



Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di laurea

**Progettazione
di un Service Recommender System
basato su Intelligenza Collettiva**

Relatore: Prof. Muffatto Moreno

Correlatore: Ing. Ruffatti Gabriele

Laureando: Zanchetta Marco

Anno Accademico 2010 - 2011

Desidero ringraziare il Professor Muffatto, non solo per il suo ruolo di relatore, ma anche per essere stato, nel corso dei due anni di laurea magistrale, un professore fortemente stimolante. Ringrazio la Engineering, sia la sede di Padova che quella di Roma, per avermi fornito un argomento di tesi senza dubbio interessante e avermi concesso la collaborazione di persone di valore. Due ringraziamenti speciali spettano quindi all'Ing. Ruffatti e a Debora Desideri. Il primo per le numerose ore dedicate alla mia tesi, nonché per aver sopportato in più occasioni l'ostinazione con cui ho affermato le mie idee. La seconda per l'attento e lungo lavoro di revisione svolto e per la piena disponibilità che mi ha sempre dimostrato.

*A me stesso,
per tutte le ore di studio,
per le numerose rinunce,
per le notti insonni.*

Sommario

Nell'ultimo decennio Internet ha subito un rapido e inaspettato sviluppo, raggiungendo livelli di penetrazione imprevedibili. Di pari passo con questo trend, molte tipologie di servizi sono state rese accessibili anche tramite la Rete, così servizi di prenotazione, di ricerca e di acquisti, ma anche Web App e servizi di Cloud Computing, hanno fatto la comparsa sul Web. Questa situazione ha determinato la moltiplicazione dei servizi disponibili e una conseguente difficoltà negli utenti a selezionare quelli più adatti a loro.

I Recommender System sono sistemi che filtrano gli oggetti presenti in un itemset, fornendo risultati personalizzati in base alle preferenze dei singoli utenti. Appare evidente come la funzione di questi sistemi ben si concili con il problema della sovrabbondanza di servizi.

L'obiettivo della tesi è dunque quello di determinare un approccio nuovo, ma soprattutto innovativo, per progettare un Recommender System che suggerisca i servizi agli utenti, in modo efficiente e personalizzato.

Un argomento che sta attirando la curiosità dei laboratori di ricerca di tutto il mondo è l'intelligenza collettiva, cioè la capacità di una comunità di effettuare un task o di risolvere un problema in maniera più efficace ed efficiente di quanto potrebbero fare i singoli individui, attraverso la collaborazione e la condivisione di conoscenze. Pur esistendo già molti esempi di utilizzo, le reali potenzialità dell'intelligenza collettiva sono ancora in fase di studio.

La proposta avanzata in questa tesi, dunque, consiste nell'applicazione dell'intelligenza collettiva ai Recommender System, con particolare riferimento all'ambito dei servizi.

Nel capitolo 1 si introduce il contesto tecnologico in cui si inserisce l'argomento della tesi; il capitolo 2 descrive l'evoluzione del mondo dei servizi, partendo da quelli tradizionali e arrivando alla cosiddetta Internet dei Servizi; il capitolo 3 e 4 fanno un punto sullo stato dell'arte, rispettivamente, dei Recommender System e della Collective Intelligence; il capitolo 5 presenta la proposta del Service Recommender System basato su intelligenza collettiva (SeReS CI); infine, nel capitolo 6, viene illustrato un esempio di applicazione del sistema progettato.

Indice

1	Introduzione	1
2	L'Internet dei servizi	5
2.1	I Servizi	5
2.1.1	Storia dei servizi	5
2.1.2	Cos'è un servizio	6
2.1.3	Definizione di servizio NEXOF-RA	7
2.1.4	Caratteristiche peculiari dei servizi	8
2.2	I Web Services	10
2.2.1	SOA	11
2.2.2	Web services e SOA	13
2.3	Dalla SOA all'Internet dei Servizi	15
2.3.1	L'avvento dell'Internet dei Servizi	15
2.3.2	Gli obiettivi dell'IoS	18
2.3.3	I campi di ricerca	22
3	I Recommender System	27
3.1	Introduzione ai Recommender System	27
3.1.1	Definizione di Recommender System	28
3.1.2	Caratteristiche di un Recommender System	28
3.1.3	Esempi di utilizzo di Recommender System	32
3.2	Analisi comparativa di Recommender System	34
3.2.1	Registrazione	34
3.2.2	Personalizzazione del profilo	36
3.2.3	I suggerimenti	38
3.2.4	Partnership	41
3.2.5	Problematiche	41
3.2.6	Analisi schematica delle caratteristiche dei tre RS	42
3.3	Stato dell'arte	43
3.3.1	Classificazione dei Recommender System	43
3.3.2	Semantic Recommender System	44
3.3.3	Social Recommender System	49
3.3.4	Mobile recommender system	51
3.3.5	Algoritmi e metodi per i Recommender System tradizionali	53
3.3.6	Metriche di valutazione dei Recommender System	57
3.3.7	Le questioni ancora aperte	59

3.4	I Service Recommender System	60
3.4.1	Caratteristiche dei Recommender System per i servizi	60
3.4.2	Stato dell'arte	61
4	La Collective Intelligence	65
4.1	Cos'è la Collective Intelligence	65
4.1.1	Le origini della Collective Intelligence	65
4.1.2	La Collective Intelligence nell'era di Internet	66
4.2	Tre livelli di Collective Intelligence	68
4.2.1	Livello 0: CI consapevole ed esplicita	69
4.2.2	Livello 1: CI consapevole ed implicita	71
4.2.3	Livello 2: CI inconsapevole ed implicita	73
4.3	Panoramica sulla Collective Intelligence	79
4.3.1	Le sorgenti di Collective Intelligence	80
4.3.2	Le componenti motivazionali nel contribuire alla CI	82
4.3.3	Come si forma la Collective Intelligence	82
4.4	La Collective Intelligence nell'ambito dei servizi	87
4.4.1	La descrizione di servizi	87
4.4.2	La ricerca e la selezione dei servizi	89
5	Proposta di un Recommender System basato su CI	91
5.1	Contesto	91
5.2	Un Service Recommender System basato su CI	93
5.2.1	Un Recommender System per i servizi	93
5.2.2	Il cervello del Recommender System: l'intelligenza collettiva	93
5.2.3	Architettura del sistema	94
5.3	Caratteristiche del sistema	96
5.3.1	Tipologie di utenti	96
5.3.2	Il Web Site	98
5.3.3	Il Widget	100
5.3.4	Facebook	101
5.3.5	Mobile App	103
5.3.6	La Collective Intelligence Platform	104
5.4	Descrizione del Recommender System	107
5.4.1	Un Vector Space Model Esteso	107
5.4.2	I dataset degli Utenti, dei Servizi e dei Service Provider	108
5.4.3	Vocabolari	108
5.4.4	Rappresentazione degli utenti	109
5.4.5	Rappresentazione dei servizi	109

5.4.6	Rappresentazione dei Service Provider	110
5.4.7	I feedback degli utenti	111
5.4.8	Le competenze della Collective Intelligence Platform	113
5.4.9	Il Recommender Engine	128
5.4.10	Proposizione dei suggerimenti	132
5.5	Possibili estensioni	135
6	Esempio di utilizzo di SeReS CI	137
6.1	Scenario di applicazione	137
6.1.1	Caratteristiche del contesto	137
6.1.2	Analisi del sistema esistente	139
6.2	Travel SeReS CI	144
6.2.1	Assunzioni	144
6.2.2	Definizione del sistema	145
6.2.3	Registrazione utente	146
6.2.4	Personalizzazione del profilo	146
6.2.5	Il sito web rinnovato	146
6.2.6	I canali addizionali	148
6.3	Analisi comparativa	149
6.3.1	Il confronto	150
6.3.2	Vantaggi e limitazioni	151
	Conclusioni	153
	Appendice 1	155
	Appendice 2	161
	Bibliografia	167

CAPITOLO 1

Introduzione

Ogni ambito della Società moderna sta mutando verso la sua forma digitale. Questo cambiamento riguarda tutti i campi della vita quotidiana: dalla salute ai trasporti, dall'informazione alla cultura, dai beni di prima necessità ai passatempi, e l'elenco potrebbe continuare. Dagli inizi degli anni Duemila, in particolare, si è assistito ad un progressivo passaggio della fruizione dei principali media verso Internet: telefonia, reti televisive, giornali sono ora disponibili sul Web. Internet, dunque, è diventata una presenza fissa nella vita di tutti i giorni e svolge sempre più un ruolo cruciale nella comunicazione tra gli esseri umani.

Hal Varian e Peter Lyman, docenti dell'Università della California a Berkeley, hanno calcolato[1], prima nel 2000 e poi nuovamente nel 2003, la produzione annuale di informazione, attraverso qualsiasi mezzo e supporto, da quello cartaceo ai dischi, dalla TV a Internet, fino ai sistemi di telecomunicazione. La stima per il 2000 era stata di 1.5 exabytes, dove un exabyte corrisponde a un miliardo di miliardi di bytes¹. Tre anni dopo, nel 2003, la produzione era salita a 3.5 exabytes, con un tasso di crescita annuo del 66%.

Nel 2008 è stato fatto uno studio, da parte della International Data Corporation (IDC) [2], nel quale si afferma che la sola informazione prodotta in formato digitale nell'anno 2007 è stata pari a 281 exabytes, cioè 80 volte quella generata con tutti i possibili mezzi solo 4 anni prima. In particolare, è interessante notare come la produzione delle cosiddette "digital shadows", cioè le informazioni generate dagli utenti semplicemente navigando sul web, sono risultate in quantità superiore ai contenuti creati volontariamente dagli stessi.

Il motivo di tale incremento vertiginoso non è unico, ma dipende da molteplici fattori. Bisogna innanzitutto considerare che il numero di soggetti che generano informazione cresce continuamente grazie alla proliferazione di strumenti di natura elettronica e informatica. Attraverso tali strumenti aumentano, infatti, le possibilità di produrre nuove aggregazioni di informazioni. Si può osservare, poi, che grandi quantitativi di informazione sono prodotti dal social networking, cioè dall'interazione attraverso le reti di milioni di persone. Infine ci sono da considerare le informazioni prodotte in automatico da strumenti di rilevamento, come ad esempio i sensori ambientali, o da strumenti autonomi di interazione su internet, come i bot, ossia agenti software che possono interagire autonomamente con altri agenti software.

Tale esplosione nella produzione di dati è strettamente legata anche all'evoluzione di Internet nel tempo, caratterizzata dal passaggio dalla condivisione di contenuti del Web 1.0, alla creazione di contenuti da parte degli utenti del Web 2.0, fino alla produzione di contenuti in collaborazione del Web 3.0. In quest'ultimo ambito si va affermando la cosiddetta Collective Intelligence (CI), cioè l'informazione utile che si può generare a partire dalle conoscenze messe a disposizione dai singoli individui. Si tratta dunque di una sorta di intelligenza artificiale, che in un'accezione più ampia comprende tra le sue origini tutti i dati presenti in Internet, infatti tutto ciò che troviamo nel web è espressione di preferenze, modi di essere,

¹ Per avere un'idea, 1.5 exabytes corrispondono a 37.000 volte il contenuto di informazione della intera Library of Congress di Washington, ovvero della più grande biblioteca al mondo con oltre 29 milioni di libri, 2.7 milioni di registrazioni, 12 milioni di foto, 4.8 milioni di mappe e 58 milioni di manoscritti, il tutto distribuito su 530 migliaia di scaffali.

idee, stati d'animo, e conoscenze. È ovvio che la mole di informazioni disponibile in Rete necessita di una organizzazione per non rimanere una massa informe di dati non significativi. Il Web 3.0, quindi, sarà un web semantico, cioè un ambiente dove i documenti pubblicati sono associati a metadati che ne specificano il contesto semantico in un formato adatto all'interrogazione, all'interpretazione e, più in generale, all'elaborazione automatica.

I molteplici cambiamenti che stanno avvenendo nella Rete hanno inevitabilmente posto problemi di non semplice risoluzione. Internet, come visto, sta crescendo a ritmi altissimi, ben oltre le aspettative originali, e gli utenti manifestano necessità crescenti per prestazioni, sicurezza e affidabilità, che non sempre possono essere soddisfatte. Inoltre, sempre più spesso ci si trova a dover fare i conti con dei limiti tecnologici dovuti alla sua attuale architettura.

Per affrontare l'Internet del futuro, la cosiddetta Future Internet (FI) e le sfide correlate, le società ICT appartenenti all'European Technology Platforms (eMobility, NEM, NESSI, ISI, e EPoSS) hanno istituito un gruppo di lavoro comune (Cross-ETP Future Internet) con l'obiettivo di definire una visione comune e formulare suggerimenti per la ricerca europea dei prossimi 10 anni. Ciò che ne è scaturito [3] è una struttura che si fonda su quattro pilastri principali: *Internet by and for People*, *Internet of Contents and Knowledge*, *Internet of Things* e *Internet of Services*.

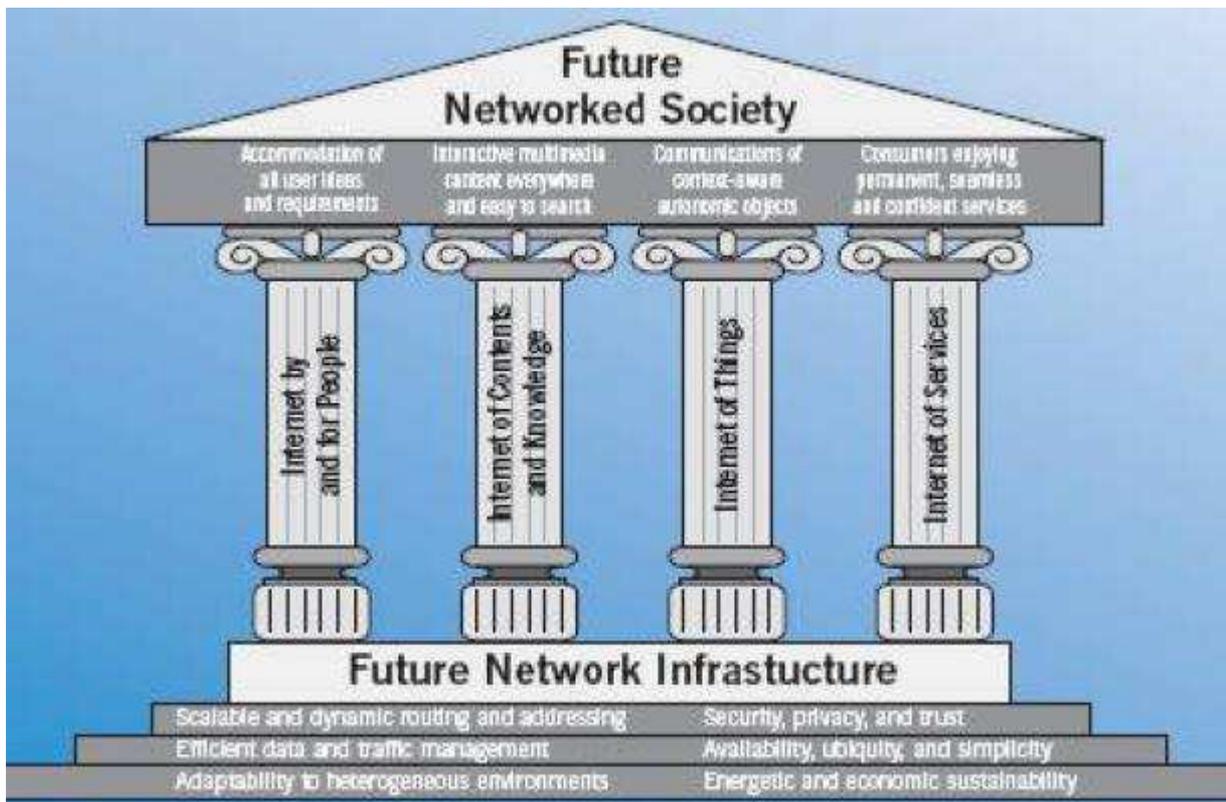


Figura 1.1: La Future Internet secondo la ETP Vision [3]

Con il primo termine si vuole indicare una infrastruttura in grado di soddisfare le esigenze e le aspettative delle persone comuni, preservando la possibilità di autocontrollo delle stesse sulle proprie attività online e sostenendo il libero scambio di idee. La FI deve anche essere in grado di facilitare la vita di tutti i giorni delle singole persone, delle comunità e delle organizzazioni; deve permettere la creazione di qualsiasi tipo di business, senza limiti di dimensione, dominio o tecnologia; deve superare le barriere esistenti tra produttori e consumatori di informazione e,

parallelamente, deve favorire l'emersione dei *prosumer*², cioè quella categoria di persone o comunità, che fanno parte sia della fase di produzione di contenuti, sia della fase di consumo. Con il ruolo sempre più centrale della comunicazione digitale all'interno della nostra società, si sta superando la fase del semplice accumulo e consumo dell'informazione, e si sta coinvolgendo sempre più attività intellettuali consapevoli, come il pensiero e l'apprendimento. Questo aspetto della FI viene espresso con il termine Internet of Contents and Knowledge. A tale scopo Internet deve supportare meccanismi di diffusione della conoscenza a livello locale ma anche globale, e fornire, oltre all'accesso alle informazioni, anche adeguati mezzi di elaborazione delle stesse. L'evoluzione del Web verso il Web 3.0 introdurrà una intelligenza conoscitiva, che permetterà alle applicazioni Web di elaborare in modo intelligente le informazioni. Il tagging delle informazioni in modo semantico sarà la base per il raggiungimento di questa nuova frontiera.

L'espressione Internet of Things (IoT), invece, sembra suggerire scenari avveniristici, nei quali gli oggetti hanno comportamenti propri e compiono azioni. In un futuro prossimo ci si attende che esista un modo per identificare univocamente ogni oggetto che ci circonda, e non solo computer, stampanti e cellulari, come già accade al giorno d'oggi. Sarà inoltre possibile l'identificazione reciproca tra gli stessi oggetti, che potranno quindi comunicare, scambiandosi informazioni, qualsiasi natura essi abbiano. In quest'ottica l'IoT può essere vista come una rete globale di oggetti interconnessi e univocamente indirizzabili, basati su un protocollo di comunicazione comune. Già oggi la varietà di dispositivi dotati di accesso a Internet è ampia, ma l'Internet of Things, sarà popolata da dispositivi con un grado di eterogeneità decisamente più vasto. In tale contesto assumeranno un ruolo centrale tecnologie come sensori e RFID (Radio-frequency Identification).

Il termine Internet of Services (IoS), infine, è un contenitore che racchiude numerosi fenomeni di interazione, a diversi livelli. Vi è infatti un esteso set di applicazioni Internet che includono ambienti pervasivi e immersivi, industriali e manifatturieri, veicolari e logistici, finanziari e di eBusiness, applicazioni eHealth e eGovernment. In questo ambito si vuole quindi capire come saranno forniti e gestiti i servizi in Internet nel futuro. I tre maggiori domini di sviluppo sono: la creazione di un Internet orientata ai servizi e scalabile, cioè che permette l'accesso a risorse fisiche, dati o a funzionalità software nella forma di servizi, senza limiti di dimensione, come già accade nel caso del Cloud Computing; la diffusione di servizi contestualizzati e proattivi; l'organizzazione del panorama dei servizi.

Per capire meglio il concetto di servizi contestualizzati e proattivi, possiamo analizzare alcune parole chiave che ne descrivono le caratteristiche. Il primo concetto chiave è la "*context awareness*", cioè la dipendenza dal contesto. L'interazione diventerà totalmente personalizzata e varierà in base alle situazioni, tenendo conto delle preferenze dell'utente, della storia d'uso, della tipologia di mezzo d'accesso utilizzato, del luogo e perfino dell'orario. Un altro concetto chiave è il termine "*seamless*", cioè la varietà dei dispositivi che possono essere utilizzati per interagire. In un tale contesto, infatti, gli utenti devono poter selezionare dinamicamente il tipo di interazione più appropriato, e quindi gli sviluppatori devono fornire un'interfaccia utente efficace per qualsiasi modalità venga scelta. Il terzo concetto chiave è l'"*end user empowerment*", ossia l'azione autonoma degli utenti finali. Sarà più facile per gli utenti scegliere da soli i servizi, e configurare autonomamente le interfacce d'accesso, tramite un mash up di risorse disponibili, ricercabili in un catalogo o in uno store di servizi. L'ultimo concetto chiave è la "*collaboration*", cioè la cooperazione tra utenti. Ci si riferisce alla

² Il termine *prosumer* deriva dalla fusione delle parole inglesi *producer* e *consumer*, e a secondo dei contesti può assumere significati diversi, se pur simili. Gli economisti, ad esempio, con *prosumer* identificano un individuo fortemente indipendente dall'economia mainstream. In generale, si riferisce ad un utente che, svincolandosi dal classico ruolo passivo, assume un ruolo più attivo nel processo che coinvolge le fasi di creazione, produzione, distribuzione, consumo.

possibilità degli utenti di condividere le loro conoscenze, ad esempio tramite sistemi di tagging, o a diffondere le applicazioni che essi stessi progettano, diventando dei veri e propri prosumer.

L'orizzonte che si profila per l'IoS, quindi, è molto vasto ed eterogeneo. Come si può facilmente dedurre da quanto detto finora, le fasi di ricerca e di selezione dei servizi saranno tra gli aspetti più delicati, e non potranno quindi essere affidati all'operato dei singoli individui. Una valida soluzione è l'utilizzo dei cosiddetti Recommender System (RS), cioè sistemi che guidano gli utenti, durante la fase di scelta e acquisto, tramite suggerimenti, sulla base di informazioni fornite dai medesimi, filtrandole e elaborandole con algoritmi più o meno complessi, che vanno dal semplice confronto della similarità dei profili-utente, fino alle più complicate tecniche di elaborazione dei dati relativi agli utenti.

Finora i RS sono stati utilizzati principalmente per fornire suggerimenti riguardanti prodotti, come nel caso di Amazon con libri, cd e dvd (Amazon Recommender System) o la Apple con la musica di iTunes (Ping³). Di recente si sono moltiplicati i siti che mettono a disposizione suggerimenti per film (es. Flixster⁴), programmi tv (es. TV Genius⁵), pagine web (es. Delicious⁶), ma anche general purpose (es. Easyrec⁷). La nuova frontiera sono i suggerimenti riguardanti i servizi. Il dominio in questo caso, come abbiamo visto, è molto più vasto ed eterogeneo, alcuni servizi saranno altamente specializzati e saranno forniti da piccole e medie imprese, altri saranno indirizzati a servizi fondamentali e legati a opportunità di business significative, quindi sarà necessario un approccio ad hoc sufficientemente generale ma allo stesso tempo ben definito.

L'obiettivo specifico di questa tesi è la progettazione di un Recommender System per i servizi. L'approccio proposto prevede l'utilizzo della Collective Intelligence come cervello di tale sistema. Nel prossimo capitolo sarà analizzato il mondo dei servizi, fornendo la definizione generale di servizio, cercando di capire chi sono gli attori in scena e studiando la cosiddetta Internet of Services. Nel capitolo 3 verrà descritto in dettaglio come è strutturato un Recommender System, andando a vedere qual è lo stato dell'arte e eseguendo un'analisi comparativa tra alcuni di essi. Nel capitolo 4 sarà affrontato il tema della Collective Intelligence e saranno forniti alcuni esempi di applicazione, con particolare interesse all'ambito dei servizi. Gli ultimi due capitoli presentano la proposta concreta del RS basato su CI progettato, mostrando, infine, una sua applicazione ad un caso pratico d'uso.

³ <http://www.apple.com/itunes/ping/>

⁴ <http://www.flixster.com/>

⁵ <http://www.tvgenius.net/>

⁶ <http://www.delicious.com>

⁷ <http://www.easyrec.org/>

CAPITOLO 2

Internet dei servizi

In questo capitolo si esplora il mondo dei servizi, da quelli tradizionali ai web service, fino alla cosiddetta Internet of Service. Nel primo sottocapitolo si fornisce la definizione di servizio e se ne analizzano le caratteristiche; nel sottocapitolo 2 si parla dei web service e dell'architettura SOA; nel sottocapitolo 3, infine, si procede a una panoramica sull'Internet dei servizi e sui campi di ricerca correlati.

2.1 I Servizi

In questa sezione vedremo come il mondo dei servizi si è evoluto nel tempo, capiremo meglio cos'è un servizio e cosa lo differenzia da un prodotto.

2.1.1 Storia dei servizi

La rivoluzione industriale ha modificato radicalmente le modalità di produzione, i consumi e i mercati, ed ha avuto come effetto la trasformazione della società dei paesi ricchi e avanzati sia dal punto di vista culturale sia sotto l'aspetto strutturale. Risolto il problema della produzione del cibo e quello della produzione di beni strumentali, la domanda si è quindi rivolta verso la produzione di servizi.

Tra i primissimi servizi creati dall'uomo troviamo il commercio, a cui hanno fatto seguito l'amministrazione della società e del territorio, la difesa della comunità, la gestione delle pratiche religiose. Oggi parliamo di rete dei servizi, varia e complessa a seconda della loro tipologia, che copre le necessità legate alla vita quotidiana come la cultura, l'informazione, la sanità, il commercio, l'economia, la mobilità, il tempo libero.

Secondo la *Classificazione internazionale dei prodotti e dei servizi al fine della registrazione dei marchi* attualmente esistono 11 classi di servizi ¹ [4]:

Classe 35. Pubblicità; gestione di affari commerciali; amministrazione commerciale; lavori di ufficio.

Classe 36. Assicurazioni; affari finanziari; affari monetari; affari immobiliari.

Classe 37. Costruzione; riparazione; servizi d'installazione.

Classe 38. Telecomunicazioni.

Classe 39. Trasporto; imballaggio e deposito di merci; organizzazione di viaggi.

Classe 40. Trattamento di materiali.

¹ La *Classificazione internazionale dei prodotti e dei servizi* è stata stabilita nell'accordo di Nizza del 1957 e poi riveduta in quello di Stoccolma nel 1967 e a Ginevra nel 1977. Attualmente siamo arrivati alla nona edizione. Le classi dalla 1 alla 34 riguardano i prodotti.

Classe 41. Educazione; formazione; divertimento; attività sportive e culturali.

Classe 42. Servizi scientifici e tecnologici e relativi servizi di ricerca e di sviluppo; servizi di analisi e di ricerca industriali; servizi di progettazione e sviluppo di computer e programmi informatici; servizi giuridici.

Classe 43. Servizi di ristorazione (alimentazione); alloggi temporanei.

Classe 44. Servizi medici; servizi veterinari; servizi d'igiene e di bellezza per persone o per animali; servizi di agricoltura, di orticoltura e di selvicoltura.

Classe 45. Servizi personali e sociali resi da terzi per soddisfare esigenze individuali; servizi di sicurezza per la protezione di beni e di individui.

I cambiamenti economici dei paesi sviluppati sono sempre più influenzati dai servizi. Gli studi sul mercato del lavoro per l'anno 2009 pubblicati dall'Ufficio Internazionale del lavoro (ILO)² [5] mostrano chiaramente come a livello mondiale, negli anni che vanno dal 1998 al 2008, l'occupazione del settore dell'agricoltura è scesa da una percentuale del 40.8% ad una del 33.5%, l'industria è passata dal 21% a poco più del 23%, mentre i servizi sono cresciuti dal 38% al 43.3%. In particolare, è interessante evidenziare, nei dieci anni di riferimento, il trend evolutivo della macroarea relativa ai paesi a capitalismo maturo (Nord America, Europa, Giappone e Oceania), dove si è assistito ad una contrazione del peso occupazionale nell'agricoltura, che è passata dal 5.8% al 3.7%, ma anche dell'industria, da circa il 28% al 25%, a fronte di un aumento dell'importanza dei servizi, passati dal 66% al 71%.

Dall'inizio degli anni 2000, con il crescere della diffusione di Internet, molte tipologie di servizi sono state rese accessibili anche tramite la Rete. Servizi di prenotazione, di ricerca, di acquisti, di e-learning, di pubblicità, solo per citarne alcuni, hanno fatto la comparsa sul Web, semplificando la vita quotidiana della popolazione mondiale. Parallelamente a questa migrazione dei servizi esistenti, si è assistito alla nascita di nuovi servizi, come SaaS o PaaS³, che possono esistere solo nel contesto della Rete. L'insieme di tutti questi servizi disponibili in Internet prende il nome di eServices.

2.1.2 Cos'è un servizio

Prima di addentrarci nel nucleo dell'argomento, è di fondamentale importanza capire cos'è sostanzialmente un servizio. Nonostante la grande diffusione del termine, ci sono molte discordanze tra le definizioni esistenti. In economia, un servizio è considerato l'equivalente non materiale della merce. Potrebbe quindi essere definito come una prestazione lavorativa o professionale che compie un soggetto a favore di chi la richiede. Con una definizione più generale può essere espresso come il risultato di attività di natura più o meno intangibile, che ha luogo nell'interazione tra il cliente e il sistema di fornitura del servizio.

Una buona definizione di servizio è stata data da Justin O'Sullivan nella sua tesi di dottorato presso la facoltà di Information Technology dell'Università di Queensland (Australia) dal titolo *Towards a Precise Understanding of Service Properties* [6]: "un servizio è essenzialmente una promessa di una parte (il fornitore) ad eseguire una funzione (il servizio) a vantaggio di un'altra parte (l'utente) in un qualche momento e luogo, attraverso qualche tipo di canale".

² www.ilo.org

³ SaaS e PaaS stanno rispettivamente per Software as a Service e Platform as a Service. Il primo è un modello di distribuzione del software applicativo dove un produttore di software sviluppa, opera (direttamente o tramite terze parti) e gestisce un'applicazione web che mette a disposizione dei propri clienti via internet. Il secondo è un modello simile, con l'unica differenza che al centro di tutto, invece di un'applicazione, c'è una piattaforma, cioè una architettura hardware con un framework software.

Una definizione più formale è stata data da Roberta Ferrario, in collaborazione con il professor Nicola Guarino, nel loro articolo *Towards an Ontological Foundation for Services Science* [7]: “un servizio è presente all’istante t e nel luogo l se e solo se, all’istante t , un agente è esplicitamente impegnato a garantire l’esecuzione di un qualche tipo di azione nel luogo l , al verificarsi di un certo evento, nell’interesse di un altro agente e sotto un accordo a priori tra i due, secondo determinate regole”. Per capire meglio il concetto, si immagini un servizio concreto, ad esempio quello fornito dai pompieri, che vengono chiamati per spegnere un incendio. L’azione viene effettuata nell’istante in cui spengono le fiamme, nel luogo in cui era scoppiato l’incendio, ma il servizio era presente anche prima e lo è anche dopo, non solo nell’istante in cui viene eseguita l’azione. Il servizio fornito dai pompieri, dunque, consiste nell’impegno che viene da essi assunto ad intervenire, e in questo caso, essendo un servizio pubblico, l’accordo tra le parti è regolamentato dalle leggi dello Stato, che prevedono il pagamento delle tasse, da parte dei cittadini, e l’azione in caso di necessità, da parte dei pompieri.

Ricapitolando, al centro di ogni servizio c’è una situazione di impegno, con il quale l’amministratore del servizio garantisce l’esecuzione di una o più azioni, tramite mezzi di un *service producer*, che può coincidere con l’amministratore stesso o essere delegato da questo, a vantaggio dell’utenza, pattuendo un certo costo e fissando determinate regole.

2.1.3 Definizione di servizio NEXOF-RA

Nell’ambito del progetto Nessi Open Framework Reference Architecture (NEXOF-RA)⁴, Antonio De Nigro e Piero Corte della società Engineering⁵, hanno prodotto un report [8] nel quale espongono le proprietà che, secondo il loro studio, sono indispensabili per la definizione del concetto di servizio. Il documento presenta 4 concetti chiave, che identificano situazioni di natura via via più complessa. Ogni concetto è stato analizzato, scomposto nelle sue componenti principali e infine rappresentato attraverso un esempio concreto (vedi appendice 1).

Tra i 4 concetti proposti, quello scelto da NEXOF-RA per essere adottato come definizione di servizio è il seguente:

Un’azione eseguita da una entità (il fornitore) che corrisponde alla richiesta di un’altra entità (richiedente), secondo un accordo precedentemente stabilito tra le due parti.

Esempio

Una persona (un uomo affamato) chiede ad un’altra persona (un cuoco) di preparargli e servirgli un pasto; dopo di che quest’ultimo (il cuoco) fa ciò che gli è stato richiesto.

⁴ Nessi Open Framework Reference Architecture (NEXOF-RA) è un progetto che rappresenta il primo passo nel processo di costruzione del Nessi Open Framework, la piattaforma aperta per la creazione e la diffusione delle applicazioni, che consentirà la creazione di un ecosistema basato sui servizi, dove fornitori di servizi e terze parti possono collaborare facilmente.

Nessi è l’acronimo di Networked European Software and Services Initiative, cioè l’ETP (European Technology Platform) dedicata ai software e ai servizi. Una ETP è una partnership pubblica e privata che mira a definire e coordinare la ricerca nelle aree economiche strategiche per l’Europa.

<http://www.nexof-ra.eu/?q=node/2>

⁵ <http://www.eng.it/>

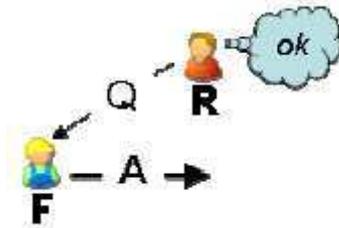


Figura 2.1: rappresentazione grafica della definizione di servizio NEXOF-RA [8]

F: il cuoco è l'entità fornitore;

R: l'uomo affamato è l'entità richiedente;

Q: chiedere di preparare e servire il pasto è la richiesta;

A(Q) preparare e servire il pasto, se è stata effettuata esattamente tale richiesta, è l'azione condizionata.

Ritenendola sufficientemente generale, la scelgo come definizione formale di servizio.

2.1.4 Caratteristiche peculiari dei servizi

Il processo produttivo di servizi presenta alcune caratteristiche specifiche che ne determinano l'assetto organizzativo e le metodologie di controllo che le aziende del settore devono adottare. È corretto precisare, tuttavia, che, con l'evolversi del panorama dei servizi, soprattutto in relazione ai servizi accessibili tramite Internet, alcune delle seguenti caratteristiche vanno riviste e rivalutate.

L'intangibilità

I servizi sono intangibili e incorporei: non possono essere toccati, visti, annusati, assaggiati o ascoltati, e non vi è nemmeno la necessità di trasporto, accumulo o stoccaggio. Inoltre un servizio non può essere posseduto o rivenduto, né tantomeno può essere restituito dal service consumer al service provider.

La deteriorabilità

I servizi sono deteriorabili in due sensi:

- a) I sistemi, i processi e le risorse rilevanti per i servizi vengono assegnati per la fornitura di un particolare servizio per un limitato intervallo di tempo. Se il consumatore di quel particolare servizio non richiede o non consuma il servizio in quell'arco temporale, il servizio non può essere fornito a lui in un altro momento. In alcuni casi questa è una opportunità di guadagno persa, in altri è possibile assegnare le risorse, i processi e i sistemi ad un altro consumatore che ne manifesti l'esigenza. Ad esempio una parrucchiera quando salta l'appuntamento con un cliente può servirne un altro. Viceversa un posto vuoto su un aereo non può essere venduto a qualcun altro una volta che l'aereo è decollato.
- b) Quando viene fornito un servizio, tale servizio scompare irreversibilmente, essendo stato "consumato" dal cliente. Ad esempio il passeggero di un aereo che è stato trasportato a destinazione non può riutilizzare il biglietto per tornare in quel luogo in un altro momento, la fornitura del servizio si è conclusa una volta ritirato l'eventuale bagaglio in stiva.

L'inseparabilità dai fornitori e dai consumatori

Il service provider è indispensabile per la consegna del servizio, in quanto lo deve generare e servire prontamente, quando il cliente lo richiede. In molti casi la consegna del servizio viene eseguita automaticamente ma il fornitore deve comunque assegnare le risorse e i sistemi preliminarmente e garantire che siano disponibili nel momento in cui vengono richiesti. Inoltre, il consumatore è inseparabile dalla consegna del servizio perché esso è coinvolto dalla fase di richiesta fino a quella di consumo. Ad esempio, il consumatore deve entrare nel negozio del parrucchiere e sedersi sulla sedia. Parallelamente, il parrucchiere deve essere in quel negozio per poter fornire il servizio.

La simultaneità tra fornitura e consumo

I servizi vengono forniti e consumati nel medesimo intervallo di tempo. Non appena il consumatore del servizio lo richiede, quel particolare servizio deve essere generato dal nulla, e istantaneamente il consumatore usufruisce dei benefici prodotti da tale azione.

La variabilità

Ogni servizio è unico. Viene generato, fornito e consumato una volta soltanto e non può essere mai ripetuto identico a se stesso nel tempo, nel luogo, nelle circostanze, nelle configurazioni, nelle risorse assegnate, anche qualora il medesimo cliente richieda il medesimo servizio. In alcuni casi, inoltre, i servizi sono considerati intrinsecamente eterogenei essendo modificati ogni volta in base alle esigenze del cliente. Si pensi, ad esempio, al servizio taxi. Ogni volta cambiano l'istante di tempo, il percorso, ma anche l'autista e l'auto.

La clientela

Particolare importanza riveste la gestione del rapporto con il cliente, soprattutto nei casi in cui vi è contatto diretto frequente tra il fornitore del servizio ed il cliente fruitore dello stesso. Infatti se il fornitore dimostra scarsa attenzione verso le esigenze, anche immateriali, che il cliente stesso si attende dal rapporto contrattuale posto in essere, si ha più facilmente una percezione di deterioramento della qualità del servizio, mettendo a rischio il rapporto di fiducia tra le due parti. Diventa quindi fondamentale coinvolgere il cliente nel processo produttivo, permettendogli di esprimere preferenze e necessità, al fine di potergli offrire un servizio che rispetti il più possibile le sue attese. Il cliente nei servizi, quindi, fa parte del processo produttivo.

La reputazione del fornitore

Sia per i beni materiali, sia per i servizi è importante che la società produttrice (o fornitrice) goda di una buona reputazione. Nel caso dei servizi, questo assume però un rilievo maggiore, infatti, basta pensare alla cena in un ristorante o ad un viaggio in aereo: se l'esperienza è negativa il primo pensiero è "in quel ristorante si mangia male" oppure "quella compagnia di voli è pessima", tendendo a generalizzare un'esperienza vissuta una volta sola. Viceversa, nel caso di beni materiali, se un paio di scarpe è difettoso oppure una penna scrive male, le nostre perplessità si rivolgono principalmente al prodotto specifico, non all'azienda che l'ha prodotto.

Ciascuna di queste caratteristiche rende la concezione di servizio molto complicata. Un marketing efficace in questo settore richiede una visualizzazione creativa del servizio che evochi un'immagine concreta nella mente del consumatore. Dal punto di vista del consumatore, queste caratteristiche rendono difficile e talvolta impossibile valutare o confrontare servizi prima di averli provati.

2.2 I Web Services

Nell'ampio contesto dei servizi, possiamo individuare un filone più specifico che sarà uno dei domini di studio di questa tesi: i web services. Secondo la definizione data dal World Wide Web Consortium (W3C), un Web Service è un'applicazione software progettata per supportare l'interoperabilità tra diversi elaboratori su di una medesima rete, identificata da un URI (Uniform Resource Identifier), le cui interfacce pubbliche e collegamenti sono definiti e descritti come documenti XML, in un formato comprensibile alla macchina (specificatamente WSDL, vedi appendice 2). La sua definizione può essere ricercata da altri agenti software situati su una rete, i quali possono interagire direttamente con il Web Service, con le modalità specificate nella sua definizione, utilizzando messaggi basati su XML (vedi appendice 2), scambiati attraverso protocolli Internet (tipicamente HTTP).

Le tecnologie su cui si basano i Web Services sono:

- XML, eXtensible Markup Language
- SOAP, Simple Object Access Protocol
- WSDL, Web Services Description Language
- UDDI, Universal Description, Discovery and Integration.

Come si può vedere in dettaglio nell'appendice 2, attraverso l'utilizzo di questi e di altri standard, i Web Services rendono possibile la comunicazione e la cooperazione, attraverso il Web, di più applicazioni (servizi) che mettono a disposizione alcune funzionalità e, allo stesso tempo, utilizzano quelle rese disponibili da altre applicazioni.

I vantaggi offerti dai Web Services sono:

- Indipendenza dalla piattaforma: i Web Services possono, infatti, comunicare fra loro anche se si trovano su piattaforme differenti.
- Indipendenza dall'implementazione del servizio: l'interfaccia che un Web Service presenta sulla rete è indipendente dal software che implementa tale servizio. In futuro tale implementazione potrà essere sostituita o migliorata senza che l'interfaccia subisca modifiche e quindi senza che dall'esterno (da parte di altri utenti o servizi sulla rete) si noti il cambiamento.
- Riutilizzo dell'infrastruttura: per lo scambio di messaggi si utilizza SOAP, che fa uso di HTTP, grazie al quale si ottiene anche il vantaggio di permettere ai messaggi SOAP di passare attraverso sistemi di filtraggio del traffico sulla rete, quali Firewall.
- Riutilizzo del software: è possibile riutilizzare software implementato precedentemente e renderlo disponibile attraverso la rete.

Il concetto di Web Service implica evidentemente un modello di architettura ad oggetti distribuiti (oggetti intesi come applicazioni), che si trovano localizzati in punti diversi della rete e su piattaforme di tipo differente. Come vedremo nella prossima sezione tutto ciò spesso si traduce in una Service Oriented Architecture (SOA).

Software Service

Prima di passare a parlare di SOA, è utile fare una precisazione su quanto detto nella sezione precedente, riguardo alla definizione di servizio. Finora avevamo parlato di servizi in un contesto generale. Ora, però, vogliamo riferirci ad un ambito particolare, cioè quello IT. In tal senso parleremo, quindi, di Software Service, e questo richiede alcuni vincoli che vanno aggiunti alla definizione di servizio data nella sezione 2.1.3 [8]: un software service è un

servizio in cui richiedente e fornitore sono agenti software, non necessariamente controllati da persone fisiche. Quindi diventa fondamentale specificare che:

1. l'interazione tra l'entità richiedente e l'entità fornitore è mediata da agenti software, detti agente richiedente e agente fornitore. Ciò implica che l'interazione diretta tra esseri umani non avviene mai.
2. può verificarsi che, per compiere la loro funzione, gli agenti software necessitino di interazioni tra umani, ma queste avvengono in modo indiretto.

D'ora in poi, in questo capitolo, utilizzeremo il termine servizio nell'accezione dell'ambito IT, e quindi nel significato di servizio accessibile tramite la Rete grazie ad applicazioni software.

2.2.1 SOA

Con il termine Service Oriented Architecture (SOA) si indica un tipo specifico di sistema distribuito in cui le entità che lo costituiscono sono proprio i servizi. Nella suddetta architettura, illustrata in Figura 2.2, si distinguono tre attori:

- *Service Provider*: chi realizza e mette a disposizione un servizio. Le caratteristiche del servizio realizzato vengono memorizzate all'interno di un registry accessibile pubblicamente (operazione di *publish*);
- *Service Directory* (o Service Broker): chi gestisce il registry e consente la ricerca di un servizio sulla base delle caratteristiche con cui questo è stato memorizzato;
- *Service Requestor*: potenziale utente del servizio. Dopo aver interagito con il Service Directory per trovare il servizio desiderato (operazione di *find*), si collega al Service Provider (operazione di *bind*) e fruisce del servizio in questione (operazione di *use*).

Affinché sia possibile realizzare un'architettura di questo tipo, è necessario specificare:

- Un linguaggio per la *descrizione* dei servizi, rivolto sia a chi lo realizza che a chi lo cerca;
- Un componente per la *memorizzazione* e il *reperimento* dei servizi;
- Un mezzo di *trasporto* delle informazioni.

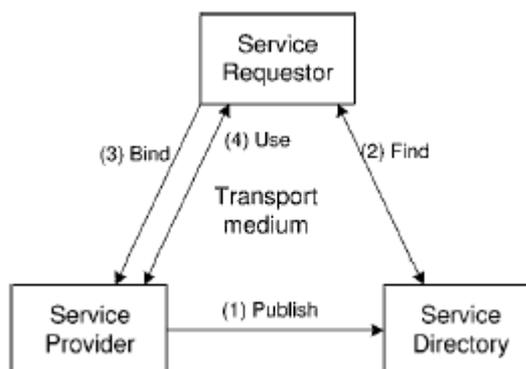


Figura 2.2: Service Oriented Architecture (SOA)

Il mezzo di comunicazione è un parametro dell'architettura, quindi questo approccio ha il vantaggio di potersi integrare con diversi ambienti, consentendo di realizzare applicazioni

multicanale, ossia fruibili attraverso dispositivi differenti.

Un sistema costruito seguendo la filosofia SOA è costituito da applicazioni ben definite ed indipendenti l'una dall'altra, che risiedono su più computer all'interno di una rete (che può essere la rete interna di una azienda, una rete di connessione fra più aziende che collaborano, o il Web). Ogni servizio mette a disposizione una certa funzionalità e può utilizzare quelle che gli altri servizi hanno reso disponibili, realizzando, in questo modo, applicazioni di maggiore complessità.

Caratteristiche

L'astrazione delle SOA non è legata ad alcuna specifica tecnologia, ma semplicemente definisce alcune proprietà, orientate al riutilizzo e all'integrazione in un ambiente eterogeneo, che devono essere rispettate dai servizi che compongono il sistema. In particolare un servizio dovrà essere:

- *ricercabile e recuperabile dinamicamente*. Un servizio deve poter essere ricercato in base alla sua interfaccia e richiamato durante l'esecuzione. La definizione del servizio in base alla sua interfaccia rende quest'ultima indipendente dal modo in cui è stato realizzato il componente che lo implementa.
- *autosufficiente e modulare*. Ogni servizio deve essere ben definito, completo ed indipendente dal contesto o dallo stato di altri servizi.
- *definito da un'interfaccia e indipendente dall'implementazione*. Deve essere definito in termini di ciò che fa, astraendosi dai metodi e dalle tecnologie utilizzate per implementarlo. Questo determina l'indipendenza del servizio non solo dal linguaggio di programmazione utilizzato per realizzare il componente che lo implementa, ma anche dalla piattaforma e dal sistema operativo su cui è in esecuzione.
- *debolmente accoppiato con altri servizi* (loosely coupled). Un'architettura è debolmente accoppiata se le dipendenze fra le sue componenti sono in numero limitato. Questo rende il sistema flessibile e facilmente modificabile.
- *disponibile sulla rete attraverso la pubblicazione della sua interfaccia* (in un Service Directory o Service Registry). Essere disponibile sulla rete lo rende accessibile da quei componenti che ne richiedono l'utilizzo. La pubblicazione dell'interfaccia deve rendere note anche le modalità di accesso al servizio.
- *dotato di un'interfaccia a "grana grossa"* (coarse-grained). Deve mettere a disposizione un limitato numero di funzionalità, in modo tale da non dover necessitare di un programma di controllo complesso. I dati che vengono trasmessi attraverso i messaggi possono essere costituiti sia dal risultato dell'elaborazione di un certo servizio sia da informazioni che più servizi si scambiano per coordinarsi fra loro. Per questo motivo e per il fatto che i servizi possono trovarsi su sistemi operativi e piattaforme diverse è necessario che i messaggi siano composti utilizzando un formato standard largamente riconosciuto (Platform Neutral).
- *realizzato in modo tale da permetterne la composizione con altri servizi*. Nell'architettura SOA le applicazioni sono il risultato della composizione di più servizi, quindi ogni servizio deve essere indipendente dagli altri, in modo tale da ottenere il massimo della riusabilità. La creazione di applicazioni o di servizi più complessi attraverso la composizione dei servizi di base viene definita Service Orchestration.

Di seguito uno schema riassuntivo delle caratteristiche proprie di un servizio all'interno di un'architettura SOA.



Figura 2.3: caratteristiche di un servizio in un'architettura SOA

2.2.2 Web services e SOA

Quando parliamo di Web Services, l'architettura in cui si inseriscono non è necessariamente di tipo SOA. Si possono, ad esempio, utilizzare i Web Services per aggiungere una funzionalità basata sui servizi ad un'architettura esistente, al fine di ottenere un certo requisito, in un sistema privato interno ad una azienda (vedi figura2.4).

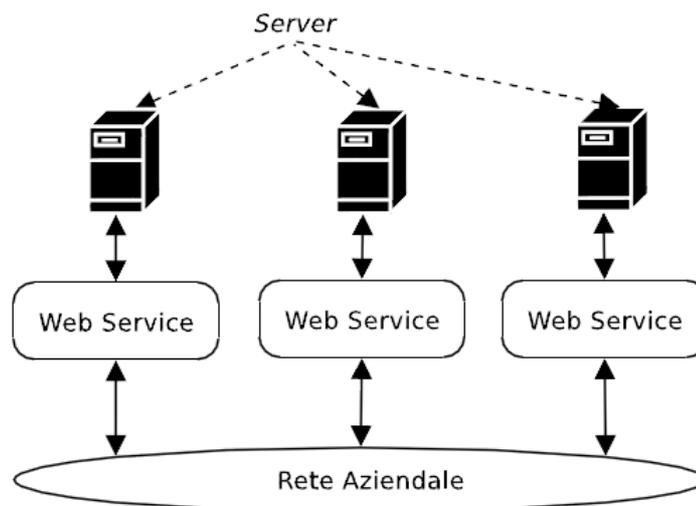


Figura 2.4: Utilizzo di Web Services in un'architettura di tipo non SOA.

In questo caso, come in altri, non è detto che si possa definire il sistema realizzato un'architettura orientata ai servizi. Ad esempio, in una rete interna ad una azienda che fa uso

della tecnologia dei Web Services, se si prevede che le modifiche che verranno apportate nel tempo al sistema non siano in misura elevata, si può pensare di memorizzare le informazioni riguardanti tutti i servizi nella configurazione di ognuno di essi, escludendo perciò dal progetto la presenza di un Service Registry, entità prevista invece dal modello di architettura SOA.

È però evidente che l'utilizzo di web services in un'architettura non SOA limita notevolmente le possibilità collaborative tra aziende, per varie ragioni:

- l'eterogeneità dei sistemi
- le differenze nelle modalità di interazione
- la mancanza di un registro dove cercare i servizi
- le difficoltà nel comporre servizi diversi

In figura2.5 è riportata la schematizzazione del funzionamento di un sistema con architettura SOA, realizzato attraverso l'uso dei Web Services.

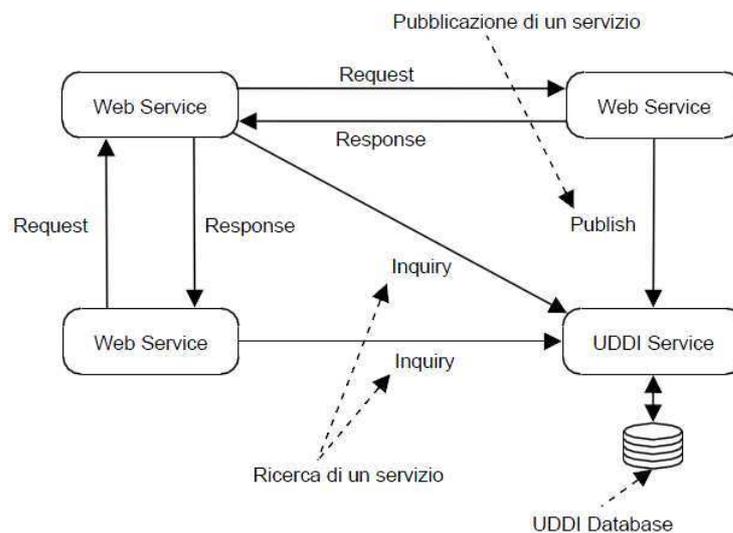


Figura2.5: Sistema con architettura Service-Oriented realizzato con la tecnologia dei Web Services.

Come si può osservare dalla figura in un'architettura SOA gioca un ruolo centrale il Service Registry. Per implementare tale componente i Web Services fanno uso di UDDI (Universal Description, Discovery and Integration). UDDI è un servizio di registro pubblico in cui le aziende possono pubblicare e ricercare Web Services. Esso mantiene informazioni relative ai servizi come l'URL e le modalità di accesso. Anche lo stesso UDDI è un Web Service, e nello specifico mette a disposizione due operazioni:

- Publish, per la registrazione del servizio
- Inquiry, per la ricerca del servizio.

Da quanto detto si deduce che una possibile architettura per Web-Services è un'istanza di una SOA dove il mezzo di comunicazione è il Web, il linguaggio per la descrizione dei servizi è WSDL e il registro dei servizi è implementato da UDDI.

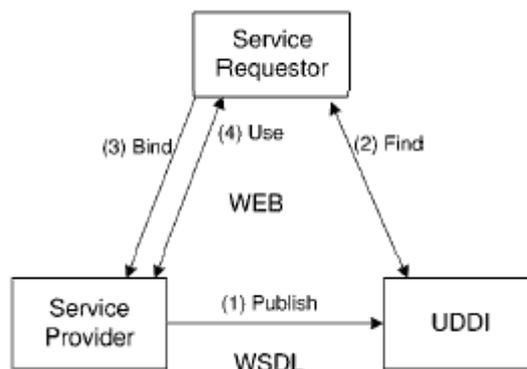


Figura 2.6: SOA per Web Service

2.3 Dalla SOA all'Internet dei Servizi

In questo sottocapitolo si studia l'avvento dell'Internet dei servizi, se ne individuano gli obiettivi, e si specificano quali saranno i principali campi di ricerca.

2.3.1 L'avvento dell'Internet dei Servizi

Molto spesso i termini Web e Internet vengono utilizzati, impropriamente, in modo interscambiabile. Tale ambiguità è dovuta alla predominanza che hanno avuto finora i web browser come principale mezzo di interazione con Internet.

Internet è una rete di computer mondiale ad accesso pubblico, e attualmente rappresenta il principale mezzo di comunicazione di massa. L'origine di Internet risale agli anni sessanta, ad opera degli USA, che misero a punto durante la Guerra Fredda un nuovo sistema di difesa e di controspionaggio, il progetto ARPANET, finanziato dalla Defence Advanced Research Projects Agency, una agenzia dipendente dal Ministero della Difesa statunitense. In una nota del 25 aprile 1963, Joseph C.R. Licklider aveva infatti espresso l'intenzione di collegare tutti i computer e i sistemi di time-sharing in una rete continentale.

Il World Wide Web, più spesso abbreviato in Web, invece, è un servizio di Internet consistente nella possibilità di navigare ed usufruire di un insieme vastissimo di contenuti multimediali e di ipertesti. Il Web è stato inizialmente implementato da Tim Berners-Lee⁶ mentre era ricercatore al CERN, sulla base di idee dello stesso Berners-Lee e di un suo collega, Robert Cailliau. La data di nascita del Web viene indicata nel 6 agosto 1991, giorno in cui Berners-Lee mise on-line su Internet il primo sito Web. Gli standard su cui è basato, in continua evoluzione, sono ora gestiti dal World Wide Web Consortium (W3C).

Internet, quindi, è l'entità, su scala mondiale, che rappresenta l'insieme di tutte le reti IP interconnesse, mentre il Web è un servizio di Internet che collega in modo ipertestuale i documenti multimediali contenuti nei server di tutto il mondo.

La rapidità dell'evoluzione tecnologica e la proliferazione di dispositivi come telefoni cellulari, netbook, tablet, lettori di eBook, ma anche televisioni, lettori mp3 e altri dispositivi multimediali, hanno moltiplicato le possibilità e le modalità di accesso ad Internet. Molte aziende si sono lanciate in nuovi avventurosi business per esplorare le opportunità offerte

⁶ Il 15 aprile 2004 Tim Berners-Lee ha ricevuto il premio Millennium Technology, (riconoscimento riservato a chi contribuisce con una innovazione tecnologica ad un miglioramento diretto della qualità della vita delle persone) per l'invenzione del World Wide Web.

dallo scenario che si è venuto a creare. Un caso esemplificativo è quello di Apple, che si è posta come leader del settore grazie alla produzione di dispositivi mobili di grande successo, quali iPod, iPhone e il recente iPad. Altre compagnie come Google, Sony, o Amazon stanno seguendo questa scia e hanno creato soluzioni analoghe che combinano dispositivi innovativi con piattaforme di distribuzione basate sul Web, fornendo quindi un valore aggiunto ai propri utenti.

Il superamento dei Web Service

Tradizionalmente l'idea di diffondere e fornire i servizi via Web è stata strettamente legata ai Web Service. Il vantaggio fondamentale di tali tecnologie risiede nel supporto che essi offrono nello sviluppo di sistemi distribuiti altamente complessi, che massimizzano il riutilizzo di componenti debolmente accoppiate. Nel tempo, quindi, sono stati creati molteplici linguaggi per la composizione di servizi in una modalità orientata ai processi. Inoltre sono stati aggiunti progressivamente anche linguaggi per la sicurezza, le transazioni, i messaggi e le notifiche. Questa serie di linguaggi, però, offrono una descrizione sintattica dei servizi, ma tralasciano quella semantica. Come conseguenza i processi di ricerca, selezione e composizione rimangono un'operazione manuale dell'utente.

Come abbiamo visto, un principio fondamentale delle SOA è la nozione di registro dei servizi, grazie al quale è possibile pubblicare i servizi prodotti, mettendoli a disposizione di tutti, e rendendoli quindi accessibili grazie ad una semplice ricerca. La pubblicazione di Web Service è stata quindi al centro della ricerca e dello sviluppo di questo settore fin dagli inizi. L'Universal Business Registry, creato all'interno di UDDI, è stato probabilmente il più grande sforzo profuso in tal senso. Sulla base di UDDI, compagnie come SAP, IBM e Microsoft, hanno creato nel 2001 questo registro universale per Web Service che poteva essere consultato pubblicamente. Non raggiungendo, il successo sperato, dopo 5 anni di attività, nel 2006, fu chiuso e il progetto accantonato.

Una delle principali ragioni per la mancanza di successo di UDDI è stata che, sebbene tali registri fossero relativamente complessi, essi non supportavano query particolarmente espressive. Inoltre la maggiore diffusione dei Web Service è nell'ambito delle aziende, le quali però si dimostrano spesso riluttanti alla divulgazione dei loro servizi, per motivi principalmente strategici, limitando di fatto la quantità di WS pubblicati nel registro.

Le Web API

Di pari passo con l'evoluzione del Web 2.0, i servizi disponibili a livello mondiale sul web, fino a quel momento limitati ai classici WS, si sono evoluti e hanno iniziato a proliferare le cosiddette Web API.

Le API, Application Programming Interface (Interfaccia di Programmazione di un'Applicazione), sono un insieme di procedure, di solito raggruppate a formare un set di strumenti specifici per un determinato compito, che consentono un'astrazione tra l'hardware e il programmatore, o tra software a basso ed alto livello. Essenzialmente le API permettono di espandere le funzionalità di un programma, fornendo gli strumenti necessari per farlo. Per uno sviluppatore mettere a disposizione un set di API di un suo software significa, quindi, dare la possibilità ad altri di interagire con la sua piattaforma e, soprattutto, estendere le funzioni e le caratteristiche della struttura base della piattaforma. In altri termini, le API sono un ottimo strumento per promuovere un'applicazione offrendo ad altri un modo per interagirci.

Nonostante la popolarità attualmente raggiunta, lo sviluppo delle Web API non è guidato da standard, modelli o linee guida. Si potrebbe dire che attualmente il loro utilizzo sia più una forma d'arte piuttosto che una scienza esatta.

Ultimamente si è assistito ad uno sforzo significativo verso questo nuovo approccio ai servizi, ma permangono alcuni limiti, come è stato evidenziato da una ricerca condotta da nel febbraio del 2010 e presentata nell' European Conference on Web Services di quell'anno. Oltre 200

Web API, scelte in modo random tra quelle offerte dal sito Programmable Web, sono state analizzate, e tra le statistiche più rilevanti si osservano i seguenti risultati:

- La qualità delle descrizioni lasciano molto a desiderare. Ad esempio solo il 28% delle API stabilisce chiaramente il tipo di dati dei parametri utilizzati, anche se c'è da dire che l'83% delle API sono comunque corredate di esempi pratici di interrogazioni.
- La descrizione degli input non è omogenea, includendo valori di default, valori alternativi, e parametri opzionali.
- Il formato di output è ancora a favore di XML, infatti ben l'85% delle API può fornire i risultati in XML, sebbene si stia piano piano affermando anche JSON (supportato dal 42% delle API).
- Più dell'80% delle API richiede alcune forme di autenticazione.

Nonostante alcune limitazioni intrinseche, le Web API stanno catturando l'attenzione di grandi compagnie come Facebook, Yahoo, o Amazon, e sono sempre più utilizzate nelle applicazioni del mercato mobile. Ad esempio, Apple ha comprato Siri, una compagnia che fornisce applicazioni per dispositivi mobili in grado di gestire prenotazioni a ristoranti o ad eventi, e ottenere informazioni utili in base alla posizione geografica dell'utente, come il meteo o le indicazioni stradali. Altre iniziative simili sono l'App Store di Apple o il Marketplace di Android, che stanno creando un vero e proprio nuovo tipo di Business Model, che fonde la distribuzione tradizionale orientata al software, con la distribuzione di servizi, inclusi aggiornamenti gratuiti, o l'accesso a servizi remoti su Internet.

Il cloud computing

Coerente ai trend esposti finora è l'esplosione del mercato del cloud computing. Essenzialmente il cloud computing fornisce agli utenti infrastrutture altamente avanzate, personalizzate e scalabili in modo trasparente attraverso Internet, permettendo a chiunque ne abbia necessità l'utilizzo di un'infrastruttura IT senza doversi sobbarcare l'investimento iniziale e le manutenzioni periodiche, lasciando tali incombenze ai fornitori delle "nuvole". I pagamenti seguono la logica del pay-per-use o di una sottoscrizione di un contratto flat.

I servizi offerti includono archiviazione, potenza computazionale, e applicazioni, tutte accessibili da remoto e in modo scalabile. Alcuni esempi di soluzioni cloud commerciali sono le App Engine, il servizio Gmail e i fogli elettronici di Google, o l'Elastic Cloud Computing di Amazon.

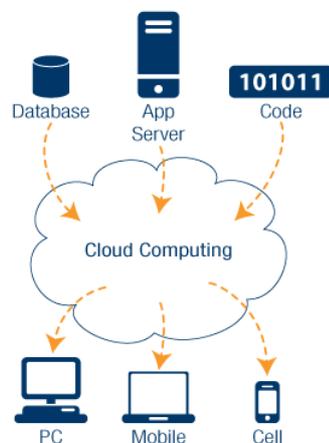


Figura 2.7: Cloud Computing

La popolarità del cloud computing è stata supportata da due trend principali: da una parte, la crescente importanza di avere elaborazioni rapide in quantità sempre maggiori, che ha creato

la necessità di strumenti di alto livello all'interno di compagnie che, però, non hanno competenze per costruire e mantenere grandi centri di elaborazione; dall'altra, la diffusione di smartphone, netbook e tablet, che hanno generato una richiesta supplementare per soluzioni basate sul cloud computing, grazie alle quali è possibile compensare le limitate risorse di cui sono dotati tali dispositivi.

La nascita dell'Internet dei Servizi

Sulla base di quanto detto, è evidente che l'interesse nel fornire eService su Internet è più accentuato che mai. Stiamo assistendo ad una evoluzione che suggerisce che l'Internet conosciuta finora sta gradualmente spostandosi da un'era industriale, predominata da una economia basata sulla produzione di software, ad un'era post-industriale basata sulla fornitura di informazione, soluzioni innovative e servizi. In aggiunta ai tradizionali modelli di business, ora troviamo un crescente numero di business model basati sui servizi come il cloud computing, o su applicazioni disponibili online.

Con Internet dei Servizi (*Internet of Services*) si intende proprio questa evoluzione di Internet, grazie alla quale saranno disponibili in rete non solo dati, informazioni e contenuti, ma anche applicazioni di vario genere, sotto forma di servizi facili da usare e da combinare, accessibili attraverso molteplici canali, che prevedono oltre al tradizionale PC, anche i dispositivi mobili, quali cellulari, e PDA. La tendenza è quella di una rapida crescita sia nel numero che nella varietà di queste applicazioni disponibili su Internet: dai servizi per la pubblica utilità (eGovernment, eHealth, eInclusion, eEnergy, eMobility) e per il business (eCommerce, eBanking), alle applicazioni rivolte all'uso personale dell'utente, i cosiddetti "personal management services" (mappe, agende, calendari, tutti on-line), dai servizi per socializzare e condividere contenuti tipici del Web 2.0, fino alle applicazioni pervasive e di prossimità, ovvero applicazioni fornite all'utente direttamente dall'ambiente in cui si trova (ad esempio tramite dispositivi mobili), in grado di fornire servizi fortemente dipendenti dalla locazione, dal contesto, dall'attività svolta.

L'Internet dei Servizi, quindi, è una visione dell'Internet del futuro dove qualsiasi applicazione software è disponibile sotto forma di servizio in Internet, dagli stessi software, agli strumenti per svilupparne di nuovi, fino alle piattaforme. Chiunque voglia sviluppare applicazioni può usare le risorse presenti nell'Internet dei Servizi, senza dover fare grossi investimenti, e quindi limitando i rischi economici. Questo nuovo scenario consentirà la proliferazione di idee innovative, che possono essere provate concretamente senza che questo significhi dover necessariamente mettere in gioco i propri capitali.

2.3.2 Gli obiettivi dell'IoS

All'attuale stato di sviluppo, è impossibile delineare requisiti specifici per l'Internet dei Servizi. Ciò che sembra chiaro è che per il momento è soltanto una combinazione di idee, tecnologie e comunità. Il ruolo principale all'interno dell'IoS appartiene certamente alle industrie del software e delle telecomunicazioni, ma ci sono anche altre entità in gioco di cui tenere conto, come i clienti, le web company e le comunità.

Attualmente non esiste alcuna visione comune sulla definizione delle infrastrutture della Future Internet nel contesto dei servizi, né tantomeno di ciò che questa infrastruttura potrà comprendere. L'investigazione di questi temi è affidata a diversi gruppi di ricercatori e certamente saranno sviluppati nuove tecniche ad hoc, ma ci sono delle domande che ci si deve porre. Quali sono gli obiettivi che ci si deve prefiggere? Quali saranno i benefici per le parti coinvolte? Quale saranno i modelli di business? Quale struttura assumerà l'IoS? Delle possibili risposte a queste domande sono state trovate da Man-Sze Li, Servane Crave, Jorge P. Muller e Steven Willmott nella loro ricerca "The Internet of Services: Vision, Scope and Issues" presentata alla conferenza eChallenge 2008. Di seguito vengono riassunte in modo schematico le principali idee emerse da tale lavoro.

