UNIVERSITA' DEGLI STUDI DI NAPOLI

"Federico II"

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

DIPARTIMENTO DI INFORMATICA E SISTEMISTICA

TESI DI LAUREA IN SISTEMI INFORMATIVI

Uso di Ontologie per sistemi di E-Learning intelligenti e adattivi –modellazione del dominio "Protocollo Informatico"

Relatore

Ch.mo Prof. Antonio d'Acierno

Correlatore

Ing. Antonio Valentino

Candidata

Deborah Vinciguerra

041/2177

Anno Accademico 2003/2004

Ai miei genitori

Ringraziamenti

Un ringraziamento particolare al Prof. Antonio d'Acierno e al Dott. Roberto Capobianco per avermi dato l'opportunità di questa esperienza in azienda e all'Ing. Antonio Valentino per il suo preziosissimo aiuto e la sua pazienza.

A mio padre e mia madre per avermi sempre sostenuto in tutti questi anni e per avermi sempre incoraggiata e senza i quali niente sarebbe stato possibile.

Alle mie sorelle Marianna e Dalila per il bene che mi vogliono.

A tutti gli amici che in questi anni universitari sono stati per me una seconda famiglia.

A Luigi mio carissimo amico e compagno di studi per aver condiviso con me gioie e fatiche di tantissimi esami.

Un ultimo ringraziamento va a Michele, la persona che più di tutti mi ha sopportata in questi anni e che spero abbia la pazienza per farlo ancora.

SOMMARIO

Introduzione

Negli ultimi anni, si sta assistendo ad un radicale cambiamento nel mondo della didattica e della formazione che sta spingendo numerose scuole, università ed aziende ad adottare le più moderne tecnologie informatiche, basate principalmente sul Web, come nuovo strumento per gestire e condividere la conoscenza.

Questo cambiamento è favorito dagli innumerevoli vantaggi che la formazione a distanza comporta, primo tra tutti un'estrema flessibilità di tempo e di spazio: il discente non è più costretto ad essere presente nel medesimo luogo del docente e può liberamente scegliere i tempi e i luoghi della fruizione. Se a questo aggiungiamo l'aumento della qualità del contenuto formativo, una sua gestione più flessibile, la possibilità di misurare facilmente i risultati e la diminuzione dei costi, comprendiamo perché la formazione a distanza è al giorno d'oggi molto appetita in tutti gli ambienti didattico/formativi. In questa ottica un sistema di e-learning può essere ancora più utile ed efficace se riesce a sfruttare le informazioni in suo possesso per elaborarle in maniera intelligente e personalizzata. In questo modo, infatti, si va verso la personalizzazione dell'insegnamento sulle reali esigenze e capacità del singolo utente.

In tale contesto nasce l'idea di un'architettura web-based per il delivery intelligente e adattivo dei corsi, idea che ha ispirato questo lavoro di tesi svolto presso l'azienda Italdata S.p.A., Gruppo Siemens.

La fase iniziale di questo lavoro di tesi è stata un'attenta selezione del materiale bibliografico sugli aspetti legati alla convergenza tra l'e-learning e la gestione della conoscenza nei nuovi sistemi di e-learning web based intelligenti e adattivi. Dopo tale analisi e valutazione di detti aspetti, si è arrivati a pensare ad un'architettura per un sistema di e-learning in grado di costruire percorsi didattici personalizzati secondo le esigenze formative dell'utente, delle sue conoscenze e delle sue preferenze di apprendimento.

Tale sistema si compone di tre moduli principali: modello del dominio, modello dello studente e mappe di competenza. Il modello del dominio, in particolare, descrive i concetti del dominio su cui gli utenti devono/vogliono formarsi e le relazioni che intercorrono tra di essi. Il modello dello studente contiene, invece, le informazioni sull'utente, il suo stato

cognitivo e le sue preferenze d'apprendimento. Le mappe di competenza, in ultimo, contengono tutte le indicazioni sulle competenze e conoscenze da acquisire al fine di guidare l'utente verso l'ottenimento dello specifico profilo professionale o abilità conoscitiva.

A partire dai tre modelli in questione il sistema, attraverso un processo di "competence gap analysis", individua le competenze che l'utente deve acquisire e utilizza queste informazioni per costruire un percorso di apprendimento costruito, oltre che sulle effettive esigenze formative dell'utente stesso (stato cognitivo), anche sulle sue preferenze, come lo stile di apprendimento e il contesto di fruizione (tipologia di device utilizzato). I dati di fruizione verranno utilizzati per aggiornare dinamicamente il profilo dell'utente e quindi le competenze che ha acquisito così come eventuali variazione allo stile di apprendimento.

Una delle problematiche principali consiste nel riuscire a rappresentare opportunamente i tre modelli che compongono il sistema e individuare tecniche adeguate che permettano di raggiungere gli obiettivi per cui il sistema stesso è pensato. In questa ottica, in questo lavoro di tesi, si è cercato di dare una risposta al problema della rappresentazione del modello del dominio, scegliendo l'ontologia come strumento per rappresentare la conoscenza. Un'ontologia identifica i termini e le relazioni di un dominio definendone un vocabolario. Questo vocabolario comune favorisce l'interoperabilità e lo scambio delle informazioni tra sistemi che l'utilizzano per rappresentare il dominio d'interesse; questo caratteristica rappresenta uno dei vantaggi principali che ha motivato la scelta di questo nuovo strumento per rappresentare la conoscenza.

Il linguaggio di codifica dell'ontologia scelto è OWL (*Web Ontology Language*), che è lo standard attualmente proposto dal W3C per la definizione di ontologie per il web semantico. Per facilitare la costruzione dell'ontologia è stato utilizzato il tool Protégé sviluppato presso l'Università di Stanford, che offre un editor grafico e interattivo per l'ontology design e per l'acquisizione della conoscenza.

Dopo aver individuato le possibili relazioni esistenti tra i concetti di un dominio didattico, si è scelto come semplice caso applicativo di rappresentare un sottoinsieme del "protocollo informatico", costituito dai concetti fondamentali legati all'automazione dei flussi documentali nell'ambito della pubblica amministrazione. La scelta di questo dominio è motivata principalmente dal fatto che Italdata ha sviluppato una soluzione di protocollo informatico da cui è stato possibile astrarre alcuni concetti base legati appunto a tale

tematica.

Per provare l'ontologia è stato, inoltre, sviluppato un applicativo Java in grado di interrogare l'ontologia e guidare in maniera "intelligente" l'utente attraverso il materiale di apprendimento per il raggiungimento degli obiettivi legati al "profilo professionale" da lui scelto.

Capitolo I

1 E-Learning Personalizzato

L'E-Learning è generalmente inteso come processo di formazione basato su tecnologie informatiche, in grado di fornire un accesso al materiale didattico e offrire un'interazione a distanza tra i soggetti coinvolti nei processi formativi.
L'interesse mostrato per tale disciplina all'interno di scuole e università, e ancor di più all'interno delle aziende, è giustificato dai numerosi vantaggi che ne comporta in termini di flessibilità di spazio e di tempo, condivisione e riuso dei contenuti, possibilità di misurare facilmente i risultati, riduzione dei costi e miglioramento delle competenze.
Tutti questi vantaggi sono il risultato dell'evoluzione che ha caratterizzato questi sistemi nel corso degli anni, passando dai primi sistemi di formazione a distanza con la didattica per corrispondenza prima e la didattica multimediale poi, fino ad arrivare alla grande evoluzione avvenuta con l'introduzione della didattica in rete.
Si supera così il termine formazione a distanza e lo si sostituisce con il termine e-learning.
Il Web diventa non solo un contenitore di risorse didattiche facilmente reperibili ma il mezzo con cui persone motivate dagli stessi interessi possano seguire le stesse lezioni, confrontarsi e discutere in modalità asincrona, offrendo la possibilità all'utente di avere libertà nella scelta dei tempi di fruizione e di interagire con altri utenti attraverso e-mail, chat, forum, o in modalità sincrona, imponendo però che le persone siano collegate nello stesso momento ma che possano comunicare tra loro attraverso strumenti di chatting, videoconferenze o aule virtuali.
Questi sistemi di e-learning sono centrati sull'erogazione di contenuti sul web e generalmente si compongono di un modulo CDS (Content Delivery System) per il delivery dei corsi e un modulo LMS (Learning Management System) per la gestione delle learning activity e degli utenti del sistema. Tali sistemi, tuttavia, ad oggi non sembrano possedere caratteristiche avanzate per la gestione delle competenze e per la

personalizzazione del learning path basata sull'analisi delle competenze, nonché l'adattamento delle competenze stesse in funzione di feedback provenienti dai comportamenti e dalle necessità di ciascun utente.

Con l'evoluzione delle tecnologie informatiche basate sul web, si comincia a parlare di **e-learning personalizzato** in cui la formazione, distribuita e resa accessibile a popolazioni sempre più ampie, **libera da vincoli di tempo e luogo,** sempre più integrata con le pratiche di lavoro, agganciata ai processi di **analisi, sviluppo e valutazione delle competenze** e collegata con i sistemi di **gestione e condivisione della conoscenza,** assume un ruolo sempre più importante e vitale per la sopravvivenza di un'azienda e consente di realizzare sistemi di apprendimento continuo: si tratta di un vero e proprio cambiamento di paradigma, sostenuto anche da nuovi valori. La conoscenza va, infatti, diffusa e resa accessibile a tutti e le competenze delle persone diventano per le aziende il patrimonio da gestire più importante.

L'e-learning personalizzato mira ad offrire un percorso d'apprendimento specifico per ogni utente individuale attraverso un processo di **competence gap analysis** tra le competenze che l'utente possiede (rappresentate nel suo modello) e quelle che deve acquisire (rappresentate attraverso mappe di competenza).

Esso opera in maniera **intelligente** e **adattiva** facendo confluire in un unico modello tecniche e tecnologie ereditate dagli ITS (Intelligent Tutoring Systems) e AHS (Adaptive Hypermedia Systems). In maniera **intelligente** sfruttando tecniche dell'Intelligenza Artificiale per ragionare sui tre modelli che compongono il sistema (modello del dominio, modello delle competenze e modello dello studente) e che contengono la conoscenza necessaria per poter comporre i corsi, valutare il livello di apprendimento dello studente e le sue competenze.

In maniera **adattiva** sia a livello di contenuti, adattando il materiale didattico in funzione del **learning style,** ossia dello stile di apprendimento dell'utente valutato in base alle sue attitudini e preferenze, e del **contesto di fruizione,** ossia la realtà in cui si trova a fruire l'utente e i dispositivi che ha a disposizione, sia adattando le mappe di competenza in funzione del **free-learning** quando l'utente ricerca autonomamente concetti non contenuti nel percorso didattico fornito dal sistema o ritenuti non soddisfacentemente esplicativi.

In questa ottica un sistema di e-learning riesce a raggiungere effettivamente gli obiettivi per i quali è nato: personalizzato alle esigenze di ogni singolo utente e fruibile nei

migliori dei modi e in ogni contesto e adattativo rispetto ad eventuali suggerimenti che provengono dall'analisi dei dati della fruizione libera.
Obiettivo di questo capitolo è descrivere i vantaggi dell'e-learning e l'importanza che esso assume in ambito aziendale per gestire la conoscenza ed accrescere la competitività e la produttività dell'azienda e presentare le caratteristiche fondamentali che portano alla definizione dei più moderni sistemi di e-learning personalizzati basati sul web.

1.1 E-Learning

L'apprendimento e l'educazione sono da sempre un elemento fondamentale di ogni società. Con il passare degli anni sono evolute non solo le conoscenze, ma anche le metodologie d'istruzione: in parte per un cambiamento sociale, in parte per lo sviluppo tecnologico. E' innegabile che l'avanzamento dell'elettronica negli ultimi decenni ha subito una crescita esponenziale, introducendo strumenti che hanno radicalmente cambiato le nostre abitudini. L'informatizzazione, prima riservata a pochi settori (militare, centri di ricerca ed altro), si è velocemente estesa, per raggiungere (quasi) ogni posto di lavoro, nonché gli ambienti domestici. Il fenomeno Internet, ormai lontano dall'essere solamente uno strumento per pochi esperti, è uno dei principali responsabili della trasformazione sociale che sta investendo tutti. Anche le metodologie d'istruzione sono coinvolte da questo processo e si sta affermando un nuovo modello d'apprendimento, l'e-learning.
L'e-learning è un fenomeno in continua espansione, con un notevole spessore commerciale e quindi appetibile ed interessante. Ma non basta Internet ed un personal computer per ottenere l'e-learning: il concetto di apprendimento a distanza è più complesso di quanto si possa supporre a partire da un'analisi superficiale, coinvolgendo un'ampia varietà di campi, dall'interazione uomo-macchina all'Intelligenza Artificiale, passando per questioni tecnologiche e sociali.
L'E-Learning risponde ad esigenze economiche, lavorative, sociali e metodologiche che finora avevano influenzato non poco l'effettivo sviluppo di progetti e aspirazioni di formazione continua, sia a livello istituzionale che aziendale e individuale. Dai tempi di utilizzo ai costi da sostenere, dalle metodologie di insegnamento a quelle di apprendimento: tutte aree in cui l'e-learning riesce a fare la differenza. Potenzialità che risultano come comune denominatore dei processi di *Web based education*.

Tale posizione è ampiamente giustificata dai vantaggi che l'e-learning introduce rispetto alla formazione tradizionale:

1. **Flessibilità spaziale e temporale:** ogni utente può scegliere liberamente i tempi di fruizione nonché i luoghi, basta avere a disposizione le tecnologie adatte a fruire il corso. Con i sistemi web based basta un computer connesso in rete per garantire l'accesso ai contenuti formativi, ciò si traduce in un ulteriore vantaggio ossia la possibilità che i contenuti siano accessibili ad un pubblico praticamente illimitato.
2. **Possibilità di misurare facilmente i risultati:** attraverso esercitazioni e test si misura facilmente il livello di apprendimento degli utenti e tali dati vengono utilizzati nei sistemi adattivi per apportare eventuali variazioni al modo di proporre i contenuti se non ritenuti sufficienti al conseguimento della spacifica competenza professionale.
3. **Condivisione e riuso dei contenuti:** i contenuti disponibili in rete possono essere facilmente condivisi tra gli utenti, nonché riutilizzabili. Grazie all'aderenza ai maggiori standard in circolazione in materia di e-learning (AICC, SCORM, IMS) i contenuti possono essere facilmente condivisi tra i vari sistemi e quindi riutilizzabili.
4. **Riduzione dei costi di formazione:** L'e-learning rappresenta un risparmio, soprattutto in abilito aziendale, rispetto alla formazione tradizionale eliminando costi per gli spostamenti, costi di gestione delle aule, costi di materiali cartacei, minimizzando gli allontanamenti dal posto di lavoro, realizzando economie di scala ove tutti i dipendenti usufruiscono di uno stesso corso.
5. **Miglioramento delle competenze del personale:** un sistema di e-learning è in grado di misurare le competenze di un utente e valutare eventuali carenze, di conseguenza sarà in grado di fornirgli il giusto materiale didattico tale da colmare le sue lacune e metterlo nello condizioni di accrescere la sua conoscenza..

1.2 Gestione della conoscenza

Il tema della gestione della conoscenza, sebbene affrontato dalla letteratura accademica da diversi decenni, diviene argomento manageriale all'inizio degli anni novanta (Nonaka & Takuechi, 1995). L'idea di fondo che ispira questa nuova disciplina

manageriale è che ogni organizzazione, al suo interno, produce un sapere originale e distintivo che, se opportunamente valorizzato può contribuire alla competitività d'impresa. Spesso, infatti, una soluzione prodotta in un luogo dell'azienda potrebbe essere utilmente riutilizzata in un altro. O ancora, idee che non sono utili in un determinato momento, potrebbero divenirlo in un momento successivo.

La gestione della conoscenza comprende, allora, tutti quegli strumenti e metodologie gestionali che facilitano un efficiente creazione e scambio del sapere a tutti i livelli dell'organizzazione col fine di creare valore per l'impresa.

Nel processo di gestione della conoscenza si possono individuare due dimensioni rilevanti:

- una dimensione riguarda la necessità di *estrarre e strutturare il proprio patrimonio di conoscenze*: indagare il contenuto e la natura della conoscenza rilevante per l'attività dell'azienda, nelle sue diverse articolazioni e tipologie, al fine di renderla comunicabile. Questa dimensione comprende tutte le iniziative relative al sistema informatico nonché al recupero e alla formalizzazione della conoscenza.
- l'altra dimensione riguarda la necessità di *sviluppare e utilizzare il proprio patrimonio di conoscenze*: comunicare, distribuire e utilizzare la conoscenza nelle diverse aree aziendali e a tutti i livelli organizzativi. Questa dimensione riguarda tutte le iniziative che toccano l'organizzazione aziendale e le risorse umane, volte a favorire la condivisione e l'acquisizione di conoscenza.

Affinché la conoscenza si trasformi in un vantaggio competitivo per le aziende, essa deve essere "gestita", ciò significa principalmente evitare la perdita e l'obsolescenza delle risorse intellettuali, questo spiega perché è sempre più frequente trovare collegato questo concetto a quello di e-learning e perché si va verso una convergenza delle due discipline.

1.3 Convergenza tra e-learning e gestione della conoscenza

Collegare la gestione della conoscenza con l'e-learning sembra immediato: l'apprendimento è un processo sociale di condivisione di conoscenza (e-learning) e la condivisione della conoscenza è una risorsa essenziale per l'organizzazione e va supportata con strumenti adeguati (KM).

Tuttavia, nella maggior parte delle aziende la divisione tra le due discipline è dovuta a motivi di organigramma: le persone che si occupano di gestione della conoscenza,

magari come attività collaterali a quelle richieste dalla funzione ufficiale, appartengono tipicamente al dipartimento dell'Information Technology o delle risorse umane.

È un fatto, tuttavia, che le iniziative di gestione della conoscenza da sempre rivestono un alto carattere strategico. Nel momento in cui si decide di definire un piano di gestione del capitale intellettuale non si può fare a meno di avere di un approccio top-down, ovvero i livelli più alti del management devono sponsorizzare l'iniziativa ed utilizzare la loro influenza per far percepire l'importanza del progetto a tutti i livelli sottostanti.

Questo per superare almeno inizialmente le resistenze al cambiamento di mentalità che sono inevitabili: la gestione della conoscenza cambia il modo di lavorare delle persone e spesso il contributo che si è tenuti a dare viene visto solo come un aggravio nel lavoro quotidiano.

L'e-learning invece è stato spesso concepito come un processo che non ha valenza strategica, ma è legato alle esigenze del momento (corsi ad hoc per colmare lacune in ambiti particolari), da cui deriva un approccio tattico per quanto riguarda le attività di pianificazione: le iniziative sono decise e promosse dal dipartimento Training e spesso riguardano i bisogni specifici di particolari unità organizzative. A rafforzare questo tipo di impostazione concorre, inoltre, un diffuso pregiudizio, secondo il quale tramite le iniziative di formazione aziendale si possono erogare contenuti legati al know-how operativo, ma non fornire un reale supporto ai livelli manageriali dell'organizzazione, i cui compiti prettamente decisionali di definizione delle strategie necessitano dell'accesso ragionato alle informazioni (tipico di iniziative di Knowledge Management), più che di corsi su come interpretare le informazioni.

È opinione ancora diffusa che questo tipo di competenze derivino dall'esperienza sul campo e difficilmente possono essere insegnate, tanto meno insegnate on-line. In realtà, in un'ottica di apprendimento continuo questa posizione non è più sostenibile: il supporto formativo alle persone deve essere dato continuamente a tutti i livelli, questo non implica che le modalità debbano però essere uguali. In questo, il collegamento delle metodologie di e-learning con gli strumenti forniti dal Knowledge Management può portare ad ottimi risultati nella disseminazione di contenuti personalizzati.

Entrambi gli approcci si occupano di organizzare, gestire e sviluppare al meglio il capitale intellettuale di un'azienda, di far circolare l'informazione anche se usano approcci differenti, ma a ben vedere complementari. L'e-learning ha come scopo quello

di colmare le carenze nelle competenze utili al raggiungimento degli obiettivi aziendali. In passato è stato spesso concepito come uno strumento da utilizzare in modo mirato e per coprire mancanze specifiche, ma ormai si sta sempre più facendo strada l'idea che la formazione sia un processo continuo, che deve affiancare i momenti di operatività e non interromperli, proponendo solo un sapere a blocchi chiusi nel tempo. Da questa esigenza deriva che il corso monolitico, seguito in modo avulso dalle attività lavorative, non dà garanzie che ciò che venga appreso sia poi messo in pratica durante il lavoro (i limiti della memoria umana e le tempistiche aziendali qui giocano una gran parte).

Per non disperdere quanto imparato, risulterebbe, quindi, più utile affiancare al corso un sistema di supporto continuo, che assista il discente quando torna ad essere operativo. Ideale sarebbe un approccio che segua i principi del concetto di Electronic Performance Support Systems (EPSS), secondo cui vengono fornite risposte mirate e immediate solo sulla parte di contenuti rilevante in un determinato contesto e soprattutto risposte aggiornate (la realtà può essere molto cambiata dal periodo di frequenza del corso).

Per realizzare questo tipo di sistema, tuttavia, bisogna avere una chiara idea delle dinamiche di lavoro aziendale ed essere in grado di reperire le informazioni più aggiornate laddove nascono e vengono utilizzate. In questo il knowledge management può dare un grande aiuto agli esperti dell'e-learning, facendo leva sugli apporti derivati da sistemi di document management, di automatizzazione dei processi, di localizzazione degli esperti della materia che possono fornire risposte, e allo stesso tempo può fornire le strutture per rimettere in circolo l'informazione tramite gli strumenti di collaborazione e di ricerca più diffusi.

Il Knowledge management, dal canto suo, deve occuparsi di recuperare ed ordinare il sapere sia tacito che esplicito, eppure spesso fallisce nel dare risposte efficaci. I materiali non sono abbastanza significativi oppure non sono facilmente identificabili e recuperabili. L'e-learning in questo può dare un grande apporto che fa leva sull'abilità propria degli Instructional designer di legare i contenuti a specifici obbiettivi di apprendimento, di riutilizzare le informazioni, che derivano spesso da interviste ad esperti della materia (dipendenti con esperienza), per presentarle in modo più efficace e diretto.

Aumentare la collaborazione tra i due ambiti potrebbe, per esempio, abituare i responsabili delle strategie di Knowledge Management a lavorare tenendo presente fin da subito il concetto di Learning Object quando si occupano di definire contenuti mirati.

Con il termine Learning Object si intende l'unità minima di contenuto che serve per far apprendere un concetto. Fondamentale è che questi oggetti portano con sé anche tutte le informazioni accessorie (metadati) che ne descrivono l'utilizzo ottimale. Durante il processo di creazione di un Learning Object, gli Instructional Designer individuano il materiale adatto (anche magari un documento pensato per un ambito operativo o le modalità di lavoro raccontate da chi ha un'esperienza adeguata) e lo ripensano per proporlo in una forma che amplifichi il suo valore formativo. Si riconoscono le informazioni di base che devono essere comunicate (ciò che deve essere appreso), si utilizza una metrica precisa per valutare l'efficacia del messaggio che viene comunicato al discente (i sistemi di test tipici della formazione di riflesso riescono a mettere in evidenza anche la validità del metodo scelto per veicolare i contenuti), lo scopo formativo è chiaro e dichiarato e, in un'ottica di riutilizzo, viene spesso identificato tramite un'attenta classificazione con metadati.

Possiamo mettere in luce i seguenti benefici che ne derivano dall'integrazione di soluzioni di e-learning e knowledge management:

- Arricchimento dell'ambiente di apprendimento;
- Creazione di contenuti tempestiva e diretta;
- Ottimizzazione della gestione di sistema;
- Risparmi sui costi e ritorno sugli investimenti;
- Aumento della competitività.

1.4 Caratteristiche di un sistema di e-learning personalizzato

L'evoluzione che ha caratterizzato i sistemi di e-learning negli ultimi anni, ha portato alla definizione di un sistema evoluto in grado di rispondere alle esigenze formative degli utenti adottando nuove tecnologie in grado di mettere in primo piano concetti come **gestione delle competenze**, **personalizzazione dei percorsi formativi** e capaci di sfruttare tutte le informazioni in loro possesso per adeguare, in maniera *intelligente*, il sistema ad eventuali cambiamenti necessari per essere *adattivo* verso gli obiettivi professionali dell'utente.

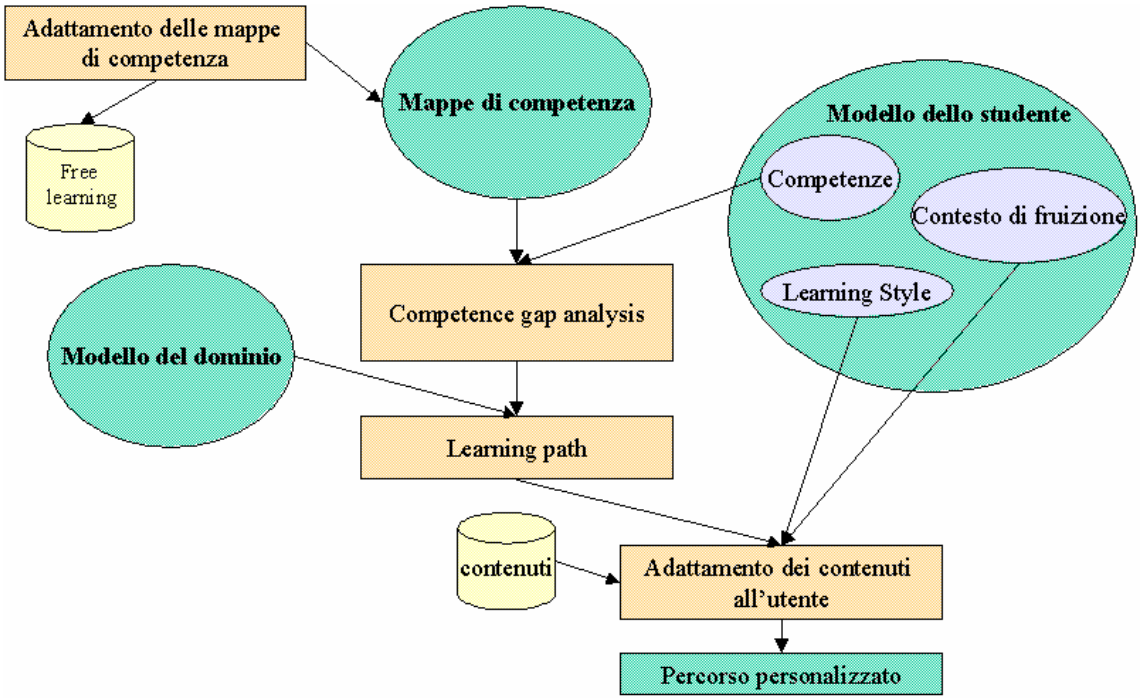


Figura 1. 1- Processi adattivi e intelligenti di un sistema di e-learning personalizzato

1.4.1 Gestione delle competenze

Le competenze rappresentano le capacità di compiere una data attività o di svolgere un dato compito ritenuto indispensabile per il processo produttivo di un'azienda.

Per poter gestire le competenze è necessario prima riuscire a rappresentarle. Le *mappe di competenza* rappresentano lo strumento utilizzato per poter definire le competenze, i legami che intercorrono tra loro e collegarle agli obiettivi formativi, ritenuti necessari al fine di conseguire la specifica competenza professionale e al modello del dominio, da cui si ricavano i concetti d'apprendere per raggiungere la competenza indicata.

La gestione delle competenze in un contesto aziendale è di fondamentale importanza per riuscire a delineare i ruoli all'interno dell'organizzazione e si traduce sicuramente in un aumento di produttività. Avere allora a disposizione uno strumento per misurare le competenze ed eventualmente accrescerle in base alla specifica esigenza del momento, diviene un aspetto importante e strategico per l'intero processo produttivo.

Le competenze devono inoltre essere mantenute aggiornate, per cui ogni utente del sistema avrà nel modello ad esso associato informazioni su proprio stato cognitivo, che deve essere automaticamente aggiornato in seguito all'acquisizione di nuove competenze.

1.4.2 Competence Gap Analysis

Il processo di competence gap analysis è mirato a valutare il divario di competenze che deve essere colmato affinché l'utente individuale raggiunga l'obiettivo professionale prefissato. Esso valuta le competenze che l'utente deve acquisire, rappresentate mediante le mappe di competenze e le competenze che egli possiede (stato cognitivo), rappresentate nel modello dell'utente. In questo modo l'utente seguirà un percorso d'apprendimento individuale basato sulle sue competenze.

