in questi casi, annulla il fattore 10 sulle regole di estensione.

3.5 L'adattatività di PerfectFit06

Finora si è presentato il sistema PerfectFit06 mostrando come esso effettua il tracking dell'utente e applica la modellizzazione secondo le tecniche presentate nel secondo capitolo.
Resta ancora da vedere a quali risultati porta questa modellizzazione.
La creazione di un grafo della conoscenza, l'astrazione delle risorse, l'identificazione di nodi principali e nodi secondari sono tutte tecniche utilizzate per permettere la riduzione di un grosso bacino informativo in nuclei più piccoli.
Tramite la gestione di un modello utente, si è mostrato che si può stabilire il grado di interesse dell'utente modellato rispetto ad una specifica risorsa.
Una pagina web di PerfectFit06 è composta da risorse a cui è associato un tasto "mostra/nascondi" di cui si è già discusso ampiamente e finora si è visto sempre questo tasto inizialmente impostato sul valore "mostra".

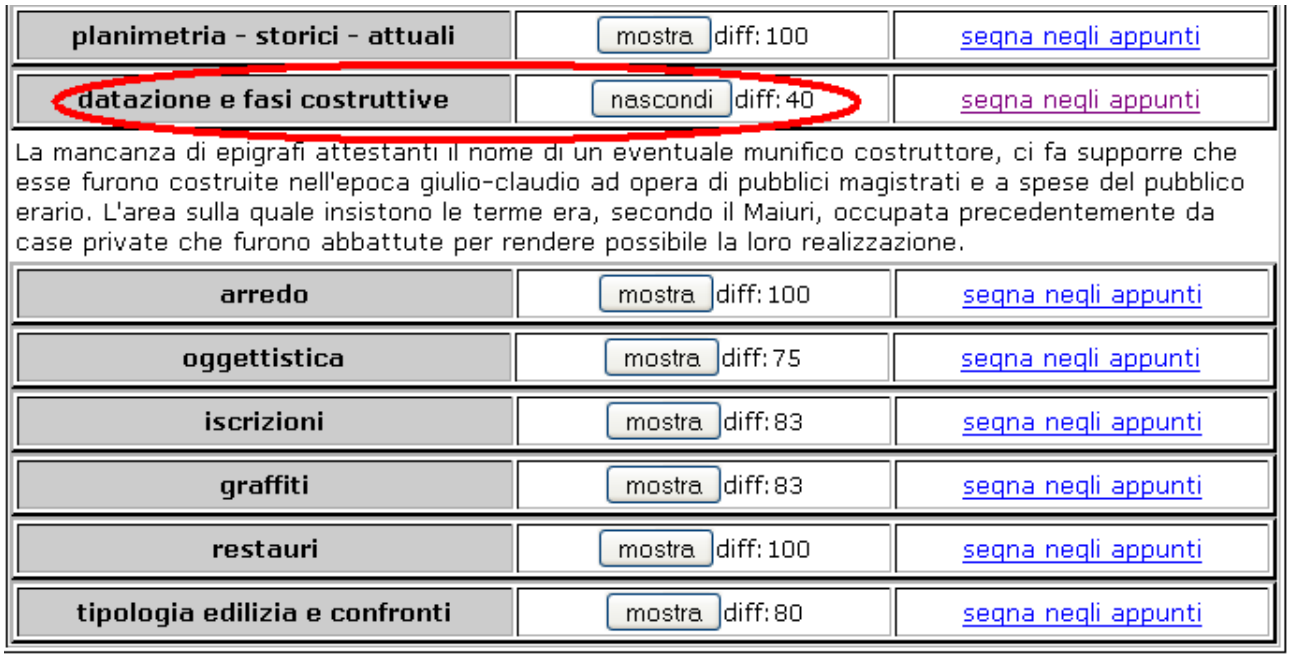


fig 3.5.1 – nuovo valore della distanza della risorsa "datazione e fasi costruttive"

Primo e più visibile effetto dell'adattività messa in atto da PerfectFit06 è proprio il fatto che, raggiunta una definizione accettabile del modello utente normalizzato, le risorse che il sistema ritiene "interessanti" per l'utente saranno automaticamente visualizzate.
Per definire il grado di interesse dell'utente per una data risorsa si usano le tecniche descritte nel par 2.15.3, il risultato di questa operazione sarà una distanza visibile, come

descritto nel par 3.2.3, nella pagina web all'interno della barra accanto al nome della risorsa.
In fig. 3.5.1 si vede come la risorsa "datazione e fasi costruttive" abbia raggiunto una distanza (diff:40) sufficiente a farla considerare interessante dal sistema e, quindi, ne sia stata visualizzato direttamente il contenuto informativo al momento del caricamento della pagina.

3.6 Esempio pratico di funzionamento

Quello che si vuole ora mostrare è come PerfectFit06 sia capace di modellare l'utente e di capire effettivamente quali sono i suoi interessi.
Per testare questa capacità è conveniente immedesimarsi in una tipologia d'utente e vedere se il sistema risponde nel modo atteso alle aspettative.
Immedesimarsi in un utente significa, sostanzialmente, realizzare una interazione col sistema simile a quella che effettuerebbe lui.
Il primo passo da fare per realizzare questa immedesimazione è vedere, regole di associazione e modelli risorsa alla mano, quali risorse lui andrebbe a visualizzare e a segnare negli appunti.
Questo passaggio è importante per l'immedesimazione al fine di garantire la coerenza con l'interpretazione delle classi *pure* effettuate dal progettista e dai suoi collaboratori.
Che poi queste scelte siano sbagliate o meno non è determinante ai fini del funzionamento ma a quello della qualità dell'adattatività realizzata.
Andare a ruota libera nell'interazione, in fase di testing, è un campo di prova più per la qualità delle regole e dei modelli che del reale funzionamento del sistema.
Se il sistema dimostra di funzionare sotto l'ipotesi di coerenza richiesta, nel caso le scelte dei modelli risorsa e delle regole di associazione risultassero poi sbagliate, la soluzione è semplicemente cambiarne i valori poiché al resto ci penserà l'adattatività strutturale di PerfectFit06.
In appendice sono consultabili, tra le altre, le tabelle riguardanti i vari modelli risorsa e le regole di associazione.
Per la verifica del sistema, non si sceglierà un utente appartenente a una classe *pura* poiché il test risulterebbe banale.

3.6.1 Test di funzionamento: utente turista con interessi storici

Si scelga, come utente di riferimento per il test, un turista con interessi storici.
Si presuppone, dunque, che comportandosi come questo tipo di utente, al termine di alcuni passi di interazione, PerfectFit06 cominci a costruire un modello utente che rispecchi i suddetti comportamenti.
Da adesso in poi si farà riferimento alle risorse degli scavi archeologici di Ercolano estratte dal grafo della conoscenza del par 2.6.
All'interno del database di PerfectFit06, per ora, sono stati caricati i dati relativi alle "Terme del foro" che fanno parte della "Insulae VI".
"Insulae VI" sarà una risorsa di tipo "Insulae" mentre terme del foro sarà di tipo "Ubicazione e denominazione", si faccia riferimento sempre al grafo del par 2.6.
Sono state scelte le "Teme del foro" poiché sono quelle che hanno un sotto grafo istanziato, e quindi un numero di istanze-risorsa (par 2.9), sufficientemente ampio e dunque un maggior numero di nodi web percorribili in fase di test.
Partendo dallo stato iniziale del sistema, col modello utente avente tutti i campi azzerati si arrivi, tramite le "risorse accessibili" (par 3.2.2), al nodo "ubicazione e denominazione" di "Terme del foro" senza selezionare nessuna altra risorsa.
Fino a questo punto il profilo utente non avrà subito variazioni e si presenterà come in figura:

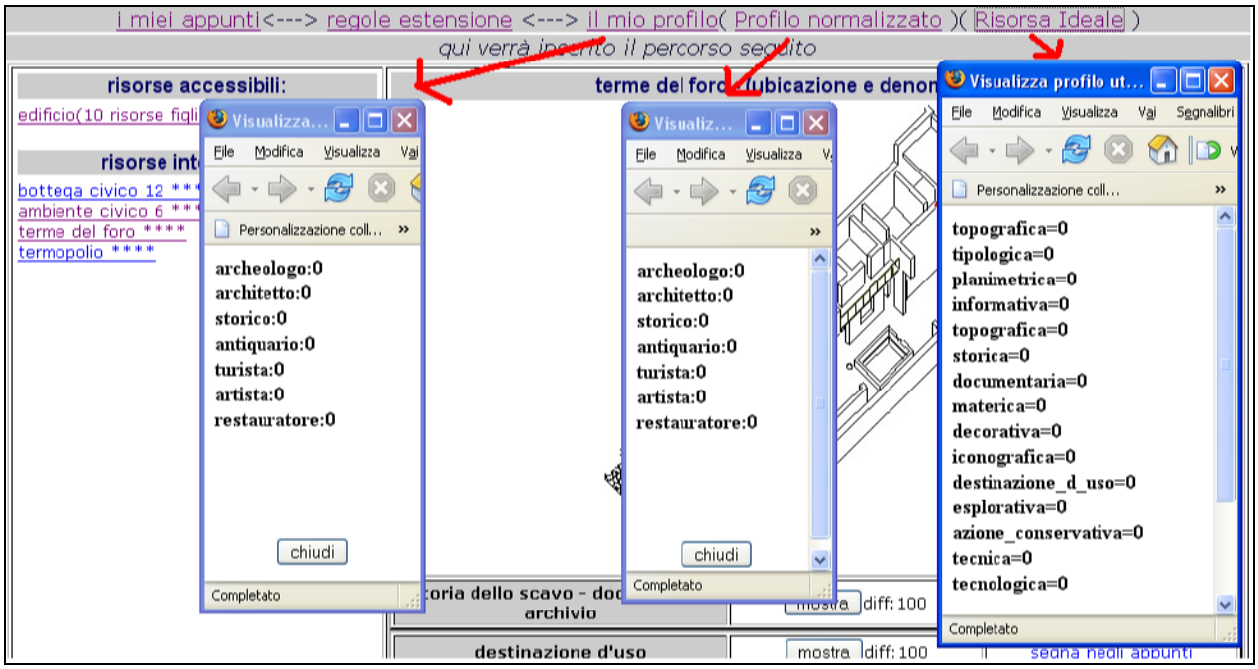


fig 3.6.1 – stato iniziale del sistema PerfectFit06

Se, a questo livello, si attiva il contenuto di “destinazione d’uso” e “resti umani” mediante il tasto “mostra” si può cominciare a vedere un interessante cambiamento nel modello utente (fig 3.6.2).

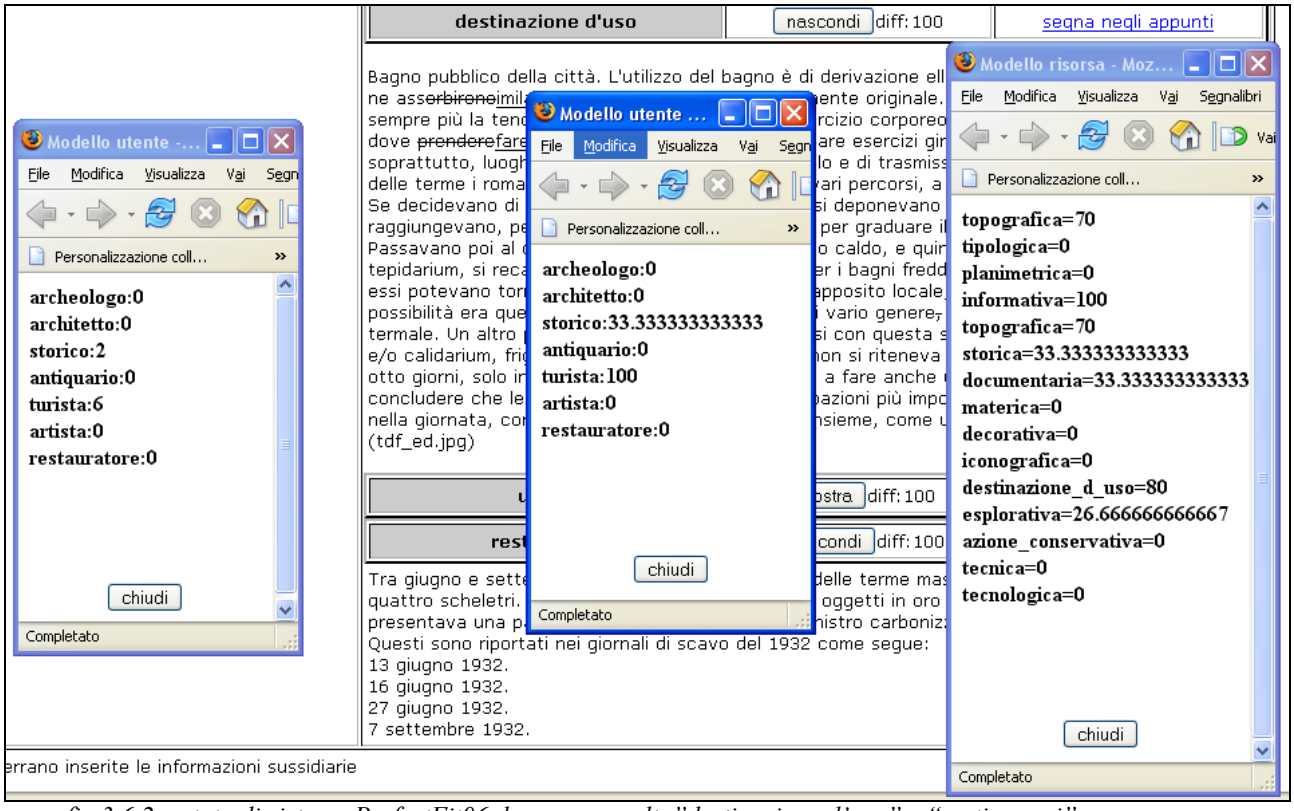


fig 3.6.2 – stato di sistema PerfectFit06 dopo aver scelto”destinazione d’uso” e “resti umani”

Inizia ad essere visibile che il sistema sta cominciando già a riconoscere l’utente come turista che ha degli interessi storici.

A questo punto, il risultato atteso, è che tutte le risorse che suscitino un minimo di interesse per un utente di questo tipo, in base alle regole di associazione, debbano mostrare una distanza inferiore a 100.

Accedendo al nodo principale “edificio” si vede che, nella nuova pagina, alcune risorse non hanno più come distanza 100 (fig 3.6.3).

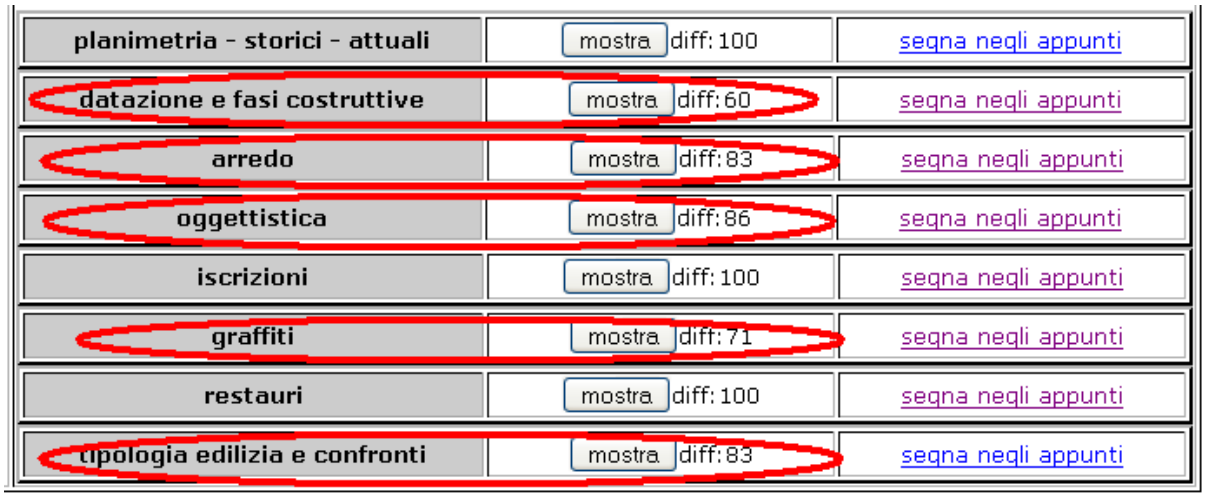


fig 3.6.3 – nuovi valori delle distanze tra risorsa ideale e risorse reali

Il risultato è già di per se incoraggiante.