La missione dell'IoS

Senza dubbio l'Internet dei Servizi deve avere un impatto positivo sulle capacità degli utenti, siano essi individui, organizzazioni, o altri servizi basati su software. Deve, cioè, produrre benefici tangibili per le persone, e non solo, nei loro differenti ruoli all'interno della società. Questa missione ha diverse e importanti implicazioni per quanto riguarda il ruolo della IoS nell'Internet del futuro.

L'IoS è un mezzo sia per la fornitura di servizi che per il consumo di servizi, ma anche per l'interazione tra fornitura e consumo, la cosiddetta "prosumption". Le tecnologie e i servizi che compongono l'IoS sono quindi mezzi per soddisfare le esigenze degli utenti, e non fini.

A livello di infrastrutture, il paradigma dell'Internet of Services deve essere "any-to-any". I servizi a questo livello devono essere condivisi, trasparenti e capaci di stare in un ambiente aperto e dinamico. Il termine aperto, in questo contesto, intercetta quattro aspetti fondamentali dell'IoS:

1. non deve essere vincolata a nessuna tecnologia o paradigma;
2. non deve essere posseduta o controllata da una singola entità;
3. il suo sviluppo e la sua crescita è fondato su una partecipazione spontanea degli utenti;
4. non deve avere pregiudizi verso modelli di business o ecosistemi, siano essi esistenti o emergenti.

L'infrastruttura dell'IoS è parte integrante della Future Internet, quindi l'innovazione di rottura con il passato non sarà totalmente senza regole, bensì si baserà su una combinazione di infrastrutture standard e infrastrutture emergenti e sulle opportunità creative che queste consentono.

Sebbene tali considerazioni generali segnalino la necessità di un approccio multidisciplinare all'IoS, esistono comunque dei temi comuni:

- il passaggio da sistemi interpretabili solo dagli umani a sistemi che facilitino l'interazione tra le macchine, sia in termini di fornitura di nuove interfacce di programmazione, sia in termini di adozione di un approccio più strutturato per definire la semantica di interazioni "macchina-macchina";
- il passaggio da una composizione statica dei servizi ad una dinamica, e il passaggio da una interoperabilità disegnata su elementi noti a priori ad una basata su elementi di volta in volta scoperti favorendo il riuso e l'assemblaggio "on-the-fly".
- l'aumento della distribuzione di applicazioni e di incapsulamento delle informazioni, nonché delle funzionalità di rete, passando dal "download and install" delle applicazioni software a modelli di accesso remoto "pay-per-use", e dal trasferimento di grandi moli di dati a query di informazioni remote "always-on". Si noti che questo cambiamento produce tre rilevanti conseguenze: la creazione di nuove classi di applicazioni distribuite; la creazione di nuovi modelli di distribuzione di funzionalità e di dati; la nascita di nuove attività e di business model.
- Il passaggio da applicazioni intra-organizzative chiuse e strettamente controllate, ad applicazioni totalmente distribuite, funzionanti in ambienti aperti e pubblici.
- il passaggio da contenuti, applicazioni, sistemi e scenari d'uso statici e scollegati fra loro, a contenuti dinamici e multimediali, servizi collaborativi e dipendenti dal contesto, reti ad hoc e ecosistemi innovativi, sistemi di sistemi globalizzati e sempre attivi.

I benefici che dovrebbe produrre l'IoS

L'intero panorama dell'Internet dei Servizi è vasto e ancora non ben distinguibile nelle sue parti, e risulta quindi impossibile determinarne un quadro definitivo. Basandoci sui mercati

esistenti, possiamo però fare delle ipotesi sui possibili utenti e fornitori che faranno parte dell'IoS, consapevoli che ciò non debba necessariamente corrispondere a quanto avverrà in futuro. Nella seguente tabella vengono elencati i probabili attori in gioco dell'Internet of Services.

Utenti		
<ul style="list-style-type: none"> • Singoli individui e comunità di individui • Organizzazioni e comunità di organizzazioni • “Things” (nel senso di Internet of Things) • Servizi basati su software 		
Fornitori		
Utility Provider	Aggregatori	Integratori
<ul style="list-style-type: none"> • Compagnie specializzate in software • Grandi compagnie specializzate nei servizi • Compagnie Web 2.0 • Compagnie telefoniche, ISP e fornitori di infrastrutture Web • Compagnie specializzate in hardware • Startup specializzate 	<ul style="list-style-type: none"> • B2B company di prima generazione • Centri industriali • Grandi compagnie specializzate nei servizi • Compagnie specializzate nel fornire e gestire web service • Fornitori EAI • Integratori di sistemi • Startup specializzate 	<ul style="list-style-type: none"> • Fornitori di software • Integratori di sistemi • Compagnie specializzate in hardware • Comunità Web 2.0

Figura 2.8: Utenti e fornitori dell'IoS

Proprio come l'Internet odierna, l'IoS dovrebbe servire i singoli individui, le business company, le organizzazioni governative e gli enti pubblici. Gli effetti che può avere sulle capacità degli utenti sono di tre tipi: tecniche, economiche e sociali.

Il miglioramento delle capacità tecniche tradizionalmente è l'obiettivo principale dell'ICT, dove grandi passi avanti sono stati fatti e sono tuttora in corso sia nell'industria che nella ricerca. Questo aspetto riguarda tutte e quattro le categorie di utenti.

Il miglioramento delle capacità economiche è legato ai costi e ai benefici, sebbene questi non siano esprimibili esclusivamente in termini monetari. Esempio tipico è la produttività. Tali vantaggi riguardano, tra gli utenti, solo i singoli individui e le organizzazioni.

I miglioramenti delle capacità sociali sono le più difficili da caratterizzare, in quanto si rivolgono a un dominio molto vasto, al quale appartengono i cosiddetti eHealth, eParticipation e eDemocracy, fino alle più recenti iniziative come la Fifth Freedom dell'Unione Europea, movimento che si occupa delle persone diversamente abili. Questi cambiamenti riguardano sia i singoli utenti, sia le comunità e le organizzazioni.

I modelli di business dell'IoS

Come descritto nella sezione 2.3.1, il modello più semplice di servizio è il servizio web che offre una soluzione attraverso l'interfaccia classica del browser. Nel tempo, questo modello è stato esteso, consentendone l'accesso anche tramite altri canali, come i cellulari e gli smartphone. Il canale alternativo, però, viene offerto come un puro gadget, senza alcun valore

aggiunto. Il passo successivo, come detto, è stata l'introduzione delle API, che hanno permesso di sfruttare il patrimonio informativo e funzionale del servizio originario per costruire dei servizi derivati. Questo modello consente a terze parti di costruire interfacce sui diversi canali o integrarli nei loro servizi, arricchendoli e distribuendo in maniera scalare i costi di produzione.

Un altro modello di business per i servizi è quello della piattaforma, che permette sempre attraverso delle API di sviluppare applicazioni terze che usino il suo patrimonio informativo e funzionale ma, in questo caso, all'interno del servizio originario. Questo vuol dire che la piattaforma gestisce tutte le regole di integrazione, distribuzione ed interfaccia. Così facendo la piattaforma cresce internamente di valore grazie alla miriade di applicazioni integrate in essa che allo stesso tempo ne beneficiano in termini di visibilità e distribuzione.

L'ultima generazione di servizi è rappresentata dal concetto di ecosistema. Per ecosistemi si intendono servizi multicanale che condividono un cuore funzionale, ma le cui interfacce sono distribuite su tutti i canali possibili. In altre parole c'è un'idea centrale di come funziona il sistema che viene però poi derivata su pc, cellulari, smartphone, tablet e lettori mp3, sfruttando al massimo le caratteristiche contestuali dell'ambiente d'uso. Il vantaggio consiste nello sfruttare il servizio in modo completo, sempre e ovunque, ottenendo un'esperienza d'uso integrata e continua. Sono formati da un insieme di oggetti, applicazioni e servizi complementari che, oltre a massimizzare l'uso del servizio, consentono alcune attività supplementari trasversali. Un esempio di ecosistema è costituito da ipod (dispositivo fisico), itunes (applicazione software), e apple store (negozi online).



Figura 2.9: esempio di ecosistema (i-pod+ itunes + apple store)

L'IoS, però, non dovrebbe essere circoscritto a modelli definiti, bensì dovrebbe lasciar aperte le porte alle nuove opportunità di business, favorendo quelli emergenti e sostenendo quelli in via di affermazione. Le principali linee guida da seguire dovranno essere: l'universalità dei servizi, l'accessibilità dei servizi e la neutralità dei servizi. La sinergia di questi tre elementi garantirà che

- 1) l'utilizzo dell'IoS sia libero e illimitato;
- 2) il modello di utilizzo sia orientato all'utente;
- 3) lo sviluppo e la crescita siano guidati direttamente dagli utenti, e non da fornitori o produttori.

Questi sono gli ingredienti basilari per la proposizione di valore dell'IoS. La creazione di valore, quindi, dovrebbe essere imperniata sul bordo della rete, dove si trovano gli utenti.

Gli ecosistemi e la struttura dell'IoS

Molti sono gli sforzi profusi nel tentativo di caratterizzare i servizi dell'Internet del futuro. Gli ecosistemi di servizi, commerciali o di altri tipi, come le AppExchange di Salesforce o gli AWS di Amazon, ma anche Facebook, Dopplr o la costellazione di lab e comunità di Google, stanno già producendo importanti miglioramenti riguardanti le capacità tecniche, come anche

per il benessere economico. Alcuni di questi fornitori pubblicano le API, i widget e altre funzionalità tecniche, e le rendono disponibili a bassi costi o gratuitamente, portando ad un'esplosione di nuovi servizi sviluppati da professionisti, ma anche da programmatori guidati solo dalla passione.

Gli attori principali dell'industria ICT e del Web 2.0 sono anche impegnati in iniziative che hanno come scopo l'interoperabilità tra le piattaforme di servizi, lo sviluppo e la continuità dei servizi tra le reti, la fusione tra il mondo fisico e il mondo virtuale. L'importanza della convergenza tecnologica ormai è stata ampiamente riconosciuta da più parti e investimenti significativi sono stati fatti per consentire l'erogazione dei servizi su reti fisse, mobili e ibride, ma anche per le reti di nuova generazione (Next Generation Network, o NGN⁷). Tutti questi importanti sviluppi dovrebbero essere i blocchi con cui costruire l'IoS, ma sebbene essi possano contribuire all'affermazione di valore dell'Internet of Services, essi non assicurano da soli un surplus positivo nella creazione del valore. Non è, infatti, chiaro se e come essi miglioreranno le capacità degli utenti. Certamente possono garantire un surplus di valore per gli utenti all'interno dei loro stessi ecosistemi, ma questi benefici sarebbero dipendenti dall'ecosistema stesso, venendo meno non appena un utente non appartiene più a quel particolare dominio.

Tenuto conto degli ecosistemi esistenti, sui quali stanno pesantemente investendo i principali fornitori di servizi, una ragionevolmente struttura dell'IoS potrebbe essere proprio una confederazione globale di tali ecosistemi.

2.3.3 I campi di ricerca

Nel dicembre del 2008 è stata indetta dalla Commissione Europea una gara di appalto per uno studio sugli effetti economici e sociali dei servizi software e dei servizi basati su software nell'ambito della definizione di una strategia europea univoca (European Software Strategy). Il contratto è stato assegnato a Pierre Audoin Consultants (PAC)⁸ e sono stati nominati come subappaltatori le società di ricerca IDATE⁹, Fraunhofer ISI¹⁰ e London Economics¹¹. Lo studio [9] è stato condotto tra l'aprile del 2009 e il giugno del 2010.

Ciò che è emerso dai risultati ottenuti segnala due trend tecnologici principali, che potrebbero accelerare o meno il processo di crescita dell'intero mercato dei Software and Software Based Service (SSBS) e in particolar modo la parte relativa all'IoS:

- da una parte, nuove tecnologie software che potrebbero rappresentare nuovi segmenti di mercato e nuove opportunità di business nell'industria SSBS. Esse potrebbero essere combinate o integrate con altre tecnologie piuttosto che essere utilizzate in modo indipendente. Il loro sviluppo è quindi subordinato agli SSBS stessi, per i quali sono

⁷ Una Next Generation Network (NGN o "rete di prossima generazione") è una rete basata su commutazione a pacchetto in grado di fornire servizi - inclusi servizi di telecomunicazioni - ed in grado di far uso di molteplici tecnologie a banda larga con QoS, nella quale le funzionalità correlate alla fornitura dei servizi siano indipendenti dalle tecnologie di trasporto utilizzate. Offre un accesso non limitato agli utenti a diversi service provider. Supporta una mobilità generalizzata consentendo la fornitura consistente ed ubiqua di servizi agli utenti.

⁸ <https://www.pac-online.com/>

⁹ <http://www.idate.org>

¹⁰ www.isi.fraunhofer.de/

¹¹ www.londecon.co.uk/

fattori di crescita interna e molto probabilmente saranno utilizzati come leve per favorirne uno sviluppo più rapido.

- dall'altra, tecnologie indipendenti dall'industria dei SSBS, utili allo sviluppo dei software stessi piuttosto che ai progressi delle tecnologie software. Mentre le esigenze dell'industria dei SSBS potrebbero accelerare lo sviluppo di queste tecnologie (aiutando a trovare nuovi business model sostenibili), quest'ultime devono essere considerate esterne a tale realtà, dipendendo in modo significativo da altre industrie (ad esempio dall'industria ICT). Il loro sviluppo, inoltre, dipenderà anche da innovazioni tecnologiche legate alla fisica, alla meccanica o alla gestione dell'energia.

Le tecnologie e i concetti critici per il futuro dell'industria SSBS appartenenti a tali trend sono:

- L'intelligenza artificiale
- Il Semantic Web
- L'Information Management
- I certificati digitali
- L'interoperabilità
- Le reti onnipresenti
- La Radio Frequency IDentification (RFID)
- La sicurezza
- I centri dati
- L'ultra-broadband

I primi tre punti dell'elenco sono di particolare interesse per l'argomento trattato nella tesi, quindi vengono approfondite di seguito.

Intelligenza Artificiale

Con il termine Intelligenza Artificiale ci si riferisce alla scienza che crea macchine in grado di svolgere funzioni e compiere collegamenti logici considerati intelligenti. Tali macchine possono raggiungere comportamenti simili a quelli umani, basandosi su programmi in grado di risolvere problemi e raggiungere obiettivi tramite procedure computazionali. Esistono due principali modi per ottenere un'intelligenza artificiale: combinare tecnologie esistenti, oppure imitare il modo di operare del cervello umano. Nel primo caso vengono utilizzati strumenti di Complex Event Processing (CEP), di Business Intelligence (BI), e piattaforme di tipo SOA. La tecnologia che invece emula le funzioni del cervello umano è data dai Multiple Agent System. Tali sistemi sono in grado di gestire livelli di complessità molto alti, come la correlazione di eventi dinamici e l'automazione di task, assicurando alle imprese, ad esempio, risposte in real time alle mutevoli condizioni di mercato.

Da un punto di vista commerciale, un agente è un clone virtuale di un elemento fisico (carrelli, stock, attrezzature, prodotti, etc.) oppure immateriale (ordine di un cliente, prenotazione, transazione commerciale, etc.). Tali agenti insieme costituiscono i multiple agent system.

Da un punto di vista tecnologico, gli agenti sono componenti software distribuite che si trovano in più server e che comunicano l'un l'altro in modo asincrono. Sono completamente integrate nel sistema informativo come componente della SOA.

Con il loro approccio innovativo, i sistemi multi agente superano le limitazioni del classico approccio SOA, che è ancora troppo rigido per i processi che variano nel tempo, come l'allocazione dinamica delle risorse, le regole di business non pianificate, o ancora eventi casuali e imprevedibili. Tale tecnologia è la piattaforma ideale per le compagnie proattive, per quelle società volte all'innovazione e per le imprese con una struttura reticolare, che fanno

dell'adattabilità e del dinamismo la chiave del loro successo.

Alcuni casi in cui i sistemi multi agente possono essere utilizzati sono:

- Customer Relationship Management: possono aiutare le compagnie ad ottimizzare la gestione dei loro clienti.
- Green IT: la disponibilità di energia sta iniziando a scarseggiare e i costi dell'inquinamento stanno crescendo, quindi le compagnie hanno bisogno di strumenti per minimizzare il consumo di energia e la produzione di rifiuti.
- Supply Chain Management: nel caso di eventi non programmati, i sistemi multi agente possono ridurre in modo significativo il tempo necessario a ridurre problemi di supply chain management allocando dinamicamente le risorse in modo efficiente.
- Manutenzione: gli agenti possono aiutare le compagnie a migliorare la fase post-vendita ottimizzandola con analisi real time in grado di rilevare i possibili problemi.

Altri ambiti d'utilizzo sono: la speech recognition, l'interpretazione del linguaggio naturale, la computer vision, i sistemi esperti, i problemi di classificazione tramite euristiche.

Semantic Web

Il web semantico è considerato il nuovo stadio evolutivo del Web e ad esso ci si riferisce con il nome di Web 3.0. Questa visione è stata teorizzata da Tim Berners-Lee, inventore del World Wide Web, che menzionò per la prima volta il Semantic Web nel 1999. Al giorno d'oggi la visione del Web Semantico sta alimentando l'innovazione e gli investimenti in nuovi prodotti e servizi. Più in generale, l'emergere di tecnologie semantiche per applicazioni consumer ed enterprise, che vanno dai social network fino alla gestione delle conoscenze personali, stanno portando molte nuove opportunità per l'industria del software. Le tecnologie semantiche emergenti hanno diverse aree di applicazioni, come ad esempio l'assistenza sanitaria, le scienze della vita, la finanza, la pubblicità, il marketing, le ricerche.

In accordo con l'idea di Tim Berners-Lee, il web semantico è un'evoluzione del World Wide Web nel quale il significato (semantica) delle informazioni e dei servizi sul web è definito, rendendo possibile per il web capire e soddisfare le richieste delle persone e delle macchine per l'accesso ai contenuti. Il Web che conosciamo oggi, fatto di reti di computer, opera su documenti e determina come questi documenti sono immagazzinati, scambiati e visualizzati. Grazie al protocollo di comunicazione http e al linguaggio HTML, possono essere capite dai computer le sintassi dei processi, consentendo la comunicazione tra computer. In realtà i computer gestiscono tali informazioni in modo cieco, non capendone il significato che vi sta dietro. Il Semantic Web va oltre. Invece di contenere soltanto documenti, esso racchiude l'essenza di persone, luoghi, eventi, musica, film, organizzazioni e di molto altro ancora. Il web semantico consente ai computer di processare il significato delle cose in modo da comprendere i concetti che stanno dietro alle pagine web e permettere collegamenti logici sulle relazioni che legano i vari oggetti. Il Semantic Web, quindi, connette i concetti per creare conoscenza.

Oltre alle classiche tecnologie di ipertesto come URI, HTML o XML, il Semantic Web è basato su quattro open standard del W3C:

- RDF (Resource Description Framework). Viene usato per descrivere concettualmente e modellare le informazioni.
- RDF Schema. Fornisce il vocabolario di base di RDF.
- SPARQL. È il linguaggio di interrogazione di RDF che permette ad agenti software e applicazioni di accedere ed estrarre informazioni dai grafi RDF.
- OWL (Web Ontology Language). è un linguaggio di markup per rappresentare esplicitamente significato e semantica di termini con vocabolari e relazioni tra gli stessi.

Lo scopo di OWL è descrivere delle basi di conoscenze, effettuare delle deduzioni su di esse e integrarle con i contenuti delle pagine web. Altro scopo di OWL è permettere alle applicazioni di effettuare delle deduzioni sui dati.

Il Semantic Web comprende una grande varietà di applicazioni e servizi. Nella tabella seguente vengono elencate le principali applicazioni attualmente disponibili.

Type of application	Application name	Company	Country	Description
Search engines	Hakia	Hakia	USA	General purpose, natural language search engines
	True Knowledge	True Knowledge	UK	
	Wolfran Alpha	Wolfran Alpha	USA	Computational knowledge engine
	Spock	Spock	USA	People search engine
	UpTake	UpTake	USA	Travel search engine
	Newssift	Financial Times	UK	Business news search engine
Search agents	Headup	SemantiNet	Israel	Presents content related to objects and terms identified on web pages
	BlueOrganizer	Adaptive Blue	USA	Displays smart link widgets on websites like book recommendations
Smart applications	Zemanta	Zemanta	Slovenia	Plug-in tool that helps users add relevant content to posts in blogs, e-mail, etc.
	SEAmail	Stanford University	USA	Semantic e-mail system that helps users route email to the correct person or group without having to know their email address
Databases	DBpedia	DBpedia	Germany /USA	Extracts structured information from Wikipedia and make this information available on the Web
	Freebase	Metaweb Technologies	USA	Open and structured database covering over 4 million topics (60% more than English Wikipedia)
Aggregation tools	BooRah	BooRah	USA	Restaurant review site aggregating reviews from food blogs, Citysearch, Tripadvisor and other large review sites
	Twine	Radar Netwroks	USA	Website used to share knowledge and information
Personal information management	Triplt	Triplt	USA	Travel planning application used to organize and share travel plans
Professional applications	Talis	Talis	UK	Platform focused on library management
	Calais	Thomson Reuters	USA	Toolkit of products that enable users to incorporate semantic functionality within their blog, content management system, website or application

Figura 2.10: le applicazione del semantic web [9]

Sebbene il Semantic Web sia molto promettente, le compagnie che se ne occupano hanno difficoltà ad affermarsi sul mercato. Tale mancanza di risultati è dovuta ad alcune barriere che devono ancora essere superate:

- La complessità tecnologica del Semantic Web rappresenta un ostacolo, soprattutto se confrontata a quella di sistemi come AJAX (utilizzato nelle applicazioni del web 2.0), che sono costruiti su tecnologie più semplici e quindi più facili da adottare.
- Le tecnologie del web semantico non saranno davvero utili finchè non saranno implementate su larga scala, ma gli investitori sono restii a impegnare denaro in

tecnologie la cui utilità non sia già stata dimostrata, e questo crea un circolo vizioso di difficile soluzione.

- Le applicazioni consumer basate sulle tecnologie semantiche potrebbero non raggiungere l’atteso successo per svariati motivi tra cui la difficoltà a monetizzare l’utilizzo da parte dei loro clienti come fanno invece i siti web 2.0, a causa dei nuovi modi di navigare in Internet, della necessità di individuare i formati pubblicitari più appropriati, o di assicurare la privacy degli utenti. Tali incertezze, quindi, rendono non sicuro il ritorno degli investimenti.

Information Management

Sempre più compagnie e governi hanno compreso che l’IT non riguarda solo l’utilizzo automatizzato dei dati, ma l’utilizzo dell’informazione come risorsa strategica per migliorare l’efficienza del business, valutare nuove opportunità, ridurre il rischio operativo, sfruttare a pieno le potenzialità di una società, di un settore, di un’economia.

La gestione delle informazioni è un segmento strategico, in quanto il valore e il vantaggio competitivo di innumerevoli imprese è basato sui loro stessi processi (SOA, Business Process Management / strumenti di analisi e ottimizzazione) e sul loro patrimonio informativo, ottenuto grazie alla diffusione crescente di dispositivi personali e di business, che intensificano la connettività tra persone, organizzazioni, istituzioni. Le aziende hanno bisogno di incanalare tutta questa intelligenza distribuita per poterla riutilizzare in modo a loro vantaggioso.

Conclusioni

Come si può desumere da quanto descritto in questo capitolo sull’IoS, la ricerca e la selezione di servizi saranno sicuramente due degli aspetti più importanti e delicati da affrontare, e non potranno quindi essere demandati all’operato dei singoli individui. Una valida soluzione può essere l’utilizzo dei cosiddetti Recommender System (RS), cioè sistemi che guidano gli utenti, durante la fase di scelta e acquisto, tramite suggerimenti, sulla base di informazioni fornite dai medesimi, filtrandole e elaborandole con algoritmi più o meno complessi, che vanno dal semplice confronto della similarità dei profili-utente, fino alle più complicate tecniche di elaborazione dei dati relativi agli utenti. Nel prossimo capitolo vedremo in dettaglio la loro struttura e ne capiremo il funzionamento.

CAPITOLO 3

I Recommender System

Questo capitolo fornisce una visione generale sul mondo dei Recommender System, studiandone le caratteristiche, confrontando tre interessanti casi di studio (Amazon, Flixster e Last.fm), e proponendo una panoramica sullo stato dell'arte. Nell'ultima parte, inoltre, saranno presentate le caratteristiche specifiche dei Recommender System applicati nell'ambito dei servizi.

3.1 Introduzione ai Recommender System

Nei primi anni '90, con la diffusione dei computer e delle prime interfacce grafiche (1984) e l'avvento del World Wide Web (1991), ci si trovò a fronteggiare il problema di gestire grandi moli di dati in formato elettronico. Questa situazione, nell'ambito delle pubblicazioni scientifiche, creava notevoli difficoltà nel trovare ciò che realmente si cercava. Peter Foltz e Susan Dumais proposero, in un loro articolo del 1992, l'idea di fornire informazioni personalizzate, operando una selezione ragionata delle risorse disponibili [10]. Eseguirono un esperimento, nel quale inviarono a 35 impiegati della Bellcore¹, per 6 mesi, degli abstract relativi a Technical Memorandum in base agli interessi da loro dichiarati. La soddisfazione rilevata ad esperimento concluso fu significativa.

Proprio in quel periodo, David Goldberg [11] e i suoi collaboratori stavano sviluppando nel centro di ricerca Xerox di Palo Alto un sistema per il filtraggio automatico delle e-mail, che chiamarono Tapestry. L'obiettivo era trovare un metodo per selezionare, in base alle preferenze dei destinatari, i numerosi messaggi di posta elettronica, il cui utilizzo cominciava ad essere consistente in quegli anni.

L'utilizzo del termine Recommender System, risale al 1997, nell'omonimo articolo di Paul Resnick e Hal Varian [12]. Fino ad allora, secondo la consuetudine, la tecnica utilizzata nei sistemi di filtraggio di dati veniva chiamata *collaborative filtering*. Gli autori giustificarono la loro scelta con due argomentazioni: la prima, che la collaborazione di chi produce i suggerimenti non è necessariamente esplicita; la seconda, che i suggerimenti proposti possono comprendere anche consigli che, adottando un rigido filtraggio, sarebbero stati esclusi.

Secondo le statistiche della Nielsen Company [13] negli Stati Uniti avvengono 10 miliardi di ricerche mensili, una media di circa 300 milioni al giorno. Verosimilmente, qualsiasi individuo che abbia accesso ad un computer ha eseguito almeno una volta nella sua vita una ricerca in Internet.

Il motore di ricerca più utilizzato è Google con una quota del 65,5% [14]. La ragione del suo successo è da ricondurre certamente all'algoritmo utilizzato, PageRank, che fornisce risultati

¹ La Bellcore è una società di ricerca nel settore delle telecomunicazioni, nata nel 1984 dalla separazione con AT&T. Attualmente è conosciuta con il nome di Telcordia. <http://www.telcordia.com/>

effettivamente rilevanti, ma anche alla personalizzazione dei risultati della ricerca. Infatti, senza dover eseguire l'accesso a un account Google, l'esperienza utente viene adattata in base alle informazioni sulle sue ricerche precedenti, identificate tramite un cookie memorizzato sul pc. I risultati cambiano anche se viene utilizzato un pc diverso, oppure se lo stesso pc viene utilizzato in luoghi differenti ².

Un Recommender System (RS), secondo la definizione, è un software di filtraggio di contenuti che guida un utente nelle sue scelte, fornendo risultati personalizzati. In tal senso, quindi, Google si può considerare un RS per le pagine web. Più in generale, un Recommender System può essere applicato in un qualsiasi contesto in cui le preferenze dell'utente siano una discriminante fondamentale per la ricerca: film, video, musica, libri, notizie, immagini, prodotti ma anche servizi.

Le società di eCommerce, in particolare, sono in permanente competizione tra loro, cercando di prevalere l'una sull'altra in una sfida senza fine. A tale scopo investono importanti cifre di denaro per capire ciò che desiderano i clienti. L'importanza dei Recommender System in questo ambito, quindi, è sempre più evidente, come testimonia anche il Netflix Prize, un concorso che mette in palio un milione di dollari per chi riesce a ideare un RS che sia migliore di quello di Netflix.

3.1.1 Definizione di Recommender System

In [15] viene proposta la seguente definizione: i Recommender system rappresentano un'ampia famiglia di strumenti che forniscono suggerimenti automatici personalizzati ad un utente, basandosi sulle sue preferenze.

Una possibile definizione formale del compito affidato ad un Recommender System la troviamo sempre in [15]: Sia C l'insieme di tutti gli utenti di una applicazione che adotta un RS. Sia S l'insieme di tutti i possibili item che il RS può proporre agli utenti. Sia R un insieme finito, discreto e totalmente ordinato (es. i numeri naturali da 1 a 10). Sia u una funzione che misura l'utilità di un oggetto $s \in S$ per un utente $c \in C$ tale che $u: C \times S \rightarrow R$. Per ogni utente c di C il Recommender System deve scegliere un oggetto s_{best} di S , che massimizzi la funzione u .

La principale difficoltà insita in questo task è che la funzione u non è definita per tutte le coppie (utente, oggetto), quindi non è possibile stabilire quale sia l'oggetto migliore per un utente. La soluzione a questo problema dipende dal tipo di RS, come vedremo nel sottocapitolo 3.3.

3.1.2 Caratteristiche di un Recommender System

Lo scopo di un Recommender System è fornire all'utente suggerimenti che corrispondano alle sue preferenze. Per raggiungere tale obiettivo, ci sono una serie di proprietà che un Recommender System dovrebbe garantire [16][17]: la trasparenza, la possibilità di correzione, l'affidabilità, l'efficacia, la capacità di persuasione, l'efficienza, la soddisfazione del cliente, l'ampiezza dei risultati, l'aggiornamento costante, la serendipità, la varietà dei suggerimenti, la robustezza, il rispetto della privacy, la scalabilità.

La trasparenza

Un RS dovrebbe spiegare il perché dei risultati che fornisce. Infatti, un utente potrebbe ricevere dal sistema dei suggerimenti che non corrispondono alle sue preferenze. Se il Recommender System spiega per ogni contenuto proposto il motivo per cui lo ha selezionato, l'utente ha modo di capire se l'errore dipende dal sistema che funziona male, oppure dalla non completezza, o dall'imprecisione, delle informazioni che lui stesso ha fornito al RS. Nel

² Ad esempio, in base all'indirizzo ip viene scelta la lingua dei risultati.

primo caso l'utente dovrebbe poter interagire in qualche modo con il sistema per potergli comunicare lo sbaglio, nel secondo caso, può intervenire direttamente sui suoi dati personali, correggendoli.

La possibilità di correzione

Dato che un Recommender System può commettere degli errori, fornendo risultati non appropriati alle caratteristiche di un utente, questi deve poter intervenire, segnalando al sistema l'errore. Le azioni possibili, in questo caso, sono di due tipi: rifiutare un determinato suggerimento, ad esempio segnalando che non è di proprio gradimento, oppure applicare un filtro definitivo verso un certo tipo di contenuti, ad esempio, nel caso di un Movie RS, impostando il blocco di suggerimenti verso film che hanno come attore Arnold Schwarzenegger. Nel primo caso, se il Recommender System è di tipo adattivo (vedi sottocapitolo 3.3), i successivi suggerimenti terranno conto dell'informazione ricevuta dall'utente, utilizzandola come ulteriore espressione di preferenze, altrimenti il sistema si limiterà a non sottoporre quel particolare contenuto nei suggerimenti futuri. Nel secondo caso, invece, si ha una modifica stabile dei criteri di ricerca.

L'affidabilità

L'utente dovrebbe incrementare la sua fiducia verso il sistema man mano che lo utilizza. Ovviamente questo dipende moltissimo dall'accuratezza dei risultati proposti, ma non solo, infatti da alcuni studi si è evidenziato che giocano un ruolo importante anche la trasparenza [18] e la possibilità di interagire con il sistema [19]. In generale, inoltre, un utente ha maggiore fiducia nei confronti di quei siti che hanno una interfaccia grafica gradevole, quindi anche per un recommender system è importante che il design con cui si presenta sia accattivante [20]. È comunque evidente che questi elementi sono sufficienti per creare una fiducia di partenza verso il sistema, ma che nel lungo periodo l'utente continuerà ad usare il RS solo se i risultati sono coerenti alle sue preferenze.

L'efficacia

Si può affermare che un Recommender System è efficace se aiuta gli utenti a compiere scelte che si rivelano soddisfacenti. Un modo per rilevare l'efficacia di un RS è l'utilizzo dei feedback, infatti, se un utente prende una decisione sulla base del suggerimento ricevuto e dopo l'interazione con l'oggetto della scelta fornisce una valutazione, è possibile capire se il suggerimento è stato efficace oppure no. Si pensi, ad esempio, ad un Recommender System per un sito che vende libri. Il sistema fornisce i titoli che dovrebbero interessare all'utente. Se dopo l'acquisto e la lettura, il libro consigliato non si rivela appropriato ai gusti del lettore, significa che l'algoritmo utilizzato dal RS non è sufficientemente efficace.

La capacità di persuasione

Un Recommender System non dovrebbe fungere solo da strumento al servizio degli utenti per guidarli nelle scelte, bensì dovrebbe essere anche un sistema di promozione dei contenuti che invogliano gli utenti a visitare una pagina Internet, a comprare un prodotto, o ad acquisire un servizio. Certamente i suggerimenti devono comunque essere validi e corrispondere alle attese dell'utente. Spingere troppo sul tasto della persuasione senza tener conto di questi aspetti è, infatti, un'arma a doppio taglio: una volta che gli utenti sono stati convinti e hanno dato retta al suggerimento, se le loro aspettative non sono state rispettate, difficilmente continueranno ad utilizzare il sistema.

L'efficienza

L'efficienza di un Recommender System consiste nel rapporto tra la qualità del servizio offerto dal Recommender System (quindi l'efficacia) e gli sforzi compiuti da un utente per

trasmettere al sistema le proprie preferenze. L'utente può comunicare le proprie inclinazioni esplicitamente, compilando il proprio profilo personale e selezionando di volta in volta i contenuti che gli piacciono, oppure implicitamente, e in questo caso è il sistema che svolge il ruolo attivo, osservando i comportamenti dell'utente, tenendo traccia dell'history di navigazione, e analizzando le caratteristiche comuni dei contenuti preferiti. Maggiore è la propensione del sistema verso questa seconda modalità e maggiore è l'efficienza del sistema.

La soddisfazione dell'utente

L'utente è soddisfatto se trova esattamente ciò che cerca. È altrettanto vero, però, che la soddisfazione dell'utente è legata anche al tipo d'interazione che questi ha con il sistema: più articolato e complesso è l'utilizzo del Recommender System e maggiore è la frustrazione dell'utente. Un RS, quindi, deve essere facile da usare, intuitivo e possibilmente curato dal punto di vista dell'interfaccia grafica. Inoltre, gli utenti apprezzano molto la presenza di descrizioni complete degli oggetti suggeriti, considerandole propedeutiche a una scelta più consapevole [18].

L'ampiezza dei risultati

I suggerimenti di un RS possono essere molto accurati ma, allo stesso tempo, possono non includere dei contenuti potenzialmente interessanti per l'utente. Ciò dipende dal *dataset* che il Recommender System ha a disposizione. La taglia dei dati a disposizione di un sistema è detta *Item Space Coverage* [17]. Un buon RS dovrebbe garantire la massima copertura possibile. Ad esempio, il dataset di libri del sito Amazon.it comprende una vastissima selezione di titoli (oltre 2 milioni³), quindi i suggerimenti forniti dal sistema sono da considerarsi esaurienti. Sempre legato all'ampiezza dei risultati è il concetto di *User Space Coverage* [17], cioè l'ampiezza della base utenti. Maggiore è il numero di utenti di un Recommender System e maggiori sono le loro interazioni con esso. Un Recommender System che sfrutta la similarità dei profili utente per i suggerimenti, secondo il principio che persone con interessi simili possono gradire gli stessi consigli, avrà una maggiore efficacia se il sistema ha una base utenti molto vasta. Allo stesso modo, se si dà la possibilità agli utenti di aggiungere descrizioni e feedback ai contenuti, un RS con molti utenti fornirà suggerimenti più accurati.

L'aggiornamento costante

Questa proprietà ha un duplice aspetto. Da una parte è necessario fornire all'utente suggerimenti sempre nuovi per stimolare la sua curiosità e dargli l'idea di un sistema vitale e sempre aggiornato. Dall'altra il RS deve poter fornire consigli riguardanti item che sono appena stati introdotti nel suo *item space*, e per cui non esistono ancora sufficienti valutazioni da parte dell'utente. La gestione di questo secondo aspetto è molto delicata. In [15] si propone un approccio di tipo *Case-based Reasoning*, cioè la risoluzione di problemi nuovi, basandosi sulla soluzione di problemi passati simili. Tradotto in questo contesto, significa dare un giudizio iniziale sul nuovo item basandosi sulla somiglianza ad altri item. Certamente non è una soluzione totalmente affidabile, però può essere considerata un buon rimedio temporaneo, in attesa delle reali valutazioni degli utenti.

La serendipity

La serendipità è la sensazione che si prova quando, nel corso di una ricerca, si scopre qualcosa di diverso dall'oggetto della ricerca, in modo del tutto inaspettato e impreveduto. In questo contesto il termine serendipity viene usato per indicare una misura di quanto sorprendenti sono i suggerimenti corretti forniti dal sistema. Ad esempio, se un utente di un sito di film ha valutato positivamente molte pellicole con un determinato attore, il

³ Dati forniti da Amazon nella sezione comunicati dell'area stampa, aggiornati all'anno 2010.
http://www.amazon.it/b/ref=footer_press?ie=UTF8&node=527386031

Recommender System di quel sito può proporgli l'ultimo film con quell'attore. Il suggerimento molto probabilmente sarà apprezzato, però non può essere considerato sorprendente. Con una gestione del tutto casuale dei suggerimenti, al contrario, si producono consigli del tutto imprevedibili, ma solo una piccola percentuale di essi sarà presa in considerazione dall'utente. La soluzione ottimale sarebbe quella di introdurre suggerimenti inattesi ma allo stesso tempo coerenti con le preferenze dell'utente.

La varietà dei suggerimenti

Se i suggerimenti sono molto simili tra loro, l'utente ha a disposizione una ristretta gamma di scelta. Spesso è molto più utile avere un ampio spettro di possibilità diverse tra loro per farsi un'idea generale, e semmai raffinare la ricerca in un secondo tempo, in base alle inclinazioni del momento. Si consideri, ad esempio, un RS per un sito di viaggi. Se il sistema, sulla base delle preferenze dell'utente (es. mare e immersioni), proponesse cinque alternative per lo stesso luogo (es. Polinesia), variando soltanto l'hotel o la compagnia aerea per raggiungere la destinazione, l'utente avrebbe una limitata possibilità di scelta. Sarebbe più vantaggioso, invece, ricevere dal Recommender System cinque proposte per mete differenti e, solo dopo aver scelto la destinazione, decidere per il volo e l'hotel.

La robustezza

La robustezza di un RS è la stabilità dimostrata dal sistema a fronte di informazioni fittizie o artefatte, introdotte di proposito. Gli utenti, tramite i loro feedback hanno la possibilità di determinare le valutazioni dei singoli item. Alcuni individui possono avere un interesse a modificare tali giudizi per scopi personali. Si pensi, ad esempio, al proprietario di un Hotel che vuole promuovere il suo esercizio. Fingendosi un cliente, può inserire una valutazione positiva riguardante il servizio offerto dall'Hotel. Ripetendo tale artificio più volte, indirizza il giudizio complessivo a suo favore, ottenendo una maggiore visibilità. Allo stesso modo, fingendosi cliente di un Hotel vicino al suo, potrebbe introdurre nel sistema feedback negativi per mettere in cattiva luce la concorrenza. Un Recommender System dovrebbe essere in grado di riconoscere comportamenti simili, ma è difficile poter distinguere le valutazioni veritiere da quelle artefatte. Quindi, dando per assodata la presenza di informazioni fittizie, la robustezza di un RS risiede nella capacità del sistema di fornire suggerimenti affidabili, nonostante tali intrusioni.

Il rispetto della privacy

In un Recommender System gli utenti rivelano in modo esplicito, o implicito, le loro preferenze e i loro desideri, con lo scopo di ottenere suggerimenti utili. Tali informazioni personali dovrebbero rimanere private, e non essere diffuse a terze parti a fini commerciali. Il pericolo è di essere letteralmente travolti da spam di ogni genere, attraverso email, ma anche con la proposizione di pubblicità "personalizzate" in siti diversi da quello a cui appartiene il RS consultato. Navigando in Internet, infatti, è abbastanza comune trovare pubblicità di Hotel di una certa località, subito dopo aver cercato su un sito di viaggi un alloggio per quella determinata destinazione. Tali situazioni, ovviamente, non sono gradevoli per un utente. Un Recommender System, quindi, dovrebbe avere una regolamentazione della privacy che garantisca agli utenti la riservatezza dei loro dati sensibili, senza cessioni a terze parti.

La scalabilità

Un RS deve gestire grandi moli di item e di profili utente. Gli algoritmi su cui si basa il sistema devono, perciò, essere in grado di gestire un rapido aumento sia degli uni che degli altri. Infatti, man mano che il sito associato al Recommender System aumenta la popolarità e, quindi, le visite, crescono gli utenti e, in proporzione, anche le loro ricerche e i loro feedback. Inoltre, si espande pure l'itemset, a causa di due processi paralleli e complementari: da una

parte c'è l'aggiornamento costante dovuto alla comparsa di nuovi item, dall'altra c'è l'inserimento, da parte degli utenti, di item che già esistevano ma che non erano conosciuti dal sistema. Il RS, perciò, oltre a dover garantire algoritmi efficienti in grado di gestire grandi moli di dati, deve anche avere sufficienti risorse, in termini computazionali e di memoria, per gestire tali problematiche legate alla scalabilità.

Idealmente un RS dovrebbe avere tutte le caratteristiche sopracitate ma nella realtà ciò non accade. Applicazioni diverse possono avere esigenze diverse, e si cerca un trade-off in base a tali necessità. Ad esempio, in alcuni casi l'accuratezza dei risultati (efficacia) può essere trascurata a favore di una maggiore varietà dei suggerimenti. In questa tesi, tuttavia, ci si propone di progettare un Recommender System che si attenga il più possibile alle proprietà precedentemente elencate.

3.1.3 Esempi di utilizzo di Recommender System

Vediamo ora tre esempi di siti web che integrano nelle loro applicazioni un Recommender System. Il primo è Amazon, che utilizza il RS a fini commerciali, per proporre ai clienti i prodotti più adeguati. Il secondo esempio è Flixster, un social network dedicato al mondo del Cinema, dove il Recommender System viene utilizzato per consigliare all'utente i film che potrebbero interessargli. Il terzo, infine, è Last.fm, un sito web che fornisce suggerimenti musicali sulla base della somiglianza tra i cantanti.

Amazon

Amazon è una compagnia di commercio elettronico statunitense con sede a Seattle, nello stato di Washington. Fondata con il nome di Cadabra.com da Jeff Bezos nel 1994 e lanciata nel 1995, Amazon.com ha iniziato come sito di vendita di libri online, ma nel giro di poco tempo ha allargato la gamma dei prodotti offerti a DVD, CD musicali, software, videogiochi, prodotti elettronici, abbigliamento, mobilio, cibo e giocattoli. Amazon ha poi esteso la sua rete, creando siti anche per i mercati di Canada, Regno Unito, Germania, Austria, Francia, Italia, Cina e Giappone.

Il Recommender System di Amazon è basato sullo storico degli acquisti di un utente, sulle valutazioni dei prodotti più venduti, e sui giudizi personali relativi a oggetti proposti in passato dal sistema. Inoltre vengono indicate anche le preferenze dei clienti con profili simili, favorendo la conoscenza di nuovi prodotti che potrebbero risultare interessanti.



Figura 3.1: logo di Amazon

Flixster

Flixster è un social network, creato da Joe Greenstein e Saran Chari nel 2005, che permette agli utenti di condividere valutazioni sui film, scoprirne di nuovi, e incontrare persone con i gusti simili ai propri. Lo slogan, infatti, è *"Watch movies. Tell friends"*.

Nel periodo Marzo – Novembre 2006 si è assistito ad un incremento delle visite giornaliere da 4 milioni a 31 milioni, mentre i visitatori unici sono passati da 328 000 a 1 milione. Nel gennaio 2007 le visite giornaliere hanno toccato i 50 milioni, facendo pensare ad un incremento crescente e continuo. In realtà un anno dopo (gennaio 2008) il numero di pagine

visitate è sceso vertiginosamente a circa 8 milioni, e ha continuato la caduta libera, fino a non riuscire a raggiungere il milione e mezzo nel luglio 2008, quando gli utenti effettivamente attivi rilevati sono stati appena 400 000.

Flixster, per porre rimedio a questo declino, ha deciso di sviluppare applicazioni per social network esterni. Tali app possiedono le principali funzionalità offerte dal sito principale, come i suggerimenti, le valutazioni, le recensioni e i quiz generati dagli utenti. La prima di queste applicazioni, *Movies*, è stata realizzata nel giugno 2007 per Facebook. Nel marzo del 2008, poi, è stata la volta di MySpace, raggiungendo nel giro di pochi mesi il secondo posto tra le applicazioni più popolari di questo social network. Successivamente, sono state realizzate le app anche per Bebo e Orkut. Nell'agosto del 2008, Flixster ha realizzato un'applicazione per iPhone, che permette agli utenti di accedere a news, recensioni e trailer, e inserire valutazioni sui film. Analoghe applicazioni sono state sviluppate, successivamente, per Blackberry e cellulari con SO Android e Windows Mobile.

Il sistema di raccomandazioni di cui si avvale Flixster si basa sulla somiglianza tra i film. Se un film A, secondo il parere degli utenti, ha una trama simile ad un film B, e l'utente X ha valutato positivamente il film A, allora il RS suggerirà B a X. Se X trova interessante B può decidere di vederlo e poi valutarlo, oppure inserirlo nella lista dei film da vedere e valutarlo in un secondo momento. Qualunque sia l'azione intrapresa da X, essa sarà registrata dal sistema e influenzerà i suggerimenti successivi.



Figura3.2: logo di Flixster

Last.fm

Last.fm è un sito di musica, che propone ai propri utenti consigli personalizzati riguardo canzoni che rispecchiano i loro gusti. Fondato nel 2002 in Gran Bretagna, possiede circa 40 milioni di utenti attivi in oltre 190 paesi. Nel Maggio del 2007 è stato acquisito da CBS Interactive per 140 000 sterline (circa 170 000 euro).

Last.fm Si avvale di un music recommender system chiamato Audioscrobbler, grazie al quale è possibile ricostruire un dettagliato profilo dei gusti di ciascun utente. A supporto di tale funzione è possibile scaricare un software, lo Scrobbler, che memorizza le canzoni ascoltate tramite i propri lettori musicali, come Windows Media Player, iTunes, Winamp, e molti altri ancora.

Lo Scrobbler invia a Last.fm dei messaggi, i cosiddetti scrobbling, per segnalare il brano che l'utente sta ascoltando. In tal modo vengono memorizzati i brani che l'utente ascolta più spesso, il numero di volte che ha ascoltato canzoni di un certo artista, gli amici che hanno gusti simili al suo e così via. Analizzando la musica ascoltata, il RS di Last.fm aiuta l'utente a scoprire nuove canzoni e artisti, coerenti ai suoi gusti, attraverso consigli personalizzati. Inoltre è possibile aggiungere tag ai brani, contribuire alle discussioni, aggiornarsi sulle ultime tendenze e scoprire le statistiche delle proprie cronologie di ascolto.

Il servizio offerto da Last.fm è quasi totalmente gratuito. Ci si può registrare, creare un profilo utente e utilizzare la maggior parte delle funzionalità senza esborsi economici, ad eccezione della radio, che non prevede costi solo negli Stati Uniti, in Gran Bretagna e in Germania. In tutti gli altri paesi è disponibile una prova gratuita che include 50 brani, scaduta la quale è necessario abbonarsi, versando una somma mensile di 3 euro.



Figura 3.3: logo di Last.fm

3.2 Analisi comparativa di Recommender System

In questo sottocapitolo vengono confrontati i Recommender System di Amazon, Flixster e Last.fm, dal punto di vista dell'utente finale, secondo diversi aspetti quali la registrazione, la personalizzazione del profilo, i suggerimenti, le partnership e le problematiche specifiche. A tale scopo ho testato in prima persona i sistemi, utilizzandoli per un mese e annotando caratteristiche, pregi e difetti. La scelta è ricaduta proprio su questi tre siti per due motivi: il primo è di natura tecnica, infatti utilizzano tre diversi approcci nel fornire suggerimenti, consentendo una visione ampia sulle possibili soluzioni; il secondo invece è prettamente economico, difatti si possono utilizzare liberamente, senza dover corrispondere alcun pagamento.

3.2.1 Registrazione

La fase di registrazione non riguarda direttamente il Recommender System, infatti serve più che altro per identificare l'utente e creare un legame tra lui e il sito di riferimento. Si deve comunque tener presente che la registrazione fornisce anche la prima impressione sul sito in cui è presente il Recommender System, e se richiede troppo tempo o necessita di troppe informazioni personali, può scoraggiare l'utente a proseguire. Poiché un RS, per essere efficace, deve avere una base utenti molto vasta, è essenziale che gli utenti siano invogliati ad utilizzarlo, assicurandosi di eliminare qualsiasi deterrente. Il tradeoff, dunque, è tra la necessità di identificare l'utente e l'esigenza di semplicità.

Amazon

La registrazione al sito è molto semplice e rapida: prevede soltanto l'inserimento del proprio nome, di un indirizzo email valido e di una password, scelta a piacere. Si noti, tuttavia, che il consenso ai termini di utilizzo e alla politica sulla privacy definiti da Amazon è implicito, e non viene espressamente scritto nella pagina di accesso al sito.

Registrazione
Sei un nuovo cliente?
Il mio nome è:
Il mio indirizzo e-mail è:
Inserisci di nuovo per confermare:
Scegli una password
Questa sarà la tua password per Amazon.
Inserisci una nuova password:
Inserisci di nuovo per confermare:

Figura 3.4: form di registrazione di Amazon

Flixster

La registrazione di Flixster prevede l'inserimento di maggiori informazioni, rispetto ad Amazon. Infatti, è necessario inserire email, nome, cognome, data di nascita, sesso, e una password scelta dall'utente. Per evitare registrazioni automatiche da parte di agenti software, viene utilizzato un CAPTCHA⁴, basato su un textbox contenente un codice con caratteri dalla forma distorta. Flixster, inoltre, fornisce agli utenti di MySpace, come alternativa alla compilazione del form, la possibilità di utilizzare il loro account per accedere al sito. In tal caso è sufficiente fornire l'indirizzo email utilizzato su MySpace e la relativa password. Infine viene segnalato esplicitamente che registrandosi a Flixster si accettano automaticamente le condizioni d'uso e la politica sulla privacy da loro definiti.

Register for Flixster
Already a member? [Click here to log in.](#)
 MySpace user? [Register with your MySpace account.](#)

Email:

First Name:

Last Name:

Date of birth: Month Day Year Hide my age

Gender: Female Male

Password:

Enter the words below:

Delisle, ireft

Figura 3.5: form di registrazione di Flixster

Last.fm

La registrazione prevede l'inserimento di un indirizzo email valido, la scelta di una password e il controllo sulle registrazioni automatiche tramite CAPTCHA. Si può affermare, quindi, che la modalità di Last.fm è una via di mezzo tra quella di Amazon (poche informazioni personali) e quella di Flixster (controllo tramite textbox). Diversamente dai due casi precedenti, tuttavia, non viene richiesto il proprio nome, ma un nickname che si può utilizzare come nome utente. Infine, come per Flixster, è necessario accettare esplicitamente i termini di utilizzo e la politica sulla privacy di Last.fm.

⁴ Un CAPTCHA è un test composto da una o più domande per determinare se l'utente sia un umano. L'acronimo deriva dalla sua definizione in inglese: Completely Automated Public Turing test to tell Computers and Humans Apart (Test di Turing pubblico e completamente automatico per distinguere computer e umani).

Figura 3.6: form di registrazione di Last.fm

3.2.2 Personalizzazione del profilo

La personalizzazione del profilo consente al sistema di conoscere l'utente, di capire quali sono le sue preferenze e i suoi principali interessi. Senza questo passaggio i primi suggerimenti che seguono il *sign in* risulterebbero eccessivamente generici, dovendosi basare soltanto sui pochi dati forniti con la registrazione. Questa fase, quindi, è da considerarsi una parte integrante del processo di registrazione e un elemento fondamentale per fornire proposte appropriate all'utente. Le interazioni successive con il sistema, inoltre, consentono di aggiornare il profilo e rendere più accurati i suggerimenti.

Amazon

Immediatamente dopo la registrazione, viene proposta la valutazione di una serie di articoli a scelta tra quelli che una persona già possiede, per conoscere meglio i gusti dell'utente. Non viene richiesta nessuna informazione esplicita riguardo le proprie preferenze, le quali vengono dedotte proprio da queste valutazioni e da quelle future, oltre che dall'analisi degli acquisti fatti e dei prodotti visionati.

Figura 3.7: personalizzazione in Amazon

Per ogni articolo specificato, il motore di ricerca del sito di Amazon fornisce uno o più risultati. L'utente può specificare se lo possiede già, spuntando un'apposita casella, e assegnargli un valore di gradimento, su una scala da una a cinque stelle, dove una stella è il minimo e cinque è il massimo.



Figura 3.8: esempio di valutazione di un articolo già acquistato

Flixster

La personalizzazione del proprio profilo in Flixster è molto più laboriosa di quella in Amazon. Si inizia dalla scelta di uno sfondo per la propria pagina personale. Il passo successivo è l'inserimento di una immagine personale. Dovrebbe essere una foto della persona, ma gli utenti possono decidere di inserire anche quella di un personaggio famoso, di un cartone animato, di un logo, oppure saltare totalmente questo passaggio. Dopo di che viene richiesto di specificare la città e lo stato di provenienza. Infine, l'ultimo passo consiste nell'elencare i film e gli attori preferiti e nell'inserire una breve descrizione di se stessi.

Per migliorare la conoscenza dei gusti dell'utente, il sistema propone un test di valutazione di 50 film tra i più famosi (come Titanic, Il signore degli anelli e Il gladiatore). Per ciascuno di essi, se si tratta di un film già visto, si può inserire una valutazione da una a cinque stelle (ma in una scala che prevede anche i mezzi voti), altrimenti si può esprimere l'intenzione di vederlo in futuro, oppure segnalare di non essere interessati.

Poiché Flixster è un social network, viene sfruttata anche la potenzialità fornita dai consigli degli amici. A tale scopo il sistema mette a disposizione un motore di ricerca, che permette di specificare nazionalità, sesso ed età delle persone con cui si vorrebbe fare amicizia. I risultati della ricerca segnalano per ciascun utente trovato il livello di compatibilità, l'*MCT score*, sulla base della similarità dei profili. La scala prevede 8 categorie: *run away, fast* (da 47 punti in giù), *terrible match* (da 48 a 52), *bad match* (53 punti) *casual buddies* (da 54 a 56), *friends* (da 57 a 60), *good friends* (da 61 a 64), *best friends* (da 65 a 69), *soul mates* (da 70 in su).

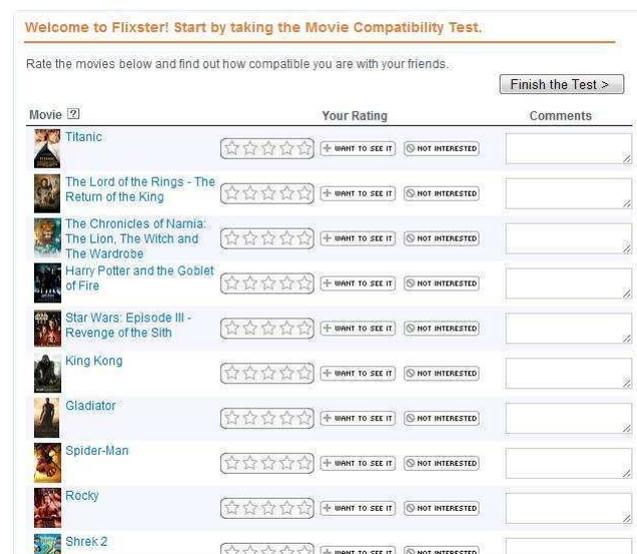


Figura 3.9: test iniziale di Flixster per conoscere i gusti dell'utente

Last.fm

L'unica informazione esplicita richiesta da Last.fm per la personalizzazione del profilo è l'elenco dei propri artisti preferiti. La maggior parte dei dati relativi a un utente, infatti, sono dedotti dall'analisi dei brani che ascolta sul sito e dalle valutazioni che, di volta in volta, può esprimere su di essi. A supporto di questo processo di osservazione, inoltre, è stato sviluppato un software, scaricabile sul proprio computer, con il quale si possono registrare i titoli e gli autori delle canzoni ascoltate con i lettori audio del pc, e inviare tali informazioni al sito.



Figura 3.10: personalizzazione del profilo in Last.fm

3.2.3 I suggerimenti

I suggerimenti sono l'elemento di maggiore importanza per un Recommender System. Devono essere appropriati, devono essere utili, devono essere proposti in una forma che stimoli l'interesse dell'utente. I Recommender System, come vedremo nel prossimo sottocapitolo (3.3 *Stato dell'arte*) possono utilizzare strategie e algoritmi diversi per formulare i suggerimenti, ma lo scopo, per tutti, è uno solo: trovare l'item che soddisfi maggiormente l'utente.

Amazon

La homepage di Amazon, com'è normale che sia per un sito di eCommerce, è formata principalmente da suggerimenti, raggruppati in categorie, e per ciascuna di esse vengono proposti 5 prodotti, con la possibilità di visualizzarne altri a discrezione dell'utente. In ordine, dall'alto verso il basso troviamo:

- *Altri prodotti da tenere presente;*
- *Simili a prodotti che hai già visto;*
- *Consigliati in base alla tua cronologia di navigazione;*
- *Consigli d'acquisto per te;*
- *In questo momento gli altri stanno guardando;*
- *Bestseller (Libri, Film e TV, Musica, Elettronica e altro);*
- *Ultimissime uscite.*

Il RS di Amazon, nel fornire suggerimenti, oltre che ai gusti dell'utente, e alla rilevanza assoluta di un prodotto, presta attenzione anche ai prezzi degli articoli. Ciò è stato dedotto da un semplice esperimento effettuato durante il processo di analisi comparativa: nella fase di registrazione al sito, dopo la valutazione dei primi 4 articoli (nello specifico, libri), che

avevano un costo medio di 14.60 euro, il Recommender System proponeva come suggerimenti del giorno alcuni libri il cui costo oscillava tra i 10.65 euro e i 18.68 euro. Proseguendo con la valutazione di ulteriori libri, sono stati presi in considerazione testi più economici, e si è potuto notare un sensibile cambiamento della fascia di prezzo dei libri consigliati: da un minimo di 4.50 euro, ad un massimo di 8.71 euro.

Altro elemento importante, peculiare dei suggerimenti forniti da Amazon nella homepage, è il continuo rinnovamento delle proposte: i consigli personalizzati cambiano ad ogni accesso al sito. Questa strategia è finalizzata al raggiungimento di tre obiettivi: il primo, incuriosire l'utente; il secondo, garantire la vitalità del sito; il terzo, spingere il cliente a visionare subito gli articoli proposti.

I suggerimenti, però, non sono confinati alla homepage, bensì sono presenti in ogni parte del sito. Le sezioni dedicate alle otto tipologie di prodotti (Libri, Musica Film e TV, Elettronica e altro, Informatica, Videogiochi, Giocattoli, Elettrodomestici, Orologi) possiedono schemi diversi per la proposizione degli articoli (ciò è dovuto alle caratteristiche esclusive di ciascuna categoria), ma è comunque possibile distinguere una struttura comune:

- *In primo piano;*
- *Consigliati per te;*
- *Novità e prossime uscite;*
- *Bestseller.*

Infine, anche le pagine dei singoli prodotti contengono suggerimenti. In questo caso, però, il sistema non tiene conto delle preferenze dell'utente, ma si basa esclusivamente sulle azioni degli altri clienti. Infatti, le tre tipologie di proposte sono:

- *Cosa comprano gli altri clienti dopo aver visto questo prodotto?;*
- *Spesso comprati insieme;*
- *I clienti che hanno visto questo articolo hanno visto anche.*

Volendo trovare una definizione all'organizzazione di suggerimenti descritta, si può dire che il Recommender System di Amazon utilizza una struttura gerarchica. Al vertice ci sono i consigli della homepage, molto generici ma altamente personalizzati, che hanno lo scopo di attirare l'attenzione dell'utente; a seguire troviamo quelli delle categorie, maggiormente specifici ma meno attenti agli interessi dell'utente; infine, al livello più basso, ci sono i suggerimenti relativi ai prodotti, che non sono correlati direttamente ai gusti del cliente, ma mirano ad allargare il campo della sua ricerca, proponendo articoli che probabilmente non stava cercando ma che possono indurlo a un acquisto non preventivato.

Flixster

Flixster è un movie social network, quindi, diversamente da Amazon, la sua homepage non contiene soltanto suggerimenti ma anche quiz, aggiornamenti relativi agli amici virtuali, trailer, foto e notizie sul mondo del cinema. Questa moltitudine di contenuti, ovviamente, riduce la rilevanza delle proposte del RS, disperdendo l'attenzione dell'utente.

I suggerimenti presenti nella pagina principale sono di due tipologie:

- film in uscita al cinema (*Opening This Week*);
- consigli personalizzati (*Recommendations For Me*).

Osservando attentamente le proposte relative ai film della prima categoria, mi sono accorto che, in realtà, sono uguali per tutti gli utenti. A mio avviso, quindi, la prima tipologia di suggerimenti non è riconducibile al Recommender System.

Diversamente dall'homepage, le pagine dedicate ai suggerimenti sono totalmente focalizzate sui consigli all'utente. Non ci sono fattori di distrazione, troviamo soltanto elementi inerenti ai film proposti. Per conferire maggiore risalto ai singoli suggerimenti, sono presentati uno alla volta. Inoltre il sistema spiega la ragione per cui è stato consigliato un certo film (pannello ...*because you liked*), e riporta le valutazioni date dagli amici.

Infine è possibile definire i parametri dei suggerimenti, specificando quali generi si vogliono includere nella ricerca. In questo modo si lascia all'utente la possibilità di agire direttamente sui risultati forniti dal Recommender System.

Passando alle pagine dei singoli film, oltre alle informazioni su trama, attori e registi, rating degli utenti e dei critici ci sono anche suggerimenti per film simili a quello selezionato (*More Like This*).

Last.fm

I suggerimenti di Last.fm sono basati sulle preferenze dell'utente e sulla somiglianza dei cantanti o delle canzoni. La misura di similarità viene calcolata automaticamente dal software Audioscrobbler in base alle abitudini di ascolto degli utenti. La regola è molto semplice: se un numero considerevole di utenti ascolta un certo artista X, ma anche Y e Z, questi ultimi due artisti saranno considerati dal sistema simili a X. La stessa cosa vale per le canzoni. Questo approccio, abbastanza semplicistico, risulta comunque efficace per il contesto di applicazione. I suggerimenti presenti nella homepage sono di tre tipi:

- *Consigli di Last.fm*
- *La radio dei tuoi consigli*
- *Brani preferiti dagli amici*

La prima tipologia, a sua volta, si suddivide in quattro sezioni:

1. *Nuovi album*. Pubblicizza gli album appena usciti, in base alle preferenze dell'utente. Fornisce anche la possibilità di scaricarli in versione digitale, ma anche di acquistare il cd, tramite siti partner come Amazon, iTunes e eBay.
2. *Musica*. Propone artisti simili a quelli ascoltati normalmente dall'utente. Il Recommender System permette di selezionare uno dei cantanti, e scegliere una sua canzone da un apposito elenco; in alternativa, ci si può affidare al sistema (cliccando sul pulsante *ascolta i tuoi consigli*), che esegue alcuni brani scelti in modo casuale tra quelli disponibili per gli artisti consigliati.
3. *Eventi*. Promuove concerti, ed eventi in generale, che siano coerenti con le inclinazioni musicali dell'utente. Fornisce informazioni su date, orari, luoghi, e permette agli utenti di inserire o modificare i dati in qualsiasi momento.
4. *Download gratuiti*. Offre mp3 gratuiti. Quando disponibili, appropriati ai gusti musicali dell'utente.

La seconda tipologia di suggerimenti della homepage è *la radio dei tuoi consigli*, una funzione a pagamento, disponibile in versione di prova gratuita per i primi 50 brani. Propone, in modo random, canzoni degli artisti preferiti, e di artisti simili. Durante l'esecuzione l'utente può esprimere un parere sulle canzoni, può inserirle tra i preferiti, o decidere di escluderle.

Il terzo tipo di suggerimenti fa riferimento alle preferenze degli amici virtuali. In questo frangente, quindi, il profilo dell'utente non è rilevante ai fini delle scelte del RS.

Nella pagina delle singole canzoni, invece, si trovano l'anteprima del brano, le relative statistiche di ascolto da parte di tutti gli utenti di Last.fm e gli album che lo contengono. Anche qui, come per le pagine dedicate ai film di Flixster, il tipo di suggerimento mira a proporre contenuti simili, infatti l'etichetta degli unici consigli presenti è "Musica simile".

3.2.4 Partnership

I Recommender System, o i siti in cui sono integrati, possono avvantaggiarsi della partnership con altri siti, conseguendo una maggiore diffusione tra i navigatori di Internet e parallelamente poter essere utilizzati anche su altre piattaforme.

Amazon

Amazon è autoreferenziale, nel senso che il suo RS è stato creato appositamente per il sito, quindi i prodotti che raccomanda sono solo quelli che effettivamente vende. L'importanza e la notorietà dell'azienda, inoltre, permette ad Amazon di non dover ricorrere a siti partner per farsi conoscere. Tuttavia la diffusione dei nuovi social media ha indotto la società a creare soluzioni innovative, in particolare l'integrazione del profilo Amazon con quello Facebook. Questa funzionalità, per il momento disponibile soltanto nella versione Beta, consente l'accesso a una sconfinata fonte di dati personali, utilissimi per società di eCommerce come Amazon. Ad esempio, si sfruttano le amicizie e le preferenze dell'utente per generare suggerimenti (es. *Popular Among Your Facebook Friends, Recommendations Based on Your Favourite Books on Facebook*).