1.4.3 Personalizzazione dei contenuti

La competenze gap analysis ci fornisce il learning path che l'utente dovrà seguire ma nulla ci dice sul materiale da proporre. La scelta dei contenuti didattici è effettuata valutando il learning style e il contesto di fruizione, informazioni contenute nel modello associato all'utente.

Il **learning style** indica le preferenze di apprendimento dell'utente valutate in base alle sue attitudini, se è portato più per materiali visivi, audio-visivi, in che lingua vuole che il materiale gli venga proposto, e così via. Il **contesto di fruizione** definisce il contesto in cui l'utente effettua la richiesta e i dispositivi di fruizione che egli utilizza. La conoscenza dovrà essere erogata "on demand" e accessibile sia da dispositivi fissi che da dispositivi mobili. Il sistema dovrà essere in grado di individuare il contesto da cui proviene la richiesta (es: posizione, canali trasmessivi disponibili per l'utente) e fornire il materiale d'apprendimento in funzione di questo.

1.4.4 Free-learning e adattamento delle mappe di competenza

All'interno di un sistema di e-learning gli utenti possono operare in due modalità differenti di fruizione:

- una fruizione "system-driven" di un percorso formativo ovvero di un intero percorso curriculare in cui il sistema guida lo studente/fruitore tra il materiale disponibile in funzione degli obiettivi didattici da raggiungere a partire dagli stati cognitivi iniziali;
- una fruizione libera per la ricerca di conoscenza o di informazione per far fronte ad una necessità operativa (meglio noto come Knowledge on Demand) oppure ad una esigenza autonoma di arricchimento culturale o di problem solving (il cosiddetto "Free-Learning").

I dati della fruizione libera andranno ad aggiornare le mappe di competenza. L'ipotesi di buon senso che si fa è che chi va ricercare liberamente delle informazioni lo fa in quanto utile per il proprio ruolo professionale o per risolvere un particolare problema legato alla propria professione. Allora se un certo numero di studenti che ricoprono lo stesso ruolo, hanno ricercato lo stesso di tipo di informazione non prevista nel proprio percorso formativo si può ritenere che tale informazione sia indispensabile nel percorso formativo e provvedere ad aggiornare le mappe di competenza.

Capitolo II

2 Web-based E-Learning adattivo e intelligente

I sistemi didattici adattivi e intelligenti basati sul web sono il risultato della confluenza e l'integrazione di diverse tecniche e tecnologie impiegate nei Sistemi Ipermediali Adattivi (AHS), negli Intelligent Tutoring System (ITS) e nei Web-based Learning Management System compatibilmente con gli standard sviluppati nell'ambito dell'e-learning (come SCORM, LOM). Tali sistemi, infatti, cercano di adattarsi in maniera più flessibile alle necessità formative e informative dell'utente costruendo e usando un modello di obiettivi, preferenze e conoscenze proprie per ogni studente (modello user-centered). Nello stesso tempo, cercano di essere anche "più intelligenti" incorporando e migliorando tecniche utilizzate nell'ambito degli ITS, come ad esempio, la generazione automatica di corsi. Le problematiche di interoperabilità delle tecnologie adottate dagli LMS e della riusabilità dei contenuti formativi (author-centered) sono altrettanto importanti e centrali nei nuovi sistemi di erogazione. Per far fronte a queste questioni, attualmente, la maggior parte dei sistemi LMS stanno adottando le specifiche standard emesse dalle maggiori organizzazione e iniziative di standardizzazione nel campo dell'e-learning come IMS, IEEE e ADL.

In questo capitolo dopo una breve descrizione delle caratteristiche fondamentali dei sistemi AHS, ITS e Web-based Learning Management vengono evidenziate le tecniche, gli approcci e le linee di ricerca che confluiscono nell'area dei nuovi sistemi avanzati per l'erogazione e la fruizione di contenuti formativi sul Web.

2.1 Origini dei sistemi didattici intelligenti e adattivi web-based

I primi pionieri di sistemi didattici intelligenti e adattivi basati sul web, furono sviluppati nel 1995-1996 (Brusilovsky et al 1996a,1996b, De Bra 1996). Da allora molti

sistemi interessanti sono stati sviluppati e resi noti. Un interesse a fornire un'educazione a distanza sul Web è stata la forza guida che sta dietro a questi sforzi di ricerca. La comunità di ricerca è stata aiutata da una serie di lavori che hanno portato un insieme di ricercatori a lavorare su AIWBES (Brusylovsky et. al 2002, Peylo 2000).

I sistemi avanzati di didattica basati sul Web, sono molto spesso riferiti come *adaptive Web-based educational systems* o come *intelligent Web-educational systems.* Questi termini non sono in realtà dei sinonimi.

Gli *adaptive systems* sono sistemi che cercano di adattarsi al singolo studente tenendo conto delle informazioni accumulate nel modello dello studente.

Gli *intelligent systems* applicano tecnologie nel campo dell'Intelligenza Artificiale (AI) per fornire un supporto più ampio e migliore agli utenti dei sistemi didattici basati sul Web.

Mentre la maggior parte dei sistemi possono essere classificati entrambi come intelligenti e adattivi, un ampio numero di sistemi cade esattamente in una di queste categorie (Figura.2.3). Per esempio molti sistemi intelligenti di diagnosi includendo German Tutor (Heift & Nicholson 2001) e SQL-Tutor (Mitrovic 2003) sono non adattivi, cioè, essi forniranno la stessa diagnosi in risposta alla stessa soluzione a un problema nonostante l'esperienza passata dello studente con il sistema.

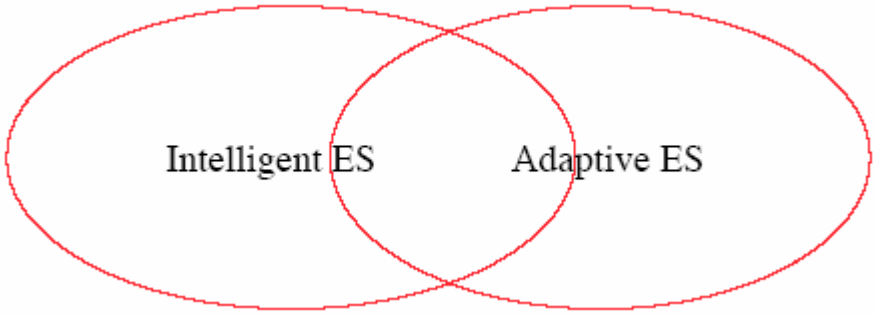


Figura 2. 1-Relazione tra sistemi didattici adattivi e intelligenti.

La ragione per focalizzarsi su entrambi i sistemi adattivi e intelligenti è che l'intersezione è ancora più larga, i confini tra "intelligent" e "non-intelligent" non sono ben definiti, e entrambi i gruppi sono certamente di interesse per la comunità AI nell'ambito dell'educazione (Mitsuhara et al. 2003).

Per tecnologie adattive e intelligenti intendiamo essenzialmente differenti modi di aggiungere funzionalità adattive e intelligenti ai sistemi per la didattica.

Un primo esame (Brusilovsky 1999) identifica cinque tecnologie principali utilizzate in AIWBES (Figura 2.2). Queste tecnologie hanno radici in due campi di ricerca che si sono ben formate prima di Internet: **Intelligent Tutoring Systems (ITS)** e **Adaptive Hypermedia** che verranno brevemente descritti nei successivi paragrafi di questo capitolo.

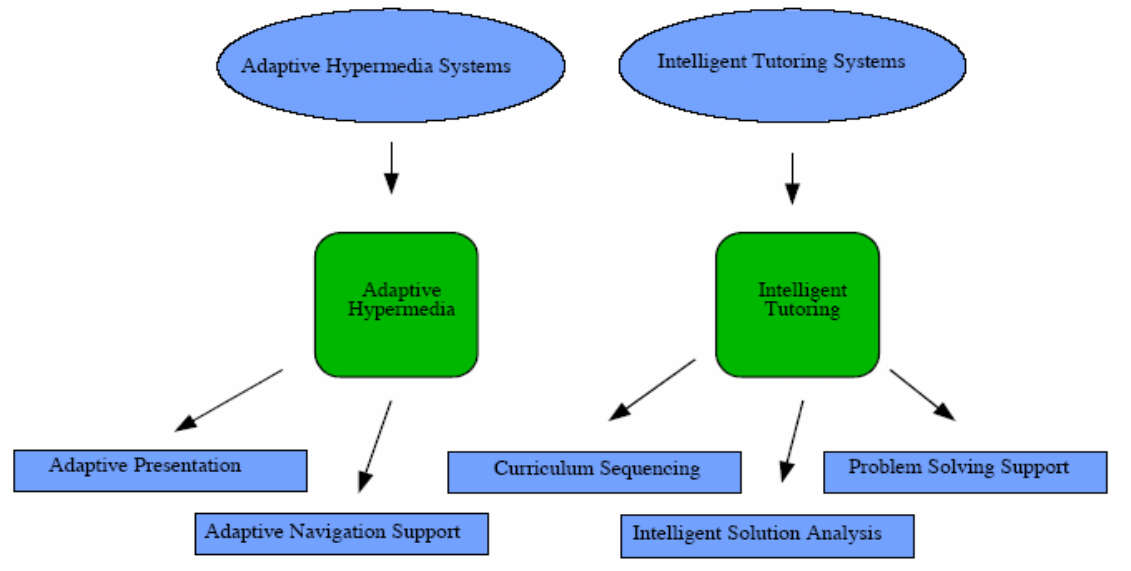


Figura 2. 2-Tecnologie classiche AIWBES e loro origini

Le maggiori tecnologie di *Intelligent Tutoring* sono: ordine del programma, analisi intelligente della soluzione, e supporto alla risoluzione dei problemi. Tutte queste tecnologie possono essere esaminate nel campo dell'ITS. Lo scopo del ***curriculum sequencing technology*** è fornire allo studente la più adatta sequenza di argomenti individualmente progettata per apprendere e compiti (esempi, domande, problemi, etc.) con cui lavorare. Esso aiuta gli studenti trovare un "optimal path" attraverso il materiale di apprendimento. Nel contesto dell'apprendimento basato sul web (WBE), curriculum sequencing technology diviene molto importante per la sua capacità a guidare gli studenti attraverso l'iperspazio di informazioni disponibili. Curriculum sequencing è stato uno dei primi ad essere implementato in maniera veloce nei sistemi AIWBES. I sistemi, ELM-ART (Weber & Brusilovsky 2001) e KBS-Hyperbook (Henze et al.2001) forniscono due buoni esempi di curriculum sequencing. In ELM-ART la sequenza è implementata in forma di collegamenti consigliati e un pulsante "next" adattivo. In KBS-Hyperbook è implementato come un percorso di apprendimento suggerito.

Intelligent solution analysis si occupa delle soluzioni degli studenti ai problemi di didattica (che può variare da semplici questioni a complessi problemi di programmazione). A differenza dei controllori non intelligenti che possono solo chiedere se la soluzione è corretta o no, gli analizzatori intelligenti possono chiedere cosa è sbagliato o incompleto e quale mancato o inesatto pezzo di conoscenza può essere responsabile dell'errore. L'analizzatore intelligente può fornire allo studente un feedback agli errori e aggiornare il modello dello studente

L'obiettivo dell'***interactive problem solving support*** è fornire agli studenti un aiuto intelligente in ogni passo della risoluzione dei problemi, per dare un suggerimento allo studente di come eseguire il prossimo passo. Come è stato dimostrato dai sistemi pionieri, implementazioni lato server pure, come PAT-Online (Ritter 1997), non possono effettivamente tener d'occhio le azioni degli studenti e possono solo fornire l'aiuto richiesto. Implementazioni lato client pure, come ADIS (Warendorf & Tan 1997) hanno una complessità limitata. La giusta funzionalità e livello di complessità per implementare un supporto interattivo alla risoluzione dei problemi, richiede implementazioni client-server come AlgeBrain (Alpert et al. 1999). Tra i sistemi esistenti ActiveMath (Melis et al. 2001) implementa un supporto interattivo alla risoluzione dei problemi, in aggiunta a questo, ELM-ART (Weber & Brusilovsky 2001) fornisce un unico esempio di *example-based problem solving support*, un supporto tecnologico interattivo che promette bene nel contesto del Web.

Supporti di presentazione adattive e navigazioni adattive sono le due maggiori tecnologie esplorate da adaptive hypertexts and hypermedia systems.

L'obiettivo della tecnologia **adaptive presentation** è di adattare i concetti presentati in ciascun nodo hypermediale (pagina) obiettivo degli studenti, conoscenza, e altre informazioni registrate nel modello dello studente. In un sistema con presentazione adattiva le pagine non sono statiche ma adattivamente generate o assemblate da ciascun utente. ActiveMath fornisce uno dei più avanzati esempi esistenti per presentazioni adattive, insieme a ELM-ART e MetaLinks (Murray 2003).

L'obiettivo della tecnologia **adaptive navigation support** è di assistere lo studente nell'orientazione e navigazione dell'iperspazio cambiando l'aspetto dei link visibili. Per esempio, un sistema adattivo ipermediale può ordinare adattivamente, annotare, o in parte nascondere i link della pagina corrente per permettere facilmente la scelta per

andare alla pagina seguente. Il supporto adattivo alla navigazione condivide gli stessi obiettivi del curriculum sequencing, aiutare gli studenti a cercare il percorso ottimale attraverso il materiale didattico. Allo stesso tempo, il supporto adattivo alla navigazione è meno direttivo e più "cooperativo" della sequenza tradizionale: esso guida gli studenti, mentre fanno le loro scelte su quale deve essere il prossimo frammento di conoscenza da apprendere e il prossimo problema da risolvere. Nel contesto del WWW, dove hypermedia è il paradigma base organizzazionale, il supporto alla navigazione adattiva diviene sia naturale che efficiente. Esso è utilizzato nelle prime tecnologie AIWBES ed esaminato in sistemi come ELM-ART, InterBook, KBS-Hyperbook (Henze& Nejdl 2001), MLTutor (Smith & Blandford 2003) e diviene forse la tecnologia più popolare nell'AIWBES.

2.1.1 Evoluzione dei sistemi AI-ED

I sistemi web-based educational adattivi ed intelligenti stanno dando vita ad un nuovo paradigma di sviluppo nel campo dell'intelligenza artificiale applicata all'educazione.
Se analizziamo la varietà dei sistemi web-based educational adattivi ed intelligenti sviluppato dalla nascita del AI-ED nel 1970, possiamo distinguere almeno tre maggiori paradigmi di sviluppo (Tabella 2.1).
La motivazione che sta dietro i primi sistemi di AI-Ed è risolvere gli evidenti problemi che dominavano nel Computer-Assisted Instruction (CAI): fornire una valutazione più intelligente della conoscenza dello studente di tradizionali domande "yes-no" e "multiple-choice" e più sequenze adattive di frammenti di istruzioni rispetto all'approccio tradizionale. Questi sistemi sono chiamati Intelligent CAI (ICAI) o **AI-CAI**. ICAI non tentano di cambiare quello che è ben stabilito nel contesto delle applicazioni dei sistemi CAI e i loro maggiori obiettivi, cioè trasferire nuova conoscenza agli studenti e garantirne l'acquisizione. Sia CAI che ICAI furono voluti per sostituire del tutto o in parte l'istruzione tradizionale in classe. Le maggiori piattaforme informatiche, originariamente dietro CAI, furono classici mainframes e più tardi mini-computers.
Alla fine del 1970 fu determinato un nuovo paradigma di "istruzione" (Brown & Burton 1978), (Clancey 1979). Esso fu più tardi propagato dalla scuola di John Anderson e divenne dominante dal 1980 fino agli inizi del 1990.

I sostenitori del nuovo paradigma rivelano che il lavoro principale dei sistemi AI-Ed non è sostituire l'insegnante in classe presentando nuovo materiale, ma fornire un tutor individuale che può supportare gli studenti nel processo di risoluzione dei problemi didattici e nella formazione della procedura di conoscenza. Quando il vecchio nome "AI-CAI" non fu più attinente, i nuovi sistemi e il loro campo d'interesse adottarono il nome di "**Intelligent Tutoring System**" (ITS).

	AI-CAI Paradigm	ITS Paradigm	AIWBES Paradigm
Time span	1970-ies	1980-1990-ies	End of 1990-ies – 2000-ies
Goal	Replace primitive CAI in transferring knowledge	Support problem solving	Comprehensive support
Context	Classroom without teachers	Classroom with a facilitator or self-study	Impendent self-study
Learning material	All learning material inside the system, most often presentations, but also exercises and problems.	No presentation material inside the system, but problems are often included.	Rich learning material on-line: presentations, examples, problems.
Technologies	Curriculum sequencing and intelligent solution analysis are the core technologies.	No course sequencing or adaptive hypermedia. Interactive problem solving support is the core technology.	Extensive use of adaptive hypermedia. Curriculum sequencing and intelligent solution analysis become widespread again. A range of Web-inspired technologies appears.
System completeness	All systems focus on single intelligent technology.	Most systems focus on single intelligent technology.	Most systems focus on several intelligent technologies.
Platform	Mainframes and mini-computers.	Personal computers	WWW

Tabella 2. 1- Confronto dei maggiori paradigmi AI-Ed

Ciò che si osserva ora, è un nuovo cambiamento del paradigma, spesso spinto da qualche estensione dei cambiamenti della piattaforma e del contesto di applicazione. Il contesto di applicazione, motivato dietro i sistemi didattici basati sul Web (WBE) è, naturalmente, quello della didattica basata sul Web. In questo contesto, con un insegnante non umano, i sistemi didattici hanno fornito una soluzione per tutti i bisogni degli studenti. La vecchia motivazione del CAI di "distribuire conoscenza" torna

centrale e diviene dominante. Il bisogno di supportare la risoluzione dei problemi rimane centrale. I nuovi bisogni specificati dal moderno WBE diventano critici, come il bisogno di supporto al lavoro collaborativo e quello di supporto al lavoro dell'insegnante remoto con la classe invisibile. Questo contesto causa la comparsa di nuove tecnologie oltre che, cambiamenti nell'uso del profilo delle tecnologie del sapere.

2.2 ITS (Intelligent Tutoring System)

Il primo ingresso dell'intelligenza artificiale nel settore della didattica risale al 1970 con un articolo di Carbonell (Carbonell 1970) che individuava alcuni limiti della tradizionale Istruzione Assistita da Calcolatore (CAI) e proponeva un nuovo tipo di CAI cosiddetto intelligente, certamente già orientato in senso cognitivista. In questi anni l'intelligenza artificiale ha promesso molto al mondo della didattica, ma, da un punto di vista pratico, ha mantenuto poco, proprio com'è avvenuto in altri settori di applicazione dell'Intelligenza Artificiale.

Due sono i figli dell'Intelligenza Artificiale d'interesse per la didattica: i sistemi esperti e soprattutto i cosiddetti Intelligent Tutoring Systems (ITS).

E' stato dimostrato che i programmi didattici basati su sistemi esperti non danno risultati particolarmente brillanti (Alpay 1989). Essi, infatti, tendono a adottare la prospettiva dell'esperto e tendono semplicemente a riproporre a chi impara il modo di ragionare dell'esperto. Essi vanno piuttosto considerati come job aids, cioè come strumenti di supporto a chi deve svolgere un determinato compito professionale e come tali possono essere utilizzati in sede didattica. E' chiaro che in una logica costruttivista essi possono facilmente trasformarsi da job aids a learning aids.

Differente è il discorso sugli ITS. Essi si sviluppano su una corrente di pensiero detta cognitivista. Il cognitivismo mette in primo piano i processi interni e i fattori cognitivi che favoriscono il raggiungimento degli obiettivi didattici, volgendo l'attenzione sia alla quantità che alla qualità dell'apprendimento. Un importante aspetto del cognitivismo è il costruttivismo che intende l'apprendimento come un impegno da parte dello studente per costruire la propria conoscenza, piuttosto che come un travaso della conoscenza dalla mente del docente a quella del discente.

SCHOLAR, il primo ITS noto della storia, fu proposto da Jaime Carbonell nel 1970. SCHOLAR è un programma per l'insegnamento della geografia dell'America Latina costituito da:

- una *base di conoscenza*, che contiene i dati e le loro connessioni rappresentati tramite una rete semantica.
- *un'interfaccia* che consente tre modalità d'utilizzo: il programma interroga lo studente, lo studente interroga il programma, oppure l'iniziativa può essere presa da ambedue le parti (es. lo studente o il programma possono rispondere a una domanda con un'altra domanda).

In seguito alla comparsa del primo ITS, nacquero numerosi altri sistemi.

SOPHIE (SOPHisticated Instructional Environment) é un sistema ICAI sviluppato da John Seely Brown, Richard Burton, e da altri colleghi tra cui Bolt Beranek and Newman (Brown, Burton, Bell 1975). E' un tutoriale che insegna ad individuare guasti in circuiti elettronici: il programma contiene un simulatore di circuiti, una base di conoscenza che "sa" come funzionano e un'interfaccia molto raffinata in grado di dialogare con l'aspirante riparatore.

Lo sviluppo del sistema **WEST** cominciò nell'ambito di SOPHIE. Esso fu costruito sul modello del gioco "How the West was Won". E' una specie di gioco da tavola, sullo schermo, con due giocatori che giocano l'uno contro l'altro. Le mosse determinate dal lancio del dado consistono in tre figure che sono combinazione di operazioni aritmetiche. West è la combinazione di un esperto "black box", che dirige le mosse e non simula intelligenza umana, e un esperto "glass box" che si occupa di analizzare le strategie sub-ottimali di gioco.

Uno dei primi sistemi a separare nettamente i "contenuti" dalla "conoscenza" è stato, invece, **ELM-ART** (ELM Adaptive Remote Tutor) (Brusilovsky et al,1996), un ITS di supporto alla programmazione in Lisp e uno dei primi, tra l'altro ad affacciarsi sul Web. ELM-ART è sviluppato sulla base del sistema ELM-PE (Wenger & Möllenberg 1994), un Intelligent Learning Environment che supporta esempi di programmazione, fa un'analisi intelligente per la risoluzione dei problemi e facilita testing e debugging.

ELM-ART può essere considerato come un libro di testo intelligente on-line, che fornisce tutto il materiale del corso (presentazione di nuovi concetti, test, esempi, e problemi) sotto forma di hypertesti.

Un più moderno sistema che coniuga la rappresentazione esplicita della conoscenza attraverso grafi concettuali con la descrizione esplicita dei learning object attraverso metadati è stato **ABITS** (Capuano, Marsella, Salerno 2000) realizzato nell'ambito del progetto EC InTraSys.

2.2.1 I componenti di un ITS.

Un Intelligent Tutoring System (ITS) è un sistema software per il supporto all'insegnamento, progettato sulla base di tecniche di Intelligenza Artificiale allo scopo di rappresentare la conoscenza e portare avanti un'interazione con uno studente. Perché questo accada, un sistema ITS deve comprendere un dominio di conoscenza (*domain model*), un modello dello studente (*student model*) e un modello pedagogico (*tutor model*) e un componente per la comunicazione del programma con lo studente (*interface*) (Woolf 1992). Si può individuare anche un quinto componente, il modello esperto (*expert model*) che qualcuno considera come facente parte del dominio della conoscenza (Figura 2.3).

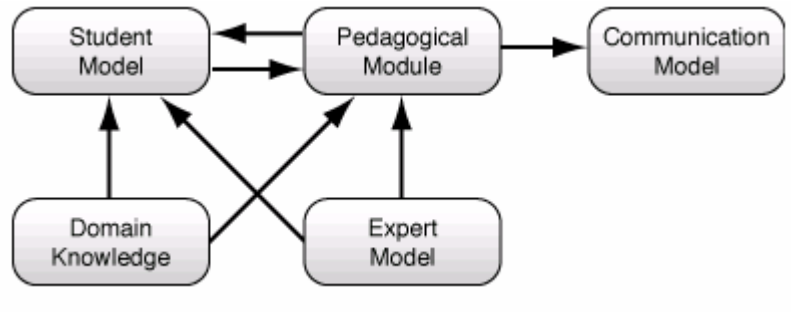


Figura 2. 3- Integrazione dei componenti in un Intelligent Tutoring System

Dominio di conoscenza

Il dominio di conoscenza definisce la conoscenza di un esperto in una certa area. E' per tale motivo spesso chiamato modello esperto. Esso consiste in una conoscenza procedurale e dichiarativa (Bobrow 1977) ed è organizzato sottoforma di liste, diagrammi strutturati di conoscenza o regole.

Contiene la rappresentazione esplicita della conoscenza da fornire allo studente, espressa sotto forma di un modello (di conoscenza), in grado, sia di comunicare all'esterno i concetti e le proprietà del dominio d'applicazione, sia di dotare il sistema di capacità dinamiche per l'elaborazione della conoscenza esperta. Il modello consente

perciò all'esperto di dominio di generare automaticamente le soluzioni ai problemi formulati allo studente nel corso del processo d'insegnamento, di spiegare i passi del ragionamento seguito e di confrontare i processi d'inferenza dello studente stesso con quelli corretti del sistema. Ne segue che è potenzialmente possibile valutare sia il livello d'apprendimento sia i progressi nello studio del dominio in esame da parte dello studente.

La *conoscenza dichiarativa* definisce i termini del dominio della conoscenza e i loro attributi, e definisce la relazione tra i termini, spesso nella forma di frames con l'eredità.

La *conoscenza procedurale* consiste in argomenti o regole che servono d'aiuto nella risoluzione di problemi. A volte la conoscenza euristica si distingue dalla dichiarativa e dalla procedurale. La *conoscenza euristica* consiste nell'esperienza e nella conoscenza della risoluzione di problemi d'esperti che non è ristretta a particolari contenuti. La conoscenza euristica è necessaria così che il tutor è capace di guidare lo studente nei processi di conoscenza e nella risoluzione dei problemi.

Ci sono essenzialmente due modelli per progettare il dominio della conoscenza: il "*black box model*" e il "*glass box model*". Con il modello scatola nera non c'è richiesta di riproduzione dell'intelligenza umana. Il programma SOPHIE I è considerato un esempio di modello scatola nera. Nel modello scatola di vetro, il dominio della conoscenza è modellato nella forma di un sistema esperto e riproduce il comportamento di un esperto umano nella risoluzione dei problemi. Il modello della scatola di vetro è chiamato con questo nome perché è trasparente, in pratica si può osservare il suo metodo lavorativo.

Modello dello studente

Il modello dello studente è quel componente dell'ITS che regista informazioni sullo stato cognitivo dello studente (Zoran & Vladav 2004).

Il paradigma standard per rappresentare il modello dello studente è l'**overlay model** (Carr et.al 1997) nel quale la conoscenza dello studente è considerata come un sotto insieme della conoscenza dell'esperto (Figura 2.4a). Con questa rappresentazione, un ITS presenta il materiale allo studente di modo che la conoscenza che egli potrà acquisire sia uguale a quella dell'esperto.

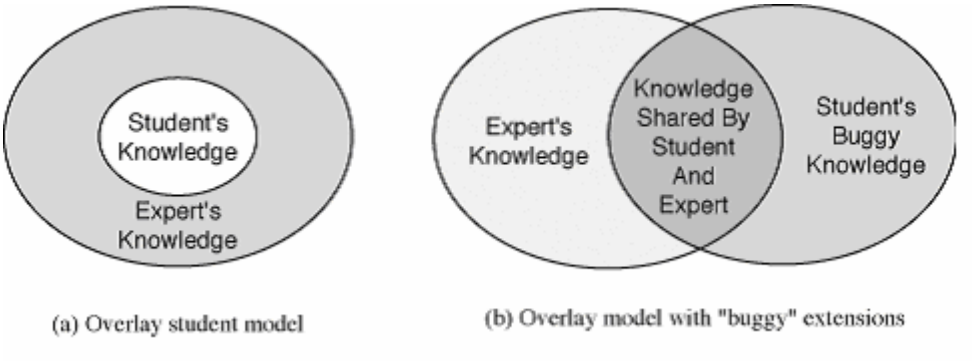


Figura 2. 4-Overly student models

L'overlay model ha una particolare debolezza: non riconoscere che lo studente può possedere delle conoscenze che non fanno parte della base di conoscenza dell'esperto. Esiste invece un'estensione dell'overlay model che rappresenta esplicitamente la conoscenza "buggy" che lo studente potrebbe avere (Figura 2.4b).

Modello pedagogico

Il modello pedagogico usa le informazioni contenute nel modello dello studente per determinare quali aspetti del dominio della conoscenza devono essere presentati all'utente. Questa informazione, per esempio, può essere nuovo materiale, una rivisita di precedenti argomenti, o feedback ad argomenti correnti. Un fatto didattico per un ITS è la selezione di una meta-strategia per il dominio d'insegnamento. Il tutor deve decidere il contenuto del materiale da presentare allo studente. Questo coinvolge decisioni sugli argomenti, i problemi e i feedback.