Se si vanno a confrontare i modelli delle risorse con le regole di associazione delle classi "turista" e "storico" si comincia a vedere che, effettivamente, il sistema sta riconoscendo quel dato comportamento.

A questo punto, analizzando i modelli risorsa e le regole di associazione, nonché i valori ottenuti come distanze in fig 3.6.3, le risorse di maggior interesse per il modello utente attuale dovrebbero essere "datazione e fase costruttive", con differenza 60, e "graffiti", con differenza 71.

In effetti, andando ad analizzare i modelli risorsa associati alle due risorse appena menzionate, essi avvalorano effettivamente il comportamento del sistema.

Se ci si continua a comportare come un turista con interessi storici, il passo successivo è cliccare sul tasto "mostra" della risorsa "graffiti" che è associata a un modello risorsa interessante per l'utente che si sta testando.

Si ricarichi ora la pagina al fine di forzare il sistema a visualizzare le distanze aggiornate, sottolineando che questa operazione ha senso solo in fase di testing.

Si otterrà il seguente risultato di figura 3.6.4.

planimetria - storici - attuali	mostra diff: 100	segna negli appunti
datazione e fasi costruttive	nascondi diff: 40	segna negli appunti
La mancanza di epigrafi attestanti il nome di un eventuale munifico costruttore, ci fa supporre che esse furono costruite nell'epoca giulio-claudio ad opera di pubblici magistrati e a spese del pubblico erario. L'area sulla quale insistono le terme era, secondo il Maiuri, occupata precedentemente da case private che furono abbattute per rendere possibile la loro realizzazione.		
arredo	mostra diff: 83	segna negli appunti
oggettistica	mostra diff: 71	segna negli appunti
iscrizioni	mostra diff: 83	segna negli appunti
graffiti	mostra diff: 57	segna negli appunti
restauri	mostra diff: 100	segna negli appunti
tipologia edilizia e confronti	mostra diff: 83	segna negli appunti

fig 3.6.4 – differenze tra risorse reali e risorsa ideale aggiornata

Si noti che, come previsto, la distanza di "graffiti" diminuisce e soprattutto viene applicata l'adattività alla risorsa "datazione e fasi costruttive" di cui ne viene automaticamente visualizzato il contenuto associato.

Riuscire ad avere un risultato in un sistema adattativo dopo solo tre passi è indice della bontà del modello utente normalizzato.

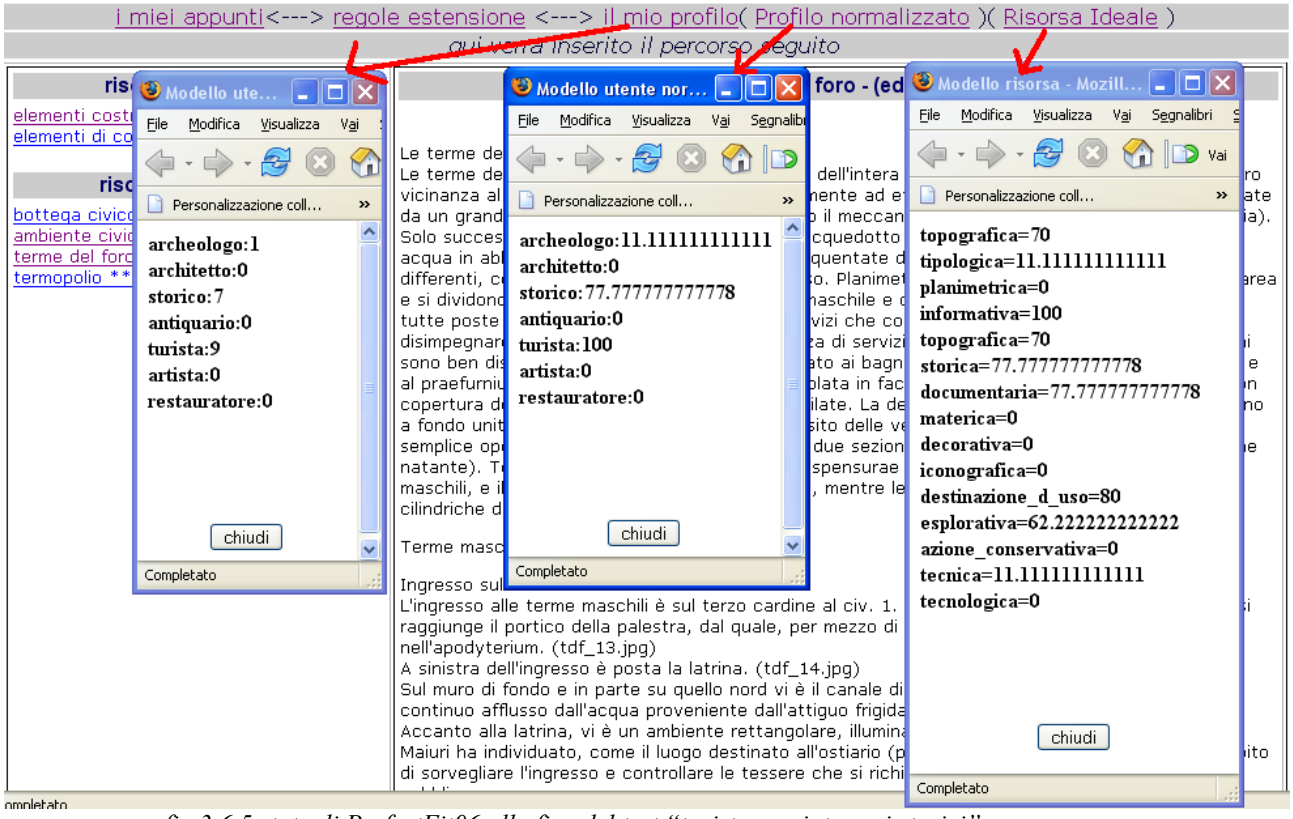


fig 3.6.5 stato di PerfectFit06 alla fine del test "turista con interessi storici"

Per PerfectFit06 è stata ovviamente impostata una distanza minima sotto la quale la risorsa viene considerata interessante, nell'attuale implementazione questo valore è 40. Si vada ora a vedere lo stato dei modelli del sistema mostrato in figura 3.6.5, come si può evincere dai valori mostrati il test è perfettamente riuscito.

3.6.2 Test di funzionamento: utente archeologo con interessi antiquari

Nel test precedente si è testato un utente per cui è stato possibile individuare immediatamente delle risorse di interesse e ci si è comportati esattamente secondo una condotta 'augurabile', cioè senza mai prendere scelte che non fossero strettamente legate all'utente di riferimento.
Anche se il test precedente è da ritenersi comunque più che accettabile, volendo essere più severi, si tenterà successivamente anche di disorientare il sistema comportandosi in maniera non immediatamente definita secondo l'utente di riferimento.
E' chiaro che se si "imbroglia" in tutta l'interazione il sistema non può far altro che assecondare l'utente.
Si provi adesso a ripetere le stesse operazioni dell'esempio precedente esibendo il comportamento di un archeologo con interessi antiquari.
Partendo dal nodo "ubicazione e denominazione" di "terme del foro", si scelga "Storia dello scavo - documenti di archivio" che è di forte interesse per uno storico con interessi archeologici.
Il modello utente attuale sarà quello presentato in figura 3.6.6.
A questo punto la componente che riguarda la sfera degli interessi rispetto all'antiquariato non è stata ancora esibita nell'ambiente di sistema.

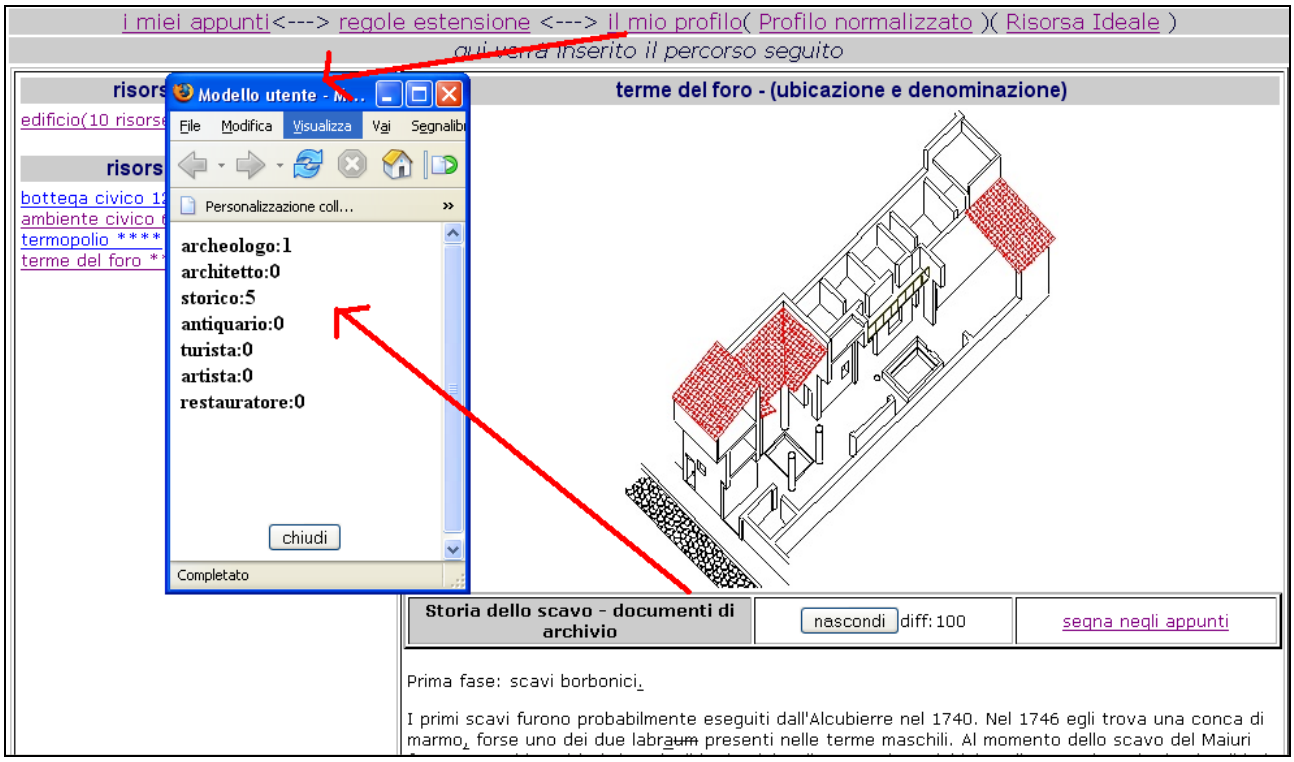


fig 3.6.6 stato di PerfectFit06 dopo la scelta di "storia dello scavo – documenti d'archivio"

Oltretutto c'è una componente di interessamento storico che, in qualche modo, è di disturbo per la definizione del modello utente di riferimento.
Continuando l'interazione, nel nodo successivo "edificio", sempre in "Terme del foro", vediamo che il sistema ha cominciato a calcolare le distanze rispetto al modello attuale di storico con interessi archeologici (fig 3.6.7).

planimetria - storici - attuali	mostra diff: 100	segna negli appunti
datazione e fasi costruttive	mostra diff: 80	segna negli appunti
arredo	mostra diff: 100	segna negli appunti
oggettistica	mostra diff: 86	segna negli appunti
iscrizioni	mostra diff: 83	segna negli appunti
graffiti	mostra diff: 71	segna negli appunti
restauri	mostra diff: 100	segna negli appunti
tipologia edilizia e confronti	mostra diff: 100	segna negli appunti

fig 3.6.7 distanze dopo la scelta di "storia dello scavo – documenti d'archivio"

Si vada ora a definire meglio il proprio comportamento scegliendo, senza necessariamente rispettare l'ordine di presentazione, prima una risorsa di interesse principalmente archeologico come "iscrizioni" e poi una di interesse prettamente antiquaria come "arredo", inserendo magari la prima negli appunti in modo da esibire in maniera decisa la componente di interesse verso l'antiquariato.
Ricaricando la pagina per forzare il sistema ad aggiornare le distanze vediamo come la risorsa "tipologie edilizie e confronti", che inizialmente il sistema aveva rilevato come per nulla interessante (fig 3.6.7), sia ora nettamente la più vicina, insieme ad "arredo", agli interessi dell'utente (fig 3.6.8).

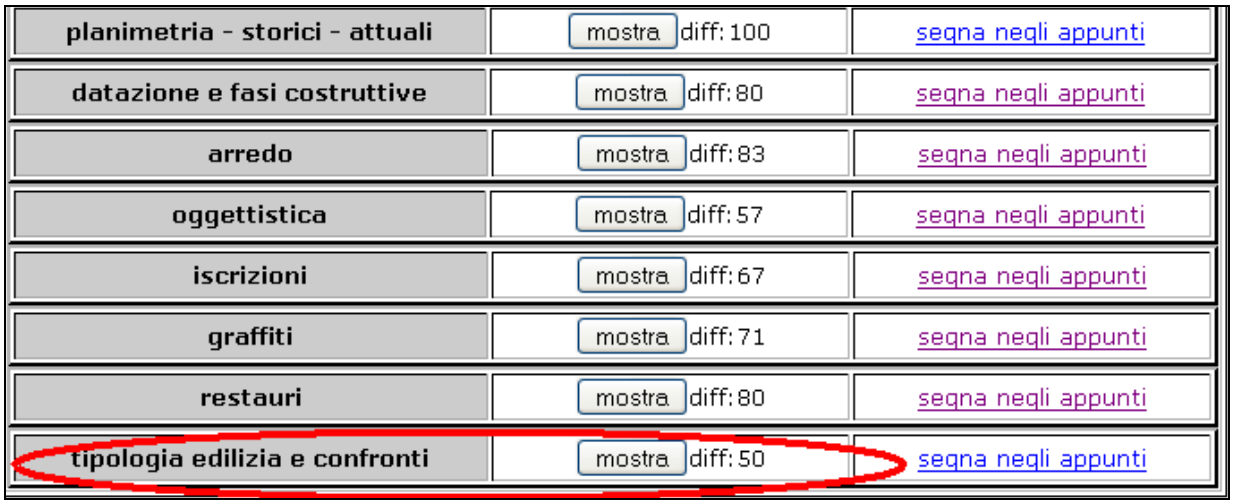


fig 3.6.8 - distanze aggiornate dopo la nuova interazione

In effetti, andando a sfogliare il modello risorsa di "tipologia edilizia e confronti" si noterà che è una risorsa di interesse proprio per un utente archeologo con interessi antiquari.
Si visualizzi lo stato del sistema come in figura 3.6.9.

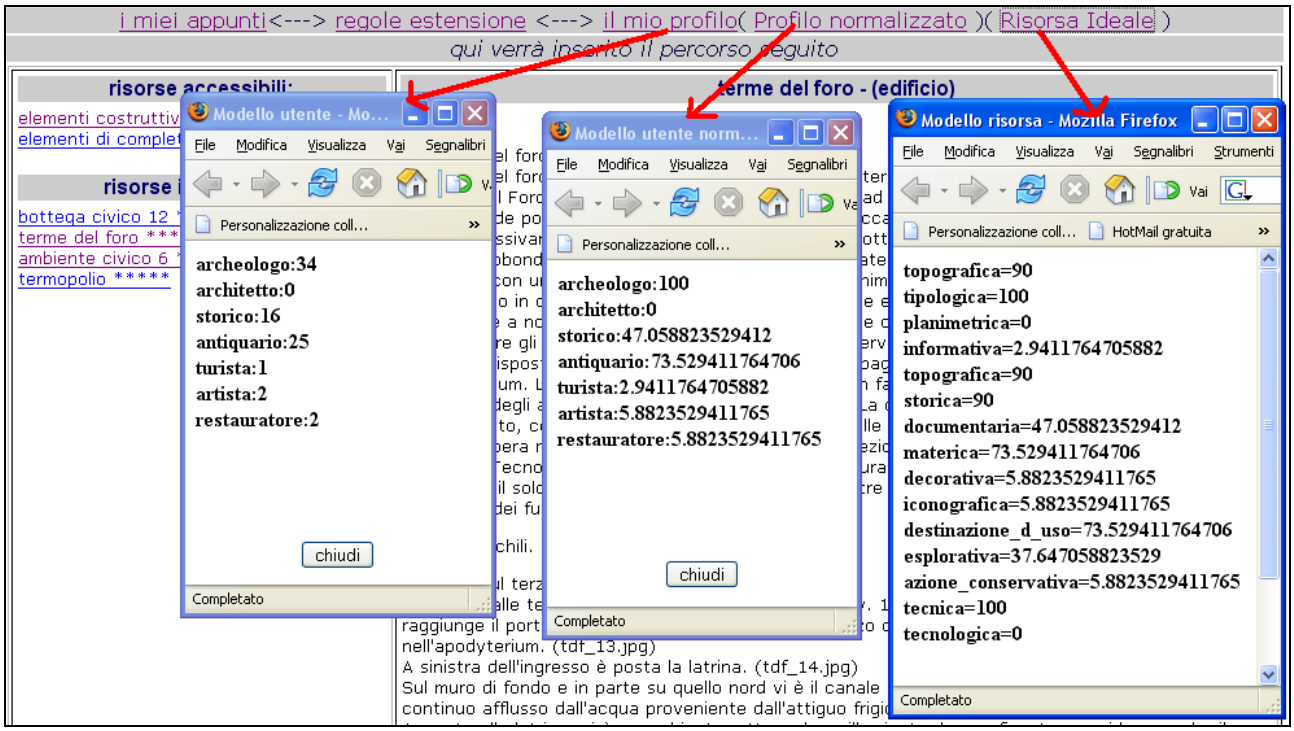


fig 3.6.9 – stato di PerfectFit06 dopo le nuove interazioni

E' nettamente evidente che PerfectFit06, pur non avendo eliminato la componente storica, è riuscito con tre interazioni a definire con sufficiente precisione l'utente di riferimento per il test.