Flixster

Flixster è un Movie Social Network, quindi i suoi obiettivi sono ben diversi da quelli di Amazon, e ciò si traduce in una maggiore apertura rispetto a possibili partnership che gli consentano, innanzi tutto di farsi conoscere da più persone possibili e secondariamente di essere utilizzato attraverso più interfacce. Tali esigenze, insieme alla diminuzione degli utenti attivi nel tempo, hanno portato alla creazione dell'applicazione *Movies* per Facebook, che sostanzialmente ricalca nella grafica e nelle funzioni il sito principale, integrando però in automatico la lista degli amici, con l'aggiunta di quelli appartenenti sia a Facebook sia all'applicazione. Tuttavia non viene sfruttata a pieno l'opportunità offerta dal Social Network di generare suggerimenti basati sulle indicazioni (esplicite o implicite) degli amici. Oltre all'applicazione Facebook, sono state sviluppate soluzioni simili per MySpace, Bebo e Orkut. A partire dal 2008, invece, Flixster si è rivolta anche al mercato *mobile*, realizzando applicazioni per iPhone, Blackberry e smartphone con SO Android e Windows Mobile. Flixster, infine, ha creato una partnership con iTunes per permettere agli utenti il download (a pagamento) dei loro film preferiti.

Last.fm

Anche Last.fm non si è potuta sottrarre al fascino di Facebook e ha creato un'applicazione ad hoc al suo interno. Il risultato però non è dei migliori, infatti, ci si accorge subito che non fornisce nessuna funzionalità utile all'utente, se non l'elenco dei brani ascoltati, e che l'unico beneficiario dell'utilizzo è Last.fm stesso, il quale può accedere ai numerosi dati personali del profilo. Questa scelta appare poco felice, in quanto, dopo la curiosità iniziale, l'utente è portato ad abbandonare l'applicazione e probabilmente a rimuoverla dal suo elenco personale. Per l'acquisto di brani ascoltati è stata stretta una collaborazione con 7digital⁵ e iTunes, mentre per avere la canzone come suoneria del cellulare è disponibile ToneFuse⁶.

3.2.5 Problematiche

Dopo aver utilizzato per circa due mesi Amazon, Flixster e Last.fm ho individuato, per ciascuno di essi, quali sono le problematiche più evidenti.

⁵ <http://us.7digital.com/>

⁶ <http://www.tonefuse.com/>

Amazon

Nel corso del periodo di analisi del RS, sono stati ricevuti suggerimenti di buona qualità e con una pertinenza crescente nel tempo. L'unica nota negativa, se proprio bisogna indicarne una, è la mancanza di un'esplicita richiesta di accettazione della politica di privacy.

Flixster

Flixster presenta molte lacune, che determinano un giudizio su di esso complessivamente negativo: innanzi tutto la personalizzazione del profilo richiede troppe valutazioni per capire i gusti dell'utente (scarsa efficienza); i suggerimenti proposti riguardano film poco recenti; i film consigliati, sebbene l'utente indichi esplicitamente una valutazione, vengono ripresentati più volte dal RS; il dataset dei film suggeriti non è aggiornato⁷; infine, vi è una palese violazione della privacy nell'utilizzo dei contatti presenti nella rubrica del proprio profilo MSN, per inviare email pubblicitarie.

Last.fm

L'esperienza utente con Last.fm è stata abbastanza soddisfacente. Il pregio più evidente è la possibilità di scaricare lo Scrobber, il software che permette di rilevare in automatico le preferenze dell'utente, in base alla musica ascoltata sul PC e sincronizzabile con il proprio lettore MP3. Gli unici problemi riscontrati, ma non di poca rilevanza, sono la limitata ampiezza dei risultati e la mancanza dei brani più recenti. I suggerimenti, infatti, sono troppo legati al paese di appartenenza dell'utente. Nel mio caso, ad esempio, i brani consigliati erano quasi esclusivamente italiani. La carenza nell'aggiornamento del dataset delle canzoni, invece, è da imputare probabilmente a questioni di diritti d'autore. Tenendo presente che Last.fm è gratuito, questo potrebbe essere anche un difetto accettabile.

3.2.6 Analisi schematica delle caratteristiche dei tre RS

In relazione alla lista delle caratteristiche che dovrebbe avere un Recommender System, elencate nella sezione 3.1.2, ho fatto un'analisi qualitativa dei tre sistemi recensiti, secondo un giudizio personale, ma basato sui dati oggettivi rilevati ed esposti nella trattazione appena conclusa. I risultati di tale studio sono stati riassunti nella seguente tabella, le cui celle hanno tre differenti colori: il blu indica che la caratteristica in questione appartiene al sistema o è stata valutata positivamente; con l'azzurro si segnala che la caratteristica è presente in parte o è stata valutata mediamente positiva; infine, le celle grigie significano che la caratteristica non è stata rilevata o è stata valutata negativamente.

	Trasparenza	Possibilità di Correzione	Affidabilità	Efficacia	Persuasione	Efficienza	Soddisfazione Cliente	Ampiezza Risultati	Agg. Costante	Serendipità	Varietà Suggerimenti	Robustezza	Rispetto Privacy	Scalabilità
Amazon														
Flixster														
Last.fm														
Legenda	Caratteristica presente o valutata positivamente Caratteristica presente in parte o valutata mediamente positiva Caratteristica assente o valutata negativamente													

Tabella 3.1: riassunto delle caratteristiche dei RS confrontati

⁷ Il film più recente consigliatomi dal RS risale al 2008.

3.3 Stato dell'arte

Per molto tempo le principali categorie di RS erano sostanzialmente due: i Collaborative Filtering (CF) e i Content Based (CB). Al primo gruppo appartengono i Recommender System che per generare i suggerimenti sfruttano la somiglianza degli utenti. Nel secondo, invece, rientrano quelli che si avvalgono della somiglianza degli item. Negli anni sono state sviluppate varianti sulla base di queste due tipologie, senza però introdurre fondamentali novità. Poi, intorno alla metà degli anni 2000, c'è stato il boom dei Semantic Recommender System e, successivamente, l'integrazione di questi con il Social Web, che si è definitivamente affermato con la diffusione mondiale di Facebook e Twitter, sebbene avesse già dato segnali importanti con i siti di media sharing (es. YouTube) e gli innumerevoli blog presenti in Rete.

In questo sottocapitolo, quindi, analizzeremo lo stato dell'arte di queste tipologie di sistemi, indagando anche nel settore dei Mobile Recommender System, vedremo quali sono gli algoritmi utilizzati dai RS per generare i suggerimenti e quali gli approcci per migliorare l'efficienza dei risultati forniti, e, infine, individueremo le questioni ancora aperte.

3.3.1 Classificazione dei Recommender System

I Recommender System possono essere suddivisi secondo più criteri [15][21][22]. In questa analisi ho preso in considerazione tre tipi di classificazioni: per tipo di suggerimenti proposti, per modalità di erogazione dei suggerimenti, e per processo di generazione dei suggerimenti.

Classificazione per tipo di suggerimenti

Adottando come criterio di classificazione di un Recommender System l'origine dei suggerimenti si possono distinguere quattro tipi di RS:

1. *Collaborative Based* (o Social based). Il RS suggerisce all'utente gli item preferiti dalle persone simili a lui
2. *Content Based*. Gli item suggeriti dal sistema sono simili a quelli che l'utente ha giudicato positivamente in passato.
3. *Economic Factor Based*. Il sistema suggerisce gli item che possiedono il miglior rapporto qualità prezzo, in base ai costi oggettivi e alle valutazioni soggettive degli utenti.
4. *Hybrid*. Nei suggerimenti di questo tipo vi è la compresenza di due o più fattori tra quelli sopracitati (preferenze degli utenti simili, item preferiti dall'utente, rapporto qualità prezzo).

Classificazione per modalità di erogazione dei suggerimenti

I suggerimenti possono essere forniti in base a specifiche richieste dell'utente oppure proposti in automatico. In base al modello di information retrieval utilizzato si distinguono tre tipi di Recommender System:

1. *Pull information model*. L'utente deve eseguire una query per ricevere il suggerimento, specificando i parametri necessari alla ricerca.
2. *Push information model*. L'utente riceve i suggerimenti in base alle informazioni contenute nel suo profilo.
3. *Hybrid*. È una combinazione dei due modelli.

Classificazione per processo di generazione dei suggerimenti

In base alle modalità di generazione dei suggerimenti si possono identificare sei diverse tipologie di Recommender System:

1. *User Centered*. Il sistema calcola la somiglianza tra un utente e tutti gli altri attraverso una misura di similarità e seleziona i cosiddetti *vicini*, cioè quegli utenti per i quali il valore della misura di similarità supera una soglia prefissata. Sulla base delle valutazioni dei vicini il RS calcola i rating medi degli item e redige una classifica.
2. *Item Based*. Il Recommender System calcola la somiglianza tra le preferenze specificate nel profilo dell'utente e le caratteristiche contenute nella descrizione degli item. Gli item che ottengono una misura di similarità sopra una certa soglia vengono proposti all'utente, gli altri vengono scartati.
3. *Mining associate rules*. Diversamente dai primi due casi, qui ci si affida ad un metodo statistico. Fissata una soglia m , se un oggetto X viene acquistato da n utenti e di questi n utenti almeno m acquistano anche Y , allora esiste una regola $X \rightarrow Y$. In tal caso il RS suggerisce Y agli utenti che sono interessati a X .
4. *Demography Based*. Il RS classifica l'utente in base ai dati demografici contenuti nel suo profilo. Per una determinata categoria, quindi, identifica le informazioni più rilevanti e su tale base determina i suggerimenti più appropriati. Il sistema, quindi, propone alle persone appartenenti alla stessa categoria i medesimi suggerimenti.
5. *Knowledge Based*. Un Recommender System può generare i suggerimenti sfruttando le sue informazioni implicite sull'utente. I tre modi possibili sono: la deduzione di preferenze a partire da dati relativi a necessità e desideri dell'utente; l'utilizzo delle esperienze passate dell'utente per determinare le sue valutazioni su item per lui nuovi; la derivazione di inferenze attraverso l'applicazione di strumenti di data mining ai dati sull'utente.
6. *Hybrid*. Il RS produce i suggerimenti sfruttando due o più strategie tra quelle sopracitate.

3.3.2 Semantic Recommender System

I suggerimenti di un Recommender System sono di natura software, ma si basano su preferenze di persone. È dunque evidente che ci sia l'esigenza di un passaggio da un linguaggio umano a uno software, con i conseguenti problemi di comunicazione (come l'eterogeneità delle rappresentazioni e le difficoltà nel riutilizzo di dati di processi differenti). I dati relativi alle risorse del Semantic Web sono definiti in modo tale che possano essere interpretati dall'uomo ma anche da un computer. Le linee guida seguite per raggiungere tale obiettivo sono state (e sono tutt'ora) due:

- il tag semantico delle risorse, separando quindi il contenuto di un documento dalla sua struttura;
- lo sviluppo di agenti software in grado di processare le risorse a un livello semantico.

Con tali strumenti a disposizione i Recommender System sono agevolati nel loro compito, consentendo un livello di comprensione più profondo sia degli utenti che degli item.

Un Recommender System è dunque considerato semantico [22] se le sue prestazioni sono basate su una struttura di conoscenza addizionale, definita attraverso un diagramma concettuale o una ontologia. Di seguito viene proposta una rassegna di Semantic Recommender System suddivisi in due categorie: Trust network based systems e Context-adaptable systems.

Trust network based systems

A questo gruppo appartengono i sistemi proposti da Adrian, Sauermann e Roth-Berghofer (2007), Szomszor, Cattuto, Alani, O'Hara, Baldassarri, Loreto e Servedio (2007), Massa e Avesani (2004), Kruk e Decker (2005), e Bedi, Kaur e Marwaha (2007).

Adrian, Sauermaann e Roth-Berghofer [23] sono gli ideatori di ConTag, un sistema che consiglia i tag più appropriati per descrivere una certa risorsa web. I tag sono generati processando la risorsa e traducendola in formato RDF. Una volta normalizzata, vengono estratti gli elementi più significativi, i quali vengono organizzati in uno schema e poi mappati su una ontologia del Semantic Web. I tag così ottenuti vengono infine proposti all'utente come possibili descrittori della risorsa. Uno dei vantaggi di tale approccio è che si minimizzano i problemi causati da sinonimi, omonimie, acronimi e varianti nelle pronunce.

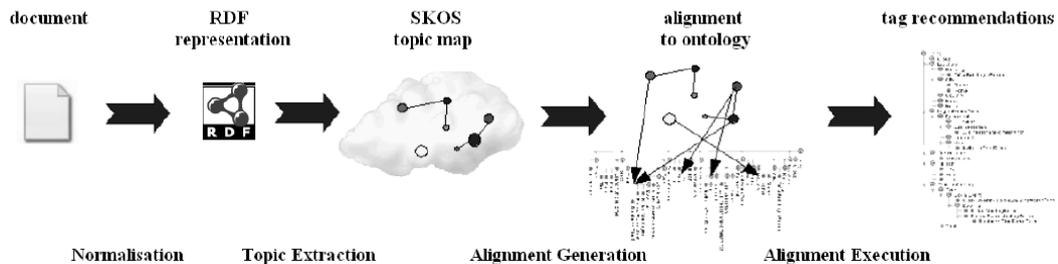


Figura3.11: il processo di generazione di suggerimenti di ConTag [23]

Szomszor e i suoi collaboratori [24] propongono una ontologia per integrare le folksonomie relative ai film con una base di conoscenza semantica che descriva le abitudini degli utenti. La folksonomia viene usata per arricchire le descrizioni dei film con recensioni, classificazioni e valutazioni. Utilizzando i tag raccolti da Internet Movie DataBase (IMDB)⁸ e le informazioni relative ai noleggi di film di Netflix, Szomszor ha effettuato degli esperimenti per capire come costruire profili accurati che riflettano il reale grado di interesse che ha un utente per un film, e quindi poter generare i suggerimenti adeguati per i film che non ha ancora visto. Le risorse sono state rappresentate tramite OWL.

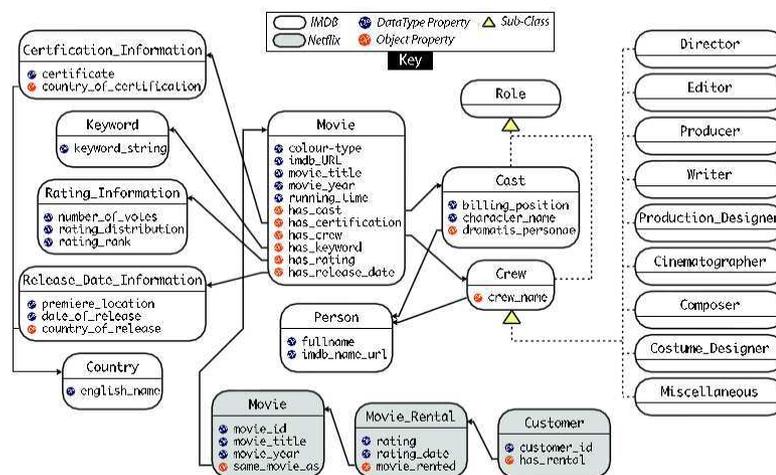


Figura 3.12 : l'ontologia utilizzata per integrare i dati di IMDB e Netflix [24]

Nei sistemi decentralizzati, nei quali chiunque è libero di creare contenuti e non vi è un'entità che vigili sulla qualità, sia dei contenuti stessi, sia degli utenti. Il sistema allora può chiedere a un utente di valutare gli altri utenti, fornendo un giudizio sulla fiducia che ripone in ciascuno di essi. Questo è proprio l'ambito di applicazione del modello proposto da Massa e Avesani [25], nel quale le valutazioni dirette di fiducia vengono propagate tra gli utenti, in modo da formare una struttura reticolare, denominata Web of Trust. I suggerimenti per un utente,

⁸ <http://www.imdb.com/>

quindi, vengono ricavati solo dai giudizi degli utenti che ritiene fidati (direttamente o indirettamente). I vantaggi di questo approccio sono due: isolare i suggerimenti provenienti da utenti non affidabili, aumentando la precisione dei risultati del RS; risolvere il problema del "cold start users", cioè di quegli utenti che hanno fornito poche valutazioni, e per i quali è difficile calcolare la similarità con altri utenti.

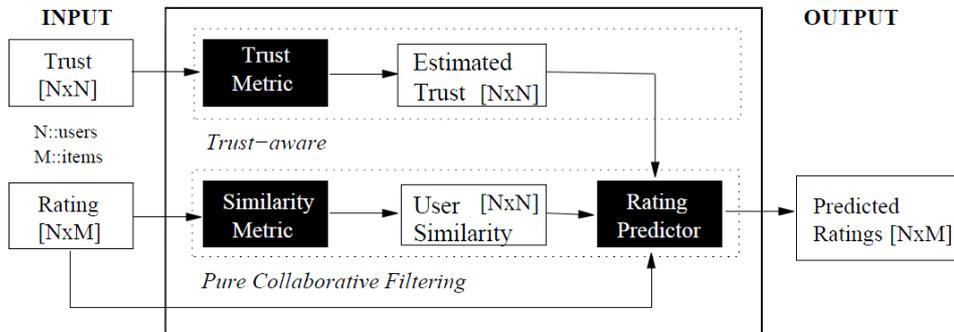


Figura 3.13: l'architettura del modello proposto da Massa e Avesani [25]

Kruk e Decker [26] hanno progettato FOAFRealm, un sistema di gestione dei profili basato sul social networking e metadati definiti secondo il vocabolario del FOAF project. In generale, i risultati delle ricerche online contengono uno o più elementi che non sono attinenti alla richiesta dell'utente. Se poi la ricerca fallisce totalmente, il più delle volte ci si risolve a chiedere ad amici o conoscenti. FOAFRealm automatizza questo processo, cercando le risposte all'interno di un social network, strutturato secondo una rete distribuita peer to peer. Anche i profili degli utenti non sono centralizzati, ma sono definiti dalle interazioni con gli altri utenti. Il modello sviluppato da Kruk e Decker è chiamato *secured semantic social collaborative filtering model*.

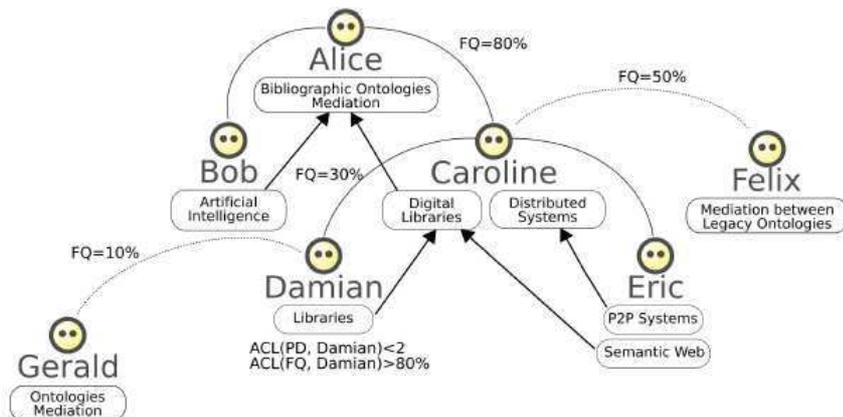


Figura 3.14: lo scenario del secured semantic social collaborative filtering model [26]

Anche Bedi e i suoi collaboratori [27] hanno definito un modello basato su reti di fiducia tra agenti software. Il Recommender System da loro proposto, tuttavia, utilizza ontologie temporanee che cambiano in tempo reale al modificarsi delle condizioni circostanti. Ciò riflette la natura dinamica dei domini in cui trovano applicazione i RS.

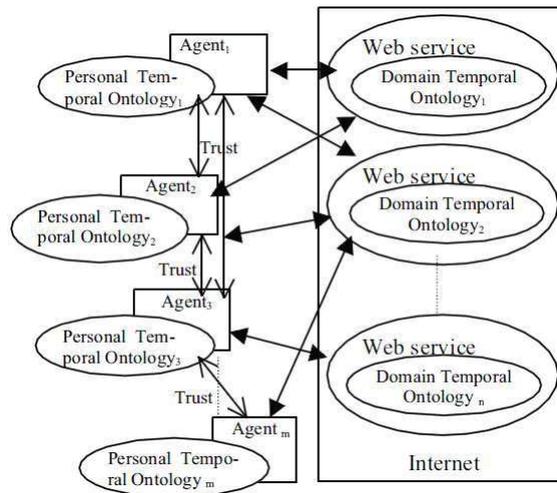


Figura 3.15: le temporal ontology proposte da Bedi et al. [27]

Context-adaptable systems

A questa categoria appartengono i sistemi proposti da Kim e Kwon (2007), Loizou e Dasmahapatra (2006), Yu, Nakamura, Jang, Kajita e Mase (2007), Laliwala, Sorathia e Chaudhary (2006), e Woerndl, Schueller e Wojtech (2007).

Kim e Kwon [28] hanno progettato un Recommender System basato sul concetto di "use context", letteralmente contesto d'uso. L'idea di fondo è che la specificità delle informazioni relative ad un item siano crescenti man mano che l'utente raffina la sua richiesta. Il primo passo del RS è determinare gli item preferiti dall'utente sulla base dei suoi acquisti passati. Successivamente i risultati del Recommender System vengono organizzati in modo gerarchico, suddividendoli per concetti, così che, scendendo i livelli, si ottengano risultati sempre più specifici e focalizzati sull'attuale ricerca dell'utente. Per determinare il contesto il sistema utilizza quattro ontologie: un'ontologia del prodotto (ontology of product), una per definire lo use context, una che annoti l'attività dell'utente nel sistema (historical user activity registry), e una specifica per l'utente (user ontology).

Loizou e Dasmahapatra [29] propongono un sistema basato su una ontologia che include informazioni contestuali riguardanti sia gli item (es. l'utilità che hanno per un utente), sia il processo di generazione dei suggerimenti (es. orario o luogo della richiesta). L'acquisizione di dati è gestita da un processo in background che riceve in input item da fonti affidabili, traduce le descrizioni disponibili nell'ontologia del sistema e li memorizza sotto forma di feature vector [nota]. Il contesto di processo è determinato principalmente dal comportamento recente dell'utente, ma tiene conto anche della sua storia passata, dei trend globali e di restrizioni fissate dall'utente nello specificare una query.

Yu e i suoi collaboratori [30] applicano i Recommender System all'e-learning. Il sistema acquisisce i dati riguardanti l'allievo (user context), le informazioni relative ai contenuti che studia e, più in generale, le peculiarità del dominio preso in considerazione. Le ontologie utilizzate per modellare e rappresentare tali conoscenze sono tre, una per ciascun tipo: Learner Ontology, Learning Content Ontology, e Domain Ontology. L'algoritmo di generazione dei suggerimenti si articola in quattro passaggi: calcolo della rilevanza semantica di un item e selezione dei più importanti, raffinamento dei risultati, generazione di un percorso di apprendimento ad hoc per lo studente, arricchimento del suggerimento con risorse aggiuntive.

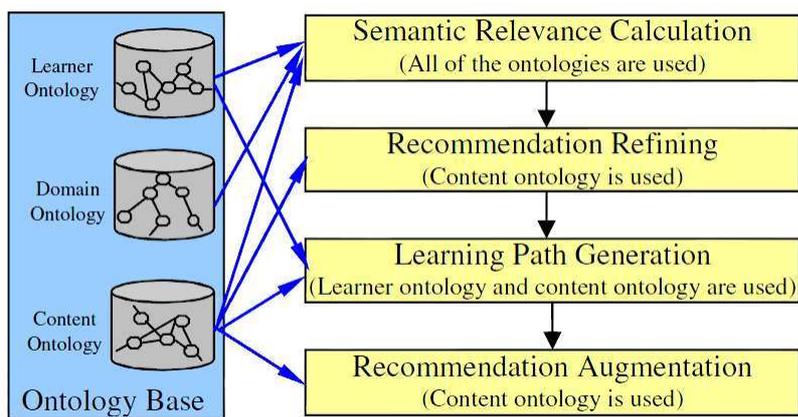


Figura 3.16: la procedura di generazione dei suggerimenti di Yu e i suoi collaboratori [30]

Laliwala, Sorathia e Chaudhary [31] propongono un Recommender System per il dominio degli Agriculture Information Services (AIS). Poichè l'ambito di applicazione è quello dei servizi, hanno scelto di utilizzare una architettura SOA come strategia implementativa. L'approccio adottato per erogare i suggerimenti è di tipo event-driven⁹. In questo caso il contesto è determinato dall'utente stesso, infatti è lui a decidere quali eventi sottoscrivere, in base ai suoi interessi. La principale novità introdotta da Laliwala e i suoi collaboratori nel loro lavoro risiede nell'utilizzo di una semantica all'interno di una SOA per definire il vocabolario comune, le conoscenze e le regole di interazione.

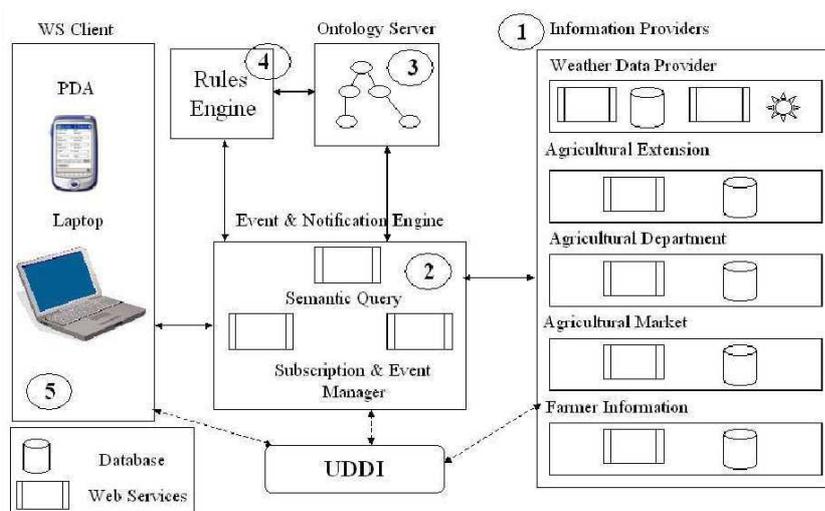


Figura 3.17: architettura del RS proposto da Laliwala, Sorathia, Chaudhary [31]

Woerndl, Schueller e Wojtech [32] si pongono come obiettivo l'integrazione di elementi contestuali nei RS appartenenti al dominio delle applicazioni mobile. Secondo l'approccio proposto, il sistema suggerisce all'utente le applicazioni in base alle scelte passate, tenendo conto anche della similarità o meno del contesto in cui sono state effettuate. L'idea è di descrivere il contesto come un vettore di attributi, analogo a quello che abbiamo visto precedentemente per utenti e item. La misura di similarità del contesto, tuttavia, varia a seconda del dominio specifico. Ad esempio, nell'ambito di applicazioni per il turismo, si può definire la similarità del contesto in relazione alle coordinate GPS: tutte le scelte effettuate

⁹ Di solito con il termine event-driven ci si riferisce a un sistema asincrono, in cui l'accadimento di un evento viene segnalato alle entità interessate.

dagli utenti trovandosi in una posizione geografica entro un certo raggio dal punto attuale sono considerate appartenenti a contesti simili.

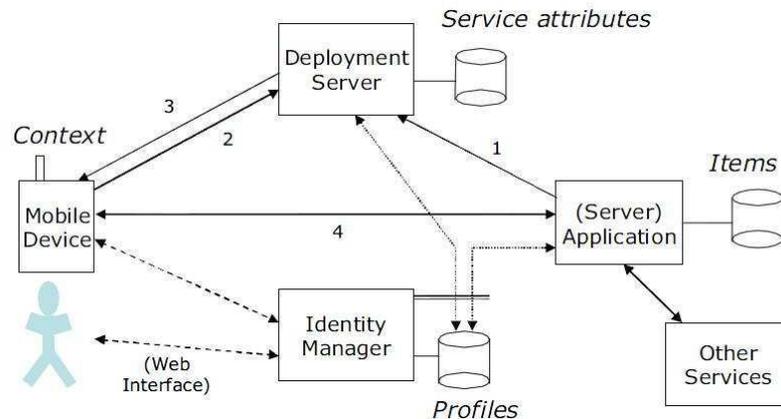


Figura 3.18 : architettura del sistema context-aware di Woerndl, Schueller e Wojtech [32]

Conclusioni

I benefici derivanti dall'utilizzo di Semantic Recommender System sono numerosi. Schematizzando quanto detto finora, i principali vantaggi si possono riassumere nel seguente elenco:

- Garantiscono l'interoperabilità dei sistemi e l'omogeneità della rappresentazione delle informazioni;
- Permettono la contestualizzazione delle preferenze degli utenti, a seconda dei domini specifici;
- Migliorano la comunicazione tra agenti software e utenti;
- Limitano il problema del cold start, completando le informazioni mancanti con inferenze;
- Estendono le descrizioni degli utenti con elementi semantici;
- Migliorano la descrizione e la rappresentazione degli elementi del sistema;
- Rafforzano la logica del sistema includendo un insieme di regole.

3.3.3 Social Recommender System

I Recommender System tradizionali sfruttano solo le informazioni esplicite relative a utenti e item per generare i suggerimenti, tralasciando fattori impliciti che, se considerati, permetterebbero una maggiore comprensione dei processi decisionali dei clienti. Inoltre, i sistemi tradizionali hanno numerose limitazioni, tra i quali il problema del cold start, la *sparsity* della matrice utenti-item, la fiducia nei suggerimenti ricevuti.

Le opportunità del Social Web

Il Web 2.0 è una preziosa e sconfinata fonte di tassonomie generate dagli utenti, le cosiddette folksonomie. Dalle classificazioni prodotte da un certo utente, ad esempio, si possono ricavare informazioni sulle sue preferenze.

Analogamente, la crescente popolarità dei social network, permette di accedere a informazioni personali degli utenti, non solo preferenze ed orientamenti, ma anche dati più privati, come le amicizie, tramite le quali è possibile definire la topologia della propria rete sociale, e più in generale la struttura della rete sociale a livello mondiale.



Figura 3.19: la mappa delle amicizie su Facebook [33]

Le soluzioni offerte

Un Social Recommender System sfrutta le opportunità messe a disposizione dal web sociale. Da una parte si apre la possibilità di utilizzare i dati prodotti dagli utenti di social network, blog e siti di media sharing, per migliorare le descrizioni degli item e delle preferenze degli utenti stessi. Dall'altra, il Social Web consente di associare ad un utente gli item preferiti dei suoi amici virtuali, conferendo attendibilità al suggerimento.

Mentre il primo aspetto verrà trattato con attenzione nel capitolo 4 (sezione 4.3.3 *Come si forma la Collective Intelligence*), qui ci soffermiamo sulla funzione di *Trust* del web sociale.

Suggerimenti fidati

Alcune proposte relative a questo concetto di fiducia sono state già presentate nella sezione *trust network based systems* (vedi 3.3.2 *Semantic Recommender System*). Di seguito, quindi, prendiamo in considerazione proposte che si rivolgono specificatamente ai Social Network.

Innanzitutto è necessario definire, in questo contesto, chi sono gli amici virtuali: sono persone che l'utente conosce nel mondo reale o, meno frequentemente, che ha incontrato in Internet, con le quali è stato instaurato un rapporto di amicizia all'interno di un Social Network. Nei loro confronti l'utente ripone un certo grado di fiducia, che può essere diverso per ciascuno di essi.

In [34] Oufaida e Nouali propongono di far valutare all'utente il livello di fiducia che si ripone in una certa amicizia virtuale, assegnando un numero, su una scala da 1 a 5 (1 = minima, 5 = massima). Basandosi su questi valori vengono poi individuati i cosiddetti "social neighborhood", cioè la propria lista di utenti considerati affidabili. Infine, ottenuta questa classificazione degli amici, si può utilizzare il valore ad essi associati per attribuire un peso ai suggerimenti o alle valutazioni provenienti da ciascun amico virtuale. Il sistema integra elementi del Social Web con elementi del Semantic Web e dei RS tradizionali (vedi figura 3.20).

Come abbiamo visto nell'analisi comparativa del sottocapitolo 3.2, un Recommender System, invece di sfruttare i dati provenienti dal Social Web dall'esterno, può essere integrato all'interno di un social network, ed è proprio ciò che accade in Last.fm e Flixster. In [35] Aranda e i suoi collaboratori esplorano proprio questa possibilità, evidenziando come sia più semplice ottenere informazioni personali sugli utenti integrando il RS in un social network, garantendo così suggerimenti affidabili e maggiormente accurati. La bontà della proposta è

stata verificata tramite un test sul database del sito BoardGameGeek¹⁰, e i risultati ottenuti hanno confermato il miglioramento qualitativo delle prestazioni.

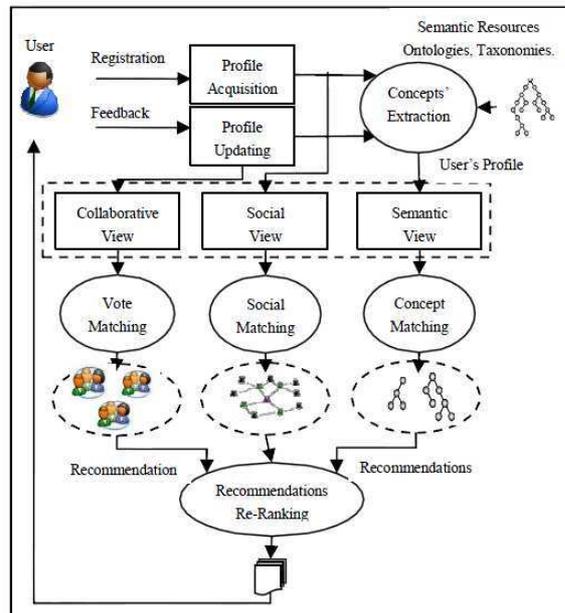


Figura 3.20: architettura del Social Recommender System di Oufaida e Nouali [34]

3.3.4 Mobile recommender system

La proliferazione di dispositivi mobili con connettività wireless e prestazioni sempre più elevate [36], insieme agli sviluppi delle tecnologie di comunicazione, come le emergenti Mobile Ad-hoc NETWORKS (MANET)¹¹, testimoniano la crescente importanza che avranno i sistemi pervasivi nel futuro prossimo. Con il profilarsi di un simile scenario è evidente l'opportunità offerta dal settore dei RS "mobili", cioè di sistemi di raccomandazioni che possano essere accessibili tramite dispositivi mobili come tablet o smartphone, e quindi disponibili sempre e ovunque.

Nell'articolo Mobile Recommender System Analysis & Design [37] Gamal Hussein descrive l'implementazione di un Recommender System per facilitare gli utenti nella ricerca proattiva di prodotti all'interno di un negozio. Utilizzando un'applicazione installata sul proprio dispositivo mobile e un'architettura client-server, basata su connessione wireless, l'utente può interagire con il Recommender System del negozio. Il paper contiene una descrizione dell'architettura del sistema e i servizi che dovrebbero essere offerti ipotizzando un possibile scenario.

¹⁰ <http://boardgamegeek.com/>

¹¹ Una MANET è un sistema autonomo di router mobili e dei loro host associati, connessi con collegamenti di tipo wireless che sono uniti formando un grafo di forma arbitraria. Tali router sono liberi di muoversi casualmente e di auto organizzarsi arbitrariamente, sebbene la topologia wireless vari rapidamente ed in modo imprevedibile. Tale rete può operare da sola oppure essere connessa alla rete Internet.

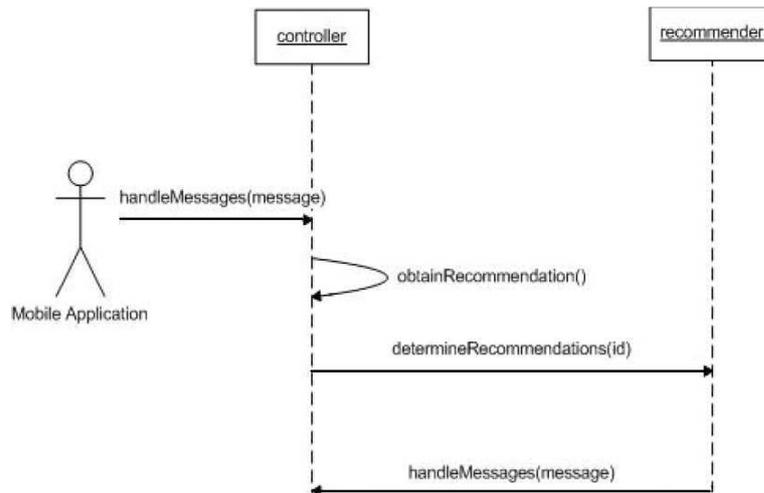


Figura 3.21: architettura del Mobile RS proposto da G. Hussein [37]

Francesco Ricci in [38] illustra il corrente stato dell'arte dei Mobile RS, con particolare attenzione ai servizi che potrebbero essere utili per i viaggiatori. I contributi principali esposti nel paper sono tre: l'analisi degli elementi fondamentali per la progettazione delle interfacce e delle interazioni; la descrizione delle tipologie di suggerimenti utili per i turisti (mappe e informazioni sui luoghi); la proposta di tre possibili architetture applicabili nel contesto dei dispositivi mobili.

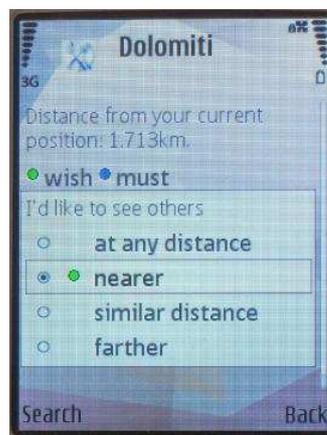


Figura 3.22: Interfaccia grafica del Mobile RS MobyRek citato da F. Ricci [38]

Rimanendo in tema di Recommender System per i turisti, M. Modsching et al. in [39] espongono i risultati di un loro esperimento: hanno considerato un campione di 421 turisti in Gorlitz (Germania) e li hanno suddivisi in tre gruppi. Al primo gruppo (*Planner*) è stato fornito un RS proattivo, che forniva percorsi personalizzati; al secondo (*Explorer*) hanno consegnato un pull RS, che consentiva di ricevere informazioni context based su richiesta; al terzo gruppo di turisti (*Logger*), infine, è stato dato un tradizionale navigatore GPS. Il risultato è stato che i *Planner* e gli *Explorer* hanno espresso un alto gradimento per i suggerimenti ricevuti, paragonando l'esperienza a un tour guidato. Confrontando, inoltre, i due gruppi dotati di RS con quello dotato di navigatore GPS, si è rilevato che i primi hanno impiegato meno della metà del tempo del secondo pur vedere il quadruplo delle attrazioni.

Criteria	Planner	Explorer	Logger
Distance	3.4 km	4.4 km	6.6 km
Duration	1.3 h	1.7 h	4 h
Visited sights	14	16	4
Visit time (mode)	4 min	4 min	2 min

Figura 3.23 : risultati dell'esperimento di M. Modsching et al. [39]

3.3.5 Algoritmi e metodi per i Recommender System tradizionali

In questa sezione vengono presentati alcuni approcci presenti in letteratura per determinare la similarità tra item e la similarità tra user, selezionare i vicini, e generare i suggerimenti

Calcolo della similarità tra item

Nell'approccio Content Based, è fondamentale trovare una misura di similarità affidabile che permetta di confrontare gli item. In generale, si assume di avere una matrice R dove ad ogni colonna corrisponde un oggetto e ad ogni riga un utente. Consideriamo due item, t_i e t_k , l'obiettivo è valutare la loro somiglianza. A tale scopo si possono utilizzare tre metriche: la correlazione di Pearson, oppure il Coseno [40].

Correlazione di Pearson: supponiamo che $\{u_1, \dots, u_L\}$ siano il sottoinsieme di utenti che hanno valutato sia l'item t_i che t_k . Indichiamo con r_{ij} l'elemento di riga i e colonna j della matrice R, cioè la valutazione che l'utente u_i ha dato per l'item t_j , e con \bar{r}_i la valutazione media che gli utenti $\{u_1, \dots, u_L\}$ hanno dato dell'item t_i . Allora definiamo la similarità di t_i e t_k come:

$$sim_{i,k} = corr_{i,k} = \frac{\sum_{j=1}^L (r_{ji} - \bar{r}_i) \cdot (r_{jk} - \bar{r}_k)}{\sqrt{\sum_{j=1}^L (r_{ji} - \bar{r}_i)^2 \cdot \sum_{j=1}^L (r_{jk} - \bar{r}_k)^2}}$$

Coseno: sia R la matrice $m \times n$ utenti-item. Allora la similarità tra due item i e j , è definita come il coseno dell'angolo formato dai vettori a n dimensioni corrispondenti alla i -esima e alla j -esima colonna della matrice R.

$$sim_{i,k} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \times \|\vec{j}\|}$$

Per capire meglio con un esempio semplificato, sia $i = A = (x_1, y_1)$ e $j = B = (x_2, y_2)$, allora:

$$\cos(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \times \|\vec{B}\|} = \frac{x_1 \cdot x_2 + y_1 \cdot y_2}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2}}$$

Calcolo della similarità tra user

Nell'approccio Collaborative Filtering, un elemento essenziale è dato dalla generazione dei vicini (neighbor) di un utente. In generale, si assume di avere una matrice R dove ad ogni

colonna corrisponde un oggetto e ad ogni riga un utente. Consideriamo due utenti, u_i e u_k , l'obiettivo è determinare quanto si assomigliano. A tale scopo si possono utilizzare due metriche: la correlazione di Pearson, oppure il Coseno [40].

Correlazione di Pearson: supponiamo che $\{i_1, \dots, i_L\}$ siano il sottoinsieme di item valutati sia da u_i che da u_k . Indichiamo con r_{ij} l'elemento di riga i e colonna j della matrice R , cioè la valutazione che l'utente u_i ha dato per l'item j , e con \bar{r}_i la valutazione media che i ha fatto sugli item $\{i_1, \dots, i_L\}$. Allora definiamo la similarità di u_i e u_k come:

$$sim_{i,k} = corr_{i,k} = \frac{\sum_{j=1}^L (r_{ij} - \bar{r}_i) \cdot (r_{kj} - \bar{r}_k)}{\sqrt{\sum_{j=1}^L (r_{ij} - \bar{r}_i)^2 \cdot \sum_{j=1}^L (r_{kj} - \bar{r}_k)^2}}$$

Coseno: se consideriamo un utente come un vettore nello spazio N -dimensionale (N è il numero di item totali), dove ogni elemento x_t del vettore corrisponde alla valutazione che l'utente ha fornito per l'item t , allora si può calcolare la similarità tra utenti come l'angolo compreso tra i due vettori che li rappresentano. Si noti che per gli item non valutati, il corrispondente elemento nel vettore vale 0. Indichiamo con r_{ij} l'elemento j -esimo del vettore associato all'utente u_i e con \bar{r}_i il valore medio degli elementi del vettore. Definiamo la similarità di u_i e u_k come:

$$sim_{i,k} = cos_{i,k} = \sum_{j=1}^N \frac{r_{ij}}{\sqrt{\sum_{j=1}^N r_{ij}^2}} \cdot \frac{r_{kj}}{\sqrt{\sum_{j=1}^N r_{kj}^2}}$$

Selezione dei vicini

Calcolando la similarità tra tutti gli utenti, indipendentemente dal metodo specifico, si ottiene una matrice simmetrica S di dimensione $M \times M$ (dove M è il numero di utenti nel sistema). Il generico elemento s_{ik} di tale matrice corrisponde alla similarità tra l'utente u_i e l'utente u_k . Ovviamente $s_{ik} = s_{ki}$. Inoltre, sulla diagonale c'è la misura di similarità di ogni utente con se stesso e non è utile tenerla in considerazione.

Consideriamo, ora, l'utente per cui vogliamo generare un insieme di suggerimenti, e lo chiamiamo *utente attivo* (u_a). Dobbiamo quindi selezionare un insieme di user che costituiscano il suo "vicinato" (*neighborhood*), cioè il gruppo di utenti a lui simili. Per tale scopo ci sono varie possibilità, e qui ne presentiamo due: center-based e per aggregazione.

Center-Based: è la soluzione più semplice. Sulla matrice S si considera la riga corrispondente all'utente u_a e si scelgono gli L valori più alti. Gli utenti corrispondenti a tali valori formano la neighborhood dell'utente attivo.

Per aggregazione: richiede un procedimento più complesso. Al primo passo si prende l'utente che ha valore di similarità più alto in relazione all'utente attivo u_a . Questo utente, insieme a u_a , costituiscono la neighborhood corrente N_C . In totale i vicini di u_a da individuare sono L . Gli $L - 1$ utenti successivi vengono selezionati applicando un algoritmo iterativo.

L' algoritmo iterativo

Supponiamo che al h-esimo passo di questo algoritmo abbiamo una neighborhood corrente formata da h utenti, più l'utente attivo. Viene calcolato un *centroide* di questi h+1 utenti secondo la formula:

$$\vec{C} = \frac{1}{h+1} \sum_{j=1}^{h+1} u_j$$

Si noti che il centroide è un vettore di N componenti come i vettori utente, e quindi confrontabile con essi. La somma di due vettori utente u_j e u_k viene fatta componente a componente.

Sia $Z = U - N_C$ l'insieme degli utenti che non fanno parte della neighborhood corrente. Tra gli elementi di Z, viene selezionato quello più simile al centroide, e viene aggiunto a N_C (ed eliminato da Z).

A questo punto la neighborhood corrente è formato da h+1 elementi, oltre all'utente attivo. Se h+2 = L l'algoritmo è terminato, altrimenti si passa all'iterazione successiva.

Generazione dei suggerimenti

Il *recommendation problem* richiede di associare ad un item i un valore r che l'utente non ha ancora espresso esplicitamente. A tale scopo si possono adottare più tecniche [40][41][42]: neighborhood based, fattorizzazione della matrice user-item, somma pesata, media pesata, association rules based.

Neighborhood based: per stimare $R^*(u,i)$, cioè il rating che l'utente u dovrebbe assegnare all'item i bisogna innanzitutto calcolare la similarità tra l'utente u e gli altri utenti u' , ad esempio utilizzando il coseno:

$$sim(u, u') = \frac{\sum_{i \in I(u, u')} R(u, i) \cdot R(u', i)}{\sqrt{\sum_{i \in I(u, u')} R(u, i)^2} \cdot \sqrt{\sum_{i \in I(u, u')} R(u', i)^2}}$$

dove $I(u, u')$ indica l'insieme di tutti gli item valutati sia dall'utente u che da u' , e $R(u, i)$ indica il rating effettivo dell'utente u riguardo all'item i .

Basandosi sulle similarità tra utenti calcolate, si determina l'insieme $N(u)$ dei *neighbor* di u più vicini. La taglia dell'insieme $N(u)$ può essere compresa tra 1 e $|U| - 1$. Il valore di $R^*(u, i)$ è calcolato come la somma pesata di tutti i rating conosciuti $R(u', i)$ degli utenti $u' \in N(u)$ secondo la seguente formula.

$$R^*(u, i) = \bar{R}(u) + \frac{\sum_{u' \in N(u)} sim(u, u') \cdot (R(u', i) - \bar{R}(u'))}{\sum_{u' \in N(u)} |sim(u, u')|}$$

Fattorizzazione della matrice user-item: Si parte dall'assunzione che il rating di un utente per un certo item è composto dalla somma dei voti riguardo le varie feature di quell'item. Usando K feature, l'utente u è associato a un vettore p_u (dove gli elementi sono le preferenze dell'utente u per le K feature) e l'item i è associato a un vettore q_i (dove gli elementi sono i

pesi dell'importanza di ciascuna feature dell'item i). La previsione di quanto un item i possa piacere all'utente u , indicata con $R^*(u,i)$ è data dal prodotto interno dei due vettori.

$$R^*(u,i) = p_u^T \bullet q_i$$

dove il pallino scuro indica il prodotto interno.

Tutti i valori degli elementi dei vettori utente e dei vettori item sono inizialmente posti a un numero arbitrario. Successivamente vengono aggiornati, variando il valore dei parametri θ e λ . L'aggiornamento viene fatto tramite un algoritmo di apprendimento, con l'obiettivo di minimizzare una certa funzione di merito. Una possibilità è la seguente [41]:

For each rating $R(u,i)$

$$err = R(u,i) - p_u^T q_i$$

$$p_u = p_u + \theta(err \times q_i - \lambda \times p_u)$$

$$q_i = q_i + \theta(err \times p_u - \lambda \times q_i)$$
End For

Infine, utilizzando i valori ottenuti di p_u e q_i , si calcola il prodotto interno dei due vettori ottenendo $R^*(u,i)$.

Somma pesata: per fare una previsione del gradimento di un utente attivo a riguardo un item i , possiamo prendere la media pesata di tutti i rating riguardanti quel determinato item ($r_{u,i}$), in accordo con la seguente formula.

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} |w_{a,u}|}$$

dove \bar{r}_a e \bar{r}_u sono i rating medi, rispettivamente, per l'utente a e l'utente u di tutti gli altri item e $w_{a,u}$ è il peso (similarità) tra l'utente a e l'utente u . La sommatoria è su tutti gli utenti u di U che hanno valutato l'item i .

Media pesata: per le previsioni item-based, si può utilizzare una media pesata per calcolare il rating dell'item i non ancora valutato da u :

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} \cdot w_{i,n}}{\sum_{n \in N} |w_{i,n}|}$$

dove le sommatorie sono su tutti gli oggetti $n \in N$ valutati dall'utente u , $w_{i,n}$ è il peso (la similarità) tra gli oggetti i e n , e $r_{u,n}$ è il rating dell'utente u per l'oggetto n .

Association rules based: supponiamo di avere n item, $I = \{i_1, i_2, \dots, i_n\}$, nella matrice R iniziale user-item. Una transazione $T \subseteq I$ è definita come un insieme di item che sono stati valutati o acquistati insieme. Una regola associativa (*association rule*) tra due insiemi di item, I_X e I_Y ,

tali che $I_X, I_Y \subseteq I$ e $I_X \cap I_Y = \emptyset$, stabilisce che se gli item dell'insieme I_X sono presenti nella transazione T , allora c'è un'alta probabilità che gli item dell'insieme I_Y siano presenti in T . Una simile regola associativa viene indicata con $I_X \Rightarrow I_Y$.

3.3.6 Metriche di valutazione dei Recommender System

Si distinguono due tipi di metriche per valutare un Recommender System: le misure per valutare la qualità e le misure per valutare le performance. Le prime analizzano i suggerimenti, le seconde il RS come sistema.

Qualità

Tra le misure per valutare la qualità dei suggerimenti di un RS possiamo distinguere l'accuratezza [40], e la copertura [42]:

Accuratezza: le tecniche ideate per quantificare l'accuratezza sono molte, e tra queste possiamo annoverare le metriche statistiche e la teoria delle decisioni. Alla prima categoria appartengono la *mean absolute error* (MAE) e la *root mean squared error* (RMSE); della seconda possiamo citare la *ROC sensivity*.

- Mean Absolute Error (MAE): uno dei metri per determinare l'accuratezza (o viceversa, l'errore di classificazione), è il Mean Absolute Error, che calcola la media della differenza assoluta tra i rating previsti dal RS e la valutazione che successivamente l'utente esprime.

$$MAE = \frac{\sum_{\{i,j\}} |p_{i,j} - r_{i,j}|}{n}$$

dove n è il numero totale di rating di tutti gli utenti, $p_{i,j}$ è il rating previsto per l'utente i sull'item j , e $r_{i,j}$ è il rating effettivo. Più basso è il valore del MAE e migliore è la predizione del rating.

A volte è utile normalizzare tale valore rispetto alla scala utilizzata nel RS specifico, in modo da poter poi comparare sistemi che utilizzano rating minimo (r_{min}) e massimo (r_{max}) diversi. In tal caso si calcola la Normalized MAE (NMAE):

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

- Root Mean Squared Error (RMSE): è la metrica prescelta per stimare la qualità dei RS nell'ambito del Netflix Prize (vedi 3.1 *Introduzione ai RS*).

$$RMSE = \sqrt{\frac{1}{n} \sum_{\{i,j\}} (p_{i,j} - r_{i,j})^2}$$

dove n è il numero totale di rating su tutti gli user, $p_{i,j}$ è il rating previsto per l'utente i sull'item j e $r_{i,j}$ è il rating effettivo.

- ROC sensivity: è una misura della potenza di previsione di un RS. Operazionalmente è data dalla *Area Under the ROC Curve* (AUC), letteralmente l'area sotto la curva ROC¹². In realtà il valore di AUC è quello dell'area sotto la curva ROC soltanto se la previsione è un problema con due possibili esiti (problema binario). In generale la ROC sensivity si calcola con la seguente formula:

$$AUC = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 \cdot n_1}$$

dove n_0 e n_1 sono rispettivamente il numero di previsioni sbagliate e di previsioni corrette, e $S_0 = \sum r_i$, dove r_i è il rank dell' i -esima previsione positiva nella lista delle previsioni, ordinata per precisione.

Copertura: La copertura è una misura della percentuale di item per i quali l'algoritmo che genera i suggerimenti può risolvere il *recommandation problem*. In alcuni casi, ad esempio a causa della sparsità della matrice user-item, il sistema non è in grado di produrre una previsione per i rating di molti item. In situazioni simili il valore della copertura certamente è basso.

Assumendo che n_i siano gli item per i quali l'utente u_i ha fornito una valutazione e np_i sia il numero degli item per cui il RS è in grado di generare una previsione del rating, allora la copertura è data dalla formula:

$$Coverage = \frac{\sum_{i=1}^m np_i}{\sum_{i=1}^m n_i}$$

dove m è il numero di righe e colonne della matrice user-item ($m \times m$).

Performance

Le metriche per misurare le performance di un RS che ho considerato sono il tempo di risposta, i requisiti di memoria e la complessità computazionale [42].

Tempo di risposta: è definito come il tempo che intercorre tra la richiesta iniziale di un utente e la risposta del sistema a tale richiesta. Si noti che si suppone di avere a che fare con RS di tipo pull, cioè dove esiste una query esplicita da parte dell'utente.

Requisiti di memoria: di solito i requisiti di memoria sono analizzati secondo due direttrici. Una è lo spazio occupato online, l'altra è lo spazio occupato offline. Gli elementi di cui si tiene conto per calcolare tali valori sono principalmente la memoria necessaria per gli utenti e quella necessaria per gli item.

Complessità computazionale: nell'esecuzione di un RS di solito si possono distinguere due fasi. Alla prima appartengono tutte le operazioni che possono essere eseguite offline, come i Preprocessing, la rappresentazione di utenti e item, il calcolo delle similarità e l'individuazione dei vicini. La seconda, invece, comprende tutte i calcoli che devono essere

¹² Una curva ROC (Receiver Operating Characteristic) è una rappresentazione in due dimensioni delle performance di un classificatore, nel quale il rate dei veri positivi è tracciata sull'asse Y e il rate dei falsi positivi è tracciato sull'asse X.

effettuati necessariamente online, in base alle interazioni con l'utente. Delle due, la fase che incide maggiormente sulla complessità computazionale del RS è la prima.

3.3.7 Le questioni ancora aperte

Asanov elenca in [43] quelli che, secondo lui, sono i principali problemi dei RS, e cioè il *cold start*, la *data sparsity* della matrice user-item, la scalabilità, l'affidabilità e la privacy. A queste Su and Khoshgoftaar [40] aggiungono i sinonimi, il problema della *gray sheep* e lo *shilling attack*. Una questione non considerata né da Asanov né dal team Su-Khoshgoftaar, è la perdita della proprietà transitiva tra vicini, che invece viene descritta da Vozalis e Margaritis in [42].

- *Cold start*. È difficile fornire suggerimenti ai nuovi utenti del sistema, infatti non avendo valutato ancora nessun item le informazioni contenute nei loro profilo sono molto limitate. Tale problema viene identificato con il nome Cold start (partenza a freddo). In alcuni RS si prova a risolvere questa difficoltà con un test iniziale. Lo stesso problema può essere riferito anche agli item appena introdotti nel dataset del sistema, i quali non sono ancora stati valutati dagli utenti .
- *Data Sparsity*. Nei sistemi in cui ci sono molti item simili tra loro è altamente probabile che un utente valuti solo un ristretto sottoinsieme degli item totali. La matrice user-item in tal caso sarà piena di “buchi”, cioè di item per cui non esiste il rating dell'utente, e per tale motivo si definisce *sparsa*. Poiché i RS creano le liste dei vicini sulla base delle valutazioni simili degli item, il loro compito necessariamente si complica quando il sottoinsieme di item valutati da due utenti è molto limitato.
- *Scalabilità*. Con la crescita del numero di utenti e degli item del sistema, il RS necessita di maggiori risorse per processare le informazioni e generare i suggerimenti. Ciò può determinare rallentamenti nei tempi di risposta del sistema. Una possibile soluzione è quella di delegare più calcoli possibili alla fase offline della computazione.
- *L'affidabilità*. L'opinione delle persone con una breve storia nel sistema potrebbe non essere rilevante quanto quella degli utenti che utilizzano il RS da molto tempo. In tal senso potrebbe essere utile diversificare il peso dei rating, assegnando maggior spessore a quelle degli utenti che hanno maggior esperienza nel sistema.
- *La privacy*. La privacy è una delle questioni più spinose, infatti i RS per formulare raccomandazioni pertinenti necessitano dei dati personali degli utenti. Maggiori sono le informazioni raccolte e migliori risultano i suggerimenti. Ciò comporta che i responsabili del RS dispongono di molti dati sensibili, e deve essere loro premura tutelarne la riservatezza.
- *Sinonimi*. Con questo termine ci si riferisce alla tendenza di un numero di item simili che hanno nomi differenti. Molti RS sono incapaci di scoprire questo sottile legame latente e quindi li trattano come item differenti. Per ovviare a tale problema si può ricorrere a un vocabolario in cui registrare quali item sono sinonimi.
- *Il problema della gray sheep*. Questa anomalia si verifica quando le opinioni di un utente non sono né troppo diverse né troppo uguali a quelle di un determinato gruppo di utenti, e quindi la generazione di una neighborhood risulta inutile. Se invece esiste un utente i cui

rating differiscono sensibilmente dalla media si parla di Black sheep, e per questo tipo di utente è pressoché impossibile fornire suggerimenti adeguati con i metodi tradizionali.

- *Lo Shilling Attack*: nei sistemi aperti, nei quali chiunque può valutare gli item, anche più volte, alcuni utenti possono decidere di inserire ripetutamente rating positivi (o negativi) per un determinato item, falsando la valutazione generale. Questo, ad esempio, è il caso di società che vogliono influenzare il RS affinché suggerisca i propri prodotti, e che quindi introducono giudizi positivi per i loro item e negativi per quelli della concorrenza.
- *Loss of Neighbor Transivity*. Si considerino gli utenti u_i e u_k , e supponiamo che abbiano un valore di correlazione alto. Si consideri ora l'utente u_l e si supponga che anche il suo valore di correlazione con u_k sia alto. Con buona probabilità allora anche u_i e u_l sono simili. Tuttavia, se l'insieme degli item comuni a u_i e u_k è molto diverso da quello di u_l e u_k , i tradizionali algoritmi di generazione dei vicini non rileveranno alcuna somiglianza tra u_i e u_l .

3.4 I Service Recommender System

Un Service Recommender System (o Service RS) è un sistema di raccomandazioni che ha come oggetto dei suggerimenti i servizi, intesi nella più ampia accezione possibile: dai servizi di trasporto a quelli di ristorazione, dai servizi per la salute a quelli di formazione, fino a quelli nati più recentemente, come i servizi web e gli eService. Per una più ampia e approfondita trattazione delle tipologie di servizi si rimanda al capitolo 2.

In questo sottocapitolo si propone una caratterizzazione di questa particolare categoria di RS, elencandone le peculiarità e studiando lo stato dell'arte.

3.4.1 Caratteristiche dei Service Recommender System

Quanto detto per i RS in generale rimane valido anche per i Service RS, quindi i tipi di Recommender System presentati (tradizionali, semantici, sociali) e le tecniche di generazione dei suggerimenti viste, possono essere applicate anche al campo dei servizi. Le peculiarità di questi sistemi di raccomandazione non risiedono tanto nella forma, bensì nei contenuti, infatti, ciò che li rende particolari è l'oggetto dei suggerimenti, cioè i servizi. Le differenze che derivano da questo contesto specifico possono essere divise in due categorie: requisiti lato RS e requisiti lato Utente. Al primo gruppo appartengono le variazioni che interessano il Recommender System, al secondo le singolarità riguardanti le interazioni dell'utente.

Requisiti lato RS

Profili utente ad hoc. I servizi sono molto più variegati e personalizzabili dei prodotti, quindi conoscere con precisione le reali preferenze di un utente diventa ancora più importante per generare suggerimenti appropriati. Se per un RS generico può essere sufficiente conoscere la storia degli acquisti di un utente e le valutazioni che ha dato su di essi, per questa tipologia di RS è necessario disporre di profili più completi e funzionali. La soluzione maggiormente adottata è la rappresentazione dei profili tramite ontologie.

Consapevolezza del contesto. Mentre i prodotti hanno un'utilità intrinseca che permane nel tempo (usura permettendo), la natura contingente dei servizi rende la loro utilità dipendente dal contesto. Anche i suggerimenti, quindi, devono tenere conto di questa componente, in modo da consentire ai RS una comprensione maggiore delle dinamiche di acquisto dei servizi e poterle sfruttare nella generazione delle raccomandazioni.

Accessibilità. Potendo ormai navigare in sicurezza in Internet con i propri smartphone e tablet, molti servizi, come le prenotazioni di voli aerei, diventano accessibili in qualsiasi luogo e in qualsiasi momento. Questo si traduce nella necessità di raccomandazioni che raggiungano l'utente anche sul suo dispositivo mobile, e quindi i Mobile RS devono essere considerati imprescindibili nella progettazione di un Service RS.

Requisiti lato Utente

Suggerimenti fidati e affidabili. Un servizio, diversamente da un prodotto, non può essere provato prima dell'acquisto, quindi assume maggiore importanza l'affidabilità dei suggerimenti e la completezza delle informazioni disponibili sui servizi. Il RS, dunque, deve fornire descrizioni accurate, ma anche rating e recensioni, preoccupandosi di garantirne la veridicità.

Conoscenza del Service Provider. Come visto nel capitolo 2, tra le caratteristiche peculiari di un servizio c'è l'importanza della reputazione del Service Provider. Un Service RS, quindi deve garantire la possibilità all'utente di verificare quale sia la reputazione dell'entità erogante il servizio che gli è stato suggerito, prevedendo tra gli elementi della sua architettura anche una repository con i profili dei Service Provider.

3.4.2 Stato dell'arte

Costa et al., nel loro lavoro [44] esplorano la possibilità di una sinergia tra RS e context-aware computing, proponendo una sistema (COReS) che integra in un Recommender System per i servizi le funzionalità di una piattaforma context aware, consentendo di riconoscere le esigenze dell'utente in base al contesto. I profili utente e i dati sul dominio di appartenenza sono rappresentati tramite ontologie.

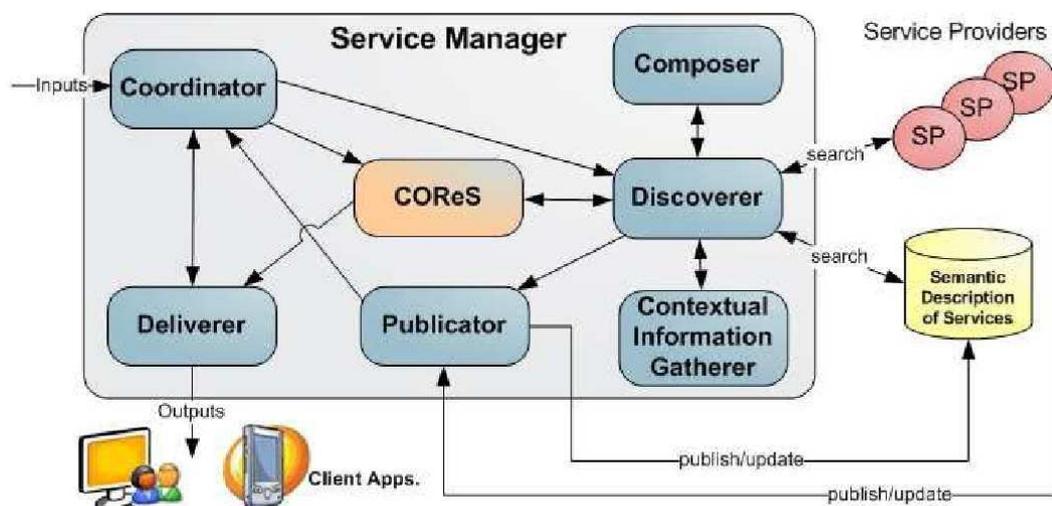


Figura 3.24: l'architettura di COReS [44]

Shahvalady et al. [45] propongono un Recommender System per i Web Service che tenga conto della QoS come discriminante di selezione. Definiscono quindi un insieme di attributi non funzionali che possono avere un impatto significativo sulla qualità del servizio offerto dal Web Service. Inoltre, per determinare la QoS complessiva specificano, per ciascun attributo, una misura appropriata. Analogamente alla proposta di Costa, i profili utente sono rappresentati con una ontologia.

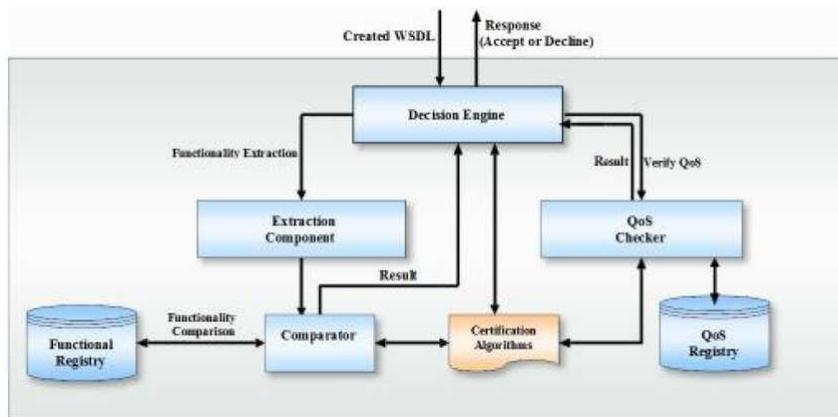


Figura 3.25: architettura del RS proposto da Shahvalady [45]

Kuo et al. [46] descrivono un sistema di raccomandazioni per i Location Based Service (LBS)¹³. Gli autori distinguono le preferenze in due gruppi: quelle a lungo termine e quelle a breve termine. La loro proposta è di diversificare l'effetto che le interazioni degli utenti con il RS producono sulla definizione delle loro preferenze. Le scelte specificate da un utente durante la selezione di un servizio influenzano solo le preferenze a breve termine, mentre quelle a lungo termine dipendono esclusivamente dalla storia dell'utente nel sistema.