Selezione di argomenti: per selezionare un argomento da presentare, il tutor deve esaminare il modello dello studente per determinare gli argomenti sui quali lo studente ha bisogno di focalizzarsi. Molte possibilità esistono per i più appropriati argomenti sui quali uno studente potrebbe lavorare. Per esempio, se la meta-strategia indica che il riesame è in ordine, il tutor può selezionare un argomento che lo studente ha già appreso. D'altro canto, se la nuova informazione è stata presentata, il tutor potrà scegliere un argomento che lo studente non ancora conosce.

Generazione di problemi: una volta che un argomento è stato selezionato, deve essere generato un problema che lo studente deve risolvere. La dimensione del problema è determinata dal dominio. Qualunque granularità del problema sia generata, è importante

che le difficoltà siano appropriate al livello d'abilità dello studente, che può essere determinato dal modello dello studente.

Feedback: molti tutor lavorano tranquillamente fintanto che gli studenti rispondono a tutto in maniera giusta. Il problema sorge, quando lo studente ha difficoltà e ha bisogno dell'aiuto del tutor. In questa situazione, il tutor deve determinare il tipo di feedback da fornire. Il modo in cui deve essere fornito questo aiuto allo studente spesso è una questione molto complicata

Una volta che il sistema decide quale feedback dare, esso deve determinare quali contenuti consigliare. Il feedback dovrebbe contenere sufficiente informazione così che gli studenti possono procedere al prossimo passo nella risoluzione del problema. Alcuni sistemi usano il modello dello studente per selezionare un suggerimento che più si avvicina al livello d'abilità dell'utente. D'altra parte, ad uno studente con alte competenze in una materia potrebbe aspirare a suggerimenti più chiari. Usando questa tecnica, allo studente non sarà richiesto di passare attraverso molti livelli di suggerimenti e riceverà un soddisfacente aiuto.

Modulo di comunicazione

Controlla l'interazione tra il sistema e l'utente. Si interessa di stabilire come il meteriale deve essere presentato allo studente.

Modello esperto

Il modello esperto è simile al dominio della conoscenza e deve contenere informazioni su cosa insegnare allo studente. Tuttavia, esso è più di una rappresentazione dei dati; è un modello di come un esperto di un particolare dominio rappresenta la conoscenza. Più comunemente, esso prende la forma di un modello esperto eseguibile, cioè uno che è capace di risolvere problemi nel dominio. Usando un modello esperto, un tutor può confrontare le soluzioni dell'alunno con quelle dell'esperto, mettendo in evidenza i punti in cui l'alunno ha difficoltà.

2.3 Sistemi Adattivi Ipermediali

Adaptive hypermedia (AH) è un'alternativa al tradizionale approccio "one-size-fits-all" nello sviluppo di sistemi ipermediali. *Adaptive hypermedia systems (AHS)* costruisce un

modello di obiettivi, preferenze e conoscenza per ogni utente individuale, e usa questo modello durante l'iterazione con l'utente (Brusilovsky 1996). Per esempio, a uno studente in un sistema didattico adattivo e ipermediale, sarà data una presentazione che è specificatamente adatta a lui e gli sarà suggerito un set dei più rilevanti links per andare avanti.

Un sistema AH può essere usato in molte aree applicative, dove utenti di sistemi ipermediali hanno essenzialmente differenti obiettivi e conoscenze e dove l'iperspazio è ragionevolmente largo. Utenti con differenti obiettivi e conoscenze possono essere interessati a diversi pezzi d'informazioni presentate su una pagina ipermediale e possono usare diversi links per navigare. AH cerca di superare questo problema usando la conoscenza rappresentata nel modello dello studente per adattare le informazioni e links presentati per ogni utente Conoscendo gli obiettivi dell'utente e la conoscenza, il sistema AH può supportare gli utenti nella loro navigazione limitando lo spazio di ricerca, suggerendo molti links rilevanti da seguire, o fornire commenti adattivi per i links visibili.

I primi pionieri di sistemi didattici adattivi ipermediali furono sviluppati tra il 1990 e 1996. Questi sistemi possono essere divisi in due flussi di ricerca, uno seguito dai ricercatori nell'area dell'Intelligent Tutoring System (ITS) l'altro da ricerche di lavoro sull'ipermedia didattici nel tentativo di costruire dei sistemi adatti agli studenti individuali.

Il segreto dell'adattività in tutti i sistemi adattivi ipermediali è "conoscenza dietro le pagine". Tutti i sistemi didattici adattivi e ipermediali modellano esplicitamente la conoscenza del dominio attraverso uno spazio di conoscenza. Per permettere ai sistemi adattivi di conoscere cosa è presentato in una particolare pagina o frammento di pagina, gli spazi di conoscenza sono collegati all'iperspazio di materiale didattico.

2.3.1 Un modello per la strutturazione dell'informazione.

La struttura dell'informazione di un tipico sistema adattivo ipermediale può essere considerata come l'interconnessione di due reti o "spazi" (Figura 2.5), una rete di concetti (spazio di conoscenza) e una rete di pagine ipertestuali con materiale didattico (iperspazio tradizionale). Di conseguenza, la progettazione di un sistema adattivo ipermediale coinvolge tre sotto passi: strutturare la conoscenza, strutturare l'iperspazio, e collegare lo spazio di conoscenza con l'iperspazio.

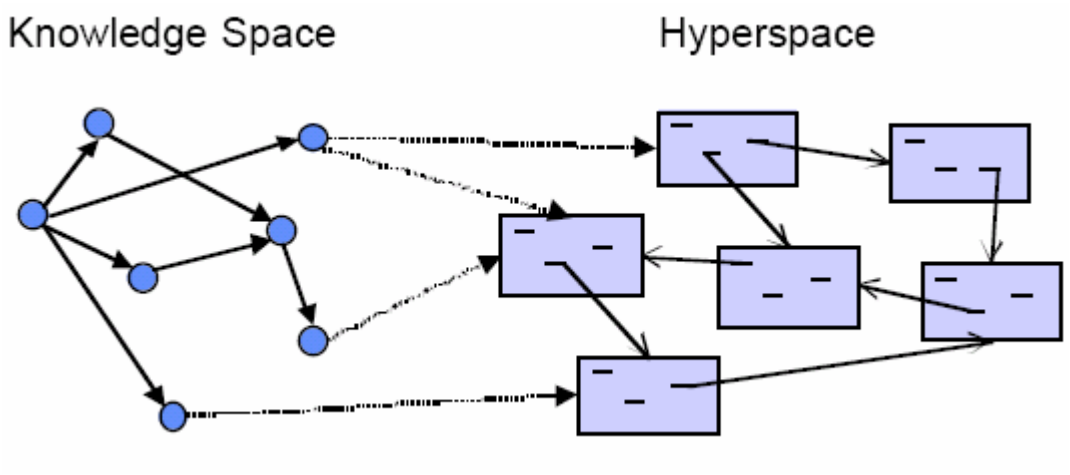


Figura 2. 5- Una tipica struttura dello spazio di informazione in un sistema ipermediale

2.3.1.1 Strutturazione della conoscenza di dominio

Per la strutturazione della conoscenza fondamentale è il concetto di ***modello del dominio*** che costituisce il cuore dell'approccio basato sulla conoscenza per lo sviluppo di sistemi adattivi ipermediali. Il modello del dominio è composto da un insieme di piccoli elementi del dominio di conoscenza, detti DKE (Domain Knowledge Element). Ciascun DKE rappresenta un frammento elementare di conoscenza per il dominio dato. I concetti DKE possono essere chiamati differentemente secondo il sistema: concetti, articoli di conoscenza, argomento, elementi di conoscenza, oggetti di apprendimento, risultati di apprendimento, ma in ogni caso essi denotano frammenti elementari del dominio di conoscenza. Un insieme di concetti del dominio rappresenta un modello del dominio. Più esattamente, un insieme di concetti indipendenti è la forma più semplice di modello del dominio. In una forma più avanzata di modello del dominio, i concetti sono collegati l'uno al altro così da formare un tipo di rete semantica. Questa rete rappresenta la struttura del dominio coperto da un sistema ipermediale. Questo tipo di modello è chiamato ***network model*** (Figura 2.6).

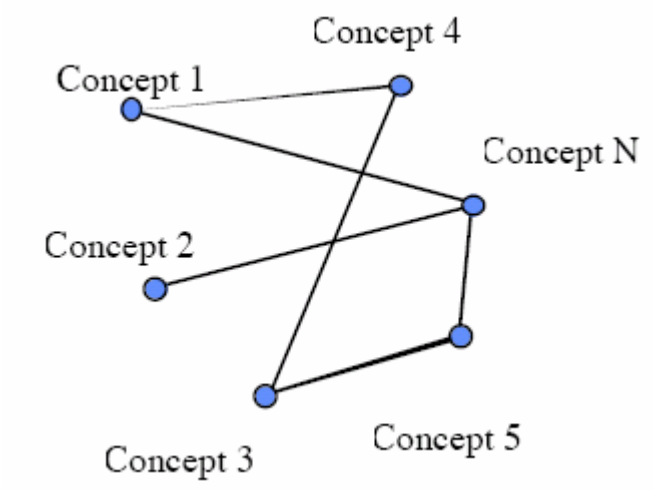


Figura 2. 6-Network domain model

La struttura del modello del dominio è stata ereditata, dai sistemi didattici adattivi ipermediali dal campo degli ITS. La ragione di ciò è che molti dei primi sistemi didattici adattivi ipermadiali hanno avuto una forte connessione con i sistemi ITS. Infatti, un numero di essi fu sviluppato nel tentativo di estendere un sistema ITS con funzionalità ipertestuali.

Il modello del dominio in un sistema didattico adattivo e ipermediale differisce molto in complessità. Qualche AHS sviluppato per insegnamento in corsi universitari sviluppa semplici vettori del modello del dominio. Allo stesso tempo, molti moderni AHS usano un modello di rete con diversi tipi di collegamenti che rappresentano differenti tipi di relazioni tra concetti. Il collegamento più popolare in AHS didattici è il collegamento tra concetti che indicano che uno dei concetti connessi è stato appreso ha altri (Grigoriadou et al. 2001). In molti AHS i link essenziali sono solo quelli tra concetti (Weber et al. 2001b) Altri tipi di link, che sono popolari in molti sistemi, sono i classici link semantico “is-a” e “part-of” (De Bra & Ruiter 2001).

Altre differenze riguardano la struttura interna dei concetti. Per la maggior parte dei sistemi educativi AHS, i concetti del dominio non sono che nomi che denotano frammenti del dominio della conoscenza. Allo stesso tempo, alcuni sistemi AH rappresentano una struttura interna per ogni concetto come un insieme di attributi.

Modello dello studente

Una delle più importanti funzioni del modello dello studente è di rappresentare il dominio della sua conoscenza. La maggior parte dei sistemi AHS usano l’*overly model*

per la conoscenza dello studente (conosciuto anche come overlay student model). L'overlay model è stato ereditato dal campo degli ITS. Il principio chiave di tale modello è che per ogni concetto del modello del dominio, il modello di conoscenza dello studente individuale memorizza alcuni dati che sono una stima del livello di conoscenza che lo studente ha di questi concetti. La maniera più semplice per fare questo è utilizzare un valore binario (conosce-non conosce) che permette al modello di rappresentare la conoscenza dello studente come una sovrapposizione del dominio della conoscenza. Mentre alcuni AHS di successo utilizzano questa classica forma di overlay model, la maggior parte dei sistemi usano un *weighted overlay model* che può distinguere vari livelli di conoscenza dei concetti da parte dell'utente usando valori qualitativi (per esempio, alto-medio-basso), un valore qualitativo intero (per esempio da 0 a 100) come mostrato in Figura 2.7 o una probabilità che lo studente conosca i concetti (Specht & Klemke 2001).

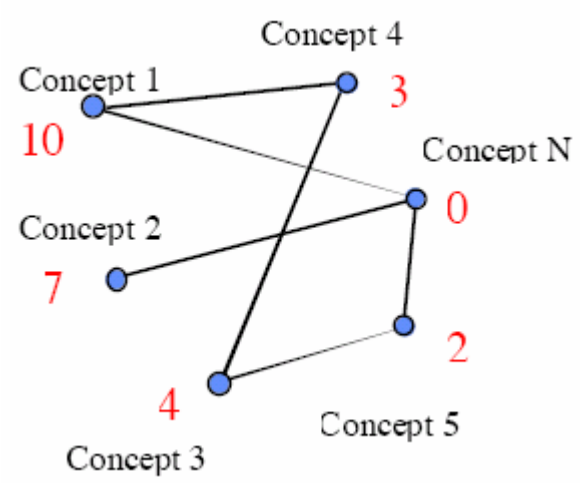


Figura 2. 7-Un esempio di overlay student model

Nel weighted overlay model la conoscenza dello studente può essere rappresentata come una coppia "concetto-valore" per ogni concetto del dominio.

In aggiunta al livello di concetti dell'overlay model, alcuni AHS mantengono un modello del livello di pagina anche conosciuto come *modello storico*. Questo modello mantiene informazioni sulle visite dell'utente a pagine individuali come il numero di visite o il tempo speso su una pagina. Mentre in alcuni AHS il modello storico era l'unico modello dello studente usato, i moderni AHS tendono ad ignorarlo o ad usarlo solo come fonte secondaria d'informazione. Tuttavia ancora oggi esistono sistemi come

ALE, (Spechi et al. 2002), che anche se recenti utilizzano come base di adattamento del modello dello studente solamente un modello storico.

Modellare un percorso didattico

L'obiettivo che si vuole raggiungere è quello di costruire un percorso didattico individuale per ogni studente del sistema. Questo approccio è anch'esso ereditato dai sistemi ITS, con esso un obiettivo didattico elementare può essere rappresentato come un sott'insieme del modello del dominio che deve essere appreso. Questo obiettivo può essere assegnato ad uno studente o dall'autore del corso o dall'insegnante, o selezionato direttamente dallo studente.

Un modo naturale per lo studente di selezionare un obiettivo didattico è di scegliere un progetto, esercizi, o altre attività didattiche come obiettivi a lungo termine. I più avanzati obiettivi didattici elementari possono essere rappresentati come un insieme di due valori: un concetto e il suo livello di conoscenza. Gli obiettivi elementari possono essere organizzati in obiettivi più complessi strutturati in sequenze (gli obiettivi vengono archiviati uno dopo l'altro) o ad albero (gli obiettivi vengono archiviati a partire dalla radice). Per esempio, in InterBook un obiettivo d'apprendimento individuale è originariamente modellato come una sequenza di insiemi (Figura 2.8) e dopo come uno stack di insiemi per permettere allo studente di prelevare da solo gli obiettivi didattici in cima allo stack (Schwarz, 1998).

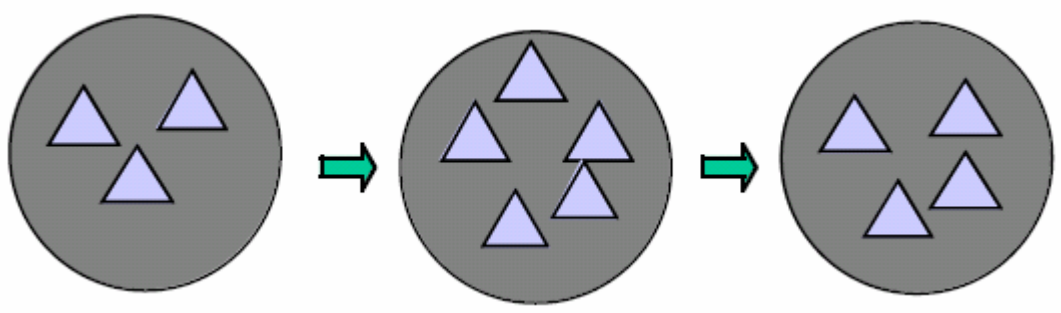


Figura 2. 8-Un obiettivo didattico modellato come una sequenza di sottoinsiemi di concetti del modello del dominio (rappresentati dai triangoli)

2.3.1.2 Connessione tra la conoscenza e il materiale didattico

Il processo per connettere la conoscenza con il materiale didattico presuppone la conoscenza di come indicizzare le pagine dei contenuti. Ci sono quattro aspetti che sono

importanti per distinguere i differenti approcci all'indicizzazione: cardinalità, granularità, navigazione e potere espressivo.

Per quel che riguarda la **cardinalità** ci sono due casi differenti: singoli concetti indicizzati, quando ogni frammento di materiale didattico è relativo ad uno ed un solo concetto del dominio e concetti multipli indicizzati, quando ogni frammento può essere relativo a più concetti. L'indicizzazione di concetti multipli è più potente, ma rende il sistema più complesso e richiede maggiore preparazione.

Per **potere espressivo** si intende l'ammontare di informazione che gli autori possono associare ai link tra concetti e pagine. Naturalmente, la più importante informazione è la vera presenza dei link. Questo caso potrebbe essere chiamato come indicizzazione piatta ed è usata nella maggior parte dei sistemi esistenti. Alcuni sistemi associano più informazioni con i link usando ruoli e/o pesi. Assegnando un ruolo ad un link questo è di aiuto per distinguere vari tipi di connessione tra concetti e pagine (Brusilovsky 2000). Un'altra strada per aumentare il potere espressivo dell'indicizzazione è specificare il peso di un link tra un concetto e una pagina. Il peso può specificare, per esempio, la percentuale di conoscenza di un concetto rappresentato nella pagina.

Il concetto di **granularità** riguarda la precisione dell'indicizzazione. I due casi più popolari sono l'indicizzazione di tutte le pagine ipertestuali con concetti e l'indicizzazione di frammenti di pagine con concetti. Ci sono anche casi in cui tutti i gruppi di pagine connesse sono indicizzate con lo stesso insieme di concetti.

Per quanto riguarda l'aspetto della **navigazione** da ogni pagina con materiale didattico lo studente può navigare tutti i concetti connessi con essa e per ogni concetto tutte le pagine indicizzate con questi concetti.

I sistemi AH suggeriscono varie strade per l'indicizzazione che differiscono per gli aspetti visti. Tutte queste differenze possono essere descritte in termini di tre approcci base che sono descritti in seguito.

Concept-Based Hyperspace è l'approccio più semplice per organizzare la connessione tra spazio di conoscenza e iperspazio è conoscere i concetti base dell'iperspazio. Questo approccio è naturalmente presente in tutti i sistemi AH che utilizzano l'indicizzazione di concetti singoli. E' possibile distinguere tra semplici concetti base dell'iperspazio (s*imple concept-based hyperspace*) e ricchi concetti base dell'iperspazio (*enhanced concept-based hyperspace*).

Simple conecpt-based hyperspace è usato in sistemi che hanno esattamente una pagina di materiale didattico per ogni concetto. Con questo approccio, l'iperspazio è costruito come una copia esatta del modello del dominio (Figura 2.9). Ogni concetto del modello del dominio è rappresentato da esattamente un nodo dell'iperspazio, mentre i link tra concetti costituiscono il percorso principale tra i nodi dell'iperspazio.

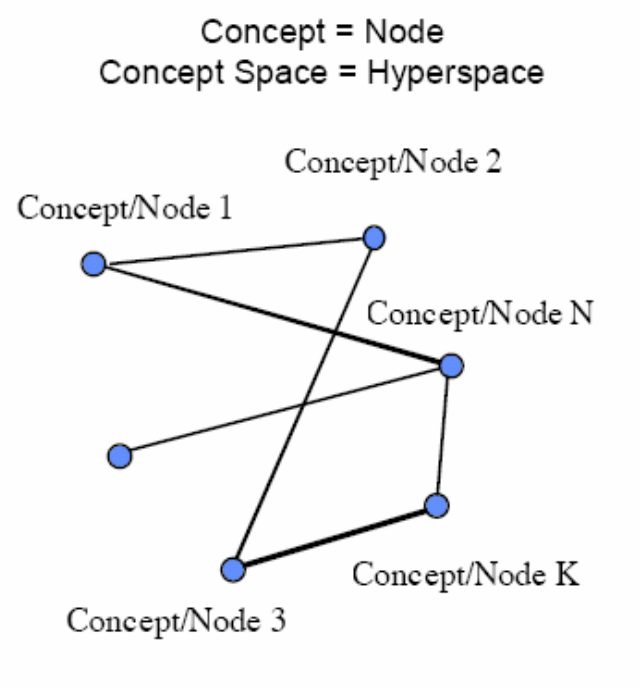


Figura 2. 9-Simple concept-based hyperspace

Gli AHS didattici con ricchi contenuti e indicizzazione di concetti singoli possono usare un approccio *enhanced concept-based hyperspace*. Con questo approccio, pagine multiple che descrivono lo stesso concetto, sono connesse con questo concetto nello spazio di informazione e nell'iperspazio. Ogni concetto ha una corrispondente pagina centrale nell'iperspazio. I concetti della pagina centrale sono connessi con link a tutte le pagine ipertestuali didattiche relative a questi concetti. Lo studente può navigare tra i concetti fulcro della pagina lungo collegamenti concettuali e verso altre pagine contenenti materiale didattico. Questo approccio potrebbe essere usato per creare AHS relativamente grandi con strutture abbastanza semplici e tener conto di un certo numero di tecniche di adattamento.

Indicizzazione delle pagine. L'approccio più diffuso per indicizzare i concetti multipli è indicizzare le pagine. Con questo approccio, tutta la pagina ipermediale (nodo) è

indicizzata con concetti del dominio. In altre parole, i link sono creati tra pagine e ogni concetto che è relativo al contenuto della pagina (Figura 2.10). Il più semplice approccio all'indicizzazione è l'indicizzazione piatta dei contenuti base quando, un concetto è incluso in una pagina indice quando qualche parte di questa pagina presenta il frammento di conoscenza designato dal concetto.

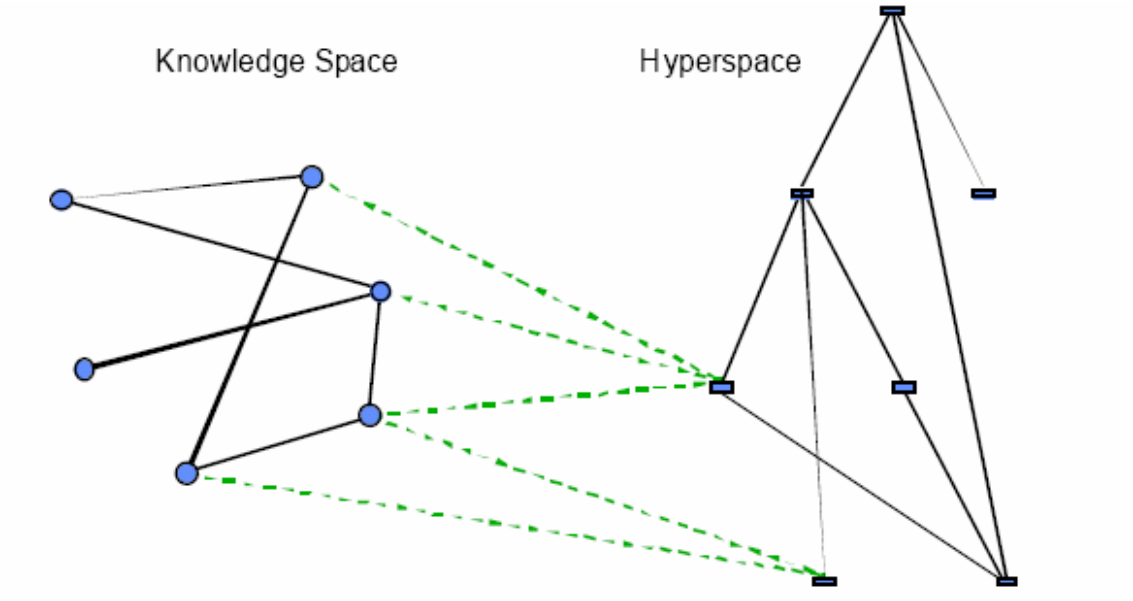


Figura 2. 10- indicizzazione di pagine con concetti multipli

Una strada più generale, ma usata meno spesso per indicizzare le pagine è di aggiungere il ruolo per ogni concetto nella pagina indice (indicizzazione basata sul ruolo). Il ruolo più popolare è "prerequisito": un concetto è incluso in una pagina indice se uno studente ha bisogno di conoscere questo concetto per capire il contenuto della pagina. Altri ruoli possono essere usati per specificare il tipo di contributo che la pagina fornisce per far apprendere il concetto (introduzione, presentazione principale, esempi, etc.). I pesi possono essere usati per indicizzare pagine che contengono concetti multipli per mostrare in che misura le pagine contribuiscono ad insegnare il concetto.
L'indicizzazione delle pagine è un approccio relativamente semplice. Esso può essere applicato persino con modelli del dominio vettoriali con nessun legame tra i concetti (Laroussi & Benahmed). Allo stesso tempo, l'indicizzazione è un meccanismo molto potente, perché fornisce al sistema conoscenza sui contenuti delle sue pagine.

Indicizzazione dei frammenti. L'indicizzazione dei frammenti non è un approccio ancora molto popolare, ma è il più preciso. L'idea di tale approccio è di dividere i

contenuti delle pagine ipermediali in un insieme di frammenti e di indicizzarne alcuni (o addirittura tutti) con concetti del modello del dominio che sono relativi ai contenuti di questi frammenti (Figura 2.11).

Similmente all'approccio d'indicizzazione delle pagine esso può essere usato con un modello del dominio vettoriale non strutturato. La differenza è che l'indicizzazione è fatta su un livello più fine e granulare. Generalmente, è usata l'indicizzazione di concetti multipli, sebbene con frammenti piccoli è spesso possibile usare esattamente un concetto per indicizzare un frammento. In ogni caso l'approccio di indicizzazione dei frammenti da al sistema una conoscenza più fine e granulare sui contenuti delle pagine: il sistema conosce ciò che è presentato in ogni frammento indicizzato. Questa conoscenza può essere effettivamente usata per proporre una presentazione adattiva allo studente. Naturalmente questo approccio ha un prezzo, la maggior precisione viene pagata con il maggior tempo speso per l'indicizzazione.

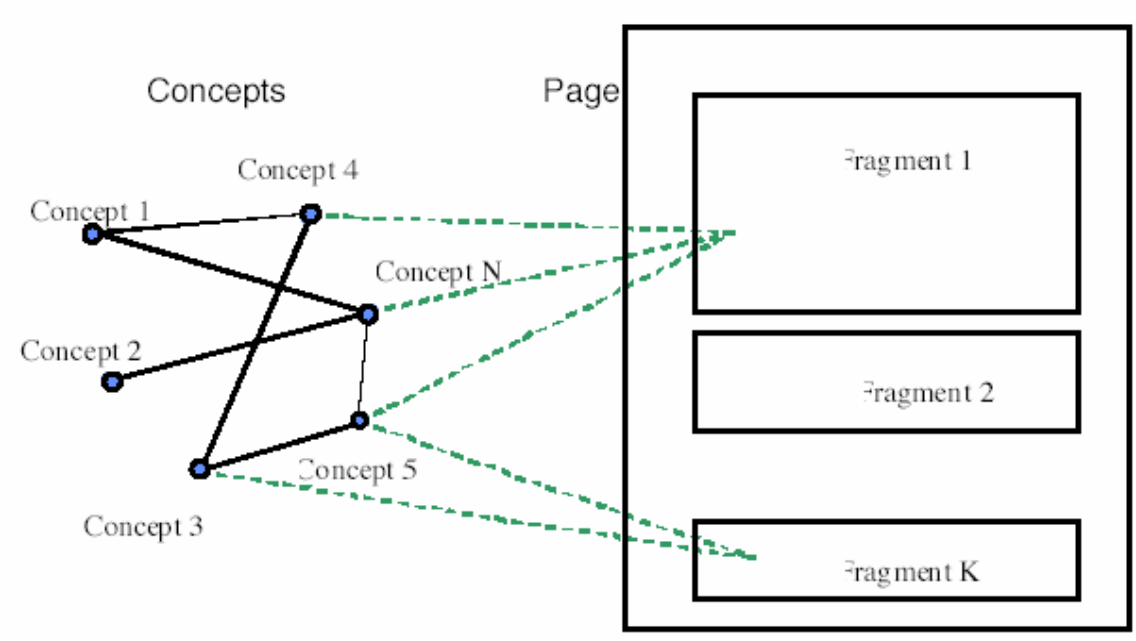


Figura 2. 11- indicizzazione dei frammenti

Un primo sistema che implementa tale approccio è MetaDoc (Boyle & Encarnacion 1994), esso non solo indicizza alcuni frammenti testuali con i relativi concetti, ma distingue anche tre tipi di frammenti: testo generale, ulteriori spiegazioni, e basso livello di dettagli. Il sistema decide se presentare frammenti testuali all'utente o nasconderglieli

e ciò dipende dal livello di conoscenza che ha dei concetti indicizzati. Un utente con buona conoscenza di un particolare concetto non vuole ulteriori spiegazioni su tale concetto e vuole invece tutti i più bassi livelli di dettaglio. Al contrario, un utente con poca conoscenza su un concetto vuole avere ulteriori spiegazioni e non bassi livelli di dettaglio. L'utente con livello medio di conoscenza vuole entrambi i tipi di frammenti testuali.
Altri sistemi che supportano l'indicizzazione dei frammenti sono AHA (De Bra & Calvi 1998), WHURLE (Brailsford et al. 2002), e ALE (Spechi et al. 2002).