Il fatto di aver aggiunto "iscrizioni" negli appunti è stato utile per rendere più evidente come, oltre all'apporto essenziale della normalizzazione del modello, anche la tecnica di raccolta appunti aumenti la velocità di modellizzazione del sistema.

3.7 Scelte progettuali in PerfectFit06

Nel realizzare questo documento si è volutamente scelto di non fare alcun cenno all'aspetto implementativo del sistema fino a questo paragrafo.
Le motivazioni di questa scelta vanno ricercate nella volontà di mettere in evidenza soprattutto le tecniche e le innovazioni del sistema PerfectFit06 aldilà di come sono state poi realizzate nella pratica.
Implementare un sistema come quello presentato non è certo semplice e le valutazioni di carattere progettuale da affrontare sono state numerose e difficili.
La versione di PerfectFit06 che è stata usata in questo capitolo, per quanto destinata principalmente a illustrare il funzionamento e le potenzialità del sistema, ha comunque reso necessario sciogliere tutti i dubbi di carattere tecnico legati all'implementazione del sistema.
Infatti, anche se a più riprese è stato detto che attualmente PerfectFit06 è un prototipo, con l'aggiunta di una interfaccia utente più amichevole il sistema è già perfettamente funzionante, come d'altronde è stato dimostrato nei paragrafi precedenti.
Si ritiene dunque ragionevole realizzare una sintesi delle analisi progettuali che hanno portato, nella pratica, alle scelte più determinanti per l'implementazione del sistema.

3.7.1 Scelta dei linguaggi di programmazione

Una prima caratteristica di PerfectFit06, che potrebbe spiazzare, è che esso non è stato realizzato in un unico linguaggio di programmazione, ma bensì in due.
La necessità di un doppio linguaggio di programmazione verrà subito resa chiara attraverso l'analisi seguente.
La prima cosa in assoluto da definire, per una corretta scelta del linguaggio di programmazione, è l'ambiente entro cui il sistema da implementare andrà a funzionare.
Come è ormai ampiamente noto, l'ambiente di PerfectFit06 è il Web.
I linguaggi disponibili per il Web sono numerosi e hanno caratteristiche e potenzialità spesso nettamente differenti tra loro.
A questo punto bisogna analizzare attentamente le esigenze che si devono andare a soddisfare per l'implementazione del sistema, sia dal punto di vista dell'utente che da quello del progettista.
Si è individuato un utente tipo di PerfectFit06 come una persona non necessariamente esperta di informatica se non addirittura poco avvezza all'utilizzo del computer.
Questo implica che il sistema deve essere potenzialmente semplice da usare, quindi usabile ed accessibile[Visciola-1], sia dal punto di vista dell'interfaccia che da quello, più delicato, della tecnologia utilizzata.
Un primo linguaggio che sembrava avere le caratteristiche adatte all'implementazione di un sistema complesso come PerfectFit06 è stato il Java.
Senza dubbio il Java offre ottime proprietà per quel che concerne la sicurezza e la portabilità del sistema, d'altro canto queste stesse caratteristiche rendono le sue applicazioni piuttosto "pesanti" per il Web sia da un punto di vista computazionale che in termini di risorse di sistema, deterrenti eccezionali per la pazienza di un utente poco esperto di informatica.
Un altro aspetto che ha poi portato a scartare il linguaggio Java per l'implementazione di PerfectFit06 è la necessità di installare una Java Virtual Machine su ogni computer client e di dover scaricare eventuali plug-in da installare per le funzioni più avanzate del sistema.
Queste operazioni non sono immediate per una utenza potenzialmente poco esperta come si è presupposta essere quella di PerfectFit06.
Il vaglio dei linguaggi di programmazione si è dunque spostato sulla classe dei linguaggi di scripting che offrono, generalmente, una buona espressività del linguaggio

e la semplicità d'uso sia per il progettista, in fase di creazione, che per l'utente, in fase di utilizzo del prodotto finito.
Un passo determinante, per la scelta del linguaggio di programmazione da usare, è stato fatto quando ci si è resi conto che PerfectFit06 ha la necessità di fare delle elaborazioni sia lato server (accesso al database, calcolo delle risorse interessanti, creazione delle regole di estensione) che lato client (memorizzazione del modello utente e relativa modifica una volta che la pagina web è stata caricata, modifica dinamica dell'interfaccia della pagina web).
Quindi è diventato evidente che il bisogno di due linguaggi di programmazione, uno lato client e uno lato server.
La scelta del linguaggio lato client è caduta immediatamente sul JavaScript, combinato con i CSS (Cascade Style Sheet), grazie alle sue caratteristiche di velocità, semplicità, portabilità e leggerezza.
Per gestire le informazioni lato client è stato utilizzato il metodo dei file-cookie.
Si ricorda che i linguaggi di programmazione lato client, generalmente, non possono accedere ai database lato server né a nessuna altra informazione su quest'ultimo.
Nasce quindi anche il problema della necessità di passare i dati tra il server e il client, compito questo che deve gravare necessariamente sul linguaggio operante lato server.
Oltretutto, come mostrato nel paragrafo 2.13 e 2.15.4, il sistema è capace di autogenerare delle sue componenti e di autoreferenziarsi mandando in esecuzione codice creato dinamicamente all'interno della stessa sessione.
Quindi, riepilogando, il linguaggio di programmazione lato server deve:

- Accedere facilmente a un database lato server.
- Accedere ai cookie lato client.
- Essere sufficientemente espressivo da poter autogenerare dinamicamente del proprio codice .
- Avere la capacità di mandare in esecuzione del proprio codice scritto nella stessa sessione di lavoro.
- Avere la capacità di definire strutture dati complesse come quelle necessarie a PerfectFit06.

La risposta a tutte queste esigenze si è trovata nel linguaggio PHP.
Infatti:

- Il PHP può accedere a qualsiasi tipo di database poiché si interfaccia con ODBC, nonché direttamente con MySQL.
- Ha funzioni apposite per la gestione dei cookie lato client.
- Il suo codice è in formato testuale, quindi con semplici funzioni di scrittura file, può autogenerare del proprio codice.
- E' un linguaggio interpretato in maniera sequenziale dal server e permette il caricamento dinamico di moduli esterni in qualsiasi punto del codice, quindi anche moduli creati un'istruzione prima.
- E' estremamente espressivo e, nelle ultime versioni, permette anche una discreta programmazione ad oggetti.

Quindi la combinazione JavaScript/CSS-PHP è stata considerata la più conveniente per l'implementazione di PerfectFit06.

3.7.2 Scelta del database

La scelta del database è stata molto meno critica rispetto a quella dei linguaggi di programmazione.
Le strutture dati utilizzate in PerfectFit06 che necessitano di essere memorizzate in un database non hanno caratteristiche tali da creare particolari esigenze.
I parametri utilizzati per la scelta sono stati dunque molto più generici rispetto a quelli utilizzati nel paragrafo precedente.
I due database che si sono presi inizialmente in considerazione per PerfectFit06 sono stati Oracle e MySQL.
Oracle è senza dubbio un database estremamente completo, complesso e performante e quindi rappresenta una scelta di prima qualità.
D'altro canto tutta questa potenza, per essere sfruttata, richiede un server con caratteristiche tecnologiche di alto profilo e, oltretutto, Oracle ha degli alti costi di acquisto per la licenza.
Questi due aspetti hanno scoraggiato la scelta di Oracle che avrebbe reso difficile lo sviluppo "decentrato" di PerfectFit06 su computer con prestazioni non sufficientemente potenti e privi di licenza.
Come è intuibile, per esclusione, la scelta finale è stata MySQL.
Anche se MySQL non offre tutte le potenzialità di Oracle, per le esigenze di PerfectFit06, si è rivelato più che sufficiente.
Le performance dei database MySQL sono notoriamente ottime, il server MySQL è gratuito e, soprattutto, PHP offre funzioni di interfaccia a questo database estremamente veloce e efficace.
In una evoluzione del sistema, come presentata nel pararagrafo 3.8.3, è ragionevole pensare ad una futura migrazione verso Oracle.
Ha comunque sicuramente influito sulla scelta finale del database la precedente esperienza del progettista con MySQL che ha permesso di ridurre i tempi di realizzazione del sistema.

3.8 Evoluzione del sistema

Anche se PerfectFit06 è stato a più riprese definito come sistema di personalizzazione dei contenuti web, la sua architettura, piuttosto complessa ed eterogenea, non ne permette una facile classificazione.
Per PerfectFit06 si è tentato di progettare un sistema che raggruppasse insieme in maniera efficiente e innovativa un numero abbastanza cospicuo di tecniche.
Infatti l'utilizzo contemporaneo delle tecniche adottate per la modellizzazione comportamentale dell'utente, la caratterizzazione in risorse delle informazioni, la capacità di auto-generare sue componenti e l'utilizzo combinato di regole di estensione e associazione sono difficilmente collocabili in un settore specifico.
La ricerca di una struttura di sistema fortemente modulare è stata voluta proprio per far sì che fosse possibile innestare, in aggiunta o sostituzione dei moduli presenti, nuovi moduli con caratteristiche specifiche più potenti.
Tutte le caratteristiche presentate nel secondo capitolo possono essere suscettibili di interpretazioni o migliorie personali al momento dell'implementazione o della stessa progettazione.
Questa versione di PerfectFit06 ha mirato a mettere delle solide basi per un sistema che può diventare molto più complesso e potente, senza però rinunciare a essere efficiente e completo rispetto ai suoi scopi primari.
Nel seguito del paragrafo verranno proposte delle possibili evoluzioni del sistema scelte per la loro fattibilità.

3.8.1 PerfectFit06 come Agente Web

Anche se nel resto della discussione non se ne è parlato apertamente, PerfectFit06 si può già, allo stato attuale, considerare un agente Web secondo i parametri presentati nel paragrafo 1.2.
Il motivo di questa reticenza è che sarebbero potute sorgere obiezioni sulla proprietà di "abilità sociale" del sistema che, attualmente, non è fortemente marcata, per quanto essa sia già presente.
Si considerino, prima di tutto, le altre proprietà atte a definire un agente rispetto le caratteristiche di PerfectFit06:

- **Autonomia:** PerfectFit06 è assolutamente autonomo nella creazione e gestione del modello utente e la conseguente scelta delle risorse interessanti per esso.
- **Reattività:** come mostrato negli esempi del paragrafo 3.6 il sistema è capace di recepire gli stimoli ricevuti dall'ambiente, come l'apertura del contenuto di una risorsa col tasto "mostra/nascondi", e di reagire tramite il cambiamento del modello utente e la conseguente adattatività sulle risorse interessanti.
- **Pro-attività:** PerfectFit06, in base al modello utente attuale, è capace di agevolare l'utente nell'esplorazione del sito proponendogli automaticamente i contenuti delle risorse interessanti e assistendolo nel perseguimento del suo scopo di raccolta di informazioni.

Come accennato, per la proprietà di "abilità sociale", il discorso è leggermente più complesso.
Prima di tutto, per mostrare abilità sociale, c'è bisogno di almeno un altro agente con cui socializzare.
Si può considerare come agente lo stesso utente, poiché PerfectFit06 recepisce l'interazione dell'utente con l'ambiente utilizzandola per cambiare la sua l'interfaccia, e quindi il modo di interagire con esso, questa è senza dubbio una abilità sociale.
D'altronde lo stesso concetto di adattatività di un sistema web si può far risalire alla proprietà di socializzazione tra sistema e utente.
Volendo riferirsi all'abilità sociale verso un altro agente, si consideri quindi uno scenario, potenzialmente realistico, in cui l'architettura Web di PerfectFit06 viene applicata a due contesti diversi, ad esempio gli scavi di Ercolano e un negozio on-line di vendita pacchetti viaggio.

Ovviamente i due sistemi avranno risorse, classi '*pure*', rispettivi modelli e regole diverse.
Ciò non esclude che le informazioni sull'utente raccolte da un sistema non possano tornare utili all'altro.
Si immagini di aver effettuato una interazione col sito degli scavi archeologici di Ercolano e di aver dunque permesso al sistema di definire un modello utente per questo sito.
I valori del modello utente saranno salvati dal sistema in dei file cookie che, senza grosse modifiche tecniche, possono essere resi persistenti sul computer client.
Si immagini ora di accedere su un sito di vendita on-line di pacchetti viaggio che utilizza l'architettura di PerfectFit06.
Il sistema potrebbe, a questo punto, utilizzare il modello utente salvato dalla versione di PerfectFit06 di Ercolano per catturare in anticipo determinate preferenze dell'utente.
Ad esempio, se l'utente ha mostrato un forte interesse per l'archeologia, potrebbe presentare in prima pagina le offerte-viaggio per siti archeologici.
Lo scenario mostrato, permette di capire come, in maniera estremamente semplice, più istanze di PerfectFit06 possano collaborare tra loro scambiandosi informazioni determinanti e di utilità per la navigazione dell'utente.
Si pensi se PerfectFit06, o dei sistemi con una gestione del modello utente simile, fossero integrati nella maggior parte delle soluzioni Web.
Tramite il salvataggio di modelli utente lato client creati in base ai vari contesti specifici, diventa evidente che i sistemi sono capaci di scambiarsi gusti e preferenze dell'utente mostrando abilità sociale.
Poiché ogni modello utente farebbe riferimento a un dato contesto, si andrebbe ad avere una definizione dell'utente sotto numerosi aspetti, ottenendo una modellizzazione sempre più efficace e a 360 gradi.
L'utente, da parte sua, ha il vantaggio di avere a disposizione sistemi che già conoscono le sue preferenze quando esso vi accede e dunque, possono servirlo al meglio già nell'immediato.
Il salvataggio dei modelli utente non è una violazione della privacy poiché l'utente è comunque assolutamente anonimo per il sistema.
Nel momento in cui si è dimostrato che PerfectFit06 può essere considerato a tutti gli effetti un agente, è il caso di provare a classificarlo come tale secondo le definizioni presentate nel paragrafo 1.2.1.

PerfectFit06 non ricalca in particolare nessuna architettura di quelle a cui si fa riferimento nel paragrafo 1.2.1 ma presenta degli aspetti caratteristici di alcune di esse.
Il sistema fa utilizzo di regole di associazione da cui, tramite funzioni matematiche, si deducono le regole di estensione e i modelli ideali.
Anche se la deduzione viene realizzata tramite funzioni matematiche, la tipologia di comportamento è riportabile ai sistemi ad agenti su base logica.
Tramite il calcolo dell'interesse dell'utente verso una risorsa del sistema (par 2.15), PerfectFit06 è capace di reagire autonomamente mappando questa situazione nell'azione, ad esempio, di mostrare o nascondere un contenuto informativo.
Il comportamento appena presentato è caratteristico degli agenti reattivi.
Anche se all'interno del sistema non c'è una formalizzazione in questo senso, esso presenta molte caratteristiche degli agenti BDI.
Infatti, si consideri il sistema in un dato momento del suo funzionamento.
Il modello associato all'utente rappresenterà la sua appartenenza alle classi *pure*, questa appartenenza rappresenta anche gli interessi dell'utente tramite le regole di associazione e, quindi, definisce le risorse del sistema a cui l'utente è interessato.
Avere la possibilità di creare una risorsa ideale dal modello utente vuol dire riuscire a descrivere le sue intenzioni, ovvero l'insieme di risorse a cui esso vorrebbe accedere motivato dall'interesse.
Il fatto che da una risorsa ideale il sistema possa stabilire quali contenuti informativi mostrare e quali no, tramite il calcolo di una distanza, significa che il sistema tenta di soddisfare i desideri dell'utente espressi implicitamente durante la sua interazione.
Quando un utente nella sua navigazione, sceglie, ad esempio, di mostrare o nascondere dei contenuti informativi di una risorsa, esso sta esprimendo delle opinioni che definiscono i suoi interessi.
L'applicazione delle regole di estensione può essere considerata la tecnica che usa il sistema per rappresentare e utilizzare le opinioni espresse dall'utente.