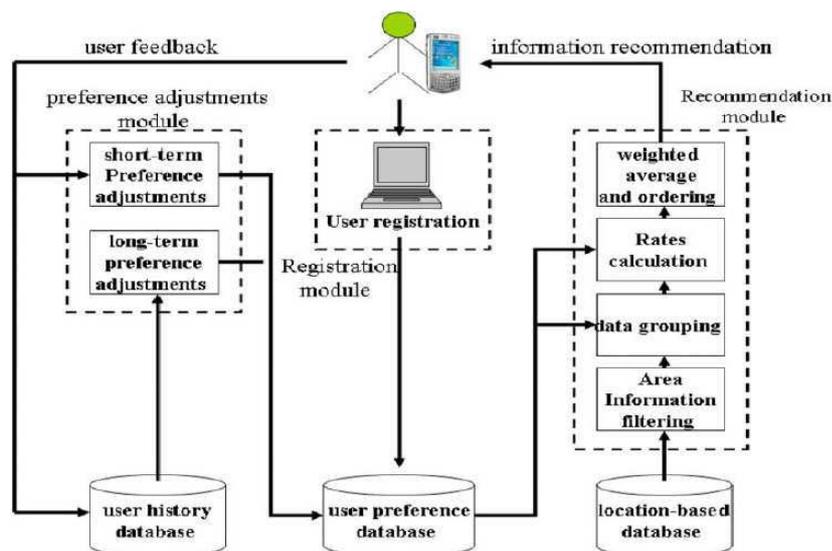


Figura3.26: schema del sistema proposto da Kuo et al. [46]

Blake e Nowlan [47] propongono, per il dominio dei Web Service, un metodo proattivo di somministrazione dei suggerimenti, generandoli in base alle abitudini dell'utente. Il punto di partenza del loro lavoro è stato un'analisi dei messaggi scambiati tra Web Service. Estrahendo i pattern significativi trovati in questi messaggi, e rielaborandoli con metodi sintattici, sono pervenuti a uno schema generale che consenta di aggregare servizi sulla base dei messaggi che utilizzano. I suggerimenti vengono quindi generati confrontando i pattern ricavabili da documenti, pagine web e chat dell'utente con i pattern associati ai Web Service, e selezionando quelli più consoni.

¹³ Rientrano in questa categoria i servizi accessibili tramite dispositivi mobili, in grado di connettersi alle mobile network e con la possibilità di rilevare la posizione geografica dell'utente.

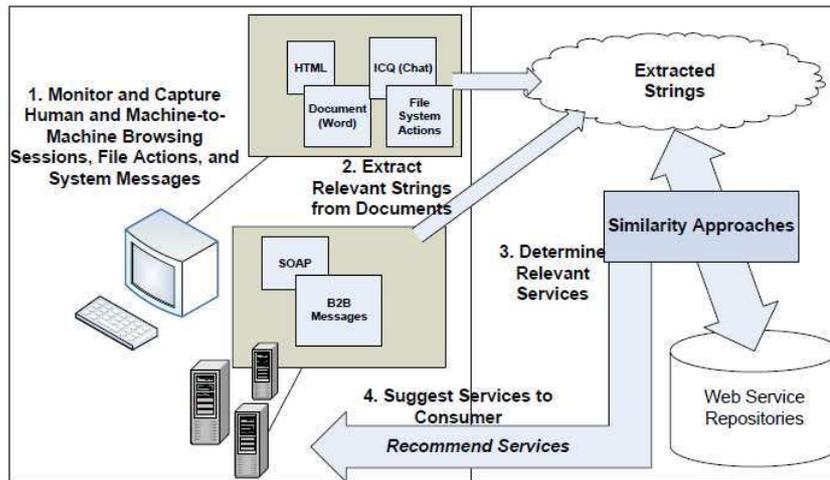


Figura 3.27: metodo di generazione di suggerimenti proposto da Blake e Nowlan [47]

Sellami et al.[48] illustrano una metodologia per la ricerca di Web Service in registri distribuiti. Tale metodologia è basata su tre elementi: la suddivisione dei Web Service in gruppi, una query di ricerca, e il profilo dell'utente. Nella prima fase si seleziona un gruppo di registri confrontando il profilo con le caratteristiche del gruppo. Questo permette una scrematura significativa delle possibilità, velocizzando di molto il processo complessivo. Successivamente, l'utente specifica una query indicando esplicitamente i parametri della sua ricerca. Il RS, infine, propone i servizi che soddisfano maggiormente i criteri indicati.

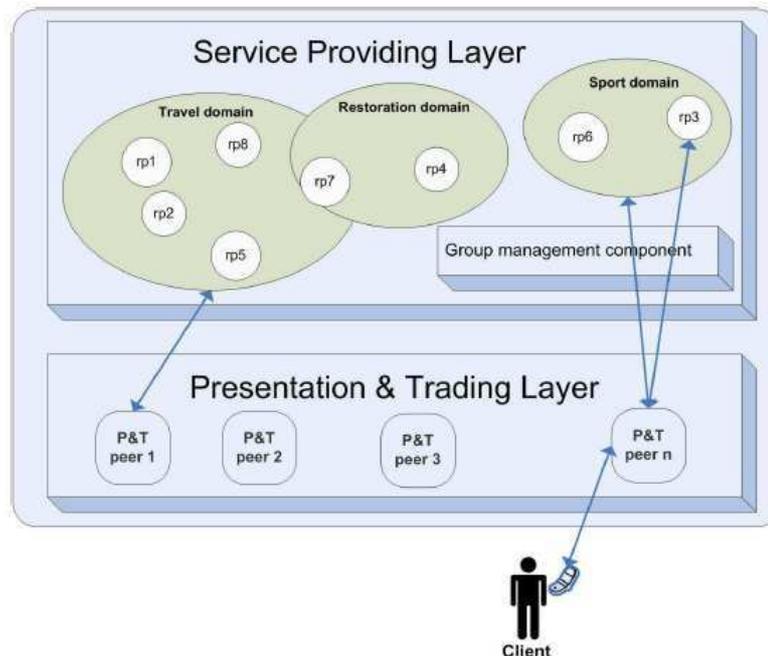


Figura 3.28: metodologia di selezione dei servizi proposta da Sellami et al. [48]

Han et al [49] propongono un Recommender System per la selezione dei servizi nell'ambito del cloud computing. L'utente specifica la richiesta del servizio attraverso un portale web, che si pone da interfaccia tra l'utente e il sistema. Il RS seleziona il servizio cloud con gli attributi di QoS e rank migliori, ottenuti da un Resource Register integrando le informazioni sui Service Provider e sui singoli servizi. Tali dati provengono da supervisori virtuali e dai feedback degli utenti. Inoltre, tra le discriminanti considerate dal sistema, c'è anche il rapporto qualità prezzo.

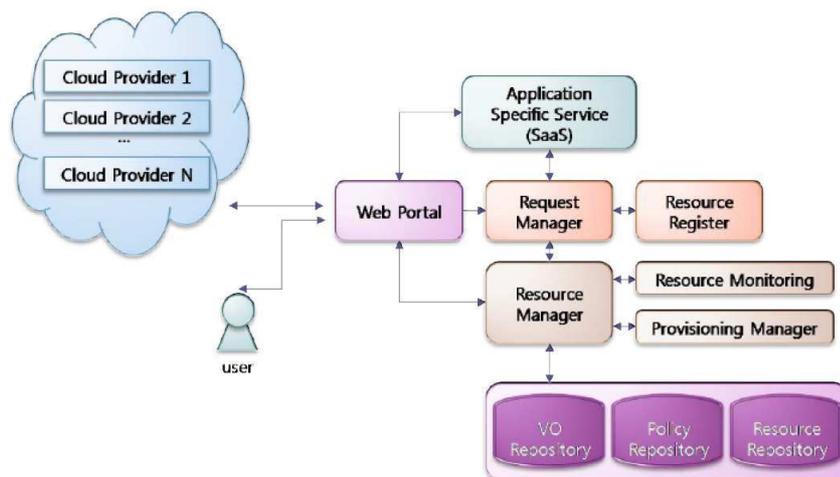


Figura 3.29: Architettura del RS per i servizi di Cloud Computing ideato da Han et al. [49]

CAPITOLO 4

La Collective Intelligence

In questo capitolo si presenta il tema della Collective Intelligence, un argomento le cui reali potenzialità sono ancora in fase di studio, ma che riscuote già molto successo, come testimoniano numerosi esempi. Nel primo sottocapitolo si spiega cos'è la Collective Intelligence, individuandone le origini; nel sottocapitolo 2 si propone una suddivisione della Collective Intelligence in tre livelli di complessità; nel sottocapitolo 3 si esplorano le sorgenti di Collective Intelligence, le componenti motivazionali che spingono alla sua formazione, e gli strumenti per generarla; nell'ultimo sottocapitolo, infine, si descrivono alcune applicazioni della Collective Intelligence nell'ambito dei servizi.

4.1 Cos'è la Collective Intelligence

Per Collective Intelligence (CI, o intelligenza collettiva) si intende la capacità di una comunità di effettuare un task o risolvere un problema in maniera più efficace ed efficiente di quanto potrebbero fare i singoli individui, attraverso la collaborazione e la condivisione di conoscenze. In questa sezione vedremo quali sono le origini di questo concetto e verrà proposta una distinzione tra tre differenti tipologie di Collective Intelligence.

4.1.1 Le origini della Collective Intelligence

Le prime tracce del concetto di intelligenza collettiva si possono trovare già nel lontano 1911, quando l'entomologo William Morton Wheeler, osservando il comportamento di una colonia di formiche, notò che esse non agivano come singoli individui, bensì erano talmente coordinate da sembrare un unico organismo [50].

Un anno più tardi, nel 1912, l'antropologo Emile Durkheim identificò nella Società l'unica sorgente del pensiero logico. La veridicità o meno di una affermazione molto spesso dipende dal senso comune, e quindi dalle conoscenze delle altre persone. In tal senso l'uomo si comporta come una formica all'interno di una colonia. Tale concetto viene espresso da Durkheim con l'aforisma: "La coscienza collettiva è l'insieme delle credenze e dei sentimenti comuni alla media dei membri di una società. Questo insieme ha una vita propria che non esiste se non attraverso i sentimenti e le credenze presenti nelle coscienze individuali" [51].

Un altro riferimento all'idea di Collective Intelligence si trova nel concetto di Noosfera, introdotto per la prima volta nel 1925 dal filosofo Pierre Teilhard de Chardin nella sua opera "L'Homínisation" [52]. Secondo de Chardin la Noosfera è una sorta di coscienza collettiva degli esseri umani che scaturisce dall'interazione fra le menti umane. Più l'umanità si organizza in forma di reti sociali complesse, più la Noosfera acquisisce consapevolezza. Interessante è la struttura della Noosfera descritta da de Chardin nella sua opera, secondo la quale esistono 3 apparati:

- un apparato ereditario, formato dalla memoria collettiva dell’umanità, trasmessa di generazione in generazione attraverso l’opera di inserimento e di educazione dell’individuo nella società;
- un apparato meccanico, dato dagli strumenti e dalle macchine inventate dall’uomo, che costituiscono il prolungamento biologico delle sue capacità naturali di agire;
- un apparato cerebrale, che non è dato dalla semplice somma delle singole autocoscienze, ma dalla loro reciproca interazione, amplificata dalla sempre più sofisticata rete di mezzi tecnici di comunicazione. Un’interazione simile a quella dei neuroni nell’encefalo umano, con la sostanziale differenza che ogni individuo conserva la propria autonomia, libertà e coscienza.

Nell’ultimo dei tre elementi descritti, viene teorizzata per la prima volta l’idea di un pensiero superiore alla somma delle singole parti, concetto fondamentale nell’odierna visione di intelligenza collettiva.

L’idea di aggregazione di conoscenze si ritrova anche nel saggio del 1937 “World Brain, The Idea of a Permanent World Encyclopedia” dello scrittore inglese Herbert George Wells [53], nel quale delinea la sua visione di una enciclopedia mondiale in grado di aiutare l’intera umanità ad utilizzare nel migliore dei modi le risorse di informazione e conoscenza esistenti. Con questa sua proposta immaginifica, impensabile da realizzare ai suoi tempi, Wells introduce l’idea di un’entità fisica che racchiude il sapere universale, inserendo una discontinuità rispetto alle teorie finora viste, le quali prevedevano soltanto entità astratte.

4.1.2 La Collective Intelligence nell’era di Internet

Il concetto di intelligenza collettiva, come visto, ha iniziato a diffondersi nei primi anni del Novecento, ma la nascita del termine Collective Intelligence è molto più recente, risale infatti al 1996, data di pubblicazione del libro “L’intelligenza collettiva. Per un’antropologia del cyberspazio” del filosofo francese Pierre Levy [54]. Egli sostiene che il fine più alto di Internet è la Collective Intelligence, che egli definisce indirettamente in un’intervista nel seguente modo:

“In primo luogo bisogna riconoscere che l’intelligenza è distribuita dovunque c’è umanità, e che questa intelligenza, distribuita dappertutto, può essere valorizzata al massimo mediante le nuove tecniche, soprattutto mettendola in sinergia. Oggi, se due persone distanti fanno due cose complementari, per il tramite delle nuove tecnologie, possono davvero entrare in comunicazione l’una con l’altra, scambiare il loro sapere, cooperare”.

Nella sua opera [54], inoltre, Levy precisa cosa non è da considerarsi Intelligenza Collettiva, cercando di fugare ogni dubbio riguardo il suo significato:

“[...] L’intelligenza collettiva, ricordiamolo, è un’intelligenza distribuita in ogni luogo, permanentemente valorizzata, coordinata e mobilitata in tempo reale. Al fine di evitare ogni malinteso, [...], preciseremo ciò che essa non è. Soprattutto non va confusa con progetti ‘totalitari’ di subordinazione degli individui a comunità trascendenti e feticizzate. In un formicaio gli individui sono ‘ottusi’, mancano di una visione d’insieme e sono ignari del modo in cui ciò che fanno si connette con gli atti degli altri individui. Ma benché le formiche isolate siano ‘stupide’, la loro interazione produce un comportamento manifesto globalmente intelligente. [...]. Il formicaio rappresenta il contrario dell’intelligenza collettiva, nel senso in cui noi la intendiamo in questo libro.[...]”.

Secondo Levy, quindi, la Collective Intelligence presuppone consapevolezza. Dopo aver spiegato ciò che non è, egli si sofferma, successivamente, a descriverne le peculiarità, dando così un'idea più esplicita di cosa invece è:

“L’intelligenza collettiva ha inizio solo con la cultura e si accresce con essa. [...] L’intelligenza dell’insieme non è più il risultato meccanico di atti ciechi e automatici, perché è il pensiero delle persone a perpetuare, inventare e mettere in movimento quello della società. E tuttavia, il collettivo intelligente cui questo testo si riferisce non s’identifica semplicemente con il livello medio della cultura. In un collettivo intelligente, la comunità si dà esplicitamente come obiettivo la negoziazione costante dell’ordine delle cose, del suo linguaggio, del ruolo di ciascuno, la suddivisione e la definizione degli oggetti, la reinterpretazione della propria memoria. [...] Questo progetto comporta un nuovo umanesimo che include e amplia il ‘conosci te stesso’ in direzione di un ‘impariamo a conoscerci per pensare insieme’ e generalizza il ‘penso dunque sono’ in un ‘noi formiamo un’intelligenza collettiva, dunque esistiamo come comunità significativa’. Si passa dal cogito cartesiano al cogitamus. Lungi dal fondere le intelligenze individuali in una sorta di magma indistinto, l’intelligenza collettiva è un processo di crescita, di differenziazione e di mutuo rilancio della specificità. [...]”

Un altro contributo al concetto di Collective Intelligence proviene dal giornalista americano James Surowiecki nel suo libro “The Wisdom of Crowds”, pubblicato nel 2004 [55]. Egli sostiene che l’aggregazione di informazioni all’interno di un gruppo durante un processo decisionale garantisce risultati migliori di quelli che potrebbe ottenere un qualsiasi membro del gruppo individualmente. Affinché un gruppo possa essere considerato saggio, secondo Surowiecki, sono tuttavia necessari 4 elementi:

- *Diversità di opinione.* Ogni individuo dovrebbe avere delle opinioni personali anche se queste dovessero dimostrarsi un’interpretazione eccentrica di fatti noti.
- *Indipendenza.* Le opinioni di un individuo non devono essere determinate o influenzate da quelle degli altri del gruppo.
- *Decentralizzazione.* Gli individui devono essere in grado di specializzarsi e di disporre di conoscenze locali.
- *Aggregazione.* Deve esistere un meccanismo per trasformare le opinioni personali in una unica decisione collettiva.

Surowiecki, parlando di saggezza della folla, spiega come si possa utilizzare una conoscenza aggregata per prendere decisioni assennate, rafforzando quanto sostenuto con numerosi esempi concreti. Quella espressa da Surowiecki, però, è solo una possibile manifestazione di Collective Intelligence e quindi anche solo un possibile caso di applicazione. Il concetto di intelligenza collettiva, infatti, ha rilevanza in molteplici campi, come la sociologia, l’economia, l’informatica, la comunicazione di massa. Per l’argomento trattato nella tesi, è interessante studiare le sue applicazioni nell’ambito informatico, dove ad interagire non sono più elementi di una medesima categoria (le formiche, nel caso della colonia, le conoscenze, nel World Brain, le persone, nella Wisdom of Crowds), bensì persone, idee e computer. Il

MIT center for collective intelligence¹ esprime il tema della loro ricerca con la seguente domanda:

“How can people and computers be connected so that — collectively — they act more intelligently than any individuals, groups, or computers have ever done before?”

Come si può evincere da alcune proprietà controverse (ad esempio la necessità di consapevolezza o meno), non esiste un'unica definizione di Collective Intelligence. Nella prossima sezione, quindi, vedremo una proposta di classificazione basata su livelli di complessità crescenti. Si assume per noto che l'ambito trattato nel seguito della tesi sarà esclusivamente quello informatico.

Giunti a questo punto è utile sintetizzare le teorie esposte finora in uno schema riassuntivo (vedi tabella 4.1), che riordini le idee emerse.

AUTORE	ANNO	CONTRIBUTO
W. M. Wheeler	1911	Prime teorie sull'esistenza di un'unica entità determinata dall'azione coordinata di più individui (superorganismo).
E. Durkheim	1912	Introduce l'idea di una condivisione di conoscenze: la coscienza collettiva
P. T. de Chardin	1925	Teorizza la possibilità di ottenere una conoscenza superiore grazie all'interazione tra le menti umane amplificata dai mezzi di comunicazione.
H.G. Wells	1937	Idea di creare un'entità fisica che racchiuda il sapere universale in modo organizzato.
P. Levy	1996	Conia il termine Collective Intelligence e descrive quelle che secondo lui sono le caratteristiche fondamentali: <ul style="list-style-type: none"> – consapevolezza e visione d'insieme – obiettivi certi – processo di crescita
J. Surowiecki	2004	Utilizzo di conoscenze personali all'interno di un gruppo in un processo decisionale.
MIT center for CI	2006	Integrazione di uomini, computer e conoscenze

Tabella 4.1: I contributi alla creazione del concetto di Collective Intelligence

4.2 Tre livelli di Collective Intelligence

Il numero di utenti di Internet, ogni anno, aumenta vertiginosamente; basti pensare che nel 2000 erano appena 360 milioni e attualmente sfiorano i 2 miliardi, con una crescita del

¹ <http://cci.mit.edu/>

444.8% nel decennio di riferimento² [57]. Questo trend è favorito anche dalle nuove possibilità di accesso alla rete, che non prevedono più soltanto PC, cellulari e palmari, ma anche oggetti in genere come televisori, lettori mp3, automobili ed elettrodomestici. Un maggior numero di utenti implica un aumento significativo dei dati prodotti dai medesimi, soprattutto dopo la creazione del cosiddetto Social Web, che si esprime sotto forma di blog, social network, wiki e siti web per il media sharing (es. YouTube, Flickr, etc.). Tali informazioni esprimono preferenze, modi di essere, idee, stati d'animo, e conoscenze. Se organizzate e convogliate nel modo corretto si possono considerare anch'esse una forma di Collective Intelligence.

Ma quanti tipi di intelligenza collettiva esistono? In realtà si potrebbero distinguere molti tipi di CI, ma per la classificazione in oggetto sono stati presi in considerazione solo 2 parametri come discriminanti: la consapevolezza dell'individuo che contribuisce alla generazione di conoscenza e la forma in cui si manifesta l'intelligenza collettiva, per un totale di 3 livelli. Pur trattandosi di una distinzione abbastanza grossolana, la si ritiene sufficientemente valida per gli scopi prefissati.

4.2.1 Livello 0: CI consapevole ed esplicita

Il livello di complessità più basso si ha quando la creazione della Collective Intelligence avviene con la consapevolezza dei singoli individui che contribuiscono alla sua composizione, e che hanno una visione d'insieme del progetto. Chi concede la propria collaborazione intellettuale, quindi, sa di farlo e lo fa di proposito. L'utilizzo successivo è immediato, non necessita di modifiche, perché lo scopo è chiaro fin dall'inizio, quindi non si richiedono passaggi intermedi per renderla utile (operativa). Questo livello di CI è già maturo e notevolmente utilizzato.

Esempi di CI di livello 0

Alcuni esempi di utilizzo della Collective Intelligence di livello 0 sono:

- *InnoCentive*³. Consente agli esperti di vari settori di venire a conoscenza di una serie di problemi irrisolti di R&S da parte di grandi aziende che pongono su tale piattaforma le questioni di loro interesse. Le aziende possono così attingere ai talenti di una community globale di ricercatori scientifici senza doverli assumere a tempo pieno, ma corrispondendo un premio prefissato a coloro che hanno generato le idee ritenute valide.
- *Wikipedia*⁴. Enciclopedia multilingue gratuita. Qualsiasi individuo può aggiungere o modificare pagine contribuendo con la sua conoscenza all'espansione dell'enciclopedia.
- *Panda Antivirus*⁵. Con un'iniziativa di Collective Intelligence nata nel 2007, vengono costantemente raccolte e condivise informazioni riguardanti ogni tipo di malware che infetta i computer protetti dall'antivirus Panda, cosicché ogni nuovo virus venga prontamente segnalato a tutti gli altri utenti, assicurando aggiornamenti in real time, che ne migliorano la sicurezza. Si noti che, in questo caso, la conoscenza condivisa è quella dei software.

² Per avere un raffronto il social network Facebook registra quasi 600 milioni di utenti (dati anno 2010) [56]

³ <http://www2.innocentive.com/>

⁴ <http://it.wikipedia.org>

⁵ <http://www.pandasecurity.com/italy/homeusers/solutions/collective-intelligence/>

Caso di studio: InnoCentive [58]

InnoCentive (IC) è una sorta di broker di conoscenza, che mette in contatto aziende di vari settori e oltre 80 000 esperti indipendenti e distribuiti in più di 150 paesi. La Collective Intelligence di IC corrisponde all'insieme delle conoscenze dei singoli individui che mettono a disposizione le loro capacità. InnoCentive, quindi, si pone come un "superorganismo" in grado di risolvere qualsiasi problema di natura principalmente scientifico-tecnologica, ma fungendo in sostanza da semplice interfaccia tra *seekers* e *solvers*.

Le società, alla ricerca di soluzioni, lavorano a stretto contatto con lo staff scientifico di InnoCentive per articolare i loro problemi interni in una forma comprensibile ad esperti del settore, i quali, essendo esterni alla realtà societaria, necessitano di una descrizione articolata e precisa della richiesta. Le due principali forme in cui si concretizzano le cosiddette "challenge" sono la stesura di *Reduction To Practice* (RTP)⁶ oppure la produzione di *paper scientifici*. Tali richieste, previo pagamento di 2000\$, vengono inviate (*postate*) sul sito di IC nella sezione *Challenge Center* (vedi figura 4.1) accompagnate da tag che ne specificano l'ambito di interesse, e dettagli che indicano la data di sottomissione, il termine ultimo per la consegna, il premio in denaro offerto dalla società committente, e il numero di esperti che sono impegnati nella risoluzione.

	Title	Posted	Deadline	Award	Solvers
	Methods or Additives that Lower the Yield Value of a Highly Concentrated Hydrophobic Suspension	1/06/11	3/07/11	\$15,000 USD	81

Figura 4.1: Innocentive Challenge Center [59]

InnoCentive, poi, sottopone via email i problemi a tutti gli iscritti (gli esperti) e li invita a partecipare alla risoluzione. Chi è interessato accetta la proposta e inizia a lavorare individualmente ad una propria soluzione. Quando una idea è pronta viene consegnata a IC, che, senza operare scremature, la inoltra alla società interessata, e solo a questo punto i responsabili interni verificano se sono state rispettate le specifiche date e selezionano una o più proposte. Ovviamente, se nessuna idea corrisponde alle aspettative, il committente può decidere di non assegnare il premio e scartare tutte le soluzioni ricevute.

⁶ Una reduction to practice, letteralmente riduzione alla pratica, corrisponde alla stesura dei dettagli necessari all'implementazione concreta di una qualche teoria, in modo che a partire da tale lavoro si possa direttamente produrre un'invenzione completa e operativa.

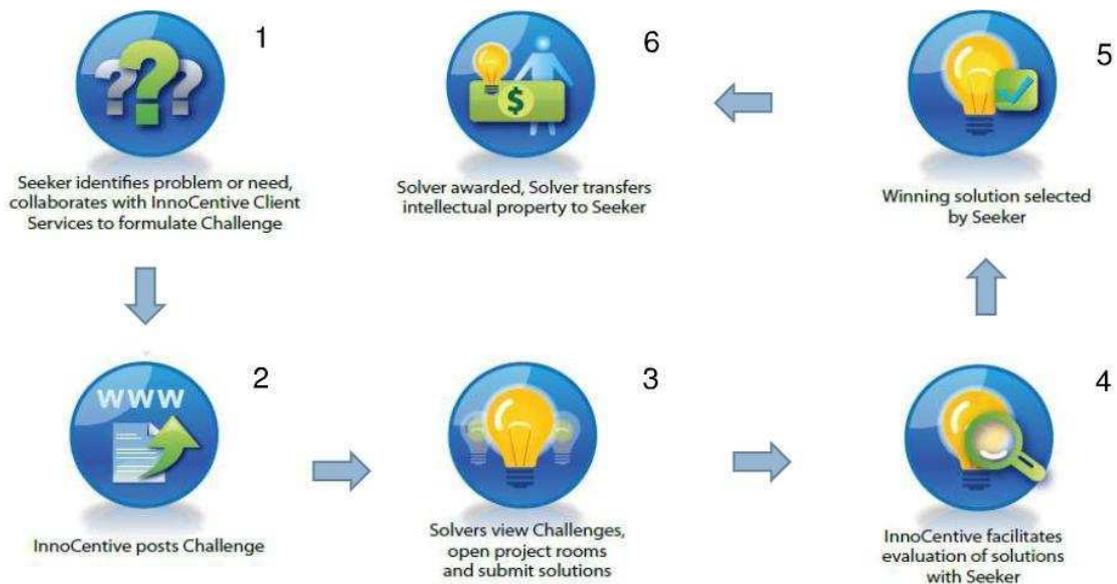


Figura 4.2: Processo di Innocentive [60]

Di particolare interesse è la singolare gestione dei diritti d'autore. Infatti nella maggior parte dei casi la Collective Intelligence non appartiene a nessuno, né a chi la genera, né a chi la sfrutta, qui invece viene prestata molta attenzione alla proprietà intellettuale, sia dei *seekers* che dei *solvers*. Quando un problema viene postato, non sono presenti i dettagli, ma soltanto un abstract che ne delinea il profilo. Chi è interessato a conoscere i particolari deve stabilire un accordo personale con la società mandataria per definire i termini di riservatezza (è vietato diffondere le informazioni ricevute) e i diritti di Intellectual Property (ai *solver* viene fornita una licenza d'uso temporanea). Una volta trovata l'intesa vengono trasmessi i dati riservati. Se successivamente la soluzione sviluppata viene giudicata interessante e quindi selezionata, InnoCentive contatta l'autore e si assicura che questi rilasci ogni diritto sulla sua opera alla compagnia richiedente e che si impegni a non utilizzare i dati riservati ricevuti, anche solo parzialmente, in lavori futuri.

Questo tipo di gestione della proprietà intellettuale, sebbene possa apparire solo una forma di tutela per la società committente, in realtà spersonalizza i contributi individuali, privando di fatto le idee della loro origine. L'insieme di conoscenze anonime che ne deriva ha un comune denominatore, InnoCentive, e dall'esterno si ha la percezione di un sapere unico e monolitico, che viene lecitamente definito Collective Intelligence.

4.2.2 Livello 1: CI consapevole ed implicita

La Collective Intelligence di livello 1 deriva da dati prodotti in modo consapevole, ma con gli attori che hanno una visione parziale della situazione e quindi non conoscono il risultato finale. Il contributo di un individuo, diversamente dal livello 0, si fonde con quelli degli altri individui, e acquisisce valore messo in relazione con essi. In altre parole, non solo l'unione delle conoscenze crea una informazione superiore alla somma dei singoli contributi, ma le conoscenze del singolo non sarebbero sufficienti per risolvere quel determinato quesito o per produrre un'informazione completa, pur essendo disposti ad accettare un'idea di qualità inferiore. Questo livello di CI corrisponde alla Wisdom of Crowds di cui parlava J. Surowiecki.

Esempi di CI di livello 1

Alcuni esempi di utilizzo della Collective Intelligence di livello 1 sono:

- *Crowdscast*⁷. Crowdcast utilizza un approccio basato su CI per effettuare previsioni di mercato, raccogliendo idee di dipendenti e partner delle società che se ne avvalgono. Vengono postate domande su metriche chiave e risultati di eventi rilevanti, come date, numeri (unità, percentuali, valute) ed esiti discreti (sì/no, rosso/verde/giallo, etc.). Chi ne ha le competenze fornisce una risposta in base alle personali conoscenze e tali risposte contribuiscono a determinare un'unica risposta finale che, man mano ci si avvicina all'evento, si allinea sempre più a quella corretta. Le risposte esatte vengono ricompensate con denaro, e maggiore è l'anticipo con cui viene fatta la previsione, maggiore è la remunerazione.
- *Tripadvisor*⁸. È uno dei maggiori siti di viaggi online. Aiuta i turisti di tutto il mondo a programmare la propria vacanza grazie a recensioni e opinioni, scritte dagli stessi utenti, relative ad alloggi, voli e ristoranti. Le informazioni riguardanti un particolare hotel, piuttosto che una pizzeria, vengono unite a quelle di altri clienti del medesimo servizio, facendo sì che ne derivi una descrizione accurata, con pregi e difetti. Maggiore è il numero di contributi per un determinato servizio presente nel sito, più affidabile è il giudizio finale che ne deriva.
- *Delicious*⁹. Quando un utente trova un sito interessante può cliccare sull'icona del bookmark, memorizzando la pagina. Delicious è un sito di social bookmarking, dove gli utenti segnalano agli amici, ma anche a tutti gli altri utenti della Rete, i loro link preferiti, assegnando dei tag che ne favoriscano la ricerca e la catalogazione. Uno studio del 2007 [61] ha mostrato che i sistemi di tagging collaborativo presentano dinamiche simili ai sistemi complessi. Sebbene non esista un vocabolario centrale, la distribuzione dei tag può essere sintetizzata in grafi di folksonomie¹⁰ che, partizionati in modo efficiente, danno luogo ad un vocabolario condiviso. Tale vocabolario è una forma di collective intelligence, che emerge dalle azioni decentrate delle comunità di utenti.

Caso di studio: Crowdcast [62]

I Marketing Manager si trovano quotidianamente davanti a importanti interrogativi. Quanto spendere in pubblicità per un nuovo prodotto o per espandere il mercato? Quanto dovrebbe costare un nuovo prodotto? Quante scorte tenere in magazzino? Quale fornitore costa meno e allo stesso tempo è meno rischioso? Analizzando i dati storici e conducendo ricerche di mercato si possono ottenere indicazioni su alcuni di questi quesiti, ma ciò va bene solo quando si ha la certezza che il futuro ricalchi il passato. Tali approcci, invece, non sono più adeguati quando viene lanciata sul mercato una nuova linea di prodotti, che non ha nessuna attinenza con quelle proposte in passato.

⁷ <http://www.crowdcast.com/>

⁸ <http://www.tripadvisor.it/>

⁹ <http://www.delicious.com/>

¹⁰ Folksonomia è un neologismo derivato dal termine inglese folksonomy che descrive una categorizzazione di informazioni generata dagli utenti mediante l'utilizzo di parole chiave (o tag) scelte liberamente. Il termine è formato dall'unione di due parole, folk e tassonomia; una folksonomia è, pertanto, una tassonomia creata da chi la usa, in base a criteri individuali.

Una valida alternativa sarebbe porre la richiesta a chi conosce a fondo quel determinato segmento di mercato, instaurando dei canali di comunicazione, trasversali alle società interessate, che connettano le persone che necessitano di risposte con quelle che possiedono le soluzioni.

Crowdcast propone esattamente questa tipologia di approccio, integrando le risposte all'interno di un sistema basato sulla ricompensa, che premia le previsioni che si rivelano corrette.

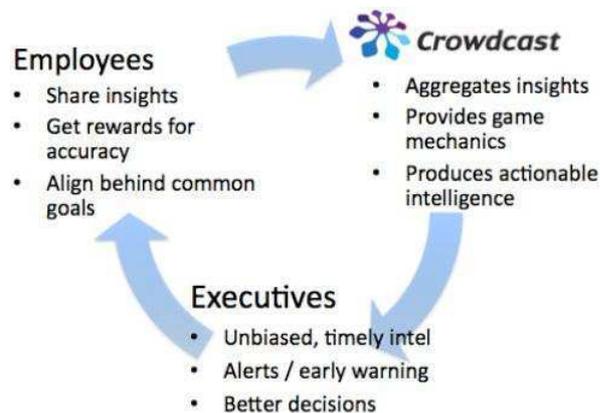


Figura 4.2: Processo di Crowdcast [62]

Esperti del settore condividono le loro intuizioni selezionando un intervallo di risultati possibili. Ad esempio, nel contesto delle spedizioni, qualcuno potrebbe predire che la consegna di un determinato prodotto avvenga entro un certo periodo, supponiamo tra il 5 e il 15 marzo. Qualcun altro, avendo informazioni diverse, potrebbe invece predire la medesima consegna in un altro periodo, ad esempio tra il 16 e il 26 Febbraio. Inoltre, ciascun partecipante può motivare la sua scelta, spiegandone le ragioni.

Il meccanismo elaborato da Crowdcast stabilisce che rispetto ad uno specifico evento vi siano delle quote probabilistiche relative a ciascun periodo, in base alle possibilità di accadimento, che si aggiornano continuamente in base alle “scommesse” degli utenti. Ovviamente, maggiore è il numero di individui che puntano su un determinato periodo, minore è la quotazione. Ad esempio se 30 persone hanno scommesso sulla data di consegna tra il 16 e il 26 febbraio e soltanto 5 sul periodo 5-15 marzo, a parità di puntata, in caso di previsione corretta, vincerebbero un compenso maggiore i secondi, e tra questi il primo più di tutti, e a scalare tutti gli altri. Ogni qualvolta una nuova predizione viene immessa nel sistema, il meccanismo analizza il range, e aggiorna le singole quote tramite una rete Bayesiana pesata(vedi figura 4.3).

Si consideri il seguente scenario: un’azienda manifatturiera che si occupa di elettrodomestici ha dei problemi a causa di previsioni poco accurate riguardanti le date di consegna dei nuovi prodotti, il numero dei potenziali nuovi clienti, i fattori di rischio e i costi dei materiali. Il processo di pianificazione è basato su dati statistici, che vengono analizzati e sintetizzati dai manager dell’azienda. Questa strategia di programmazione richiede un dispendio temporale eccessivo, senza produrre risultati che possano giustificare tale spreco (alcune tipologie di previsioni non raggiungono nemmeno il 20% dell’accuratezza).

Sfida: Migliorare le performance del programma di promozione dei prodotti, incrementando la visibilità sul mercato, velocizzando i tempi di consegna e ottimizzando la gestione.



Figura 4.3: Il metodo di Crowdcast [62]

Soluzione: Crowdcast sfrutta le conoscenze di 500 dipendenti in tema di marketing, R&D, finanza, spedizioni e industria manifatturiera per creare una strategia programmatica efficiente ed affidabile in tempi rapidi.

Vediamo come cambia il processo:

PRIMA	DOPO
I piani di lancio dei prodotti venivano stabiliti dai responsabili societari specialisti del settore	Gli specialisti di Crowdcast appartengono a tutte le categorie e settori (maggiori conoscenze e informazioni disponibili)
Si era incentivati ad essere ottimistici	Si è incentivati ad essere realisti
Previsioni aggiornate di rado	Previsioni in real time, aggiornate ogni qual volta è disponibile una nuova informazione

Cambiando il metodo di pianificazione e le campagne promozionali, e sfruttando il coinvolgimento dei dipendenti, si sono ottenuti risultati sorprendenti:

- Previsioni più accurate del 37%;
- Diminuzione dei tempi di pianificazione dalle 2 settimane a pochi giorni;
- Riduzione delle scorte di magazzino con conseguenti risparmi economici;
- Maggiore coinvolgimento di tutti i dipendenti, che si sentono più partecipi e importanti, migliorando il complessivo morale societario.

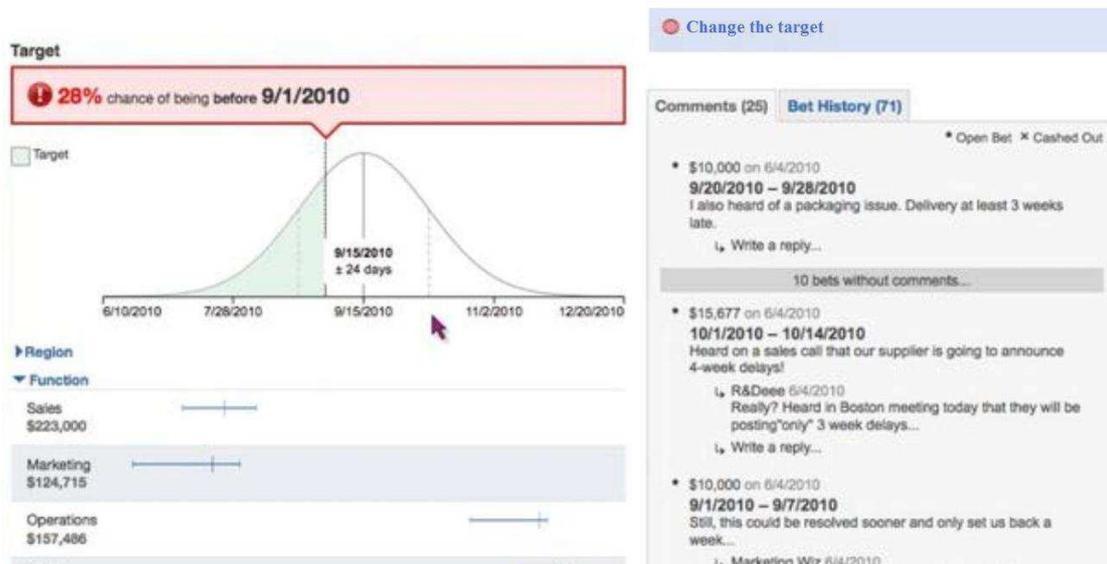


Figura 4.4: Esempio di schermata Crowdfunder [62]

4.2.3 Livello 2: CI inconsapevole ed implicita

In questo caso la Collective Intelligence scaturisce dai dati prodotti in modo implicito dagli utenti di applicazioni e servizi e senza che quest'ultimi siano consci di fornirla. Ad esempio, la prenotazione di un volo aereo ha come scopo l'acquisto di un biglietto, ma la transazione andata a buon fine può essere utilizzata a fini statistici per rilevare le preferenze sulle destinazioni raggiunte da una compagnia. Appartengono a questo livello anche le conoscenze che si possono produrre a seguito dell'analisi di dati grezzi tramite strumenti sofisticati di Business Intelligence che sono in grado di individuare relazioni interessanti e che prima non esistevano. Questa è la forma più alta di Collective Intelligence, ma la meno diffusa, e la più difficile da ottenere a causa della natura eterogenea dei dati e delle molteplici fonti da cui si può attingere. Per farlo emergere sono quindi necessari mezzi di descrizione, ricerca e selezione.

Esempi di CI di livello 2

A causa della complessa natura di questo livello di CI, attualmente non si trovano molti esempi di utilizzo. Uno di questi è Baynote.

- *Baynote*¹¹. È una società di marketing digitale che, utilizzando un "motore di affinità", chiamato UseRank, analizza i micro-comportamenti di clienti e società produttrici, determina le corrispondenze tra domanda e offerta, in modo da far pervenire il messaggio corretto alle persone appropriate nel più breve tempo possibile. Tutto ciò viene gestito in modo automatico grazie ad una Collective Intelligence Platform.

Caso di studio: Baynote

Baynote fornisce strumenti di ricerca che garantiscono risultati altamente affidabili, suggerimenti di contenuti e prodotti coerenti con gli interessi dell'utente, e offerte ad hoc via email. Migliaia di professionisti di eCommerce, marketing e consulenza si affidano a Baynote per incrementare il numero dei propri clienti. A tale scopo viene utilizzata una piattaforma, la Adaptive Web Suite, che include applicazioni adattive, un centro di ottimizzazione e una piattaforma di Collective Intelligence.

¹¹ <http://www.baynote.com/>

Le applicazioni adattive offrono ricerche e suggerimenti personalizzati (ad esempio mostrando all'utente contenuti che interessano ad altri con profilo simile al suo), email commerciali mirate alle necessità del destinatario, e pagine web che variano in base alle caratteristiche e alle preferenze del visitatore.

Il centro di ottimizzazione espone un'interfaccia integrata che monitora e gestisce tutti i prodotti dell'Adaptive Web Suite, fornendo report, test e previsioni. Inoltre, Baynote mette a disposizione dei suoi clienti il *Search Insights*, uno strumento che memorizza i termini più ricercati da una community e che può essere utilizzato dalle società per attrarre nuovi clienti con pubblicità accattivanti, contenenti tali espressioni.

La Collective Intelligence Platform (CIP) è la base su cui si fonda l'intera Adaptive Web Suite. Il motore di questa componente è la tecnologia brevettata *UseRank*, un'applicazione capace di rilevare i gradi di affinità tra utenti e di analizzarne i cosiddetti micro comportamenti (tendenze comuni), potendo così individuare le loro preferenze.

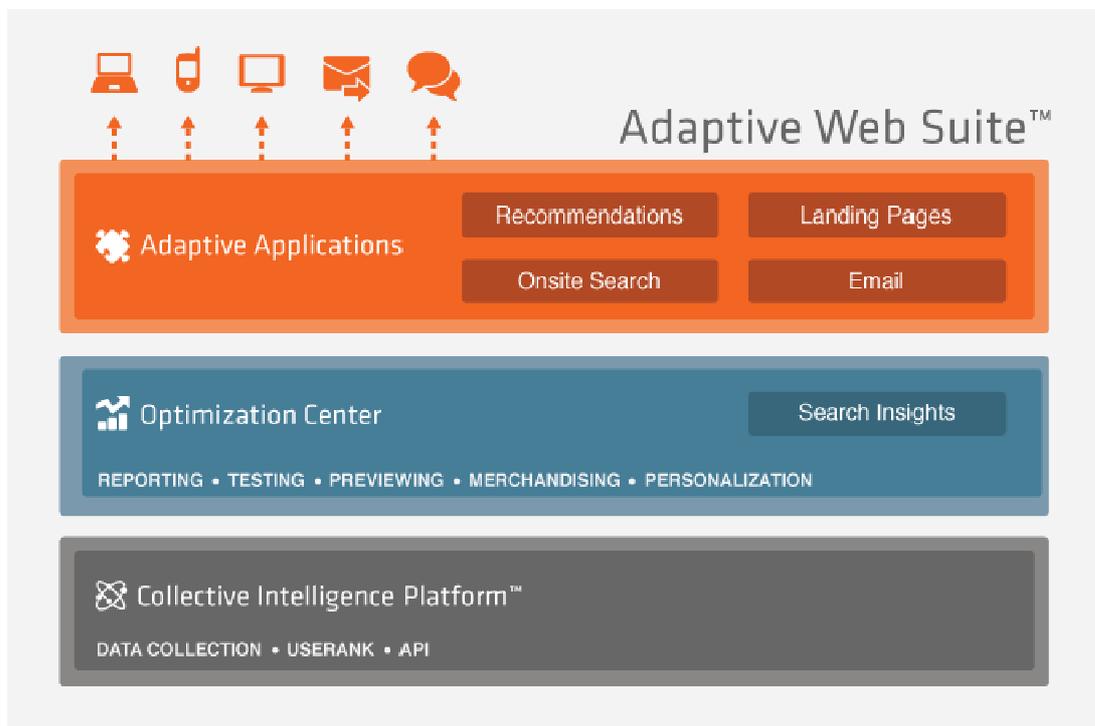


Figura 4.5: La Adaptive Web Suite di Baynote [63]

UseRank adotta un approccio totalmente differente da qualsiasi altra tecnologia web che produca rapporti su dati di alto livello, come le conversioni¹², di cui hanno bisogno gli operatori per dare un senso ai dati. UseRank analizza i micro comportamenti per determinare le esigenze dei clienti, identifica altri clienti che condividono i medesimi bisogni, e infine promuove contenuti e prodotti che quest'ultimi trovano utili e attrattivi. Il risultato finale è che agli utenti vengono offerti contenuti altamente coerenti con le loro preferenze del momento, anche qualora l'interesse dovesse cambiare tra una visita e la successiva.

La raccolta di dati di Baynote tiene traccia di tutti i micro comportamenti del visitatore di una

¹² Nella pubblicità online si verifica una conversione quando un clic su un annuncio genera un comportamento vantaggioso all'azienda da parte dell'utente (ad esempio un acquisto, una registrazione, o la visualizzazione di una pagina)

pagina web, come i movimenti del mouse, lo scorrimento di una pagina, la sottolineatura di un testo, ma anche di comportamenti più palesi, come i click-path (la sequenza di link seguita in un certo sito web), le conversioni, il tempo di sosta (il tempo dedicato ad una pagina), le parole che compaiono nei link più cliccati e così via. Le soluzioni tradizionali ignorano la maggior parte dei comportamenti descritti a causa dell'enorme volume di dati che sono coinvolti in tale processo, ma anche perché spesso gli utenti di tali soluzioni non sono in grado di tradurre le statistiche raccolte in dati significativi. UseRank, invece, aggrega in automatico tali dati utilizzando euristiche di vario genere (Baynote ne ha brevettate 24), che permettono di ricavare in tempo reale informazioni utili. UseRank, ad esempio, assegna un punteggio alla forza attrattiva che ciascun utente conferisce ai differenti prodotti e contenuti. Per massimizzare la rilevanza dell'esperienza-utente, UseRank identifica una massa critica di clienti che condividono le medesime preferenze, e analizza i prodotti più utili e accattivanti per quel determinato gruppo. Le applicazioni della Adaptive Web Suite successivamente promuovono tali prodotti ai singoli individui. Basandosi sulle preferenze (*intent*), la forza attrattiva dei contenuti (*content engagement*) e la *Collective Intelligence*, è stata dimostrata una maggiore efficacia nella proposizione di contenuti rilevanti ed una migliore esperienza utente, rispetto alle tradizionali statistiche usate (conversioni, dati storici e informazioni sul profilo).

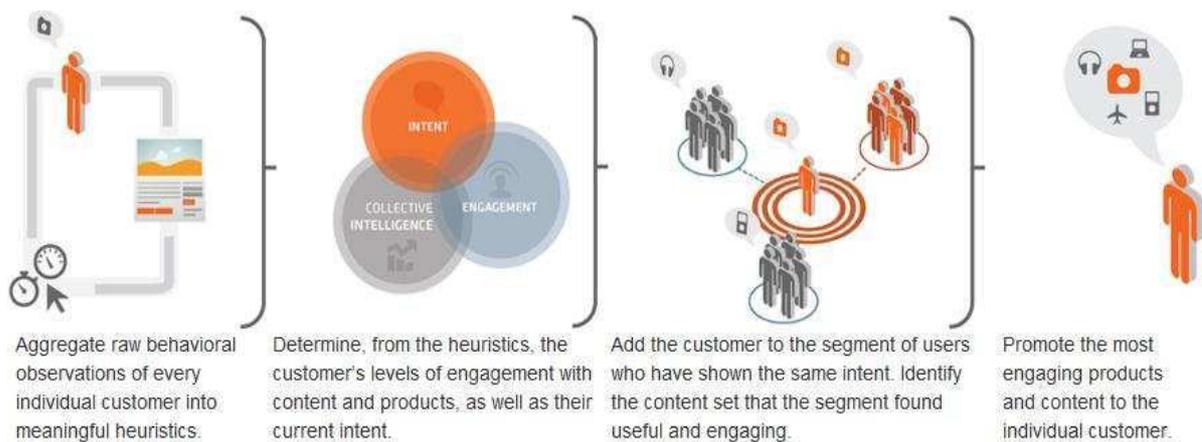


Figura 4.6: Il funzionamento di UseRank [64]

Un esempio dell'efficacia del servizio offerto da Baynote si può osservare nel caso della Stelter Company. In qualità di fornitore leader di servizi di marketing relativo alle donazioni, la Stelter aiuta più di 2600 organizzazioni no-profit a livello nazionale nel raccogliere fondi principalmente attraverso donazioni differite (come lasciti, fondi fiduciari, o Charitable gift annuities¹³). Questi tipi di donazioni sono indicate come "planned gifts" perché richiedono una pianificazione e, spesso, l'aiuto di consulenti professionali. A differenza delle offerte libere, le donazioni pianificate sono tipicamente costituite da beni che rimangono di proprietà del donatore fino alla sua morte, e solo successivamente passano al beneficiario.

Benché l'85 per cento degli americani regolarmente faccia beneficenza, meno del 10 per cento si affida a donazioni pianificate. La causa principale di una così bassa percentuale è da ricercare nella mancanza di conoscenza dei vari tipi di opzioni disponibili. Stelter mette a disposizione le sue competenze per aiutare le persone a destreggiarsi tra le varie possibilità. Per conto dei loro clienti, esegue campagne promozionali utilizzando email e inserzioni sul

¹³ Charitable gift annuities: In cambio di una donazione in contanti o di titoli, una fondazione si impegna a versare un importo fisso annuale al donatore o a un beneficiario

web. Circa il 30 per cento delle entrate della Stelter proviene dai contenuti pubblicati su circa 1.200 siti web no profit a livello nazionale, tra cui Make-AWish Foundation, Georgetown University, Harvard Medical School, American Diabetes Association, CARE USA.

La Stelter Company, quindi, deve occuparsi di pubblicare contenuti analoghi ma in siti web gestiti da organizzazioni diverse, il che significa sapersi rivolgere a utenti eterogenei tra loro. La sfida consiste nell'analizzare i diversi comportamenti dei donatori, per poterli assistere al meglio nella scelta della forma di beneficenza a loro più adatta, e trovare una strategia efficace di promozione delle donazioni pianificate. Stelter ha così scelto di affidarsi al sistema di suggerimenti fornito da Baynote.

La Stelter Company, dovendo gestire il sistema di programmazione di contenuti di centinaia di organizzazioni no profit, ha a disposizione una vasta rete di utenti e una banca dati di contenuti condivisi. Utilizzando queste fonti, grazie agli strumenti forniti da Baynote, ha la possibilità di raccogliere i termini più cercati e i dati comportamentali presenti nei vari siti gestiti. Aggregando le informazioni relative ai vari clienti in un'unica vista globale, può giungere così ad una comprensione più approfondita dei comportamenti dei visitatori.

I contenuti suggeriti sono raggruppati in una lista, stilata in collaborazione con la Stelter Company, e, poiché non tutti gli enti di beneficenza sono in grado di supportare ogni tipo di attività benefica, sono stati elaborati meccanismi integrati, come liste nere e blocchi, per filtrare i contenuti.

In seguito alle modifiche apportate da Baynote, si sono potuti riscontrare risultati positivi già nel breve periodo:

- i visitatori sono maggiormente catturati da risultati pertinenti alle loro preferenze, e il tempo che dedicano alla ricerca della propria soluzione è raddoppiato;
- viene calcolato un rating per ogni contenuto in base all'attrattiva che produce nei clienti, che può essere utilizzato come misura del valore del contenuto e consente all'azienda di stimare anche la qualità del servizio offerto alla coda lunga (la long tail¹⁴), che ha interesse per tipi di donazioni meno comuni;
- l'analisi di queste informazioni ha permesso di capire quale tipo di contenuti i visitatori ricercano effettivamente e, poiché in questo settore la qualità spesso è più importante della quantità¹⁵, sono sufficienti anche poche donazioni per ottenere un significativo ritorno sugli investimenti.

¹⁴ L'espressione coda lunga è stata coniata da Chris Anderson in un articolo dell'ottobre 2004 su Wired Magazine per descrivere alcuni modelli economici e commerciali, come ad esempio Amazon.com o Netflix. In questi modelli una popolazione ad alta frequenza è seguita da una popolazione a bassa frequenza, che diminuisce gradatamente (tail off).

¹⁵ Per molti enti di beneficenza la donazione a lungo termine media è di circa 50.000 dollari

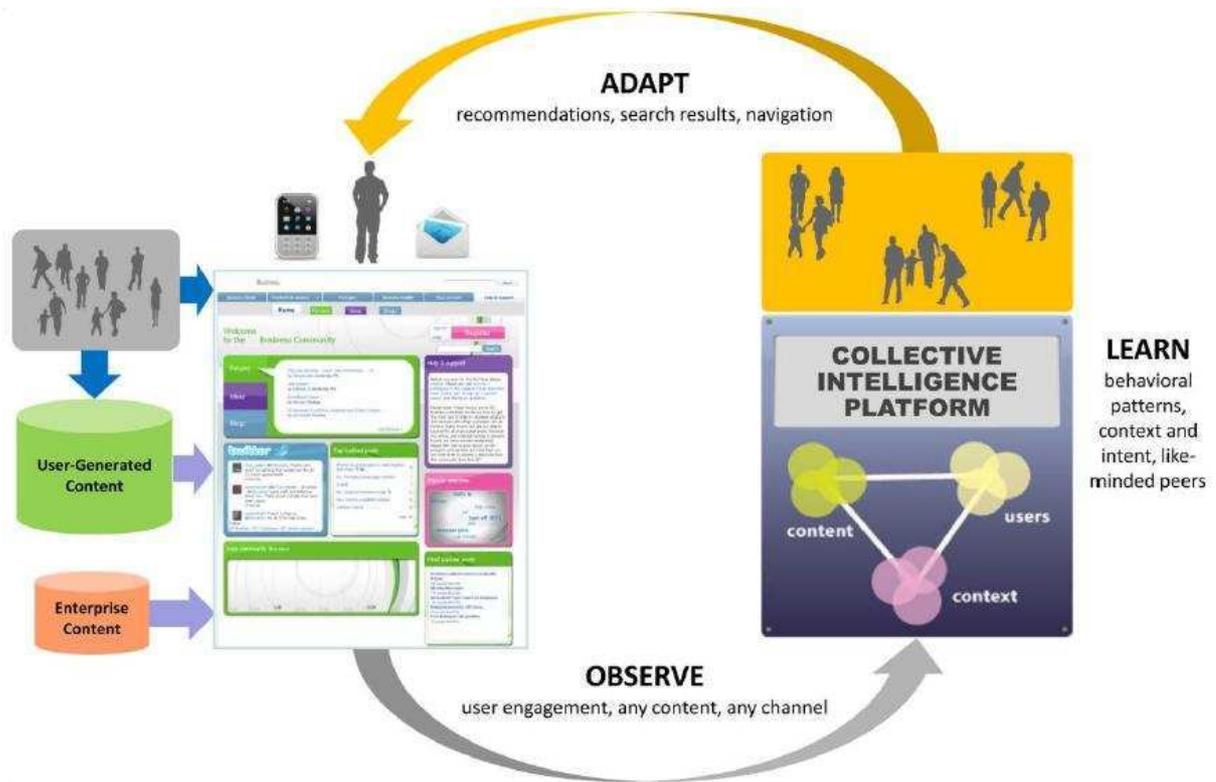


Figura 4.7: Il processo di Baynote [65]

4.3 Panoramica sulla Collective Intelligence

Le applicazioni e i siti del Web 2.0 non solo contengono le conoscenze da cui si origina la CI, ma costituiscono anche le piattaforme in cui le persone, interagendo e collaborando, creano l'Intelligenza Collettiva. Gli utenti, quindi, giocano un ruolo attivo contribuendo con la generazione di contenuti personali. Video e immagini vengono caricati in siti di condivisione come Flickr, Panoramio e YouTube. Tali contenuti possono essere accompagnati dai cosiddetti *user metadata*, cioè descrizioni, tag e altri tipi di informazioni aggiuntive, prodotte dagli utenti. I contenuti possono anche essere testi strutturati (come nel caso di Wikipedia), le cui composizioni derivano dalla collaborazione di singoli individui. Forum e blog sono altre fonti di Collective Intelligence, poiché le conversazioni e i post degli utenti possono essere recuperate e analizzate, come vedremo nella sezione 4.3.3. Infine, un'altra importante risorsa sono i feedback degli utenti. Tali informazioni possono essere raccolte da vari siti relativi a prodotti e servizi, contenenti recensioni degli utenti. Siti web che pubblicizzano o vendono oggetti elettronici, computer, libri, viaggi, sono solo alcuni esempi nei quali gli utenti possono commentare e valutare i prodotti e i servizi da loro stessi acquistati.

Questo sottocapitolo è una sorta di riorganizzazione logica, e allo stesso tempo un approfondimento, dei concetti espressi finora, fornendo una classificazione delle molteplici sorgenti dell'intelligenza collettiva, spiegando quali motivazioni si celano dietro il contributo volontario alla generazione della CI, e descrivendo alcuni approcci che si possono adottare per ottenere la Collective Intelligence.

4.3.1 Le sorgenti di Collective Intelligence

Finora abbiamo trattato l'Intelligenza Collettiva come un'entità astratta, normalmente generata dalle conoscenze di una moltitudine di individui, ma senza mai precisare chi fossero queste persone e in che termini fornissero il loro contributo alla realizzazione della CI. Nel corso del progetto WeKnowIt [66], sono state individuate cinque fonti di Collective Intelligence che possono essere prese in considerazione singolarmente o combinate tra loro:

1. *Personal Intelligence*. Contributi di singoli individui che forniscono informazioni di interesse.
2. *Media Intelligence*. Contenuti digitali e informazioni contestuali, prodotte appositamente dagli utenti o preesistenti.
3. *Mass Intelligence*. Feedback numericamente rilevanti, che consentano l'estrazione e il monitoraggio di trend.
4. *Social Intelligence*. Interazioni sociali tra utenti, dalla cui analisi è possibile ricavare informazioni utili per il controllo e la gestione delle reti sociali.
5. *Organizational Intelligence*. Interazioni a livello aziendale, in cui sono coinvolte le gestioni di diversi livelli di conoscenze.

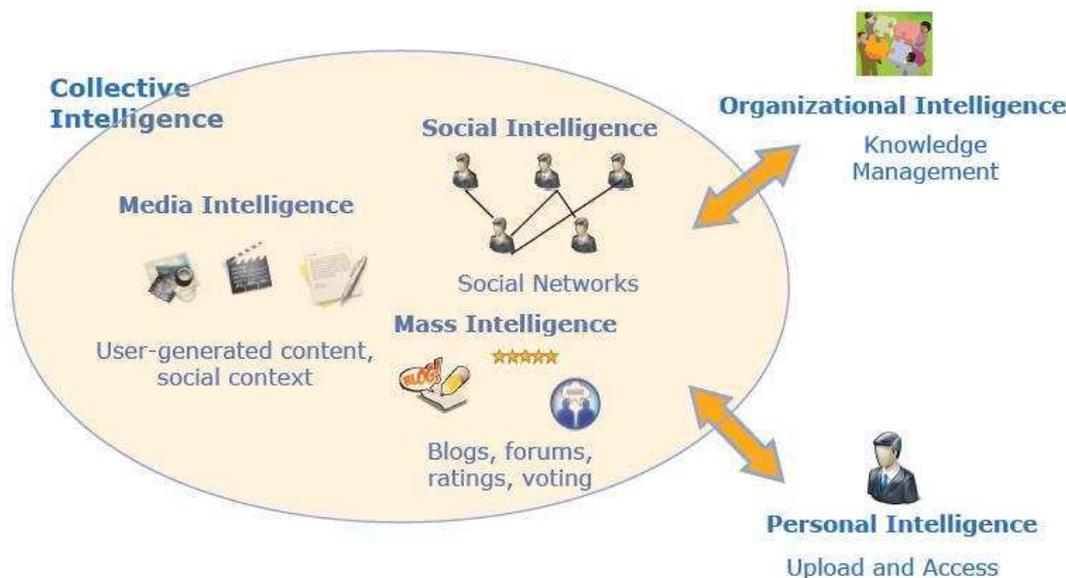


Figura 4.8: Le fonti di Collective Intelligence secondo WeKnowIt [66]

Personal Intelligence

I singoli individui possono produrre informazioni utili attraverso i propri dispositivi connessi ad Internet (pc, telefoni, smartphone, etc.). Come si può evincere anche dalla figura 4.8, la Personal Intelligence è allo stesso tempo sorgente e consumatrice di Collective Intelligence. Un caso esemplare a tal proposito è Wikipedia, dove gli utenti producono individualmente i contenuti, che poi altri utenti, sempre a livello individuale, utilizzano per informarsi riguardo ad argomenti a loro prima ignoti.

Media Intelligence

La Media Intelligence è l'intelligenza riscontrabile nei contenuti digitali (immagini, video, audio, testi) e che può essere arricchita dall'analisi di informazioni contestuali a livello spazio temporale (luoghi e orari), a livello personale (profili utente e preferenze) e a livello sociale

(tag, valutazioni e profili di gruppo). A questo scopo, vengono utilizzate tecniche automatiche di analisi dei contenuti per i diversi media, che consentano di estrarre conoscenza dai contenuti. Dovendo gestire grandi quantità di dati, spesso mal organizzati, i suddetti metodi comprendono tecniche di machine learning¹⁶, di data mining¹⁷ e di information retrieval¹⁸.

Mass Intelligence

La Mass Intelligence risiede essenzialmente nei feedback degli utenti e acquisisce valore e significatività all'aumentare del numero di pareri raccolti. L'analisi di opinioni prodotte in massa permette il raggruppamento e la classificazione delle informazioni prodotte, così come la categorizzazione di informazioni ed eventi. Le tipologie di Mass Intelligence individuate da WeKnowIt possono essere riassunte in quattro punti:

1. *Mass question answering*. Risposte a domande specifiche.
2. *Mass interaction feedback*. Feedback espliciti ed impliciti prodotti dall'interazione di utenti.
3. *Mass classification and clustering*. Classificazioni e valutazioni.
4. *Mass evolution analysis*. Analisi delle variazioni comportamentali degli utenti.

Social Intelligence

La Social Intelligence si può rilevare a partire dalle interazioni che nascono nel cosiddetto Social Web, cioè, in quelle piattaforme, come blog, siti di media sharing e social network, che consentono agli utenti di comunicare direttamente o indirettamente tra loro. L'analisi di queste interazioni permette di monitorare e riconoscere le preferenze e le necessità degli utenti. La Social Intelligence, inoltre, fornisce un mezzo per combinare ed amplificare l'impiego di Media Intelligence, Mass Intelligence e Organizational Intelligence.

Organizational Intelligence

L'Organizational Intelligence si manifesta a livello aziendale, e quindi nelle interazioni che avvengono in un ambiente lavorativo relativamente chiuso. Si noti, però, che questo tipo di conoscenza non deriva solo dai contributi dei singoli, bensì dall'azione combinata delle sorgenti descritte in precedenza, sebbene in un ambito specifico. Come nel caso della Personal Intelligence, L'Organizational Intelligence possiede il duplice ruolo di sorgente e consumatrice di CI. In particolare, l'utilizzo che se ne può fare è relativo ai processi di *decision making*, fornendo alle persone giuste l'informazione giusta al momento giusto.

Riassumendo, gli input della CI provengono dalle conversazioni degli utenti, dall'upload di media, da votazioni, note e tag. Tali input, grazie agli strumenti del Web 2.0, hanno già creato un tipo di intelligenza sotto forma di suggerimenti, e layout personalizzati. Usando tag e analisi statistiche, l'intelligenza che ne deriva raggiunge un livello di complessità maggiore, fornendo servizi personalizzati, trend, connettività sociale ed estrazione di significato dai contenuti.

¹⁶ Il machine learning consiste in tecniche di apprendimento automatico. A partire da dati o esempi, oppure tramite algoritmi di clustering è possibile sintetizzare nuove forme di conoscenza.

¹⁷ Il data mining ha per oggetto l'estrazione di un sapere o di una conoscenza che prima non esistevano a partire da grandi quantità di dati

¹⁸ L'information retrieval è l'insieme delle tecniche utilizzate per il recupero mirato dell'informazione in formato elettronico.

4.3.2 Le componenti motivazionali nel contribuire alla CI

La ricerca delle sorgenti di Collective Intelligence è stata oggetto anche di una indagine del MIT Center for Collective Intelligence, e il risultato è stato pubblicato in un paper dal titolo *Harnessing Crowd: Mapping the Genome of Collective Intelligence* [67]. Il termine utilizzato dagli autori dell'articolo per riferirsi al concetto di sorgente è stato gene, con un chiaro riferimento figurativo all'unità ereditaria fondamentale degli organismi viventi. La distinzione riguardo alle sorgenti in questo caso è molto più grezza, facendo risalire l'Intelligenza Collettiva a soli due possibili tipologie di sorgenti: le organizzazioni gerarchiche e le masse. Di maggiore interesse è invece l'analisi delle motivazioni che spingono gli individui a produrre consapevolmente forme di Collective Intelligence: la ricompensa economica, la passione per l'argomento, il miglioramento della propria reputazione.

La ricompensa economica

La prospettiva di ricevere una ricompensa economica per molte persone è un importante incentivo a mettere a disposizione le proprie conoscenze. Per alcune, probabilmente, è anche l'unica motivazione. In certi casi i pagamenti sono diretti, una sorta di salario, come accade nelle challenge di Innocentive; in altri, la partecipazione può rappresentare una speranza di vedere migliorate le proprie condizioni economiche future, ad esempio migliorando la propria visibilità, come nel caso dei programmatori che partecipano allo sviluppo dei moduli di Linux.

La passione per l'argomento

La passione per un certo argomento è un altro importante fattore motivazionale, abbastanza forte da poter esistere anche qualora non ci sia una prospettiva di guadagno. Il "gene" della passione può manifestarsi in varie forme: le persone possono essere motivate dalla gioia intrinseca alla partecipazione ad un'attività che li appassiona, oppure dall'opportunità di collaborare con altre persone che hanno interessi simili e con cui ci può essere uno scambio di conoscenze, o ancora dall'idea di contribuire ad un progetto di grande rilevanza che li fa sentire importanti. Un esempio che racchiude tutte e tre le espressioni di questa componente motivazionale è sicuramente Wikipedia.

Il miglioramento della propria reputazione

Il riconoscimento delle proprie capacità da parte di altre persone è una fonte di motivazioni, che però senza una delle due precedenti componenti difficilmente può essere sufficiente. Ad esempio, i programmatori di software open source sono motivati dal desiderio di essere apprezzati per il loro contributo, ma sono spinti anche da una forte passione per la programmazione. Un modo per accrescere la spinta fornita da questa espressione motivazionale può essere quello di pubblicare il nome di chi si adopera maggiormente in una sorta di classifica dei "top contributor".