Approccio misto. I tre approcci presentati sono basilari per l'organizzazione dell'iperspazio nei sistemi AH. Essi però sono molto complicati perché si basano sullo stesso dominio e modelli utente. Usando più di un approccio si apre la strada all'uso di tecniche di adattamento perché ogni approccio supporta il proprio insieme di tecniche. Per esempio, ISIS-Tutor usa un approccio basato sulla conoscenza per costruire una parte dell'iperspazio rappresentando i concetti di un linguaggio di programmazione. Un'altra parte dell'iperspazio (dove le pagine sono problemi ed esempi) è organizzata con pagine indicizzate con concetti. InterBook è basato sulla stessa idea di ISIS-Tutor usando una combinazione di base di conoscenza e indicizzazione delle pagine.

2.3.1.3 Strutturazione dell'iperspazio

Un importante passo nel progetto di un AHS didattico è strutturare l'iperspazio. L'iperspazio è formato da pagine di contenuti connesse con link di navigazione. La topologia delle connessioni tra pagine è generalmente riferita come struttura dell'iperspazio. Nei moderni AHS didattici la strutturazione dell'iperspazio è un passo del progetto separato, indipendente (nel senso di scelta di approccio di progetto) dal passo di strutturazione della conoscenza. Vediamo vari approcci seguiti per strutturare l'iperspazio di conoscenza.
Il primo approccio è *unstructured approach*, è il caso in cui questo passo di progetto è omesso. Questo era l'approccio predominante nei primi AHS didattici ed è ancora probabilmente il più popolare. Questo per due ragioni. Primo, l'approccio dell'iperspazio basato sui concetti che è usato in molti AHS produce iperspazi particolarmente ricchi e ben strutturati che rendono non necessaria la richiesta di altre organizzazioni. Secondo, spesso nei sistemi che non usano un approccio basato sui

concetti l'iperspazio è strutturato in molti modi, allora le pagine sono spesso connesse con altre pagine con collegamenti classici tra ipertesti (Figura 2.12). Ci sono AHS didattici che non usano né l'approccio basato sui concetti né la struttura esplicita dell'iperspazio e lavorano con una struttura arbitraria dell'ipertesto. Comunque, se non è usata una speciale strategia nel progettare l'iperspazio, esso risulterà una struttura caotica che crea problemi di navigazione e orientamento per gli utenti.

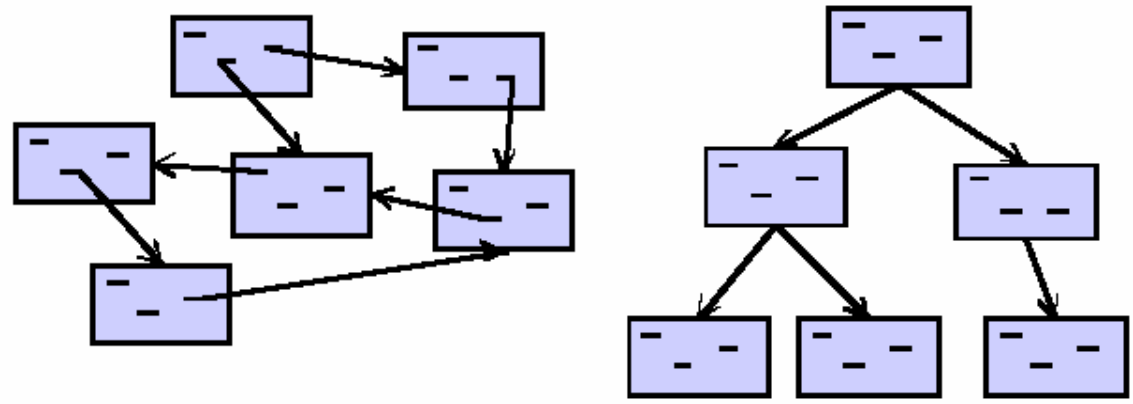


Figura 2. 12- Unstructured hyperspace (left) and hierarchically structured hyperspace (right)

L'approccio più popolare per strutturare l'iperspazio è quello gerarchico. ELM-ART è stato il primo sistema adattivo ipermediale che ha strutturato gerarchicamente l'iperspazio. Quest'approccio è stato rifinito in InterBook e più tardi usato in molti AHS didattici. Con l'approccio gerarchico, l'iperspazio assomiglia ad una struttura gerarchica di un libro di testo – con capitoli, sessioni, e sottosessioni (Figura 2.13). La struttura può essere percorsa seguendo due approcci "depth first" o "breadth first". Uno dei benefici conosciuti nel strutturare gerarchicamente gli ipertesti è la miglior navigazione e orientamento, ciò è particolarmente importante per quelle categorie di utenti che hanno di questi problemi (Lin, 2003).

Un altro efficiente approccio alla strutturazione dell'iperspazio è "ASK". L'obiettivo di tale approccio è di costruire un iperspazio "conversazionale", con esso un progettista prova ad anticipare un numero successivo di domante che l'utente può porre dopo aver visitato una pagina e fornire i link per pagine dell'iperspazio che possono dare le risposte a tali domande.

2.3.2 Moderne tecnologie AIWBES

Una rivisita sulle tecnologie AIWBES (Brusilovsky 1999) ne identifica cinque gruppi (Figura 2.13). Le tecnologie classiche di Adaptive Hypermedia e Intelligent Tutoring rimangono inalterate, mentre si introducono nuove tecnologie ispirate al Web: Adaptive Information Filtering , Intelligent Class Monitoring, e Intelligent Collaboration Support. In Tabella 2.2 sono riportate tali tecnologie con esempi di sistemi che le utilizzano.

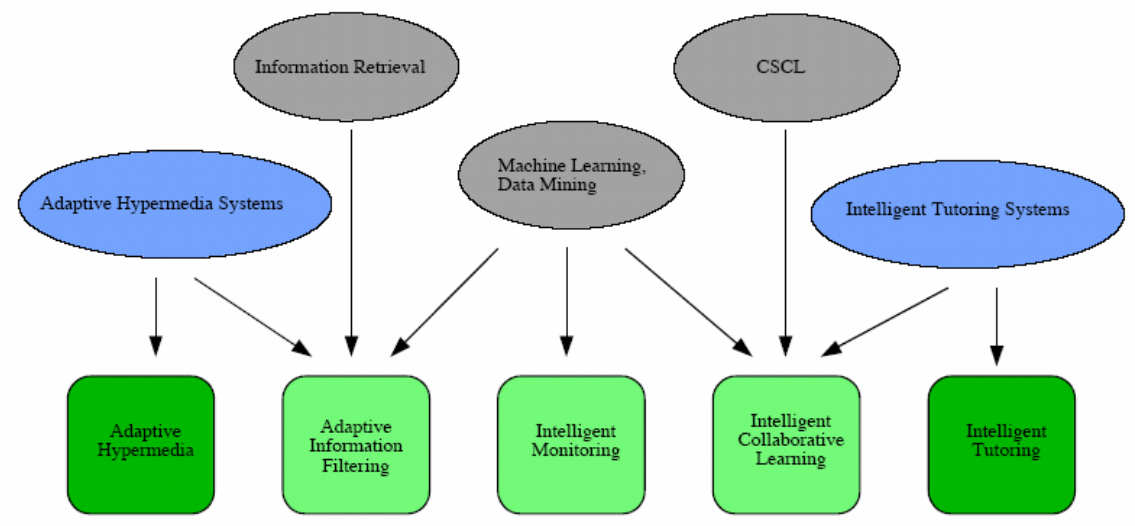


Figura 2. 13-Cinque gruppi di moderne tecnologie AIWBES

Adaptive Information Filtering (AIF) è una classica tecnologia nel campo dell'**Information Retrieval**. Il suo obiettivo è cercare il materiale che è rilevante per l'interesse dell'utente in un largo pool di documenti. Nel Web questa tecnologia è stata usata sia nel contesto della ricerca che del browsing. Essa è stata applicata per adattare i risultati della ricerca sul Web, usando filtraggio e ordinamento e segnalare i documenti più interessanti tra quelli a disposizione attraverso la generazione di link. Mentre i motori usati dai sistemi AIF sono molto differenti dai motori adattivi hypermediali, a livello d'interfaccia basata sul Web AIF molto spesso usa tecniche adattive di supporto alla navigazione. Ci sono essenzialmente due tipi di motori AIF che possono essere considerati come due tecnologie AIF differenti, filtraggio basato sul contesto e filtraggio d'insieme. Il primo si basa sul contenuto dei documenti mentre il secondo ignora completamente il contenuto, tentando piuttosto di accoppiare gli utenti che sono interessati agli stessi documenti.

Sources of AIWBES technologies	Technologies	Sample systems
Adaptive Hypermedia	Adaptive navigation support Adaptive presentation	AHA (De Bra, et al., 1998) InterBook (Brusilovsky, Eklund, & Schwarz, 1998) **KBS-Hyperbook (Henze, & Nejdl, 2001)** **MetaLinks (Murray, 2003)** **ActiveMath (Melis, et al., 2001)** **ELM-ART (Weber, & Brusilovsky, 2001)** INSPIRE (Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, Submitted)
Adaptive Information Filtering	Content-based filtering Collaborative filtering	**MLTutor (Smith, & Blandford, 2003)** WebCOBALT (Mitsuhara, et al., 2002)
Intelligent Class Monitoring		HyperClassroom (Oda, Satoh, & Watanabe, 1998)
Intelligent Collaborative Learning	Adaptive group formation and peer help Adaptive collaboration support (coaches and monitors) Virtual students	PhelpS (Greer, et al., 1998) HabiPro (Vizcaíno, Contreras, Favela, & Prieto, 2000) **COLER (Constantino Gonzalez, Suthers, & Escamilla De Los Santos, 2003)** EPSILON (Soller, & Lesgold, 2003)
Intelligent Tutoring	Curriculum sequencing Intelligent solution analysis Problem solving support	VC-Prolog-Tutor (Peylo, Teiken, Rollinger, & Gust, 1999) **SQL-Tutor (Mitrovic, 2003)** **German Tutor (Heift, et al., 2001)** **ActiveMath (Melis, et al., 2001)** **ELM-ART (Weber, et al., 2001)**

Tabella 2. 2- Tecnologie AIWBES, origini e sistemi che le utilizzano

Intelligent collaborative learning è un interessante gruppo di tecnologie sviluppate dall'incrocio di due campi originariamente molto distanti tra loro: computer supported collaborative learning (CSCL) e ITS. Il recente flusso di lavori sull'uso di tecniche di AI per supportare la didattica collaborativa ha dato come risultato un incremento del livello di interazione tra questi campi. Al momento possiamo elencare, come minimo, tre distinte tecnologie all'interno del gruppo di intelligent collaborative learning: adaptive group formation e peer help, adaptive collaboration support, e virtual student. Un buon esempio di supporto di collaborazione adattiva è fornire da COLER (Gonzalez et al. 2003).

Le tecnologie per adaptive group formation e peer help cercano di usare la conoscenza (spesso rappresentata nel loro modello dello studente) per formare un gruppo coordinato capace di risolvere problemi collaborativi.
Le tecnologie per adaptive collaboration support, cercano di fornire un supporto interattivo per un processo di collaborazione, proprio come i sistemi interattivi di supporto ai problemi assistono uno studente individuale nella risoluzione di un problema. In tal modo, i sistemi di supporto alla collaborazione come COLER (Gonzalez et al. 2003) o EPSILON (Soller & Lesgold 2003) possono consigliare ed aiutare.
La tecnologia virtual student è a confronto più vecchia. Questa tecnologia cerca di introdurre differenti tipi di studenti virtuali in un ambiente di apprendimento che collaborano tra loro soprattutto attraverso e-mail, chat e forums.

Intelligent class monitoring e un'altra tecnologia AIWBES motivata da WBE. Nel contesto di WBE un "insegnante remoto" non può leggere i segni di comprensione o confusione sulle facce degli alunni. Questa mancanza di feedback può rendere difficile l'identificazione degli studenti difficili che hanno bisogno d'ulteriori attenzioni, studenti brillanti che hanno bisogno di essere stimolati, oltre che quella del materiale di apprendimento che può essere troppo facile, troppo difficile o confusionario. Il sistema WBE può seguire ogni azione dello studente, ma è spesso impossibile per un insegnante umano capire il senso del volume di dati che sono collezionati. I sistemi intelligent class monitoring cerca di usare l'AI per aiutare l'insegnante in questo contesto. Il pioniere di questo flusso di lavoro fu HyperClassroom (Oda et al. 1998) che usava la tecnologia fuzzy per identificare gli studenti nel WBE e negli ultimi due anni ne sono stati riportati altri esempi (Merceron & Yacef 2003), (Romero et al. 2003).

2.4 Una piattaforma di e-learning

Come esempio di piattaforma di e-learning, viene presentata la suite Docent Enterprice, una piattaforma di e-learning in grado di gestire le problematiche connesse con l'erogazione di un servizio di formazione in modalità asincrona.
La suite Docent è costituita dai moduli Docent LMS (Learning Management System) e Docent LCMS (Learning Content Management System).

In dettaglio il software applicativo è così composto:

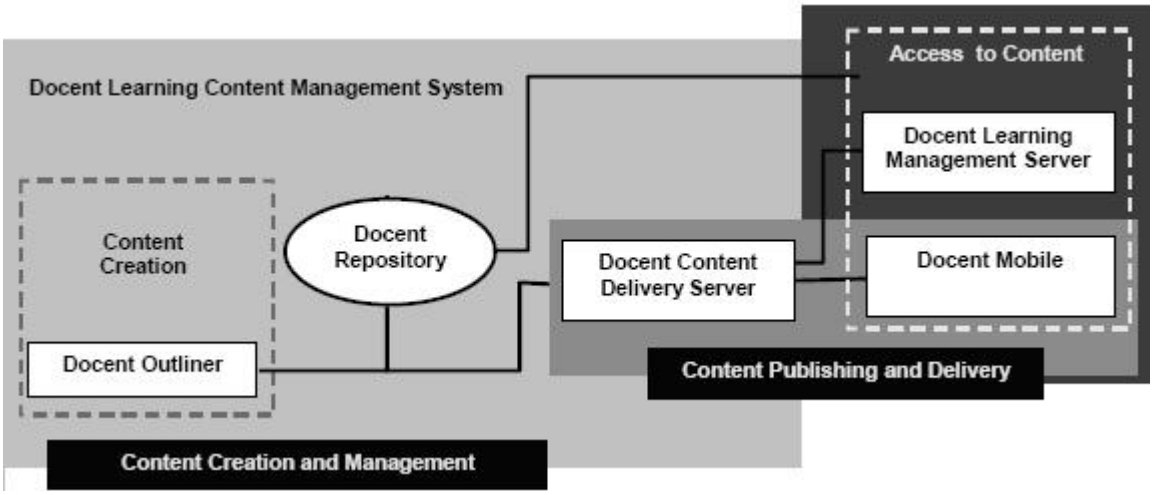


Figura 2. 14

- *Docent Learning Management Server*: Per la gestione e l'amministrazione dei processi di learning;
- *Docent Learning Content Management Server*: Per la gestione delle attività relative alla creazione, pubblicazione ed erogazione dei contenuti didattici. Questo modulo è articolato nei seguenti prodotti:
    - ° Docent Content Delivery Server: Per l'assessment, il content delivery ed il tracking dei risultati dei test;
    - ° Docent Mobile: Per l'erogazione di corsi in modalità off-line agli studenti;
    - ° Docent Outliner: Per lo sviluppo di Learning Object erogabili attraverso la piattaforma;
    - ° Docent Repository: Per l'archiviazione ed il riuso dei contenuti utilizzati nella creazione dei Learning Object. Offre funzioni di ricerca e di "version control".

Tale piattaforma e conforme ai maggiori standard in circolazione in materia di e-learning (AICC, SCORM, IMS) per garantire l'interoperabilità e l'integrabilità nel sistema di materiale didattico proveniente da LMS differenti.

2.5 Importanza degli standard nei sistemi LMS

Con lo sviluppo sempre più crescente di sistemi didattici basati sul web, è sorta la necessità di rendere riutilizzabile, interoperabile e accessibile la conoscenza codificata nei diversi sistemi di e-learning. Per far fronte a tali necessità, sono sempre più numerosi i sistemi LMS che adottano specifiche standard emesse da organizzazioni che operano nel campo di standardizzazione dell'e-learning come:

- **IMS Global Consortium (IMS):** E' un consorzio che produce specifiche basate su XML descriventi caratteristiche chiave di corsi, lezioni, esercitazioni e gruppi di lavoro. Il lavoro dell'IMS è descrivere in modo uniforme le risorse di apprendimento, con l'intento di facilitare la ricerca, la memorizzazione e lo scambio d'informazioni. Le specifiche più note: IMS Meta-data, IMS Content Packaging, IMS QTI (Question and Test Interchange).
- **Advanced Distributed Learning (ADL):** L'ADL è un'organizzazione (finanziata dal governo americano) che realizza ricerca e sviluppo nel campo dell'e-learning. La pubblicazione maggiormente rilevante è lo SCORM (*Shareable Content Object Reference Model*): combina elementi di specifiche IEEE, IMS e AICC ed è un "pilastro" dell'istruzione a distanza odierna.
- **Aviation Industry CBT Committee (AICC):** Creato nel 1988, è un gruppo internazionale che crea linee guida per CBT destinati all'aviazione. Hanno prodotto le linee guida **CMI** (Computer-managed istruction), ampiamente utilizzate (includono record sulle performance degli studenti, diari delle lezioni).
- **IEEE Learning Technology Standards Committee (IEEE LTSC):** All'interno dell'IEEE (*Institute for Electrical and Electronic Engineers*) è stata fondata una commissione **LSTC** atta a produrre specifiche e standard sulle tecnologie orientate al learning. La specifica più nota è il LOM (Learning Object Metadata), usata sia da ADL che da IMS.

2.5.1 Standard per i metadati

Esistono attualmente due metadata standard per il dominio didattico: *IMS* e *LOM.*

2.5.1.1 IMS

L'IMS global Learning Consortium propone un insieme di standard per l'e-learning. Il primo e il più importante di questi è IMS Metadata Specification (IMS 2001).

Usualmente uno standard per i metadati definisce un insieme di categorie che raggruppano la conoscenza su un oggetto individuale del dominio. Le categorie proposte dall'IMS Metadata Specification sul dominio didattico usano la notazione di Learning Object (LO) per indicare un oggetto atomico del dominio diattico. Un LO è generalmente una risorsa, come un documento testuale, slide, figure e così via,che rappresenta un pezzo di conoscenza.

Le principali categorie IMS che descrivono i Learning Object sono:

1. **General:** contiene informazioni generali come l'identificativo del LO, il titolo, la lingua ecc..
2. **Life-Cycle:** Descrive la storia della risorsa, il suo stato attuale, l'ultima modifica,…
3. **Metametadata:** descrive il sistema di classificazione stesso.
4. **Tecnical:** contiene informazioni sui requisisti tecnici per utilizzare i LO, come ad esempio il computer la potenza ecc..
5. **Educational:** Raggruppa le caratteristiche didattiche e pedagogiche dei LO.
6. **Right:** Esplicita informazioni su eventuali diritti d'autore sul LO.
7. **Relation:** indica le caratteristiche del LO rispetto ad altri LO.
8. **Annotation:** Contiene annotazioni e commenti sulla risorsa.
9. **Classification:** da la classificazione del LO in base ad una specifica tassonomia.

2.5.1.2 LOM

LOM sta per Learning Object Metadata (IEEE P1484.12) ed è uno standard definito dal Learning Tecnology Standardization Committee (LTSC) dell'IEEE. Come L'IMS Specification Metadata lo standard LOM ha una struttura gerarchica. Ognuno dei suoi elementi può essere *mandatiry, optional* o *conditional* (in tal caso il valore dell'attributo dipende dalla presenza o assenza di un altro attributo).

I principali elementi dello schema LOM sono attualmente 47 e possono essere raggruppati in 9 categorie come mostrato in Tab 2.3.

Tabella 2.3- Categorie LOM

Category	Description	Elements
General	Groups all context independent features together with the semantics descriptors of the resource.	Identifier, Title, CatalogEntry, Language, Description, Keywords, Coverage, Structure, Aggregation Level.
Life Cycle	Groups the features related to the life cycle of the resource.	Version, Status, Contribute.
Meta Metadata	Groups the features of the description itself rather than the resource which is described.	Identifier, CatalogEntry, Contribute, Metadata Scheme, Language.
Technical	Groups the technical features of the resource.	Format, Size, Location, Requirements, Installation Remarks, Other Platform Requirements, Duration.
Educational	Groups the didactic and pedagogical features of the resource.	Interactivity Type, Learning Resource Type, Interactivity Level, Semantic Density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Description, Language
Rights Management	Groups the resource features depending on the type of use we plan for it.	Cost, Copyright and Other Restrictions, Description.
Relation	Groups the resource features which relate it to other resources.	Kind, Resource, Identifier, Description, CatalogEntry.
Annotation	Allows comments to be made about the resource's educational content.	Person, Date, Description.
Classification	Describes a particular classification system in which the resource is defined (such as taxonomies, conceptual graphs, and so on).	Purpose, TaxonPath, Description, Keywords.

2.5.2 Standard per il modello dello studente

2.5.2.1 IMS Learner Information Package (LIP)

La specificazione IMS LIP fornisce una strada per assemblare le informazioni sugli studenti e renderle scambiabili tra i vari LMS (IMS Lip).

Le informazioni sull'utente sono divise in undici categorie principali (Figura 2.15):

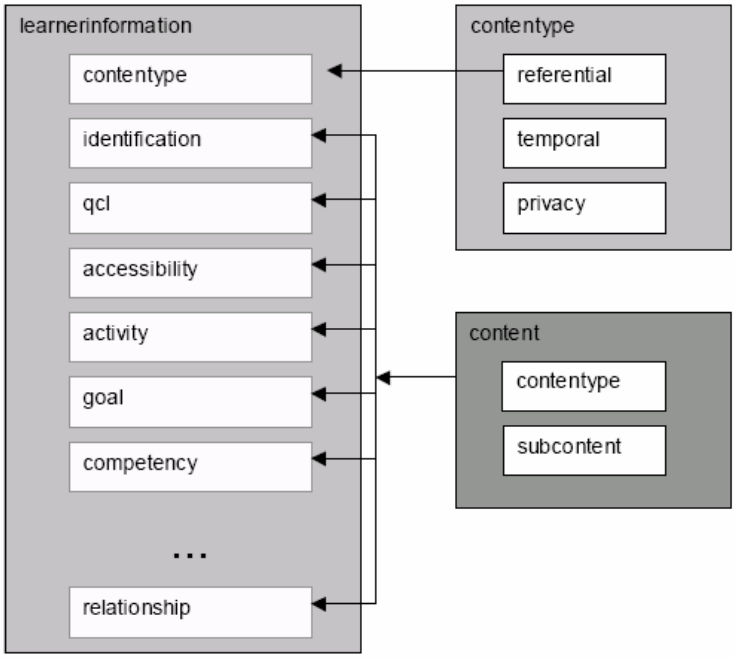


Figura 2. 15-IMP LIP data structures

- *Identification*: Contiene le informazioni personali rilevanti per ogni utente;
- *Qlc*: è usato per l'identificazione delle qualifiche,certificazioni e licenze di autorità competenti;
- *Accessibility*: Riguarda l'accesso generale alle informazioni del learner cioè capacità di linguaggio, incapacità, eleggibilità e preferenze di apprendimento;
- *Activity*: Può contenere informazioni su attività di apprendimento formale e non formale, incluso esperienze lavorative, servizio militare o civile.
- *Goal*: Rappresenta l'apprendimento,la carriera ed altri obiettivi del learner;
- *Competency*: Fornisce uno spazio per gli obiettivi, le esperienze e la conoscenza acquisita;
- *Affiliation*: Rappresenta le informazioni registrate circa i membri di un'organizzazione professionale;
- *Transcript*: rappresenta il sommario dei risultati accademici;
- *Interest*: può contenere informazioni riguardanti hobby e attività ricreative;

- *Relationship*: punta alle relazioni all'interno dei dati;
- *Security key* :per settare password e chiavi assegnate al learner.

2.5.2.2 IEEE PAPI

PAPI (Public and Private Information), è lo standard proposto dall'IEEE LTSC (PAPI 2002) per strutturare le informazioni sugli utenti nel seguente modo:

- *Personal information*: contiene le informazioni circa il nome, contatti e indirizzi del learner.
- *Relations information*: fornisce una categoria di informazioni del learner in relazione ad altre persone;
- *Security* : punta a fornire spazi per credenziali e accessi corretti alle risorse;
- *Preference*: indica il tipo di dispositivi e oggetti che il learner è abile a riconoscere;
- *Performance*: riguarda le informazioni storiche circa la misurazione della performance di un learner attraverso il materiale di apprendimento;
- *Portfolio*: riguarda le esperienze precedenti di un utente;

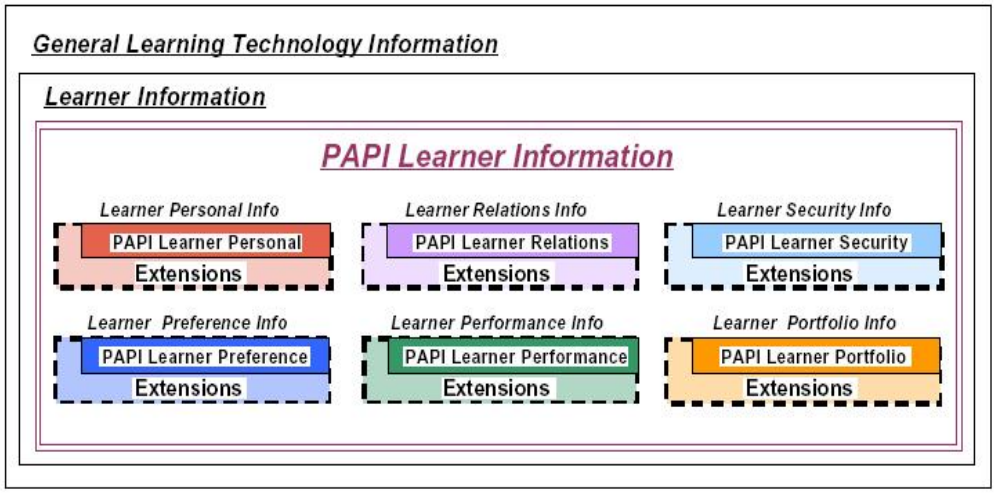


Figura 2. 16-Categorie dello standard PAPI

Capitolo III

3 Un'Architettura per il delivery intelligente e adattivo dei contenuti

3.1 Introduzione

In questo capitolo viene definita in linea di massima un'architettura Web-based per un sistema di e-learning intelligente e adattivo. Intelligente perché sfrutta le tecniche e metodologie dell'Intelligenza Artificiale per l'automazione di alcuni processi formativi, adattivo perché è in grado di adattarsi alle reali esigenze e preferenze dell'utente.
Attraverso l'architettura viene descritto l'insieme dei moduli che consente di creare un percorso formativo personalizzato (*learning path*) a partire da un processo di *competence gap analisys* applicato congiuntamente alla rappresentazione delle mappe di competenza e alla rappresentazione dello stato cognitivo dello studente. L'aggiornamento dei parametri significativi del sistema avviene dopo la fruizione del materiale didattico attraverso due tipologie:

- aggiornamento "system-driven"
- aggiornamento "free-learning"

Nell'aggiornamento di tipo "system-driven" i dati della fruizione "system-driven", ovvero della fruizione delle risorse didattiche pianificate dal sistema, andranno ad aggiornare dinamicamente le competenze acquisite dall'utente. Nell'aggiornamento di tipo "free-learning", invece, i dati della fruizione "free-learning", ovvero della fruizione libera avulsa dalla pianificazione, andranno ad aggiornare le mappe di competenza.