3.8.2 Modulo esperto

Il sistema PerfectFit06 fa ampio utilizzo di regole, sia di associazione che di estensione. Per quel che concerne le regole di estensione, come mostrato nel paragrafo 2.15.4, esse sono calcolate dinamicamente.
Le regole di associazione sono invece definite 'manualmente' rispetto alle classi *pure*.
Per come è realizzata l'architettura di PerfectFit06 si può pensare facilmente di inserire un modulo esperto capace di "migliorare" le regole di associazione inserite inizialmente.
Il modulo avrebbe il compito di calcolare l'efficienza della regole di associazione, verificando che le risorse scelte dall'utente rispecchino effettivamente il modello utente.
Si potrebbe così stabilire anche una stima dell'efficienza del sistema e indicare dove, e in che peso, andare a modificare le regole.
Il modulo potrebbe anche avere l'autonomia di variare le regole dove l'errore è palese, in fondo "l'esperienza è il nome che si da ai propri errori"[Wilde-1].
Se, ad esempio, risulta che spesso un utente in cui la componente di archeologo è forte, sceglie una risorsa il cui modello ha poco a che vedere con l'archeologo, il modulo esperto potrebbe decidere di cambiare le regole di associazione della classe *pura* archeologo o suggerire come modificare il modello risorsa.
Sarebbe interessante anche partire senza regole di associazione definite e utilizzare un modulo esperto per una sessione di apprendimento del sistema definendo così, man mano, le regole da utilizzare e validandole, eventualmente, richiedendo all'utente a quale modello esso si sente più vicino.

3.8.3 Miglioramenti generici del sistema

La gamma di miglioramenti apportabili al sistema è, come per ogni progetto di questo tipo, piuttosto ampia.
Se si vede il forte utilizzo che si fa del database in PerfectFit06 e si nota come sono strutturate le tabelle, viene quasi spontaneo immaginare una organizzazione in classi di queste ultime.
Si possono individuare agevolmente delle classi e dei metodi per il database di PerfectFit06 e in particolare per le risorse e l'applicazione o creazione delle regole.
L'utilizzo di un database ad oggetti, come ad esempio Oracle, è un passo del tutto naturale nell'evoluzione del sistema e porterebbe numerosi vantaggi soprattutto nell'organizzazione delle risorse e nel calcolo delle regole di estensione.
Un altro aspetto che andrebbe ridefinito in PerfectFit06 è il grafo della conoscenza attualmente disponibile per gli scavi archeologici di Ercolano.
Questo grafo è stato realizzato con scopi molto diversi dall'utilizzo per il web.
Per quanto la sua descrizione sia efficace, per il sistema sarebbe ideale se fosse realizzato per modellare gli interessi o i gradi di approfondimento tecnici di ogni nodo.
Anche le risorse del sistema PerfectFit06, di conseguenza, risultano essere poco adatte all'ambiente in cui sono state inserite e una riformulazione anche dei contenuti, magari con gradi di complessità e dettaglio tecnico diversificati, renderebbe maggiormente evidente l'utilità del sistema.
Nel caso non si faccia uso di sistemi esperti, le regole di associazione richiederebbero per la loro formulazione, l'aiuto di esperti di psicologia e analisi comportamentale.
Uno studio completo da questo punto di vista però, porterebbe via troppo tempo e risorse umane rendendo più interessante l'applicazione di un modulo esperto come indicato nel paragrafo 3.8.2.
Approfittando del fatto che il modello risorsa è molto simile strutturalmente a una ontologia, si potrebbe pensare di esportarlo in XML o RDF per utilizzarlo per il Web semantico.
Dall'altro canto, anche il modello utente utilizzato come nello scenario del paragrafo 3.8.1 potrebbe tornare utile per il Web semantico.
Volendo migliorare il modello utente, si può pensare di modellizzare anche aspetti tecnici dell'utente, come la velocità di collegamento o il grado di conoscenza dell'ambiente Web, per permettere una adattatività più spinta definendo magari le risorse anche da un punto di vista tecnologico.

Si potrebbe così estendere l'adattatività ai contenuti tecnologici come i video, le immagini oppure identificare risorse adatte per il Web classico, per il Wap etc.
Un altro aspetto che si può notevolmente affinare in PerfectFit06 è la gestione del grado di interesse dell'utente, che può essere utilizzato in maniera molto più utile della semplice visualizzazione del contenuto della risorsa.
Il grado di interesse dell'utente si potrebbe, ad esempio, utilizzare per definire i nodi web dove sono accessibili il maggior numero di risorse interessanti, stabilire delle priorità nella ricerca all'interno del sito in funzione dell'interessamento e altro ancora.
Quanto descritto è solo una piccola parte del repertorio di possibili miglioramenti del sistema che, comunque, rende ancora più evidente come PerfectFit06 risulti essere un sistema innovativo.

Appendice A – Regole di associazione

	archeologo	architetto	storico	antiquario	turista	artista	restauratore
topografica	90	0	0	0	70	0	0
tipologica	100	0	0	100	0	0	0
planimetrica	0	80	0	0	0	0	0
informativa	0	0	0	0	100	0	0
storica	90	0	100	60	0	0	0
documentaria	0	0	100	0	0	0	0
materica	0	0	0	100	0	0	100
decorativa	0	0	0	0	0	100	80
iconografica	0	0	0	0	0	100	0
destinazione_d_uso	0	0	0	100	80	0	0
esplorativa	0	0	80	0	0	0	0
azione_conservativa	0	100	0	0	0	0	100
tecnica	100	100	0	0	0	0	0
tecnologica	0	90	0	0	0	0	0

Appendice B – Regole di estensione calcolate da PerfectFit06

	archeologo	Architetto	storico	antiquario	Turista	artista	restauratore
insulae	0	0	0	0	0	0	0
ubicazione e denominazione	0	0	0	0	0	0	0
Storia dello scavo - documenti di archivio	1	0	5	0	0	0	0
destinazione d'uso	0	0	0	0	6	0	0
edificio	0	0	0	0	1	0	0
utenti	0	0	0	0	3	0	0
resti umani	0	0	2	0	0	0	0
planimetria – storici – attuali	0	0	0	0	0	0	0
datazione e fasi costruttive	0	0	4	0	2	0	0
elementi costruttivi	0	0	0	2	1	0	0
spazi ed elementi sotterranei	0	4	0	0	0	0	1
pre-esistenze	0	3	5	0	0	0	0
elementi di completamento	0	0	0	0	1	3	2
arredo	0	0	0	3	1	2	2
oggettistica	1	1	0	1	0	5	0
iscrizioni	3	0	1	2	0	0	0
graffiti	1	0	5	0	3	0	0
restauri	0	2	0	0	0	0	7
tipologia edilizia e confronti	3	0	0	5	0	0	0
spazi annessi - aperti - chiusi	0	2	1	0	0	0	1
murature	6	2	0	1	0	0	3
coperture	0	0	0	0	0	0	1
strutture orizzontali - solai - volte	0	0	0	1	0	0	2
scale	0	0	0	1	0	0	2
portici e meniani	0	1	0	0	0	0	2
impianti tecnologici	0	1	0	1	0	0	2
pavimenti	0	1	0	0	0	7	2
balconi finestre porte e portoni	0	1	0	0	0	0	2
rivestimenti – interni - esterni	0	1	0	0	0	7	2

Appendice C – Modelli risorsa

arredo	
topografica	0
tipologica	90
planimetrica	90
informativa	70
storica	0
documentaria	0
materica	90
decorativa	80
iconografica	20
destinazione_d_uso	0
esplorativa	0
Azione_conservativa	0
tecnica	0
tecnologica	0

coperture	
topografica	0
tipologica	100
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	100
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	100
tecnologica	0

balconi finestre porte e portoni	
topografica	0
tipologica	30
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	60
decorativa	30
iconografica	0
destinazione_d_uso	0
esplorativa	0
Azione_conservativa	50
tecnica	100
tecnologica	100

datazione e fasi costruttive	
topografica	0
tipologica	0
planimetrica	0
informativa	50
storica	100
documentaria	100
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	40
esplorativa	50
azione_conservativa	0
tecnica	0
tecnologica	0

destinazione d'uso	
topografica	70
tipologica	0
planimetrica	0
informativa	90
storica	0
documentaria	0
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	100
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	0

elementi costruttivi	
topografica	0
tipologica	80
planimetrica	0
informativa	80
storica	0
documentaria	0
materica	80
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	0
tecnica	100
tecnologica	100

edificio	
topografica	0
tipologica	0
planimetrica	0
informativa	50
storica	0
documentaria	0
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	100
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	70

elementi di completamento	
topografica	0
tipologica	0
planimetrica	0
informativa	80
storica	0
documentaria	0
materica	60
decorativa	60
iconografica	60
destinazione_d_uso	0
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	60

graffiti	
topografica	40
tipologica	0
planimetrica	0
informativa	80
storica	100
documentaria	90
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	50
azione_conservativa	25
tecnica	80
tecnologica	0

insulae	
topografica	100
tipologica	0
planimetrica	80
informativa	100
storica	0
documentaria	0
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	0

impianti tecnologici	
topografica	0
tipologica	20
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	80
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	100
tecnologica	100

iscrizioni	
topografica	0
tipologica	90
planimetrica	0
informativa	0
storica	100
documentaria	0
materica	100
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	50
azione_conservativa	25
tecnica	80
tecnologica	0

murature	
topografica	80
tipologica	100
planimetrica	0
informativa	0
storica	70
documentaria	0
materica	80
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	80
tecnica	100
tecnologica	90

pavimenti	
topografica	0
tipologica	0
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	100
decorativa	100
iconografica	100
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	80
tecnologica	0

oggettistica	
topografica	0
tipologica	100
planimetrica	0
informativa	0
storica	40
documentaria	0
materica	0
decorativa	90
iconografica	60
destinazione_d_uso	30
esplorativa	0
azione_conservativa	0
tecnica	70
tecnologica	70

planimetria - storici - attuali	
topografica	0
tipologica	0
planimetrica	100
informativa	0
storica	100
documentaria	0
materica	80
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	60

portici e meniani	
topografica	0
tipologica	30
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	60
decorativa	30
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	100
tecnologica	100

restauri	
topografica	0
tipologica	0
planimetrica	80
informativa	0
storica	0
documentaria	0
materica	100
decorativa	100
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	100
tecnica	90
tecnologica	0

pre-esistenze	
topografica	0
tipologica	0
planimetrica	90
informativa	0
storica	100
documentaria	80
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	20
esplorativa	100
azione_conservativa	40
tecnica	80
tecnologica	90

resti umani	
topografica	0
tipologica	0
planimetrica	0
informativa	0
storica	0
documentaria	100
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	70
azione_conservativa	60
tecnica	0
tecnologica	0

rivestimenti - interni - esterni	
topografica	0
tipologica	0
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	100
decorativa	100
iconografica	100
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	80
tecnologica	0

spazi annessi - aperti - chiusi	
topografica	90
tipologica	0
planimetrica	100
informativa	0
storica	90
documentaria	0
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	80
esplorativa	0
azione_conservativa	90
tecnica	0
tecnologica	90

scale	
topografica	0
tipologica	50
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	80
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	100
tecnologica	0

spazi ed elementi sotterranei	
topografica	0
tipologica	30
planimetrica	100
informativa	30
storica	0
documentaria	0
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	90
tecnica	100
tecnologica	40

Storia dello scavo - documenti di archivio	
topografica	100
tipologica	80
planimetrica	0
informativa	0
storica	100
documentaria	70
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	70
azione_conservativa	0
tecnica	0
tecnologica	0

strutture orizzontali - solai - volte	
topografica	0
tipologica	50
planimetrica	0
informativa	0
storica	0
documentaria	0
materica	80
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	50
tecnica	100
tecnologica	0

tipologia edilizia e confronti	
topografica	80
tipologica	80
planimetrica	0
informativa	30
storica	100
documentaria	0
materica	70
decorativa	0
iconografica	0
destinazione_d_uso	100
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	0

ubicazione e denominazione	
topografica	70
tipologica	0
planimetrica	0
informativa	100
storica	0
documentaria	80
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	0
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	0

utenti	
topografica	0
tipologica	0
planimetrica	0
informativa	90
storica	0
documentaria	0
materica	0
decorativa	0
iconografica	0
destinazione_d_uso	100
esplorativa	0
azione_conservativa	0
tecnica	0
tecnologica	0

Appendice D – Pagina web di PerfectFit06

PerfecFit06 genera tutto il sito da una sola pagina se si escludono quelle per visualizzare lo stato del sistema viste nel terzo paragrafo.
La pagina qui riportata è **perfectfit06.php** che codice sia il codice PHP che quello JavaScript immerso nell'HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<link media="screen" href="ercolano.css" title="screen" type="text/css"
rel="stylesheet" />
<?php
 session_start();

 //Invoco i parametri di connessione al db, cioè username e password
 require("connessione.php");
 //Invoco la libreria di funzioni di perfectFit06
 require("funzioni.php");

 //verifica se i cookie sono attivi-->$_COOKIE['CookiesOn']

 //Inizializzo il profilo utente se non è stato creato
 if (!session_is_registered("profilo_utente_attuale"))
 {
  session_register("profilo_utente_attuale");
  $profilo_utente_attuale = array();
  inizializza_profilo_utente($profilo_utente_attuale);
  crea_cookie_profilo_utente($profilo_utente_attuale);
 }
 else
 {
  aggiorna_cookie_profilo_utente($profilo_utente_attuale);
 }

 //Crea dinamicamente le regole di estensione
 crea_regole_estensione();

?>

<SCRIPT type='text/javascript'>

 <?php
   //Genera dinamicamente il codice Javascript per l'aggiornamento del profilo
utente
   $str = "function aggiorna_profilo(id, ";
   $sql = "SELECT COUNT(*) AS NP FROM base_regole_associazione";
   $result = mysql_query($sql);
   $riga = mysql_fetch_array($result);
   for ($i=0; $i < $riga['NP']; $i++)
   {
    $str .= "var".$i;
      if ($i < $riga['NP']-1) $str .= ", ";
   }
   $str .= ")\n";
   $str .= "{\n";
```

```
   $str .= "if (id != 'null')\n my_object = getObject(id); \nelse\n
my_object=null\n\n";
   $str .= "if (id=='null') segno = 1\n else\n";
   $str .= " if (my_object.className=='visibile')\n     segno=1;\n else\n
segno=-1;\n\n";
   $sql = "SELECT * FROM base_regole_associazione";
   $result = mysql_query($sql);
   $i=0;
   while ($riga = mysql_fetch_array($result))
   {
     $str .=
"\nval=parseInt(GetCookie('".$riga['profilo_utente_riferimento']."'));\n";
     $str .= "val += (segno*var".$i.");\n";
     $str .= "\n//garante coerenza profilo\n";
     $str .= "if (val <  0) val =  0;\n";
     $str .= "if (val >100) val = 100;\n";
     $str .= "SetCookie('".$riga['profilo_utente_riferimento']."',val);\n";
       $i++;
   }
   echo $str ."\n}";
 ?>

//Funzioni JavScript di gestione dei cookie
function GetCookie (name)
{
var arg = name + "=";
var alen = arg.length;
var clen = document.cookie.length;
var i = 0;
while (i < clen) {
var j = i + alen;
if (document.cookie.substring(i, j) == arg)
return getCookieVal (j);
i = document.cookie.indexOf(" ", i) + 1;
if (i == 0) break;
}
return null;
}

function getCookieVal (offset) {
var endstr = document.cookie.indexOf (";", offset);
if (endstr == -1)
endstr = document.cookie.length;
return unescape(document.cookie.substring(offset, endstr));
}

function SetCookie (name,value,expires,path,domain,secure) {
document.cookie = name + "=" + escape (value) +
((expires) ? "; expires=" + expires.toGMTString() : "") +
((path) ? "; path=" + path : "") +((domain) ? "; domain=" + domain : "") +
((secure) ? "; secure" : "");
}