4.3.3 Come si forma la Collective Intelligence

Molte applicazioni del web 2.0 aggregano e analizzano dati. Il tipo di contenuti, la dimensione e l'analisi variano molto tra le diverse applicazioni. Ci sono due aspetti principali della CI che vale la pena sottolineare. Il primo riguarda la dimensione dei dati, più l'applicazione produce informazioni che si possono considerare un'istanza di CI. Il secondo, invece, riguarda quanto l'elaborazione e l'analisi di dati sono coinvolti nella produzione di conoscenze. Mentre in alcuni casi presentare i semplici risultati in modo aggregato è sufficiente per fornire informazioni aggiuntive, in molti altri è necessaria un'ulteriore analisi per produrre informazioni utili e conoscenza. Vediamo, ora, alcuni esempi di siti web ben conosciuti in relazione a queste due dimensioni, cioè la dimensione e l'analisi di dati.

Nel caso di Wikipedia, l'analisi dei contenuti è abbastanza bassa e ci sono processi supportati solo al livello di organizzazione e presentazione dei dati. Si rileva comunque una forte partecipazione collettiva da parte degli utenti e l'aggregazione di molte risorse.

YouTube usa l'analisi dei tag per la ricerca, per i servizi di personalizzazione, e per i suggerimenti, e si avvale di tecniche di analisi dei contenuti per l'identificazione delle violazioni di copyright.

In modo simile Flickr usa i tag per la ricerca e i suggerimenti personalizzati ma anche per raggruppare elementi simili in cluster (Tag Cluster), che sono il risultato dell'analisi dei contenuti caricati dagli utenti su larga scala.

Anche Digg, un sito "sociale" di notizie, con il quale è possibile scoprire e condividere contenuti, usa metodi di analisi statistica delle valutazioni degli utenti.

Amazon crea profili personali, e sulla base delle history degli acquisti, sulle visite, sulle recensioni e valutazioni degli utenti suggerisce i prodotti in modo customizzato.

Infine, Google ha sviluppato PageRank, un metodo di classificazione delle pagine web. Con tale approccio, ogni pagina web vota le pagine web che sono collegate ad essa, e dall'analisi dei link definisce l'importanza di una pagina web.

Sebbene in alcuni degli esempi sopra citati un minimo di analisi dei dati viene effettuata, attualmente le metodologie presenti nelle applicazioni del web 2.0 non intercettano un livello di intelligenza nascosta, generata dalle conoscenze degli utenti. Per ottenerlo sono necessarie analisi maggiormente accurate e integrazioni di conoscenze provenienti da più fonti (contenuti, contesti, feedback, interazione sociale, etc.).

	Action	Display	Feedback
Digg	Submitting a news story	Upcoming, popular, homepage	Digg, share and bury stories
Amazon	Writing a product review	Mostful, most recent	Is this Helpful? Report this, Comment
Netflix	Rating a movie	Recommended movies	Add to queue, Rate movie, Not interested
Google	Writing a web page	Results based on relevancy	Link between web pages, Click on search results
Wikipedia	Starting an article	Article page	Edit article over time
Delicious	Saving and tagging a bookmark	Most popular, Related tags, all tags	Copy bookmarks
Flickr	Uploading and tagging a picture	Interestingness, popularity, clusters	Tagging, setting Favorites
YouTube	Uploading a video	Youtube interface, embedded in blogs	Favorite it, Report it, Embed it

Tabella 4.2: manifestazioni di collective intelligence in alcuni siti del Web 2.0 [68]

Il modello generale

Al fine di correlare gli utenti con i contenuti e gli utenti tra di loro, c'è bisogno di un linguaggio comune per calcolare la correlazione tra gli item¹⁹, tra gli utenti, e tra gli utenti e gli item. Inoltre, poiché molti contenuti esistono sotto forma di testi non strutturati, è utile capire come i metadati possano essere ricavati a partire da questo tipo di testi. Una soluzione a

¹⁹ Per item, in questo contesto si intendono tutti i possibili contenuti di un sito o di una applicazione del Web 2.0.

queste problematiche si può trovare nel libro *Collective Intelligence in Action* di Satnam Alag [69].

Come mostrato nella figura seguente (4.9), la maggior parte delle applicazioni comprendono utenti e item. Gli item possono essere testi, video, foto, blog, post con domande e risposte, ma anche prodotti o servizi. Ad ogni item sono associati un certo numero di metadati, che possono essere espressi nella forma di keyword (fornite direttamente o estratte tramite algoritmi di analisi testuale), tag, rating, o attributi di vario genere.

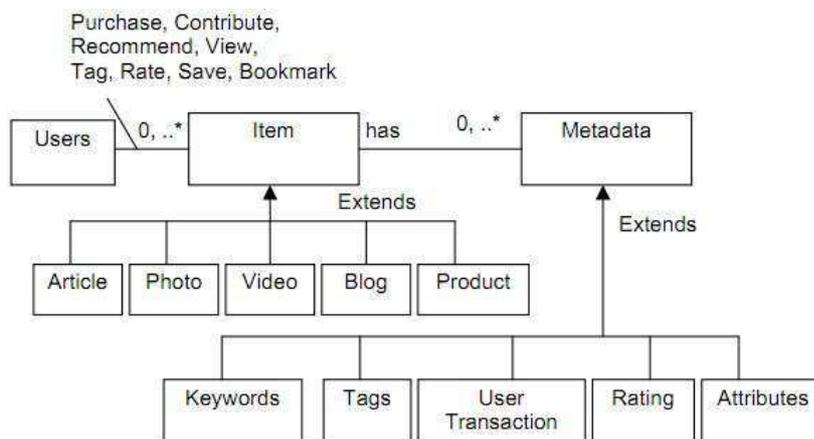


Figura 4.9: modello utenti-item-metadati [69]

Spesso anche agli utenti sono associati dei metadati. In questo caso sono riassunti in un profilo dell'utente, e specificano dati personali, come età, sesso, posizione geografica, professione.

Rappresentazione dei metadati

Ciascun item e ciascun utente, quindi, sono definiti in base ai metadati a loro associati. Un modo per rappresentarli è l'utilizzo dei *term vector*, cioè una rappresentazione vettoriale dei metadati. In tal caso si parla di *vector space model*²⁰, come modello di rappresentazione dei documenti. L'insieme di tutti i termini e tag definiscono il vocabolario del sistema. Il vocabolario relativo ai metadati degli utenti e quello relativo ai metadati degli item possono coincidere o essere distinti. Nel secondo caso è possibile confrontare utenti con item, nel primo invece no.

Ad ogni termine, inoltre, è associato un peso, che dipende da due fattori, la *term frequency* (TF), cioè il numero di volte che un termine appare nelle descrizioni dell'item, e la *inverse document frequency* (IDF), che rileva termini normalmente poco usati, e proprio per questo motivo la loro presenza risulta significativa. Una volta che si dispone dei *term vector* per due oggetti è possibile confrontarli sulla base della somiglianza dei due vettori che li descrivono.

Quando si ha a che fare con singole keyword, tag o rate è facile creare un *term vector*, in quanto ciascun termine corrisponde ad un elemento del vettore. I problemi sorgono quando si ha a che fare con item rappresentati da testi non strutturati.

²⁰ Il *vector space model* è un modello algebrico per rappresentare documenti di testo (ma anche oggetti in generale) come vettori. Di solito l'utilizzo di questo modello è riscontrabile nell'*information filtering*, nell'*information retrieval*, nell'indicizzazione di testi e nella classificazione di contenuti.

Approccio per testi non strutturati

Termini composti e frasi devono essere smembrati in singoli elementi e convertite nella loro versione primitiva, adottando un procedimento chiamato stemming.

Si consideri, ad esempio, il titolo di un libro: “La solitudine dei numeri primi”. In questa frase ci sono cinque termini, *la*, *solitudine*, *dei*, *numeri*, e *primi*. La prima operazione da fare è eliminare punteggiatura, congiunzioni, articoli, preposizioni e avverbi. In questo caso rimangono solo i termini *solitudine*, *numeri* e *primi*. Se ci sono delle maiuscole, va trasformato tutto in minuscolo. Nell’esempio proposto non sono necessarie modifiche. A questo punto si devono troncare le parole, lasciando solo la radice. Il termine *solitudine* viene trasformato in *solo*; si noti che lo stesso risultato si ottiene da parole come “soli” (plurale), “solitario” (aggettivo) e “soltanto” (avverbio). I termini *stemmizzati*, infine, sono inseriti nel *term vector*.

Ricapitolando, i passaggi necessari per estrarre termini significativi da un testo non strutturato sono quattro:

1. Dividere il testo in termini singoli
2. Eliminare la punteggiatura, le congiunzioni, gli articoli, le proposizioni e gli avverbi
3. Normalizzare le parole rendendole tutte minuscole
4. Applicare lo stemming a ciascun termine

Calcolare le similarità tra item e/o utenti

Come detto, per confrontare due item (o due utenti, oppure un item con un utente) è sufficiente calcolare la similarità tra i vettori che li rappresentano. Ad ogni entry corrisponde un termine, e l’elemento memorizzato è il peso. Supponiamo di avere due vettori, V_1 e V_2 , entrambi di n componenti:

$$V_1 = (x_1, x_2, \dots, x_n) \qquad V_2 = (y_1, y_2, \dots, y_n)$$

Il primo passo da effettuare è normalizzare i vettori nel seguente modo:

$$N(V_1) = \frac{(x_1, x_2, \dots, x_n)}{\sqrt{(x_1^2 + x_2^2 + \dots + x_n^2)}} \qquad N(V_2) = \frac{(y_1, y_2, \dots, y_n)}{\sqrt{(y_1^2 + y_2^2 + \dots + y_n^2)}}$$

Ora è possibile confrontare i due vettori tramite una misura di similarità:

$$\text{Similarità}(V_1, V_2) = \frac{(x_1 \cdot y_1 + x_2 \cdot y_2 + \dots + x_n \cdot y_n)}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2} \cdot \dots \cdot \sqrt{x_n^2 + y_n^2}}$$

Osservazione

Si noti che in quest’ultima formula, i valori delle componenti di V_1 e V_2 sono quelli normalizzati al passo precedente e non quelli iniziali.

Estrarre CI dalle interazioni degli utenti con gli item

Per estrarre una forma d’intelligenza dalle interazioni degli utenti con gli item, non è sufficiente osservare la history di navigazione, cercando di capire a quali contenuti sono interessati. È invece più significativo quantificare la qualità delle loro preferenze. Ad un utente potrebbe piacere un certo item oppure no, ma questi sono due casi estremi, bisogna invece capire quanto gli piace quell’item rispetto agli altri item. Le soluzioni per ottenere una misura qualitativa di tale interesse sono due:

1. far sì che l'utente esprima esplicitamente il suo livello di gradimento, ad esempio, attraverso un valore numerico o una scala di giudizio;
2. derivare il livello di interesse da informazioni implicite, ad esempio, osservando la distribuzione dei click sui link.

Il rating e le recensioni fanno parte della categoria delle espressioni esplicite. Nella categoria delle manifestazioni implicite, invece, troviamo la condivisione di un contenuto, il salvataggio di un bookmark, l'acquisto di un item.

Estrarre Collective Intelligence dai tag

Con il termine tagging si indica il processo di arricchimento degli item con parole chiave freeform (cioè qualsiasi termine di senso compiuto), o brevi frasi, agli item.

Possiamo classificare i tag sulla base di chi li genera. Esistono tre tipi di tag: generati da professionisti (*professionally generated*), generati dagli utenti (*user-generated*) e generati in automatico (*machine-generated*).

Alla prima categoria appartengono i tag generati da esperti, cioè da persone qualificate, che hanno grande familiarità con i contenuti da loro taggati, e che molto spesso sono pagate per tale servizio. Le caratteristiche principali dei tag *professionally generated* sono le seguenti:

- colgono la semantica del contesto a cui appartengono gli item
- forniscono un punto di vista globale
- l'insieme delle parole usate appartengono ad un vocabolario ben definito
- riuniscono i sinonimi in un'unica parola
- possono essere termini composti

L'impiego di professionisti nella produzione di tag specialistici, tuttavia, comporta dei costi, che possono diventare rilevanti se gli item sono molto numerosi.

Un'alternativa molto più economica, addirittura a costo zero, è lasciare che siano gli utenti a taggare gli item di cui usufruiscono. In questo caso, però, i termini utilizzati non appartengono ad un insieme finito e determinato, infatti si lascia totale libertà all'utente di inserire qualsiasi parola ritenga adatta a descrivere l'item o la propria esperienza con esso. Le principali caratteristiche dei tag *user-generated* sono le seguenti:

- comprendono termini familiari all'utente
- colgono la semantica del contesto a cui appartengono gli item
- possono essere termini composti
- forniscono informazioni sull'interazione dell'utente con l'item
- possono racchiudere modi di dire

I tag generati dagli utenti non hanno il problema di avere un costo proprio perché non sono tag professionali, ma per il medesimo motivo risultano anche di qualità inferiore, includendo sinonimi, termini al plurale, e a volte parole volgari. Necessitano, quindi, del medesimo approccio, visto precedentemente, che si riserva ai testi non strutturati.

Un metodo che individua un trade-off tra qualità ed economicità è l'utilizzo di un algoritmo che generi i tag in modo automatico. La condizione necessaria per l'applicazione di un simile metodo ad un insieme di item è che questi siano già descritti in modo testuale. A partire dalle descrizioni, un algoritmo di parsing è in grado di analizzare il contenuto e creare tag appropriati. Le caratteristiche principali dei tag *machine-generated* sono le seguenti:

- utilizzano termini presenti nella descrizione degli item, o loro sinonimi;

- sono quasi sempre termini atomici;
- possono esserci degli errori dovuti all'ambiguità di termini che hanno più significati.

Questi tipi di tag sono particolarmente utili nel caso di item testuali, come i post presenti nei blog e nei social network.

4.4 La Collective Intelligence nell'ambito dei servizi

L'obiettivo di questa tesi è quello di realizzare un service recommender system che sfrutti la Collective Intelligence per fornire suggerimenti affidabili e personalizzati. In questo sottocapitolo, quindi, vediamo quali contributi può dare l'intelligenza collettiva a un sistema di questo tipo. In primo luogo capiremo come la CI possa integrare le descrizioni dei servizi fornite dai service provider, aumentando le informazioni a disposizione degli utenti. Successivamente sarà presentato un approccio, basato su Collective Intelligence, per la ricerca e la selezione dei servizi. Con riferimento al sottocapitolo 4.2, per la descrizione dei servizi vengono utilizzati tutti e tre i livelli di CI, mentre per la ricerca e la selezione si fa riferimento solo a quella di livello 0.

4.4.1 La descrizione di servizi

Come abbiamo visto nel capitolo 2, tra i più interessanti campi di ricerca attualmente in corso di esplorazione troviamo l'applicazione degli strumenti del Semantic Web al mondo dei servizi. Una delle possibili soluzioni offerte da questo connubio è la descrizione semantica dei servizi per renderli facilmente ricercabili e interpretabili, sia da essere umani che da agenti software.

Un eService, secondo [70], può essere descritto semanticamente da cinque elementi complementari:

- *Information Model Descriptions*. Definiscono, tramite l'XML Schema²¹, i modelli di dati utilizzati per la rappresentazione di input, output e messaggi²².
- *Functional Descriptions*. Descrivono le funzionalità dei servizi, per mezzo di un'interfaccia WSDL e le operazioni annesse²³.
- *Non-Functional Descriptions*. Definiscono i dettagli relativi al provider, all'implementazione e all'ambiente di esecuzione (prezzo, versione, nome, autore, URL, etc). I protocolli usati in questo caso sono numerosi e includono WS-Policy, WS-

²¹ L'XML Schema è il linguaggio di descrizione del contenuto di un file XML. Il suo scopo è delineare quali elementi sono permessi, quali tipi di dati sono ad essi associati e quale relazione gerarchica hanno fra loro gli elementi contenuti in un file XML.

²² Per input si intende ciò che un cliente richiede, per output ciò che riceve.

²³ WSDL è un linguaggio formale in formato XML utilizzato per la descrizione di Web Service. Definisce l'interfaccia pubblica, che specifica le operazioni messe a disposizione dal servizio, il protocollo di comunicazione da utilizzare per accedere al servizio, il formato dei messaggi accettati in input e restituiti in output dal servizio e l'endpoint del servizio, cioè l'indirizzo - in formato URI - a cui reperire il Web Service.

Agreement, WS-Reliability, WS-Security²⁴.

- *Behavioral Descriptions*. Definiscono il comportamento esterno (il protocollo che un cliente deve seguire per usufruire di un servizio) e interno (il modo in cui le funzionalità del servizio sono aggregate tra loro) del servizio. Le specifiche utilizzabili sono WS-CDL, per il primo, WS-BPEL, per il secondo²⁵.
- *Technical Descriptions*. Descrivono i dettagli riguardanti il formato dei messaggi, i protocolli di comunicazione e gli access point del servizio. Per i primi due sono utilizzati WSDL Binding, per l'ultimo WSDL service.

Il compito di realizzare nel concreto la descrizione semantica di un servizio è di competenza del service provider, ma questo comporta due limitazioni:

1. gli utenti sono esclusi dal processo di descrizione del servizio, precludendo la possibilità di inserire la componente esperienza-utente.
2. i service provider sono restii a utilizzare descrizioni semantiche dei loro servizi a causa della complessità degli approcci.

La novità introdotta in [70] è la possibilità, da parte degli utenti, di integrare le descrizioni dei service provider con quelle fornite da loro stessi. Per raggiungere questo obiettivo, è stato introdotto un nuovo elemento per la descrizione semantica dei servizi, chiamato *social contract*. Questa nuova componente tiene conto del modo in cui gli utenti percepiscono il servizio, del perché lo utilizzano e in quali circostanze. Tali metadati si possono derivare in modo esplicito o implicito:

- *Esplicito*, quando i metadati vengono aggiunti dagli utenti, perché vogliono esprimere una descrizione del servizio dal loro punto di vista. Gli utenti possono annotare i servizi, similmente a quanto già fanno per prodotti, contenuti e file multimediali in siti come Amazon, Flickr e YouTube. Gli utenti, quindi, hanno la possibilità di spiegare perché usano un certo servizio, e di aggiungere note riguardanti gli attributi, come gli input e gli output. In alcuni casi, gli utenti possono esprimere il loro grado di soddisfazione riguardo la qualità e la facilità d'uso. In ogni caso, l'espressione di alcuni attributi rimangono di competenza del service provider, come ad esempio le Technical Descriptions. L'approccio proposto da [70] per tali scopi, utilizza dei meccanismi di tagging. L'uso dei tag è semplice e diretto, e gli utenti posseggono una buona familiarità con essi. Si possono utilizzare stringhe libere, cioè non vincolate da un dizionario, oppure termini appartenenti a un insieme predefinito. Di solito i service provider preferiscono l'utilizzo di vocabolari strutturati, mentre gli utenti preferiscono avere maggiore libertà d'azione, e quindi la possibilità di inserire termini svincolati da ogni costrizione lessicografica.
- *Implicito*, quando le informazioni riguardanti i servizi possono essere inferite dall'osservazione dei comportamenti degli utenti mentre interagiscono con essi. Ad

²⁴ Maggiori informazioni relative a questi protocolli sono reperibili ai seguenti indirizzi:

<http://www.w3.org/Submission/WS-Policy/>, <http://www.ogf.org/documents/GFD.107.pdf>, http://docs.oasis-open.org/wsrn/ws-reliability/v1.1/wsrn-ws_reliability-1.1-spec-os.pdf, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

²⁵ Maggiori informazioni relative a WS CDL e WS BPEL sono reperibili ai seguenti indirizzi

<http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/>, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

esempio, se un numero significativo di utenti che utilizza un certo servizio A, subito dopo usufruisce anche di un servizio B, si può dedurre l'informazione che B è un servizio utile a chi utilizza A. Ovviamente il rilevamento di dati impliciti è maggiormente complesso di quello relativo ai dati espliciti e richiede l'impiego di strumenti di data mining, secondo diversi livelli di complessità.

4.4.2 La ricerca e la selezione dei servizi

Come abbiamo già visto nel capitolo 3, la ricerca e la selezione dei servizi è un argomento di estrema attualità. Il problema della ricerca di un servizio è strettamente connesso a quello della sua descrizione e allo stesso tempo a quello della selezione, infatti un servizio deve rispettare le aspettative di chi lo richiede (quindi deve esserci un match semantico tra i termini presenti nella descrizione e quelli nella query di ricerca), e deve possedere delle metriche che ne permettano la distinzione tra omologhi e, quindi, la selezione del migliore. Il tema della descrizione di un servizio è stato affrontato nella precedente sezione; in questa vedremo come effettuare una scelta tra servizi formalmente equivalenti.

La scelta di un servizio è influenzata dalla misura della QoS (Quality of Service), che fornisce una valutazione qualitativa riguardo alle prestazioni tecniche del servizio (come il tempo di risposta di un sistema o la sua affidabilità). Tuttavia, nel momento in cui un utente deve decidere per quale servizio optare, si preoccupa maggiormente della qualità del servizio erogato. Tale metrica viene indicata da [71] con l'espressione Quality of Experience (QoE). Con questa terminologia ci si riferisce alla valutazione della qualità del servizio percepita dagli utenti passati, cioè da coloro che hanno già usufruito del servizio, e che successivamente hanno espresso il grado di apprezzamento riguardo la loro esperienza.

Nel paper *Selecting Web Services Based on Past User Experiences* di P. Leitner et al. [72], gli autori propongono un approccio al rilevamento della QoE basato sull'impiego di feedback strutturati e di tag liberi:

- *Feedback strutturati*. L'utente, dopo aver usufruito di un servizio, esprime una valutazione attraverso un valore che può essere paragonato oggettivamente a quelli forniti da altri utenti, ad esempio attraverso un numero che esprime il grado di soddisfazione su una scala ben definita (rating numerico).
- *Tag liberi*. L'utente, sulla base della propria esperienza diretta, esprime un giudizio attraverso delle parole chiave (singole o composite) senza restrizioni di vocabolario. In tal modo la valutazione del servizio risulta più espressiva, fornendo ai futuri utenti informazioni aggiuntive che non possono essere rappresentate con un semplice numero.

Sempre in [72] troviamo anche una definizione del processo di selezione del servizio, secondo la quale si possono distinguere cinque fasi:

1. Definire le funzionalità che il servizio deve garantire.
2. Definire i vincoli non funzionali
3. Selezionare il servizio migliore in base alla qualità offerta
4. Usare il servizio
5. Fornire un feedback

Si noti che, normalmente, solo i primi tre punti fanno riferimento direttamente alla scelta di un servizio, ma se consideriamo un approccio in cui sono fondamentali i feedback degli utenti, diventa fondamentale comprendere anche gli ultimi due punti. Se così non fosse si andrebbe

incontro al fenomeno del *free riding*, cioè all'utilizzo di risorse senza corrispondere un contributo e lasciando questa incombenza ad altri²⁶.

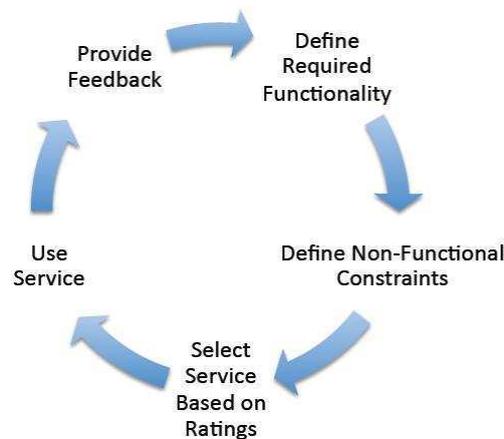


Figura 4.10: il ciclo di selezione di un servizio [72]

Le prime due fasi sono prettamente di ricerca. La prima dipende da com'è stato descritto formalmente il servizio (vedi sezione 4.4.1) e specifica il tipo di servizio richiesto. La seconda serve a scremare i risultati ottenuti al passo precedente, selezionando quelli che soddisfano determinati vincoli, generalmente concernenti la presenza o meno di caratteristiche accessorie, e che vengono espressi tramite i tag liberi. La terza fase riguarda la selezione vera e propria. I feedback strutturati permettono di redigere una classifica qualitativa, che poi può essere esplorata nel dettaglio osservando anche le informazioni aggiuntive fornite dai tag liberi. La quarta fase corrisponde all'utilizzo del servizio e quindi alla formazione dell'esperienza utente. La quinta e ultima fase chiude il cosiddetto ciclo di selezione del servizio; l'utente riporta le sue valutazioni sotto forma di feedback strutturati e tag liberi, agevolando le ricerche future con informazioni aggiuntive.

Conclusioni

In questo capitolo abbiamo visto cos'è la Collective Intelligence, l'abbiamo suddivisa in tre livelli, abbiamo analizzato le sorgenti da cui deriva, abbiamo studiato come può essere formata, e abbiamo spiegato in che modo può essere applicata ai servizi. Nel prossimo capitolo vedremo la proposta di RS basato su Collective Intelligence, studiando alcuni possibili approcci innovativi.

²⁶ Si consideri il caso di Wikipedia. Gli utenti di tale sito leggono le pagine create da altri e allo stesso tempo ne aggiungono di nuove oppure aggiornano quelle esistenti. Chi usufruisce dei contenuti presenti in Wikipedia ma non fornisce mai il proprio contributo è considerato un free rider. Ciò diventa un serio problema quando i free rider sono in numero eccessivo rispetto agli utenti che invece condividono le loro conoscenze. Supponiamo, estremizzando il concetto, che in un certo istante T tutti gli utenti di Wikipedia decidessero di non partecipare più alla creazione di nuove pagine. Nel lungo periodo le informazioni su Wikipedia diventerebbero obsolete e quindi inutili.

Capitolo 5

Proposta di un Recommender System basato su Intelligenza Collettiva

In questo capitolo si presenta una proposta di Recommender System per i servizi, basato sull'intelligenza collettiva. Si precisa fin da subito che non si tratta di una proposta implementativa ma di una progettazione ad alto livello. Tuttavia, basandosi sulle linee guida contenute nella seguente trattazione, si può pensare, come sviluppi futuri, a una effettiva implementazione del RS.

Il sottocapitolo 1 fornisce una panoramica del contesto in cui si inserisce la proposta del Recommender System. Il sottocapitolo 2 introduce il sistema, delineandone l'architettura. Il sottocapitolo 3 è incentrato sulle caratteristiche del sistema. Il sottocapitolo 4 provvede una descrizione del RS, entrando nei dettagli. L'ultimo sottocapitolo descrive possibili estensioni.

5.1 Contesto

Prima di iniziare una progettazione di qualsiasi genere, è fondamentale individuare due elementi: il problema e la soluzione. Spiegando meglio il significato di questi due termini, il problema corrisponde alle esigenze, di uno o più individui, che si vogliono soddisfare; la soluzione, invece, consiste nei mezzi a disposizione del progettista che possono permettere di risolvere il problema.

Il problema, in questo caso può essere riassunto dalla seguente citazione.

“What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.”

H.A. Simon

H. A. Simon¹ [73], quindi, sostiene che al crescere delle informazioni disponibili, diminuisce l'attenzione di chi le riceve, e diventa dunque necessario trovare metodi per indirizzare efficientemente l'interesse tra tutte le sorgenti a disposizione. Questo, come spiegato nel capitolo 3, è proprio il compito affidato a un Recommender System.

¹ Herbert Alexander Simon (1916 – 2001) è stato un economista, uno psicologo e un informatico statunitense. Esperto di Intelligenza artificiale, scienze cognitive e in processi decisionali, ha ricevuto il premio Nobel per l'Economia nel 1975 e il premio Turing nel 1978.

La sovrabbondanza di informazioni, nel contesto considerato, riguarda la mole di servizi, che con l'avvento di Internet si sono moltiplicati e diversificati (vedi capitolo 2). Il problema da risolvere consiste pertanto nel trovare un metodo efficiente per selezionare, tra tutti i servizi disponibili, quelli più adatti a un determinato individuo.

La soluzione proposta in questa tesi è l'utilizzo della Collective Intelligence (vedi capitolo 4), per generare suggerimenti appropriati e completi, in grado di guidare l'utente nella scelta di un servizio. Tra le sorgenti di CI individuate (vedi sezione 4.3.1 *Le sorgenti di Collective Intelligence*) vi è la Social Intelligence, cioè le informazioni che si possono ricavare dal Social Web. All'interno di questo ambito certamente ricopre un ruolo di primario interesse il social network per eccellenza: Facebook. L'aggregazione di contenuti presenti in esso (dati personali, immagini, video, applicazioni, interazioni sociali, etc.) costituiscono una preziosa fonte di informazioni.

Nell'articolo di Leah Betancourt² *How Companies Are Using Your Social Media Data* [74], vengono indagati alcuni degli scopi a cui possono essere destinati i dati ricavabili dal Social Web. Ad esempio, intervistando Jules Polonetsky, direttrice e co-presidente del Future of Privacy Forum³, è emerso che aggregatori come Rapleaf⁴ raccolgono e catalogano informazioni sugli individui, per poi rivenderle alle aziende che vogliono conoscere meglio i loro clienti e scoprire le loro abitudini online. Entità come le compagnie aeree, infatti, possono utilizzare questi dati per trovare nuovi clienti o classificare quelli già esistenti. Società di servizi finanziari come banche e istituti di credito utilizzano gli stessi servizi di datamining per finalità di marketing e per prendere decisioni sulle richieste di prestito.

Un altro aspetto importante di Facebook è il numero di utenti che raggiunge, circa 600 milioni (nell'anno 2010), rendendolo, di fatto, non solo depositario di informazione ma anche un rilevante veicolo di informazione. Un esempio eccellente è la campagna *Inspired by Iceland*⁵ promossa dall'ente del turismo Islandese. In questo caso è stato pensato di utilizzare il social network per veicolare una campagna di marketing turistico. La pagina Facebook non è così una semplice presenza istituzionale ma una community fresca e attiva. Si trovano informazioni utili sempre aggiornate ed i visitatori sono chiamati a partecipare in maniera attiva con contenuti (post, immagini, video e link) e suggerimenti personali.

Ma le potenzialità della Collective Intelligence sono ancora maggiori se si considera che gli strumenti per interagire con Internet nell'ultimo decennio, oltre ad essere sempre più diffusi, sono anche cambiati, includendo dispositivi mobili come telefoni cellulari, smartphone, PDA e tablet. Ciò ha consentito di instaurare un collegamento permanente tra gli individui e la Rete. Donald A. Norman⁶ in una recente intervista sul Corriere della Sera [75] ha affermato che “[...] *Le funzioni del computer verranno sempre più inglobate dentro altri dispositivi, come le cosiddette tavolette. Userà il pc in senso classico solo chi, come gli scrittori o gli ingegneri, non potrà fare a meno dello schermo e della tastiera. Per tutti gli altri, non sarà più così necessario [...]*”. La teoria di Norman sembra essere supportata da alcuni dati statistici riportati di seguito.

In una ricerca della Nielsen Company del 2009 [76] si evidenziava una penetrazione degli smartphone in Italia pari al 28% (più di un italiano su quattro) e negli USA del 32% (circa un americano su 3); in altri paesi, dove la penetrazione registrata è minore, come in Gran Bretagna (solo il 12% della popolazione nel 2009 possedeva uno smartphone), il dato sulla crescita, rispetto ad un anno prima, è comunque significativo (in UK pari al 70%).

² Leah Betancourt è una giornalista esperta in media di nuova generazione, con una particolare passione per i Social Media.

³ <http://www.futureofprivacy.org/>

⁴ <https://www.rapleaf.com/>

⁵ <http://www.facebook.com/inspiredbyiceland>

⁶ Donald Arthur Norman è uno psicologo e ingegnere statunitense. Attualmente professore alla Northwestern University, è anche professore emerito del MIT, ed è stato vicepresidente del gruppo di ricerca sulle tecnologie avanzate della Apple.

Uno studio di eMarketer⁷ [77] prevede per il 2014 una diffusione degli smartphone in Italia del 67%, negli Stati Uniti del 55%, in Gran Bretagna del 32%.

Gartner⁸ e IDC⁹ hanno reso disponibili nello scorso mese di gennaio i propri risultati di sintesi sull'andamento delle vendite di PC per l'anno 2010 [78][79]. Sebbene su base annuale l'incremento sia pari al 13,6% per IDC e al 13,8% per Gartner, nel quarto trimestre si evidenzia una flessione della crescita di vendite, infatti, nell'ultimo periodo del 2010 la crescita si assesta intorno al 3%. Secondo l'analista di IDC David Daoud, a impattare negativamente sulle vendite di PC a livello globale, soprattutto nel corso della seconda metà del 2010, è il successo di mercato delle soluzioni Tablet PC, soprattutto con riferimento all'iPad di Apple. Le vendite di queste soluzioni mobile hanno, in pratica, cannibalizzato il mercato dei sistemi PC tradizionali.

Se gli utenti sono orientati sempre di più verso dispositivi mobili, i Recommender System, per sfruttare al meglio questa opportunità, devono necessariamente seguirli.

5.2 Il sistema SeReS CI

Il Service Recommender System basato su Collective Intelligence (*SeReS CI*) che presentiamo in questo sottocapitolo ha come soggetto delle raccomandazioni i servizi e come "cervello" del Recommender Engine l'intelligenza collettiva.

5.2.1 Un Recommender System per i servizi

La ricerca e la selezione di servizi, come abbiamo visto nel capitolo 2, sono due degli aspetti più importanti e delicati dell'IoS, infatti l'aumento dei servizi disponibili rappresenta un problema rilevante per i consumatori che, trovandosi di fronte a innumerevoli proposte, fanno fatica a scegliere il servizio migliore per le proprie esigenze. La selezione ottimale e personalizzata dei servizi costituisce un'attività che può essere espletata efficientemente da un Recommender System. Nel capitolo 3, sono state elencate le caratteristiche di un RS per i servizi, presentando un esempio per il mercato del Cloud Computing. In questo capitolo, invece, ci si propone di presentare un sistema che sfrutti non solo le preferenze degli utenti, ma anche le conoscenze sui servizi e sui service provider, e più in generale la Collective Intelligence.

5.2.2 Il cervello del Recommender System: l'intelligenza collettiva

Come può essere utilizzata l'intelligenza collettiva per migliorare le performance di un Service Recommender System? La risposta a questa domanda si trova in questa sezione e, in modo più approfondito, nelle prossime.

La prima considerazione da fare è che la Collective Intelligence è una conoscenza composita, ma condivisa, e proprio perché è valida in generale potrebbe sembrare illogico applicarla in un ambito come quello dei Recommender System, dove rivestono un ruolo di notevole importanza le preferenze dei singoli utenti, cioè entità diverse e variegata tra loro. Ciò è vero se si guarda all'intelligenza collettiva come possibile fonte di un insieme di preferenze comuni; tuttavia non è questa la prospettiva corretta.

Nel capitolo 4 abbiamo già visto che la Collective Intelligence può essere utilizzata nell'ambito dei Servizi per la descrizione, la ricerca e la selezione. Nella prossima sezione vedremo altre applicazioni, come ad esempio la definizione della reputazione di un service provider.

⁷ <http://www.emarketer.com/>

⁸ <http://www.gartner.com/>

⁹ <http://www.idc.com/>

Focalizzando l'attenzione sui Recommender System, finora la Collective Intelligence non è stata sfruttata a sufficienza per le potenzialità che offre. Esistono, infatti, numerose possibili applicazioni della CI a questa tipologia di sistemi, come ad esempio la correzioni di suggerimenti sbagliati, l'aggiornamento del dataset, l'associazione di suggerimenti specifici a gruppi di utenti.

La seconda considerazione è che l'intelligenza collettiva è sparsa per la Rete e necessita di regole e strumenti per essere prima raccolta e poi organizzata in modo omogeneo. Per catturare tali conoscenze distribuite è necessario creare un sistema in grado di estrarre informazioni da qualsiasi fonte possibile, una rete capillare di interfacce pronte a raccogliere i dati prodotti, esplicitamente e implicitamente, dagli utenti. L'idea è che se la Collective Intelligence è pressoché ovunque, è necessario strutturare il sistema affinché sia capace di cogliere ogni sua espressione. Da questa base teorica nasce la proposta di un Recommender System che segua l'utente ovunque si trovi, che sia disponibile per dispositivi fissi ma anche per dispositivi mobili, e che offra ben quattro paradigmi di interfacce diversi all'utente per relazionarsi con il sistema: un sito web, un widget per PC, Facebook, e una Mobile App.

5.2.3 Architettura del sistema

L'architettura del sistema prevede tre elementi: i dispositivi di accesso, le interfacce, e SeReS CI, il Recommender System vero e proprio.

I dispositivi di accesso

I dispositivi di accesso sono i mezzi fisici che un utente può utilizzare per accedere al sistema. In sede di progettazione ne sono stati scelti due:

- *dispositivi fissi*. Rientrano in questa categoria PC e notebook, cioè quei dispositivi che permettono all'utente di avere uno schermo sufficientemente grande, una potenza computazionale di discreto livello e che di norma sono utilizzati in condizioni di comodità e tranquillità.
- *dispositivi mobili*. Fanno parte di questa categoria smartphone, palmari e tablet, cioè quel tipo di dispositivi che forniscono prestazioni di basso livello (se confrontate con quelle di un computer medio) e che in genere vengono utilizzati in circostanze di provvisorietà. Il vantaggio di questi dispositivi risiede nella possibilità di poterli utilizzare mentre ci si sposta da un luogo ad un altro.

Le interfacce

Le interfacce per poter accedere a SeReS CI sono quattro, le prime tre sono utilizzabili con un dispositivo fisso, la quarta è destinata a dispositivi mobili:

- *il sito web*. È il canale di accesso principale al Recommender System. Permette una ricerca completa, contenente dettagli sui servizi, immagini, video e altri tipi di contenuti correlati alla ricerca.
- *il widget per il computer*. Se non è disponibile una connessione Internet può essere utile disporre di uno strumento offline che consenta di ottenere i suggerimenti in qualsiasi momento. I dati disponibili, ovviamente, sono limitati nei contenuti, per non appesantire troppo l'applicazione. Inoltre, memorizza informazioni sull'utente e le trasmette al sistema centrale non appena è disponibile una connessione.
- *le applicazioni per Facebook*. Il social network per antonomasia archivia nei suoi server preziose informazioni su oltre 500 milioni di utenti. L'integrazione del sistema con Facebook ha un triplice scopo: far conoscere SeReS CI, promuoverne l'utilizzo tramite gli strumenti del social network, intercettare le informazioni private degli utenti.

- *le app per i dispositivi mobili.* Come abbiamo visto nel capitolo 3, per i Semantic Recommender System, e in particolare per la categoria dei Context-Aware, è importantissimo poter avere accesso a suggerimenti che siano attinenti al contesto ambientale in cui si trova l'utente. Inoltre, le app per dispositivi mobili possono inviare informazioni al RS (in tempo reale se è disponibile una rete wireless, o appena è disponibile una connessione, in caso contrario), favorendo l'archiviazione di dati utili per suggerimenti successivi.

SeReS CI

L'elemento principale del sistema è il Recommender System (SeReS CI). Esso è costituito da cinque componenti:

- *L'ontologia del sistema.* Definisce il vocabolario del sistema, la descrizione degli utenti e dei servizi, la rappresentazione dell'Intelligenza Collettiva e le interazioni tra questi elementi e il recommender engine.
- *Gli utenti.* Sono una delle due risorse fondamentale del sistema. Sono i destinatari dei suggerimenti e il profilo di ciascuno di essi permette, congiuntamente alle interazioni con il Recommender System, di raccogliere informazioni per le proposte future.
- *I servizi.* Sono l'altra risorsa fondamentale del sistema. Sono l'oggetto dei suggerimenti, e per questo devono essere corredate di descrizioni complete, in modo da poter garantire all'utente una scelta consapevole.
- *I Service Provider.* La rappresentazione di un provider e la sua reputazione derivano dalle descrizioni dei servizi che eroga e dalla valutazione delle prestazioni fornite. Tali informazioni sono archiviate e aggregate sotto forma di Collective Intelligence nella *CI platform*.
- *La Collective Intelligence platform.* È il cervello del RS. Contiene le conoscenze legate a utenti, servizi e service provider, in modo ordinato e organizzato. I tre livelli di Intelligenza Collettiva (vedi capitolo 4) consentono la generazione di suggerimenti appropriati, migliorano le descrizioni di utenti e servizi, e determinano la reputazione dei service provider.
- *Il recommender engine.* È il cuore del sistema, genera i suggerimenti sulla base dei dati provenienti da utenti, servizi e dalla Collective Intelligence platform, secondo quanto specificato dagli algoritmi in esso contenuti.

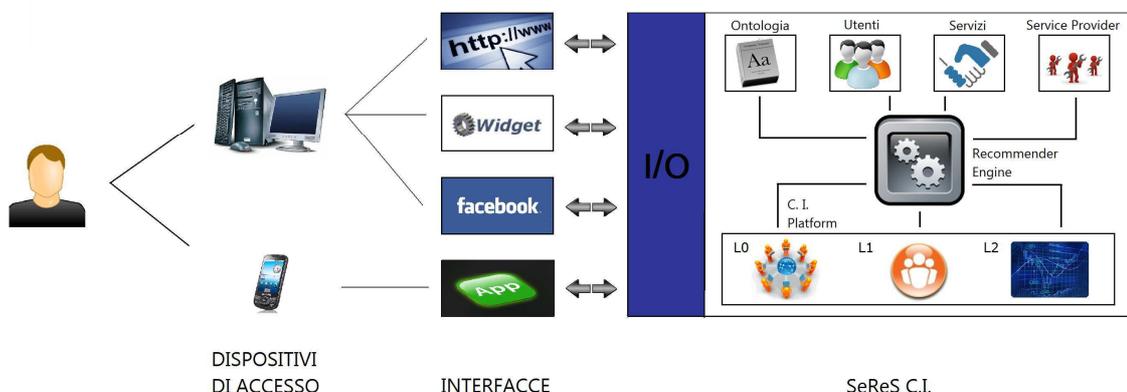


Figura 5.1: Architettura del sistema

5.3 Caratteristiche del sistema

In questo sottocapitolo si espongono le caratteristiche peculiari del sistema progettato, e in particolare le tipologie di utenti, le interfacce del sistema e le prerogative della Collective Intelligence Platform.

5.3.1 Tipologie di utenti

In base alle possibili modalità di interazione con il sistema, sono state previste quattro tipologie di utenti per SeReS CI: i visitatori, i clienti, gli utenti premium, e i superutenti. Accanto a queste quattro categorie di utenti veri e propri si aggiungono due classi atipiche: i collaboratori esterni e i Service Provider.

Visitatori

Un utente è considerato un visitatore quando utilizza il sistema senza fornire le proprie credenziali. Questa possibilità deve essere garantita per tre ragioni:

- consentire un utilizzo immediato senza barriere d'ingresso;
- incuriosire gli utenti “di passaggio” con un *sample*;
- promuovere il sistema anche tra le persone che vogliono mantenere l'anonimato.

I *visitatori*, tuttavia, non possono ricevere suggerimenti personalizzati, e in questo caso ci si affida ai suggerimenti generati dalla Collective Intelligence platform, in grado di definire le preferenze di utenti tipo.

Clienti

Appartengono a questa categoria gli utenti che accedono al sistema dopo essersi registrati e che quindi possiedono un profilo. Le informazioni fornite sono le minime indispensabili per la registrazione (nome utente, indirizzo email e password) ma possono essere aggiornate attraverso successive modifiche dell'utente, o in base alle sue interazioni con il sistema. Un'alternativa alla registrazione è l'accesso tramite il proprio profilo Facebook (il sistema in questo caso entra in possesso delle informazioni personali che l'utente ha nel profilo Facebook).

I *clienti* hanno accesso ai suggerimenti personalizzati, ma il livello di specificità dipende dal livello di completamento del proprio profilo:

- *Basso*. Le informazioni fornite sono solo quelle di base. I suggerimenti risultano piuttosto generici.
- *Medio*. Le informazioni fornite sono complete ma ci sono state poche interazioni con il sistema. I suggerimenti sono mediamente personalizzati.
- *Alto*. Le informazioni fornite sono complete e ci sono state molte interazioni con il sistema. Il livello di personalizzazione dei suggerimenti è molto elevato.

Utenti premium

Un cliente (con un livello di completamento del profilo *Alto*) che abbia acquistato servizi e poi li abbia valutati e recensiti nel sistema diventa un utente premium. Ci sono tre livelli di utenti premium:

1. *Standard*. Utenti premium che hanno valutato al massimo 20 servizi (effettivamente acquistati) negli ultimi 2 anni.
2. *Plus*. Utenti premium che hanno valutato al massimo 50 servizi (effettivamente acquistati) negli ultimi 2 anni.

3. *Deluxe*. Utenti premium che hanno valutato più di 50 servizi (effettivamente acquistati) negli ultimi 2 anni.

Il vantaggio di essere un utente premium non consiste nell'aver suggerimenti molto accurati (questo è già possibile per i *clienti* con un *Alto* livello di completamento del profilo), bensì nell'ottenere vantaggi economici, in cambio dei feedback che producono. Gli utenti premium, infatti, possono ricevere uno sconto sull'acquisto di nuovi servizi, proporzionale al livello di appartenenza (ad esempio, potrebbe essere pari al 5% per quelli *Standard*, 10% per i *Plus*, 20% per i *Deluxe*). Per verificare che la valutazione sia stata eseguita dopo aver effettivamente acquistato il servizio, il service provider rilascia al cliente, all'atto del pagamento, un codice identificativo univoco. Il sistema permette di inserire il giudizio sul servizio solo dopo aver verificato il codice.

Superutenti

Un *superutente* è un utente premium di livello *Deluxe*, le cui valutazioni sono da tenere in considerazione perché simili a quelle di molti utenti, oppure a quelle di un utente in particolare. Si distinguono due tipologie di *superutenti*:

- *superutente del sistema*. Sia X un utente *Deluxe* che ha effettuato N valutazioni. Siano $\{X_1, \dots, X_K\}$ i K utenti del sistema. Supponiamo che il generico utente X_i abbia giudicato S_i suggerimenti contenenti una o più valutazioni di X . Se almeno il 33% dei K utenti del sistema hanno giudicato positivamente almeno la metà degli S_i suggerimenti, allora X è un *superutente del sistema*.
- *superutente relativo*. Sia X un utente *Deluxe* che ha effettuato N valutazioni. Sia Y un utente del sistema. Supponiamo che Y abbia valutato S suggerimenti contenenti una o più valutazioni di X . Se Y ha giudicato positivamente almeno la metà degli S suggerimenti, allora X è un *superutente di Y* ¹⁰.

Un *superutente* non ha particolari benefici. Il sistema, però, lo tiene in considerazione perché i suoi giudizi sono significativi e possono essere utili per consigli generici ai visitatori.

I collaboratori esterni

Una tipologia di utente atipica presente nel sistema è quella dei *collaboratori esterni*. Questa categoria è anomala perché non è formata da fruitori del RS veri e propri (anche se non è escluso che lo siano), bensì da persone che accettano di collaborare alla formazione di Collective Intelligence di livello 0 attraverso contributi personali di vario genere (descrizioni, recensioni, suggerimenti, ricerche di mercato, etc.), in cambio di un compenso economico. È fondamentale, in questo caso, verificare con attenzione l'identità di tali persone e monitorare le loro azioni, affinché non introducano nel sistema informazioni errate o inesatte. La modalità di impiego di queste persone è una via di mezzo tra quelle dei casi Wikipedia e Innocentive, visti nel capitolo 4. La remunerazione prevista dipende dalla validità dei singoli contributi¹¹.

All'interno di questa categoria viene selezionato un ristretto sottoinsieme di persone affidabili e competenti, le quali costituiscono l'*authority* del sistema, cioè una entità garante, responsabile del corretto funzionamento del RS, dei controlli sulle attività degli utenti, e anche degli altri collaboratori.

¹⁰ Si noti che essere un *superutente relativo* non è riflessivo, infatti, se X è un *superutente di Y* , X è un utente *Deluxe* con metro di giudizio simile a Y . Certamente è vero che Y ha gusti simili a X , però questo non significa che Y sia necessariamente un utente *Deluxe*.

¹¹ La validità di un contributo dipende dal numero di suggerimenti che contengono quell'informazione e sono ritenuti utili dagli utenti

I Service Provider

Anche i Service Provider possiedono un profilo utente, sebbene sia un account particolare, destinato non tanto all'utilizzo del RS, bensì all'immissione nel sistema di informazioni e aggiornamenti sui servizi da loro erogati.

5.3.2 Il Web Site

È il canale di accesso principale al Recommender System, l'elemento che forgia la prima impressione dell'utente e quindi deve fornire un'idea di semplicità e allo stesso tempo di serietà. L'interfaccia grafica della homepage deve essere intuitiva e con pochi elementi essenziali, possibilmente solo due: un campo testo per specificare il tipo di servizio a cui si è interessati e un pulsante per il login. La scelta di una pagina iniziale così spartana vuole privilegiare l'immediatezza e la facilità di utilizzo, così da poter consentire l'interazione con il sistema anche ai meno esperti.

La registrazione al sito

Sebbene si possa utilizzare il Recommender System anche senza fornire le credenziali, come detto nella sezione precedente, per avere suggerimenti personalizzati è necessario accedere al sistema. L'accesso può essere eseguito in tre modi:

1. *Credenziali Facebook.* La modalità più rapida e semplice per entrare nel sistema è utilizzare il proprio profilo Facebook. Scegliendo questa possibilità si permette al sistema di accedere alle informazioni personali registrate nel Social Network.
2. *Registrazione.* Se non si possiede un profilo e nemmeno un account Facebook, ci si può registrare al sito fornendo tre informazioni basilari: un nome utente, un indirizzo email e una password.
3. *Login.* Se si possiede già un account di SeReS CI, per entrare nel sistema si utilizzano il nome utente e la password forniti in fase di registrazione.

La personalizzazione del profilo

Se la modalità scelta per accedere al sito è stata la registrazione standard (nome utente, email e password), il sistema non ha ancora nessuna informazione personale sull'utente e non può fornirgli alcun suggerimento specifico. Se l'accesso avviene tramite account Facebook, di solito si hanno a disposizione molte informazioni sull'utente, ma non è detto che queste siano rilevanti per la formulazione di un suggerimento per un particolare servizio.

Subito dopo la registrazione, quindi, è necessario proporre un test (facoltativo) che permetta al sistema di conoscere alcune informazioni utili per generare consigli in modo appropriato. Tale test deve essere accurato, però, allo stesso tempo, deve risultare piacevole e non durare troppo, altrimenti l'utente è portato a non completarlo.

La prima fase del test chiede all'utente di specificare a quale delle 11 classi di servizi (vedi capitolo 2) è interessato. È ragionevole pensare che un utente possa essere interessato a più di una classe, tuttavia si suppone che ad ogni utilizzo del sistema la tipologia di servizio desiderata sia una sola.

La seconda fase dipende dalla classe selezionata al passo precedente, infatti vengono fatti valutare ("Utilizzerei questo servizio/Non utilizzerei questo servizio") 6 servizi della tipologia scelta, fornendo all'utente una accurata descrizione del servizio stesso, le recensioni di altri utenti, e le reputazioni dei service provider.

La terza fase, infine, invita l'utente a specificare il livello di importanza delle feature associate alla classe del servizio, per capire quali caratteristiche sono rilevanti nella scelte che normalmente compie.

A questo punto il sistema ha elementi sufficienti per fornire suggerimenti discretamente personalizzati all'utente. Maggiori saranno le interazioni dell'utente con il sistema e più accurate saranno le proposte specifiche future.



Figura 5.2: ipotesi grafica della Homepage del sito web SeReS CI

Il primo suggerimento

Indipendentemente dal fatto che l'utente abbia eseguito l'accesso oppure no (questo passaggio influisce soltanto sulla qualità del suggerimento), la prima azione che deve compiere l'utente per ottenere i suggerimenti relativi a un certo tipo di servizio è scegliere la classe di appartenenza. Il sistema, per facilitare il compito dell'utente, permette di digitare nel campo testo sia la classe (es. classe 43 – Servizi di ristorazione e alloggi temporanei), sia il nome di un servizio generico (es. Viaggi), fornendo anche una funzione di completamento automatico. Nel primo caso l'utente viene dirottato immediatamente nella pagina specifica (es. *SeReS CI 43*).

Nel secondo caso, invece, prima di raggiungere la pagina specifica, viene richiesto di selezionare la categoria di appartenenza tra quelle possibili (es. classe 39 o classe 43) per evitare ambiguità (Viaggi nella *classe 39 – trasporti e organizzazione di viaggi* oppure nella *classe 43 – alloggi temporanei*), fornendo anche informazioni aggiuntive per scegliere la classe corretta (la classe 39 comprende i servizi consistenti in INFORMAZIONI relative ai viaggi, le tariffe, gli orari, fornite da agenzie turistiche o da mediatori; la classe 43 comprende i servizi relativi alla PRENOTAZIONE di camere d'albergo da parte di agenzie di viaggio o di mediatori).

La pagina specifica possiede caratteristiche diverse in base alla tipologia di servizi che deve consigliare. La struttura di base, comunque, è la presenza di sei suggerimenti (in primo piano)

e di alcuni strumenti per raffinare la ricerca (a margine). Le sei proposte sono scelte dal sistema, distinguendo se l'utente ha eseguito l'accesso oppure no.

- *Ha eseguito l'accesso.* Il RS utilizza le conoscenze sulle preferenze dell'utente per suggerirgli i servizi più adeguati;
- *Non ha eseguito l'accesso.* Il Recommender Engine ricorre alla Collective Intelligence Platform e, combinando le conoscenze collettive con i profili degli utenti, individua, attraverso un algoritmo di clustering, tre modelli di utente. Per ogni tipologia propone due suggerimenti specifici.

Il raffinamento della ricerca consente di impostare esplicitamente dei vincoli addizionali. Ogni parametro definito dall'utente viene memorizzato dal sistema nel profilo e riproposto all'accesso successivo. Si noti che in questo modo chi non ha eseguito l'accesso ottiene un profilo implicitamente. Per associare l'utente al profilo (anonimo) creato, senza fornire le credenziali, si utilizza un file cookie. Sorge un problema qualora l'utente utilizzi più computer. In tal caso, a seconda del PC che utilizza, il profilo sarà diverso.

Accessi successivi

Dopo il primo utilizzo, ogni volta che l'utente torna nel sito, la prima pagina che gli si presenta non è più la Homepage bensì la pagina specifica del servizio che, fino a quel momento, ha cercato più spesso. Affinché ciò sia possibile (sia per gli utenti registrati sia per quelli che non hanno voluto fornire le credenziali), si ricorre al sopracitato file cookie.

5.3.3 Il Widget

Il *SeReS CI Widget* è un software per computer che permette di utilizzare il sistema offline, con capacità ridotte, oppure online, per accedere alle funzionalità principali. Inoltre, consente di interagire con il sistema (online), scaricando dati e suggerimenti o, viceversa, caricando dati di vario genere.

Funzioni offline

Quando non è disponibile una connessione a Internet, il Widget consente di utilizzare alcune funzionalità del sistema. In particolare permette di:

- *Rivedere i suggerimenti recenti.* Il Widget memorizza i suggerimenti ricevuti dall'utente nell'ultimo mese.
- *Valutare i suggerimenti.* I suggerimenti, soprattutto quando si hanno poche informazioni su un utente, possono non essere appropriati. Segnalando l'inadeguatezza di un suggerimento, ma anche l'opposto, il sistema può apprendere (implicitamente) i gusti dell'utente.
- *Aggiornare il proprio profilo.* Si può caricare un'immagine, specificare interessi personali, modificare i dati immessi in precedenza.
- *Valutare un servizio acquistato.* Dopo aver usufruito di un servizio si possono aggiungere, tramite il Widget, la propria recensione, il rating delle feature, immagini e video, ma anche contribuire con una descrizione personale del servizio stesso.

L'insieme delle azioni eseguite offline, non appena è disponibile una connessione alla Rete, vengono trasmesse al sistema centrale, che aggiorna le informazioni relative ai profili, e ai servizi.

Funzioni online

Il Widget connesso ad Internet dispone di maggiori funzionalità, che si vanno ad aggiungere a quelle offline. Oltre alla citata trasmissione dei dati al sistema centrale, consente anche di:

- *Ricevere suggerimenti.* L'utente riceve direttamente sul Widget i suggerimenti personalizzati, gli stessi che può visionare anche nel sito. L'aggiornamento è costante e tiene conto in tempo reale dei cambiamenti che l'utente apporta (implicitamente o esplicitamente) al suo profilo.
- *Utilizzare una chat per comunicare con altri clienti.* Spesso è utile confrontarsi con gli utenti che hanno già utilizzato un servizio per chiedere loro informazioni e porre domande specifiche. Questo è possibile tramite la funzione chat (disponibile per chi scarica il Widget).

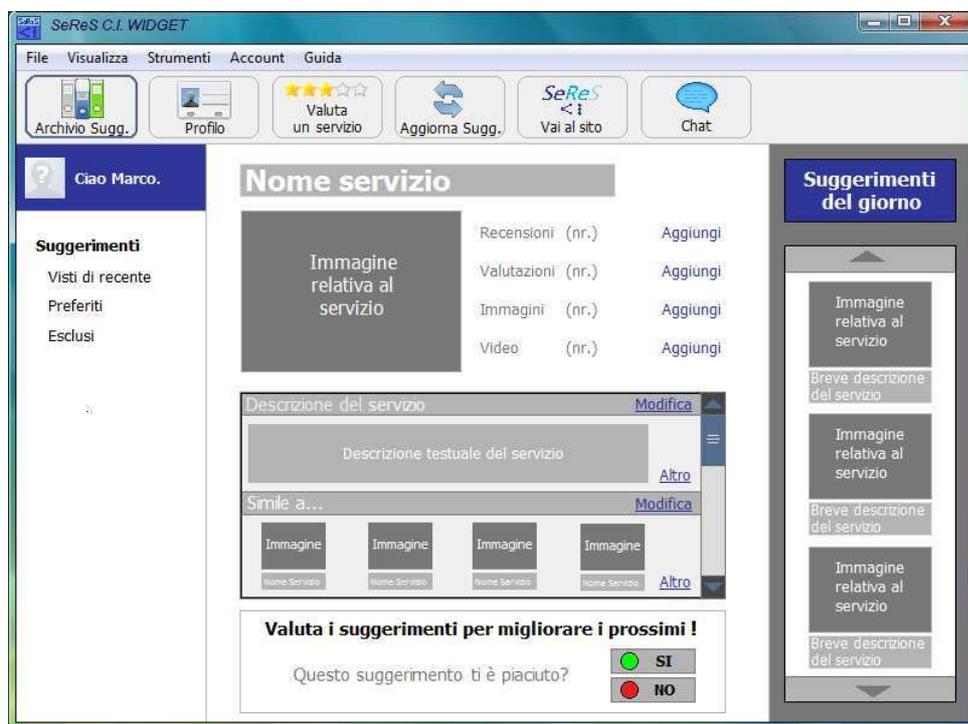


Figura 5.3: ipotesi grafica del SeReS CI Widget

5.3.4 Facebook

SeReS CI è stato pensato per interoperare con Facebook. L'inserimento di applicazioni connesse al Recommender System all'interno del Social Network consente di far conoscere il sistema in breve tempo e senza particolari sforzi, di veicolare i suggerimenti attraverso le pagine, e di raccogliere informazioni utili sotto forma di CI. Le applicazioni all'interno di Facebook sono di tre tipi, ciascuna con funzioni specifiche:

- *SeReS CI App.* È la versione Facebook del Recommender System. Sostanzialmente si accede al sistema attraverso il Social Network. Si tratta, quindi, di un'ulteriore possibilità di accesso al RS, senza togliere né aggiungere alcuna funzionalità.
- *SeReS CI Ad.* Sono dei suggerimenti personalizzati inseriti nelle pagine personali di quel sottoinsieme di utenti Facebook che sono anche fruitori del Recommender System.

- *Collective Intelligence Agents* (CIA). Sono delle applicazioni a scopo ludico di vario genere (Giochi, Test, Quiz, Video), che consentono l'accesso ai dati personali di utenti Facebook (e quindi non solo degli utenti del RS). Dalla raccolta di tali dati, in forma anonima, il RS genera una forma di conoscenza identificabile con una Collective Intelligence di livello 2.

I Collective Intelligence Agent

L'idea di utilizzare delle applicazioni ludiche per accedere ai dati sensibili della popolazione di Facebook proviene da un dato, non tanto sorprendente, riguardante il tipo di app presenti nel social network: su 23160 applicazioni totali, ben 9609 sono per puro divertimento (quiz, test e simili), e altre 2500 sono giochi. Circa la metà delle applicazioni di Facebook, quindi, sono a scopo ricreativo, testimoniando il successo che questo settore riscuote tra gli utenti.

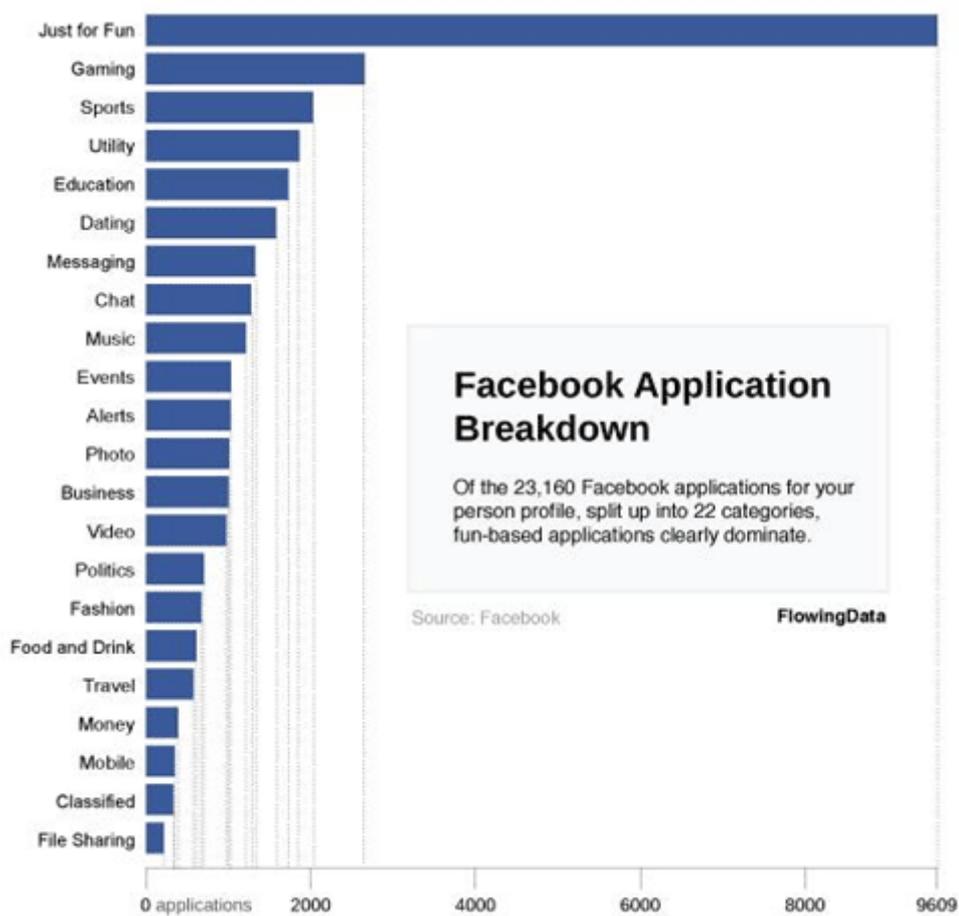


Figura 5.4: le applicazioni di Facebook divise per tipologia [80]

Un altro dato interessante sono le informazioni alle quali hanno accesso tali applicazioni, che all'apparenza possono sembrare solo passatempi innocui. Infatti, ogni volta che si utilizza una nuova app, gli si permette di disporre come meglio crede dei propri dati personali, quali informazioni di base (nome, sesso, reti, lista degli amici, etc.), informazioni del profilo (preferenze, data di nascita, informazioni lavorative, orientamento politico, etc.), informazioni di contatto (rilevamento presenza online), relazioni (parenti e situazione sentimentale), foto e video (qualsiasi immagine o filmato caricato dall'utente), e persino informazioni sugli amici (tutto ciò che un utente può vedere dei suoi amici).



Figura 5.5: le informazioni a cui hanno accesso le applicazioni di Facebook

5.3.5 Mobile App

La diffusione di dispositivi mobili descritta in apertura di capitolo, si traduce nell'ambito dei RS in una importante occasione offerta dal settore *Mobile* nell'intercettare i bisogni dei clienti, e poter soddisfare adeguatamente le loro esigenze. La SeReS CI App è un'applicazione per smartphone e tablet pensata per raggiungere tre obiettivi principali:

- *Agevolare l'utilizzo del sistema ovunque ci si trovi.* Soprattutto nel caso degli smartphone, lo schermo di limitate dimensioni può creare problemi nel visualizzare suggerimenti e dati relativi ai servizi. Una possibile soluzione potrebbe essere la creazione di una versione Mobile del sito Web, ma l'utilizzo richiederebbe la disponibilità di una connessione ad Internet, e gli hotspot WiFi hanno ancora una diffusione troppo marginale. Una App, invece, permette di utilizzare molte funzionalità del sistema anche offline, similmente a quanto descritto per il Widget.
- *Caricare Immagini e Video nel sistema.* Se un servizio può essere fotografato, come nel caso di un hotel o un ristorante, si può scattare una foto con lo smartphone e associare l'immagine alla App, che provvede a memorizzare automaticamente informazioni contestuali, ad esempio il luogo tramite le coordinate del GPS, o l'orario in base all'orologio interno. L'utente inoltre può aggiungere subito il rating o altre annotazioni. Una volta connesso il dispositivo mobile al PC, avviene la sincronizzazione dei dati con il Widget e l'utente può inserire descrizioni e valutazioni del servizio comodamente dalla tastiera.
- *Fornire suggerimenti contestuali.* Se è disponibile un hotspot WiFi si possono ricevere suggerimenti personalizzati in tempo reale contestuali alla posizione e all'orario. Il GPS individua la posizione geografica dell'utente e il sistema propone dei servizi che possono essergli utili, in base alle informazioni che possiede. Se non è disponibile una connessione ad Internet, un'alternativa è che l'utente scarichi nella memoria del suo dispositivo i dati dei servizi di interesse presenti nella zona di destinazione, prima di uscire di casa. I

dettagli dei suggerimenti in questo caso sono limitati, per evitare di occupare troppo spazio.

- *Acquisire informazioni sull'utente.* Come il Widget per il computer, la App per dispositivi mobili monitora le interazioni dell'utente con il sistema e trasmette tutti i dati al Recommender System non appena possibile. Le informazioni interessanti possono essere le più svariate: dagli itinerari seguiti, ai suggerimenti ritenuti utili, dall'acquisto di un servizio, alla valutazione che ne segue.