3.2 Descrizione del sistema

Il sistema si sviluppa intorno ai tre fondamentali modelli di rappresentazione, che sono:

- modello del dominio;
- modello dello studente ;
- mappe di competenza;

e comprende i seguenti principali processi:

- creazione dell'ontologia di dominio;
- competence gap analisys;
- generazione del learning path e presentazione dei contenuti;
- aggiornamento delle competenze e preferenze dell'utente (system-driven);
- aggiornamento delle mappe di competenze (free-learning);

Una visione complessiva dei modelli e relativi processi coinvolti è mostrata in Figura 3.1. A partire dai contenuti grezzi verrà costruito un modello del dominio (ontologia di dominio) in due step successivi, uno automatico per l'estrazione della semantica dai dati e l'altro semiautomatico per la definizione da parte di un esperto del dominio del modello. Le mappe di competenza rappresentano le competenze che l'utente deve acquisire e vengono definite una prima volta da un esperto (ad esempio l'ufficio del personale). L'idea è rappresentarle in maniera isomorfa al modello del dominio per consentire un confronto più semplice tra i due modelli ma nel contempo avere anche una rappresentazione che ne consenta l'aggiornamento dinamico.
Il processo di pianificazione del percorso formativo personalizzato è un processo inferenziale di "competence gap analysis" sulla rappresentazione della mappa di competenza e delle competenze specifiche dello studente.

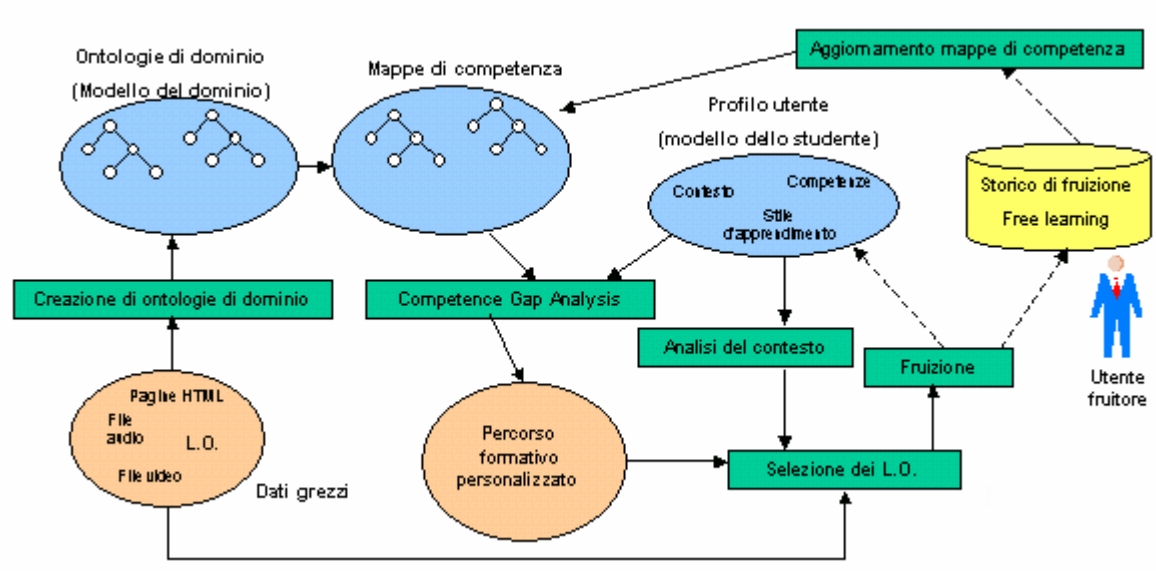


Figura 3. 1-Visione generale del sistema

Analizzando infatti le competenze che lo studente possiede e quelle che deve raggiungere si individuano le competenze che deve ancora acquisire.

Il percorso così generato è una pianificazione di massima che non istanzia ancora i learning object da utilizzare. Questo compito viene svolto successivamente andando a individuare il device di fruizione (desktop, mobile), lo stile di apprendimento dello studente e il rating dei vari learning object. I dati della fruizione andranno ad aggiornare dinamicamente il profilo dell'utente (le competenze che ha acquisito oppure qualche variazione allo stile di apprendimento). I dati della fruizione libera andranno invece ad aggiornare le mappe di competenza. L'ipotesi di buon senso che si è fatta è che chi va ricercare liberamente delle informazioni lo fa in quanto utile per il proprio ruolo professionale o per risolvere un particolare problema legato alla propria professione. A tale ipotesi si aggiunge che si provvederà l'aggiornamento delle mappe di competenza solo su grandi numeri cioè dopo che un certo numero di studenti che ricoprono lo stesso ruolo hanno ricercato lo stesso di tipo di informazione non prevista nel proprio percorso formativo.

Nella Figura 3.2 è presentata, a scopo esemplificativo, una rappresentazione dei tre modelli.

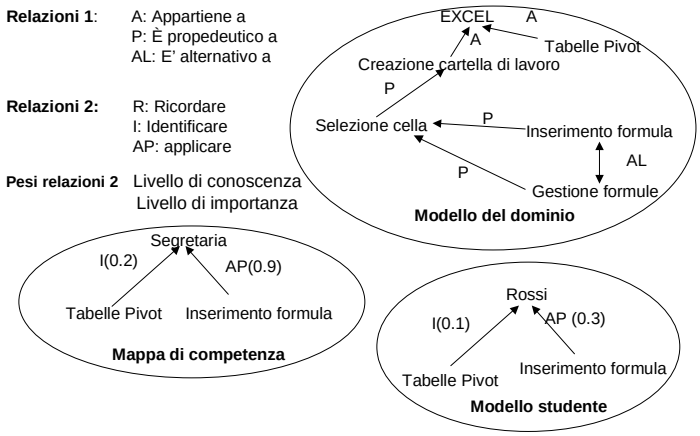


Figura 3. 2- Una possibile rappresentazione esemplificata dei tre modelli

Il dominio didattico è quello del programma Excel, e si compone di una serie di sottoconcetti. Lo studente in questione (Rossi) ha uno stato cognitivo, che sono le competenze che egli possiede sui concetti "Tabelle Pivot" e "Inserimento formula", con

il relativo livello di conoscenza. Per assumere le competenze di "Segretaria", così come indicato nelle mappe di competenza, l'utente deve essere capace di identificare le "Tabelle Pivot" con un livello di 0.2 e di applicare l'"Inserimento formula" con livello di 0.9.. Dalla competence gap analysis tra il modello dello studente e le mappe di competenza si evince che "Rossi" ha un livello di conoscenza sugli argomenti non sufficiente a fargli assumere la competenza di "Segretaria", deve allora seguire un percorso di apprendimento, che dal modello del dominio si deduce essere (Tabelle Pivot, Inserimento formula) o in alternativa (Tabelle Pivot, Gestione formule).

3.3 Architettura

L'architettura concettuale può essere schematizzata come in Figura 3.3 in cui individuiamo i seguenti componenti:

1. Modulo per la costruzione automatica dell'ontologia;
2. Modulo per la costruzione manuale dell'ontologia;
3. Modulo per la costruzione di learning path;
4. Modulo per l'estrazione di learning object;
5. Modulo per la profilazione utente;
6. Modulo per l'adattamento delle mappe di competenza;
7. Modulo per la pubblicazione dei Learning Object;
8. Modulo per la coordinazione delle transazioni tra Web Services.

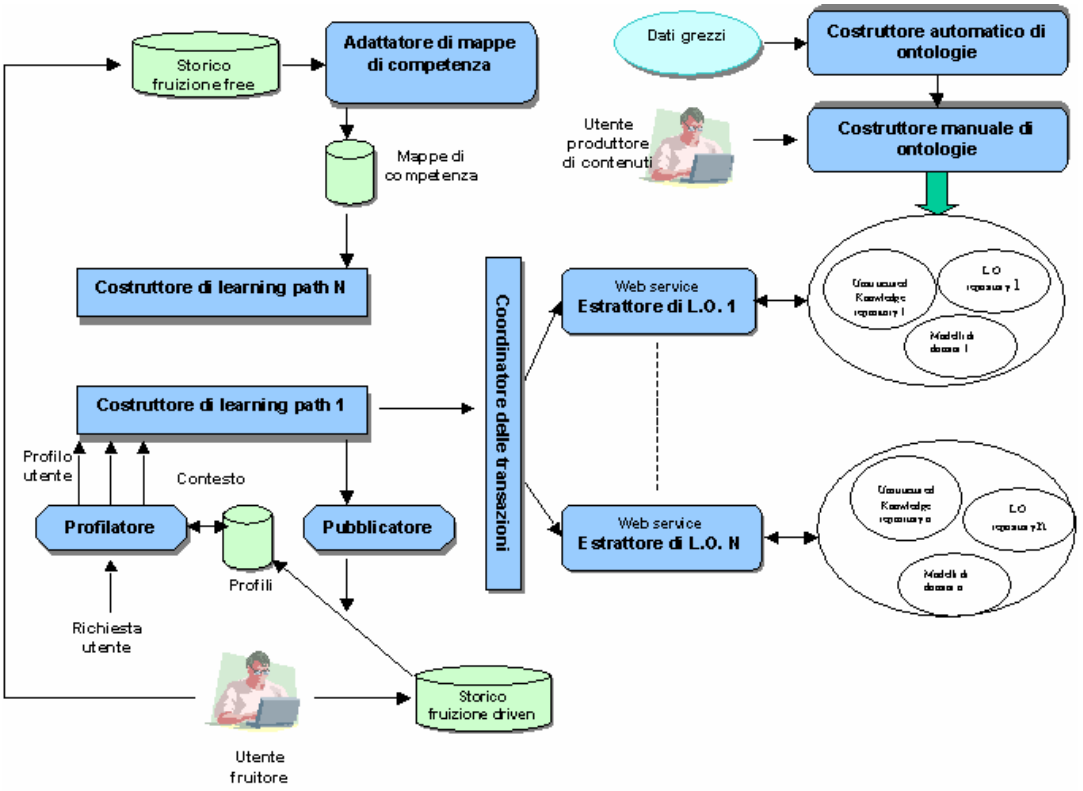


Figura 3. 3-Architettura

3.3.1 Modulo per la costruzione automatica delle ontologie

Il **modulo per la costruzione automatica di ontologie** è un processo automatico di estrazione ed aggiunta di metadati di primo livello (non necessariamente conformi ad uno standard). Il suo compito è quello di costruire, in maniera automatica ed a partire dai dati grezzi (filmati, documenti, ecc.), i metadati ad essi relativi.

L'indicizzazione automatica dei dati ha però dei limiti dovuti all'impossibilità di un qualsiasi tool automatico di discernere il contesto in cui un documento si pone. Di fatto un estrattore automatico si limiterà ad estrapolare dal testo le informazioni più significative, non potrà mai inserire informazioni non contenute nel testo, anche se magari implicite ad esso (il contesto appunto). Inoltre l'estrazione automatica rimane sempre molto esposta alla ambiguità dei termini. Del resto non si può pensare di fare a meno di essa.

Diamo allora una descrizione generica e semplificata di un tool di estrazione automatica.

Una prima fondamentale componente del tool sarà un Text Zoner: un programma che si accolla il compito di suddividere il testo in parti strutturali (title, body, etc...). Fatto

questo, il testo è passato a un Preprocessor che fa una analisi morfologica delle frasi, stabilisce cioè di che parti è costruita una frase (soggetto, predicato, oggetto). Un Filter eliminerà dal testo le frasi e le sentenze ritenute irrilevanti.

A questo punto si cercherà di capire quali sono le informazioni più interessanti. Attraverso un Named Entity Recognizer si possono identificare strutture lessicali minime come nomi propri, date, numeri, nomi di società, etc. Tutte queste informazioni verranno organizzate quindi da un Parser che isolerà i risultati significativi e ne fornirà la gerarchia di relazioni, ordinandoli secondo un albero.

Un Lexical Disambiguation dovrà assicurasi che i termini con più significati siano tradotti in un unico modo, parole con significato molto vicino o espressioni composte simili saranno ricondotte ad un unico termine.

A questo punto dobbiamo iniziare a produrre metadati; qualche cosa che potremmo chiamare un Semantic Interpreter riporterà gli elementi significativi riscontrati su strutture ontologiche possedute a priori. Si traduce il lavoro svolto dai componenti precedenti in relazioni espresse da ontologie che descrivono il dominio di nostro interesse.

Questa beve descrizione fa capire molto bene come il tipo di vocabolari ed ontologie utilizzate dalle applicazioni di estrazione dei metadati influiscano enormemente sui loro risultati. Il tool cercherà quei termini che avrà in una sua lista, riconoscerà le strutture sintattiche che gli saranno state fatte conoscere, stabilirà vicinanze semantiche in base a vocabolari e tesauri specifici.

3.3.2 Modulo per la costruzione manuale dell'ontologia

Il **modulo per la costruzione manuale di ontologie** consente a partire dai metadati generati automaticamente, sia di aggiungere metadati di secondo livello (conformi ad uno standard) sia di costruire l'ontologia di dominio e agganciare i metadati ai concetti di tale ontologia.

Il processo di costruzione dell'ontologia di dominio è un processo delicato che richiede la conoscenza del dominio da rappresentare, perciò in questa fase si richiede l'aiuto di un esperto del dominio per individuare i concetti chiave e le relazioni che intercorrono tra essi.

Il modo in cui viene costruito un dominio didattico, facendo uso di ontologie, sarà ampiamente discusso nel capitolo 4.

3.3.3 Modulo per la costruzione di Learning Path

Il **Modulo per la costruzione di Learning Path** è un sistema software in grado di interrogare il modello dello studente ed estrarre informazioni sugli obiettivi da raggiungere (goal). In base a tali obiettivi va ad interrogare le mappe di competenza per individuare le competenze da acquisire.
Un processo di *competence gap analysis* confronterà le competenze che lo studente deve possedere con quelle che possiede e con il modello del dominio ed individuerà il giusto percorso formativo che deve seguire; in uno step successivo, attraverso *l'estrattore di Learning Object* si provvederà ad associare al percorso formativo i learning object che meglio si prestano al particolare contesto di fruizione e alle preferenze individuali di ogni singolo utente.

La problematica che si deve affrontare è relativa all'individuazione di metodologie di rappresentazione delle mappe di competenza e di metodi di inferenza che consentano il confronto tra mappe di competenza ed i profili degli utenti al fine di comporre un percorso formativo personalizzato. L'obiettivo è:

1. consentire l'effettuazione di una competence gap analysis per individuare le competenze/livello di competenza che l'utente deve acquisire per ricoprire un determinato ruolo;
2. navigare nelle ontologie di dominio per individuare i Learning Object da estrarre dal repository dei dati grezzi filtrandole anche in base al contesto in cui verranno fruite.

Trovare quindi un modo per rappresentare le mappe di competenza funzionale agli obiettivi del progetto è di fondamentale importanza. Alcuni prodotti di e-learning disponibili in commercio utilizzano una gerarchia di competenze per catturare gli skill necessari per ricoprire determinati ruoli. In tali gerarchie le competenze sono mappate ai corsi che possono aumentare le competenze di un impiegato in una certa area. Non esiste uno standard industriale per la rappresentazione di queste gerarchie di competenze anche se sono in atto degli sforzi in tal senso -standard HRXML01- o anche IMS (Learner Information Packaging e Reusable Competency Definitions).
Queste rappresentazioni fornite dai prodotti in commercio, però, non sono in grado di catturare la semantica che può essere catturata, invece, usando una rappresentazione basata su ontologie. Con essa si possono catturare le relazioni tra varie competenze e le

relazioni con altre ontologie (ad esempio ontologie di dominio). Tali relazioni possono consentire ad un sistema intelligente di effettuare ragionamenti ed inferenze sulle competenze.

3.3.4 Modulo per l'estrazione di Learning Objects

Il **modulo per l'estrazione di Learning Objects** ha come compito quello di interrogare il repository dei contenuti ed estrarre il giusto materiale da presentare agli studenti tenendo conto delle preferenze contenute nel modello dello studente.
Tali preferenze riguardano la lingua, la presentazione del materiale ecc…
L'estrazione della giusta informazione è possibile grazie all'indicizzazione dei dati, processo che associa ai Learning Object i relativi metadati, ossia dati che li descrivono.

3.3.5 Modulo per la profilazione dell'utente

Il **modulo per la profilazione** si occupa della profilazione dell'utente. Con l'espressione *profilazione dell'utente* si intende quell'insieme di strutture dati, algoritmi e strumenti di supporto che concorrono a rappresentare ed individuare in maniera corretta il profilo degli utenti che accedono al sistema, abilitando all'erogazione di servizi e risorse formative ed informative che si presume possano effettivamente interessare l'utente.
Gli obiettivi possono essere sintetizzati come segue:

- individuazione delle principali caratteristiche da considerare per la definizione di un profilo utente nell'ambito di servizi orientati alla formazione ed informazione dell'utente;
- definizione, per ciascuna delle macro aree in cui il profilo utente va segmentato, le modalità di rappresentazione più idonee (rappresentazione dei dati e delle caratteristiche utente) e delle strategie di adattamento ed aggiornamento.

La rappresentazione del profilo di un utente è assimilabile ad una quadrupla:

$$\{\Pi_{LS,}\ \Pi_{C,}\ \Pi_{FL,}\ \Pi_{CX}\}$$

In cui i diversi elementi sono inerenti ad una delle macro caratteristiche che vengono considerate per rappresentare il profilo, ed in particolare:

- Π_{LS} rappresenta la n-upla contenente le informazioni inerenti le risposte fornite dall'utente al questionario di rilevazione del learning style, nonché il learning style che il modello inferenziale ha individuato a partire dalle risposte dell'utente ed il grado di *belief* che il modello ha relativamente al learning style assegnato all'utente;
- Π_{C} rappresenta il riferimento, ad esempio un URI, che consente di recuperare la rappresentazione della competenza (in termini di obiettivi raggiunti ed obiettivi da raggiungere) associata all'utente;
- Π_{FL} rappresenta la rilevazione delle aree in cui l'utente si trova a navigare in situazioni di free learning (*User History* ricerca di risorse formative non necessariamente legate ad un percorso formativo assegnato, ma ad un'esigenza che l'utente manifesta al momento);
- Π_{CX} rappresenta la n-upla inerente le informazioni di contesto di fruizione dipendenti da dispositivo impiegato e localizzazione dell'utente.

Per ciascuna di tali aree, si cercherà un modello di rappresentazione. Un punto di partenza è sicuramente l'utilizzo di standard di rappresentazione per il learner IEEE PAPI e IMS-LIP. Verranno determinati opportuni test sets, ove possibile con esempi positivi e negativi, che costituiranno la base di conoscenza dei modelli di inferenza impiegati per le diverse aree in cui il profilo è segmentato A seconda del successo o meno della scelta effettuata, il modello di inferenza specifico provvede in maniera automatica all'aggiornamento della rispettiva base di conoscenza. Si investigheranno i seguenti criteri per determinare il successo o meno della scelta operata dal sistema:

- superamento da parte degli utenti degli obiettivi didattici;
- accettazione esplicita, da parte dell'utente, della risorse informative suggerite, nello stesso ordine di presentazione;
- accettazione esplicita, da parte dell'utente, della risorse informative suggerite, in un ordine differente da quello di presentazione;

In particolare, l'aggiornamento avviene al termine di ogni attività di verifica dello studente considerando i risultati ottenuti ai test (che, essendo Learning Object, sono essi

stessi legati ai concetti del dominio) e mediandoli con i risultati dei test precedenti relativi agli stessi concetti. Terminata questa fase di aggiornamento, si ricorre all'osservazione congiunta del materiale didattico utilizzato e delle conoscenze acquisite al fine di determinare il grado di ricettività dello studente ai vari tipi di stimoli derivanti dalle varie tipologie di materiale.

3.3.5.1 Learning style

Le principali metodologie per l'individuazione dei learning styles (stili di apprendimento) sono essenzialmente riconducibili, come punto di partenza, a David A. Kolb (1975) a cui si deve la definizione del Learning Style Inventory, cioè di una metodologia di rilevazione, basata sull'impiego di questionari le cui domande / risposte sono orientate alla determinazione di come uno studente preferisca apprendere.

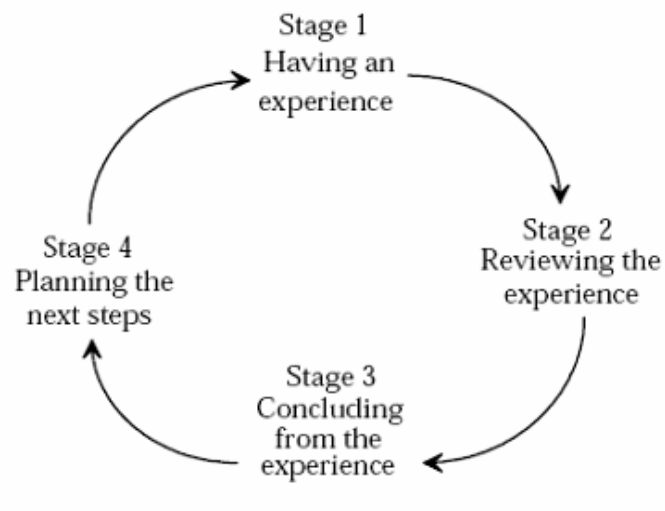


Figura 3. 4- Ciclo di apprendimento

Secondo Kolb, l'apprendimento da parte di uno studente avviene secondo un ciclo a quattro stati (Figura 3.4). In un primo momento, l'utente si trova a "vivere" una concreta esperienza, che induce in lui un processo di osservazione e di riflessione in merito a quanto "vissuto".

Dopo il processo di riflessione, avviene il processo di assimilazione con la definizione di concetti e generalizzazioni che portano l'utente, alla fine del ciclo, ad interagire in modo differente con il mondo esterno. Questo modello a sua volta riflette una suddivisione dei learning styles su di un piano secondo due assi cartesiani in cui i quattro versi possibili rappresentano, rispettivamente, l'approccio *Concrete Experience*

(CE), l'*Abstract Conceptualisation (AC)*, l'*Active Experimentation (AE)* e il *Reflective Observation (RO)*. Abbiamo quindi in questo modo quattro quadranti, uno per ognuno dei learning style:

- **Accomodator** (preferenza nell'apprendimento tramite esperienze concrete);
- **Diverger** (preferenza nell'apprendimento esaminando la situazione da diverse prospettive);
- **Assimilator** (preferenza nell'apprendimento a partire da diverse sorgenti informative, per poi operare astrazioni);
- **Converger** (preferenza nell'apprendimento tramite la determinazione di utilizzi pratici per idee e teorie).

Come detto, la rilevazione del learning style avviene attraverso l'impiego di opportuni questionari che, in funzione dei punteggi ottenuti dallo studente a tali risposte, consentono di fornire una prima indicazione relativamente il *profilo di apprendimento* dello studente.

3.3.5.2 Competenze – Stato Cognitivo

La rappresentazione del modello di competenze all'interno del profilo dello specifico utente (il suo stato cognitivo) è fondata su di un sotto insieme proprio degli attributi impiegati all'interno delle mappe di competenza

Per rappresentare le competenze nel modello dello studente, si possono usare algoritmi per consentire l'impiego di metodi Bayesiani combinati con logica fuzzy per associare, a ciascuna delle voci riportate nell'elenco delle competenze in possesso dell'utente, sia il grado con cui l'utente possiede quell'abilità, sia il grado di affidamento che il sistema ripone in questa stima. Il criterio di aggiornamento di queste informazioni è chiaramente dipendente dal superamento o meno delle attività di verifica presenti all'interno del percorso formativo: l'avere superato queste verifiche rafforza ad esempio l'indicatore di abilità relativo a questo skill ed anche la stima che il sistema ripone in ciò. Un insuccesso, implicherà un aggiornamento di tipo contrario. In particolare, per la rappresentazione dello stato cognitivo verranno investigate strategie, da sottoporre a validazione attraverso sperimentazione, i cui capisaldi possono essere così riassunti:

- L'obiettivo didattico è assimilabile alla competenza/ruolo professionale che la persona deve acquisire: il vocabolario di tale attributo è quindi identico al vocabolario impiegato per le ontologie inerenti le mappe di competenza;

- L'elenco delle "abilità" già acquisite dall'utente è, da un punto di vista concettuale, equiparabile ad una lista. Anche in questo caso, i vocabolari associati alle abilità sono coincidenti con quelli impiegati nelle ontologie di competenza e di dominio;

Per ogni abilità, si dovrà esprimere sia il grado di conoscenza (stimato) dal sistema, sia il grado di confidenza che il sistema ripone sulla sua stima, sia di come debbano modificarsi queste informazioni in base alle attività di free – learning (es.: la fruzione di un elemento a puro carattere di lettura è da discriminarsi rispetto alla fruizione di un elemento avente valenza di test o di valutazione).

3.3.5.3 User History

I dati raccolti durante le navigazioni degli utenti dei focus group durante l'effettuazione dei test bed, nonché dati derivanti da "navigazioni dummy" il cui scopo è l'inizializzazione della base di conoscenza, consentiranno al motore di knowledge discovery di evidenziare a quale pattern il comportamento dell'utente attuale possa essere attribuito. Questa informazione ha chiaramente una duplice utilità:

- da un lato consente di verificare la congruenza o meno di questo pattern con quanto atteso relativamente ad un profilo professionale a cui è associata la rispettiva mappa di competenza. Cio viene operato in base alle seguenti considerazioni:
    - è possibile riscontrare che una percentuale significativa di utenti manifestino la necessità di impiegare learning object che afferiscono ad una determinata ontologia di dominio, a sua volta associata ad una abilità professionale che, insieme ad altre, contribuisca al raggiungimento di una determinata competenza C_1 ;
    - in teoria tali utenti, a cui è stato assegnato il conseguimento di un determinato obiettivo didattico e quindi, attraverso le ontologie, risulterebbero essere associati ad una competenza C_2 e a determinate ontologie di dominio, non avrebbero motivo di ricercare quei learning object;
    - poiché la ricerca di contenuti è correlata alla necessità, da parte degli utenti, a specifici bisogni afferenti l'ambito lavorativo e non ad un mero divertimento, questo lascia intendere la necessità di operare un adattamento della mappa di competenza..

- Dall'altro lato la disponibilità di queste informazioni è necessaria per suggerire all'utente, quando accede al sistema per ricercare dei contenuti, risorse che potrebbero essere utili anche prima di avviare un'ipotetica ricerca.

3.3.5.4 Dispositivo di fruizione e localizzazione

La necessità di profilare l'erogazione di un contenuto in funzione del dispositivo e della localizzazione riveste un ruolo molto importante sia per quel che riguarda il tipo di *presentation layer* da impiegare, sia per quel che riguarda la possibilità, da parte del sistema, di fare pushing verso l'utente finale di contenuti dipendenti da dove l'utente stesso sta effettuando la richiesta. Ad esempio, in merito a questo punto, se l'utente si trovasse agli Uffizi e volesse ottenere informazioni inerenti un quadro del Botticelli, il sistema potrebbe rispondere con informazioni afferenti il periodo storico, da chi venne commissionato e così via, laddove se la stessa richiesta venisse effettuata da un utente che si trovasse presso uno degli uffici di Christie, il sistema dovrebbe rispondere con informazioni afferenti la quotazione, esperti che ne hanno certificato l'autenticità e così via.

3.3.6 Modulo per l'adattamento delle mappe di competenza

Il modulo per l'adattamento delle mappe di competenza è una rete di agenti intelligenti che fungono contemporaneamente da ausilio nella definizione delle mappe di competenza associate ai percorsi di certificazione ed alle aree di interesse, nonché da strumenti di adattamento automatico delle medesime mappe, dei percorsi formativi e del catalogo di contenuti di ausilio. In entrambi i casi tali agenti impiegheranno, fra le diverse fonti dati interessati, gli storici della fruizione libera inerenti la medesima area di competenza. In particolare, lo storico fruizione free potrà fornire al Competence Map Adapter le informazioni necessarie alla supervisione dello stato e relativo adattamento dei percorsi formativi erogati, nonché del catalogo di contenuti messi a disposizione dell'utente e delle mappe di competenza. Le mappe di competenza a loro volta contengono tutte le indicazioni inerenti le competenze e conoscenze da acquisire al fine di conseguire, da parte dell'utente, l'ottenimento dello specifico profilo professionale o abilità conoscitiva.