//Funzione che permette di "ottenere" il riferimento a un oggetto a prescindere
dal browser
 function getObject(objectId)
 {
      if(document.getElementById)
      {
        return document.getElementById(objectId)
      }
      else if (document.all)
      {
```

```
        return document.all(objectId)
      }
      else if(document.layers)
      {
        return document.layers[objectId];
      }
      return null;
 }

 //Funzione che gestisce il contenuto informativo delle risorse e il tasto
mostra/nascondi
 function toggle(id, button_id)
 {
   my_object = getObject(id);
   my_button = getObject(button_id);
   if (my_object.className == 'visibile')
   {
    my_object.className = 'nascosto';
      my_button.value='mostra  ';
   }
   else
   {
    my_object.className = 'visibile';
    my_button.value='nascondi';
   }
 }
</SCRIPT>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Scavi di Ercolano - Official Web Site</title>
</head>
<body>

<?php
 //Crea la lista di link principali e secodari in base alla risorsa ideale
 global $principale, $secondario;
 $principale = array();
 $secondario = array();

 if (isset($_GET['id_ist']))
  genera_lista_nodi($_GET['id_ist'], $_GET['padre_nodo'], $principale,
$secondario);
 else
  genera_lista_nodi("null", "null", $principale, $secondario);
?>

<table width='100%' class='principale'>
  <tr ID="riga_superiore" class='intestazione_tabella'>
    <td width="100%" align='center'>
       <img src="image/vesuvio_calmo.jpg" alt="impossibile caricare l'immagine"
width="100%" height="100%" />
      </td>
  </tr>
  <tr ID="riga_di_ricerca" class='intestazione_tabella'>
    <td  align='center'>
       <a href='mostra_appunti.php' target='_blank'>i miei appunti</a><--->
       <a href='mostra_regola_estensione.php' target='_blank'>regole
estensione</a> <--->
       <a href='visualizza_profilo_utente.php' target='_blank'>il mio
profilo</a>(
       <a href='visualizza_profilo_utente_normalizzato.php'
target='_blank'>Profilo normalizzato</a> )(
```

```
          <a href='visualizza_risorsa_ideale.php' target='_blank'>Risorsa
Ideale</a> )
      </td>
  </tr>
  <tr ID="riga_di_percorso" class='intestazione_tabella'>
   <td  align='center'>
      <i>qui verr&agrave; inserito il percorso seguito</i>
    </td>
  </tr>
  <tr ID="riga_centrale" width='30%'>
    <td>
       <table class="centrale" border="1" width='100%'>
       <tr>
        <td width='30%' align="left" valign="top" ID="colonna_link_principali">
         <table width='100%' border='0' align="left" valign="top">
            <tr>
             <td>
              <div class="riga_intestazione_link" align="center"><b>risorse
accessibili:</b></div>
             </td>
            </tr>
            <tr>
             <td>
              <?php
              //Elenca i nodi principali
             for ($i=0; $i < count($principale); $i++)
             {
              $risp = $principale[$i];
              $str = "<a href='perfectfit06.php?padre=".$risp-
>padre."&padre_nodo=".$risp->padre_nodo."&id_ist=".$risp-
>id_istanza_risorsa."'>";
          if ($risp->id_struttura_albero > 2)
               $str .= $risp->nome_nodo;
              else
               $str .= $risp->nome_istanza;
              $str .= "(".$risp->grado." risorse figlio)</a><br/>";
              echo $str;
             }
            ?> <br/>
             </td>
            </tr>
            <tr>
             <td>
              <div class="riga_intestazione_link" align="center"><b>risorse
interessanti:</b> </div>
             </td>
            </tr>
            <tr>
             <td>
            <?php
             //Generazione della lista delle risorse interessanti
             risorse_interessanti($profilo_utente_attuale, $lista_link);

             for ($i=0; $i < count($lista_link); $i++)
             {
              $risp = $lista_link[$i];
              $str = "<a href='perfectfit06.php?padre=".$risp-
>padre."&padre_nodo=".$risp->padre_nodo."&id_ist=".$risp-
>id_istanza_risorsa."'>";
          if ($risp->id_struttura_albero > 2)
               $str .= $risp->nome_nodo;
              else
               $str .= $risp->nome_istanza;
```

```
              //in questo caso il grado è quello di interesse
            $str .= " ";
              for ($cn=0 ; $cn <= (round( ( (100 - $risp->grado) /20) )-1);
$cn++)
               $str .= "*";
              echo $str."<br/>";
             }

         ?>
             </td>
          </tr>
         </table>
         </td>
         <td width="100%" id="colonna_foglie">
            <table border='0' width='100%'>
         <tr id='intestazione'>
              <td class="riga_intestazione_link" width='100%'>
              <div align="center">
              <b>
              <?php
           if (isset($_GET['id_ist']))
               {
                 $sql = "SELECT * FROM struttura_albero NATURAL JOIN
istanze_risorsa WHERE id_istanze_risorsa='".$_GET['id_ist']."'";
                    $result = mysql_query($sql);
                    $riga = mysql_fetch_array($result);
             echo $riga['nome_istanza']." - (".$riga['nome_nodo'].")";
               }
              ?>
              </b>
              </div>
              </td>
             </tr>
         <tr>
              <td>
               <?php
            if (isset($_GET['id_ist']))
                {
                   include "risorse/".$riga['contenuto'];
                }
           ?>
              </td>
             </tr>
            </table>

        <?php
             //Mostra l'elenco dei nodi secondari, ovvero delle risorse
accessibili all'interno dell pagina
             for ($idx=0; $idx < count($secondario); $idx++)
             {
          if ($risp = $secondario[$idx])
              {
               $riso = new profilo_risorsa;
               crea_profilo_risorsa_ideale($profilo_utente_attuale, $riso);
           $sql = "SELECT COUNT(*) AS NP FROM base_regole_associazione";
           $result = mysql_query($sql);
               $riga = mysql_fetch_array($result);
           $np = $riga['NP'];
               $str = " <table border=2 width='100%'>\n   <tr width='100%'>\n
<td bgcolor='#CCCCCC' WIDTH='40%' ALIGN='center'><b>".$risp-
>nome_nodo."</b>\n</td>\n  <td WIDTH='30%' ALIGN='center'>\n  <input
name='button' type=button ID='butt".$risp-
>id_istanza_risorsa."'onClick='toggle(\"ris".$risp-
```

```
>id_istanza_risorsa."\",\"butt".$risp-
>id_istanza_risorsa."\");aggiorna_profilo(\"ris".$risp-
>id_istanza_risorsa."\",";   $i = 0;
             $query = "SELECT * FROM base_regole_associazione";
           $result = mysql_query($query);
           while($riga = mysql_fetch_array($result))
           {
                $str .= calcola_regole_estensione($risp->id_struttura_albero,
$riga['profilo_utente_riferimento']);
              if ($i <= $np-1) $str .= ", ";               $i++;
           }
               if ((profilo_utente_definito($profilo_utente_attuale, 10)) &&
(differenza_risorse($risp->id_struttura_albero, $riso) <= 40))$str .= "$i);'
value='nascondi'>diff:".differenza_risorse($risp->id_struttura_albero, $riso);
         else
            $str .= "$i);' value='mostra'>diff:".differenza_risorse($risp-
>id_struttura_albero, $riso);
           echo $str."</td><td width='30%'>";
           echo "<div align='center'><a href='appunti.php?id_risorsa=".$risp-
>id_istanza_risorsa."' target='_blank' onclick='aggiorna_profilo(\"null\",";
             $i = 0;
             $query = "SELECT * FROM base_regole_associazione";
           $result = mysql_query($query);
           while($riga = mysql_fetch_array($result))
           {
                echo calcola_regole_estensione($risp->id_struttura_albero,
$riga['profilo_utente_riferimento']) * 10;
              if ($i <= $np-1) echo ", ";
                  $i++;
           }
                echo "$i)'>segna negli appunti</a></div></td></tr>\n  </td>\n
</tr></td>\n <table>\n";

              echo "<tr><td border='3' width='100%'";
              if ((profilo_utente_definito($profilo_utente_attuale, 10)) &&
(differenza_risorse($risp->id_struttura_albero, $riso) <= 40))              echo
" class='visibile' id='ris".$risp->id_istanza_risorsa."'>";
               else
                echo " class='nascosto' id='ris".$risp-
>id_istanza_risorsa."'>";
              $r = "risorse/".$risp->nome_file;
              include($r);
              echo "</td></tr></table>";
       }
           }
        ?>
       </td>
    </tr>
   </table>
  </td>
  </tr>
  <tr align="left" valign="top" id="riga_inferiore" width='100%'>
    <td>qui verrano inserite le informazioni sussidiarie </td>
  </tr>
</table>
</body>
</html>
```

Appendice E – Libreria di funzioni di PerfectFit06

La pagina **perfectfit06.php** dell'appendice D fa uso della libreria di funzioni PHP definita nella pagina **funzioni.php** di seguito riportata.

```
<?php
  //Classe che descrive un profilo risorsa
  class profilo_risorsa
  {
    var $topografica;
    var $tipologica;
    var $planimetrica;
      var $informativa;
    var $topografica;
    var $storica;
      var $documentaria;
    var $materica;
    var $decorativa;
      var $iconografica;
    var $destinazione_d_uso;
    var $esplorativa;
    var $azione_conservativa;
    var $tecnica;
    var $tecnologica;
  }

  //Classe per la definizione e la descrizione dei link riguardanti le risorse
  class rif
  {
      var $nome_file;
        var $id_istanza_risorsa;
        var $nome_istanza;
        var $id_struttura_albero;
        var $padre_nodo;
        var $padre;
        var $nome_nodo;
        var $grado;
  }

  /*
   questa funzione crea dinamicamente le regole di estensione utilizzate da
PerfectFit06 e le
   inserisce all'interno del database del sistema.
  */
  function crea_regole_estensione()
  {
    //Elimina le regole di estensione precedenti
      $sql = "DROP TABLE `regole_associazione`";
      $result = mysql_query($sql);

    //prepara la stringa per la creazione della tabell destinata alle regole di
estensione
    $stringa = "CREATE TABLE `regole_associazione` (`id_regole_associazione`
BIGINT NOT NULL AUTO_INCREMENT, `id_struttura_albero` BIGINT NOT NULL";

      //Carica le classi utente "pure" dalle regole di associazione utilizzate
per la generazione dinamica delle regole di estensione
      $sql = "SELECT * FROM base_regole_associazione";
```

```
    $result = mysql_query($sql);
    while ($riga = mysql_fetch_array($result))
     $stringa = $stringa.", `".$riga['profilo_utente_riferimento']."` SMALLINT
DEFAULT '0' NOT NULL";
    $stringa = $stringa.", PRIMARY KEY ( `id_regole_associazione` ));";
      $result = mysql_query($stringa);

    //Carica la struttura del grafo della conoscenza per definire le risorse del
sistema
    $sql = "SELECT * FROM struttura_albero";
    $result = mysql_query($sql);
    while ($riga = mysql_fetch_array($result))
    {
       //Prepara la stringa per l'inserimento di una riga all'interno del
database
       $stringa_i = "INSERT INTO `regole_associazione` ( `id_struttura_albero`";
       $stringa_f = "VALUES ('".$riga['id_struttura_albero']."'";

     $inner_sql = "SELECT * FROM base_regole_associazione";
     $inner_result = mysql_query($inner_sql);
     while ($inner_riga = mysql_fetch_array($inner_result))
       {
         $stringa_i = $stringa_i.",
`".$inner_riga['profilo_utente_riferimento']."`";
         //Calcola la singola regola di estensione per una data risorsa rispetto
alla specifica classe 'pura'
         $stringa_f = $stringa_f.",
'".calcola_regole_estensione($riga['id_struttura_albero'],
$inner_riga['profilo_utente_riferimento'])."'";
       }
       $stringa = $stringa_i.") ".$stringa_f.");";
       mysql_query($stringa);
      }
  }



  /*questa funzione calcola il numero di figli istanziati della risorsa corrente
  VAR:
  id_istanze_risorsa  : l'id dell'istanza della risorsa cui si vuole conoscere
il numero di figli
  id_struttura_albero : l'id del nodo padre nella struttura ad albero

  RETURN:
  il numero di istanze-figlio del nodo passato in ingresso
  */
  function ottieni_grado($id_istanze_risorsa)
  {
   //Calcolo grado
   $query="SELECT COUNT(*) AS NF FROM struttura_albero NATURAL JOIN
istanze_risorsa WHERE istanze_risorsa.id_istanza_padre
='".$id_istanze_risorsa."'";
   $result = mysql_query($query);
   $riga = mysql_fetch_array($result);
   $grado = $riga['NF'];
   return $grado;
  }

  /*
  La funzione analizza il nodo corrente e genera due vettori di istanze della
classe "rif".
  Il vetore 'principale' conterrà i riferimenti delle risorse che collegheranno
a nuove pagine (grado > 0)
```

```
  il vettore 'secondario' conterrà i riferimenti alle risorse foglia
  */
  function genera_lista_nodi($istanza, $padre_nodo, &$principale, &$secondario)
  {
   //Seleziono tutti i nodi figli del nodo corrente
   if ($istanza != "null")
    $query="SELECT * FROM struttura_albero NATURAL JOIN istanze_risorsa WHERE
istanze_risorsa.id_istanza_padre ='".$istanza."'";
   else
    $query="SELECT * FROM struttura_albero NATURAL JOIN istanze_risorsa WHERE
struttura_albero.padre IS NULL";

   $result = mysql_query($query);

   //Azzero gli array che andranno a contenere i link principali e quelli
secondari
   $principale = array();
   $secondario = array();
    //contatori per la gestione dei vettori
   $cn_p=0;
   $cn_s=0;

   //Scorro tutti i nodi figli del nodo corrente
   while ($riga = mysql_fetch_array($result))
   {
    $grado = ottieni_grado($riga['id_istanze_risorsa'],
$riga['id_struttura_albero']);

    $riferimento = new rif;
    if ($grado > 0)
    {
     //Tratto in maniera speciale i nodi-risorsa "Insulae" ed "Edificio"
       if (($riga['id_struttura_albero']==1) or
($riga['id_struttura_albero']==2))
     {
      $riferimento->id_istanza_risorsa = $riga['id_istanze_risorsa'];
      $riferimento->padre_nodo = $riga['id_struttura_albero'];
        $riferimento->padre      = $riga['id_istanze_risorsa'];
        $riferimento->nome_nodo  = $riga['nome_nodo'];
      $riferimento->grado      = $grado;
      $riferimento->id_struttura_albero = $riga['id_struttura_albero'];
      $riferimento->nome_istanza = $riga['nome_istanza'];
     }
     else
     {
      $riferimento->id_istanza_risorsa = $riga['id_istanze_risorsa'];
      $riferimento->padre_nodo = $riga['id_struttura_albero'];
        $riferimento->padre      = $riga['id_istanza_padre'];
        $riferimento->nome_nodo  = $riga['nome_nodo'];
      $riferimento->grado      = $grado;
      $riferimento->id_struttura_albero = $riga['id_struttura_albero'];
      $riferimento->nome_istanza = $riga['nome_istanza'];
     }
     $principale[$cn_p] = $riferimento;
     $cn_p++;
    }
    else
    {
     $riferimento->nome_file  = $riga['contenuto'];
     $riferimento->id_istanza_risorsa = $riga['id_istanze_risorsa'];
     $riferimento->id_struttura_albero = $riga['id_struttura_albero'];
     $riferimento->nome_nodo  = $riga['nome_nodo'];
     $secondario[$cn_s] = $riferimento;
```

```
     $cn_s++;
    }
   }
 }


  /*questa funzione inizializza il profilo passatogli, in questa implementazione
azzera tutti i campi
   VAR:
   profilo_utente_attuale: l'indirizzo del profilo utente da inizializzare
  */
function inizializza_profilo_utente(&$profilo_utente_attuale)
{
 $sql = "SELECT * FROM base_regole_associazione";
 $result = mysql_query($sql);
 $profilo_utente_attuale = array();
 while ($riga = mysql_fetch_array($result))
 {
  $profilo_utente_attuale[$riga['profilo_utente_riferimento']] = 0;
 }
}

/*Calcola dinamicamente una regola di estensione, ovvero quanto cambia il
profilo utente nel momento in cui viene selezionata una risorsa.