5.3.6 La Collective Intelligence Platform

La Collective Intelligence Platform costituisce il cervello del Recommender System, infatti, monitora tutte le informazioni associabili ai servizi e le archivia in modo ordinato e organizzato, per poi poterle esibire sotto forma di conoscenza al Recommender Engine. Secondo la proposta del capitolo 4, esistono 3 livelli di Collective Intelligence e ciascuno di essi può essere sfruttato per ottenere obiettivi specifici.



Figura 5.6: Le tre componenti della Collective Intelligence Platform

CI di livello 0

La Collective Intelligence di livello 0 corrisponde alla conoscenza creata con la consapevolezza dei singoli individui che contribuiscono alla sua composizione, e che hanno una visione d'insieme del progetto. Con questo livello di CI è possibile:

- *Descrivere i servizi.* La descrizione di un servizio può essere vista come una prerogativa del service provider che eroga il servizio, però diventa più completa e dettagliata se si permette agli utenti di inserire descrizioni proprie, controllate da un'autorità garante centrale, analogamente a quanto è possibile fare con Wikipedia.
- *Recensire i servizi.* La descrizione di un servizio offre una visione oggettiva di ciò che offre un service provider, tuttavia l'esperienza del cliente è una questione totalmente soggettiva e contingente. L'informazione riguardante un servizio, quindi, risulta più esauriente se è corredata di recensioni che raccontino le esperienze individuali degli utenti.
- *Aggiornare il dataset.* Uno delle esigenze dei Recommender System è mantenere aggiornato il proprio dataset. Alcuni service provider modificano il proprio servizio o non lo erogano più, altri entrano nel mercato fornendo nuovi servizi. È difficile rimanere al passo con la dinamicità del settore dei servizi, quindi, un modo per evitare obsolescenze, è consentire che siano gli utenti stessi ad apportare le modifiche. Anche in questo caso è necessario monitorare e regolare le operazioni di aggiornamento per intercettare errori, volontari o involontari che siano.

CI di livello 1

La Collective Intelligence di livello 1 deriva da dati prodotti in modo consapevole, ma con gli attori che hanno una visione parziale della situazione e quindi non conoscono il risultato finale. Con la CI di livello 1 si possono:

- *Valutare i servizi.* Ogni servizio è definito da un insieme di feature che ne descrivono le proprietà. La valutazione di un servizio, quindi, è l'espressione numerica della qualità di ciascuna feature. I rating dei clienti, singolarmente rappresentano un giudizio più o meno negativo legato alla loro esperienza personale; messi insieme, invece, diventano una valutazione attendibile del servizio.
- *Definire la reputazione dei service provider.* Nel mondo dei servizi riveste particolare importanza la reputazione di chi eroga il servizio. Se il service provider ha una cattiva fama è molto probabile che il cliente sia portato a non sceglierlo. In un Recommender System per i servizi una simile componente non può essere ignorata nella formulazione dei suggerimenti e l'utente, all'occorrenza, deve poter rintracciare tale informazione. La reputazione di un service provider è data da due elementi: la qualità dei servizi che eroga (possono essere uno solo o più di uno) e il rapporto che ha con la clientela. La reputazione si ottiene, quindi, integrando i giudizi forniti dai singoli utenti riguardo ai servizi e all'esperienza con il provider.
- *Rappresentare semanticamente i servizi.* Un servizio può essere descritto semanticamente da un insieme di tag. Ricorrendo al vector space model, si possono confrontare i servizi sulla base della similarità dei vettori a loro associati. I tag che compongono questi vettori sono inseriti dagli utenti, ma poi vengono stemmizzati e normalizzati, ad esempio eliminando i sinonimi, per evitare di estendere eccessivamente il vocabolario del sistema.
- *Determinare i servizi di maggiore successo.* Un Recommender System deve fornire suggerimenti personalizzati, tuttavia anche il mondo dei servizi non si sottrae al concetto di moda. Se un servizio è particolarmente utilizzato e richiesto, può essere ritenuto un valido suggerimento anche se non rispecchia le prerogative dell'utente. Raccogliendo e classificando le preferenze degli utenti è possibile stilare una graduatoria dei servizi più in voga.

CI di livello 2

In questo caso la Collective Intelligence scaturisce dai dati prodotti in modo implicito dagli utenti di applicazioni e servizi, senza che questi siano consci di fornirla. Le opportunità offerte dalla CI di livello 2 sono:

- *La definizione delle preferenze di utenti tipo.* A partire dall'insieme delle rappresentazioni degli utenti, si possono derivare attraverso un algoritmo di clustering 3 tipologie di utenti. Come visto in sede di descrizione del sistema, non sempre il RS dispone di elementi sufficienti per conoscere le preferenze di un utente (ad esempio, nel caso dei *Visitatori*). In tali casi può essere utile, per agevolare il compito del Recommender System, fare riferimento a profili standard, così da poter proporre suggerimenti secondo le preferenze più comuni.
- *Il completamento dei profili utente.* In fase di registrazione un utente fornisce, a livello di dati personali, soltanto il proprio indirizzo email. Successivamente aggiorna il profilo e più informazioni inserisce, maggiore è l'appropriatezza dei suggerimenti che riceverà. Tuttavia, anche qualora non abbia completato la compilazione delle informazioni che lo riguardano, e abbia effettuato solo poche interazioni con il sistema, comunemente si aspetta una risposta efficace dal Recommender System. I dati mancanti possono essere completati facendo un confronto tra il suo profilo e quello degli altri utenti. Selezionando i profili più simili al suo (*neighbor profile*), e scegliendo le caratteristiche più comuni all'interno di questo insieme, il sistema completa provvisoriamente le informazioni mancanti. Ovviamente tali dati sono utilizzabili solo dal Recommender System al fine di produrre i suggerimenti e alla fine dell'interazione dell'utente con il sistema vengono scartati.

- *L'associazione di uno o più servizi a gruppi di persone.* Sulla base delle caratteristiche degli utenti, raccolte tramite sito Web, Widget, CIA e App, è possibile ottenere una descrizione a tutto tondo degli utenti e raggrupparli secondo uno o più parametri. Analizzando i servizi più diffusi all'interno di una categoria si possono produrre suggerimenti "di gruppo" che non derivano più dalle preferenze personali, bensì da quelle collettive.
- *Rappresentazione semanticamente dei servizi.* Come detto, un servizio può essere descritto semanticamente da un insieme di tag. Questo tipo di descrizione può essere esplicita (CI di livello 1) oppure derivata automaticamente dalle recensioni degli utenti, estrapolando le parole chiave già presenti nel vocabolario di sistema.

Di seguito si riporta una tabella riassuntiva, nella quale si specifica anche la sorgente, sia per tipo di utente (vedi sezione 5.3.1), che per tipo di fonte .

Livello di CI	Applicazione	Sorgente	Componente Motivazionale
Livello 0	Descrivere i servizi	Service Provider (PI)	Ric. economica (pubblicità)
		Clienti (PI)	Passione
		Utenti premium (PI)	Ric. economica (sconto)
		Superutenti (PI)	Ric. economica (sconto)
		Collaboratori esterni (PI)	Ric. economica (retribuzione)
	Recensire i servizi	Utenti premium (MeI)	Ric. economica (sconto)
		Superutenti (MeI)	Ric. economica (sconto)
	Aggiornare Dataset	Service Provider (PI)	Ric. economica (pubblicità)
		Clienti (MeI)	Passione
		Utenti premium (MeI)	Passione
Superutenti (MeI)		Passione	
Collab. esterni (PI)		Ric. economica (retribuzione)	
Livello 1	Valutare i servizi	Utenti premium (MaI)	Ric. economica (sconto)
		Superutenti (MaI)	Ric. economica (sconto)
	Definire reputazione del service provider	Utenti premium (MaI)	Ric. economica (sconto)
		Superutenti (MaI)	Ric. economica (sconto)
	Rappresentare semanticamente i servizi	Service Provider (PI)	Ric. economica (pubblicità)
		Clienti (MaI)	Passione
		Utenti premium (MaI)	Ric. economica (sconto)
		Superutenti (MaI)	Ric. economica (sconto)
		Collab. esterni (PI)	Ric. economica (retribuzione)
	Determinare i servizi di maggiore successo	Utenti premium (SI)	Ric. economica (sconto)
Superutenti (SI)		Ric. economica (sconto)	
Livello 2	Definire utenti tipo	Clienti (SI)	COLLABORAZIONE INCONSAPEVOLE
		Utenti premium (SI)	
		Superutenti (SI)	
	Completare i profili	Clienti (SI)	COLLABORAZIONE INCONSAPEVOLE
		Utenti premium (SI)	
		Superutenti (SI)	
	Associare servizi a gruppi di utenti	Clienti (SI)	COLLABORAZIONE INCONSAPEVOLE
		Utenti premium (SI)	
		Superutenti (SI)	
	Rappresentare semanticamente i servizi	Clienti (MeI)	COLLABORAZIONE INCONSAPEVOLE
Utenti premium (MeI)			
Superutenti (MeI)			

Tabella 5.1: le funzioni della Collective Intelligence Platform nel sistema

Il significato delle sigle usate nella precedente tabella è specificato di seguito:

Personal Intelligence - PI, Media Intelligence - MeI, Mass Intelligence - MaI, Social Intelligence – SI, e Organizational Intelligence - OI; Per maggiori chiarimenti si faccia riferimento alla sezione 4.3.1 *Le sorgenti di CI*.

5.4 Descrizione del Recommender System

In questo sottocapitolo si descrivono le componenti del Recommender System, specificandone la struttura e le funzioni. Non si tratta, tuttavia, di una proposta implementativa, bensì di una progettazione ad alto livello del sistema, elaborata con il solo scopo di consentire una migliore comprensione dei contenuti teorici, esposti in precedenza.

Le prime sezioni riguardano i modelli scelti per rappresentare gli utenti, i servizi, i Service Provider e i vocabolari di sistema. Nelle successive, invece, si descrivono le funzionalità della Collective Intelligence Platform e del Recommender Engine. Infine, si espongono le strategie che il RS adotta per presentare i suggerimenti in modo personalizzato, adattandosi, di volta in volta, alle caratteristiche dell'utente.

5.4.1 Un Vector Space Model Esteso

Il Vector Space Model (detto anche term vector model) è un modello algebrico utilizzato per rappresentare un oggetto (ad esempio un documento di testo) tramite vettori di identificatori, ad esempio parole chiave [81]. Di solito è impiegato nell'ambito dell'information filtering, dell'information retrieval, dell'indicizzazione e delle classificazioni.

In questo contesto si definisce un Vector Space Model Esteso, un modello in cui i vettori non sono composti solo da identificatori, ma anche e soprattutto da strutture dati come array e liste e da elementi tipizzati. In particolare, utenti e servizi sono rappresentati come vettori e per ciascuna categoria sono definite funzioni di confronto specifiche (vedi le funzioni *HowSimilarService* e *HowSimilarUser* della sezione 5.4.8).

Le convenzioni usate per rappresentare i tipi di dati contenuti negli elementi dei vettori sono le seguenti:

- *array[]*: indica una lista di termini o di strutture, che può essere ordinata oppure no.
- *array[A,B]*: indica una lista in cui ogni termine è composto da due elementi.
- *array[array[], A, B]*: indica una lista in cui ogni termine è composto da tre elementi e il primo di questi è a sua volta una lista.
- *array[][]*: indica una tabella di due colonne e un numero non specificato di righe.
- *string*: indica un testo composto da caratteri (alfabeto, numeri e simboli).
- *int*: indica un numero intero, ma per semplicità di notazione anche decimale.
- *struct*: indica una struttura specifica di cui è necessario dichiarare il formato.
- *struct data*: è la struttura utilizzata per indicare una data e ha il formato gg/mm/aaaa.
- *struct context*: è la struttura utilizzata per indicare il contesto di acquisto di un servizio ed è formata da due campi, coordinate GPS e orario. Il formato è [(latitudine, longitudine), hh : mm]

Si precisa, altresì, che i tipi di dati e le strutture specificati non sono da intendersi come scelte implementative vincolanti, bensì esempi a scopo dimostrativo, per aiutare il lettore nella comprensione dell'approccio proposto.

5.4.2 I dataset degli Utenti, dei Servizi e dei Service Provider

Gli utenti del sistema, i servizi oggetto dei suggerimenti e i service provider che li erogano, sono memorizzati sotto forma di vettori in tre dataset specifici, ciascun gruppo con una struttura vettoriale propria. La struttura di un utente è definita nella sezione 5.4.4, quella dei servizi nella sezione 5.4.5, mentre la struttura dei Service Provider è descritta nella sezione 5.4.6.

5.4.3 Vocabolari

Un vocabolario, in questo contesto, è una struttura che memorizza i termini usati all'interno di un sistema, definendone la struttura, i tipi di dati ed eventuali sinonimi che possono essere ricondotti a un unico termine principale. Un vocabolario può essere esteso aggiungendo nuovi termini, ma a condizione che rispettino le regole definite per esso.

Per ogni dataset (Utenti, Servizi e Service Provider) sono stati ideati più vocabolari, per limitare i termini che possono essere inseriti in un certo elemento del vettore. A questi tre è stato aggiunto un vocabolario per i suggerimenti (*voc_TI_SU*) che consente di specificare come è stato generato (*User Based, Content Based, Group Based, Superuser Based, Popularity Based, Trust Based* o *Context Based*. Vedi sezione 5.4.10)

Vocabolari per il dataset Utenti

Nel sistema sono stati previsti dieci vocabolari per il dataset Utenti. Facendo riferimento alla figura 5.7, *Nomi propri, Cognomi, Provincia di residenza e Paese di residenza* non hanno bisogno di spiegazioni. *Sesso* permette i due valori Maschio e Femmina. *Lingua Madre* consente di inserire una delle 100 lingue più parlate al mondo. *Istruzione e Lavoro, Sport, Arte e Intrattenimento, Attività e Interessi* fanno riferimento ai rispettivi database di Facebook. La presenza di una seconda colonna tiene in considerazione la possibilità di inserire sinonimi. Dovesse esserci più di un sinonimo, la seconda colonna diventa (per ogni riga) una lista.

Nome Vocabolario	Sigla	Struttura
Vocabolario Nomi propri	voc_NO	<i>array[string]</i>
Vocabolario Cognomi	voc_CO	<i>array[string]</i>
Vocabolario Sesso	voc_SE	{F, M}
Vocabolario Lingua Madre	voc_LI_MA	<i>array[string]</i>
Vocabolario Provincia di residenza	voc_PR_RE	<i>array[string]</i>
Vocabolario Paese di residenza	voc_PA_RE	<i>array[string]</i>
Vocabolario Istruzione e Lavoro	voc_IS_LA	<i>array[string][string]</i>
Vocabolario Sport	voc_SP	<i>array[string][string]</i>
Vocabolario Arte e Intrattenimento	voc_AR_IN	<i>array[string][string]</i>
Vocabolario Attività e Interessi	voc_AT_IN	<i>array[string][string]</i>

Figura 5.7: Vocabolari per il dataset degli utenti

Vocabolari per il dataset Servizi

Per il dataset Servizi sono stati pensati cinque vocabolari. *Proprietà Servizi* contiene le feature associabili a un servizio. *Service Provider* fornisce un elenco degli identificativi dei Service Provider che erogano servizi presenti nel sistema. *Categoria Servizi* specifica le tipologie di servizi oggetto dei suggerimenti. *Descrizione Semantica* racchiude tutti i tag associati ai servizi; nella prima colonna c'è la parola chiave principale, nella seconda la lista di eventuali sinonimi. Infine, *Stato Servizio* comprende i termini *attivo* e *inattivo*, utili per segnalare se il servizio può essere suggerito oppure no.

Nome Vocabolario	Sigla	Struttura
Vocabolario Proprietà Servizi	voc_PR_SE	<i>array[string]</i>
Vocabolario Service Provider	voc_SE_PR	<i>array[ID_Provider]</i>
Vocabolario Categoria Servizi	voc_CA_SE	<i>array[string]</i>
Vocabolario Descrizione Semantica	voc_DE_SE	<i>array[string][string]</i>
Vocabolario Stato Servizio	voc_ST_SE	{attivo, inattivo}

Figura 5.8: Vocabolari per il dataset dei servizi

Vocabolari per il dataset Service Provider

I vocabolari per il dataset Service Provider sono solo due: *Tipologia Provider* e *Stato Provider*. Il primo permette di specificare se un Provider eroga servizi pubblici o privati. Il secondo comprende i termini *attivo* e *inattivo*, utili per segnalare se il Service Provider eroga ancora servizi oppure no.

Nome Vocabolario	Sigla	Struttura
Vocabolario Tipologia Provider	voc_TI_PR	{pubblico, privato}
Vocabolario Stato Provider	voc_ST_PR	{attivo, inattivo}

Figura 5.9: Vocabolari per il dataset dei Service Provider

5.4.4 Rappresentazione degli utenti

Il vettore che rappresenta un utente è composto da 23 elementi, suddivisi in 4 gruppi di dati: registrazione, demografia, RS_Info e Social. Nel primo rientrano le informazioni fornite all'atto della registrazione; nella seconda ci sono informazioni di carattere demografico; la terza contiene informazioni specifiche per il RS; la quarta contiene i social metadata.

È opportuno sottolineare le strutture relative al gruppo delle RS_Info: il campo *Servizi_Visti* è formato da due elementi, l'identificativo del servizio e la data di visione. La stessa cosa vale per *Servizi_Acquistati* e *Servizi_Valutati*. Per quest'ultimo c'è anche un terzo elemento che indica il rating dell'utente. Per quanto riguarda i campi *Suggerimenti_Acceptati* e *Suggerimenti_Rifiutati* la struttura prevede l'identificatore del servizio, il tipo di suggerimento, e la data in cui è stato preso in considerazione. *Peso_Proprietà* definisce, per ogni categoria di servizio, i pesi che un utente assegna alle varie feature. Infine, il campo *Utenti_Simili* elenca gli S profili più simili all'utente, specificando anche il valore di similarità. (vedi figura 5.10)

5.4.5 Rappresentazione dei servizi

I vettori che rappresentano i servizi sono composti da 15 elementi, suddivisi in 2 gruppi di dati: base e descrizione. Nel primo rientrano le informazioni utili a individuare le caratteristiche del servizio; nella seconda, invece, ci sono i dati prodotti dagli utenti, che permettono una descrizione completa e approfondita del servizio.

Soffermandoci sulle strutture delle componenti descrittive, il primo elemento è *Descrizione_Servizio* che contiene una caratterizzazione oggettiva di ciò che offre il servizio. *Recensioni*, invece, contiene i racconti delle esperienze utente, identificando anche l'autore e la data di stesura. Il campo *Rating* è formato da tre elementi: un elenco delle proprietà del servizio, ciascuna corredata da un giudizio, l'identificatore dell'autore e la data di valutazione. *Immagini* e *Video* hanno una struttura simile: il nome del contenuto multimediale e la data di upload. Il campo *Descrizione_Semantica* è semplicemente un elenco di tag. Di maggiore interesse sono i quattro campi successivi: *Acquistato_Da* e *Valutato_Da* forniscono un elenco, rispettivamente, di chi ha comprato il servizio e di chi lo ha anche giudicato, specificando nel

primo caso la data di acquisto e il contesto, nel secondo il rating e la data di valutazione; *Servizi_Simili* è un elenco dei servizi che un alto grado di affinità con il vettore in questione (e per ciascuno se ne indica il valore di similarità), mentre *Stato_Servizio* indica se il servizio è attivo o inattivo e la data dell'ultima modifica di stato.

Tipo dato	Nome elemento	Struttura
Registrazione	1 - ID_Utente	<i>int</i>
	2 - Nome_Utente	<i>string</i>
	3 - Indirizzo_email	<i>string</i>
	4 - Password	<i>string</i>
Demografia	5 - Nome	<i>voc_NO</i>
	6 - Cognome	<i>voc_CO</i>
	7 - Sesso	<i>voc_SE</i>
	8 - Data di Nascita	<i>struct data</i>
	9 - Lingua_Madre	<i>voc_LI_MA</i>
	10 - Provincia_Residenza	<i>voc_PR_RE</i>
	11 - Paese_Residenza	<i>voc_PA_RE</i>
RS_Info	12 - Servizi_Visti	<i>array</i> [ID_Servizio, <i>struct data</i>]
	13 - Servizi_Acquistati	<i>array</i> [ID_Servizio, <i>struct data</i>]
	14 - Servizi_Valutati	<i>array</i> [ID_Servizio, <i>struct data</i> , <i>int</i>]
	15 - Suggerimenti_Acceptati	<i>array</i> [ID_Servizio, <i>voc_TI_SU</i> , <i>struct data</i>]
	16 - Suggerimenti_Rifiutati	<i>array</i> [ID_Servizio, <i>voc_TI_SU</i> , <i>struct data</i>]
	17 - Peso_Proprietà	<i>array</i> [<i>voc_CA_SE</i> , <i>array</i> [<i>voc_PR_SE</i> , <i>int</i>]]
Social	18 - Utenti_Simili	<i>array</i> [ID_Utente, <i>int</i>]
	19 - Amici	<i>array</i> [ID_Utente]
	20 - Istruzione e Lavoro	<i>array</i> [<i>voc_IS_LA</i>]
	21 - Sport	<i>array</i> [<i>voc_SP</i>]
	22 - Arte e Intrattenimento	<i>array</i> [<i>voc_AR_IN</i>]
	23 - Attività e Interessi	<i>array</i> [<i>voc_AT_IN</i>]

Figura 5.10: Struttura di un vettore utente

Tipo dato	Nome elemento	Struttura
Base	1 - ID_Servizio	<i>int</i>
	2 - Nome_Servizio	<i>string</i>
	3 - Service_Provider	<i>voc_SE_PR</i>
	4 - Categoria_Servizio	<i>voc_CA_SE</i>
	5 - Proprietà_Servizio	<i>array</i> [<i>voc_PR_SE</i> , <i>int</i>]
Descrizione	6 - Descrizione_Servizio	<i>array</i> [<i>string</i> , ID_Utente]
	7 - Recensioni	<i>array</i> [<i>string</i> , ID_Utente, <i>struct data</i>]
	8 - Rating	<i>array</i> [<i>array</i> [<i>voc_PR_SE</i> , <i>int</i>], ID_Utente, <i>struct data</i>]
	9 - Immagini	<i>array</i> [ID_image, ID_Utente, <i>struct data</i>]
	10 - Video	<i>array</i> [ID_video, ID_Utente, <i>struct data</i>]
	11 - Descrizione_Semantica	<i>array</i> [<i>voc_DE_SE</i>]
	12 - Acquistato_Da	<i>array</i> [ID_Utente, <i>array</i> [<i>struct data</i> , <i>struct context</i>]]
	13 - Valutato_Da	<i>array</i> [ID_Utente, <i>array</i> [<i>int</i> , <i>struct data</i>]]
	14 - Servizi_Simili	<i>array</i> [ID_Servizio, <i>int</i>]
	15 - Stato_Servizio	<i>voc_ST_SE</i> , <i>struct data</i>

Figura 5.11: Struttura di un vettore servizio

5.4.6 Rappresentazione dei Service Provider

Il vettore che rappresenta i Service Provider è composto da 13 elementi, suddivisi in 3 gruppi: base, contatti e descrizione. Tralasciando i primi due, di facile intuizione, volgiamo l'attenzione al terzo: il campo *Servizi_Offerti* consiste in una lista dei servizi erogati dal Service Provider. *Reputazione_Servizi* offre, per ciascun servizio, un riepilogo dei rating

(medi) di tutte le proprietà. Il campo *Rapporto_Clientela*, invece, è un elenco di valutazioni testuali del rapporto dei clienti con il Service Provider. Infine, *Stato_Provider* indica lo stato del Service Provider (attivo o inattivo) e la data dell'ultima variazione di stato.

Tipo dato	Nome elemento	Struttura
Base	1 - ID_Provider	<i>int</i>
	2 - Nome_Provider	<i>string</i>
	3 - Tipologia_Provider	<i>voc_TI_PR</i>
	4 - Nome_Responsabile	<i>string</i>
	5 - Cognome_Responsabile	<i>string</i>
Contatti	6 - Sede	<i>string</i>
	7 - Numero_Telefono	<i>array[int]</i>
	8 - Numero_Fax	<i>array[int]</i>
	9 - Indirizzo_email	<i>array[string]</i>
Descrizione	10 - Servizi_Offerti	<i>array[ID_Servizio]</i>
	11 - Reputazione_Servizi	<i>array[ID_Servizio, array[voc_PR_SE, int]]</i>
	12 - Rapporto_Clientela	<i>array[ID_Servizio, array[ID_Utente, string, struct data]]</i>
	13 - Stato_Provider	<i>voc_ST_PR, struct data</i>

Figura 5.12: Struttura di un vettore Service Provider

5.4.7 I feedback degli utenti

Nel capitolo 4 abbiamo visto l'importanza che hanno i feedback degli utenti nell'ambito della descrizione, della ricerca, e della selezione di un servizio. Il compito di un RS è proprio quello di cercare un servizio in base alle preferenze di un utente, proporglielo, fornendogli anche informazioni sufficienti a una scelta consapevole. SeReS CI considera sei tipi di feedback:

- Descrizioni delle caratteristiche del servizio
- Recensioni delle esperienze utente
- Rating della qualità del servizio
- I tag per la descrizione semantica del servizio
- Valutazione del Service Provider
- Valutazione dei suggerimenti

Certamente i feedback possono presentare delle discrepanze tra loro. Quando queste diversità sono molto evidenti rispetto alla media, il RS provvede a segnalare l'anomalia all'authority che provvede a un controllo, verificando la reale situazione. Nel caso di difformità riguardanti le recensioni o le valutazioni del Service Provider, la suddetta verifica non è necessaria, in quanto sono espressioni di esperienze individuali ed è normale che presentino delle incongruenze.

Le descrizioni

Le descrizioni devono poter spiegare agli utenti che le leggono le caratteristiche oggettive di un servizio, in particolare devono indicare quali attributi sono presenti, e quali invece mancano.

La descrizione di un servizio ha, per ciascuna caratteristica, tre campi: *nome caratteristica*, *presenza (o meno)*, e *descrizione*. La struttura quindi è del tipo:

<Nome Caratteristica> <Sì / No> <descrizione testuale>

Le recensioni

Ogni utente può raccontare la propria esperienza individuale riguardante il servizio usufruito attraverso una recensione. L'oggettività richiesta nella descrizione di un servizio, qui lascia spazio alla soggettività della cosiddetta Quality of Experience (vedi capitolo 4, sezione 4.4.2). Le recensioni di un servizio X possono essere aggiunte dagli utenti ogni volta che acquistano il servizio. Il formato con cui si presentano è simile a un elenco, ordinato in modo tale da avere ai primi posti le recensioni più attuali. La struttura di ogni recensione è la seguente:

<Nome Utente> <Data> <Testo Recensione>

I rating

Le recensioni permettono di raccontare le esperienze individuali a parole, l'aggregazione dei singoli rating, invece, consentono una stima numerica della qualità del servizio. Per ogni proprietà, infatti, l'utente può indicare una valutazione specifica che esprima il livello del suo gradimento.

Il rating può essere espresso tramite un valore da 1 a 5, e comprende anche i mezzi voti (1.5, 2.5, 3.5, 4.5). Ad ogni numero intero corrisponde un giudizio, che aiuta l'utente nella scelta della valutazione più appropriata:

- 1 = pessimo
- 2 = insufficiente
- 3 = sufficiente
- 4 = buono
- 5 = ottimo

I mezzi voti forniscono agli utenti indecisi tra due giudizi una soluzione intermedia.



Figura 5.13 : Rating della proprietà di un servizio

I tag

Un utente può associare a un determinato servizio tag che esprimono delle caratteristiche del servizio erogato (es. il termine *bagno* per un servizio di trasporto qualora sia presente una toilette a bordo del veicolo), oppure tag a livello qualitativo, che segnalano aspetti più o meno positivi del servizio (es. il termine *economico* per un servizio di ristorazione qualora il ristorante pratichi prezzi medio-bassi). Per guidare l'utente nell'inserimento di tag, il RS mette a disposizione un elenco di parole chiave già utilizzate nel sistema per servizi della medesima categoria. Questi termini, riprendendo la distinzione tra le due finalità possibili di un tag, sono organizzati in due classi: tag descrittivi e tag qualitativi. I tag qualitativi, inoltre, sono suddivisi in tre gruppi: positivi, neutri e negativi (in base al giudizio che si vuole esprimere). L'utente, oltre a scegliere uno dei tag suggeriti, può decidere di aggiungere termini propri in ciascuna categoria:

INSERISCI TAG

Tag descrittivi

Seleziona dall'elenco ▼

Inserisci un tag personale

+
Aggiungi

+
Aggiungi

Tag qualitativi

POSITIVI

Seleziona dall'elenco ▼

Inserisci un tag personale

+
Aggiungi

+
Aggiungi

NEUTRI

Seleziona dall'elenco ▼

Inserisci un tag personale

+
Aggiungi

+
Aggiungi

NEGATIVI

Seleziona dall'elenco ▼

Inserisci un tag personale

+
Aggiungi

+
Aggiungi

Figura 5.14 : associazione di tag a un servizio

Le valutazioni dei provider

I rating e le recensioni esprimono in formato numerico e testuale il giudizio dell'utente sul servizio. Le medesime forme di valutazione, riferite però al service provider costituiscono congiuntamente la reputazione dell'entità erogante. La *valutazione numerica*, permette di stimare la reputazione del service provider con un numero da 1 a 5, mezzi voti compresi. La *valutazione descrittiva*, invece, consente una comprensione completa della qualità del provider, tramite descrizioni testuali delle singole esperienze utente.

La valutazione di un service provider, quindi, ha la seguente struttura:

<Nome Service Provider> <valutazione numerica> <valutazione descrittiva>

La valutazione dei suggerimenti

I suggerimenti ricevuti da un utente possono rivelarsi appropriati oppure no. L'utente ha la possibilità di esprimere il giudizio sulla proposta ricevuta, cosicché il sistema apprenda le reali preferenze dell'utente. Se la valutazione del suggerimento è positiva vuol dire che il Recommender Engine ha svolto correttamente il suo compito. Se invece è negativa, l'utente, oltre a indicare che il suggerimento non era appropriato, può specificare cosa non l'ha convinto, fornendo così al sistema informazioni utili per la formulazione dei consigli successivi.

Valuta i suggerimenti per migliorare i prossimi !

Questo suggerimento ti è piaciuto?

SI

NO

Figura 5.15: valutare i suggerimenti

5.4.8 Le competenze della Collective Intelligence Platform

La Collective Intelligence platform è la base su cui poggia l'intero Recommender System. Nella tabella 5.1 di questo capitolo sono state elencate le opportunità che i tre livelli di CI mettono a disposizione del sistema. Tale lista, insieme a quattro operazioni su utenti e servizi, due di ricerca e due di confronto, costituiscono le funzionalità offerte dalla CI platform:

- Descrivere un servizio
- Recensire un servizio
- Aggiornare i dataset
- Valutare i servizi
- Definire la reputazione del service provider
- Rappresentare semanticamente i servizi (caso esplicito)
- Determinare i servizi di maggior successo
- Definire le preferenze di utenti tipo
- Completare i profili
- Associare servizi a gruppi di utenti
- Rappresentare semanticamente i servizi (caso implicito)
- Trovare servizi simili (routine *FindSimilarService*)
- Confrontare servizi (funzione *HowSimilarService*)
- Trovare utenti simili (routine *FindSimilarUser*)
- Confrontare utenti (funzione *HowSimilarUser*)

In questa sezione, quindi, vengono descritte alcune strategie possibili, che possono essere adottate per ottenere le sopracitate funzionalità. Le proposte avanzate, si pongono come unico obiettivo una spiegazione più concreta di come poter realizzare in pratica quanto detto sul piano teorico, pur rimanendo volutamente lontani da una efficiente proposta implementativa.

Descrizione di un servizio

Per aggiornare la descrizione, l'utente inserisce un testo con il quale aggiunge dettagli relativi al servizio. Tale azione comporta la modifica della componente *Descrizione_Servizio* del vettore corrispondente: viene inserito nell'array l'elemento contenente la nuova descrizione e l'identità dell'autore, che si va ad aggiungere ai precedenti.

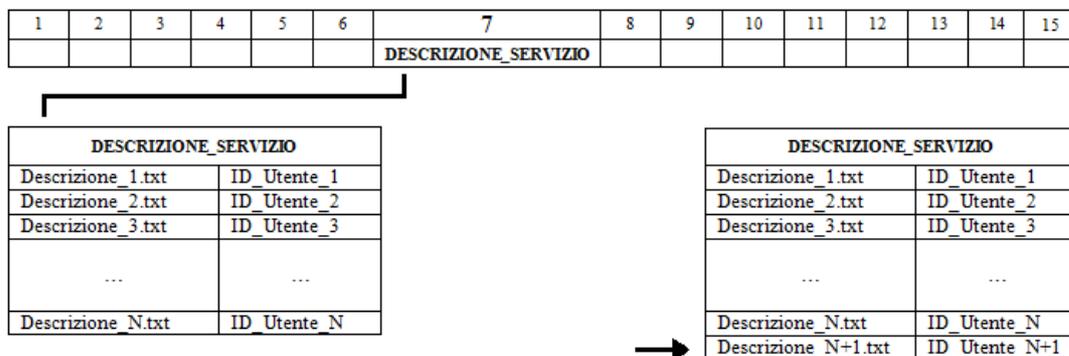


Figura5.16: Aggiornamento Descrizione di un servizio

Recensione di un servizio

L'inserimento di una recensione è un processo analogo a quello enunciato per la descrizione dei servizi. Le uniche due differenze sono: la componente del vettore servizio interessata questa volta è *Recensioni*; viene inserito il codice del servizio nel vettore utente alla voce *Servizi_Valutati*. Si noti che l'array *Servizi_Valutati* viene mantenuto ordinato (per identificatore di Servizio crescente).

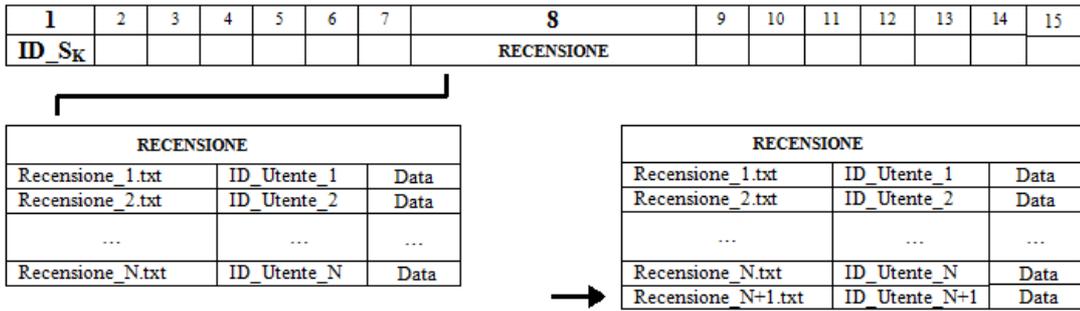


Figura5.17: Aggiornamento Recensioni - parte Servizio

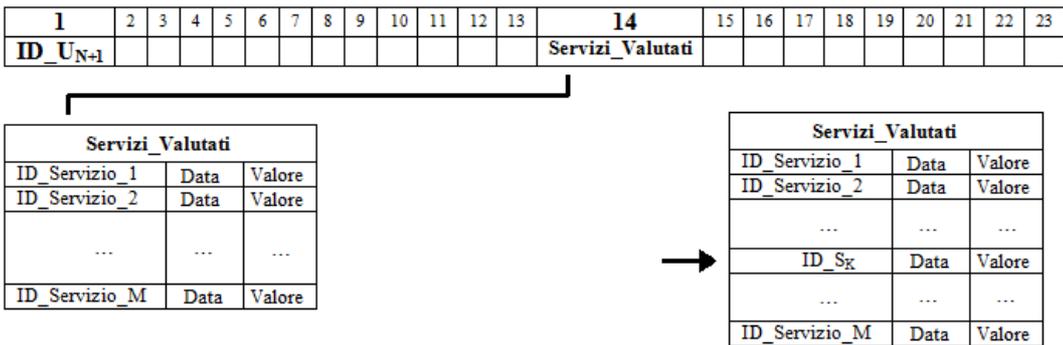


Figura5.18: Aggiornamento Recensioni - parte Utente

Aggiornamento dei dataset

L'operazione di aggiornamento del dataset dei servizi si può suddividere in cinque possibili azioni:

1. *Inserimento di un service provider.* L'inserimento di uno nuovo service provider richiede la specifica delle *Informazioni di base*, dei contatti e della descrizione. In mancanza di tali dati non è possibile effettuare l'inserimento. Prima di validare l'inserimento del service provider nel sistema (*Stato_Provider = attivo*), viene effettuato un controllo sulla veridicità delle informazioni da parte dell'authority.
2. *Cancellazione di un service provider.* La cancellazione di un service provider comporta una reazione a catena. Innanzitutto i servizi da lui erogati non sono più disponibili e quindi dovrebbero essere eliminati dal dataset. Tuttavia, se si procedesse in questo modo, tutti i riferimenti degli utenti ai servizi con cui hanno interagito in passato andrebbero persi. Una alternativa, quindi è contrassegnare i servizi da cancellare come inattivi (vedi punto 5). La medesima soluzione viene applicata ai service provider che hanno terminato la loro attività e devono essere rimossi dal dataset (campo *Stato_Provider* settato a *inattivo*).

1	2	3	4	5	6	7	8	9	10	11	12	13
Informazioni di base					Contatti			Descrizione				
												Stato_Provider

Figura5.19: Vettore di un service provider appena aggiunto al dataset

3. *Inserimento di un servizio.* Inserire un nuovo servizio nel dataset richiede la specifica delle informazioni di base all'interno del vettore, mentre gli elementi descrittivi sono facoltativi e possono essere aggiunti in un secondo momento da altri utenti. Il campo *Servizi_Simili* del vettore viene completato da una routine *FindSimilarService* della Collective Intelligence Platform che procede al confronto del servizio inserito con quelli già esistenti. Il campo *Stato_Servizio* viene impostato *attivo* dopo un controllo dell'authority sulla reale esistenza del servizio.
4. *Aggiornamento della descrizione di un servizio esistente.* Il processo di aggiornamento, per la parte di inserimento delle nuove descrizioni, è uguale all'operazione *Descrizione di un servizio* presentata in precedenza. Nel caso in cui ci siano delle correzioni che comportano l'eliminazione di una o più descrizioni, invece, si deve procedere alla rimozione di uno o più elementi.
5. *Cancellazione di un servizio.* Per i medesimi motivi enunciati per la cancellazione di un Service Provider, il servizio non viene effettivamente cancellato dal dataset, bensì viene impostata su *inattivo* la variabile *Stato_Servizio*.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Informazioni di base													Servizi_simili	Stato_Servizio

Figura5.20: Vettore di un servizio appena aggiunto al dataset

Valutare i servizi

Dato l'insieme di P proprietà (*feature*) di un servizio, elencate nel campo *Proprietà_Servizio* del relativo vettore, un utente che ha usufruito del servizio (e quindi in possesso del codice associato) può esprimere una valutazione numerica per ciascuna proprietà, e il tutto viene memorizzato nel campo *Rating* del vettore, insieme all'identificatore del cliente e alla data. Calcolando la media di tutti i giudizi espressi dagli utenti si ottiene la valutazione qualitativa del servizio in questione. Infine, l'aggiornamento del servizio comporta l'invocazione della routine *FindSimilarService*, il cui risultato viene inserito nel campo *Servizi_Simili*.

Oltre alla modifica del vettore del servizio, il sistema compie un aggiornamento dell'elemento *Servizi_Valutati* del vettore Utente, e di *Reputazione_Servizi* del vettore Service Provider.

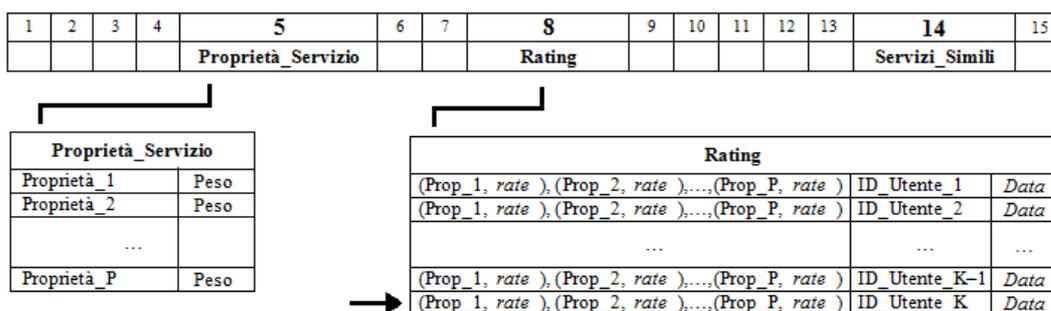


Figura 5.21: Valutazione di un servizio

Definire la reputazione del service provider

La definizione della reputazione di un service provider dipende da due elementi: le valutazioni degli utenti dei servizi erogati (vedi *Valutare i servizi*) e il rapporto con la clientela, espresso sotto forma di giudizi testuali. La prima componente permette di classificare e di ordinare i service provider secondo un parametro numerico. La seconda componente, invece, consente una comprensione più completa della qualità del provider, fornendo descrizioni testuali delle singole esperienze utente.

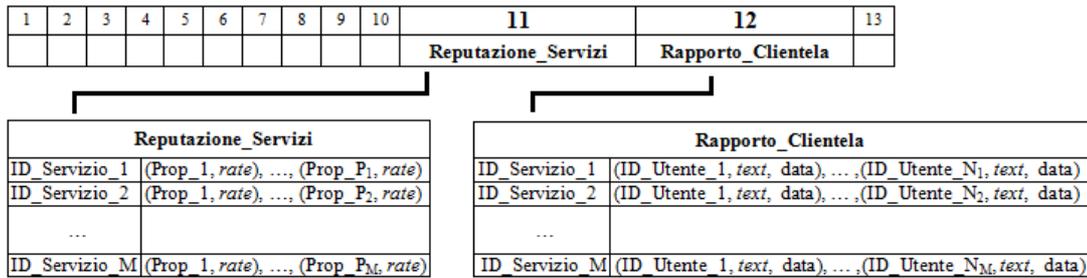


Figura 5.22: Definire la reputazione del service provider

Rappresentare semanticamente i servizi (caso esplicito)

Gli utenti premium, oltre a scrivere una descrizione o una recensione del servizio acquistato, possono anche associargli uno o più tag, rappresentativi delle caratteristiche del servizio. L'insieme dei tag, come visto nel capitolo 3 (vedi 3.3.2 *Semantic Recommender System*), esprimono la descrizione semantica del servizio. Tale insieme è memorizzato nel campo *Descrizione_Semantica* del vettore servizio. Inoltre il sistema, come risposta all'aggiornamento di questo elemento, chiama la routine *FindSimilarService*, i cui risultati vengono inseriti nel campo *Servizi_Simili*.

Per quanto riguarda il vettore utente, il sistema inserisce l'identificativo del servizio in questione nel campo *Servizi_Valutati*.

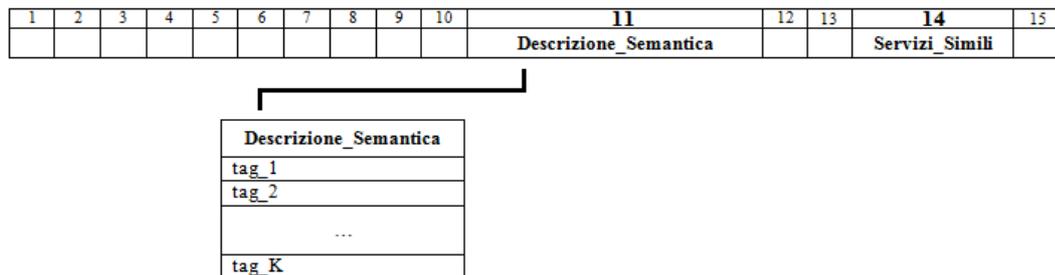


Figura 5.23: descrizione semantica di un servizio

Determinare i servizi di maggior successo

Data una categoria di servizi (hanno il medesimo campo *Categoria_Servizio*), si vogliono determinare quelli di maggior successo. I parametri che influenzano questa scelta sono due: il numero di utenti che hanno acquistato il servizio (campo *Acquistato_Da*) e le valutazioni numeriche degli utenti che hanno anche giudicato il servizio (campo *Rating*). La prima componente permette una scrematura dei servizi, selezionando solo quelli che sono stati visionati da una base utenti sufficientemente estesa (si fissa una soglia minima, proporzionale

al numero di utenti totali del sistema); la seconda componente, invece, determina una classificazione qualitativa dei servizi scelti, mettendo ai primi posti quelli con giudizi più alti. Si noti che si può anche stilare una graduatoria ristretta a un particolare intervallo temporale (servizi del mese, della settimana, del giorno, etc.), prendendo in considerazione solo le viste e i rate dopo una certa data (nella figura seguente d_{Si} e D_i indicano, rispettivamente, gli indici della prima e dell'ultima data, appartenenti all'intervallo di interesse, relativi alle viste dell'utente i ; mentre m e M sono gli indici del primo e dell'ultimo utente che hanno fornito un rating nel periodo specificato).

Servizio S_a

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
							Rating				Acquistato_Da			

Rating		
(Prop_1, rate), (Prop_2, rate), ..., (Prop_P, rate)	ID_Utente_1	data1
(Prop_1, rate), (Prop_2, rate), ..., (Prop_P, rate)	ID_Utente_2	data2
...
(Prop_1, rate), (Prop_2, rate), ..., (Prop_P, rate)	ID_Utente_m	datam
...
(Prop_1, rate), (Prop_2, rate), ..., (Prop_P, rate)	ID_Utente_M	dataM

Acquistato_Da		
ID_Utente_1	(data1 ... datad _{S1} ... dataD ₁)	contesto
ID_Utente_2	(data1 ... datad _{S2} ... dataD ₂)	contesto
...
ID_Utente_K	(data1 ... datad _{SK} ... dataD _K)	contesto

$$\text{Viste totali* : } \sum_{i=1}^K (D_i - d_{Si})$$

$$\text{Rating medio* : } \frac{1}{P} \sum_{\rho=1}^P \frac{1}{M - m} \left(\sum_{i=m}^M \text{rate_utente_i} (\text{Prop_}\rho) \right)$$

* = Relativi al periodo di interesse

Figura 5.24: calcolo delle viste totali e del rating medio di un servizio per la selezione dei servizi di maggior successo

Definire le preferenze di utenti tipo

Come abbiamo visto in sede di presentazione del sistema, il Recommender System può essere utilizzato anche da utenti che non possiedono un profilo, oppure hanno fornito elementi insufficienti ad individuare le loro preferenze. In circostanze simili il RS deve comunque essere in grado di rispondere alle esigenze dell'utente in modo soddisfacente e, a tale scopo, si è deciso di adottare una strategia di clustering, che individua 3 tipologie di utenti, creati in modo differente, facendo sì che i suggerimenti all'utente, nella prima fase della sua interazione con il RS, siano appropriate alle caratteristiche dei 3 profili, e poi sia lui stesso a scremare i risultati proposti con scelte esplicite, permettendo al sistema di conoscere meglio le sue reali preferenze. Gli algoritmi di clustering utilizzati sono tre, e si applicano all'insieme degli utenti che hanno un livello di completamento del profilo alto (vedi 5.3.1 *Tipologie di utenti*):

1. *Clustering gerarchico agglomerativo.* Supponiamo che il sistema sia popolato da N utenti. Si parte dall'utente con ID più basso, e si prendono lui e il primo elemento della lista dei suoi vicini (il primo elemento del campo *Utenti_Simili* del vettore corrispondente), inserendoli nello stesso cluster. Tolti questi due utenti dagli N iniziali, si continua col medesimo procedimento per l'utente successivo, e si itera il processo

finche non sono stati esauriti tutti gli N utenti (se rimane un solo utente costituisce un cluster a se)¹².

A questo punto si hanno $N/2$ cluster, ciascuno formato da 2 elementi: A_1 e B_1 . È importante notare che gli utenti di un cluster sono stati scelti secondo due direttive: il primo elemento (A_1) è l'utente con ID più basso tra quelli rimasti; il secondo elemento (B_1) è l'utente con profilo più simile al primo elemento, sempre tra gli utenti rimasti. Quest'ultima regola comporta due conseguenze: non è certo che B sia effettivamente l'utente più simile ad A_1 , infatti, detto C_1 tale utente, potrebbe essere che C_1 sia stato già scelto per un altro cluster; non è vero in generale che se B_1 è l'utente più simile ad A_1 , allora A_1 è l'utente più simile a B_1 (non è una proprietà riflessiva).

Sulla base delle due precedenti considerazioni si enuncia la seguente regola, fondamentale per i passi successivi dell'algoritmo:

Regola 1

Siano X_1, \dots, X_M gli utenti di uno degli $N/2$ cluster. Sia $N_i = \{ n_{i1}, \dots, n_{iS} \}$ l'insieme dei vicini del generico utente X_i , ordinati per similarità decrescente. Detto n_i il primo elemento di N_i che non sia presente nel cluster (*vicino massimo*), indichiamo con v_i il valore di similarità di questo elemento con X_i . Allora, si elegge rappresentante del cluster l'utente X_j tale che v_j è il massimo tra i v_i , per i da 1 a M .

Applicando questa a regola a tutti i cluster si ottengono $N/2$ rappresentanti R_y , ciascuno con vicino massimo n_y con valore di similarità v_y , definendo quindi la tripla (R_y, n_y, v_y) . Partendo dal rappresentante R_z che ha il valore v_z più alto (tra tutti i v_y , per y da 1 a $N/2$), si seleziona il cluster che contiene l'utente n_z e si crea un cluster unico, formato dalla fusione del cluster a cui appartiene R_z e da quello a cui appartiene n_z . Tolti questi due elementi dall'insieme degli $N/2$ cluster iniziali, si procede ad accorpare nello stesso modo gli altri cluster, fino ad esaurirli tutti. Se un rappresentante ha come vicino massimo un utente appartenente ad un cluster già rimosso, e quindi il suo cluster non si può fondere con quello del suo vicino massimo, si procede al ricalcolo del rappresentante, tenendo presente che il suo vicino massimo deve appartenere a uno dei cluster ancora da accorpare.

Ad ogni iterazione l'algoritmo dimezza il numero di Cluster ($N/2, N/4, N/8, \dots$). Gli utenti del sistema sono N , ognuno ha una lista di vicini di S elementi, con $N \gg S$. Arrivati al passo L tale che $N/2^L = S$, si hanno S cluster, ciascuno di N/S elementi. Di questi, viene scelto il cluster con grado di similarità più alto, in base a questa seconda regola:

Regola 2

Siano $X_1, \dots, X_{N/S}$ gli utenti di uno degli S cluster. Sia $N_i = \{ n_{i1}, \dots, n_{iS} \}$ l'insieme dei vicini del generico utente X_i , ordinati per similarità decrescente. Siano $\{ v_{i1}, \dots, v_{iS} \}$ i valori di similarità dei vicini $\{ n_{i1}, \dots, n_{iS} \}$ con X_i . Si selezionano gli elementi di N_i che appartengono al cluster e di questi si sommano i valori di similarità. La

¹² Se il primo elemento della lista dei vicini di un utente A è già stato inserito in un altro cluster, si considera il secondo elemento, poi eventualmente il terzo, e si continua finché non si trova un elemento che non appartiene ancora a nessun cluster. Se si dovesse esaurire la lista senza successo, si ricorre alla funzione *HowSimilarUser*, applicandola a tutti gli utenti non ancora inseriti nel cluster e scegliendo quello più simile ad A .

stessa cosa viene fatta per tutti gli X_i . Alla fine si sommano le N/S somme parziali e si ottiene il grado di similarità del cluster.

$$\text{deg} = \frac{1}{N / S} \sum_{i=1}^{N / S} \sum_{j=1}^S v_{ij}^*$$

$$\text{con } v_{ij}^* = \begin{cases} v_{ij} & \text{se } n_{ij} \in \text{Cluster} \\ 0 & \text{se } n_{ij} \notin \text{Cluster} \end{cases}$$

Applicando questa regola agli S cluster, si ottengono S valori di gradi di similarità e si seleziona il cluster con grado massimo. A questo punto si dispone di un insieme composto da N/S utenti simili tra loro. L'obiettivo, però, era definire un utente tipo, quindi si procede a un merge dei profili utente per quanto riguarda i campi *Servizi_Visti*, *Servizi_Acquistati*, *Servizi_Valutati*, *Suggerimenti_Acceptati*, *Suggerimenti_Rifiutati*, mentre per i campi *Data_Di_Nascita* e *Peso_Proprietà* si effettua una media dei valori dei singoli utenti. Per quanto riguarda *Lingua_Madre* e *Paese_Di_Residenza* ci si rifà all'origine dell'indirizzo IP (esistono molti tools specifici), mentre per le informazioni di base, per i campi *Nome*, *Cognome*, *Provincia_Di_Residenza*, e per le informazioni *Social* si omette il completamento.

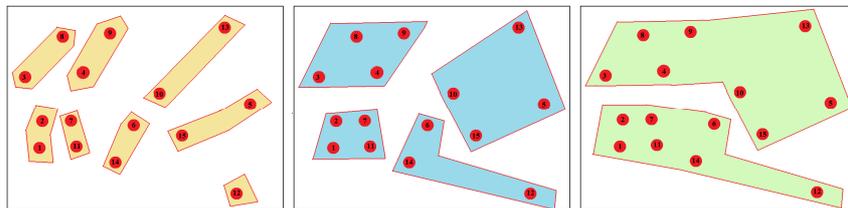


Figura 5.25: Esempio di applicazione dell'algoritmo di clustering gerarchico

2. *Clustering partitivo*. Anche in questo caso si vuole definire un utente medio, tuttavia, diversamente dal primo algoritmo, si punta ad indicare come utente tipo un utente realmente esistente, rappresentativo di un gruppo sufficientemente esteso. A partire dagli N utenti iniziali, si crea una partizione di S cluster, ciascuno composto da N/S elementi. La strategia di scelta degli elementi appartenenti ad un cluster non è rilevante, e si può pensare di selezionare gli utenti in base al loro identificatore: partendo dall'ID più basso si scelgono i primi N/S , e si prosegue finché non si esauriscono tutti gli utenti. A questo punto si dispone di S gruppi, ciascuno di N/S elementi, e per ogni partizione si vuole determinare un centroide, in base al seguente procedimento:

Regola 3

Dato un cluster di N/S utenti, si effettua un merge dei profili utente dei campi *Servizi_Visti*, *Servizi_Acquistati*, *Servizi_Valutati*, *Suggerimenti_Acceptati*, *Suggerimenti_Rifiutati*, mentre per i campi *Data_Di_Nascita* e *Peso_Proprietà* si effettua una media dei valori dei

singoli utenti. Gli altri campi non vengono completati. Il profilo che ne risulta è definito centroide del cluster.

Calcolato il centroide (c_y) di ogni cluster (C_y), si valuta per ogni utente N (tramite la funzione *HowSimilarUser*) qual è il profilo più simile tra quelli degli S centroidi, formando una nuova partizione degli utenti, dove ogni cluster è formato dagli elementi che hanno selezionato il medesimo centroide. Si ripete la procedura, calcolando ad ogni iterazione il centroide di ciascun gruppo e valutando per ogni utente il centroide più vicino. Questa fase dell'algorithm termina quando le partizioni rimangono le stesse oppure si entra in un loop, cioè si ritorna ad una partizione già ottenuta in qualche passo precedente.

Ottenuta la partizione degli utenti con l'algorithm di clustering, ora si deve eleggere l'utente tipo. Per fare questo si seleziona il cluster con il maggior numero di utenti e all'interno di questo gruppo si prende l'utente più simile al centroide.

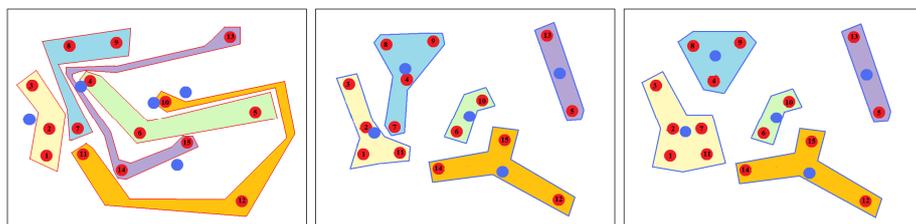


Figura 5.26: Esempio di applicazione dell'algorithm di clustering partitivo

3. *Clustering ibrido*. Questo algorithm mira a creare il profilo dell'utente medio combinando i due algoritmi descritti in precedenza.

L'idea di base è che utilizzando l'algorithm gerarchico si può determinare il cluster con grado di similarità più basso, fornendo così un insieme di utenti simili, ma allo stesso tempo sufficientemente variegati. Sugli elementi di tale sottoinsieme si applica, successivamente, l'algorithm partitivo, producendo una partizione degli utenti in cluster con alto grado di similarità interna. Calcolando, infine, il centroide dei centroidi, si ottiene l'utente tipo cercato.

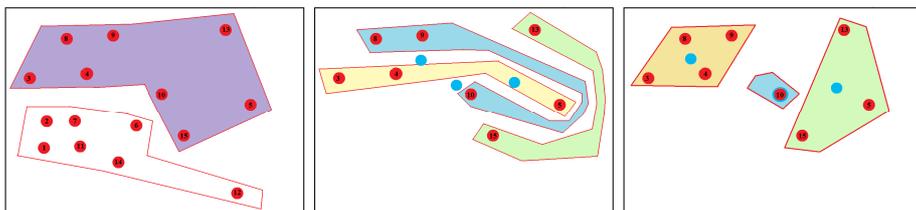


Figura 5.27: Esempio di applicazione dell'algorithm di clustering ibrido

Applicando i tre algoritmi all'insieme degli utenti con livello di completamento del profilo *alto*, si ottengono tre profili differenti. In presenza di un utente non registrato (visitatore) il RS genera un egual numero di suggerimenti per ciascun tipologia, fornendo “un profilo” provvisorio al visitatore. Si noti che il riferimento alle caratteristiche dei tre utenti tipo è utile solo per i primi istanti dell'interazione del visitatore con il sistema, infatti, le successive azioni dell'utente vengono rilevate dal RS e memorizzate in un profilo inizialmente “vuoto”, consentendo la creazione di un profilo implicito.

Completare i profili

Diversamente dal caso appena visto, per il quale non si aveva alcuna informazione riguardo l'utente, qui si suppone di disporre almeno di una conoscenza parziale (livello di completamento del profilo *medio*, vedi 5.3.1 *Tipologie di utenti*).

A partire dagli S vicini dell'utente X , elencati nel campo *Utenti_Simili* del corrispondente vettore, e che indichiamo con $N = \{n_1, \dots, n_S\}$, si selezionano solo le caratteristiche condivise da almeno $\frac{2}{3}$ degli S neighbor di X . Ad esempio, se un servizio S_K è presente nel campo *Servizi_Visti* di R neighbor, con $R > \frac{2}{3} \cdot S$, allora S_K viene inserito nel profilo temporaneo di X e preso in considerazione dal RS durante la generazione dei suggerimenti. Al termine dell'interazione dell'utente con il sistema i dati generati con questo metodo vengono rimossi.

Associare servizi a gruppi di utenti

Determinati servizi possono essere particolarmente interessanti, su base statistica, per utenti con caratteristiche simili. L'idea è di creare gruppi di utenti affini e determinare quali sono i servizi più apprezzati all'interno di ciascun insieme.

Il primo passo è determinare i gruppi di utenti simili, e questo si può fare con un algoritmo di clustering gerarchico (vedi *Definire le preferenze di utenti tipo* in questa sezione). A questo punto si determinano i servizi di maggior successo per ciascun cluster (vedi *Determinare i servizi di maggior successo* in questa sezione) e, partendo dal servizio più quotato, e a seguire tutti gli altri, si propongono agli utenti quelli che essi devono ancora visionare (cioè quelli non presenti nel campo *Servizi_Visti* dei vettori utente).

Rappresentare semanticamente i servizi (caso implicito)

Un servizio può essere descritto semanticamente da un insieme di tag. Questo tipo di descrizione può essere esplicita (CI di livello 1) oppure derivata automaticamente dalle recensioni degli utenti, estrapolando le parole chiave già presenti nel vocabolario di sistema. Nel capitolo 4 abbiamo visto che a tale scopo si può utilizzare la tecnica dello stemming. I passaggi necessari per estrarre tag da un testo non strutturato, qual è una recensione, sono quattro:

1. Dividere il testo in termini singoli.
2. Eliminare la punteggiatura, le congiunzioni, gli articoli le preposizioni e i pronomi.
3. Normalizzare le parole rendendole tutte minuscole.
4. Applicare lo stemming a ciascun termine (troncare le parole, lasciando solo la radice).

A queste quattro operazioni è necessario aggiungere un ulteriore passaggio per rendere i tag utilizzabili dal sistema:

5. Selezionare solo i tag presenti nel vocabolario *voc_DE_SE* (descrizione semantica).

La quinta operazione consente di includere nella descrizione semantica soltanto elementi che siano interpretabili dal RS.

Esempio

Testo originale: *La stanza era spaziosa e pulita e in ordine, durante il ns soggiorno abbiamo trovato disponibilità e cortesia sui servizi non saprei in quanto non li abbiamo utilizzati ne richiesti. L'hotel è un pò fuori ma tranquillo e vicino all'autostrada e in pochissimo tempo in auto si può raggiungere il centro. Per noi è stato un'ottima scelta.*

- Passo 1: Il testo viene suddiviso in singole stringhe, ma nella forma rimane uguale.
- Passo 2: stanza era spaziosa pulita ordine durante ns soggiorno abbiamo trovato disponibilità cortesia servizi non saprei quanto non abbiamo utilizzati richiesti hotel è pò fuori tranquillo vicino autostrada pochissimo tempo auto può raggiungere centro è stato ottima scelta.
- Passo 3: I termini rimasti erano già tutti minuscoli, quindi non sono necessarie variazioni.
- Passo 4: stanza essere spazio pulizia ordine ns soggiorno avere trovare disponibilità cortesia servizio non sapere non avere utilizzare richiedere hotel essere poco fuori tranquillo vicino autostrada poco tempo auto potere raggiungere centro essere stare ottimo scelta
- Passo 5: spazio - pulizia - ordine - disponibilità - cortesia – tranquillità - ottimo

La routine *FindSimilarService*

Per determinare la similarità tra coppie di servizi la CI platform ricorre ad una routine, che viene chiamata ogni qual volta che viene introdotto un nuovo servizio nel dataset, o quando vengono aggiornate le informazioni relative ad un servizio esistente.

La routine consta di sei fasi:

1. La CI Platform, per ogni servizio, a partire dal vettore ad esso associato, genera un vettore semplificato, che chiamiamo *service comparison vector*, di 4 elementi:
 - *categoria servizio*: categoria a cui appartiene il servizio;
 - *proprietà servizio*: le feature che descrivono il servizio con il relativo peso;
 - *rating*: elenco delle valutazioni medie di ogni feature;
 - *descrizione semantica*: tag con cui è rappresentato semanticamente il servizio.
2. Si effettua la prima scrematura: se due servizi appartengono a categorie differenti il valore di similarità è pari a 0 e per essi non è necessario applicare i passi successivi della routine.
3. Se due servizi A e B appartengono alla medesima categoria hanno necessariamente le stesse K feature che li descrivono, tutt'al più, se alcune caratteristiche non sono presenti per mancanze del service provider, il valore delle corrispondenti feature è pari a 0. Un primo livello di confronto tra due servizi, quindi, è il calcolo della similarità qualitativa, che si ottiene valutando per ogni feature f_i la seguente misura di similarità:

$$Sim_i(A, B) = \frac{\sum_{j=1}^L (r_{jA} - \bar{r}_A) \cdot (r_{jB} - \bar{r}_B)}{\sqrt{\sum_{j=1}^L (r_{jA} - \bar{r}_A)^2 \cdot \sum_{j=1}^N (r_{jB} - \bar{r}_B)^2}}$$

Si noti che $Sim_i(A, B)$ è la *Correlazione di Pearson* presentata nel capitolo 3 (vedi *Calcolo della similarità tra item* della sezione 3.3.4). Indichiamo con $\{u_1, \dots, u_L\}$ il sottoinsieme di utenti che hanno valutato sia A che B, con r_{jA} la valutazione che

l'utente u_j ha dato per la feature i del servizio A, e con \bar{r}_x la valutazione media che la feature i di A ha ricevuto dagli utenti $\{u_1, \dots, u_L\}$ (per B è analogo). Tale calcolo viene effettuato per tutte le K feature di A e B.

4. A questo punto si dispone di K valori di similarità relativi alle feature dei servizi, ma non tutti sono egualmente importanti, tant'è vero che ogni utente assegna un peso differente a ciascuno di essi. Moltiplicando ciascuno dei K valori per la relativa componente peso (*weight*), memorizzata nell'elemento *proprietà servizio* del *comparison vector*, e poi sommando i prodotti ottenuti, si ottiene il valore di similarità qualitativa dei due servizi.

$$QS(A, B) = \sum_{i=1}^K (Sim_i(A, B) \cdot (weight(i)))$$

5. Rimane soltanto da misurare la similarità semantica, che si deriva confrontando l'insieme dei tag associati ad A (T_A) con quello dei tag associati a B (T_B). Supponiamo che T_A sia formato da N_A elementi e T_B ne abbia N_B . Indichiamo con N_{AB} il numero di tag che sono presenti sia in T_A che in T_B . La similarità semantica di A e B allora si misura attraverso la seguente formula:

$$SS(A, B) = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

6. L'ultimo passo consiste nell'integrazione di $QS(A, B)$ con $SS(A, B)$. Anche qui si effettua una somma pesata, in quanto l'importanza della similarità semantica è limitata rispetto a quella della similarità qualitativa. Il rapporto proposto è di 1 a 4 a favore della QS , però può essere modificato a seguito di risultati sperimentali in fase di implementazione del sistema. Indicati con W_{QS} e W_{SS} i pesi delle due misure di similarità, la similarità complessiva tra due servizi A e B è data da:

$$Sim(A, B) = W_{QS} \cdot QS(A, B) + W_{SS} \cdot SS(A, B)$$

Calcolati i valori di similarità del servizio A rispetto a tutti gli altri, e stilato un elenco ordinato per valore di similarità decrescente, si memorizzano i primi S_{SS} servizi nel campo *Servizi_Simili* di A, e si aggiornano i campi *Servizi_Simili* degli altri servizi nel seguente modo: dato un servizio D, si consideri il suo campo *Servizi_Simili*, e in particolare l'ultimo elemento (E, $Sim(D, E)$). Se $Sim(D, E)$ è minore di $Sim(A, D)$, allora (E, $Sim(D, E)$) viene eliminato e viene inserito (A, $Sim(A, D)$), mantenendo ordinata la lista per valore di similarità decrescente.