Quello che allora si deve poter fare, è aggiornare in maniera dinamica le regole che stanno alla base delle associazioni fra mappe di competenza e percorsi formativi e

quindi sviluppare e validare un insieme di metodologie che permettano di aggiornare tali meccanismi di associazione. Questo scopo può essere perseguito in due modalità fondamentali:

1. Esplicitazione delle informazioni contenute nei dati di utilizzo dei contenuti da parte degli utenti, con particolare riguardo ai concetti di *proficiency*, ossia il grado di possesso di una determinata competenza, nonché di *criticality*, cioè della scala di importanza da associare a tali competenze. I dati di uso dei contenuti in modalità *free learning* (in cui l'utente è definisce autonomamente il proprio percorso) possono sicuramente fornire un insieme di indicazioni, soprattutto di tipo statistico, che possono essere di ausilio nell'aggiornamento delle insieme di regole che sottendono al filtraggio dei contenuti ed alla definizione automatica dei percorsi formativi
2. Adattamento semimanuale delle regole di associazione in base a nuove esigenze, evoluzione dei domini applicativi, ridefinizioni delle conoscenze per la definizione di particolari skill e figure professionali.

In questa fase dobbiamo trovare il modo per trattare l'incertezza e poter quindi incorporare metodi per la rappresentazione di fenomeni probabilistici all'interno di linguaggi per la definizione di ontologie e poter così fornire a tali linguaggi una maggiore potenza espressiva per quantificare il grado di sovrapposizione o di inclusione fra concetti, effettuare ricerche di similarità, ad esempio fra concetti simili, ed eventualmente effettuare efficientemente integrazioni semantiche.
Lo sviluppo di sistemi che siano in grado di adattarsi dinamicamente a diversi contesti ed alle necessità sempre più personalizzate dell'utente passa necessariamente attraverso la realizzazione di sistemi in grado di interagire con il mondo esterno e di interpretare dati incerti in termini di modelli probabilistici. L'area dei modelli grafici probabilistici costituisce sicuramente uno dei maggiori sforzi della comunità scientifica in questa direzione. Molti autori ritengono infatti i modelli grafici probabilistici un meccanismo generale di inferenza bayesiana in grado cioè di conciliare informazione *a priori* basata sui metodi classici di acquisizione della conoscenza da parte di esperti e relative rappresentazioni, con metodi di adattamento che tengano conto delle distribuzioni osservate acquisendo i dati nelle particolari applicazioni.

Scopo di questa fase è quello dello sviluppare e applicare metodologie per il mapping di ontologie tenendo conto dell'incertezza. Le logiche del primo ordine possono, infatti, essere efficientemente applicate per rappresentare se una certa affermazione circa un dato è o meno vero. Tuttavia gran parte dei domini applicativi reali, come ad esempio lo sviluppo di sistemi per la personalizzazione della conoscenza e dell'apprendimento, contengono conoscenza incerta che è vera solo con certi gradi di "accuratezza" (ad esempio il livello di competenza acquisita da un discente o il livello di confidenza che un determinato percorso didattico risolva un particolare *knowledge gap*, etc.). La teoria della probabilità potrebbe essere efficientemente combinata con la logica per catturare o modellare l'incertezza. Diversi studi preliminari sono stati sviluppati, negli anni '90, per aumentare direttamente le logiche del primo ordine con la probabilità (Bacchus1990), altri studi sono invece finalizzati ad integrare con metodi probabilistici sottoinsiemi specifici come sistemi basati su regole (Poole, 1993) o sistemi basati su logica descrittiva (Giugno & Lukasiewicz, 2002), altri approcci ancora fanno uso della logica fuzzy per aumentare l'espressività di logica descrittiva (Straccia, 2001).

3.3.7 Modulo per la pubblicazione dei Learning Object

Il modulo per la pubblicazione dei Learning Object provvederà alla multicanalità fornendo le informazioni nel formato migliore avvalendosi delle informazioni sul contesto di utilizzo fornite dallo strumento di profilazione. In dettaglio, tale modulo terrà conto sia della ricchezza multimediale del contenuto, sia dei possibili canali di trasmissione (es: satellite, ADSL, …) con cui fruire del contenuto.

Tale approccio si fonda sulla descrizione basata su ontologie:

- del materiale formativo;
- del contesto di utilizzo;
- del profilo degli utenti

fornendo così un accesso personalizzato a questi materiali di apprendimento.

I criteri per la gestione della presentazione all'utente sono quindi riconducibili ai seguenti input:

- Ricezione, *dal modulo per l'estrazione di Learning Object,* di tutti i learning object inerenti la competenza da acquisire (in caso di fruzione *system driven* da parte dell'utente) o i criteri di ricerca ricevuti dall'esterno (in caso di fruizione *free learning*);

- Ricezione, dal *modulo per la profilazione*, delle informazioni afferenti all'utente a cui erogare i learning objects, inerenti il contesto di fruizione (dispositivo, localizzazione) ed il learning style;
- Gestione del rating, vale a dire considerazione automatica, da parte del sistema, dell'efficacia riscontrata in situazioni precedenti, nella fruizione di risposte da parte degli utenti.

L'output prodotto è relativo ai learning object che rispondono ai criteri presenti nel profilo (primo filtro – contesto, secondo filtro learning style), al rating ed alla metodologia di presentazione che si vuole sperimentare.

3.3.8 Modulo per la coordinazione delle transazioni

Il Modulo per la coordinazione delle transazioni coordinerà i servizi forniti dai moduli per l'estrazione dei Learning Object gestendo la transazionalità distribuita attarverso i Web Services.

Un Web Service è un servizio disponibile in rete ad uso di altri programmi, che può essere pubblicato, localizzato e invocato attraverso il Web. A sua volta, un servizio espone una o più funzionalità che devono essere specificate in termini di contratto tra il fornitore e l'utilizzatore. Una funzionalità offerta da un Web Service può andare da una semplice richiesta a complesse elaborazioni.

I Web Services comunicano direttamente con altri Web Services attraverso Internet e possono essere utilizzati come building block per realizzare applicazioni in sistemi distribuiti aperti, senza preoccuparci dell'implementazione dei componenti software che forniscono i servizi, in quanto ciò che risulta essere significativo, ai fini dei servizi offerti, è ciò che viene esposto e non come ciò venga realizzato.

I Web Services utilizzano protocolli e formati dati che sono standard per il Web, come HTTP (Hypertext Transfer Protocol) e XML (eXtensible Markup Language). Tale scelta fa sì che gli stessi servizi possano essere fruiti in ambiti particolarmente eterogenei per quel che riguarda la connettività (Internet – Intranet – Extranet sia in modalità wired che wireless) e non risentano di specificità tecnologiche proprie dei diversi fornitori. Inoltre, l'applicazione client che consuma i Web Services può essere implementata in un qualsiasi linguaggio di programmazione capace di creare e interpretare i messaggi definiti per l'interfaccia dei Web Services.

In definitiva, le caratteristiche fondamentali dei Web Services sono:

- **Interoperabilità**: tutti i Web Services possono interagire tra loro senza intermediazioni.
- **Accessibilità**: le applicazioni, sviluppate in qualsivoglia linguaggio di programmazione, possono essere esposte come Web Services velocemente usando appositi tools, e invocate da qualsiasi ambiente di sviluppo. Tutti i dispositivi, che supportano fondamentalmente le tecnologie HTTP e XML, possono accedere ai Web Services ed usarli.
- **Design:** i servizi possono essere realizzati seguendo metodologie tecnologiche di sviluppo standard, attraverso le quali individuare le responsabilità dei diversi componenti in maniera univoca (logica applicativa, logica di presentazione, logica di coordinamento). Ciò garantisce una migliore qualità progettuale, un maggior riutilizzo dei componenti sviluppati ed una maggiore trasparenza rispetto al particolare device impiegato dall'utente finale nella fruizione dei servizi formativi/formativi.

Queste tre caratteristiche unite all'estrema apertura li rendono particolarmente adatti ad essere utilizzati in contesti distribuiti dove è necessario integrare più applicazioni Web. E' per questo motivo che abbiamo ritenuto di basare l'architettura sui Web Services

Nella nostra architettura l'intelligenza per navigare nelle informazioni attraverso le ontologie sarà implementata in maniera distribuita nei Web Service. Questo implica la necessità di gestire meccanismi transazionali distribuiti.

Capitolo IV

4 Rappresentazione del dominio di conoscenza: un modello basato sulle ontologie

4.1 Modello del dominio di conoscenza.

Il concetto di *modello del dominio di conoscenza* costituisce il cuore dell'approccio basato sulla rappresentazione della conoscenza per lo sviluppo di sistemi di e-learning personalizzati, così come descritto nel precedente capitolo. In generale, il modello del dominio è composto da un insieme di concetti ciascuno rappresentativo di un frammento elementare di conoscenza per il dominio dato.

Nella forma più semplice di modello di dominio, l'insieme dei concetti può essere rappresentato attraverso l'uso di vettori i cui elementi sono, appunto, i concetti stessi. Tale forma di rappresentazione, tuttavia, va bene solo per rappresentare un insieme di concetti indipendenti, ovvero privi di connessioni semantiche. Nella Figura 4.1 viene mostrato un caso di modello di dominio di tipo vettoriale.

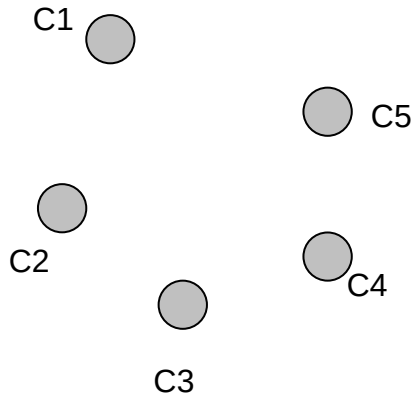


Figura 4. 1- Esempio di modello vettoriale

In una rappresentazione più complessa e più ricca semanticamente, invece, i concetti sono opportunamente collegati tra di loro attraverso relazioni del dominio d'interesse in modo da formare una rete di conoscenza o "knowledge network", come mostrato nella Figura 4.2.

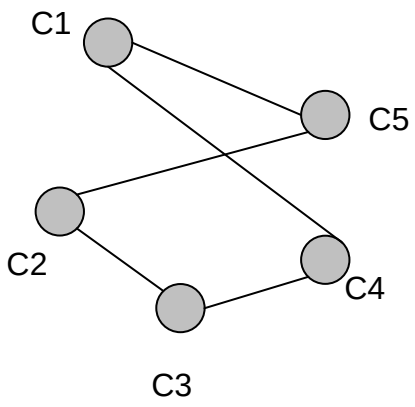


Figura 4. 2- Esempio di knowledge network model

Nelle reti di conoscenza l'informazione esplicita è descritta dai nodi della rete e dai collegamenti, in generale entrambi etichettati. La conoscenza implicita, invece, è derivata da opportuni algoritmi che seguono i collegamenti della rete. Diversamente da come accede nei metodi di rappresentazione basati sulla logica, come nella logica del primo ordine, in cui l'informazione esplicita è data fondamentalmente da una lista di stringhe e la conoscenza implicita è dedotta usando sistemi di ragionamento automatico che lavorano su un insieme di assiomi e regole caratteristici dello specifico sistema formale scelto.

Da questa differenza di base seguono due importanti conseguenze. La prima riguarda il formato di rappresentazione della base di conoscenza che nel caso dei sistemi basati sulla logica è di tipo testuale, mentre nel caso delle reti di conoscenza è di tipo grafico e quindi, in generale, più semplice da comprendere.

La seconda conseguenza riguarda il costo computazionale dell'inferenza della conoscenza implicita. I sistemi per l'inferenza logica devono effettuare, per ogni step di derivazione, un confronto tra i simboli delle formule finché non si verifica l'unificazione del goal con le variabili della clausola. In una struttura a rete, invece, non è necessario controllare tutta la knowledge base per effettuare un match, in quanto è sufficiente che l'algoritmo segua i collegamenti per trovare i relativi fatti che si stanno

cercando. In generale allora si può dire che le rappresentazioni che impiegano reti di conoscenza sono computazionalmente più veloci di quelle che impiegano la logica. Tuttavia, gli approcci basati su reti di conoscenza non hanno una potenza espressiva ricca come la logica del primo ordine essendo fondamentalmente basati su una metafora "Description Logic-Like".

I sistemi basati su frame e le reti semantiche sono due formalismi per rappresentare la conoscenza strutturata in rete. Le reti semantiche derivano dal lavoro di Charles Pierce, che propose nel lontano 1896, una notazione grafica chiamata "existential graph", adesso nota come "semantic network".

I sistemi a frame e le reti semantiche usano la stessa metafora: gli oggetti del dominio (concetti) sono nodi in un grafo, tali nodi sono organizzati in una struttura gerarchica, e i link tra i nodi rappresentano relazioni binarie. Nei sistemi a frame, in particolare, le relazioni sono trattate come slot in un frame che sono riempiti da altri, mentre, nelle reti semantiche esse sono pensate come frecce tra i nodi. A parte, dunque, la notazione grafica (box o grafi) il significato e l'implementazione dei due tipi di sistemi sono in effetti identici. Esempi di sistemi a frame sono: OWL, FRAIL, KODIAK, mentre esempi di reti semantiche includono: SNEPS, NETL e grafi concettuali.

Di regola i sistemi a frame e le reti semantiche possono essere traslati nel linguaggio DL (Description Logic). Da ciò scaturisce un fatto importante: i sistemi a frame e le reti semantiche sono decidibili, ovvero se lo spazio delle soluzioni è finito allora gli algoritmi basati su tali sistemi di rappresentazioni convergono in un tempo finito.

I metodi di rappresentazione della conoscenza, che si tratti di "semantic network" o di logica del primo ordine, forniscono naturalmente un framework generale per esprimere la conoscenza esplicita (i fatti del dominio) e per derivare la conoscenza implicita. Ad esempio, l'approccio basato su reti semantiche indica come modellare la conoscenza del dominio, cioè attraverso una gerarchia di concetti e relazioni binarie, ma non specifica la natura delle relazioni o i vincoli sulla specie di fatti da rappresentare. Tutto ciò è ragionevole se si pensa che differenti domini hanno differenti proprietà da modellare e un modello per la rappresentazione della conoscenza non può vincolarsi a tali proprietà. Da qui, però, segue una importante conseguenza, cioè che differenti applicazioni usando differenti linguaggi di rappresentazione della conoscenza, non sono in grado di scambiarsi la conoscenza sullo stesso dominio. Per superare questo limite il linguaggio da adottare per la descrizione semantica della conoscenza da scambiare deve essere

quanto più uniforme possibile e possibilmente standard. Questo vale a maggior ragione per la formazione a distanza basata sul web in cui la condivisione e il riuso della conoscenza del dominio e dei contenuti formativi sono due aspetti fondamentali. Per rispondere alla necessità di una rappresentazione condivisa della conoscenza ci viene in aiuto la definizione di un opportuno insieme di vincoli che possono essere imposti sulle descrizioni della conoscenza e il linguaggio OWL per la formalizzazione della realtà d'interesse. L'insieme di vincoli sulla rappresentazione delle descrizioni della conoscenza è costituito da vocabolari condivisi di termini e di relazioni tra di essi. I vocabolari e le relazioni possono essere condivisi tra differenti applicazioni che lavorano sullo stesso dominio. In generale, le descrizioni standard ricadono su due livelli, differenti per potenza espressiva e complessità computazionale: metadata e ontologie. Metadata e ontologie sono metodologie derivate dalle reti semantiche che uniscono i vantaggi della rappresentazione delle conoscenza link-based, descritti in precedenza, con gli aspetti di standardizzazione che consentono di scambiare e condividere informazioni tra differenti sistemi.

Il linguaggio standard OWL, per la definizione delle ontologie, consente una formalizzazione, più generale possibile, degli aspetti semantici (concetti e relazioni) di un dominio di conoscenza attraverso la strutturazione in classi/sottoclassi dei concetti, l'uso di oggetti, o "individuals", della realtà di interesse e di relazioni binarie, o "properties", tra gli oggetti. Tale linguaggio può essere utilizzato per la rappresentazione esterna della conoscenza, cioè per la descrizione della conoscenza che deve essere scambiata tra due sistemi. Ciascun sistema singolarmente provvederà successivamente a trasformare la rappresentazione ricevuta nella propria rappresentazione interna sulla quale potrà utilizzare gli strumenti di ragionamento specifici.

La metodologia per la rappresentazione della conoscenza basata sull'uso combinato di metadata, ontologia e OWL, in definitiva, si presta bene a formalizzare il dominio didattico. Questo dominio può essere visto come una specializzazione di un dominio di conoscenza più complesso al caso didattico in cui i concetti sono da una parte collegati tra loro con relazioni tipiche del mondo didattico e dall'altra a risorse formative che vanno a spiegare, appunto, tali concetti.

Nella Figura 4.3 viene illustrato un esempio di dominio didattico "Matematica".

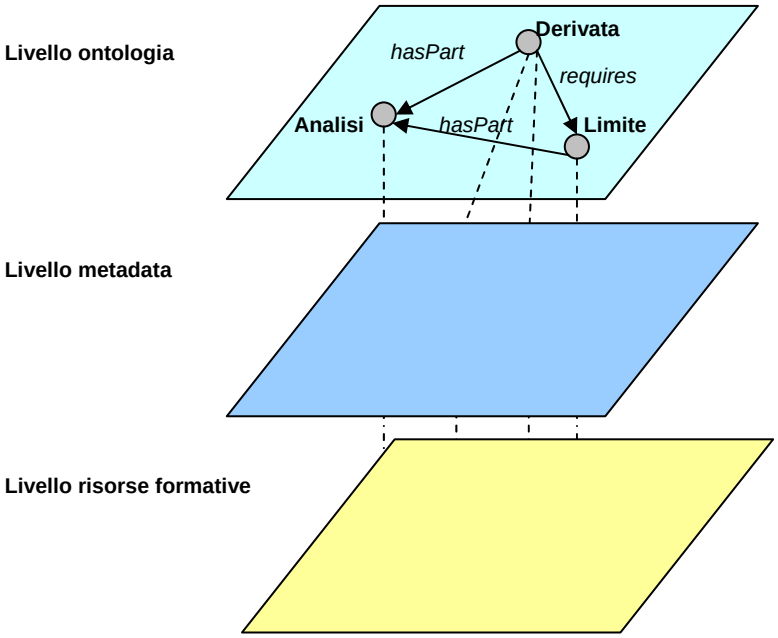


Figura 4. 3- Esempio di dominio didattico “Matematica”

Dall’ontologia dell’esempio si evince che per apprendere il concetto “Derivata” si richiede la conoscenza del concetto “Limite”, dato che i due concetti sono legati tra loro dalla relazione “requires”. I concetti sono separati dalle risorse formative (contenuti didattici che spiegano i concetti) attraverso il livello metadata.
Nei paragrafi successivi verranno descritte le ontologie e verrà definito un modello del dominio didattico.

4.2 Un approccio per la rappresentazione della conoscenza: l’ontologia

L’ontologia, “lo studio dell’essere in quanto essere”, - diceva Aristotele – è usualmente concepita come una disciplina strettamente filosofica. Eppure, negli ultimi anni grazie all’esplosione delle comunicazioni in rete, gli aspetti ontologici dell’informazione hanno acquistato un valore strategico. Tali aspetti sono intrinsecamente indipendenti dalle forme di codifica dell’informazione stessa, che può essere quindi isolata, recuperata, organizzata, integrata in base a ciò che più conta: il suo contenuto.
La standardizzazione dei contenuti dell’informazione risulta oggi cruciale nella prospettiva delle aziende integrate e del commercio elettronico ed è indispensabile per semplificare i processi di comunicazione. In generale, infatti, la mancanza di un’interpretazione condivisa porta alla povertà di comunicazione tra le persone e le loro

organizzazioni. Nel contesto della costruzione di un sistema IT (Information Tecnology), tale mancanza di comprensione porta a delle difficoltà nell'identificare i requisiti e nel definire le specifiche del sistema. Molti tools software, metodi di modellazione, paradigmi e linguaggi limitano l'interoperabilità tra i sistemi, il loro riuso e la loro condivisione.

E' proprio per superare i problemi precedenti che s'introduce l'ontologia, che cerca di eliminare o, almeno, ridurre le confusioni concettuali o terminologiche, in modo da avere un'interpretazione condivisa, in altre parole un vocabolario comune, con un significato per i vari termini su cui tutti sono d'accordo.

Sebbene l'ontologia sia nata nell'ambito filosofico, negli ultimi anni, si è affermata una nuova scuola di pensiero, che propone una caratterizzazione logica rigorosa delle categorie ontologiche fondamentali utilizzate nei sistemi informativi, con lo scopo di aumentarne la trasparenza semantica e l'interoperabilità. Tale approccio coinvolge attività di modellazione concettuale e di ingegneria della conoscenza in una prospettiva fortemente interdisciplinare.

4.3 Definizioni di ontologia

Sono state date diverse definizioni dell'ontologia, oltre a quella prettamente filosofica:

1. *"Un'ontologia identifica i termini basilari e le relazioni di un determinato dominio, definendone in questo modo il vocabolario, e le regole per combinare tali termini e tali relazioni, andando oltre il vocabolario stesso"* (Neches et al. 1991).

Tale definizione indica il modo di procedere per costruire un'ontologia: identificare i termini basilari e le loro relazioni; identificare le regole che li combinano; provvedere a definizioni di tali termini e relazioni. In base a tale definizione, un'ontologia non include solo i termini che sono esplicitamente definiti in essa, ma anche quelli che possono essere derivati usando tali regole.

2. *"Un'ontologia è un insieme di termini gerarchicamente strutturati per descrivere un dominio che può essere usato come fondamenta per una base di conoscenza"* (Swartout 1999).
3. *"Un'ontologia è un mezzo per descrivere esplicitamente la concettualizzazione presente dietro la conoscenza rappresentata in una base di conoscenza"* (Bernaras 1996).

Una definizione del termine "ontologia" largamente adottata, soprattutto nell'ambito delle "artificial intelligence communities", è quella proposta da Gruber, secondo cui un'ontologia è una "*specifica esplicita e formale di una concettualizzazione condivisa*" (Gruber 1995).
La *concettualizzazione* si riferisce ad un modello astratto di un qualche fenomeno avendone identificato i concetti; *esplicita* significa che i tipi di concetti usati e i vincoli sul loro uso sono esplicitamente definiti; *formale* si riferisce al fatto che l'ontologia dovrebbe essere "machine-readable"; *condivisa* riflette il fatto che l'ontologia cattura la conoscenza consensuale, cioè quella non propria di un individuo, ma accettata da un gruppo.

In questo lavoro di tesi, l'ontologia può essere vista come "*l'insieme dei termini e delle relazioni, che denotano i concetti utilizzati in un dominio*".
Essa può essere rappresentata come un insieme di concetti, una gerarchia di concetti o tassonomia, un insieme di relazioni tra concetti e un set di assiomi :

$$O = \{C, H^C, R, A\}$$

- C: insieme dei concetti che costituiscono l'ontologia;
- H^C: esprime una relazione gerarchica o tassonomica tra due concetti del dominio. $H^C(C_1, C_2)$ indica che il concetto C_1 è sottocncetto di C_2.
- R: insieme delle relazioni definite sul dominio, che legano tra loro elementi del dominio con elementi del condominio o range.
- A: indica un set di assiomi espressi in un appropriato linguaggio logico.

Con ontologia dunque ci si riferisce a quell'insieme di termini che, in un particolare dominio applicativo, denotano in modo univoco una particolare conoscenza e fra i quali non esiste ambiguità poiché sono condivisi dall'intera comunità d'utenti del dominio applicativo stesso.
Un'ontologia può essere vista come un thesaurus arricchito in cui, oltre alle definizioni e alle relazioni tra i termini di un dato dominio (fornite dal thesaurus), viene rappresentata e fornita una conoscenza più concettuale.
L'ontologia viene spesso confusa con una knowledge base; in realtà si tratta di due cose differenti, infatti, l'ontologia è una knowledge base particolare, che descrive fatti

considerati sempre veri dalla comunità d'utenti, mentre la knowledge base può descrivere fatti e asserzioni relative ad un particolare "state of affaire", che in genere non vale sempre.

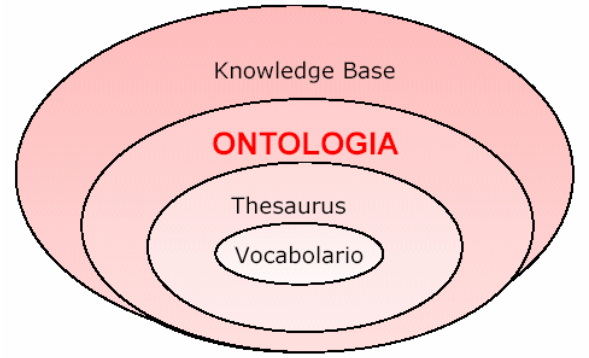


Figura 4. 2- Spazio concettuale

Sebbene l'ontologia non sia l'unico modo di specificare una concettualizzazione, esso ha l'interessante proprietà di permettere la condivisione della conoscenza tra software di IA.

4.4 Funzioni dell'ontologia

In base alla definizione d'ontologia e al significato della nozione di concettualizzazione, possiamo dedurre che un'ontologia consiste di:

- termini o concetti – termini generali che esprimono le categorie principali in cui è organizzato il mondo, come *cosa, entità, sostanza, uomo, oggetto fisico,* ecc. oppure termini particolari che descrivono un dominio d'applicazione specifico (ontologie di dominio);
- definizione dei termini;
- relazioni che li associano o impongono particolari vincoli.

L'ontologia svolge la funzione di:

- *Un lessico comune*. La descrizione di un ambiente target necessita di un lessico concordato tra le persone coinvolte. Un notevole contributo è dato dai termini contenuti in un'ontologia;
- *Spiegazione di ciò che è stato lasciato implicito*. In tutte le attività umane, si trovano assunzioni e presupposti impliciti, come la definizione di termini comuni e basilari, le relazioni ed i vincoli tra questi e i punti di vista nell'interpretazione dei fenomeni;

- *Sistematizzazione della conoscenza*. Richiede concetti/lessico ben stabiliti, in base ai quali le persone descrivono fenomeni, teorie, ecc. Un'ontologia, perciò, fornisce la backbone della sistematizzazione della conoscenza;
- *Funzione di meta-modello*. Un modello è generalmente un'astrazione di un oggetto reale. Un'ontologia fornisce concetti e relazioni che sono usati come blocchi costitutivi del modello, cioè specifica il modello da costruire, dando le direttive e i vincoli che dovrebbero essere soddisfatti in questo modello;

Un'ontologia può presentarsi in varie forme, ma include sempre un vocabolario di termini e una descrizione dettagliata del loro significato.

4.5 Tipi di ontologia

Vi sono diversi tipi di ontologia, che possono variare lungo tre dimensioni chiave: grado di formalità, natura del soggetto e scopo (campi di applicazione delle ontologie) (Uschold & King 1996).

4.5.1.1 Grado di formalità

Riguarda il grado di formalità in base al quale è creato un vocabolario e vengono specificati i significati dei vari termini; tale grado può essere:

- *altamente informale*: l'ontologia viene espressa in linguaggio naturale;
- *semi-informale*: viene utilizzata una forma ristretta e strutturata di linguaggio naturale, in modo da aumentare la chiarezza e ridurre l'ambiguità. Ne è un esempio l'Enterprise Ontology, una collezione di termini e definizioni rilevanti per il business dell'impresa
- *semi-formale*: l'ontologia viene espressa in un linguaggio artificiale, in modo formale, come ad es. il linguaggio Ontolingua;
- *rigorosamente-formale*: i termini vengono definiti in un linguaggio con semantiche formali e teoremi. Ne è un esempio il progetto TOVE (TOronto Virtual Enterprise) il cui obiettivo è creare un'ontologia per l'impresa.

4.5.2 Natura del soggetto

Riguarda la natura del soggetto che viene caratterizzato attraverso l'ontologia. Sono presenti le seguenti categorie:

- *ontologie di rappresentazione della conoscenza o meta-ontologie*: descrivono le primitive di rappresentazione usate per formalizzare la conoscenza in una base di conoscenza (come concetti, attributi, relazioni, ecc.) (Es. FrameOntology e Mereologia, che definisce la relazione delle *parti* e le sue proprietà);
- *ontologie generali/comuni*: includono vocabolari relativi alle cose, eventi, tempo, spazio, causalità, comportamento, funzioni, ecc. (Es. CYC);
- *top-level ontologies:* descrivono concetti molto generali (sotto cui sono legati tutti i termini delle ontologie esistenti) come spazio, tempo, materia, oggetto, evento, azione, ecc., che sono indipendenti da un particolare problema o dominio(Es. Guarino);
- *ontologie di dominio*: descrivono il vocabolario relativo a un generico dominio, le attività presenti in tale dominio, le teorie e i principi elementari che governano tale dominio (ad esempio un dominio medico, o automobilistico, o relativo all'impresa ad esempio Enterprise Ontology);
- *task ontologies*: descrivono il vocabolario relativo ad un generico obiettivo (come la diagnostica e le vendite), dando una specializzazione dei termini introdotti nella top-level ontology.
- *application ontologies*: contengono la conoscenza necessaria per modellare una particolare applicazione.