 VAR:
  id_struttura_albero: l'identificativo nel database della risorsa rispetto la
quale si sta calcolando la risorsa
  nome_profilo_utente_riferimento: la classe pura rispetto cui si calcola la
regola

*/
function calcola_regole_estensione($id_struttura_albero,
$nome_profilo_utente_riferimento)
{
  $sql = "SELECT * FROM base_regole_associazione WHERE
profilo_utente_riferimento ='".$nome_profilo_utente_riferimento."'";
  $result = mysql_query($sql);
  $modello_ideale_associato = mysql_fetch_array($result);

  $sql = "SELECT * FROM profilo_risorse WHERE
id_struttura_albero='".$id_struttura_albero."'";
  $result2 = mysql_query($sql);
  $modello_reale = mysql_fetch_array($result2);

  //inizializzazione della differenza
  $differenza = 0;

  //calcolo della regola di estensione.
  /*
   Se un campo del modello risorsa reale è minore dello stesso campo del modello
risorsa associato al modello utente di riferimento
   all'interno dell regola di associazione viene incrementata la distanza.
   Praticamente, quando un campo del modello reale "soddisfa" l'interesse
definito nella regola d'associazione nello stesso
   campo del modello ideale , la differenza non subisce variazioni
  */
  if ($modello_reale['topografica'] - $modello_ideale_associato['topografica'] <
0)          $differenza += abs($modello_ideale_associato['topografica'] -
$modello_reale['topografica']);
  if ($modello_reale['tipologica'] - $modello_ideale_associato['tipologica'] <
0)           $differenza += abs($modello_ideale_associato['tipologica'] -
$modello_reale['tipologica']);
```

```
  if ($modello_reale['planimetrica'] -
$modello_ideale_associato['planimetrica']< 0)         $differenza +=
abs($modello_ideale_associato['planimetrica'] - $modello_reale['planimetrica']);
  if ($modello_reale['informativa'] - $modello_ideale_associato['informativa']<
0)          $differenza += abs($modello_ideale_associato['informativa'] -
$modello_reale['informativa']);
  if ($modello_reale['storica'] - $modello_ideale_associato['storica']< 0)
             $differenza += abs($modello_ideale_associato['storica'] -
$modello_reale['storica']);
  if ($modello_reale['documentaria'] -
$modello_ideale_associato['documentaria']< 0)          $differenza +=
abs($modello_ideale_associato['documentaria'] - $modello_reale['documentaria']);
  if ($modello_reale['materica'] - $modello_ideale_associato['materica']< 0)
$differenza += abs($modello_ideale_associato['materica'] -
$modello_reale['materica']);
  if ($modello_reale['decorativa'] - $modello_ideale_associato['decorativa']< 0)
$differenza += abs($modello_ideale_associato['decorativa'] -
$modello_reale['decorativa']);
  if ($modello_reale['iconografica'] -
$modello_ideale_associato['iconografica']< 0)         $differenza +=
abs($modello_ideale_associato['iconografica'] - $modello_reale['iconografica']);
  if ($modello_reale['destinazione_d_uso'] -
$modello_ideale_associato['destinazione_d_uso']< 0)   $differenza +=
abs($modello_ideale_associato['destinazione_d_uso'] -
$modello_reale['destinazione_d_uso']);
  if ($modello_reale['esplorativa']-$modello_ideale_associato['esplorativa']<0)
$differenza += abs($modello_ideale_associato['esplorativa'] -
$modello_reale['esplorativa']);
  if ($modello_reale['azione_conservativa']-
$modello_ideale_associato['azione_conservativa']< 0)  $differenza +=
abs($modello_ideale_associato['azione_conservativa'] -
$modello_reale['azione_conservativa']);
  if ($modello_reale['tecnica']-$modello_ideale_associato['tecnica']< 0)
$differenza += abs($modello_ideale_associato['tecnica'] -
$modello_reale['tecnica']);
  if ($modello_reale['tecnologica']-$modello_ideale_associato['tecnologica']< 0)
$differenza += abs($modello_ideale_associato['tecnologica'] -
$modello_reale['tecnologica']);


  //Calcola quanti campi del modello risorsa ideale e reale sono stati
utilizzati nel calcolo della differenza tra le risorse
  $tot = 0;

  if ($modello_reale['topografica'] - $modello_ideale_associato['topografica'] <
0)          $tot++;
  if ($modello_reale['tipologica'] - $modello_ideale_associato['tipologica'] <
0)        $tot++;
  if ($modello_reale['planimetrica'] -
$modello_ideale_associato['planimetrica']< 0)      $tot++;
  if ($modello_reale['informativa'] - $modello_ideale_associato['informativa']<
0)        $tot++;
  if ($modello_reale['storica'] - $modello_ideale_associato['storica']< 0)
           $tot++;
  if ($modello_reale['documentaria'] -
$modello_ideale_associato['documentaria']< 0)         $tot++;
  if ($modello_reale['materica'] - $modello_ideale_associato['materica']< 0)
$tot++;
  if ($modello_reale['decorativa'] - $modello_ideale_associato['decorativa']< 0)
$tot++;
  if ($modello_reale['iconografica'] -
$modello_ideale_associato['iconografica']< 0)         $tot++;
```

```
  if ($modello_reale['destinazione_d_uso'] -
$modello_ideale_associato['destinazione_d_uso']< 0)   $tot++;
  if ($modello_reale['esplorativa']-
$modello_ideale_associato['esplorativa']<0)$tot++;
  if ($modello_reale['azione_conservativa']-
$modello_ideale_associato['azione_conservativa']< 0) $tot++;
  if ($modello_reale['tecnica']-$modello_ideale_associato['tecnica']< 0)
$tot++;
  if ($modello_reale['tecnologica']-$modello_ideale_associato['tecnologica']< 0)
$tot++;

  //Genera una proporzione tra la distanza calcolata e i campi utilizzati nel
suo calcolo
  if ($tot)
    $risultato = $differenza / $tot;
  else
    $risultato=0;

  //Ritorna il valore della regola di estensione secondo la formula seguente:
  $ret = round((100-$risultato)/10)-3;
  if ($ret >=0 )
   return $ret;
  else
   return 0;
}

/*
 Questa funziona genera l'elenco delle regole di estensione calcolate al fine
della stampa a video
*/
 function stampa_regole_estensione()
  {
    echo "<table border='1'><tr><td><b><div align='center' background='#CCCCCC'
color='#000033' font='16px'>nome risorsa</div></b></td>";
      $sql = "SELECT * FROM base_regole_associazione";
    $result = mysql_query($sql);
      $stringa="";
    while ($modello_ideale_associato = mysql_fetch_array($result))
     $stringa = $stringa."<td><b><div
align='center'>".$modello_ideale_associato['profilo_utente_riferimento']."</b></
div></td>";
    $stringa = $stringa."</tr>";
    echo $stringa;

    $sql = "SELECT * FROM struttura_albero";
    $result = mysql_query($sql);
      while ($modello_ideale_associato = mysql_fetch_array($result))
    {
       $stringa_i = "<tr><td>".$modello_ideale_associato['nome_nodo']."</td>";
       $stringa_f = "VALUES ('"."'";

     $inner_sql = "SELECT * FROM base_regole_associazione";
     $inner_result = mysql_query($inner_sql);
     while ($inner_riga = mysql_fetch_array($inner_result))
       {
         $stringa_i =
$stringa_i."<td>".calcola_regole_estensione($modello_ideale_associato['id_strutt
ura_albero'], $inner_riga['profilo_utente_riferimento'])."</td>";
       }
       echo $stringa_i."</tr>";

      }
  }
```

```
/*Crea la risorsa ideale partendo dal profilo risorsa attuale*/
function crea_profilo_risorsa_ideale($profilo_utente_attuale, &$risorsa_ideale)
{
 //Creazione del modello risorsa ideale e azzeramento dei campi
 $risorsa_ideale = new profilo_risorsa;

 $risorsa_ideale->topografica=0;
 $risorsa_ideale->tipologica=0;
 $risorsa_ideale->planimetrica=0;
 $risorsa_ideale->informativa=0;
 $risorsa_ideale->topografica=0;
 $risorsa_ideale->storica=0;
 $risorsa_ideale->documentaria=0;
 $risorsa_ideale->materica=0;
 $risorsa_ideale->decorativa=0;
 $risorsa_ideale->iconografica=0;
 $risorsa_ideale->destinazione_d_uso=0;
 $risorsa_ideale->esplorativa=0;
 $risorsa_ideale->azione_conservativa=0;
 $risorsa_ideale->tecnica=0;
 $risorsa_ideale->tecnologica=0;


 $sql = "SELECT COUNT(*) AS NP FROM base_regole_associazione";
 $result = mysql_query($sql);
 $regola_associazione = mysql_fetch_array($result);

 //Carica la tabella delle regole di associazione
 $sql = "SELECT * FROM base_regole_associazione";
 $result = mysql_query($sql);

 //inizializza un nuovo profilo utente che conterrà la normalizzazione di quello
attuale
 inizializza_profilo_utente($nuovo_profilo);
 normalizza_profilo_utente($profilo_utente_attuale, $nuovo_profilo);

 //il profilo utente normalizzato diventerà quello usato per i calcoli
 $profilo_utente_attuale = $nuovo_profilo;


 //Scansione delle regole di associazione
 while ($regola_associazione = mysql_fetch_array($result))
 //Viene verificato se il particolare campo del profilo utente è definito
 if
($profilo_utente_attuale[$regola_associazione['profilo_utente_riferimento']])
 {
  if ($risorsa_ideale->topografica <
ABS(($regola_associazione['topografica'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']])
   $risorsa_ideale->topografica =
ABS(($regola_associazione['topografica'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']];
  if ( $risorsa_ideale->tipologica <
ABS(($regola_associazione['tipologica'])/100)*$profilo_utente_attuale[$regola_as
sociazione['profilo_utente_riferimento']])
   $risorsa_ideale->tipologica =
ABS(($regola_associazione['tipologica'])/100)*$profilo_utente_attuale[$regola_as
sociazione['profilo_utente_riferimento']];
  if($risorsa_ideale->planimetrica <
ABS(($regola_associazione['planimetrica'])/100)*$profilo_utente_attuale[$regola_
associazione['profilo_utente_riferimento']])
```

```
  $risorsa_ideale->planimetrica =
ABS(($regola_associazione['planimetrica'])/100)*$profilo_utente_attuale[$regola_
associazione['profilo_utente_riferimento']];
  if($risorsa_ideale->informativa <
ABS(($regola_associazione['informativa'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']])
  $risorsa_ideale->informativa =
ABS(($regola_associazione['informativa'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']];
  if($risorsa_ideale->storica <
ABS(($regola_associazione['storica'])/100)*$profilo_utente_attuale[$regola_assoc
iazione['profilo_utente_riferimento']])
  $risorsa_ideale->storica =
ABS(($regola_associazione['storica'])/100)*$profilo_utente_attuale[$regola_assoc
iazione['profilo_utente_riferimento']];
  if($risorsa_ideale->documentaria <
ABS(($regola_associazione['documentaria'])/100)*$profilo_utente_attuale[$regola_
associazione['profilo_utente_riferimento']])
  $risorsa_ideale->documentaria =
ABS(($regola_associazione['documentaria'])/100)*$profilo_utente_attuale[$regola_
associazione['profilo_utente_riferimento']];
  if($risorsa_ideale->materica <
ABS(($regola_associazione['materica'])/100)*$profilo_utente_attuale[$regola_asso
ciazione['profilo_utente_riferimento']])
   $risorsa_ideale->materica =
ABS(($regola_associazione['materica'])/100)*$profilo_utente_attuale[$regola_asso
ciazione['profilo_utente_riferimento']];
  if($risorsa_ideale->decorativa <
ABS(($regola_associazione['decorativa'])/100)*$profilo_utente_attuale[$regola_as
sociazione['profilo_utente_riferimento']])
  $risorsa_ideale->decorativa =
ABS(($regola_associazione['decorativa'])/100)*$profilo_utente_attuale[$regola_as
sociazione['profilo_utente_riferimento']];
  if (  $risorsa_ideale->iconografica <
ABS(($regola_associazione['iconografica'])/100)*$profilo_utente_attuale[$regola_
associazione['profilo_utente_riferimento']])
  $risorsa_ideale->iconografica =
ABS(($regola_associazione['iconografica'])/100)*$profilo_utente_attuale[$regola_
associazione['profilo_utente_riferimento']];
  if($risorsa_ideale->destinazione_d_uso <
ABS(($regola_associazione['destinazione_d_uso'])/100)*$profilo_utente_attuale[$r
egola_associazione['profilo_utente_riferimento']] )
  $risorsa_ideale->destinazione_d_uso =
ABS(($regola_associazione['destinazione_d_uso'])/100)*$profilo_utente_attuale[$r
egola_associazione['profilo_utente_riferimento']] ;
  if ($risorsa_ideale->esplorativa <
ABS(($regola_associazione['esplorativa'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']])
   $risorsa_ideale->esplorativa =
ABS(($regola_associazione['esplorativa'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']];
  if($risorsa_ideale->azione_conservativa <
ABS(($regola_associazione['azione_conservativa'])/100)*$profilo_utente_attuale[$
regola_associazione['profilo_utente_riferimento']])
  $risorsa_ideale->azione_conservativa =
ABS(($regola_associazione['azione_conservativa'])/100)*$profilo_utente_attuale[$
regola_associazione['profilo_utente_riferimento']];
  if($risorsa_ideale->tecnica <
ABS(($regola_associazione['tecnica'])/100)*$profilo_utente_attuale[$regola_assoc
iazione['profilo_utente_riferimento']])
  $risorsa_ideale->tecnica =
ABS(($regola_associazione['tecnica'])/100)*$profilo_utente_attuale[$regola_assoc
iazione['profilo_utente_riferimento']];
```

```
  if($risorsa_ideale->tecnologica <
ABS(($regola_associazione['tecnologica'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']])
   $risorsa_ideale->tecnologica =
ABS(($regola_associazione['tecnologica'])/100)*$profilo_utente_attuale[$regola_a
ssociazione['profilo_utente_riferimento']];
 }
}


/*Calcola il profilo utente normalizzato prendendo come valore di riferimento
quello più alto tra i campi del modello utente
  passatogli come parametro

 VAR:
  profilo_iniziale: profilo utente iniziale
  profilo_finale: grazie al passaggio per indirizzo setta il profilo utente
normalizzato

*/
 function normalizza_profilo_utente($profilo_iniziale, &$profilo_finale)
 {
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);

  //Cerca il massimo valore per usarlo come massimo di riferimento
  $max = 0;
  while ($riga = mysql_fetch_array($result))
   {
     if ($max < $profilo_iniziale[$riga['profilo_utente_riferimento']])
        $max = $profilo_iniziale[$riga['profilo_utente_riferimento']];
   }

  //effettua la normalizzazione
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);
  if ($max)
  while ($riga = mysql_fetch_array($result))
   {
     $profilo_finale[$riga['profilo_utente_riferimento']] = 100 *
($profilo_iniziale[$riga['profilo_utente_riferimento']]/$max);
   }
 }

 //Calcola la differenza tra un modello risorsa reale e un modello risorsa
ideale
 function differenza_risorse($id_struttura_albero, $risorsa_ideale)
 {
  $sql = "SELECT * FROM (istanze_risorsa NATURAL JOIN profilo_risorse) WHERE
istanze_risorsa.id_struttura_albero ='".$id_struttura_albero."' GROUP BY
nome_istanza";

  $sql = "SELECT * FROM profilo_risorse WHERE id_struttura_albero
='".$id_struttura_albero."'";
  $result_ued = mysql_query($sql);
  $modello_risorsa_reale= mysql_fetch_array($result_ued);

  $differenza=0;
  $tot=0;

  //Verifica il numero di campi impostati del modello risorsa reale
  if ($modello_risorsa_reale['topografica']) $tot++;
  if ($modello_risorsa_reale['tipologica']) $tot++;
```