La funzione *HowSimilarService*

I sei passi della routine *FindSimilarService* appena descritti, costituiscono una funzione, che chiamiamo *HowSimilarService*, la quale valuta la similarità di due servizi A e B. La routine applica tale funzione iterativamente, al fine di confrontare un certo servizio A con tutti gli altri servizi del sistema. La funzione, però, può essere applicata anche in altri contesti, ad esempio nella generazione di suggerimenti content-based (vedi la sezione 5.4.9 *Il Recommender Engine*).

La routine *FindSimilarUser*

Innanzitutto è fondamentale precisare che ho definito la similarità di due utenti X e Y secondo tre contesti: commerciale, demografico, e sociale. La routine *FindSimilarUser*, quindi, associa all'utente Y una terna di valori, corrispondenti alla similarità di Y a X, rispettivamente, nel contesto commerciale, in quello demografico e in quello sociale. Poiché l'oggetto dei suggerimenti del RS è il mondo dei Servizi, l'affinità di due utenti dipende soprattutto dalla somiglianza commerciale, secondariamente da quella demografica e infine da quella sociale. Quindi se la routine associa a X e Y la terna $S = (s_1, s_2, s_3)$, e agli utenti X e Z la terna $T = (t_1, t_2, t_3)$, per capire se X assomiglia più a Y o a Z, si confrontano s_1 e t_1 , a parità di valore si confrontano s_2 e t_2 , e soltanto se anch'essi sono uguali si prendono in considerazione anche le componenti s_3 e t_3 .

Ogni qualvolta si registra un nuovo utente (N) oppure viene aggiornato un profilo già esistente il RS provvede ad avviare la routine *FindSimilarUser*, la quale compila (o aggiorna) il campo *Utenti_Simili* del vettore associato a N. Inoltre, se un utente X risulta simile a N con un valore di similarità pari a $K = (K_1, K_2, K_3)$, e nel campo *Utenti_Simili* di X l'ultimo elemento (quello con valore di similarità più basso rispetto a X) è l'utente Y, al quale è associata la terna $L = (L_1, L_2, L_3)$, con $L < K$, allora Y viene rimosso dalla lista degli utenti simili e viene inserito X (mantenendo ordinata la lista per valore di similarità decrescente).

La routine consta di 3 fasi:

1. La CI Platform, per ogni utente, a partire dal vettore ad esso associato, genera un vettore semplificato, che chiamiamo *user comparison vector*, di undici elementi:
 - *servizi acquistati*: elenco dei servizi acquistati;
 - *servizi valutati*: elenco dei rating assegnati ad ogni servizio
 - *peso proprietà*: importanza che assegna a ogni proprietà di un servizio;
 - *sesso*: maschio o femmina;
 - *età*: calcolata a partire dalla data di nascita;
 - *lingua madre*: la lingua preferita;
 - *provincia di residenza*: dove abita;
 - *paese di residenza*: Stato in cui si trova attualmente;
 - *istruzione*: dove ha studiato;
 - *lavoro*: impiego;
 - *sport*: sport praticato.

I primi due elementi (*servizi acquistati* e *peso proprietà*) definiscono il contesto commerciale; dal terzo al settimo (*sesso*, *età*, *lingua madre*, *provincia di residenza*, *paese di residenza*) fanno riferimento alla somiglianza demografica; gli ultimi tre (*istruzione*, *lavoro*, *sport*) appartengono all'ambito sociale.

2. Dati due utenti X e Y, ogni elemento del *comparison vector* di X viene confrontato con l'omologo del *comparison vector* di Y, tramite una funzione ad hoc. Di seguito si definiscono le 11 funzioni:

$$- F_{\text{acquisti}} : (X_{\text{acquisti}}, Y_{\text{acquisti}}) \rightarrow \frac{\sum_{i=1}^K f_i}{K}$$

Siano $\{s_1, \dots, s_K\}$ i K servizi acquistati da almeno uno tra X e Y e ordinati per ID; sia f_i il valore di verità della proprietà "il servizio s_i è stato acquistato da

entrambi gli utenti” (vale 0 se è falso, 1 se è vero). La funzione F_{acquisti} associa alla coppia (acquisti di X, acquisti di Y) il rapporto tra gli acquisti effettuati da entrambi e il numero di servizi totali presi in considerazione. Il valore minimo è 0 quello massimo è 1.

$$- F_{\text{rating}} : (X_{\text{rating}}, Y_{\text{rating}}) \rightarrow \frac{\sum_{j=1}^L (r_{jX} - \bar{r}_X) \cdot (r_{jY} - \bar{r}_Y)}{\sqrt{\sum_{j=1}^L (r_{jX} - \bar{r}_X)^2 \cdot \sum_{j=1}^L (r_{jY} - \bar{r}_Y)^2}}$$

Questa funzione è la *Correlazione di Pearson* presentata nel capitolo 3 (vedi *Calcolo della similarità tra user* della sezione 3.3.4). Sia $\{s_1, \dots, s_L\}$ il sottoinsieme di servizi valutati sia da X che da Y. Indichiamo con r_{jX} la valutazione media che l’utente X ha dato per il servizio s_j ¹³, e con \bar{r}_X la valutazione media che X ha fatto sui servizi $\{s_1, \dots, s_L\}$ (per Y è analogo). La funzione F_{rating} associa alla coppia (*valutazioni di X, valutazioni di Y*) la correlazione di Pearson tra i due insiemi.

$$- F_{\text{pesi}} : (X_{\text{pesi}}, Y_{\text{pesi}}) \rightarrow \frac{1}{L} \left(\sum_{i=1}^L \frac{\min\{w_{iX}, w_{iY}\}}{\max\{w_{iX}, w_{iY}\}} \right)$$

Siano $\{p_1, \dots, p_L\}$ le L feature possibili per descrivere un servizio; sia w_{iX} il peso assegnato dall’utente X alla feature p_i (per Y è analogo). La funzione F_{pesi} associa alla coppia (pesi di X, pesi di Y) la media dei rapporti tra i pesi assegnati da X alle feature e quelli assegnati da Y (l’utilizzo delle funzioni min e max è giustificato dalla necessità di avere frazioni il cui valore sia compreso tra 0 e 1).

$$- F_{\text{sesto}} : (X_{\text{sesto}}, Y_{\text{sesto}}) \rightarrow \{0,1\}$$

Associa alla coppia (sesto di X, sesso di Y) il valore 0 se sono di sesso diverso (o se per almeno uno dei due manca il dato), il valore 1 se sono dello stesso sesso.

$$- F_{\text{età}} : (X_{\text{età}}, Y_{\text{età}}) \rightarrow \frac{\min\{X_{\text{età}}, Y_{\text{età}}\}}{\max\{X_{\text{età}}, Y_{\text{età}}\}}$$

Associa alla coppia (età di X, età di Y) il rapporto tra l’età dell’utente più piccolo e quella dell’utente più grande. Se manca il dato per un utente, o per entrambi, il valore di similarità è 0.

$$- F_{\text{lingua}} : (X_{\text{lingua}}, Y_{\text{lingua}}) \rightarrow \{0,1\}$$

Associa alla coppia (lingua di X, lingua di Y) il valore 0 se parlano lingue diverse (o se per almeno uno dei due manca il dato), il valore 1 se parlano la medesima lingua.

$$- F_{\text{provincia}} : (X_{\text{provincia}}, Y_{\text{provincia}}) \rightarrow \{0,1\}$$

Associa alla coppia (provincia di X, provincia di Y) il valore 0 se risiedono in province diverse (o se per almeno uno dei due manca il dato), il valore 1 se risiedono nella medesima provincia.

¹³ La valutazione media del servizio s_j rispetto all’utente X è data dalla media pesata delle votazioni che ha fornito X per ogni caratteristica, in base all’importanza assegnata dall’utente X a ciascuna caratteristica.

- $F_{paese} : (X_{paese}, Y_{paese}) \rightarrow \{0,1\}$
 Associa alla coppia (paese di X, paese di Y) il valore 0 se risiedono in Stati diverse (o se per almeno uno dei due manca il dato), il valore 1 se risiedono nello stesso Stato.
- $F_{istruzione} : (X_{paese}, Y_{paese}) \rightarrow \{0,1\}$
 Associa alla coppia (istruzione di X, istruzione di Y) il valore 0 se hanno un background formativo differente (o se per almeno uno dei due manca il dato), il valore 1 se hanno ricevuto la medesima formazione.
- $F_{lavoro} : (X_{paese}, Y_{paese}) \rightarrow \{0,1\}$
 Associa alla coppia (paese di X, paese di Y) il valore 0 se lavorano in settori diversi (o se per almeno uno dei due manca il dato), il valore 1 se il settore è lo stesso.
- $F_{sport} : (X_{paese}, Y_{paese}) \rightarrow \{0,1\}$
 Associa alla coppia (sport di X, sport di Y) il valore 0 se praticano sport diversi (o se per almeno uno dei due manca il dato), il valore 1 se almeno uno degli sport praticati è lo stesso.

3. A questo punto si dispone di undici valori di similarità. Per ottenere i tre valori cercati (commerciale, demografico, sociale) si devono raggruppare i risultati delle undici funzioni per contesto e poi calcolarne i valori medi. Poiché all'interno di ciascun contesto, ogni funzione ha rilevanza differente, il calcolo da effettuare è una media pesata. A tale scopo si definiscono tre maschere (*weight mask*), una per ogni contesto, che determinano l'importanza di ciascuna funzione. Una proposta per le tre *weight mask* è la seguente:

- Per il contesto commerciale

$$\text{Mask}_C (F_{acquisti}; F_{rating}; F_{pesi}) = [0.30; 0.50; 0.20]$$

- Per il contesto demografico

$$\text{Mask}_D (F_{sesso}; F_{età}; F_{lingua}; F_{provincia}; F_{paese}) = [0.25; 0.25; 0.25; 0.05; 0.20]$$

- Per il contesto sociale

$$\text{Mask}_S (F_{istruzione}; F_{lavoro}; F_{sport}) = [0.40; 0.50; 0.10]$$

Per ogni gruppo, quindi, si applicano le maschere ai singoli valori di similarità e si sommano i risultati, nel seguente modo:

- Similarità commerciale:

$$S_C = F_{acquisti} \cdot M_C[1] + F_{rating} \cdot M_C[2] + F_{pesi} \cdot M_C[3]$$

- Similarità demografica:

$$S_D = F_{sesso} \cdot M_D[1] + F_{età} \cdot M_D[2] + F_{lingua} \cdot M_D[3] + F_{provincia} \cdot M_D[4] + F_{paese} \cdot M_D[5]$$

- Similarità sociale;

$$S_S = F_{istruzione} \cdot M_S[1] + F_{lavoro} \cdot M_S[2] + F_{sport} \cdot M_S[3]$$

dove $M_I[K]$ corrisponde al K -esimo elemento della maschera relativa al contesto I .

La funzione HowSimilarUser

I tre passi della routine *FindSimilarUser* appena descritti, costituiscono una funzione, che chiamiamo *HowSimilarUser*, la quale valuta la similarità di due utenti X e Y. La routine applica tale funzione iterativamente, al fine di confrontare un certo utente X con tutti gli altri utenti del sistema. La funzione, però, può essere applicata anche in altri contesti, come abbiamo visto nella descrizione del clustering partitivo (vedi *Definire le preferenze di utenti tipo*).

5.4.9 Il Recommender Engine

Il Recommender Engine ha il compito di generare i suggerimenti sulla base dei dati provenienti da utenti, servizi e dalla Collective Intelligence platform, tenendo conto degli algoritmi in esso contenuti e delle impostazioni personali di ogni utente. Infatti, il Recommender Engine è in grado di generare cinque tipologie diverse di suggerimenti, ma non le propone in misura uguale a tutti gli utenti, ciò dipende da alcuni fattori, come la tipologia di suggerimenti che l'utente predilige, oppure l'interfaccia utilizzata per l'accesso.

Le cinque categorie di suggerimenti generate dal Recommender Engine sono:

- *Suggerimenti User Based*: il RS consiglia all'utente i servizi preferiti dagli utenti con profilo simile al suo.
- *Suggerimenti Content Based*: il RS consiglia all'utente servizi simili a quelli che in passato ha valutato positivamente.
- *Suggerimenti Group Based*: il RS consiglia servizi in base al gruppo di appartenenza dell'utente.
- *Suggerimenti Superuser Based*: il RS consiglia all'utente i servizi preferiti dal suo superutente relativo e da quello del sistema.
- *Suggerimenti Popularity Based*: il RS consiglia i servizi più popolari relativamente a uno specifico intervallo temporale.
- *Suggerimenti Trust Based*: i consigli provengono direttamente dalle persone che l'utente conosce e di cui si fida.
- *Suggerimenti Context Based*: il RS consiglia servizi acquistati nel medesimo contesto in cui si trova in quel momento l'utente.

Pur sembrando una considerazione superflua, ritengo comunque opportuno specificare che si assume la regola, valida per tutti le categorie di suggerimenti, secondo la quale a un utente X non vengono suggeriti servizi che sono già presenti nel campo *Servizi_Visti* del vettore a lui associato.

Suggerimenti User Based

Sia X l'utente destinatario dei suggerimenti e sia $L_X = \{l_1, \dots, l_S\}$ la lista dei suoi S vicini, contenuta nel campo *Utenti_Simili* del vettore rappresentante X. All'interno della categoria User Based, definiamo due sottoclassi di suggerimenti: rating based e recommendation based.

La prima sottoclasse è basata sulle valutazioni individuali dei servizi fornite dai vicini di X, la seconda, invece, prende in esame i suggerimenti valutati positivamente dai vicini di X.

1. *Rating Based*. Per ogni utente l_i si considera la lista *Servizi_Valutati*, i cui elementi sono del tipo $(ID_Servizio, data_rating, rate)$, e si selezionano soltanto i servizi che hanno ricevuto un rating medio superiore ad una certa soglia (th_R). Indichiamo con $C_X(i) = \{(c_1(i), r_1(i)), \dots, (c_M(i), r_M(i))\}$ l'insieme delle M coppie (*servizi, rating*) selezionate dalla lista dell'utente l_i . Indicato con $Sim(X, l_i)$ il valore di similarità tra X e l_i , si moltiplicano gli r_j (per j da 1 a M) per $Sim(X, l_i)$, ottenendo $WR(c_j(i))$, cioè il rating del servizio $c_j(i)$ pesato rispetto alla similarità tra X e l_i .

$$WR(c_j(i)) = r_j(i) \cdot Sim(X, l_i)$$

Tale valore viene calcolato per tutti i servizi c_j dell'utente l_i , e il medesimo procedimento viene applicato a tutti gli utenti dell'insieme L_X . Alla fine del processo si ottiene un insieme $C_X = \{(c_1, WR(c_1)), \dots, (c_T, WR(c_T))\}$, contenente tutte le coppie (*servizio, rating pesato*), che costituiscono la lista dei suggerimenti candidati¹⁴. Il Recommender Engine, quindi, redige un elenco ordinato (per valore di *rating pesato* decrescente) degli elementi di C_X , e seleziona i primi α servizi¹⁵, proponendoli all'utente X come suggerimenti.

2. *Recommendation Based*. Per ogni utente l_i si considera la lista *Suggerimenti_Accettati*, i cui elementi sono del tipo $(ID_Servizio, data_suggerimento, qualità_suggerimento)$, e si selezionano soltanto i servizi che hanno ricevuto un rating medio superiore ad una certa soglia (th_S). Indichiamo con $C_X(i) = \{(c_1(i), q_1(i)), \dots, (c_M(i), q_M(i))\}$ l'insieme delle M coppie (*ID_servizio, qualità_suggerimento*) selezionate dalla lista dell'utente l_i . Indicato con $Sim(X, l_i)$ il valore di similarità tra X e l_i , si moltiplicano gli r_j (per j da 1 a M) per $Sim(X, l_i)$, ottenendo $WQ(c_j(i))$, cioè la qualità del suggerimento per il servizio $c_j(i)$ pesato rispetto alla similarità tra X e l_i .

$$WQ(c_j(i)) = r_j(i) \cdot Sim(X, l_i)$$

Tale valore viene calcolato per tutti i servizi c_j dell'utente l_i , e il medesimo procedimento viene applicato a tutti gli utenti dell'insieme L_X . Alla fine del processo si ottiene un insieme $C_X = \{(c_1, WQ(c_1)), \dots, (c_T, WQ(c_T))\}$, contenente tutte le coppie (*servizio, rating pesato*), che costituiscono la lista dei suggerimenti candidati¹⁴. Il Recommender Engine, quindi, redige un elenco ordinato (per valore di *rating pesato* decrescente) degli elementi di C_X , e seleziona i primi β servizi¹⁵, proponendoli all'utente X come suggerimenti.

Suggerimenti Content Based

Sia X l'utente destinatario dei suggerimenti e sia $V_X = \{v_1, \dots, v_H\}$ la lista degli H servizi valutati finora, contenuta nel campo *Servizi_Valutati* del vettore rappresentante X. Indichiamo con $C_X(v_i) = \{(c_{1i}, Sim(v_i, c_{1i})), \dots, (c_{Si}, Sim(v_i, c_{Si}))\}$ l'insieme delle S coppie (*ID_servizio_simile_a_v_i, valore_di_similarità*), elencate nel campo *Servizi_Simili* del vettore associato a v_i . Di tutti i c_{ji} vengono selezionati solo gli elementi con valore di similarità

¹⁴ Se nella lista C_X un servizio è ripetuto più volte, si prende in considerazione solo quello con valore di WR più alto.

¹⁵ I valori di α , β , θ e μ devono essere decisi in fase di implementazione del sistema, in base alla strategia di presentazione dei suggerimenti scelta

superiore ad una soglia minima (th_{Min}) sufficientemente restrittiva. Scelti secondo questa discriminante i servizi di $C_X(v_i)$, e effettuata questa operazione per ogni v_i dell'insieme V_X , si ottiene una lista $C_X = \{c_1, \dots, c_T\}$ dei servizi candidati.

Il campo rating dei vettori servizio ha la seguente struttura:

$$\left\{ \begin{array}{l} [[(proprietà_{11}, voto_{11}), \dots, (proprietà_{P1}, voto_{P1})], utente_1, data], \\ \dots, \\ [[(proprietà_{1N}, voto_{1N}), \dots, (proprietà_{PN}, voto_{PN})], utente_N, data] \end{array} \right\}$$

Per ottenere una valutazione specifica del rating medio dei servizi, pesandola sulla base dell'importanza che l'utente X assegna a ogni proprietà di un servizio, si deve tener conto del campo *Peso_Proprietà* del vettore associato all'utente X. La formula che permette di calcolare il rating medio del servizio c_i (relativo a X) è la seguente:

$$WR(i) = \sum_{j=1}^P w_j \cdot \left(\frac{1}{N} \sum_{u=1}^N voto_{ju}(i) \right)$$

dove $voto_{ju}(i)$ indica il rating della proprietà j , da parte dell'utente u , relativa al servizio c_i . Calcolato $WR(i)$ per tutti i servizi c_i dell'insieme C_X , il Recommender Engine ora può stilare una classifica di servizi, ordinata per rating decrescente, e scegliere i primi γ , come suggerimenti per l'utente X.

Suggerimenti Group Based

Come abbiamo visto in questa sezione, nel corso della trattazione dell'argomento *Associare servizi a gruppi di utenti*, determinati servizi possono essere particolarmente interessanti, su base statistica, per utenti con caratteristiche simili. Definiti i gruppi di utenti affini e determinati i servizi più apprezzati all'interno di ciascun insieme (abbiamo già visto che questo è uno dei compiti della CI platform), al Recommender Engine resta soltanto da individuare a quale gruppo appartiene l'utente X, destinatario dei suggerimenti e selezionare i primi θ servizi¹⁵.

Suggerimenti Superuser Based

Sia X un utente per il quale il RS deve generare i suggerimenti. Il Recommender Engine, per adempiere al suo dovere, necessita di determinare il superutente relativo a X e il superutente del sistema.

Partiamo dal primo obiettivo: individuare il superutente relativo di X. Si consideri l'insieme dei K utenti premium deluxe $D = \{D_1, \dots, D_K\}$. Si prendano, uno alla volta, tutti i D_i , e per ciascuno di essi si determini il numero di suggerimenti, contenenti una o più valutazioni di D_i , ricevuti da X e valutati positivamente. Se il numero rilevato è superiore alla metà dei suggerimenti totali ricevuti da X allora D_i diventa un candidato.

Alla fine, tra tutti i candidati, viene scelto come superutente relativo a X quello che ha contribuito al maggior numero di suggerimenti valutati positivamente da X.

Passando al secondo obiettivo, in base alla definizione di superutente assoluto, per prima cosa si deve individuare l'insieme dei K utenti premium deluxe $D = \{D_1, \dots, D_K\}$. Successivamente, selezionando uno alla volta tutti gli utenti deluxe D_i , si verifica quale tra tutti i candidati verifica la definizione. Sia $I = \{X_1, \dots, X_N\}$ l'insieme degli N utenti del sistema e supponiamo che il generico utente X_i abbia giudicato S_i suggerimenti contenenti una o più valutazioni di

D_i . Se almeno un terzo degli N X_i ha giudicato positivamente almeno $S_i/2$ suggerimenti allora D_i è un possibile candidato.

Tra tutti i candidati ottenuti, alla fine viene scelto come superutente del sistema quello che ha ricevuto il maggior numero di consensi.

Suggerimenti Popularity Based

Anche in questo caso la maggior parte del lavoro viene svolta dalla Collective Intelligence platform, alla quale è affidato l'onere di determinare i servizi di più popolari (vedi *Determinare i servizi di maggior successo* nella sezione precedente). Il compito del Recommender Engine, quindi è molto limitato e si riduce a tre azioni:

1. Scegliere un intervallo di tempo (*giorno, settimana, mese, anno, sempre*) rispetto al quale si vuole determinare i servizi più venduti, basandosi sulle preferenze dell'utente X .
2. Comunicare la scelta dell'intervallo (o degli intervalli) alla CI platform.
3. Selezionare i primi μ ¹⁵ servizi della lista (o delle liste) ricevuta come risposta dalla CI platform.

Suggerimenti Trust Based

Questa tipologia di suggerimenti è stata pensata per la SeReS CI App di Facebook. Una funzionalità aggiuntiva che proviene dall'integrazione del RS nel Social Network è la possibilità di condividere le proprie preferenze con gli amici virtuali. Infatti, come dimostrato in *How Online Recommendations Can Benefit From Social Networking* [82], i suggerimenti provenienti direttamente dalle conoscenze, se pur limitati nel numero (perché richiede una partecipazione attiva), tuttavia vengono presi in considerazione più di quelli generati secondo altre tecniche. Si noti che il Recommender Engine, in questo caso, non ha un ruolo attivo nella produzione di suggerimenti, agisce soltanto da tramite tra l'utente e i suoi amici.

Suggerimenti Context Based

Questa categoria di suggerimenti è stata pensata in modo specifico per l'utilizzo del Recommender System con dispositivi mobili che integrano un GPS, i quali permettono la geo-localizzazione degli utenti. Il *context* di cui si parla in questo caso è formato da due componenti: coordinate GPS e orario. Le opportunità offerte dalla conoscenza del *context* sono di due tipi: indicare all'utente i servizi acquistati più spesso nel contesto in cui si trova; avvisare l'utente della presenza di servizi vicini alla posizione che occupa in quel momento.

1. *I servizi più visti nel contesto attuale.* La tipologia di suggerimento rientra nei *Popularity Based*, con una differenza: si effettua una scrematura dei servizi a monte, prendendo in considerazione soltanto quelli che appartengono ad un contesto simile a quello in cui si trova attualmente l'utente. La misura di similarità dipende da due indici: l'indice di vicinanza spaziale (*SPI*) e l'indice di similarità temporale (*TSI*). Le impostazioni di default del Recommender Engine prevedono che i suggerimenti abbiano un valore di *SPI* inferiore a ϵ km e *TSI* massimo pari a η secondi. L'utente, come vedremo parlando delle modalità di presentazione dei suggerimenti (vedi la sezione *5.4.10 Proposizione dei suggerimenti*) può agire su questi parametri, imponendo condizioni più restrittive, oppure più rilassate.

Spatial Proximity Index (SPI)

Date le coordinate GPS (*latitudine, longitudine*) $C_X = (c_1(X), c_2(X))$, relative alla posizione dell'utente X e $C_Y = (c_1(Y), c_2(Y))$, relative alla posizione in cui è stato acquistato un certo servizio Y , il valore di *SPI* è dato dalla formula:

$$AC \left(S \left(\frac{\pi \cdot c_1(Y)}{180} \right) \cdot S \left(\frac{\pi \cdot c_1(X)}{180} \right) + C \left(\frac{\pi \cdot c_1(Y)}{180} \right) \cdot C \left(\frac{\pi \cdot c_1(X)}{180} \right) \cdot C \left(\left| \frac{\pi \cdot c_2(X)}{180} - \frac{\pi \cdot c_2(Y)}{180} \right| \right) \right) \cdot R$$

la quale indica, in km, la distanza in linea d'aria di Y da X. I simboli AC, S, C, R indicano rispettivamente le funzioni ArcCos, Sen, Cos, e l'approssimazione del Raggio terrestre (6371 km).

Temporal Similarity Index (TSI)

Dato l'orario riferito all'utente X, $H_X = (hh_X, mm_X, ss_X)$, e quello relativo all'acquisto di un certo servizio Y, $H_Y = (hh_Y, mm_Y, ss_Y)$, il valore di TSI è dato dalla formula:

$$TSI = (hh_X \cdot 3600 + mm_X \cdot 60 + ss_X) - (hh_Y \cdot 3600 + mm_Y \cdot 60 + ss_Y)$$

la quale indica, in secondi, la differenza di orario tra H_X e H_Y .

2. *Servizi vicini alla posizione attuale.* L'obiettivo di questo tipo di suggerimenti è di consigliare servizi in modo proattivo, cioè il RS non aspetta che sia l'utente a manifestare l'esigenza, bensì propone, in base ai gusti dell'utente, i servizi disponibili nelle vicinanze. In questo caso la modalità di generazione dei suggerimenti appartiene sia alla categoria *User Based* che a quella *Content Based*, tuttavia vengono presi solo in considerazione quelli disponibili entro una certa distanza fissata ϵ , espressa in chilometri. Come nel caso precedente, l'utente può agire sul valore di ϵ in base alle sue esigenze.

5.4.10 Proposizione dei suggerimenti

Come detto nella sezione precedente, il Recommender Engine genera sette tipi di suggerimenti: *User Based*, *Content Based*, *Group Based*, *Superuser Based*, *Popularity Based*, *Trust Based* e *Context Based*¹⁶. La strategia di presentazione di questi suggerimenti, come ad esempio la scelta della tipologia da privilegiare per un certo utente, dipende dall'interfaccia utilizzata (sito web, widget, Facebook, mobile app), cioè le proposte sono modellate sulla base delle caratteristiche della piattaforma di accesso al sistema, sfruttando al massimo le opportunità che ciascuna di esse offre. Indipendentemente dalla tipologia ci sono comunque delle caratteristiche valide per tutte le interfacce. Partiamo dal descrivere proprio questi elementi:

- *Generazione dei suggerimenti.* Il Recommender Engine che genera i suggerimenti è unico (o tutt'al più distribuito) e la generazione di alcune tipologie di suggerimenti, come i *Group Based*, *Superuser Based* e *Popularity Based*, si possono affidare ad un processo in background, per limitare il carico sulla macchina. Per ogni categoria il RS genera R suggerimenti. In base alle strategie adottate poi ne vengono presentati uno, più di uno o anche nessuno.
- *Eliminazione di un suggerimento.* Dopo che un suggerimento di tipo T è stato selezionato, viene rimosso dalla lista degli R suggerimenti della tipologia T. Inoltre, se l'utente accede

¹⁶ Gli ultimi due tipi sono riservati, rispettivamente, alle applicazioni di Facebook e alla Mobile App.

tre volte al RS e non seleziona mai un certo suggerimento proposto, il sistema interpreta questa mancanza come una segno di disinteresse per il suggerimento e lo rimuove dalla suddetta lista. La medesima cosa accade, indipendentemente dagli accessi, se passa una settimana dalla generazione senza che l'utente visualizzi la proposta.

- *Frequenza di generazione dei suggerimenti.* In generale, senza necessità di scadenze precise, quando la lista di R suggerimenti di una categoria specifica si riduce della metà, il RS si attiva e ne genera altri $R/2$. Tuttavia per le categorie legate ad un intervallo temporale (es. Popularity Based), la frequenza di aggiornamento è ben definita: ogni ora per i suggerimenti giornalieri, ogni giorno per i suggerimenti settimanali, mensili e annuali.

I suggerimenti nel Web Site

Il sito web permette un'esplorazione approfondita dei suggerimenti generati dal Recommender System, quindi si può ricorrere a una struttura gerarchica, simile a quella proposta da Amazon (vedi capitolo 3). I suggerimenti presenti nella prima pagina hanno il compito di attrarre l'interesse dell'utente, quindi devono avere come oggetto servizi che rispecchino effettivamente le sue preferenze, ma per non risultare troppo monotoni devono proporre anche i servizi di maggior successo. Quindi, al primo livello troviamo in primo piano 5 suggerimenti *Content Based* e 5 *Superuser Based*, e un po' più defilati 10 suggerimenti *Popularity Based*, suddivisi in *Servizi del giorno* (5) e *Servizi del mese* (5).

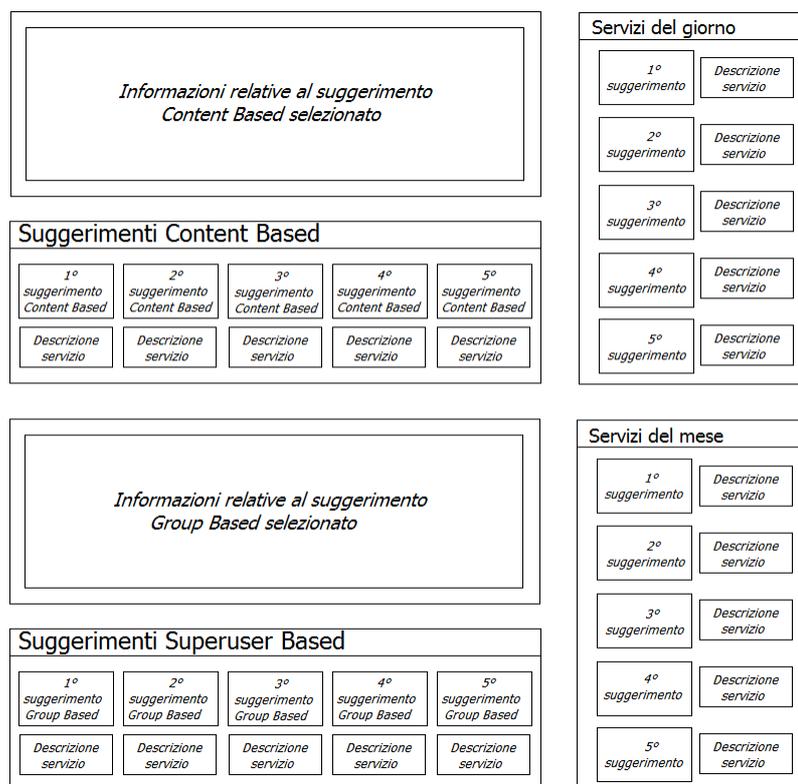


Figura 5.28: i suggerimenti nel sito web (1° livello)

Selezionato uno dei servizi proposti, si arriva in una pagina dedicata, contenente la descrizione del servizio, il rating, le recensioni, i tag associati, ed eventuali immagini e video. A questo livello i suggerimenti proposti hanno un duplice scopo: la raccomandazione di servizi simili a quello scelto, l'estensione del ventaglio di possibilità con servizi non correlati

all'attuale ricerca. A tale scopo vengono mostrati in primo piano 5 suggerimenti *Content Based*, e più defilati 5 suggerimenti *User Based* e 5 *Group Based*.

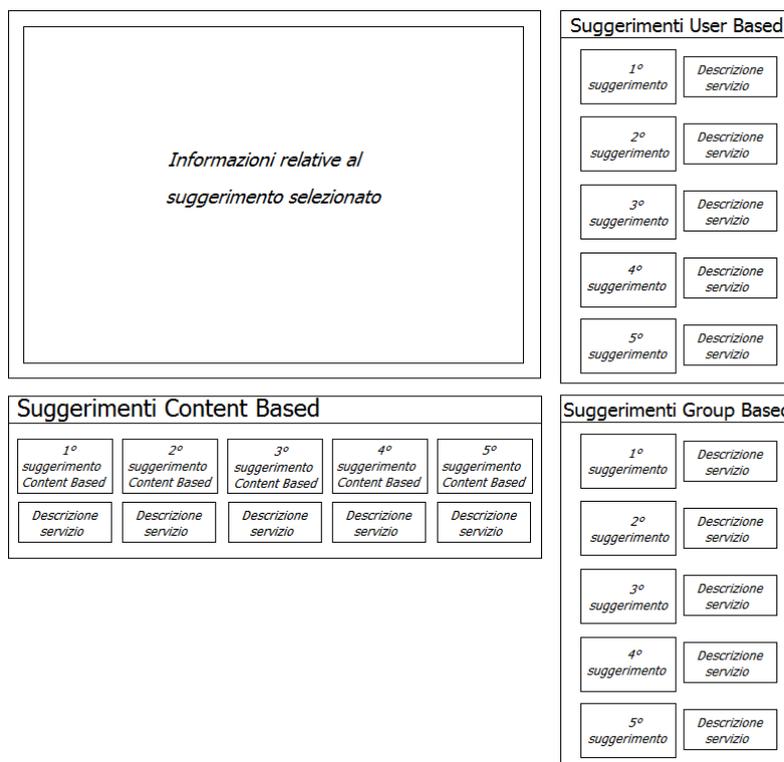


Figura 5.29: i suggerimenti nel sito web (2° livello)

I suggerimenti nel Widget

Il widget, per quanto concerne i suggerimenti, può essere considerato uno strumento di supporto al web site, infatti memorizza i suggerimenti ricevuti, permette di rivederli e di valutarli. Non essendo stato pensato per una esplorazione approfondita, il widget deve proporre suggerimenti efficaci. Non esiste, tuttavia, una categoria di suggerimenti migliore in assoluto, dipende da utente a utente. Per tale motivo è utile calcolare per ogni profilo quale tipologia di suggerimenti ha maggior successo. Per determinare questa preferenza si ricorre ai campi *Suggerimenti Accettati* e *Suggerimenti Rifiutati* e per ogni tipologia α di suggerimenti si calcola la frazione di suggerimenti accettati sul totale di quelli di tipo α proposti:

$$utilità \alpha = \frac{\#suggerimenti \text{ accettati } \alpha}{\#suggerimenti \text{ accettati } \alpha + \#suggerimenti \text{ rifiutati } \alpha}$$

Vengono quindi proposti 5 suggerimenti della categoria preferita. Se la tipologia più utile si rivela essere la *Popularity Based*, i consigli sono riferiti ai servizi più popolari del giorno.

I suggerimenti in Facebook:

In Facebook si può utilizzare la stessa strategia adottata per il sito web, ma con una opportunità in più per gli utenti: condividere le proprie preferenze con gli amici virtuali. Quindi, strutturati i suggerimenti in modo gerarchico, le proposte *Trust Based* vengono collocate in primo piano, sopra quelle *Content Based* e *Superuser Based*.

I suggerimenti per la Mobile App

I suggerimenti della Mobile App sono organizzati secondo due paradigmi: *tradizionale* e *topologico*. Il paradigma *tradizionale* ricalca sostanzialmente lo schema di erogazione dei suggerimenti del widget per PC, con la funzionalità aggiuntiva dei suggerimenti *Context Based*.

Il paradigma topologico, invece, propone suggerimenti esclusivamente *Context Based*, indicandoli direttamente sulla mappa geografica. I servizi proposti sono 5 per volta, e cambiano in base alla scala con cui si visualizza la mappa. L'utente può decidere se visualizzare i *servizi più visti nel contesto attuale* oppure i *servizi vicini alla posizione attuale* (vedi *Suggerimenti Context Based* nella sezione precedente).



Figura 5.30: i suggerimenti per la mobile app

5.5 Possibili estensioni

Le funzionalità del Recommender System presentato possono essere estese e migliorate, ricorrendo ad alcune soluzioni che non hanno trovato spazio in questa tesi o per motivo di tempo o perché non rientravano nelle prerogative di progettazione.

Includere un meccanismo automatico di controllo dei feedback

Il RS permette agli utenti di inserire descrizioni, rating, recensioni e tag riguardanti i servizi, tuttavia il controllo dei dati immessi nel sistema è limitato alla verifica delle anomalie da parte dei *collaboratori*. Una possibile miglioria per rendere più efficace il RS, quindi, potrebbe essere l'introduzione di una funzione di controllo automatico dei feedback.

Specificare gli effetti dei feedback sui suggerimenti

Anche i suggerimenti, come spiegato nella sezione 5.4.7 (*I feedback degli utenti*), possono essere valutati dagli utenti che li ricevono. Tuttavia non è stato specificato quali siano gli effetti dell'approvazione o meno della proposta ricevuta sulle azioni successive del Recommender Engine. Si potrebbe pensare, ad esempio, di includere un meccanismo che

agisce sui pesi delle proprietà, oppure che deduce preferenze in base alle caratteristiche dei servizi suggeriti.

Determinare la rilevanza del feedback di un utente

Le aggregazioni dei rating degli utenti determinano sia la qualità dei servizi che quella dei Service Provider. Tuttavia non tutti i giudizi espressi hanno la stessa valenza. Se un professionista fornisce un feedback per un servizio di cui è esperto, questa votazione dovrebbe avere maggiore importanza rispetto a quella di un utente che ha usato un determinato servizio occasionalmente. La questione, quindi, è duplice: innanzi tutto è necessario elaborare un metodo per determinare quali sono le competenze di ciascun utente; secondariamente si devono pesare i feedback degli utenti in base alle conoscenze specifiche del servizio.

Estrarre informazioni implicite dai profili Facebook

I Collective Intelligence Agents permettono al sistema di accedere ai profili Facebook degli utenti che li utilizzano. Ciò che manca, però, è uno strumento di raccolta dei dati disponibili, in grado di catalogare le informazioni implicite, insite ad esempio nei link che un utente pubblica nella propria pagina, oppure nei “mi piace” assegnati a determinate pagine o contenuti. Queste social metadata sono fonti di inestimabile valore che se organizzati correttamente permettono una precisa e profonda conoscenza degli utenti.

Rendere più aperto il sistema

Il sistema di feedback di SeReS CI è stato studiato in modo tale che soltanto gli utenti che effettivamente acquistano un servizio possano poi aggiungere contenuti e commenti personali all'elemento che lo rappresenta nel sistema. A tale scopo, infatti, si utilizza un codice univoco che il Service Provider fornisce all'utente, tramite il RS, all'atto del pagamento del servizio, e che deve essere esibito ogni qual volta si aggiornano le risorse relative a quel servizio.

Tutto ciò rende SeReS CI un sistema chiuso, e se da una parte questo è sintomo di presunta qualità, dall'altra limita la quantità di contributi che potrebbe ricevere, sia da utenti del sistema, sia da individui esterni.

Per quanto riguarda gli utenti registrati, la ricerca di una possibile soluzione rimanda alla questione di come determinare la rilevanza dei feedback in base alle competenze. Ma non solo, infatti sarebbe utile anche poter far valutare agli utenti l'utilità di un feedback, in modo da definire la qualità degli utenti in base alla bontà dei loro pareri.

Volgendo l'attenzione agli utenti sprovvisti di profilo, invece, il problema è molto più insidioso e merita un'attenta valutazione sulla base dei rischi e dei benefici che comporterebbe un'apertura così radicale.

Analizzare le relazioni sociali

L'integrazione del RS con Facebook, tra le tante opportunità, consente anche di conoscere le relazioni sociali di ciascun profilo. Analizzando la rete sociale che si può ottenere integrando i dati sulle amicizie di tutti gli utenti, si possono ricavare informazioni relative a gruppi e dedurre dinamiche globali. Esiste una branca della sociologia che ha come scopo proprio quello di individuare schemi e pattern all'interno delle reti sociali, chiamata Social Network Analysis (SNA).

Capitolo 6

Esempio di utilizzo di SeReS CI: Integrazione in un sito di viaggi

In questo capitolo si è preso in esame il sito Booking.com, che offre un servizio di prenotazione di Hotel e numerose informazioni sulle strutture, compresi i feedback dei clienti. Partendo dall'esigenza degli utenti di avere suggerimenti personalizzati sui possibili alberghi in cui andranno a soggiornare, è stata definita una integrazione di SeReS CI all'interno del sito considerato. Nel primo sottocapitolo si esamina lo scenario di applicazione, eseguendo un'analisi del sistema esistente. Nel secondo sottocapitolo si descrivono le variazioni determinate dall'utilizzo del RS. Nel terzo sottocapitolo, infine, si propone un'analisi comparativa.

6.1 Scenario di applicazione

Booking.com è una Online Travel Agency (OTA), cioè una società che consente, tramite un Computer Reservations System (CRS) la prenotazione di strutture alberghiere in Internet. Il sito web, oltre a fornire un servizio di prenotazione che non prevede alcuna commissione e a tariffe spesso agevolate, è anche un portale informativo gratuito, dove gli utenti possono leggere le descrizioni delle strutture e visionare le recensioni e le valutazioni dei clienti.

Attualmente ha un bacino di utenza pari a 2 175 579 utenti unici giornalieri [83], distribuiti principalmente in 33 paesi, con UK (11.3%), Italia (10.4%), e Spagna (9.2%) che si pongono come principali fruitori [84]. Gli hotel presenti nel database di Booking.com sono oltre 120 000, sparsi in 99 paesi. I collaboratori in tutto il Mondo sono circa 1600 [85].

Il modello di business dell'azienda si basa sulle commissioni (variabili da un minimo del 1% a un massimo del 50%, sul totale pagato dal cliente) che gli albergatori versano per avere una posizione privilegiata nella pagina dei risultati. I clienti, invece, non hanno alcun costo aggiuntivo, la registrazione è gratuita e di solito pagano direttamente in hotel. Solo per ricevere speciali sconti, in alcuni casi, si richiede un pagamento anticipato.

La proposta è di integrare all'interno di Booking.com il sistema SeReS CI, fornendo un valore aggiunto al sito, con suggerimenti mirati per i suoi utenti.

6.1.1 Caratteristiche del contesto

Per integrare al meglio SeReS CI nel sito Booking.com è utile individuare chi sono effettivamente gli attori presenti in questo contesto, procedendo a un'analisi delle caratteristiche di utenti, servizi e Service Provider.

Gli utenti

In generale la segmentazione del mercato del turismo prevede 4 categorie di clienti [86]:

- turisti singoli (*leisure individual*);
- gruppi (*leisure group*);
- uomini d'affari (*corporate individual*);
- partnership con aziende (*corporate marketing*).

Booking.com, si rivolge principalmente ai turisti, tralasciando il settore corporate, come testimonia il tipo di classificazione utilizzata per distinguere i suoi utenti:

- Famiglie con figli piccoli;
- Famiglie con figli grandi;
- Coppie giovani;
- Coppie mature;
- Gruppi di amici;
- Viaggiatori singoli.

I servizi

In questo contesto si possono distinguere tre tipi di servizi:

- il servizio di prenotazione fornito da Booking.com;
- i servizi di carattere informativo presenti nel sito, la cui origine è riconducibile sia a Booking.com, sia agli utenti;
- gli hotel, intesi come servizi di alloggio temporaneo, erogati dagli albergatori;

Di queste tre tipologie, ovviamente, sarà oggetto dei suggerimenti proposti dal RS soltanto l'ultima, quindi da qui in poi qualsiasi riferimento ai servizi (e quindi anche ai Service Provider), è da intendersi rivolto a servizi di alloggio temporaneo.

Gli hotel sono stati classificati in base a tre parametri:

- star rating (Numero di stelle);
- tipo di alloggio;
- catena di appartenenza.

I Service Provider

La proprietà di un albergo può essere di un individuo o di un gruppo di persone. In entrambi i casi è possibile che il proprietario (o i proprietari) possieda più strutture. Si distinguono quindi due tipi di Service Producer: i *singoli albergatori* e le *catene*, intendendo con il primo termine uno o più individui che possiedono un hotel, e con il secondo termine la compagnia che possiede la catena di hotel.

In questo contesto Service Producer e Service Provider non sempre coincidono, anzi il più delle volte il proprietario non è coinvolto nell'erogazione del servizio, che è invece delegata al personale della struttura alberghiera.

All'interno del sito Booking.com l'unico riferimento riguardante i Service Provider è il rating dello staff, che però fornisce una valutazione complessiva del personale senza entrare nel merito delle singole caratteristiche qualitative, come la correttezza, la professionalità o la

disponibilità verso il cliente. Un'idea più precisa dell'entità erogante può essere estrapolata dalle recensioni dell'utente, ma tali informazioni sono sparse e disorganizzate.

6.1.2 Analisi del sistema esistente

Vediamo, ora, com'è strutturato il sito web esistente, analizzando il profilo utente, la home page, la query di ricerca, la descrizione degli hotel, i feedback degli utenti, le informazioni utili e i suggerimenti.

Il profilo utente

La registrazione al sito prevede l'inserimento dell'indirizzo email, del nome e cognome, e di una password. In automatico vengono assegnati lingua e paese di residenza (probabilmente attraverso il rilevamento dell'indirizzo IP). Dopo aver eseguito il login è possibile specificare anche l'indirizzo di residenza e il numero di telefono.

Gli hotel consultati non vengono registrati nel profilo, bensì in un cookie, ma questo comporta lo svantaggio che utilizzando computer (o anche browser) diversi si perde l'informazione.



I miei dati

Utilizzeremo questi dati per velocizzare le tue future prenotazioni.

Nome
marco zanchetta

e-mail
mzank84@libero.it

Indirizzo
Italia

Lingua
Italiano

Paese
Italia

Telefono

Figura 6.1: profilo utente in Booking.com

La homepage

L'elemento centrale della homepage è il motore di ricerca (*search engine*), nel quale si possono specificare il luogo in cui si desidera prenotare l'hotel (*Dove?*), la data di arrivo (*data check-in*) e quella in cui si lascerà l'hotel (*data check-out*). Inoltre sono presenti due opzioni per definire fin da subito alcuni dettagli della ricerca: *Non ho ancora deciso le date del mio soggiorno* e *Viaggio in gruppo/con bambini*.

Immediatamente sotto in *search engine* troviamo una sezione dedicata alle destinazioni più richieste, nella quale sono presentate le 8 città in cui ci sono state maggiori prenotazioni di hotel. Per ciascuna località sono indicati nazione, numero di hotel disponibili e a quando risale l'ultima prenotazione.

A destra degli elementi principali (*search engine* e *destinazioni più richieste*) c'è un elenco degli ultimi 5 hotel visionati dall'utente. Come spiegato in sede di presentazione del profilo, questa informazione è memorizzata in un cookie.

Nella parte più bassa della pagina, visibile soltanto facendo scorrere la barra laterale del browser, si possono visualizzare la mappa dei paesi in cui Booking.com è presente, il form di iscrizione alla newsletter, un riquadro contenente l'elenco dei vantaggi forniti dal sito, e tre destinazioni tra le più ambite (in base alla provenienza dell'utente), per ognuna della quale si indicano cinque sistemazioni.

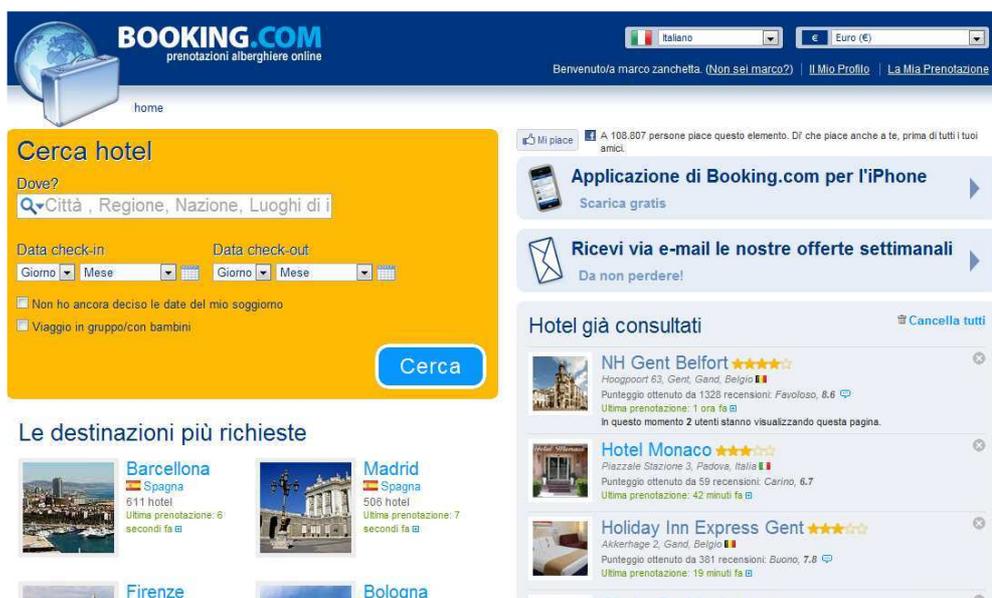


Figura 6.2: homepage di Booking.com

La query di ricerca

Soffermandoci sull'elemento principale della homepage, il search engine, vediamo come un utente arriva a trovare l'hotel che desidera.

Innanzitutto inserisce i parametri di ricerca elencati precedentemente. Questa operazione porta l'utente in una pagina contenente l'elenco degli hotel disponibili nel luogo e nel periodo che ha specificato. Si noti che se il nome della destinazione è ambiguo vengono proposte una serie di possibili alternative.

Il secondo passo consiste nel raffinare i risultati della ricerca. L'elenco proposto inizialmente è ordinato mettendo ai primi posti gli hotel più quotati. In realtà, come spiegato nella presentazione del sito, gli albergatori pagano una commissione a Booking.com per avere una posizione privilegiata in questa lista. L'utente può decidere di riordinare tale elenco in base a una delle sei caratteristiche disponibili, oppure restringere le opzioni proposte, specificando ulteriori parametri:

- *Riordino della lista.* L'utente seleziona una delle opzioni di ordinamento tra *Più quotati*, *Disponibilità*, *Prenotati di Recente*, *Stelle*, *Tariffa*, *Valutazione clienti*. A parità di valore per la classificazione specificata, vige comunque la regola che chi paga le commissioni più alte ha diritto a una posizione migliore.
- *Specifica di nuovi parametri.* Le tipologie di parametri che si possono indicare per restringere il campo della ricerca sono sette: *fascia di prezzo*, *star rating* (numero di stelle), *tipo di alloggio*, *servizi accessori* (forniti dall'hotel), *tema dell'hotel* (romantico, per famiglie, di lusso, etc.), *posizione* (quartiere in cui si trova), *catena di appartenenza*.

Terminata la fase di affinamento della ricerca, l'utente può iniziare a prendere in considerazione le singole alternative, esplorando le informazioni disponibili nelle relative pagine. Per ogni hotel, le pagine dedicategli nel sito sono due: *Descrizione e disponibilità* e *Giudizi del cliente*.

Trovato l'hotel che più lo soddisfa, l'utente può, infine, procedere alla prenotazione.

Cerca hotel
Dove? Q Gand
Distanza: seleziona...
Data check-in: Mar 19 - Aprile '11
Data check-out: Mer 20 - Aprile '11
Cerca

43 Hotel trovati a Gand, 37 Disponibili, Risultati 1 – 20

Ordina per: Più Quotati | Disponibilità | Prenotati Di Recente | Stelle | Tariffa | Valutazione Clienti

NH Gent Belfort ★★★★★ Favoloso, 8.6
Hooppoort 63, Gent, Gand • Visualizza mappa
L'hotel NH Gent Belfort è situato vicino allo storico palazzo del municipio di Ghent. Ospita alloggi eleganti, un raffinato ristorante e una palestra in loco comprensiva di sauna. [Maggiori Informazioni](#)
Questo hotel sarà presto sold-out!
Ultima prenotazione: 7 minuti fa

Tipologie Disponibili	Persone	Disponibilità	Tariffa
Camera Matrimoniale Standard Cancellazione GRATUITA	2	5 camere rimaste	€ 129 - € 119
Appena prenotato			
Junior Suite Ultima camera! Cancellazione GRATUITA	2	Ultima occasione! 1 camera rimasta	€ 179
Camera Matrimoniale Superior Cancellazione GRATUITA	2	5 camere rimaste	€ 144 - € 134
Camera Matrimoniale Deluxe Cancellazione GRATUITA	2	5 camere rimaste	€ 159 - € 149

Ghent Marriott Hotel ★★★★★ Eccellente, 9
Korenlei 10, Gent, Gand • Visualizza mappa
L'hotel Marriott a Gand è situato sulle rive del fiume Lys e si affaccia sul Korenlei. La struttura coniuga un design moderno con il fascino di edifici storici. [Maggiori Informazioni](#)
Ultima prenotazione: ieri

Tipologie Disponibili	Persone	Disponibilità	Tariffa
Camera Qualità Marriott Cancellazione GRATUITA	2	4 camere rimaste	€ 199 - € 139

Holiday Inn Express Gent ★★★★★ Buono, 7.8
Akkerhage 2, Gand • Visualizza mappa
L'hotel offre ampie camere con prima colazione e connessione WiFi gratuite vicino all'autostrada E17, a soli 4 km dal centro della città medievale di Gent. [Maggiori Informazioni](#)
Ultima prenotazione: 56 minuti fa

Tipologie Disponibili	Persone	Disponibilità	Tariffa
Camera Standard con Divano-Letto Cancellazione	2	Disponibile	€ 129 - € 109

Cerca per:
Fascia Di Prezzo (a notte)
Star Rating
Tipo Di Alloggio
Servizi
Tema Dell'hotel
Quartieri
Catene

Figura 6.3: i risultati di una query di ricerca in Booking.com

La descrizione degli hotel

Come abbiamo appena visto, le pagine dedicate ad un hotel sono due: *Descrizione e disponibilità* e *Giudizi del cliente*. Nella pagina descrittiva (la prima delle due) si possono individuare alcune sezioni:

- *Descrizione*. Presenta informalmente l'hotel attraverso una descrizione, una galleria di immagini e video, e la valutazione media ricevuta dagli utenti.
- *Disponibilità*. Descrive le camere disponibili per il periodo specificato nella query, le condizioni tariffarie e il prezzo praticato.
- *L'hotel e i suoi servizi*. Elenco dei servizi accessori offerti dall'hotel, come ad esempio la connessione ad Internet, il deposito bagagli, l'aria condizionata o le cassette di sicurezza.
- *Condizioni dell'hotel*. Specifica le norme generali dell'hotel, come gli orari di check-in e check-out e i metodi di pagamento accettati.
- *Ristorazione* (sezione opzionale). Qualora ci sia un ristorante all'interno dell'hotel, in questa sezione sono elencati gli orari di apertura, le caratteristiche dei piatti cucinati ed eventuali offerte.
- *Informazioni sulla zona* (sezione opzionale). Qui i proprietari dell'hotel possono inserire informazioni riguardanti la zona in cui è situata la loro struttura.
- *Informazioni importanti* (sezione opzionale). Sezione
- *Termini e condizioni generali*.

L'albergatore può decidere se inserire i dati relativi alle sezioni contrassegnate con il termine *opzionale*, mentre è obbligato a inserire i tutti gli altri dati.

Oltre alle sezioni descritte troviamo anche, defilati nella parte destra della pagina, alcuni elementi informativi, riguardanti i punti di interesse nei dintorni dell'hotel, come monumenti, musei, parchi, stazioni e aeroporti.

NH Gent Belfort ★★★★★  [Visualizza mappa](#)

Hoogpoort 63, Gent, 9000 Gand  [Visualizza mappa](#)

 Mi piace  A 76 persone piace questo elemento. Di che piace anche a te, prima di tutti i tuoi amici.

Descrizione e disponibilità **Giudizi del cliente**



L'hotel NH Gent Belfort è situato vicino allo storico palazzo del municipio di Ghent. Ospita alloggi eleganti, un raffinato ristorante e una palestra in loco comprensiva di sauna.

Le camere dell'Hotel NH Gent Belfort sfoggiano pavimenti in legno e mobili di design. Presentano una TV a schermo piatto e un ampio bagno fornito di sapone e shampoo in omaggio.

Il ristorante Van Artevelde, caratterizzato da una calda atmosfera, propone specialità tradizionali della cucina locale. Il bar è un luogo d'incontro ideale in cui gustare la birra belga e squisiti cocktail.

Nelle giornate più calde potrete godervi il sole sulla terrazza esterna dell'hotel. Se desiderate rilassarvi dopo una lunga giornata, il lounge-bar sito in giardino è il posto che fa al caso vostro.

Da questa posizione privilegiata potrete raggiungere facilmente a piedi tutte le attrazioni principali, tra cui la chiesa di San Nicola o il centro commerciale Post Plaza.

Camere: 174. Catena Alberghiera : NH Hoteles.



Favoloso, 8.6
Punteggio ottenuto da [1328 recensioni](#)

"The staff were excellent. The breakfast buffet was very good and the restaurant service excellent."

 **James, Loughborough**
19 marzo 2011

Figura 6.4: la sezione *Descrizione* di un hotel

I feedback degli utenti

Le tipologie di feedback che un utente può esprimere sono il rating e la recensione, e possono essere inseriti solo da chi ha effettivamente soggiornato nell'hotel. Booking.com, qualche giorno dopo la data del check-out invia una mail al cliente, all'interno della quale c'è un link da utilizzare per inserire (contestualmente) il rating e la recensioni, e questo è l'unico modo possibile per fornire feedback sull'hotel.

Per quanto riguarda la visualizzazione dei giudizi da parte degli utenti che vogliono avere maggiori informazioni su un hotel, essa si può circoscrivere in base alla categorie di clienti che le hanno espresse (*Famiglie con figli piccoli; Famiglie con figli grandi; Coppie giovani; Coppie mature; Gruppi di amici; Viaggiatori singoli*).

Rating

Le caratteristiche qualitative degli hotel, considerate da Booking.com, e su cui gli utenti possono esprimere una valutazione, sono:

- Pulizia;
- Comfort;
- Posizione;
- Servizi accessori;
- Staff;
- Rapporto qualità-prezzo.

La votazione minima è 1 e la massima 10. Calcolando le medie dei voti espressi da tutti gli utenti per ciascun attributo si ottengono sei valori, che esprimono i livelli qualitativi dell'hotel per ogni caratteristica.

In base al rating complessivo (determinato come media aritmetica dei sei valori), viene assegnato all'hotel un giudizio tra quelli dell'elenco seguente (tra parentesi è specificato l'intervallo di rating corrispondente):

- Scarso (da 1 a 3.9)
- Deludente (da 4 a 4.9)
- Discreto (da 5 a 5.4)
- OK (da 5.5 a 5.9)
- Carino (da 6 a 6.9)
- Buono (da 7 a 7.9)
- Ottimo (da 8 a 8.5)
- Favoloso (da 8.6 a 8.9)
- Eccellente (da 9 a 9.4)
- Eccezionale (da 9.5 a 10)

Si noti che viene assegnato un rating complessivo all’hotel (e quindi un giudizio) soltanto se è stato valutato da almeno 5 utenti.

Recensioni

Gli utenti, dopo aver soggiornato in un Hotel prenotato con Booking.com, hanno la possibilità di scrivere una recensione, con commenti differenziati per le caratteristiche positive e per quelle negative. I primi, nella visualizzazione agli altri utenti, sono indicate con il simbolo + all’interno di un quadrato verde, le seconde con il simbolo – all’interno di un quadrato rosso. Oltre alla descrizione, il cliente può indicare anche la categoria di viaggiatore alla quale appartiene e il luogo di provenienza.

NH Gent Belfort ★★★★★

Hoogpoort 63, Gent, 9000 Gand [Visualizza mappa](#) [Prenota ora](#)

Mi piace A 76 persone piace questo elemento. Di che piace anche a te, prima di tutti i tuoi amici.

Descrizione e disponibilità **Giudizi del cliente**

I giudizi dei clienti che hanno soggiornato presso **NH Gent Belfort**.
I clienti descrivono la loro esperienza presso questo hotel.

Totale 8.6	Pulizia 8.7	Visualizza valutazione di:
Comfort 8.8	Posizione 9.7	Tutti gli ospiti 1328
Servizi 8.4	Staff 8.4	Famiglie con figli piccoli 56
Rapporto qualità-prezzo 8		Famiglie con figli grandi 39
		Coppie mature 439
		Gruppi di amici 198
		Coppie giovani 425
		Viaggiatori singoli 171

Giudizi individuali degli ospiti di NH Gent Belfort
I giudizi dei clienti sono classificati in ordine di lingua e di data (massimo 25 giudizi per pagina)

Pagina precedente Risultati 1 - 25 (Totale 1328) Pagina successiva

Cristian
Gruppo di amici
Noceto, Italia
05 aprile 2011
+ la posizione centrale, il confort delle camere. **10**

Anna
Gruppo di amici
parma, Italia
30 marzo 2011
+ la posizione è ottima e in particolare abbiamo apprezzato la colazione che è eccezionale. Comodissimo avere il parcheggio interno. **8.8**

Daniele
Viaggiatore singolo
Milano, Italia
27 marzo 2011
+ ottima posizione per la visita del centro città
- la griglia in leno che separava la zona soggiorno dalla camera della mia suite non veniva spolverata da anni. **6.7**

Figura 6.5: la pagina *Giudizi del cliente* di un hotel

I suggerimenti

Nella pagina di ogni hotel è presente anche una sezione *I clienti che hanno visualizzato “Nome dell’hotel” hanno consultato anche:*, nella quale si possono trovare le descrizioni dei tre hotel più visti dagli stessi utenti che hanno guardato anche la pagina dell’hotel corrente.

Quella appena descritta è l'unica forma di suggerimento basato su preferenze (implicite) presente nel sito Booking.com. Poiché questo portale è utilizzato principalmente da turisti informati, che organizzano i loro viaggi per conto proprio e hanno esigenze ben definite, sarebbe utile rendere il servizio offerto da Booking.com maggiormente personalizzato e interattivo.

6.2 Travel SeReS CI

In questo sottocapitolo si presenta la versione di SeReS CI integrata all'interno di Booking.com, che possiamo denominare, per ovvie ragioni, Travel SeReS CI. Si fanno alcune assunzioni preliminari riguardo il bacino di utenza, la taglia dell'itemset, la veridicità delle informazioni e la lingua usata. Dopo di che si passa alla descrizione vera e propria.

6.2.1 Assunzioni

Non avendo una effettiva implementazione del sistema non è possibile studiare un reale caso d'uso, ciò comporta la necessità di specificare alcune assunzioni, in particolare, riguardo il numero di utenti che utilizzano il sistema, la taglia del dataset, il livello di completamento dei profili utente e la veridicità delle informazioni.

Bacino d'utenza

Si suppone che gli utenti del sistema esistente (circa 2 milioni¹) siano anche i fruitori del servizio che ci apprestiamo a descrivere. In particolare si assume che di questi sia classificabile² come *Cliente* il 20% (400 000), il 30% (600 000) come *Utente premium*, lo 0.3% (6000) come *Collaboratori esterni*, l'1% (20 000) come *Service Provider* (anche se in questo caso sarebbe più appropriato il termine *Service Producer*) e il restante 48.7 % come semplice *Visitatore*. A scopo di rendere più semplice la trattazione si ipotizza che il grado di completamento del profilo dei *Clienti* sia *Alto* e che gli *Utenti premium* siano tutti di livello *Plus*.

Taglia e caratterizzazione dell'itemset

Gli item considerati, in questo caso, sono i servizi di alloggio temporaneo, quindi la dimensione dell'itemset corrisponde al numero di Hotel suggeribili. Ancora, si suppone che gli hotel prenotabili tramite il sistema esistente (circa 120 000) siano anche presenti nell'itemset del RS.

Come detto nel primo sottocapitolo, la classificazione delle strutture può essere fatta per numero di stelle, per tipo di struttura e per catena di appartenenza. Si propone, dunque, una possibile ripartizione degli hotel rispetto a questi tre parametri³.

Per quanto riguarda il numero di stelle si assume che il 3% (3600) sia a 1 stella, il 13% (15 600) a 2 stelle, il 28% (33 600) a tre stelle, un altro 28% (33 600) a quattro stelle, il 4% (4800) a 5 stelle e il 24% (28 800) non classificato⁴.

¹ Il dato è riferito all'anno 2011

² Rispetto alle tipologie di utenti specificate nella sezione 5.3.1

³ Tutte le scelte sono state effettuate basandosi sull'analisi statistica degli alberghi di 25 città, considerando sia metropoli (come Londra, New York e Parigi), sia città di medie e piccole dimensioni (come Gent, Genova e Padova).

⁴ Con il termine non classificato ci si riferisce a una struttura per la quale non è prevista l'assegnazione di stelle oppure per la quale non è stata fornita l'informazione.

I tipi di strutture sono quattro, cioè *Appartamento*, *Albergo*, *Ostello*, *Altre strutture*, e la suddivisione dell'itemset proposta è, rispettivamente, 14% (16 800), 68% (81 600), 15% (18 000), 3% (3600).

Gli hotel possono essere di proprietà dei singoli albergatori o appartenenti a catene di alberghi. Si suppone che appartengano al primo gruppo l'80% delle strutture (96 000) e al secondo gruppo il restante 20% (24 000).

Veridicità delle informazioni

L'obiettivo di questo capitolo è proporre una possibile integrazione di SeReS CI con il sito Booking.com e come descritto in fase di presentazione del RS (vedi capitolo 5), nel sistema è previsto un controllo da parte dell'*authority* sulla veridicità delle informazioni fornite dagli utenti. Per semplificare la proposta di applicazione si suppone a priori che le informazioni contenute nei dataset siano veritiere, senza dover quindi descrivere il passaggio di verifica in presenza di anomalie. Per tali ragioni l'ipotesi formulata non è da considerarsi rilevante ai fini della valutazione qualitativa.

Sistema monolingua

Per non introdurre complicazioni superflue per questo tipo di trattazione, si assume che la lingua utilizzata da utenti e Service Provider sia l'italiano. Certamente nella realtà non è così, ma si potrebbe facilmente ovviare a questo problema introducendo un vocabolario multilingua che permetta la traduzione delle informazioni contenute nel sito nella lingua dell'utente. Una funzionalità di questo tipo è già presente nel sistema esistente per la pagina *Descrizione e disponibilità* e per la sezione rating della pagina *Giudizi dell'utente*. Un po' più complesso è il compito di tradurre le recensioni.

6.2.2 Definizione del sistema

Ricorrendo alle strutture vettoriali proposte nel capitolo 5, possiamo definire utenti, servizi e Service Provider. Prima di far questo è però necessario descrivere i vocabolari relativi al dataset Servizi. Si noti che non è necessario specificare quelli del dataset Utenti e del dataset Service Provider perché non hanno peculiarità legate al contesto che valga la pena discutere esplicitamente.