Questa distinzione tra le varie categorie di ontologia non è mai netta e una stessa ontologia può appartenere a più categorie.

4.5.3 Campi di applicazione delle ontologie

Riguarda i vari campi di applicazione o usi che vengono fatti dell'ontologia. Diamo una panoramica di tali campi di applicazione.

4.5.3.1 Esplicitazione e riutilizzo della conoscenza di dominio

La conoscenza gioca ormai un ruolo di fondamentale importanza, sia all'interno della singola impresa, che tra tutti gli attori dell'intera filiera produttiva. La conoscenza infatti è un asset nascosto che non compare sui bilanci annuali, ma che è molto importante per la creazione di valore e per i potenziali guadagni futuri. Diversi, infatti, sono i vantaggi che si ottengono integrando la conoscenza nelle strategie di business e nei processi

aziendali più importanti; alcuni di questi benefici sono l'evitare di avere costi errati, la condivisione delle pratiche migliori, la risoluzione più rapida dei problemi, tempi di sviluppo più veloci, migliori soluzioni e servizi per i consumatori e nuovi guadagni.
Le imprese stanno adottando delle strategie di business basate su due spinte importanti. La prima spinta consiste nel condividere la conoscenza, che esiste già, tra tutti gli attori. La seconda spinta consiste, invece, nell'innovazione, nella creazione di nuova conoscenza e nella sua commercializzazione, come se fosse un prodotto o un servizio molto prezioso.
La conoscenza può essere tacita o esplicita. La conoscenza tacita, o implicita, è personale, dipende dal contesto, esiste soltanto nella mente delle persone, e perciò è difficile da formalizzare e comunicare. La conoscenza esplicita, invece, è quella che può essere catturata e codificata nei manuali, nelle procedure o nelle regole, e che quindi si può trasmettere in un linguaggio formale e sistematico.
In tutte le attività umane, si trovano assunzioni e presupposti impliciti, come la definizione di termini comuni e basilari, le relazioni ed i vincoli tra questi e i punti di vista nell'interpretazione dei fenomeni. Ogni base di conoscenza si fonda sulla concettualizzazione del suo costruttore ed è generalmente implicita. L'ontologia è un'interpretazione della conoscenza implicita, cioè deve esplicitare tutto ciò che è stato lasciato implicito, in modo da non creare confusioni concettuali o terminologiche.
Un altro importante uso dell'ontologia è il fatto che essa deve permettere il riutilizzo della conoscenza di dominio. Supponiamo che un gruppo di ricercatori sviluppi un'ontologia di dominio, ad esempio dell'impresa; se altre persone hanno bisogno di svilupparne un'altra, sempre per lo stesso dominio, non occorre cominciare ex-novo, ma è sufficiente estendere l'ontologia già esistente. E' possibile anche riutilizzare un'ontologia generale (che comprende concetti presenti in molti domini diversi, come ad esempio quello di tempo), estendendola per descrivere il dominio di interesse.

4.5.3.2 Comunicazione tra le persone

Le ontologie permettono la comunicazione tra persone con differenti bisogni e punti di vista, che lavorano in contesti diversi; riducendo la confusione concettuale e terminologica e provvedendo a framework integrato all'interno dell'organizzazione. In questo caso può essere sufficiente un'ontologia informale, ovvero in linguaggio naturale, e non ambigua.

E' utile considerare in dettaglio alcuni aspetti dell'uso dell'ontologia che facilitano le comunicazioni tra le persone all'interno dell'organizzazione:

- *Modelli normativi*. In ogni sistema software largamente integrato, differenti persone devono avere una conoscenza condivisa di tale sistema e dei suoi obiettivi. Usando l'ontologia, è possibile costruire un modello normativo del sistema, fornendo ad esso una semantica.
- *Network di relazioni*. E' possibile usare le ontologie per creare una rete di relazioni. Una rete è implicita in un sistema, ma le persone spesso hanno diverse prospettive e usano assunzioni differenti. Di conseguenza manca una conoscenza condivisa della natura delle relazioni all'interno del sistema. Le ontologie servono a esplicitare tutte le assunzioni, attraverso l'identificazione delle connessioni logiche tra i vari elementi del sistema.
- *Consistenza e mancanza di ambiguità*. Uno dei ruoli più importanti che l'ontologia gioca nella comunicazione, è provvedere a definizioni non ambigue dei termini usati in un sistema software. Ogni set di tools software deve garantire la consistenza tra i vari tools stessi e le ontologie. Un'ontologia usata dall'utente è differente da una che supporta i tool; di conseguenza occorre identificare le assunzioni rilevanti usate da diverse persone, tools o ontologie, catturare vari sinonimi e utilizzarli nella traslazione al diverso pubblico.
- *Integrazione di prospettive utenti differenti*. Se si ha un sistema con molti agenti che comunicano, diventa vitale l'integrazione di prospettive diverse. Ad es., le persone che occupano posizioni differenti nell'organizzazione possono avere diverse visioni di che cosa fa l'organizzazione, quali sono gli obiettivi da raggiungere, e come raggiungerli. C'è anche il problema di integrare le viste locali e globali del sistema. Usando un'ontologia per provvedere ad un modello normativo del sistema, l'integrazione può essere ottenuta assistendo i partecipanti nella comunicazione e venendo ad un punto di incontro.

4.5.3.3 Interoperabilità

Molte applicazioni di ontologie si occupano del problema dell'interoperabilità, in cui vi sono utenti diversi, che hanno bisogno di scambiare i dati, o che utilizzano diversi tools

software. Uno dei maggiori temi per l'utilizzo delle ontologie in un dominio, come in un modello dell'impresa o un'architettura multiagente, è la creazione di un ambiente integrato per diversi tools software.

Supponiamo, ad esempio, che diversi siti web contengano informazioni riguardo i termini usati in un'impresa. Se tali siti condividono la stessa ontologia, allora diversi tools possono estrarre e aggregare informazioni da questi siti differenti. I tools possono usare queste informazioni aggregate per rispondere alle domande degli utenti o come dati in input ad altre applicazioni.

4.5.3.3.1 Ontologie come Inter-Lingua

Ogni ambiente IT per il business process reengineering o sistema multiagente dovrebbe usare modelli di impresa integrati, abbracciando le attività, le risorse, le organizzazioni, gli obiettivi, i prodotti e i servizi. Questi modelli di impresa integrati servono come un deposito comune, accessibile da molteplici set di tools. Essi servono anche ad integrare i depositi di dati esistenti, standardizzando la terminologia tra gli utenti diversi o provvedendo a delle fondamenta semantiche per le traslazioni tra i differenti utenti.

Per aiutare l'interoperabilità, le ontologie possono essere usate come supporto alla traslazione tra diversi tools software, linguaggi, paradigmi e metodi di modellazione. Un approccio possibile è progettare un unico traslatore per ogni due parti interessate allo scambio (Figura 4.3); tale approccio, però, richiede $O(n^2)$ traslatori per n ontologie differenti. Usando invece le ontologie come "inter-lingua", cioè come un formato di interscambio, per supportare la traslazione, è possibile ridurre il numero di traslatori a O(n) per n differenti ontologie, perché sono necessari solo i traslatori dall'ontologia originaria all'ontologia di interscambio (Figura 4.4).

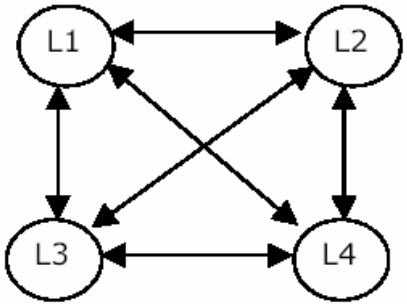


Figura 4.3 Unico traslatore per ogni due parti interessate allo scambio

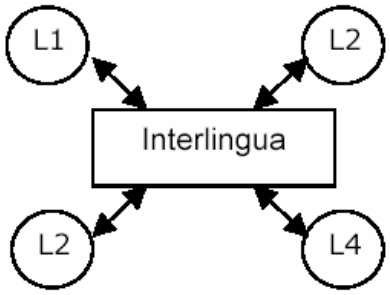


Figura 4.4 per traslare dal linguaggio Li a Lj e viceversa è richiesto un traslatore tra Li e l'inter-lingua ed un altro tra l'inter-lingua a Lj

4.5.3.3.2 *Dimensioni dell'interoperabilità*

Vi sono varie dimensioni dell'interoperabilità:

- *Interoperabilità interna*. In questo caso, tutti i sistemi che richiedono interoperabilità sono sotto il controllo diretto della stessa unità organizzativa.
- *Interoperabilità esterna*. Vi è un'unità organizzativa che desidera isolare se stessa dai cambiamenti imposti su di essa dall'esterno, dove "esterno" può significare un altro dipartimento nella stessa organizzazione.
- *Ontologie integrate tra i domini*. Si tratta di integrare le ontologie provenienti da diversi domini in modo da supportare alcuni lavori. Ad es., un'ontologia che supporta i sistemi di workflow management ha bisogno di integrare le ontologie che riguardano i processi, le risorse, i prodotti, i servizi e le organizzazioni.
- *Integrazione di ontologie tra i tools.* Tools differenti, che operano nello stesso dominio, possono usare ontologie diverse; per ottenere l'interoperabilità tra i vari tools, è necessaria un'ontologia comune, che deriva dall'integrazione delle varie ontologie, che tutti i tools possano utilizzare.

4.5.3.4 System Engineering

Le ontologie svolgono un ruolo importante nella progettazione e nello sviluppo dei sistemi software, nel migliorare la loro affidabilità e il loro riutilizzo e nel velocizzare

l'acquisizione di conoscenza da parte di tali sistemi software. Quanto detto è stato descritto più in dettaglio nel seguito.

4.5.3.4.1 Specificazione

L'ontologia aiuta il processo di identificazione dei requisiti e la definizione delle specifiche per un sistema software. Il ruolo che l'ontologia gioca nelle specifiche, varia con il grado di formalità e automazione all'interno della metodologia di progettazione del sistema.

In un approccio informale, le ontologie facilitano il processo di identificazione dei requisiti del sistema e la comprensione delle relazioni tra i vari componenti del sistema stesso. Questo è molto importante per i sistemi che riguardano i team distribuiti di progettisti che lavorano in domini differenti.

In un approccio formale, un'ontologia provvede ad una specifica dichiarativa di un sistema software, che permette di capire per che cosa il sistema è progettato, piuttosto che come il sistema supporta questa funzionalità.

4.5.3.4.2 Affidabilità

Le ontologie informali, ovvero in linguaggio naturale, possono migliorare l'affidabilità dei sistemi software, fungendo da base per il controllo manuale della progettazione, in base alle specifiche richieste. Una rappresentazione formale, mediante assiomi e teoremi, rende possibile la consistenza dell'ontologia, attraverso un controllo dei risultati mediante software affidabili. In aggiunta, le ontologie formali possono essere usate per rendere esplicite le varie assunzioni fatte da componenti differenti di un sistema software, facilitando la loro integrazione.

4.5.3.4.3 Riusabilità

Per essere effettive, le ontologie devono supportare la riusabilità, in modo che possano importare ed esportare moduli diversi tra sistemi software diversi. Il problema è che quando i tools software vengono applicati a un nuovo dominio, le performance che si ottengono non sono quelle che ci si aspettava, poiché tali tools fanno affidamento su assunzioni soddisfatte nel dominio originale, ma non nel nuovo.

Le ontologie provvedono ad un framework per determinare quali aspetti di un'ontologia sono riutilizzabili tra differenti domini e classi.

Le ontologie provvedono anche ad una libreria di entità, attributi, processi e loro relazioni nel dominio di interesse, che è facile da riutilizzare.
L'obiettivo ultimo di tale approccio è la costruzione di una libreria di ontologie che può essere riutilizzata e adattata a diverse classi di problemi e a diversi ambienti. Per essere utili, le ontologie in tale libreria, devono poter essere personalizzate, a seconda delle classi di problemi e delle classi di utenti, siano essi manager, consulenti o ingegneri. Inoltre, tali ontologie devono poter essere estendibili, permettendo l'incorporamento di nuove classi di regole e la specializzazione di concetti e di regole per un particolare problema.

4.5.3.4.4 Acquisizione di conoscenza

La velocità e l'affidabilità dei sistemi software può essere incrementata utilizzando l'ontologia esistente, come punto di partenza e base per guidare l'acquisizione della conoscenza, quando vengono costruiti sistemi knowledge-based.

4.5.3.5 Supporto al Web Semantico

Un altro importante uso dell'ontologia è il supporto che essa fornisce alla realizzazione del cosiddetto Web semantico.
Allo stato attuale, il Web contiene più di trecento milioni di oggetti che forniscono un'ampia varietà di informazioni. Trovare ed estrarre una determinata informazione dal web è già da anni un problema al centro dell'attenzione degli studiosi. Tale problema diventerà sempre più difficile se la crescita del web avverrà con la rapidità prevista dal W3C (il comitato di standardizzazione del web).
L'intelligenza artificiale ha una tradizione consolidata nello sviluppo di metodi, strumenti e linguaggi atti a strutturare la conoscenza e l'informazione in generale. Un *agente intelligente* è un programma che filtra le informazioni in base agli interessi dell'utente, in modo automatico, magari periodicamente.
Sembra quindi naturale applicare tali tecniche di intelligenza artificiale per affrontare i problemi di ricerca ed estrazione delle informazioni sul web.
Sfortunatamente non è conveniente applicare le tecniche dell'IA direttamente a documenti semistrutturati scritti in linguaggio naturale.

All'atto pratico, l'impiego di strumenti per il ragionamento automatico è possibile quando si ha a disposizione una rappresentazione della semantica dei documenti che sia *machine-processable* ("processabile dalla macchina").

Il WWW fu proposto con l'idea che esso non dovesse essere solo un mezzo di comunicazione tra umani, ma anche tra macchine.

Sfortunatamente, la seconda di queste speranze non è ancora realizzata, con il risultato frustrante che un'enorme mole di dati disponibile all'uomo non può però essere analizzata e rielaborata dalle macchine. Proprio affinché tali dati possano essere utilizzati direttamente dalle macchine, è nata l'esigenza di pensare alla terza generazione del web, ossia al *Web Semantico*. L'idea del Web semantico è fornire sufficiente flessibilità per rappresentare tutte le basi di dati e le regole logiche per collegarle insieme, con un consistente valore aggiunto, in modo da passare dall'informazione "*machine-readable*" all'informazione "*machine-processable*", cioè rendere le informazioni direttamente utilizzabili dagli elaboratori, e realizzare un motore di ricerca semantico, facendo in modo che la ricerca dei documenti non sia basata sulle semplici parole chiave, ma sul significato di un concetto o di più concetti legati tra loro.

Per avere a disposizione un Web semantico, i computer devono poter accedere a una collezione strutturata di informazioni e determinare poi delle regole di inferenza da utilizzare per effettuare dei ragionamenti automatici. Due database possono usare due diversi identificatori per riferirsi allo stesso concetto, allora un programma che vuole confrontare o combinare informazioni provenienti da questi due database deve essere in grado di sapere che questi due termini vengono utilizzati per attribuire lo stesso significato a una determinata cosa.

La soluzione al problema è fornita proprio dall'ontologia, usata come base per strutturare semanticamente e indicizzare un deposito di informazioni; essa può essere pensata come un documento o un file, che definisce le relazioni tra i termini. La forma più tipica di ontologia per il Web ha una tassonomia e una serie di regole di inferenza; la tassonomia definisce sia le classi di oggetti sia le relazioni *is-a* tra tali classi. Le regole di inferenza associate alle ontologie forniscono a Internet una notevole potenza. Ad es. attraverso di esse, un programma è in grado di capire immediatamente che l'indirizzo della Cornell University, che ha sede a Ithaca, deve essere all'interno dello stato di New York, che a sua volta è in America e per questo dovrebbe essere formattato seguendo gli standard americani.

Le ontologie possono migliorare le funzionalità di Internet in molti modi, ad esempio incrementando l'accuratezza delle ricerche: il programma di ricerca potrebbe guardare solo quelle pagine che si riferiscono ad un preciso concetto, invece che tutte quelle che usano keyword ambigue.
Applicazioni più avanzate potrebbero utilizzare le ontologie per collegare l'informazione presente in una pagina con la relativa base di conoscenza e/o con le regole di inferenza, dando luogo ai cosiddetti agenti intelligenti.

4.6 Metodologie di sviluppo delle ontologie

Sebbene siano state fatte molte esperienze collettive sullo sviluppo e l'uso delle ontologie non esiste ancora una metodologia standard per costruire un'ontologia. La scelta dipende soprattutto dallo scopo per cui l'ontologia viene costruita. Infatti, non c'è un modo corretto per modellare un dominio: la soluzione migliore è legata alla singola applicazione; in secondo luogo, lo sviluppo di un'ontologia è un processo iterativo; infine, nell'ontologia i concetti dovrebbero essere strettamente legati agli oggetti (fisici o logici) e alle loro relazioni nel dominio d'interesse.
Alcune delle metodologie per lo sviluppo delle ontologie sono riportate in (Tabella 4.1).

Tabella 4. 1 – Metodologie per costruire le ontologie

Methodologies for building ontologies from the scratch
TOVE Methodology
Enterprise Methodology
Sensus methodology
Bernaras, Laresgoiti, Corera Methodology
Methontology
CyC methodology (www.cyc.com)
Uschold and King
Gruninger and Fox
Kactus methodology
Onto-To-Knowledge Methodology (http://www.ontoknowledge.org/)

Tali metodologie sono delle linee guida e definiscono la sequenza di passi da seguire per la progettazione di un'ontologia senza tuttavia specificare nel dettaglio come eseguire ogni singolo passo. Quasi tutti i passi, infatti, sono difficilmente formalizzabili ed in ognuno entra in gioco la creatività, l'esperienza e la conoscenza del dominio.

Come detto in precedenza non esiste un'unica metodologia per costruire un'ontologia. Tuttavia di seguito viene illustrata la metodologia proposta da Mike Uschold e da Martin King opportunamente ampliata in alcune fasi; tale metodologia si basa sull'esperienza di sviluppo dell'Enterprise Ontology, ed è costituita dalle seguenti fasi (Uschold & King 1995) (Figura 4.5):

1. Studio di fattibilità
2. Identificazione del dominio e del fine dell'ontologia (fase di scoping)
3. Costruzione dell'ontologia
4. Valutazione
5. Mantenimento

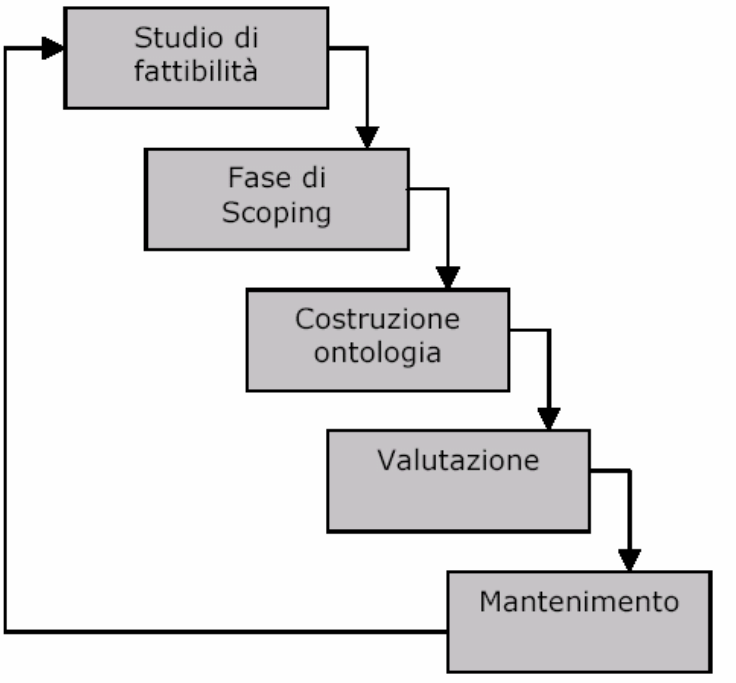


Figura 4. 5 - Fasi del processo di sviluppo di un'ontologia di dominio

Dopo una prima elaborazione, quasi certamente si avrà bisogno di rivedere e correggere l'ontologia iniziale. Questo processo di realizzazione dell'ontologia è dunque un processo iterativo.

Nella fase di progettazione dell'ontologia si possono delineare alcune linee guida, dei criteri cui attenersi affinché l'ontologia possa essere davvero d'aiuto a raggiungere lo scopo preposto. Tali criteri vengono di seguito descritti:

- *Chiarezza*. Un'ontologia dovrebbe comunicare in modo oggettivo e non ambiguo il significato dei termini, indipendentemente dal contesto per cui sono definiti. Il mezzo attraverso il quale è possibile perseguire tale fine è la formalizzazione. Ogni dichiarazione dovrebbe inoltre essere affiancata da una spiegazione in linguaggio naturale.
- *Coerenza*. Un'ontologia deve essere internamente consistente. La coerenza deve essere applicata agli assiomi di definizione e anche alle parti delle definizioni che non sono assiomatiche, così come la documentazione in linguaggio naturale e gli esempi.
- *Estendibilità*. Un'ontologia deve essere progettata in modo da essere estesa facilmente a nuovi domini, senza svilupparne una ex-novo; si dovrebbe, cioè, poter definire nuovi termini per usi speciali basati sul vocabolario esistente, in modo da non richiedere la revisione delle definizioni esistenti.
- *Indipendenza dalla codifica*. La concettualizzazione dovrebbe essere definita a livello di conoscenza senza alcuna dipendenza da un particolare linguaggio di codifica. Questo perché gli agenti e le applicazioni potrebbero essere implementate facendo riferimento a sistemi di rappresentazioni diversi.
- *Approccio.* Per scegliere quali termini definire per primi, si può procedere in tre modi differenti:
    - *top-down*: si definiscono prima i concetti più generali e poi li si specializza;
    - *bottom-up:* si definiscono prima i concetti più specifici e poi li si organizza in classi più generali*;*
    - *combinato:* si definiscono prima i concetti più salienti e poi li si generalizza e specializza*.*

Di seguito sono state analizzate in dettaglio le varie fasi del processo di sviluppo di un'ontologia di dominio.

4.6.1 Studio di fattibilità

Un qualsiasi sistema di knowledge management può funzionare in modo soddisfacente solo se è opportunamente integrato. Molti fattori, oltre a quello tecnologico, possono determinarne il successo o il fallimento. Per analizzare questi fattori si deve effettuare uno studio di fattibilità per identificare i problemi o le aree d'opportunità e le potenziali soluzioni. Tale studio di fattibilità può aiutare a determinare la fattibilità tecnica ed economica del progetto in esame, selezionare l'area su cui focalizzarsi e determinare la soluzione migliore.

4.6.2 Fase di Scoping

La seconda fase di tale metodologia viene definita *fase di scoping* (spesso si parla anche di fase di *specificazione* dell'ontologia).

Tale fase ha l'obiettivo di definire chiaramente il campo d'applicazione dell'ontologia e lo scopo per cui essa viene realizzata, in modo da essere chiari sui destinatari cui ci si rivolge.

Questo passo è di fondamentale importanza, il progettista deve dare risposte chiare e precise alle domande: "Per quale motivo e per raggiungere quale scopo sto definendo la mia ontologia? Quale sarà il suo utilizzo? Sarà un componente di un sistema più complesso? Se si quale sarà il suo ruolo? Le risposte date influenzeranno la fase successiva di costruzione dell'ontologia.

Questa fase si compone di varie sottofasi:

- Identificare e caratterizzare il range d'utenti dell'ontologia (manager, personale tecnico, programmatori, etc.).
- Consultare il range di scopi e gli usi dell'ontologia.
- Identificare i vari scenari e le competency questions, che aiutano a chiarire gli usi e i meccanismi dell'ontologia.

4.6.2.1 Scelta del grado di formalità

Dopo aver stabilito lo scopo che s'intende raggiungere con l'ontologia che si vuole realizzare, il passo successivo è di stabilirne il grado di formalità. Questo è determinato prevalentemente dallo scopo e dagli utenti dell'ontologia. Ad es., se gli utenti sono persone non tecniche e lo scopo principale è provvedere ad un *vocabolario condiviso* per facilitare la comunicazione tra le persone, allora un glossario informale può essere

sufficiente. Se, invece, l'ontologia deve supportare l'interoperabilità o il riuso e la condivisione di basi di conoscenza, allora sarà necessaria una rappresentazione più formale. In generale, il grado di formalità richiesto aumenta con il grado d'automazione dei processi che l'ontologia deve supportare. In alcuni casi, può essere richiesta sia un'ontologia informale che formale, per soddisfare gli utenti tecnici e non.

4.6.2.1.1 Cattura degli scenari d'uso

Una volta determinato lo scopo e il livello di formalità dell'ontologia, occorre identificarne la portata. Per delineare un chiaro quadro della portata dell'ontologia, è utile considerare gli scenari in cui l'ontologia verrà utilizzata.

4.6.2.1.2 Formulazione di competency questions

E' possibile usare gli scenari come base per definire un insieme completo di *competency questions*, ovvero di domande alle quali dovrebbe essere in grado di rispondere una base di conoscenza fondata sull'ontologia. Queste domande permettono di chiarire cosa deve contenere l'ontologia e cosa no. Occorre tener presente che le risposte a tali domande potrebbero cambiare durante il ciclo di vita di realizzazione dell'ontologia.

4.6.3 Costruzione dell'ontologia

La fase di costruzione di un'ontologia, che rappresenta il cuore del processo di sviluppo dell'ontologia, è costituita dalle seguenti sotto-fasi:

- *Cattura dell'ontologia*;
- *Ralizzazione della meta-ontologia*, ovvero scelta dei termini basilari (classi e relazioni) che saranno utilizzati per specificare l'ontologia;
- *Costruzione della tassonomia*, ovvero tracciamento delle relazioni *is-a* tra i vari termini del dominio, in base alle loro definizioni;
- *Scelta del linguaggio di rappresentazione dell'ontologia*, per rappresentare graficamente tutte le possibili relazioni tra i concetti dell'ontologia;
- *Scelta del linguaggio di codifica dell'ontologia*, che permetta di passare dalla rappresentazione grafica, "*human-readable*", alla scrittura del codice, "*machine-readable*";
- *Scrittura del codice*;
- *Integrazione d'ontologie esistenti*.

4.6.3.1 Cattura dell'ontologia

Questa viene chiamata anche sotto-fase di *concettualizzazione* e si articola in due steps:
1) Identificazione: identificazione dei concetti chiave e delle relazioni nel dominio d'interesse. Per procedere in questa fase, si possono utilizzare le seguenti tecniche:

- *Brainstorming:* confrontare le proprie idee con altri su un problema specifico.
- *Grouping:* strutturare i termini in base alle aree di lavoro corrispondenti, formando dei sotto-gruppi. Si possono raggruppare termini simili e potenziali sinonimi per considerazioni successive.
- *Competency Questions*: se non c'è una competency question che richiede l'utilizzo di un concetto, allora tale concetto non deve essere incluso nell'ontologia.

2) Definizione: produzione di definizioni testuali precise e non ambigue per riferirsi ai concetti e relazioni individuati nello step precedente.
Questo processo di cattura dell'ontologia è indipendente dal particolare linguaggio che viene utilizzato per la codifica.

4.6.3.2 Realizzazione della meta-ontologia

La realizzazione della *meta-ontologia* riguarda la scelta dei termini basilari (classi e relazioni) che saranno utilizzati per specificare l'ontologia.

4.6.3.3 Costruzione della tassonomia

La terza sotto-fase della fase di costruzione dell'ontologia consiste nel tracciare le relazioni *is-a* tra i vari termini del dominio, in base alle loro definizioni, ottenendo la tassonomia.
Uno dei principali ruoli delle tassonomie è dare una struttura ad un'ontologia, per facilitare la comprensione umana e permettere l'integrazione delle varie ontologie. Le ontologie si limitano a dare definizioni conservative, ovvero definizioni nel senso tradizionale di introdurre terminologia senza aggiungere conoscenza a proposito del mondo. Infatti, per mantenere un sistema "aperto", l'ontologia non è basata su una gerarchia fissa di categorie, ma su un insieme di distinzioni, a partire dalle quali la gerarchia è generata automaticamente.