```
  if ($modello_risorsa_reale['planimetrica']) $tot++;
  if ($modello_risorsa_reale['informativa']) $tot++;
  if ($modello_risorsa_reale['storica']) $tot++;
  if ($modello_risorsa_reale['documentaria']) $tot++;
  if ($modello_risorsa_reale['materica']) $tot++;
  if ($modello_risorsa_reale['decorativa']) $tot++;
  if ($modello_risorsa_reale['iconografica']) $tot++;
  if ($modello_risorsa_reale['destinazione_d_uso']) $tot++;
  if ($modello_risorsa_reale['esplorativa']) $tot++;
  if ($modello_risorsa_reale['azione_conservativa']) $tot++;
  if ($modello_risorsa_reale['tecnica']) $tot++;
  if ($modello_risorsa_reale['tecnologica']) $tot++;

  $num=0;
  //Verifica il numero di campi impostati del modello risorsa reale
  if ($risorsa_ideale->topografica) $num++;
  if ($risorsa_ideale->tipologica) $num++;
  if ($risorsa_ideale->planimetrica) $num++;
  if ($risorsa_ideale->informativa) $num++;
  if ($risorsa_ideale->storica) $num++;
  if ($risorsa_ideale->documentaria) $num++;
  if ($risorsa_ideale->materica) $num++;
  if ($risorsa_ideale->decorativa) $num++;
  if ($risorsa_ideale->iconografica) $num++;
  if ($risorsa_ideale->destinazione_d_uso) $num++;
  if ($risorsa_ideale->esplorativa) $num++;
  if ($risorsa_ideale->azione_conservativa) $num++;
  if ($risorsa_ideale->tecnica) $num++;
  if ($risorsa_ideale->tecnologica) $num++;
  /*
    Se un campo modello risorsa reale è diverso da zero ed è minore del
rispettivo della risorsa ideale
    la differenza viene incrementata di una unità.
  */
  if ($modello_risorsa_reale['topografica'])
   if ($modello_risorsa_reale['topografica'] < $risorsa_ideale->topografica)
$differenza++;
  if ($modello_risorsa_reale['tipologica'])
   if ($modello_risorsa_reale['tipologica'] < $risorsa_ideale->tipologica)
$differenza++;
  if ($modello_risorsa_reale['planimetrica'])
   if ($modello_risorsa_reale['planimetrica'] < $risorsa_ideale->planimetrica)
$differenza++;
  if ($modello_risorsa_reale['informativa'])
   if ($modello_risorsa_reale['informativa'] < $risorsa_ideale->informativa)
$differenza++;
  if ($modello_risorsa_reale['storica'])
   if ($modello_risorsa_reale['storica'] < $risorsa_ideale->storica)
$differenza++;
  if ($modello_risorsa_reale['documentaria'])
   if ($modello_risorsa_reale['documentaria'] < $risorsa_ideale->documentaria)
$differenza++;
  if ($modello_risorsa_reale['materica'])
   if ($modello_risorsa_reale['materica'] < $risorsa_ideale->materica)
$differenza++;
  if ($modello_risorsa_reale['decorativa'])
   if ($modello_risorsa_reale['decorativa'] < $risorsa_ideale->decorativa)
$differenza++;
  if ($modello_risorsa_reale['iconografica'])
   if ($modello_risorsa_reale['iconografica'] < $risorsa_ideale->iconografica)
$differenza++;
  if ($modello_risorsa_reale['destinazione_d_uso'])
```

```
   if ($modello_risorsa_reale['destinazione_d_uso'] < $risorsa_ideale-
>destinazione_d_uso) $differenza++;
  if ($modello_risorsa_reale['esplorativa'])
   if ($modello_risorsa_reale['esplorativa'] < $risorsa_ideale->esplorativa)
$differenza++;
  if ($modello_risorsa_reale['azione_conservativa'])
   if ($modello_risorsa_reale['azione_conservativa'] < $risorsa_ideale-
>azione_conservativa) $differenza++;
  if ($modello_risorsa_reale['tecnica'])
   if ($modello_risorsa_reale['tecnica'] < $risorsa_ideale->tecnica)
$differenza++;
  if ($modello_risorsa_reale['tecnologica'])
   if ($modello_risorsa_reale['tecnologica'] < $risorsa_ideale->tecnologica)
$differenza++;
  //Realizza una proporzione per stabilire quanti campi del modello reale
soddisfano il modello ideale
  $finale = round(100-(($differenza / $tot)* 100));
  return $finale;
 }

 /*Setta una copia del profilo utente nella 'memoria' cookie per far si che il
codice JavaScript vi possa accedere*/
 function crea_cookie_profilo_utente(&$pu)
 {
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);
  while ($riga = mysql_fetch_array($result))
  {
setcookie($riga['profilo_utente_riferimento'],$pu[$riga['profilo_utente_riferime
nto']]);
  }
 }
 /*aggiorna il cookie del profilo utente in base al profilo passatogli*/
 function aggiorna_cookie_profilo_utente(&$pu)
 {
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);
  while ($riga = mysql_fetch_array($result))
  {
   $pu[$riga['profilo_utente_riferimento']] =
$_COOKIE[$riga['profilo_utente_riferimento']];
  }
 }
 /*E' una funzione booleana che definisce se il profilo utente si può
considerare
   ragionevolmente definito*/
 function profilo_utente_definito($pu, $limite)
 {
  $somma=0;
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);
  while ($riga = mysql_fetch_array($result))
  {
   $somma += $pu[$riga['profilo_utente_riferimento']];
  }
  if ($somma > $limite)
   return 1;
  else
   return 0;
 }
  /*Definisce un insieme di risorse interessanti raggruppandole in base alle
risorse-figlio, l'attuale implementazione è fittizia*/
 function risorse_interessanti($profilo_utente, &$lista_link)
```

```
 {
  $lista_link = array();
  $ris = new profilo_risorsa;
  crea_profilo_risorsa_ideale($profilo_utente, $ris);
   $sql = "SELECT *, (";
  $sql .=     "((ABS(AVG(topografica) - $ris->topografica) + (AVG(topografica) -
$ris->topografica))/2)";
  $sql .= " +  ((ABS(AVG(tipologica )  - $ris->tipologica) + (AVG(tipologica ) -
$ris->tipologica ))/2)";
  $sql .= " +  ((ABS(AVG(planimetrica) - $ris->planimetrica) +
(AVG(planimetrica)  - $ris->planimetrica))/2)";
  $sql .= " +  ((ABS(AVG(informativa ) - $ris->informativa ) + (AVG(informativa
) - $ris->informativa  ))/2)";
  $sql .= " +  ((ABS(AVG(storica     ) - $ris->storica     ) + (AVG(storica) -
$ris->storica))/2)";
  $sql .= " +  ((ABS(AVG(documentaria) - $ris->documentaria) +
(AVG(documentaria)  - $ris->documentaria))/2)";
  $sql .= " +  ((ABS(AVG(materica    ) - $ris->materica) + (AVG(materica) -
$ris->materica))/2)";
  $sql .= " +  ((ABS(AVG(decorativa  )  - $ris->decorativa) + (AVG(decorativa) -
$ris->decorativa))/2)";
  $sql .= " +  ((ABS(AVG(iconografica) - $ris->iconografica) +
(AVG(iconografica) - $ris->iconografica))/2)";
  $sql .= " +  ((ABS(AVG(destinazione_d_uso)  - $ris->destinazione_d_uso) +
(AVG(destinazione_d_uso)- $ris->destinazione_d_uso))/2)";
  $sql .= " + ((ABS(AVG(esplorativa) - $ris->esplorativa) + (AVG(esplorativa) -
$ris->esplorativa))/2)";
  $sql .= " + ((ABS(AVG(azione_conservativa) - $ris->azione_conservativa) +
(AVG(azione_conservativa) - $ris->azione_conservativa))/2)";
  $sql .= " + ((ABS(AVG(tecnica)  - $ris->tecnica) + (AVG(tecnica)  - $ris-
>tecnica))/2)";
  $sql .= " + ((ABS(AVG(tecnologica) - $ris->tecnologica) + (AVG(tecnologica) -
$ris->tecnologica))/2)";
  $sql .= ")/14 AS differenza FROM (istanze_risorsa NATURAL JOIN
profilo_risorse) WHERE istanze_risorsa.id_struttura_albero <> '1' GROUP BY
nome_istanza ORDER BY differenza ";
  $result_ued = mysql_query($sql);
  $idx=0;
  while ($riga= mysql_fetch_array($result_ued))
  {
    $sql = "SELECT * FROM istanze_risorsa WHERE
nome_istanza='".$riga['nome_istanza']."' AND id_struttura_albero='2'";
      $result = mysql_query($sql);
      $riga_ris = mysql_fetch_array($result);
      $riferimento = new rif;
    $riferimento->id_istanza_risorsa = $riga_ris['id_istanze_risorsa'];
    $riferimento->padre_nodo = 2;
      $riferimento->padre      = 2;
      $riferimento->nome_nodo  = $riga['nome_istanza'];
    $riferimento->grado      = $riga['differenza'];
    $riferimento->id_struttura_albero = 2;
    $riferimento->nome_istanza = $riga['nome_istanza'];
      $lista_link[$idx] = $riferimento;
      $idx++;
  }
 }
 ?>
```

Appendice F – Codice delle pagine per la visualizzazione dello stato di PerfectFit06

Codice della pagina **visualizza_profilo_utente.php** per visualizzare il profilo utente attuale.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<SCRIPT type='text/javascript'>
 function chiudi()
 {
  window.close();
 }
</script>
<?php
  session_start();
  require ('connessione.php');
  require ('funzioni.php');
?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Modello utente</title>
</head>
<body>
<?php
  $pu = array();
  aggiorna_cookie_profilo_utente($pu);
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);
  echo "<b>";
  while ($riga = mysql_fetch_array($result))
  {
   echo
$riga['profilo_utente_riferimento'].":".$pu[$riga['profilo_utente_riferimento']]
."<br/>";
  }
  echo "</b>";
?>
<br/><br/><br/><br/>
<table border=0 align='center'>
<tr ALIGN='center'><td>
<input name='button' type=button  value='chiudi' onclick='chiudi();'/>
</td></tr>
</table>
</body>
</html>
```

Codice della pagina **visualizza_profilo_utente_normalizzato.php** usato da PerfectFit06 per visualizzare una pagina col modello utente normalizzato ricavato da quello attuale.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<SCRIPT type='text/javascript'>
 function chiudi()
 {
  window.close();
 }
</script>
<?php
  session_start();
  require ('connessione.php');
  require ('funzioni.php');
?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Modello utente normalizzato</title>
</head>
<body>
<?php
  $pu = array();
  aggiorna_cookie_profilo_utente($pu);
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);

  inizializza_profilo_utente($nuovo_profilo);
  normalizza_profilo_utente($pu, $nuovo_profilo);
  $pu = $nuovo_profilo;


  echo "<b>";
  while ($riga = mysql_fetch_array($result))
  {
   echo
$riga['profilo_utente_riferimento'].":".$pu[$riga['profilo_utente_riferimento']]
."<br/>";
  }
  echo "</b>";
?>
<br/><br/><br/><br/>
<table border=0 align='center'>
<tr ALIGN='center'><td>
<input name='button' type=button  value='chiudi' onclick='chiudi();'/>
</td></tr>
</table>
</body>
</html>
```

Codice della pagina **visualizza_profilo_risorsa.php** utilizzato per mostrare il modello della risorsa ideale associato al modello utente normalizzato corrente.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<SCRIPT type='text/javascript'>
 function chiudi()
 {
  window.close();
 }
</script>
<?php
  session_start();
  require ('connessione.php');
  require ('funzioni.php');
?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Modello risorsa</title>
</head>
<body>
<?php
  $pu = array();
  aggiorna_cookie_profilo_utente($pu);
  $sql = "SELECT * FROM base_regole_associazione";
  $result = mysql_query($sql);

  inizializza_profilo_utente($nuovo_profilo);
  normalizza_profilo_utente($pu, $nuovo_profilo);
  $pu = $nuovo_profilo;

  $risorsa_ideale = new profilo_risorsa;
  crea_profilo_risorsa_ideale($pu, $risorsa_ideale);
  echo "<b>";
  echo "topografica=".$risorsa_ideale->topografica."<br/>";
  echo "tipologica=".$risorsa_ideale->tipologica."<br/>";
  echo "planimetrica=".$risorsa_ideale->planimetrica."<br/>";
  echo "informativa=".$risorsa_ideale->informativa."<br/>";
  echo "topografica=".$risorsa_ideale->topografica."<br/>";
  echo "storica=".$risorsa_ideale->storica."<br/>";
  echo "documentaria=".$risorsa_ideale->documentaria."<br/>";
  echo "materica=".$risorsa_ideale->materica."<br/>";
  echo "decorativa=".$risorsa_ideale->decorativa."<br/>";
  echo "iconografica=".$risorsa_ideale->iconografica."<br/>";
  echo "destinazione_d_uso=".$risorsa_ideale->destinazione_d_uso."<br/>";
  echo "esplorativa=".$risorsa_ideale->esplorativa."<br/>";
  echo "azione_conservativa=".$risorsa_ideale->azione_conservativa."<br/>";
  echo "tecnica=".$risorsa_ideale->tecnica."<br/>";
  echo "tecnologica=".$risorsa_ideale->tecnologica."<br/>";
  echo "</b>";
?>
<br/><br/><br/><br/>
<table border=0 align='center'>
<tr ALIGN='center'><td>
<input name='button' type=button  value='chiudi' onclick='chiudi();'/>
</td></tr>
</table>
</body>
</html>
```

Appendice G – Codice MySQL

Di seguito è riportato la descrizione del codice MySQL per generare le tabelle del database di PerfectFit06.

Il codice è stato realizzato con una operazione di dump nell'ambiente PHPMyAdmin[PHP-3], non sono stati inseriti i valori all'interno delle tabelle ma solo la struttura di quest'ultime.

```
# phpMyAdmin SQL Dump
# version 2.5.3
# http://www.phpmyadmin.net
#
# Host: localhost
# Generato il: 23 Apr, 2006 at 09:48 PM
# Versione MySQL: 4.0.15
# Versione PHP: 4.3.3
#
# Database : `ercolano`
#

# --------------------------------------------------------

#
# Struttura della tabella `appunti`
#

CREATE TABLE `appunti` (
  `id_appunti` bigint(20) NOT NULL auto_increment,
  `session_id` text NOT NULL,
  `id_istanze_risorsa` bigint(20) NOT NULL default '0',
  PRIMARY KEY  (`id_appunti`)
) TYPE=MyISAM AUTO_INCREMENT=66 ;

# --------------------------------------------------------

#
# Struttura della tabella `base_regole_associazione`
#

CREATE TABLE `base_regole_associazione` (
  `id_base_regole_associazione` bigint(20) NOT NULL auto_increment,
  `topografica` smallint(6) NOT NULL default '0',
  `tipologica` smallint(6) NOT NULL default '0',
  `planimetrica` smallint(6) NOT NULL default '0',
  `informativa` smallint(6) NOT NULL default '0',
  `storica` smallint(6) NOT NULL default '0',
  `documentaria` smallint(6) NOT NULL default '0',
  `materica` smallint(6) NOT NULL default '0',
  `decorativa` smallint(6) NOT NULL default '0',
  `iconografica` smallint(6) NOT NULL default '0',
  `destinazione_d_uso` smallint(6) NOT NULL default '0',
  `esplorativa` smallint(6) NOT NULL default '0',
  `azione_conservativa` smallint(6) NOT NULL default '0',
  `tecnica` smallint(6) NOT NULL default '0',
  `tecnologica` smallint(6) NOT NULL default '0',
  `profilo_utente_riferimento` varchar(255) NOT NULL default '',
  PRIMARY KEY  (`id_base_regole_associazione`)
```

```
) TYPE=MyISAM AUTO_INCREMENT=38 ;

# --------------------------------------------------------

#
# Struttura della tabella `istanze_risorsa`
#

CREATE TABLE `istanze_risorsa` (
  `id_istanze_risorsa` bigint(20) NOT NULL auto_increment,
  `nome_istanza` varchar(255) NOT NULL default '',
  `id_struttura_albero` bigint(20) NOT NULL default '0',
  `id_istanza_padre` bigint(20) default NULL,
  `contenuto` varchar(255) default 'blank.htm',
  PRIMARY KEY  (`id_istanze_risorsa`)
) TYPE=MyISAM AUTO_INCREMENT=68 ;

# --------------------------------------------------------

#
# Struttura della tabella `percorso_utente`
#

CREATE TABLE `percorso_utente` (
  `id_percorso_utente` bigint(20) NOT NULL auto_increment,
  `session_id` text NOT NULL,
  `id_risorsa` bigint(20) NOT NULL default '0',
  `ora` datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  (`id_percorso_utente`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;

# --------------------------------------------------------

#
# Struttura della tabella `profilo_risorse`
#

CREATE TABLE `profilo_risorse` (
  `id_profilo_risorse` bigint(20) NOT NULL auto_increment,
  `id_struttura_albero` bigint(20) NOT NULL default '0',
  `topografica` smallint(6) NOT NULL default '0',
  `tipologica` smallint(6) NOT NULL default '0',
  `planimetrica` smallint(6) NOT NULL default '0',
  `informativa` smallint(6) NOT NULL default '0',
  `storica` smallint(6) NOT NULL default '0',
  `documentaria` smallint(6) NOT NULL default '0',
  `materica` smallint(6) NOT NULL default '0',
  `decorativa` smallint(6) NOT NULL default '0',
  `iconografica` smallint(6) NOT NULL default '0',
  `destinazione_d_uso` smallint(6) NOT NULL default '0',
  `esplorativa` smallint(6) NOT NULL default '0',
  `azione_conservativa` smallint(6) NOT NULL default '0',
  `tecnica` smallint(6) NOT NULL default '0',
  `tecnologica` smallint(6) NOT NULL default '0',
  PRIMARY KEY  (`id_profilo_risorse`)
) TYPE=MyISAM AUTO_INCREMENT=30 ;

# --------------------------------------------------------
#
# Struttura della tabella `regole_associazione`
#

CREATE TABLE `regole_associazione` (
```

```
  `id_regole_associazione` bigint(20) NOT NULL auto_increment,
  `id_struttura_albero` bigint(20) NOT NULL default '0',
  `archeologo` smallint(6) NOT NULL default '0',
  `architetto` smallint(6) NOT NULL default '0',
  `storico` smallint(6) NOT NULL default '0',
  `antiquario` smallint(6) NOT NULL default '0',
  `turista` smallint(6) NOT NULL default '0',
  `artista` smallint(6) NOT NULL default '0',
  `restauratore` smallint(6) NOT NULL default '0',
  PRIMARY KEY  (`id_regole_associazione`)
) TYPE=MyISAM AUTO_INCREMENT=32 ;
# --------------------------------------------------------
#
# Struttura della tabella `risorse`
#
CREATE TABLE `risorse` (
  `idRisorsa` bigint(20) NOT NULL auto_increment,
  `nomeRisorsa` varchar(255) NOT NULL default '',
  `tipologiaRisorsa` bigint(20) NOT NULL default '0',
  PRIMARY KEY  (`idRisorsa`)
) TYPE=MyISAM AUTO_INCREMENT=5 ;
# --------------------------------------------------------
#
# Struttura della tabella `struttura_albero`
#
CREATE TABLE `struttura_albero` (
  `id_struttura_albero` bigint(20) NOT NULL auto_increment,
  `padre` bigint(20) default NULL,
  `nome_nodo` varchar(255) NOT NULL default '',
  PRIMARY KEY  (`id_struttura_albero`)
) TYPE=MyISAM AUTO_INCREMENT=30011 ;
```