Vocabolari del dataset Servizi

I vocabolari relativi al dataset Servizi, come specificato nel capitolo precedente, sono cinque:

- *Vocabolario Proprietà Servizi*. Ho individuato otto proprietà di un hotel che possono influenzare la scelta di un utente: {*ampiezza stanze*; *arredamento stanze*; *comfort*; *isolamento acustico*; *pulizia stanze*; *qualità della zona*; *rapporto qualità prezzo*; *servizi accessori*}.
- *Vocabolario Service Provider*. È un elenco formato dai nomi di tutti i 120 000 hotel, anche se in realtà con quei nomi si intende identificare il personale che vi lavora all'interno.
- *Vocabolario Categoria Servizi*. In questo caso è un vocabolario superfluo perché c'è soltanto una categoria, cioè il servizio di alloggio temporaneo.
- *Vocabolario Descrizione Semantica*. Contiene l'elenco di tutti i tag specificati dagli utenti, e gli eventuali sinonimi.
- *Vocabolario Stato Servizio*. Anche questo vocabolario non è indispensabile, perché è sempre lo stesso in tutti i contesti. I termini che contiene sono: {attivo, inattivo}.

Vettore utente

Il vettore utente definito nel capitolo 5 non deve essere adattato in base al tipo di servizio, quindi rimane valida la struttura presentata. L'unico elemento che potrebbe essere utile modificare è il campo *Peso_Proprietà*, perché legato alle proprietà specifiche del servizio. Si può dunque pensare di ridurre l'ampiezza di questo campo, tenendo soltanto i pesi relativi alle proprietà specificate nel vocabolario *Proprietà Servizi*.

Vettore servizio

Anche per il vettore servizio non ci sono molte modifiche da apportare. Una è eliminare il campo *categoria_servizio*, essendoci soltanto un'unica categoria; l'altra è specificare nel campo *proprietà_servizio* i termini contenuti nel vocabolario *Proprietà Servizi*.

Vettore Service Provider

Come per gli altri due vettori, il vettore Service Provider rimane sostanzialmente immutato, se non per il campo *Tipologia_Provider*, la quale può essere eliminata per il medesimo motivo del campo *categoria_servizio* del vettore servizio.

6.2.3 Registrazione utente

Un utente non è costretto a possedere un account per utilizzare il RS, tuttavia per ricevere suggerimenti maggiormente personalizzati è consigliabile registrarsi al sito.

La registrazione di un nuovo utente consiste nell'inserimento di un nome utente, di un indirizzo email e di una password. Il nuovo processo di registrazione, quindi, non richiede maggiori informazioni, anzi, elimina la richiesta del cognome e sostituisce il nome della persona con un nickname. Tuttavia, rispetto a quello precedente, il profilo utente è in grado di memorizzare più dati, i quali possono essere inseriti subito dopo il primo login, in un secondo momento oppure ricavate implicitamente dal sistema (vedi i campi delle categorie *RS_INFO* e *Social* del vettore utente nella sezione 5.4.5).

Come spiegato nel capitolo 5, le modalità di accesso possibili sono due. L'alternativa alla registrazione standard appena descritta, è l'utilizzo del proprio profilo Facebook. Optando per questa seconda modalità si accetta di condividere con il RS le informazioni in esso contenute.

6.2.4 Personalizzazione del profilo

Subito dopo il primo login, qualsiasi sia l'alternativa di accesso scelta, viene proposto un test di valutazione di sei hotel con lo scopo di apprendere informazioni rilevanti riguardo le preferenze dell'utente ed essere in grado fin da subito di produrre suggerimenti personalizzati, risolvendo così il problema del cosiddetto cold start. Per ogni hotel si propone una descrizione riassuntiva delle caratteristiche, cinque rating e cinque recensioni degli utenti, e la reputazione del Service Provider.

Gli hotel sottoposti all'attenzione dell'utente sono scelti diversi tra loro e presentano valutazioni con recensioni singolari (es. viene espressa una lamentela sulla mancanza di un deposito bagagli, oppure viene lodata la qualità della colazione). Ciò è necessario per catturare le preferenze implicite degli utenti, visto che spesso pensano di sapere ciò che vogliono, ma poi manifestano esigenze diverse da quelle inizialmente espresse.

Per quanto riguarda le preferenze esplicite, si permette all'utente di indicare con un numero da 1 a 5 (mezzi voti compresi) il livello di importanza delle caratteristiche di un Hotel, contenute nel vocabolario *Proprietà Servizi* (vedi sopra).

6.2.5 Il sito web rinnovato

Poiché i suggerimenti non possono essere proposti direttamente come risultato della query a causa del business model di booking.com, secondo il quale gli albergatori pagano una

commissione per avere una posizione privilegiata nei risultati delle ricerche, si può apportare una modifica non invasiva, aggiungendo un'opzione *Suggerimenti personalizzati* alle possibilità di ordinamento dei risultati.

La homepage

Non è necessario modificare la struttura della homepage, tuttavia si possono sostituire i riquadri delle destinazioni più ambite e degli ultimi hotel visionati con i consigli del RS. In questo modo l'aspetto che assume è diverso per ogni utente in base alle singole preferenze individuali.

L'homepage personalizzata, e i suggerimenti in particolare, sono visibili agli utenti che hanno effettuato il login, tuttavia, sfruttando il meccanismo dei cookie, si possono rendere disponibili anche a tutti gli altri utenti.

La query di ricerca

Come detto in apertura di sezione, la strategia di ricerca non può essere modificata a causa del business model di Booking.com. Si potrebbe anche pensare ad un diverso modello, tuttavia si aprirebbero questioni di carattere economico che non sembra il caso di trattare in questo contesto. Non cambiando la strategia si è deciso di lasciare immutato anche il Search Engine e il relativo form (specifica del luogo e della data del viaggio).

La soluzione proposta per somministrare i suggerimenti agli utenti, allora, è l'aggiunta di un tasto *Suggerimenti Personalizzati* tra le opzioni di ordinamento dei risultati della query di ricerca. In automatico, così, il sistema mette ai primi posti gli hotel che più incontrano le preferenze (esplicite e implicite) dell'utente. Rimangono comunque disponibili gli strumenti per raffinare la ricerca attraverso la selezione diretta di parametri specifici.

La descrizione degli hotel

Lo sdoppiamento delle pagine dedicate agli hotel in *Descrizione e disponibilità* e *Giudizi del cliente* sembra un buon approccio, quindi si ritiene opportuno non modificarlo. Si può invece cambiare l'origine delle informazioni contenute nella sezione descrittiva, permettendo anche agli utenti (oltre che ai Service Producer e Service Provider) la modifica e l'aggiunta di dati, in base alle conoscenze in loro possesso. L'approccio proposto è quello descritto nella sezione 5.4.7 e, per intenderci, ricalca a tutti gli effetti il meccanismo di aggiunta di risorse alle pagine di Wikipedia.

I feedback degli utenti

Gli utenti sono stimolati a produrre feedback di qualità nella prospettiva di uno sconto sulle successive prenotazioni. Già adesso Booking.com prevede condizioni tariffarie agevolate distribuite a pioggia (non esiste un criterio meritocratico, vige la regola del "chi prima arriva meno paga"), tuttavia se diminuisse (o eliminasse) gli sconti per gli utenti che non producono feedback (la categoria *Clienti*), e aumentasse invece quelli per chi fornisce valutazioni significative sulla sua esperienza personale, si avrebbe un portale con più informazioni, organizzate meglio e di ottima qualità.

I tipi di feedback che l'utente può fornire al RS sono sei: *descrizioni*, *recensioni*, *rating*, *tag*, *valutazione del personale*, *valutazione dei suggerimenti*, *contenuti multimediali* (immagini e video). Il processo di valutazione è un po' più lungo rispetto al precedente, ma questa perdita di tempo è certamente compensata dagli sconti offerti (per di più incrementali, al crescere del numero di feedback corrisposti).

I suggerimenti

Il RS seleziona gli hotel, le cui caratteristiche maggiormente concordano con le preferenze dell'utente. L'approccio gerarchico definita nella sezione 5.4.10 è riproposto anche all'interno

del sito. Nella homepage si mira ad attirare l'interesse dell'utente, proponendo quindi in primo piano i suggerimenti *Hotel Simili Ai Miei preferiti* (content based) e *Hotel Consigliati Per Me* (superuser based), e più defilati le categorie *Hotel Del Giorno* e *Hotel Del Mese*. Nella pagina successiva, quella dei risultati della query, l'obiettivo cambia, e si mira a dirottare l'attenzione dell'utente sulle scelte che più dovrebbero convincerlo a prenotare. Non c'è una sezione specificatamente dedicata ai suggerimenti, infatti, questa funzione viene delegata all'opzione di ordinamento *Suggerimenti Personalizzati*, descritta in precedenza. Infine, nella pagina dedicata ai singoli hotel, i consigli sono presenti soltanto sul fondo della pagina. Lo scopo del RS è di garantire una visione completa del ventaglio di possibilità, poiché ormai l'utente ha un'idea precisa di ciò che vuole, ma rischia di non esplorare tutte le alternative. In un'ottica simile sono stati inseriti a questo livello le categorie di suggerimenti *Hotel simili* (content based) e *Altri Hotel Che Potrebbero Piacermi* (user based e group based).

6.2.6 I canali aggiuntivi

Come illustrato nell'architettura di SeReS CI, oltre al sito web sono previste altre tre interfacce: il widget per PC, l'applicazione per Facebook e la app per dispositivi mobili.

Il widget per PC

Il widget è uno strumento di supporto al web site, che permette di utilizzare il sistema offline, con capacità ridotte, oppure online, per accedere alle funzionalità principali. Inoltre, consente di interagire con il sistema (online), scaricando dati e suggerimenti o, viceversa, caricando dati di vario genere. Non essendo stato pensato per un'esplorazione approfondita, propone soltanto i suggerimenti più efficaci, in base alla storia dell'utente (vedi sezione 5.4.10).

L'applicazione per Facebook

Già adesso esiste un'applicazione di Booking.com per Facebook, tuttavia essa è semplicemente un link al sito principale, che quindi non sfrutta le potenzialità del Social Network. La nuova versione che integra il RS permette agli utenti di indicare direttamente ai loro amici gli hotel in cui sono stati, fornendo preziosi consigli (trust based), che provenendo da persone conosciute, sono considerati con maggiore attenzione e spesso apprezzati. Oltre a ciò l'utente può ricevere direttamente nella sua pagina i suggerimenti generati per lui dal RS (popularity based). Questa possibilità, tuttavia, non deve essere abusata e prima di attivarla è necessario chiedere agli utenti la frequenza con cui vogliono ricevere le proposte (da due a trenta al mese).

La app per dispositivi mobili

In questo dominio è di particolare interesse l'utilizzo della App per i dispositivi mobili, infatti le opportunità di creare un valore aggiunto sono molto allettanti. Oltre alla possibilità per il RS di memorizzare molte informazioni dell'utente e sull'utente (vedi sezione 5.3.5), migliorando così la formulazione delle proposte, vi è anche, per chi utilizza la App, la convenienza di avere sempre con se preziosi suggerimenti sugli Hotel, nonché informazioni utili sui servizi disponibili nella zona.

La validità di una simile applicazione può essere facilmente compresa dalle statistiche presentate nelle due figure seguenti [87]. Nella prima (fig. 6.6) si evidenzia come due utenti su cinque trascorrono almeno un'ora a settimana su siti di viaggi attraverso un dispositivo mobile. Nella seconda (fig. 6.7), invece, si può vedere che circa un utente su due tra i clienti *mobile* utilizza il proprio dispositivo per cercare destinazioni di viaggi e leggere recensioni, mentre uno su quattro addirittura per prenotare le vacanze. Considerando che quelli elencati sono proprio tutti i servizi messi a disposizione dal portale Booking.com, sono immediate le conclusioni a cui si può giungere.

Percentuale di utenti che hanno trascorso almeno 1 ora a settimana su siti mobile nelle seguenti categorie

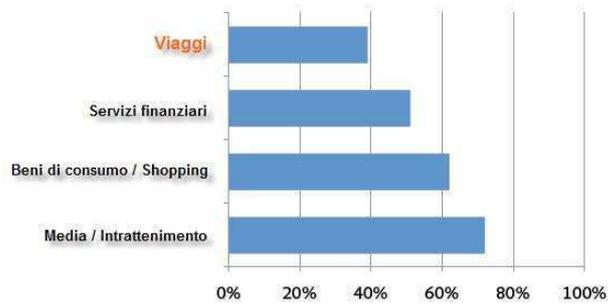


Figura 6.6: statistiche preferenze utenti mobile [87]

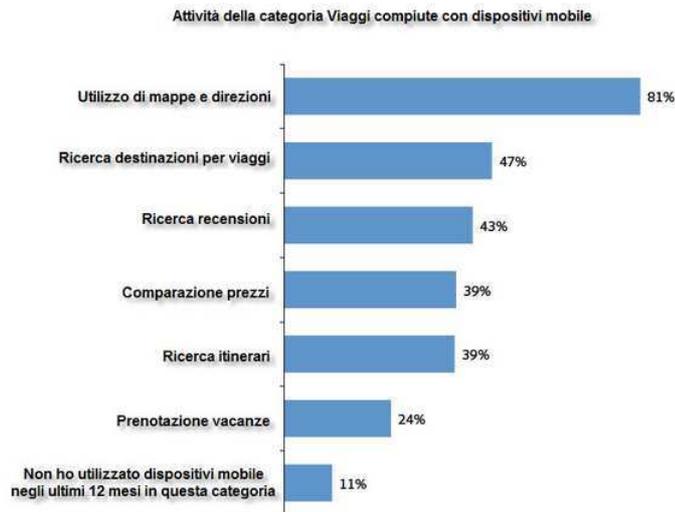


Figura 6.7: attività della categoria viaggi compiute con dispositivi mobili

6.3 Analisi comparativa

Mancando un'effettiva implementazione non è possibile condurre test per valutare statisticamente la qualità dei suggerimenti, e di conseguenza non si può nemmeno confrontare il sistema con quelli già esistenti della medesima categoria. Non è banale nemmeno poter affermare con certezza se l'integrazione del RS nel sito possa portare miglioramenti nel sistema e accrescere la soddisfazione degli utenti che lo utilizzano. Tuttavia, si può procedere a un'analisi comparativa che si pone come obiettivo il confronto oggettivo tra le funzionalità del sistema esistente e le nuove feature.

6.3.1 Il confronto

Il confronto tra il sistema esistente e quello proposto può essere fatto su due livelli: il processo di prenotazione e la qualità dei contenuti.

Processo di prenotazione

Il processo di prenotazione comprende tutti i passaggi che vanno da quando l'utente arriva nel sito, fino alla effettiva prenotazione. Le fasi che si possono distinguere sono sei: valutazione degli elementi della homepage, specificazione dei parametri di ricerca, ordinamento della lista dei risultati, scrematura secondo vincoli aggiuntivi, valutazione dei singoli hotel e prenotazione. Come si può dedurre dalla seguente tabella riassuntiva, le fasi 2 e 4 rimangono le stesse, mentre tutte le altre hanno subito variazioni, più o meno significative.

FASI	PRIMA	DOPO
1. Valutazione degli elementi della homepage	Tutti gli utenti avevano a disposizione gli stessi hotel in vetrina	Ogni utente ha una homepage personalizzata contenente gli hotel più adatti a lui
2. Specificazione dei parametri di ricerca	L'utente specifica il luogo e la data del viaggio	L'utente specifica il luogo e la data del viaggio
3. Ordinamento della lista dei risultati	L'utente può ordinare la lista secondo diversi criteri ma non più di uno per volta	L'utente può ordinare la lista in base alle sue preferenze venendo guidato dal RS
4. Scrematura secondo vincoli aggiuntivi	L'utente può specificare alcuni vincoli per restringere il ventaglio delle possibilità	L'utente può specificare alcuni vincoli per restringere il ventaglio delle possibilità
5. Valutazione dei singoli hotel	L'utente valuta i singoli hotel e in fondo alla pagina trova suggerimenti per strutture simili	L'utente valuta i singoli hotel e in fondo alla pagina trova suggerimenti per farsi un'idea più ampia
6. Prenotazione	L'utente può prenotare l'hotel scelto a condizioni tariffarie vantaggiose	L'utente può prenotare l'hotel scelto a condizioni economiche particolari, a seconda dei feedback prodotti

Tabella 6.1: confronto sul processo di prenotazione

Qualità dei contenuti

I contenuti presenti nel sito sono: descrizioni, immagini, video, recensioni, rating, tag, valutazioni del personale e suggerimenti. La principale differenza che si evince dalla tabella seguente è che nella versione originaria la maggior parte dei contenuti era prodotta dai Service Provider o dai Service Provider, mentre nella versione con il RS hanno maggiore importanza gli utenti. Questo è sicuramente un valore aggiunto perché in tal modo gli utenti si sentono più partecipi e quindi sono portati a legarsi maggiormente al sito e al servizio che offre.

Tipo	PRIMA	DOPO
1. Descrizioni	Le descrizioni dell'hotel sono di competenza dei proprietari dell'Hotel	Le descrizioni sono definite dai clienti e possono essere controllate dai proprietari dell'Hotel
2. Immagini	Le immagini possono essere inserite soltanto dai proprietari dell'hotel	Le immagini possono essere inserite dai clienti e possono essere controllate dai proprietari dell'Hotel
3. Video	I video possono essere inseriti dai proprietari dell'hotel	I video possono essere inseriti dai clienti e possono essere controllate dai proprietari dell'Hotel
4. Recensioni	I clienti possono descrivere la loro esperienza attraverso una recensione	I clienti se vogliono degli sconti particolari devono descrivere la loro esperienza attraverso una recensione
5. Rating	I clienti possono valutare alcune proprietà dell'hotel attraverso un rating	I clienti se vogliono degli sconti particolari devono valutare le proprietà dell'hotel attraverso un rating
6. Tag	NON PRESENTI	L'utilizzo dei tag permette un riconoscimento semantico degli Hotel, migliorando anche la ricerca
7. Valutazioni del personale	La valutazione del personale è limitata al rating della proprietà Staff	La valutazione del personale prevede un rating delle qualità previste e recensioni esplicative
8. Suggerimenti	I suggerimenti sono basati sulle commissioni pagate dai proprietari degli Hotel, quindi, uguali per tutti	I suggerimenti sono personalizzati e differenziati per scopo, a seconda della pagina in cui si trovano

Tabella 6.2: confronto sui contenuti

6.3.2 Vantaggi e limitazioni

L'unico scopo della proposta di applicazione del Service Recommender System basato su intelligenza collettiva al sito Booking.com, è di fornire un esempio pratico che possa far meglio comprendere il sistema nel suo complesso. Nella seguente tabella si elencano, in modo riassuntivo, i principali vantaggi e limitazioni dell'integrazione del RS nel portale. Mancando un'effettiva implementazione non è possibile un riscontro concreto, quindi la valutazione riguarda i possibili effetti prodotti su Utenti e Service Provider, e non tanto sulla qualità del sistema in se.

VANTAGGI	LIMITAZIONI
Gli utenti vengono maggiormente coinvolti e si sentono protagonisti	I contenuti prodotti da professionisti sono di maggiore qualità rispetto agli UGC
Sconti incrementali	Minori sconti per gli utenti che non vogliono produrre feedback potrebbero scoraggiare alcuni utenti
Fidelizzazione degli utenti	Dando più spazio ai suggerimenti personalizzati potrebbero diminuire gli investimenti pubblicitari
Maggiori informazioni sugli Hotel	L'operazione di inserimento dei feedback necessita di maggior tempo
Ricerche personalizzati, più precise e a più ampio spettro	
Possibilità di avere maggiori informazioni sul personale	

Tabella 6.3 : vantaggi e limitazioni della proposta

Conclusioni

Il lavoro compiuto nel redigere questa tesi può essere suddiviso in due fasi: la prima di ricerca, la seconda di progettazione. Nella prima parte, attraverso la lettura e lo studio di numerosi documenti, ho potuto apprendere lo stato dell'arte degli argomenti trattati. In questa fase ho anche eseguito delle analisi comparative e ho formulato alcune teorie personali. Tra queste, l'elemento che ritengo di maggior valore, è la classificazione dell'intelligenza collettiva secondo tre livelli di complessità. Nella seconda parte, invece, ho potuto mettere in pratica le conoscenze acquisite, valorizzandole con una proposta personale.

Negli ultimi due capitoli della tesi si presenta, prima con una descrizione ad alto livello, poi con l'esempio di una possibile applicazione, un Service Recommender System basato su intelligenza collettiva. La proposta avanzata prevede l'impiego di una Collective Intelligence Platform come cervello del sistema. Le funzionalità di questa piattaforma, descritte accuratamente nel capitolo 5, costituiscono l'elemento innovativo dell'approccio esposto.

Il mio lavoro di tesi, ponendosi come obiettivo non tanto l'implementazione di un Recommender System, bensì la ricerca di un approccio nuovo e originale al cosiddetto *recommendation problem*, si è fermato a una progettazione ad alto livello, ponendosi come base teorica a eventuali sviluppi più concreti. Ciò che, invece, probabilmente manca è un'analisi sui punti di forza e di debolezza del sistema proposto e un confronto qualitativo con i sistemi esistenti della medesima categoria.

Le possibili estensioni al Recommender System progettato sono state elencate nell'ultima sezione del capitolo 5. Questi elementi potrebbero costituire una valida integrazione al lavoro svolto, ma rappresentano solo un piccolo sottoinsieme degli sviluppi futuri ipotizzabili. Da questa tesi, infatti, si possono trarre spunti in più direzioni. Per iniziare, si può considerare la questione implementativa del sistema, realizzando un prototipo sulla base dell'architettura specificata. Secondariamente si può studiare la progettazione delle quattro interfacce (sito web, widget, facebook, mobile app) messe a disposizione dell'utente, essenziali per raccogliere i *geni* (secondo l'accezione del MIT) dell'intelligenza collettiva. Ulteriori lavori futuri possono riguardare sia argomenti di ricerca (es. indagare se esiste una intelligenza collettiva di livello 3), sia proposte implementative (es. realizzare un collective intelligence agent in grado di recuperare e inserire in una struttura dati predefinita le informazioni presenti nei profili Facebook).

Appendice 1

Definizione di Servizio NEXOF-RA

Nell'ambito del progetto Nessi Open Framework Reference Architecture (NEXOF-RA)¹, Antonio De Nigro e Piero Corte della società Engineering², hanno prodotto un report [8] nel quale espongono le proprietà che, secondo un loro studio, sono indispensabili per la definizione del concetto di servizio.

Il documento presenta quattro concetti chiave, che identificano situazioni di natura via via più complessa. Ogni concetto è stato analizzato, scomposto nelle sue componenti principali e infine rappresentato attraverso un esempio concreto. Al termine del lavoro è stato scelto uno di questi quattro concetti come definizione formale di servizio, identificando, così, le proprietà minime che un servizio deve possedere.

Vediamo ora i quattro concetti e i relativi esempi:

C1: Una azione eseguita da una entità (il fornitore) a beneficio di un'altra (il cliente)

Il concetto C1 racchiude i seguenti fatti:

- una entità fornitore (F) esegue un'azione (A);
- una entità cliente (C) ha un beneficio (B) dall'azione eseguita.

Quindi possiamo dire che il concetto C1 dipende da F, da C e da A. A sua volta A dipende dalla percezione che ha C di B.

Possiamo esprimere tali dipendenze tramite una definizione testuale, una rappresentazione grafica e uno schema UML³:

- Testuale

$C1 = \text{def } [F, A(B), C]$

Utilizzando questa notazione possiamo anche esprimere il concetto di identità tra due istanze x e y di C1:

siano $x = [f1, a1(b1), c1]$ e $y = [f2, a2(b2), c2]$

¹ Nessi Open Framework Reference Architecture (NEXOF-RA) è un progetto che rappresenta il primo passo nel processo di costruzione del Nessi Open Framework, la piattaforma aperta per la creazione e la diffusione delle applicazioni, che consentirà la creazione di un ecosistema basato sui servizi, dove fornitori di servizi e terze parti possono collaborare facilmente. <http://www.nexof-ra.eu/?q=node/2>

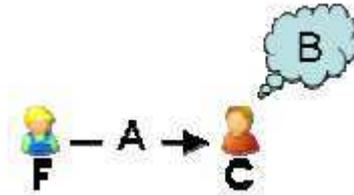
² <http://www.eng.it/>

³ L'Unified Modeling Language (UML) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. Questo linguaggio è nato con l'intento di unificare approcci precedenti, raccogliendo le *best practices* nel settore e definendo così uno standard industriale unificato. UML svolge un'importantissima funzione di lingua franca nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa UML per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. <http://www.uml.org/>

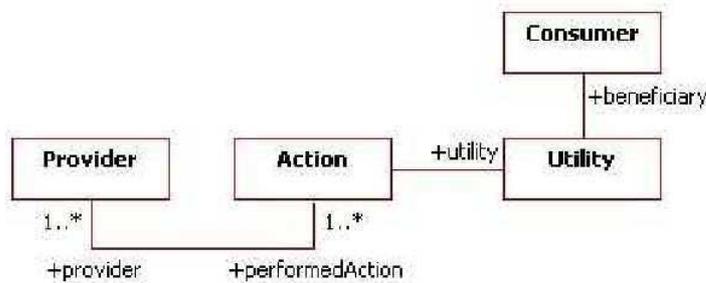
allora $x == y \leftrightarrow f1 == f2$, $a1(b1) == a2(b2)$ e $c1 == c2$

(N.B. Non deve valere necessariamente $b1 == b2$).

- Grafica



- UML



Vediamo ora un esempio che può essere facilmente riconducibile al concetto C1 e ai due sottoconcetti contenuti in esso:

Esempio 1

Una persona (il cuoco) prepara e serve un pasto ad un'altra persona (l'uomo affamato)

F: il cuoco è l'entità fornitore;

C: l'uomo affamato è il cliente;

B: il pasto preparato e servito che soddisfa l'esigenza di C è il beneficio percepito da quest'ultimo;

A(B) : preparare e servire il pasto, se questo soddisfa l'esigenza di C, è l'azione.

Vediamo ora il secondo concetto.

C2: Un'azione eseguita da un'entità (il fornitore) che corrisponde con la richiesta di un'altra entità (l'entità richiedente), secondo l'interpretazione di quest'ultima.

Questo secondo concetto racchiude i due seguenti fatti:

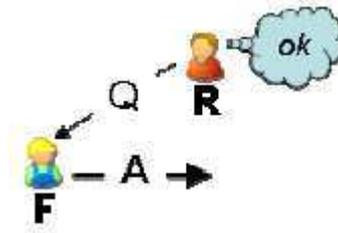
- una entità richiedente (R) fa una richiesta (Q) a un'entità fornitore (F) per eseguire una certa azione (A);
- un'entità fornitore (F) esegue un'azione richiesta (A).

Possiamo dunque dire che C2 dipende dall'entità fornitore, dall'entità richiedente e da un'azione che a sua volta dipende da una richiesta esplicita. Il tutto, come prima, può essere scritto nelle tre forme:

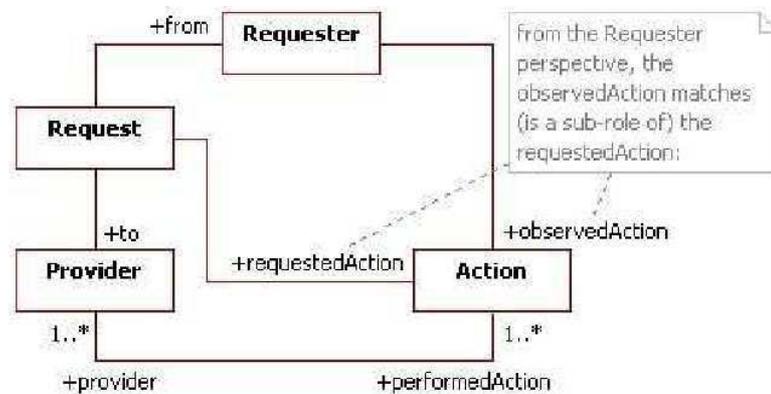
- Testuale

$C2 = \text{def}[F, A(Q), R]$

- Grafica



- UML



Come per il concetto precedente, possiamo utilizzare un esempio concreto per renderlo più comprensibile.

Esempio 2

Una persona (l'uomo affamato) chiede ad un'altra persona (il cuoco) di preparargli e servirgli un pasto, dopo di che quest'ultimo (il cuoco) fa ciò che gli è stato richiesto.

F: il cuoco è l'entità fornitore;

R: l'uomo affamato è l'entità richiedente;

Q: chiedere di preparare e servire il pasto è la richiesta;

A(Q) preparare e servire il pasto, se è stata effettuata esattamente tale richiesta, è l'azione condizionata.

Osservazione

È importante riconoscere due elementi che sono stati introdotti nell'esempio 2:

1. il cuoco interpreta la richiesta ed esegue l'azione di conseguenza;
2. l'uomo affamato osserva l'azione eseguita e la interpreta per verificare se corrisponde alla sua richiesta.

Ci sono due interpretazioni eseguite da entità distinte, e non necessariamente sono le stesse. Il risultato potrebbe essere, ad esempio, che l'entità fornitore è sicura di aver eseguito ciò che gli

era stato richiesto, ma che l'altro non è d'accordo. C2 si basa sull'interpretazione dell'entità richiedente.

Al fine di rimuovere tale asimmetria intrinseca a C2 e far sì che l'interpretazione sia univoca dobbiamo introdurre un nuovo concetto.

C3: Un'azione eseguita da una identità (il fornitore) che corrisponde alla richiesta di un'altra entità (entità richiedente), secondo un precedente accordo stabilito tra le due parti.

Il concetto C3 racchiude i seguenti fatti:

- una entità richiedente (R) fa una richiesta (Q) a un'entità fornitore (F) di eseguire una certa azione (A);
- un'entità richiedente e un'entità fornitore stabiliscono un accordo (G) su cosa deve essere fatto;
- un'entità fornitore (F) esegue l'azione richiesta (A) secondo l'accordo (G).

Quindi, possiamo dire che C3 dipende da un'entità fornitore, da un'entità richiedente e da un'azione, che dipende dall'accordo stabilito, che a sua volta dipende da una richiesta esplicita. Possiamo esprimere il tutto succintamente usando le solite tre notazioni:

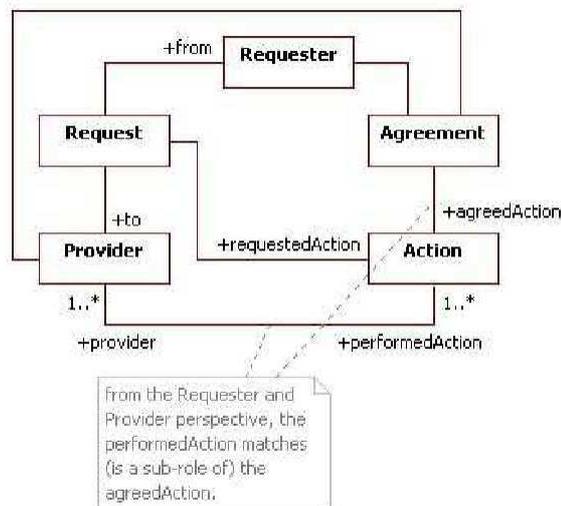
- Testuale

C3 = def[F, A(G(Q)), R]

- Grafica



- UML



Vediamo ora un esempio che rappresenti tale concetto:

Esempio 3

Una persona (l'uomo affamato) chiede ad un'altra persona (il cuoco) di preparargli e servirgli un pasto, si accordano sulla preparazione e sul servizio di pesce fresco, dopo di che il secondo dei due (il cuoco) fa ciò che gli era stato richiesto.

F: il cuoco è l'entità fornitore

R: l'uomo affamato è l'entità richiedente

Q: chiedere al cuoco di preparare e servire un pasto è la richiesta

G(Q): accordarsi sulla preparazione e il servizio di pesce fresco se viene richiesto di preparare e servire un pasto è l'accordo risultante dalla richiesta iniziale

A(G(Q)): preparare e servire pesce fresco se ci si è accordati di preparare e servire pesce fresco è l'azione che dipende dall'accordo.

Osservazione

Finora è stato dato per certo che l'entità fornitore sia sempre pronta a ricevere una richiesta dall'entità richiedente, ma non è detto che sia sempre così. Sarebbe opportuno considerare un processo specifico che l'entità richiedente possa riconoscere come il contesto appropriato, nel quale sottoporre le sue richieste (prima di accordo e poi di servizio). Dobbiamo quindi introdurre un quarto concetto che tenga conto di tale specifica.

C4: Un'azione eseguita da un'entità (il fornitore), pronta a ricevere richieste e stabilire accordi, che corrispondano alle richieste di un'altra entità (il richiedente) secondo accordi stabiliti precedentemente dalle due parti.

Il concetto C4 implica 4 sottoconcetti:

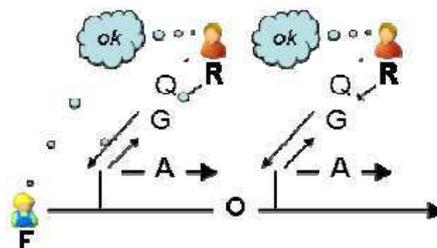
- Un'entità fornitore (F) esegue un processo (O) per ricevere richieste (Q) da entità richiedenti (R) e per stabilire accordi (G) con essi;
- Un'entità richiedente (R) fa una richiesta (Q) tramite un processo (O) a un'entità fornitore (F) di eseguire una certa azione (A);
- Un'entità richiedente e un'entità fornitore stabiliscono un accordo (G) su ciò che deve essere fatto;
- Un'entità fornitore (F) che esegue l'azione richiesta (A) in base all'accordo stabilito.

Quindi, possiamo dire che C4 dipende da un'entità fornitore, da un'entità richiedente e da un'azione, che dipende da un accordo, che dipende da una richiesta esplicita, che a sua volta dipende dall'esistenza di uno specifico processo in corso. Con le tre notazioni:

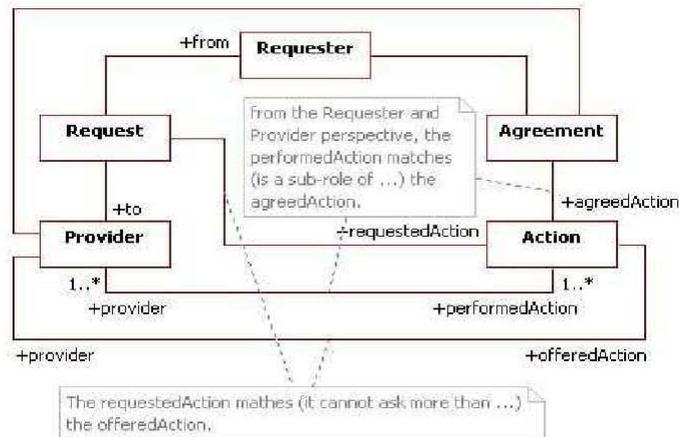
- Testuale

$C4 = \text{def } [F, A(G(Q(O))), R]$

- Grafica



- UML



Facciamo ora un esempio che rappresenti il concetto descritto.

Esempio 4

Una persona (un cuoco) ogni mattina apre il suo ristorante per essere pronto ad accogliere le persone che vogliono avere un pasto. Ogni giorno persone diverse (i clienti) entrano nel suo ristorante. Ogni cliente gli chiede di preparare e servire un pasto e si accorda con il cuoco sulla preparazione e il servizio di un pasto specifico. Infine il cuoco fa ciò che gli è stato chiesto.

F: il cuoco è l'entità fornitore

O: l'apertura del ristorante ogni giorno da parte del cuoco, le ordinazioni dei clienti e l'accordo che fanno clienti e cuoco costituiscono il processo che consente di ricevere le richieste e raggiungere l'accordo;

R: il cliente è l'entità richiedente;

Q(O): il cliente che chiede al cuoco di preparargli e servirgli un pasto dopo che è entrato nel ristorante è la richiesta fatta attraverso il processo;

G(Q(O)): un cliente e il cuoco che si accordano sulla preparazione e il servizio di un piatto specifico se viene richiesto di preparare e servire un pasto è l'accordo risultante dalla richiesta iniziale;

A(G(Q(O))): preparare e servire il piatto specifico se c'è stato un accordo sulla preparazione e il servizio di un piatto specifico è l'azione che dipende dall'accordo.

Tra i quattro concetti esposti, quello scelto da NEXOF-RA per essere adottato come definizione di servizio è C2. Quindi un servizio è definito come:

Un'azione eseguita da una entità (il fornitore) che corrisponde alla richiesta di un'altra entità (richiedente), secondo un accordo precedentemente stabilito tra le due parti.

Tratto da:

A. De Nigro e P. Corte (Engineering); marzo 2009; NESSI Open Framework – Reference Architecture - Service Description Investigation Team (Final results)

<http://www.nexof-ra.eu/sites/default/files/D2.1%20Appendix%20B%20-%20Service%20Description%20IT%20Report.pdf>

Appendice 2

Linguaggi e Protocolli per l'IoS

A2.1 XML

XML (eXtensible Markup Language) è attualmente considerato come il linguaggio standard per lo scambio dei dati. È un linguaggio a tag che permette di identificare il tipo di dato trasmesso associandone un significato.

Banalmente si può paragonare XML all'HTML (derivano entrambi dallo standard SGML¹ che fornisce una notazione formale per la definizione dei linguaggi di markup) con la differenza che, mentre in HTML i tag servono per esprimere come un dato deve essere visualizzato (font, dimensione, colore, ecc.), in XML i tag rappresentano il significato del dato, quindi non è affatto necessario che esistano dei parser che rendano il contenuto dei file XML human-readable, come invece accade con HTML.

Per rendere più chiara la descrizione verrà ora mostrato un esempio di file XML che descrive un ipotetico magazzino contenente due libri, il primo, la "Divina Commedia" di Dante Alighieri, che ha codice 111 e costa 40 euro, il secondo, i "Promessi Sposi" di Alessandro Manzoni, che ha codice 222 e costa 30 euro.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<magazzino>
  <libro>
    <IDLibro>111</IDLibro>
    <Titolo>Divina Commedia</Titolo>
    <Prezzo>40</Prezzo>
    <Autori>Dante Alighieri</Autori>
  </libro>
  <libro>
    <IDLibro>222</IDLibro>
    <Titolo>Promessi Sposi</Titolo>
    <Prezzo>30</Prezzo>
    <Autori>Alessandro Manzoni</Autori>
  </libro>
</magazzino>
```

¹ SGML è l'acronimo di Standard Generalized Markup Language ed è un metalinguaggio definito come standard ISO avente lo scopo di definire linguaggi da utilizzare per la stesura di testi destinati ad essere trasmessi ed archiviati con strumenti informatici. L'idea centrale di SGML è quella di definire linguaggi di marcatura generica chiamata "marcatura descrittiva"; ogni linguaggio obiettivo definisce le caratteristiche strutturali dei documenti che governa. L'organizzazione di un documento in un linguaggio obiettivo non si preoccupa primariamente della sua resa visiva, ma piuttosto dei ruoli logico-semantici che rivestono le parti nelle quali il documento si articola

Il documento inizia con una riga di intestazione che informa il parser sulle caratteristiche del documento, dopo la quale inizia il documento vero e proprio nel quale i vari tag, che devono seguire regole ben precise perché il documento sia accettato, “spiegano” il contenuto dei dati. Dal documento si nota inoltre come le informazioni vengano descritte creando una struttura gerarchica ad albero.

Per rendere possibile una validazione da parte del ricevente ogni documento XML può contenere la descrizione della sua struttura (DTD), cosicché il parser ricevente possa controllare se il documento è “well formed” (ben formato), cioè rispetta la sintassi XML e “valid” (valido), cioè rispetta le regole presenti nel DTD.

In sintesi XML non è nient’altro che un sistema per dare un significato al contenuto di un documento. Per questo trova grande applicazione, nell’ambito dello scambio di dati tra strutture non omogenee, come possono essere aziende diverse.

A2.2 SOAP

SOAP (Simple Object Access Protocol) è il protocollo che fornisce un meccanismo per lo scambio di informazioni strutturate (scritte sfruttando le potenzialità di XML) in un sistema distribuito decentralizzato.

Definendo solo la struttura del messaggio che le parti si scambiano, e alcune regole per elaborarlo, SOAP riesce a rimanere ad alto livello e quindi ad astrarsi, risultando completamente indipendente, dal protocollo di trasporto sottostante e risultando così implementabile da qualsiasi linguaggio di programmazione e da qualsiasi sistema operativo. Un’altra caratteristica chiave di SOAP è la sua semplicità che gli ha permesso di affermarsi tra i protocolli di comunicazione e che lo rende estensibile a seconda delle esigenze e degli sviluppi futuri. Inoltre con SOAP è possibile usare un qualsiasi modello di programmazione e non solo l’RPC al quale è spesso associato.

SOAP è diviso in tre parti:

1. L’envelope (busta), che è una struttura globale per la definizione di cosa sta nel messaggio e di chi se ne deve occupare.
2. Un insieme di regole di encoding, che sono il meccanismo con il quale poter ottenere lo scambio di informazioni.
3. Una rappresentazione RPC, che fornisce la convenzione usata per le eventuali chiamate e risposte a procedure remote.

Mostriamo ora un messaggio SOAP e analizziamone la struttura, come protocollo di trasporto per il messaggio è stato scelto l’HTTP.

Tale messaggio rappresenta la richiesta al web service dell’esecuzione del metodo `GetLastTradePrice` e passa come parametro una stringa, racchiusa tra i tag `<symbol></symbol>` che rappresenta il prodotto del quale si vuole conoscere il prezzo.

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
```

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

```

SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
<SOAP-ENV:Header>
  <t:Transaction
    xmlns:t="some-URI"
    SOAP-ENV:mustUnderstand="1">
    5
  </t:Transaction>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <m:GetLastTradePrice xmlns:m="Some-URI">
    <symbol>DEF</symbol>
  </m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Si nota subito come l'envelope del messaggio sia divisa in due parti:

1. L'Header: una parte opzionale del messaggio che consente l'aggiunta di caratteristiche al messaggio SOAP stesso ed è sfruttato spesso per l'estensibilità del protocollo, vi possono essere infatti entries riguardanti l'autenticazione o la gestione della transazione.
2. Il Body: parte obbligatoria che contiene le informazioni da spedire al destinatario, quali possono essere il marshalling di una chiamata RPC, comprensivo dell'invio di tutti i parametri necessari alla stessa.

L'applicazione che riceverà il messaggio deve:

1. Identificare le parti del messaggio SOAP destinate ad essa.
2. Verificare che tutte queste parti siano supportate e in tal caso elaborarle.
3. Nel caso in cui l'applicazione non sia il destinatario finale del messaggio, deve rimuovere tutte le parti identificate nel punto 1 e inoltrare il messaggio.

Nel caso subentrino degli errori l'applicazione risponderà al mittente con degli speciali messaggi d'errore che specificano il tipo e le cause dell'errore stesso, tali messaggi sono detti SOAP Fault.

A2.3 WSDL

Per ricevere i servizi da parte di un web service è sufficiente inviargli una opportuna richiesta SOAP. Ma come si può sapere che tipo di richiesta si aspetta un web service? WSDL (Web Services Description Language) si occupa proprio di questo, consente infatti ai web services di specificare il formato delle richieste che si attendono dai vari client. I file WSDL associati ai web services sono documenti scritti in XML che descrivono cosa essi possono fare (ad esempio i metodi da invocare con i relativi parametri e valori di ritorno), dove questi servizi risiedono, in che modo invocarli e come formatteranno le risposte alle richieste che gestiranno. Tutti i metodi riportati sono descritti in maniera astratta e sono associati a un concreto protocollo di rete e formato di messaggi.

Le descrizioni dei servizi web fatte con WSDL sono composte dai seguenti elementi:

1. **Types**: un contenitore per le definizioni dei tipi di dati realizzato usando un sistema come XSD (XML Schema Definition).
2. **Message**: una definizione astratta e tipizzata dei dati che verranno scambiati, contiene cioè i parametri di richiesta e di risposta del servizio.
3. **Operation**: una descrizione astratta di un'azione supportata del servizio web costituita da messaggi di richiesta e risposta.
4. **Port Type**: un insieme di operazioni astratte supportate da uno o più endpoint.
5. **Binding**: descrive il protocollo supportato, le operazioni consentite e i relativi input e output per un particolare port type.
6. **Port**: un singolo endpoint definito come una combinazione di un binding e di un indirizzo di rete.
7. **Service**: un insieme di endpoints collegati, mostrati dalla locazione del binding associato al servizio.

A2.4 UDDI

Definiti il linguaggio per la descrizione dei tipi di dati trasmessi (XML), il protocollo per lo scambio di informazioni strutturate (SOAP), e il linguaggio per effettuare richieste di servizi ad un web service, rimane da capire come un cliente possa individuare il servizio web adatto alle sue esigenze. UDDI (Universal Description Discovery and Integration) ha proprio questa funzionalità, permette infatti di localizzare e conoscere i servizi offerti dalle aziende iscritte, proprio come se fosse una sorta di elenco telefonico dei servizi residenti sul web. Più precisamente si tratta di una specifica per registry di informazioni sui servizi web (quale ad esempio i dettagli per collegarsi ad esso) nella quale i fornitori di servizi e i servizi stessi sono descritti in XML. UDDI si basa sul protocollo SOAP in quanto sia le richieste che le risposte sono oggetti UDDI incapsulati all'interno di messaggi SOAP.

Il tradizionale scenario di utilizzo di UDDI viene rapidamente ricondotto a queste tre fasi fondamentali:

1. **Pubblicazione (publishing)**: il fornitore del servizio per renderlo pubblico contatta il service broker che provvede ad inserirlo nel registry tramite UDDI.
2. **Ricerca (finding)**: alla richiesta di un servizio, il service broker provvede a cercare quelli che meglio rispondono alle esigenze del richiedente.
3. **Collegamento (binding)**: nel momento in cui il service broker fornisce la risposta può stabilirsi il collegamento tra il richiedente del servizio web e il fornitore.

A2.5 BPEL

BPEL (Business Process Execution Language) è un linguaggio in grado di specificare il comportamento di un processo di business, e quindi di descrivere le caratteristiche di un aggregato di web services secondo lo standard WS-BPEL². L'introduzione di tale linguaggio permette un ulteriore grado di astrazione nella pila dei servizi web, infatti grazie al suo utilizzo si è in grado di mascherare all'utente finale non solo il funzionamento interno di ogni

² <http://www.oasis-open.org/committees/download.php/14616/wsbpel-specification-draft.htm>

singolo servizio (cosa che era già possibile con WSDL) ma anche le modalità di interazione con ogni singolo servizio, mostrando all'utente finale l'aggregato di servizi come un'entità unica. BPEL esprime le specifiche degli aggregati di servizi tramite un file XML per struttura molto simile ai file WSDL.

A2.6 XSRL

XSRL (XML Service Request Language) è un linguaggio per formulare richieste ad aggregati di web services. XSRL, basandosi sull'uso di specifici tag XML, fornisce la possibilità di formulare una precisa richiesta a un aggregato di servizi (o business process) scomponendola nei vari obiettivi e sott'obiettivi necessari per il raggiungimento della stessa. E' infatti possibile scomporre gli obiettivi per specificarne un ordine temporale, tramite la relazione basata sull'uso del tag `<BEFORE> goal1 </BEFORE> <THEN> goal2 </THEN>` che è soddisfatta se goal1 è soddisfatto e successivamente anche goal2 viene soddisfatto, o esprimere un obiettivo tramite la comparazione di una variabile ad un valore, ad esempio grazie all'uso della relazione `<LESS> variabile </LESS> <THAN> valore </THAN>` che è soddisfatta solo se la variabile assume un valore minore di quello specificato.

Per rendere più chiari i concetti enunciati creiamo ora, come esempio, una possibile richiesta di un utente a un servizio e la sua realizzazione tramite XSRL.

L'utente vuole acquistare un'auto con cilindrata maggiore a 1600cc e, se possibile, spendere meno di 15.000 euro.

Nell'esempio l'obiettivo dell'utente è essenzialmente diviso in due obiettivi:

1. La cilindrata DEVE essere maggiore di 1600cc.
2. PREFERISCE un prezzo minore di 15000 euro.

I due obiettivi hanno una differenza fondamentale: il soddisfacimento del primo è condizione necessaria per il soddisfacimento della richiesta (possiamo dire che il primo obiettivo sia vitale), mentre il secondo, che possiamo definirlo opzionale, è considerato sempre soddisfatto, in quanto non pregiudica in alcun caso il buon esito dell'intera richiesta.

XSRL esprime questa differenza mediante l'uso di due diverse relazioni espresse rispettivamente dai tag `<VITAL-MAINT>` e `<OPTIONAL-MAINT>`.

Da ciò deriva che il semplice esempio venga tradotto in XSRL in questo modo:

```
<XSRL>
  <ACHEIVE-ALL>
    <VITAL-MAINT>
      <GREATER>CilindrataAuto</ GREATER >
      <THAN>1600</THAN>
    </VITAL-MAINT>
    <OPTIONAL-MAINT>
      <LESS>PrezzoAuto</LESS>
      <THAN>15000</THAN>
    </OPTIONAL-MAINT>
  </ACHEIVE-ALL>
</XSRL>
```

Dove i tag `<XSRL>` `</XSRL>` rappresentano i delimitatori di inizio e fine del codice XSRL mentre i tag `<ACHEIVE-ALL>` `</ACHEIVE-ALL>` racchiudono all'interno una serie di obiettivi che devono necessariamente essere tutti soddisfatti.

Bibliografia

- [1] Peter Lyman and Hal R. Varian; *How Much Information*; 2003.
<http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/>
- [2] John F. Gantz et al.; Marzo 2008; *The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011*; IDC white paper.
<http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf>
- [3] Dimitri Papadimitriou et al.; Gennaio 2009; “*Future Internet - The Cross-ETP Vision Document*”; eMobility, NEM, Nessi, ISI, EPoSS.
http://www.future-internet.eu/fileadmin/documents/reports/Cross-ETPs_FI_Vision_Document_v1_0.pdf
- [4] CLASSIFICAZIONE INTERNAZIONALE DEI PRODOTTI E DEI SERVIZI AI FINI DELLA REGISTRAZIONE DEI MARCHI
http://www.uibm.gov.it/index.php?option=com_content&view=article&id=2004516&lang=it
- [5] Global Employment Trends, Update May 2009
http://www.ilo.org/empelm/what/pubs/lang--en/docName--WCMS_114102/index.htm
- [6] J. O’Sullivan e D. Edmond; 2006; *Towards a precise understanding of service properties*; tesi di dottorato presso la facoltà di Information Technology, Queensland University of Technology (Sidney).
http://eprints.qut.edu.au/16503/1/Justin_O'Sullivan_Thesis.pdf
- [7] R. Ferrario e N. Guarino; 2008; *Towards an ontological foundation for services science*; Proceedings of future internet symposium 2008. Springer, Heidelberg
<http://www.loa-cnr.it/Papers/FerrarioGuarinoFIS08Proceedings.pdf>
- [8] A. De Nigro e P. Corte (Engineering); marzo 2009; NESSI Open Framework – Reference Architecture - Service Description Investigation Team (Final results)
<http://www.nexof-ra.eu/sites/default/files/D2.1%20Appendix%20B%20-%20Service%20Description%20IT%20Report.pdf>
- [9] Pierre Audoin Consultants SAS (PAC); agosto 2010; Economic and Social Impact of Software & Software-Based Services
<http://cordis.europa.eu/fp7/ict/ssai/docs/study-sw-report-final.pdf>
- [10] P. Foltz, S. Dumais; 1992; *Personalized Information Delivery: An Analysis of Information Filtering Methods*; In Communications of the ACM, December, 1992.
- [11] D. Goldberg et al.; 1992; *Using collaborative filtering to weave an information tapestry*; Commun. ACM 35, 12 (Dec.1992), 61—70.
- [12] P. Resnick e H. R. Varian; 1997; Recommender systems; Communications of the ACM, 40(3):56-58, March 1997.

- [13] Nielsen Company; Febbraio 2010; http://blog.nielsen.com/nielsenwire/online_mobile/u-s-web-searches-top-10-2-billion-in-january/
- [14] comScore; Marzo 2011; http://www.comscore.com/Press_Events/Press_Releases/2011/3/comScore_Releases_February_2011_U.S._Search_Engine_Rankings
- [15] Adomavicius, A. Tuzhilin; Giugno 2005; *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*; IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 6, pp 734-749.
<http://www.inf.unibz.it/~ricci/ATIS/papers/state-of-the-art-2005.pdf>
- [16] N. Tintarev, J. Masthoff; 2007; *A Survey of Explanations in Recommender System*; Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop (2007), pp. 801-810.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4401070
- [17] G. Shani, A. Gunawardana; Novembre 2009; *Evaluating Recommendation Systems*; Technical report, No. MSR-TR-2009-159.
<http://research.microsoft.com/pubs/115396/EvaluationMetrics.TR.pdf>
- [18] R. Sinha, K. Swearingen; 2002; *The role of transparency in recommender systems*; Conference on Human Factors in Computing Systems 2002; ACM, New York.
<http://portal.acm.org/citation.cfm?id=506619>
- [19] A. Felfernig, B. Gula; 2006; *An empirical study on Consumer behavior in the interaction with knowledge-based recommender applications*; ECAI 2006 Workshop on Recommender Systems; IEEE Computer Society, Washington DC.
<http://portal.acm.org/citation.cfm?id=1153176>
- [20] B. J. Fogg et al; 2003; *How do users evaluate the credibility of web sites?: a study with over 2,500 participants*; Proceedings of the 2003 conference on Designing for User Experiences, pages 1-15; ACM, New York.
<http://portal.acm.org/citation.cfm?id=997097>
- [21] M. Chen; 2010; *Research on Recommender Technology in E-commerce Recommendation System*; In Proceeding of the 2nd International Conference on Education Technology and Computer (ICETC); V4 pp409-413
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5529654
- [22] E. Peis, J. M. Morales-del-Castillo, J. A. Delgado-López; 2008; *Semantic Recommender System. Analysis of the state of the topic*; Hipertext.net, num. 6, 2008.
<http://www.hipertext.net/english/pag1031.htm>
- [23] B. Adrian, L. Sauermann, T. Roth-Berghofer; 2007; *ConTag: A Semantic Tag Recommendation System*; Proceedings of I-Semantics' 07 pp 297-304; JUCS.
<http://www.dfki.uni-kl.de/~sauermann/papers/adrian+2007a.pdf>

- [24] M. Szomszor et al; 2007; *Folksonomies, the Semantic Web, and Movie Recommendation*. Proc. of the ESWC'07, pp. 71-84.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.6082&rep=rep1&type=pdf>
- [25] P. Massa, P. Avesani; 2004; *Trust-Aware Collaborative Filtering for Recommender Systems*; Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE; Berlin, Heidelberg: Springer, pp. 3-17.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.5213&rep=rep1&type=pdf>
- [26] S.R. Kruk, S. Decker; 2005; *Semantic Social Collaborative filtering with FOAFRealm*; Proc. Of the Semantic Desktop ISWC2005.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.6306&rep=rep1&type=pdf>
- [27] P. Bedi, H. Kaur, S. Marwaha; 2007; *Trust based Recommender System for the Semantic Web*; Proc. of the IJCAI07, pp. 2677-2682.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.2839&rep=rep1&type=pdf>
- [28] S. Kim, J. Kwon; 2007; *Effective Context-aware Recommendation on the Semantic Web*; IJCSNS International Journal of Computer Science and Network Security, VOL. 7 No.8, pp 154-159.
http://paper.ijcsns.org/07_book/200708/20070822.pdf
- [29] A. Loizou, S. Dasmahapatra; 2006; *Recommender systems for the Semantic Web*; ECAI 2006 Recommender Systems Workshop, Trento, Italy.
http://eprints.ecs.soton.ac.uk/12584/2/semanticRSecai2006_submission.pdf
- [30] ref3.21 Z. Yu et al.; 2007; *Ontology-Based Semantic Recommendation for Context-Aware E-Learning*; Proc. of the 4th Conference on Ubiquitous Intelligence and Computing, VOL 4611, pp. 898-907; Springer, Berlin.
<http://ebookbrowse.com/ontology-based-semantic-recommendation-for-context-aware-e-learning-pdf-d74100660>
- [31] Z. Laliwala, V. Sorathia, S. Chaudhary; 2006; *Semantic and rule based event-driven services oriented agricultural recommendation system*; Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst. Workshops; pp. 24–29.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1648914&tag=1
- [32] W. Woerndl, C. Schueller, R. Wojtech; 2007; *A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications*; IEEE 23rd International Conference on Data Engineering Workshop 17-20 April 2007; pp.871-878.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4401078&isnumber=4400943>
- [33] <http://notebookitalia.it/facebook-mappa-le-amicizie-10130>
- [34] H. Oufaida, O. Nouali; Luglio 2009; *Exploiting Semantic Web Technologies for Recommender Systems. A Multi View Recommendation Engine*; The Twenty-first

International Joint Conference on Artificial Intelligence (IJCAI-09); Pasadena, California.

<http://www.dcs.warwick.ac.uk/~ssanand/itwp09/papers/Oufaida.pdf>

- [35] J. Aranda et al.; 2007; *An Online Social Network-based Recommendation System*; report; University of Toronto, Canada.
<http://www.cs.toronto.edu/syslab/courses/csc2231/07au/projects/2/aranda.pdf>
- [36] Paolo Ottolina; Marzo 2011; *Gli smartphone so' piezz' e (dual)core: Lg Optimus Dual, la prova*; Corriere della Sera.it, 31/03/2011.
http://malditech.corriere.it/2011/03/smartphone_piezz_e_dual_core_1.html
- [37] G. Hussein; 2009; *Mobile Recommender System Analysis & Design*; First International Conference on Networked Digital Technologies, 2009. NDT '09.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05272223>
- [38] F. Ricci; Gennaio 2010; *Mobile Recommender Systems*; In Proceedings of the First Italian Information Retrieval Workshop, pages 71-75, Padova, Italia.
<http://www.inf.unibz.it/~ricci/papers/mobile-recommenders.pdf>
- [39] M. . Modsching et al.; 2007; *Effectiveness of Mobile Recommender Systems for Tourist Destinations: A User Evaluation*; In Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2007).
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=4488505
- [40] X. Su e Taghi M. Khoshgoftaar; Gennaio 2009; *A Survey of Collaborative Filtering Techniques*; In Advances in Artificial Intelligence archive Volume 2009; Hindawi Publishing Corp. New York, NY, United States.
<http://portal.acm.org/citation.cfm?id=1722966>
- [41] G. Adomavicius e Y. Kwon; 2011; *Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques*; In IEEE Transactions on Knowledge and Data Engineering 03 Jan. 2011; IEEE computer Society Digital Library; IEEE Computer Society. <http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.15>
- [42] E. Vozalis e K. G. Margaritis; 2003; *Analysis of Recommender Systems' Algorithms*; In The 6th Hellenic European Conference on Computer Mathematics & its Applications (HERCMA), Athens, Greece, 2003, pp. 732-745.
<http://www.mendeley.com/research/analysis-of-recommender-systems-algorithms-1/>
- [43] D. Asanov; 2011; *Algorithms and Methods in Recommender Systems*; Berlin Institute of Technology, progetto didattico. Berlino, Germania.
http://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1011/snet-project/recommender-systems_asanov.pdf
- [44] A. C. M. Costa et al; 2007; *COReS: Context-aware, Ontology-based Recommender system for Service recommendation*; In Proceedings of the 19th international conference on advanced information systems engineering (CAiSE'07).
<http://www.inf.ufes.br/~rguizzardi/publications/costa-guizz-guizz-gonc-UMICS07-cameraready.pdf.pdf>

- [45] S.H. Shahvalady et al; 2009; *A web service recommender system using user ontology*; In Proceedings of the seventh international conference on computer science and information technologies(CSIT2009), p.234–8.
<http://www.csit.am/2009/proceedings/4AIMSS/12.pdf>
- [46] M. H. Kuo, L.C. Chen, C.W. Liang; 2009; *Building and evaluating a location-based service recommendation system with a preference adjustment mechanism*; In Expert Systems with Applications 2009.
<http://sclab.yonsei.ac.kr/courses/11mobile/11mobile.files/paper/12.pdf>
- [47] M. B. Blake, M.F. Nowlan; 2007; *A web service recommender system using enhanced syntactical matching*; In Proceedings of the 2007 IEEE international conference on web services(ICWS2007); p.575–82.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4279646>
- [48] M. Sellami; 2009; *A recommender system for web services discovery in a distributed registry environment*; In Proceedings of the fourth international conference on internet and web applications and services (ICIW 2009); p.418–23.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5072554>
- [49] S. M. Han et al.; 2009; *Efficient service recommendation system for cloud computing market*; In Proceedings of the second international conference on interaction sciences (ICIS2009) p.839–45.
http://portal.acm.org/ft_gateway.cfm?id=1656078&type=pdf
- [50] W. M. Wheeler, 1911; *The Ant Colony as an Organism*; Journal of Morphology, 22: 307–325
- [51] E. Durkheim, 1912; *Le forme elementari della vita religiosa*
- [52] P. T. de Chardin, 1957; *L’Hominisation*; Francia
- [53] H.G. Wells, 1937; *World Brain, The Idea of a Permanent World Encyclopedia*
- [54] P. Levy, 1996; *L’intelligenza collettiva. Per un’antropologia del cyberspazio*; Feltrinelli editore
- [55] J. Surowiecki, 2004; *The Wisdom of Crowds*; Doubleday; Anchor
- [56] <http://www.ilsole24ore.com/art/tecnologie/2011-01-06/demografia-facebook-064613.shtml?uuid=AYR3dYxC>
- [57] <http://www.internetworldstats.com/stats.htm>
- [58] K. R. Lakhani, L. B. Jeppesen, P. A. Lohse, J. A. Panetta; 2007; *The Value of Openness in Scientific Problem Solving*; pag. 28; Harvard University.
<http://www.hbs.edu/research/pdf/07-050.pdf>
- [59] http://www.slideshare.net/sarasarouf/innocentive-power-point?src=related_normal&rel=1748872

- [60] http://www.slideshare.net/GaryCooper/innocentive?src=related_normal&rel=4256823
- [61] H. Halpin, V. Robu, H. Shepherd; 2007; *The complex dynamics of collaborative tagging*; Proceedings of the 16th international conference on World Wide Web; 211—220; ACM, New York.
<http://portal.acm.org/citation.cfm?id=1242572.1242602>
- [62] Crowdcast Inc.; 2010; *How CPG Companies Use Collective Intelligence Tools to Improve Operational Performance*; White Paper.
- [63] <http://www.baynote.com/why-baynote/integrated-suite/>
- [64] <http://www.baynote.com/why-baynote/userank/>
- [65] Allen Bonde; May 2010; *Why Collective Intelligence is Essential to Social CRM*; White Paper.
- [66] WeKnowIt; 2009; *First white paper on Collective Intelligence*; White Paper.
<http://www.weknowit.eu/sites/default/files/D9.4.1.pdf>
- [67] T. W. Malone, R. Laubacher, C. Dellarocas; 2009; *Harnessing Crowds: Mapping the Genome of Collective Intelligence*; MIT Center for Collective Intelligence, Working Paper.
http://www.realtechsupport.org/UB/MRIII/papers/CollectiveIntelligence/MIT_CollectiveIntelligence2009.pdf
- [68] <http://glennas.files.wordpress.com/2009/11/collective-intelligence.jpg>
- [69] S. Alag; Ottobre 2008; *Collective Intelligence in Action*; Manning, Greenwich CT.
- [70] Nikolaos Loutas, Vassilios Peristeras, Konstantinos Tarabanis; 2010; *Rethinking the Semantic Annotation of Services*; Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops: 540-549; Springer, Berlino.
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-540/ugs2009_submission_4.pdf
- [71] A. V. Moorsel; 2001; *Metrics for the Internet Age: Quality of Experience and Quality of Business*; HP Labs, Tech. Rep.
<http://www.hpl.hp.com/techreports/2001/HPL-2001-179.pdf>
- [72] P. Leitner, A. Michlmayr, F. Rosenberg, S. Dustdar; Dicembre 2009; *Selecting Web Services Based on Past User Experiences*; Services Computing Conference, 2009. APSCC 2009: 205-212; IEEE Asia-Pacific
http://www.infosys.tuwien.ac.at/Staff/sd/papers/apsc09_Leitner.pdf
- [73] H.A. Simon; 1971; *Designing organizations for an information-rich world*; Computers, Communications, and the Public Interest (1971), 38–52.
- [74] L. Betancourt; Marzo 2010; *How Companies Are Using Your Social Media Data*.
<http://mashable.com/2010/03/02/data-mining-social-media/>

- [75] D.A. Norman; Marzo 2011; *Più smartphone che computer: cambiano i linguaggi digitali*.
http://www.corriere.it/cultura/11_marzo_22/rastelli-smartphone-computer_a4461720-547a-11e0-a5ef-46c31ce287ee.shtml
- [76] Nielsen Company; Settembre 2009; *With Smartphone Adoption on the Rise, Opportunity for Marketers is Calling*.
http://blog.nielsen.com/nielsenwire/online_mobile/with-smartphone-adoption-on-the-rise-opportunity-for-marketers-is-calling/
- [77] eMarketer; Giugno 2010; *Smartphone ownership and mobile ad spending also rise*.
<http://www.emarketer.com/Article.aspx?R=1007747>
- [78] Gartner; Gennaio 2011; *Gartner Says Worldwide PC Shipments in Fourth Quarter of 2010 Grew 3.1 Percent; Year-End Shipments Increased 13.8 Percent*
<http://www.gartner.com/it/page.jsp?id=1519417>
- [79] IDC; Gennaio 2011; *PC Market Records Modest Gains During Fourth Quarter of 2010, According to IDC*.
<http://www.idc.com/getdoc.jsp?containerId=prUS22653511>
- [80] <http://flowingdata.com/2008/05/01/chart-of-the-day-a-breakdown-of-facebook-applications/>
- [81] G. Salton, A. Wong, e C. S. Yang; 1975; *A vector space model for automatic indexing*; In Magazine Communications of the ACM Volume 18 Issue 11, Nov. 1975; ACM, New York, NY, USA <http://portal.acm.org/citation.cfm?doid=361219.361220>
- [82] P. Bonhard, M.A. Sasse, C. Harries; 2007; *"The devil you know knows best": How Online Recommendations Can Benefit From Social Networking*; Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1; British Computer Society; Swinton, UK.
<http://portal.acm.org/citation.cfm?id=1531305>
- [83] <http://siteanalytics.compete.com/booking.com/>
- [84] <http://www.serverinsiders.com/domain/booking-com.html>
- [85] <http://www.slideshare.net/AngeloOldani/analisi-alcuni-siti-turisticci>
- [86] <http://www.danilopontone.it/web-marketing-turistico/whr-seminar-2010-elementi-di-revenue-giancarlo-tamburi/>
- [87] <http://www.turismoecconsigli.com/statistiche-web-mobile-2010-abitudini-preferenze-degli-utenti/>