Glossario specifico di perfectFit06

Classe pura: è una classe d'utente ideale, cioè con comportamenti perfettamente definiti e noti.

Distanza: valore che quantifica l'interesse di un utente rispetto a una data risorsa reale, maggiore è la distanza minore l'interesse dell'utente verso essa.

Modello risorsa: struttura dati rappresentante caratteristiche peculiari della risorsa.

Modello risorsa ideale (risorsa ideale): modello di una risorsa astratta, generata tramite le regole di associazione dal modello utente attuale.

Modello risorsa reale (risorsa reale): modello risorsa di una risorsa realmente esistente, cioè di un contenuto informativo presente nel sistema

Modello utente: struttura dati rappresentante caratteristiche peculiari dell'utente.

Modello utente attuale: modello utente durante l'interazione col sistema, quindi, con valori impostati in maniera specifica per un utente che realmente sta interagendo con PerfectFit06.

Modello utente normalizzato: è una normalizzazione del modello utente attuale realizzata prendendo come massimo di riferimento il valore più alto tra i campi del modello utente attuale.

Regola di associazione: regola che stabilisce il grado di interesse di una classe *pura* rispetto una caratteristica del modello risorsa.

Regola di estensione: regola che stabilisce in che quantità un campo del modello utente deve essere modificato se una risorsa viene selezionata.

Risorsa: contenuto informativo atomico gestito dal sistema.

Risorsa ideale (modello risorsa ideale): modello di una risorsa astratta, generata tramite le regole di associazione dal modello utente attuale.

Risorsa interessante: una risorsa reale la cui distanza dagli interessi di un utente è inferiore a una soglia stabilita come parametro del sistema.

Risorsa reale (modello risorsa reale): modello risorsa di una risorsa realmente esistente, cioè di un contenuto informativo presente nel sistema.

Regola di associazione: regola che stabilisce il grado di interesse di una classe *pura* rispetto una caratteristica del modello risorsa.

Regola di estensione: regola che stabilisce in che quantità un campo del modello utente deve essere modificato se una risorsa viene selezionata.

Bibliografia consultata

[Brusilovsky-1] P. Brusilovsky: **"Methods and Techniques of Adaptive Hypermedia."**, User Modeling and User-Adapted Interaction v.6, pag. 87-129 , 1996.

[Brusilovsky-2] P. Brusilovsky: **"Adaptive Hypermedia"**, User Modeling and User-Adapted Interaction v.11, pag. 87-110, 2001.

[Burattini-1] E. Burattini: **"Leggere e scrivere ipertesti"**, Editoriale scientifica, Napoli, 2001.

[Burattini-2] E. Burattini, F. Gaudino, L. Serino: **"Hypermedia knowledge acquisition and a BDI agent for navigation assistance. A case study: Herculaneum Excavations"**, European Conference on Cognitive Science '99, Siena, Ottobre 1999.

[De Bra-1] P. De Bra, A. Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, N. Stash: **"AHA! Adaptive Hypermedia Architecture"** , reading, http://citeseer.ist.psu.edu/, 2003.

[De Bra-2] H. Wu, E. de Kort, P. De Bra:**"Design issues for general-purpose adaptive hypemedia system"**, reading, http://citeseer.ist.psu.edu/, 2000.

[De Bra-3] P. De Bra, G.J. Houben, H. Wu: **"AHAM: A Dexter-based reference model for adaptive hypermedia"**, reading, http://citeseer.ist.psu.edu/, 1999.

[Decker-1] K. Decker, K. Sycara, M. Williamson**: "Intelligent Adaptive Information Agents**" , reading, http://citeseer.ist.psu.edu/, 1997.

[Ercolano-1] E. Burattini, F. Gaudino, L. Serino: **"Ipertesto multimediale degli scavi di Ercolano"**, ipertesto multimediale in Toolbook.

[Etzioni-1] O. Etzioni, D.S. Weld: "Intelligents agents on the Internet: Fact, Fiction e forecast", reading, http://citeseer.ist.psu.edu/, 1995

[Greenwood-1] D. Greenwood, M. Calisti: **"Engineering Web service – Agent integration"**, reading, http://citeseer.ist.psu.edu/, 2004

[Javascript-1] D. Goodman: **"JavaScript bible – gold edition"**, Hungry Minds, 2005.

[Kobsa-1] A. Kobsa: **"Generic User Modeling System"**, User Modeling and User-Adapted Interaction v.11, pag. 49-63, 2001.

[Newell-1] A. Newell, H. Simon: **"Human problem solving"**, Prentice Hall, Englewood Cliff, NJ, USA, 1972.

[Pagano-1] M. Pagano: **"Ercolano: itinerario archeologico ragionato"**, T&M, Napoli, 1997.

[Pazzani-2] M. J. Pazzani , D. Billsus: **"Adaptive Web Site Agents"** , reading, http://citeseer.ist.psu.edu/, 1998.

[Pazzani-3] M. J. Pazzani , D. Billsus: **"Learning Probabilistic User Models"** , reading, http://citeseer.ist.psu.edu/, 1997.

[PHP-5] T.Converse, J.Park: **"PHP5 and MySQL Bible"**, Wiley Publishing Inc., 2005.

[Quinn-1] E. Quinn, M. O'hEigeartaigh, P. Pathak:**"Effectively Deploying an Adaptive Web Based e-Learning System by Extending UM"**, 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eindhoven,The Netherlands, August 2004.

[Rojas-1] G. Rojas, V. Pelechano, J. Fons: **"Navigational Properties and User Attributes for Modelling Adaptive Web Applications"**, Proceeding of AH 2000, Trento, I Agosto 2000.

[Rossi-1] S. Rossi, **"Agenti intelligenti per la navigazione in un sistema ipermediale su Web"**, Tesi del corso di Laurea in Fisica dell'Università degli studi di Napoli "Federico II", 2001.

[Tasso-1] C. Tasso , P. Omero: **"La personalizzazione dei contenuti Web"**, Franco Angeli, 2002

[Totò-1] regia C. Matrocinque, sceneggiatura E. Anton: **"Totò Peppino e la … malafemmina"**, film, 1956.

[Visciola-1] M. Visciola : **“Visibilità dei siti Web”**, Apogeo, 2000

[Weiss-1] G. Weiss: **“Multiagent System”**, The MIT press, 2000

Bibliografia di riferimento

[Belkin-1] N.J. Belkin, W.B. Croft: **"Information filtering and information retrieval: two sides of the same coin?"**, Communication of the ACM 35(12), 1992, 29-38.

[Benway-1] J.P. Benway: **"Banner blidness: The irony of attention grabbing on the World Wide Web"**, proceedings of the Human Factors and Ergonomic Society, 42nd Annual Meeting, 1998.

[Bradshawn-1] J. Bradshawn: **"Software Agent"**, AAAI/MIT press, Cambridge (MA), 1997.

[Brusilovsky-3] P. Brusilovsky, T. Santic, P. De Bra: **"A flexible layout model for a Web-based Adaptive Hypermedia Architecture"**, Proceeding of the AH2003 workshop, Budapest, Hungary 2003.

[Card-1] S. Card, T. Moran, A. Newell:**"The psicology of Human Computer Interaction"**, Hillsdale, NJ, Erlbaum Associates, 1983.

[Carrol-1] J. Carroll, M.B. Rosson: "**The Paradox of the Active User",** Interfacing Thought : Cognitive Aspects of Human-Computer Interaction, Cambridge, MA, MIT Press, 1987

[Casasola-1] E. Casasola, S. Gauch: **"Intelligent Information Agents for the World Wide Web"**, Information and Telecomunication Technology Center, Thecnical Report, 1997.

[Diaz-1] O. Diaz, N.W. Paton: **"Active Database System"**, ACM Computing Surveys 31, 1999.

[Eirinaki-1] M. Eirinaki, M. Vazirgiannis:**"Web Mining for Web Personalization"**, ACM Transactions on Internet Technology, 2003.

[Folz-1] P.W. Folz, S.T. Dumais : **"Personalized information delivery : an analysis of information filtering methods"**, Communication of the ACM 35(12), 1992.

[Garrigòs-1] I. Garrigós, J. Gómez, C. Cachero: **"Modelling Dynamic Personalization in Web Applications"**, 3rd International Conference on Web Engineering (ICWE'03), LNCS, Oviedo, Spain,2003.

[Genesereth-1] M. Genesereth, S. Katchpel: **"Software agents"**, Communications of the ACM37 , 147, 1994.

[Harley-1] R. Harley:**"Why most web sites fails"**, Forrester Research Report. Report September 1999.

[Hothi-1] J.Hothi, W. Hall: **"An evalutation of adapted hypermedia techniques using static user model"**, proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, 1998

[Kaplan–1] C. Kaplan, J. Fenwick, J. Chen : **"Adaptive hypertext navigation based on user goals and context"**, User Models and User Adapted Interaction 3(3), 1993.

[Malone-1] T. Malone, K. Grant, F. Turbak, S. Brobst, M. Cohen: **"Intelligent information shrink system"**, Communication of the ACM 43(8), 1987.

[Munoz-1] L.S. Munoz, J.P.M. de Oliveira: **"Adaptive Web-Based Courseware Development using Metadata Standards and Ontologies"**, CAiSE Conference, Riga, Latvia, June 2004.

[Nielesen-1] J. Nielsen, **"Web Usability"**, Edizione italiana: Apogeo, 2000

[Newell-1] A. Newell, H. Simon: **"Human problem solving"**, Prentice Hall, Englewood Cliff, NJ, USA, 1972.

[Norman-1] D. Norman:**"The psychology of everyday thing"**, Basic Book Perseus, New York, 2002.

[Norman-2] D. Norman:**"The invisible computer invisibile",** MIT Press, Cambridge MA, 1998.

[Pazzani-1] M. J. Pazzani , J. Muramatsu, D. Billsus: **"Syskill & Webert: Identifying interesting web sites"**, Proceedings of the National Conference on Artificial Intelligence, Portland, OR,1996.

[Pearl-1] J. Pearl: **"Fusion, propagation, and structuring in belief networks"**, Artificial Intelligence 29, 1986.

[Perkowitz-1] M. Perkowitz, O. Etzioni: **"Adaptive Web Sites: Automatically Synthesizing Web Pages."**, In Proceedings of AAAI98, 1998

[Rich-1] E. Rich: **"Users are Individuals:Individualizing Users Models"**, International Journal of Man-Machines studies 18, 1983.

[Shoshana-1] L. Shoshana: **"Architecting personalised delivery of multimedia information"**, Communication of the ACM 35(12), 1992.

[Tasso-2] M. Minio, C. Tasso: **"User Modeling for information filtering on Internet Services:exploiting an extended version of the UMT shell"**, UM96 workshop on User Modeling for information filtering on the World Wide Web", Kailua-Kona, Hawaii, USA, Gennaio 1996.

[Terveen-1] L. Terveen, W. Hill: **"Beyond raccomender system: Helping people help each other"**, Addison-Wesley, reading, MA, USA 2001.

[Thomas-1] C.G. Thomas, G. Fischer: **"Using Agents to Improve the Usability and Uselfulness of the WWW"**, Fifth International Conference on User Modeling, UM-96, 1996 Hawaii.

[Wexselblat-1] A. Wexelblat, P. Maes: **"Using History to Assist Information Browsing"**, RIAO'97: Computer-Assisted Information Retrieval on the Internet, Montreal, 1997.

[Wooldridge-1] Wooldridge, Jennings: **"Intelligent Agents: Theory and Pratice"**, Knowledge Engeneering Review, January 1995.

[Ziegler-1] G. Szwillus, J. Ziegler: **"From adaptive hypermedia to adaptive Web"** ,2003.

Siti web consultati

[Adapt-1] http://adaptweb.sourceforge.net/

[AH-1] http://ah2000.itc.it/

[AH-2] www.ah2004.org

[AH-3] http://polux.lcc.uma.es/AH2002/index.html

[Amazon-1] www.amazon.com

[Browser-1] http://browsershots.org

[Citeseer-1] http://citeseer.ist.psu.edu/

[Ercolano-2] www.ercolano.unina.it

[Google-1] www.google.it

[HCI-1] www.hcibib.org

[HCI-2] http://hci-journal.com

[HCI-3] www.hcii.cmu.edu

[HCI-4] http://hci.stanford.edu

[Htext-1] www.ht04.org

[Html-1] www.html.it

[Iota-1] http://www.iota.org/Winter99/recommend.html

[Java-1] http://java.sun.com/

[JavaScript-2] http://javascript.internet.com/

[JavaScript-2] www.javascript.com

[Lee-1] http://www.sciam.com/2001/0501issue/0501berners-lee.html

[Manuali-1] www.manuali.it

[Moviefinder-1] www.moviefinder.com

[Mozilla-1] www.mozilla.org

[Mozilla-2] www.mozillaitalia.org

[MSN-1] www.msn.it

[PHP-1] http://freephp.html.it

[PHP-2] www.easyphp.org

[PHP-3] www.phpmyadmin.net

[PHP-4] www.php.net

[Pompei-1] www.pompeiisites.org

[SemWeb-1] www.semanticweb.org

[Umuai-1] http://www.hcirn.com/res/period/umuai.php

[Umuai-2] www.umuai.org

[Umuai-3] http://gate.ac.uk/conferences/um2005/um05.html

[usabilità-1] www.usabilityfirst.com

[usabilità-2] www.usabile.it

[usabilità-3] www.usableweb.com

[usabilità-4] www.useit.com

[W3C-1] www.w3.org

[Wikipedia-1] http://en.wikipedia.org/wiki/