4.6.3.4 Scelta del linguaggio di rappresentazione dell'ontologia: le reti semantiche

La tassonomia permette di dare una struttura all'ontologia; tuttavia, poiché essa comprende solo le relazioni *is-a* esistenti tra i vari termini, per rappresentare tutte le altre possibili relazioni (non solo quelle *is-a*) è necessario considerare altri strumenti.

Vi sono due correnti di pensiero diverse, circa il modo in cui rappresentare la conoscenza: la corrente logica e la corrente anti-logica. La corrente logica propone di utilizzare gli strumenti della logica matematica per studiare e sviluppare i linguaggi di rappresentazione della conoscenza.

La logica come strumento di rappresentazione della conoscenza, nell'ambito dell'Intelligenza Artificiale, è stata criticata da diversi punti di vista. Le rappresentazioni logiche sono poco strutturate; la conoscenza è rappresentata sotto forma di molti enunciati tra loro indipendenti. Di conseguenza, le informazioni che vertono su un dato oggetto, o concetto o evento, possono essere sparpagliate in molteplici formule diverse della base di conoscenza. Di qui l'esigenza di organizzare le rappresentazioni in un formato più consono agli scopi computazionali dell'Intelligenza Artificiale. Inoltre gli sviluppatori di ontologie non sempre conoscono la disciplina della logica matematica; di conseguenza, non sono in grado di usare la logica per rappresentare le ontologie.

Le reti semantiche sono tra i formalismi per la rappresentazione della conoscenza che sono stati sviluppati come alternativa alla logica.

4.6.3.5 Scelta del linguaggio di codifica dell'ontologia

Questa quinta sotto-fase della fase di costruzione dell'ontologia riguarda la scelta del linguaggio di codifica dell'ontologia che permetta di passare dalla rappresentazione grafica, "*human-readable*", alla scrittura del codice, "*machine-readable*". Diversi linguaggi possono essere utilizzati per codificare l'ontologia, che differiscono nella loro sintassi, terminologia, espressività e semantica.

4.6.3.6 Scrittura del codice

Dopo aver scelto il linguaggio che dovrà essere utilizzato per la codifica dell'ontologia, la fase successiva è ovviamente quella della scrittura del codice, che può essere

effettuata mediante l'aiuto di appositi tools che supportano il linguaggio di codifica scelto o attraverso il semplice notepad.

4.6.3.7 Integrazione delle ontologie esistenti

Durante la fase di cattura e di scrittura dell'ontologia, c'è il problema di come e quando utilizzare le ontologie già esistenti. In genere, si tratta di un problema molto complesso. Infatti, è abbastanza semplice identificare i sinonimi ed estenderli ad un'ontologia in cui non esistono realmente concetti. Invece, quando ci sono concetti simili definiti nelle ontologie esistenti, è, in genere, difficile capire come e quando tali concetti possono essere adattati e riutilizzati. Un modo per cercare di integrare le ontologie esistenti è quello di rendere esplicite tutte le assunzioni che sono alla base dell'ontologia.

4.6.4 Valutazione

Ci sono molti tipi di criteri che potrebbero essere usati nella valutazione di un'ontologia. Alcuni sono generali, per cui applicabili ad una qualsiasi ontologia, altri molto specifici, e quindi legati ad un particolare esempio.

Alcuni dei criteri generali sono quelli indicati nelle linee guida delineate nel paragrafo 4.5.

Gomez-Pérez (Gomez-Pérez et al.1995) fornisce una buona definizione della valutazione nel contesto della tecnologia della conoscenza condivisa:

" dare un giudizio tecnico sulle ontologie, sugli ambienti software ad esse associate, e sulla documentazione rispetto a un frame di riferimento…Il frame di riferimento può essere costituito da specifiche dei requisiti, competency questions, e/o il mondo reale."

4.6.5 Mantenimento

Le specifiche su cui si basa l'ontologia possono spesso cambiare in base ai cambiamenti del mondo reale. Per riflettere questi cambiamenti, le ontologie devono essere frequentemente aggiornate.

Il mantenimento dell'ontologia è un compito molto cruciale dal momento che devono essere rispettate sempre due importanti caratteristiche:

- *adeguatezza*, che è relativa a come l'ontologia di un dominio rappresenta correttamente la realtà e perciò riguarda la sua efficacia.

- *completezza,* che riguarda la definizione completa di tutti i concetti appartenenti all'ontologia.

In questa fase di mantenimento rientra anche la definizione dei documenti necessari per documentare l'ontologia, il tipo e lo scopo della stessa. Questa fase di documentazione è importante perché una delle principali barriere per un'effettiva condivisione della conoscenza, è l'inadeguata documentazione sulle ontologie e sulle knowledge base esistenti. Per superare tale barriera tutte le assunzioni importanti dovrebbero essere documentate, sia quelle riguardanti i concetti principali definiti nell'ontologia, sia quelle sulle primitive usate per esprimere le definizioni nell'ontologia (a cui ci si riferisce come meta-ontologia).

4.7 Rappresentazione della conoscenza di un dominio didattico.

Per rappresentare la conoscenza di un dominio didattico si utilizzano tre livelli di astrazione (Figura 4.6). A livello più basso troviamo i Learning Object che rappresentano le risorse didattiche elementari necessarie a spiegare un dato argomento del dominio. Il livello immediatamente sopra è occupato dai metadati che servono ad indicizzare le risorse e a descrivere quindi i Learning Object. A livello ancora più alto troviamo le ontologie che legano i concetti del dominio ai metadati.

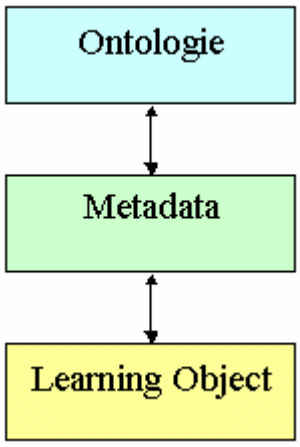


Figura 4. 6- I tre livelli della conoscenza in un dominio didattico

4.7.1 Learning Object

I **learning object** rappresentano tutte quelle risorse didattiche utilizzate nell'ambito dell'e-learning con le seguenti proprietà:

- **Condivisibile, riusabile**: deve essere possibile riutilizzare un contenuto didattico pensato per un determinato corso. Deve essere possibile condividere l'oggetto tra più utilizzatori. La riusabilità è l'elemento cardine dell'intera "economia" dei LO, in quanto consente di non reinventare contenuti già sviluppati da altri e di ottimizzare quindi gli investimenti nell'e-learning.
- **Digitale**: sebbene una delle definizioni ufficiali più accreditate (IEEE) includa nel concetto di LO anche risorse non digitali, come libri o qualunque altra risorsa didattica, una definizione operativa legata all'e-learning e basata su piattaforme tecnologiche come il WWW deve per forza di cose limitare il suo campo alle sole risorse digitali, utilizzabili direttamente da sistemi informatici.
- **Modulare**: gli oggetti di apprendimento non sono interi corsi monolitici, con un inizio ed una fine e senza possibilità di scomposizione, ma piuttosto unità di contenuti più piccole, utilizzabili in diversi contesti.
- **Interoperabile**: interoperabilità costituisce un elemento fondamentale per l'"economia" dei LO. Non ha senso creare contenuti didattici utilizzabili all'interno di una sola piattaforma tecnologica, senza possibilità di passarli ad un diverso sistema.
- **Accessibile**: nel senso di "facilmente recuperabile"

I Learning Object definiti nella nostra ontologia godono di due importanti proprietà:

- **Atomici**: per ogni concetto del dominio esiste uno ed un solo LO ad esso associato e viceversa ogni LO è associato ad un solo concetto del dominio.
- **Auto-consistenti**: se un LO è associato ad un concetto esso deve essere capace di contenere tutta la conoscenza per spiegare quel concetto.

4.7.2 Collegare l'ontologia ai Learning Object: I metadati

Per collegare l'ontologia con il materiale didattico occorre innanzi tutto definire il tipo di indicizzazione da adottare, ossia il modo in cui collegare il materiale didattico ai concetti del dominio.

Dal punto di vista della cardinalità l'indicizzazione può essere singola o multipla. L'*indicizzazione singola dei concetti* è la forma più semplice di indicizzazione in cui ad ogni concetto del dominio è associato uno ed un solo Learning Object.

Nell'*indicizzazione multipla dei concetti* la relazione tra concetti e contenuti è di tipo molti a molti per cui ad un concetto sono associati più Learning Object e viceversa.

Una volta definito il tipo di indicizzazione occorre stabilire come collegare i concetti del dominio con i Learning Object, è qui che intervengono i metadati.

I metadati possono essere associati alla risorsa che descrivono in diversi modi: possono essere memorizzati all'interno della risorsa stessa (come potrebbe avvenire utilizzando i tag META di un documento HTML) oppure mantenuti separatamente attraverso documenti XML/RDF. Questa seconda soluzione appare più consona alla natura del Web, in quanto i database dei metadati, centralizzati in server specializzati o distribuiti in una serie di nodi, possono contenere riferimenti di tipo ipertestuale alla locazione del contenuto descritto, senza difficoltà.

L'area dei metadati è probabilmente la più prolifica in materia di proposte e specifiche di standardizzazione precedente, nel 2002 l'IEEE ha emesso la specifica 1484.12.1 "Standards for Learning Object Metadata", conosciuta con la sigla LOM. In precedenza, una delle prime organizzazioni che si è occupata di metadati è stata Dublin Core, la cui proposta è rivolta alla descrizione di qualunque risorsa (quindi anche non specificamente didattica) presente sul Web.

4.7.2.1 IEEE/LOM P1484.12

L'IEEE/LOM consiste in un insieme di circa 47 elementi descrittivi, di cui molti obbligatori, suddivisi in 9 gruppi, specificamente destinati alla descrizione di risorse didattiche. Si tratta dell'unico standard emesso da un'organizzazione di "alto livello" nel settore dell'e-learning.

Le categorie in cui è suddiviso il modello LOM sono:

1. *General*: comprende informazioni generali che descrivono l'oggetto nel suo complesso. Alcuni dei descrittori di questo gruppo sono la descrizione, il titolo, il livello di aggregazione (corso, modulo, lezione).
2. *Lifecycle*: questa categoria raggruppa i descrittori relativi alle versioni del LO e allo stato attuale come numero di versione, chi ha contribuito.
3. *Meta-Metadata:* include informazioni sui metadati stessi.
4. *Technical:* in questo gruppo sono indicati i requisiti tecnici necessari per il funzionamento del LO e le caratteristiche tecniche del LO stesso come il formato, la dimensione e la dipendenza da particolari sistemi operativi.
5. *Educational*: questa categoria contiene le caratteristiche pedagogiche ed educative del LO.

6. *Rights:* sono raggruppati gli elementi che descrivono i diritti di proprietà intellettuale e le eventuali condizioni di utilizzo del LO come il costo e informazioni di copyright.
7. *Relation*: questo gruppo descrive le eventuali relazioni (del tipo "è parte di", "richiede", "si riferisce a") con altri LO. Se il LO ha diverse relazioni possono essere inserite diverse indicazioni di relazione per ognuna di esse.
8. *Annotation*: questa categoria comprende descrittori che consentono di inserire commenti sull'utilizzo educativo dei LO, inclusa l'identificazione di chi ha creato l'annotazione.
9. *Classification*: attraverso gli indicatori di questa categoria è possibile classificare il LO in relazione ad un particolare sistema di classificazione.

Una categoria molto importante è *Relation*, essa definisce il modo in cui le varie risorse sono collegate tra loro, le possibili relazioni sono definite in Tabella 4.2, attraverso queste relazioni è possibile collegare tra loro i vari Learning Object.

Relation	Descrizione
IsVersionOf	Indica se la risorsa descritta è la versione o l'adattamento di un'altra risorsa
hasVersion	Indica se la risorsa descritta ha un'altra versione o l'adattamento
isReplacedBy	Indica se la risorsa descritta è sostituita da un'altra risorsa
replaces	Indica per la risorsa descritta quale è quella che la sostituisce
isRequiredBy	Indica per la risorsa descritta quella che la richiede
Requires	Indica per la risorsa descritta quale altra risorsa richiede
isPartOf	Indica per la risorsa descritta di quale risorsa fa parte logicamente o fisicamente
hasPart	Indica per la risorsa descritta quali risorse ne fanno parte logicamente o fisicamente
isReferencedBy	Indica per la risorsa descritta quale risorsa la referenzia
references	Indica per la risorsa descritta quale risorsa referenzia

isFormatOf	Indica per la risorsa descritta quale altra risorsa ha il suo stesso contenuto ma presentata in un formato diverso
hasFormat	Indica per la risorsa descritta quale altra risorsa preesistente ha il suo stesso contenuto ma presentata in un formato diverso

Tabella 4. 2- Relation in IEEE/LOM P1484.12

4.7.3 Ontologia per il dominio didattico

Per rappresentare il dominio didattico attraverso ontologie è necessario definire un vocabolario di termini, che rappresentano i concetti del dominio, le relazioni che intercorrono tra essi e le regole sottoforma di assiomi.

4.7.3.1 I concetti del dominio

In un dominio didattico possiamo individuare i seguenti concetti:

- *Concetti generici:* possono essere considerati tali i corsi che devono essere seguiti nell'ambito di un determinato contesto didattico (scuola, università, azienda ecc..).
- *Concetti specifici:* particolarizzano i concetti generici e nel caso specifico, possono essere considerati tali gli argomenti che compongono un corso.

4.7.3.2 Relazioni tra i concetti

Le tipiche relazioni che intercorrono in un dominio didattico sono:

- *IsPastOf(x,y)*: indica che il concetto x è parte del concetto y.
- *Requires(x,y)* : è utilizzato per implementare la propedeuticità, vuol dire che x ha bisogno di y come prerequisito.

In alcuni casi può essere implementata anche una propedeuticità più lasca usando una relazione del tipo:

- *SuggestedOrder(x,y)*: per indicare che è preferibile aver appreso prima x e dopo y. Tale relazione è un vincolo solo sull'ordine di come vengono proposti i concetti del dominio ma non è necessario conoscere x se si è interessati solo a y.

Per collegare i Learning Object ai relativi concetti bisogna individuare i così detti *concetti atomici* ossia quei concetti per cui la relazione *IsPartOf(x,y)* è tale che *x=y*, per cui il concetto x non fa parte di nessun altro concetto.
Allora per ogni concetto atomico esiste una relazione che lo collega ai L.O. che lo spiegano, attraverso i metadati:

- *IsExplainedBy(c,m)*: indica che il concetto c può essere spiegato dal L.O. descritto dal metadata m.

4.7.3.3 Regole

Una volta stabilite le relazioni tra i concetti del dominio, bisogna definire le regole che si devono rispettare per una corretta interpretazione della conoscenza contenuta nell'ontologia.
La relazione *IsPastOf* impone un ordine da seguire per navigare tra i concetti dell'ontologia. Poiché l'ontologia deve rappresentare un grafo aciclico che collega i concetti del dominio, deve valere il seguente assioma:

$$\forall x, y \in D : IsPartOf(x, y) \Rightarrow \neg \exists x, y : IsPartOf(y, x)$$

La relazione *Requires(x,y)* ci impone l'ordine con cui gli argomenti devono essere appresi e questo si traduce in un certo ordine con cui saranno presentati i materiali didattici.
Possiamo definire per tale relazione una proprietà transitiva:

$$\forall x, y, z : \mathrm{Re}\, quires(x, y), \mathrm{Re}\, quires(y, z) \Rightarrow \mathrm{Re}\, quire(x, z)$$

4.8 Linguaggi di codifica delle ontologie

Come già detto, una delle sottofasi nella costruzione delle ontologie riguarda la scelta del linguaggio di codifica dell'ontologia che permetta di passare dalla rappresentazione grafica, "*human-readable*", alla scrittura del codice, "*machine-readable*".
Per fare ciò si usano i linguaggi ontologici, alcuni basati su XML altri su RDF/RDFS (Figura 4.7).

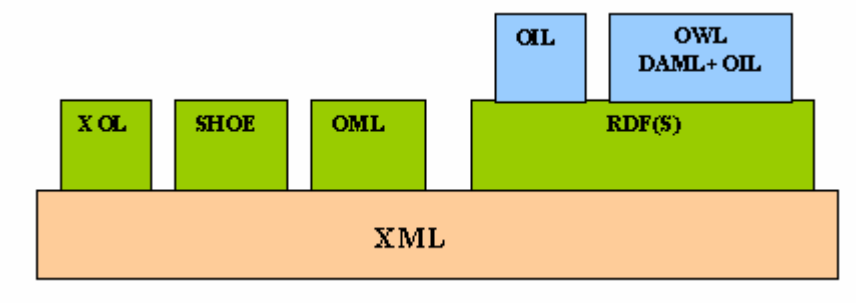


Figura 4. 7-Stack dei linguaggi

4.8.1 XML e XML Schema

XML (eXtension Markup Language) (W3C 2004) è un linguaggio di markup destinato alla descrizione di documenti, strutturati in maniera arbitraria. A differenza di HTML, che è utilizzato per descrivere la modalità con cui documenti ipertestuali con strutture fisse sono visualizzati, XML separa il contenuto del documento dalla sua modalità di visualizzazione. E' dunque uno standard elaborato per garantire l'interoperabilità sintattica, ossia per fare in modo che le informazioni non siano semplicemente formattate per facilitarne il reperimento per un utente umano, ma anche facilmente elaborate da agenti software.

XML definisce una struttura ad albero per i documenti dove ciascun nodo individua un tag ben definito (*elemento*) mediante il quale è possibile in qualche modo interpretare le informazioni che esso racchiude (*attributi*).

```
<?xml version="1.0" ?>
- <person>
    <name>Pippo</name>
  - <personal_info>
      <office_number>123</office_number>
      <phone_number>7894</phone_number>
    </personal_info>
  </person>
```

Figura 4. 8-Semplice documento XML

É possibile imporre delle restrizioni con cui i tag XML possono essere usati e quali annidamenti di tali tag sono permessi mediante l'uso di due tecnologie:

DTD (Document Type Definition): Contengono le regole che definiscono i tag usati nel documento XML, in altre parole ne definiscono la struttura. Questi possono essere dei file esterni o specificati direttamente all'interno del documento;

XML Schema (W3C 2001): Spesso i documenti condividono una struttura specifica per un certo dominio, per consentire sia al software sia agli umani di riconoscere il contenuto del XML si richiede che tali strutture siano documentate in un formato comprensibile ad entrambi. A tal scopo è stato introdotto XML Schema, il quale permette di definire un vocabolario utilizzabile per descrivere documenti XML facendo uso della loro stessa sintassi. Le sue specifiche assumono che si faccia uso di almeno due documenti, un'istanza ed uno schema. La prima contiene le informazioni che interessano realmente, mentre il secondo descrive struttura e tipo della precedente.

XML, tuttavia, non gestisce la semantica dei contenuti: essa non è specificata in modo esplicito, ma è "incorporata" nei nomi dei tag, ossia il vocabolario degli elementi e le loro combinazioni non sono prefissati, ma possono essere definiti ad hoc per ogni applicazione. Tale semantica, quindi, non è definita formalmente e può risultare eventualmente comprensibile solo all'uomo e non alla macchina: un entità software riconosce i contenuti, ma non è in grado di attribuire loro un significato.

XML, quindi, fornisce l'interoperabilità sintattica ma non riesce a garantire quella semantica né a fornire meccanismi di classificazione o ragionamento e quindi deve necessariamente essere affiancato ad altri linguaggi più potenti.

4.8.2 RDF e RDF Schema

RDF (Resource Description Framework) (W3C RDF) è lo standard che consente l'aggiunta di semantica a un documento XML e quindi si pone ad un livello direttamente superiore rispetto ad esso. Esso è in un certo senso un'applicazione di XML: se XML è un'estensione del documento, RDF può essere visto come un'estensione dei dati introdotti da XML.

Il modello base dei dati RDF è composto da:

- *Risorse*: con questo termine si intende qualsiasi cosa possa essere descritta. Una risorsa può essere ad esempio una pagina Web oppure un qualsiasi

oggetto anche se non direttamente accessibile via Web (ad esempio un libro, una persona). Una risorsa viene identificata univocamente attraverso un URI.

- *Proprietà*: una proprietà è una caratteristica, una relazione che descrive una risorsa. Il significato, l'insieme di valori che può assumere, i tipi di risorse a cui può riferirsi sono tutte informazioni reperibili dallo schema RDF in cui essa è definita.
- *Asserzioni*: una asserzione è costituita da un soggetto (la risorsa descritta) , un predicato (la proprietà) e un oggetto (il valore attribuito alla proprietà), dove l'oggetto può essere una semplice stringa o un'altra asserzione.

Un semplice esempio di utilizzo del modello di RDF può essere fornito dalla seguente asserzione: " Pippo's phone_number is 123 ", che può essere schematizzata graficamente come in Figura 4.9:

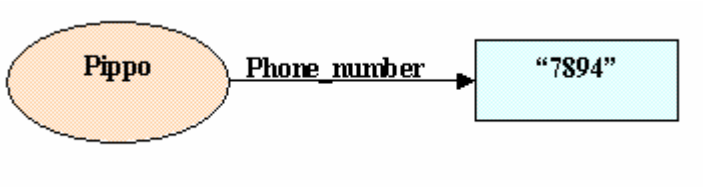


Figura 4. 9- Semplice esempio di un'asserzione RDF

La precedente asserzione è formalizzata in RDF/XML come in Figura 4.10:

```
<?xml version="1.0" ?>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:s="http://www.example.com/props/"
    xmlns:base="http://www.organization.com/people">
  - <rdf:Description rdf:about="Pippo">
      <s:Phone_number>7894</s:Phone_number>
    </rdf:Description>
  </rdf:RDF>
```

Figura 4. 10-Un semplice documento RDF/XML

Sintatticamente, i concetti espressi con RDF vengono serializzati mediante XML. RDF definisce, quindi, un semplice modello dei dati per descrivere proprietà e relazioni fra

risorse. In RDF però non esistono livelli di astrazione: ci sono le risorse e le loro relazioni, tutte organizzate in un grafo piatto. In altri termini non è possibile definire tipi (o classi) di risorse con loro proprietà specifiche.

Vista l'utilità di poter definire classi di risorse, RDF è stato arricchito con un semplice sistema di tipi detto **RDF Schema**. Il sistema di tipi RDF Schema ricorda i sistemi di tipi dei linguaggi di programmazione object-oriented (tipo Java). Una risorsa può, per esempio, essere definita come istanza di una classe (o di più classi) e le classi possono essere organizzate in modo gerarchico.

RDF Schema utilizza il modello RDF stesso per definire il sistema di tipi RDF, fornendo un insieme di risorse e proprietà predefinite che possono essere utilizzate per definire classi e proprietà a livello utente. E' possibile, inoltre, definire vincoli di dominio e di range sulle proprietà ed alcuni tipi di relazioni (comprese quelli di sottoclasse di una risorsa e sottotipo di una proprietà). L'insieme di tali elementi è detto vocabolario dell'RDF Schema.

Il linguaggio di specifica RDFS è un linguaggio dichiarativo poco espressivo ma molto semplice da implementare. Si esprime attraverso la sintassi di serializzazione RDF/XML e si avvale del meccanismo dei namespace XML per la formulazione delle URI che identificano in modo univoco le risorse (definite nello schema stesso) consentendo il riutilizzo di termini definiti in altri schemi. Concetti e proprietà già dichiarati per un dominio possono essere impiegati di nuovo o precisati per incontrare le esigenze di una particolare comunità di utenti.

Sebbene, in prima battuta, RDF e RDF Schema sembrano essere un buon strumento per la definizione di un linguaggio di markup per il Web Semantico (ad esempio, per determinare le relazioni semantiche tra termini differenti), in realtà essi mostrano di non avere sufficiente potere espressivo: non consentono, infatti, di specificare le proprietà delle proprietà, le condizioni necessarie e sufficienti per l'appartenenza alle classi e gli unici vincoli che si possono definire sono quelli di dominio e range delle proprietà.

Il loro utilizzo nei sistemi di rappresentazione della conoscenza è limitato, inoltre, da un'altra caratteristica: non permettono di specificare meccanismi di ragionamento, ma rappresentano semplicemente un sistema a frame. I meccanismi di ragionamento devono essere costruiti, quindi, ad un livello superiore.

4.8.3 DAML+OIL

Il linguaggio DAML+OIL (DARPA Agent Markup Language) è uno standard che consente la rappresentazione delle informazioni in modo che il loro significato sia comprensibile alle macchine. I due risultati più interessanti degli ultimi anni in fatto di sviluppo verso il Web Semantico, DAML-ONT e OIL, si sono fusi in un unico linguaggio, DAML+OIL appunto, che incorpora caratteristiche provenienti dal lavoro del gruppo americano (DARPA) e del gruppo europeo (progetto On-To-Knowledge, IST).

DAML+OIL riesce a garantire l'interoperabilità sintattica e semantica sfruttando la sintassi e le caratteristiche dei linguaggi che si trovano ai livelli più bassi dello stack del WEB Semantico, ossia XML e RDF, ed in più aggiunge, rispetto a questi, una maggiore forza espressiva ottenuta a partire da primitive di modellazione ereditate dai sistemi basati su Frame e la capacità di produrre asserzioni in logica formale ed effettuare ragionamenti in modo automatico, traendo spunto dalle Description Logics.

4.8.3.1 Sistemi basati su Frame

I sistemi basati su Frame somigliano agli approcci orientati agli oggetti, ma considerano le cose da un punto di vista differente. Le primitive di modellazione sono le classi (o frame), ognuna delle quali ha determinate proprietà (o slot), chiamate attributi. Essi non hanno una visibilità globale, ma sono applicabili alle sole classi per cui sono definiti. Un frame fornisce un certo contesto per la modellazione di un aspetto del dominio in esame.

Dai sistemi basasti su Frame, DAML+OIL eredita le primitive di modellazione essenziali. DAML+OIL, infatti, è basato sulla nozione di classe, la definizione delle sue proprietà e delle sue superclassi e sottoclassi. Le relazioni possono essere definite come entità indipendenti aventi un certo dominio e intervallo. Come le classi, anche le relazioni possono essere organizzate in una gerarchia.

Ad incrementare il potere espressivo contribuiscono, poi, due aspetti: i costruttori e i tipi di assiomi. I costruttori permettono la creazione, oltre che di classi intese nel senso classico, con un nome ed associate ad un URI, anche di classi intese come espressioni, cioè come il prodotto di un certo numero di operatori. Una classe può essere ottenuta dall'unione o dall'intersezione di altre classi, elencandone gli elementi, può essere complementare di un'altra, può essere definita come collezione degli elementi che

hanno almeno o al più un certo numero di valori distinti di una proprietà. Il secondo aspetto è rappresentato dagli assiomi, che permettono la dichiarazione di relazioni di classificazione o equivalenza tra classi e/o proprietà, la disgiunzione tra classi, l'equivalenza o meno di oggetti diversi, le proprietà delle proprietà.

4.8.3.2 Description Logics

Le Description Logics rappresentano una classe di logiche che sono specificatamente designate a modellare vocabolari. Esse descrivono la conoscenza in termini di concetti (paragonabili ai frame) e restrizioni (paragonabili agli slot) che sono utilizzate per derivare automaticamente classificazioni tassonomiche.
DAML+OIL eredita dalle Description Logics la semantica formale e il supporto per il ragionamento, stabilendo una corrispondenza tra concetti logici e tag DAML+OIL che li rappresentano. A partire da un documento DAML+OIL è possibile, infatti, creare una Knowledge Base: una KB è un database con capacità di ragionare in maniera automatica, in grado non solo di rispondere a query grazie a dei match ma anche effettuando ragionamenti.

4.8.4 OWL

OWL (*Web Ontology Language*) è lo standard attualmente proposto dal W3C per la definizione di ontologie per il web semantico (W3C 2004). Sviluppato a partire da DAML+OIL e come esso estende RDF ed RDF Schema e usa la sintassi XML.

OWL prevede tre livelli di complessità crescente:

- *OWL Lite*, è un sottoinsieme di OWL DL che è facilmente utilizzabile ed implementabile. Aggiunge a RDF Schema molte funzionalità utili per supportare le applicazioni web. Comprende molte delle caratteristiche più usate di OWL ed è destinato agli sviluppatori che vogliono utilizzare OWL ma vogliono iniziare con un set relativamente semplice di caratteristiche del linguaggio;
- *OWL DL*, inserisce alcuni vincoli sul modo di combinare OWL con RDF Schema. E' destinato agli utenti che vogliono la massima espressività senza perdere l'efficienza e la completezza computazionali, e i benefici dei sistemi che ragionano. E' stato realizzato per supportare le Description Logics esistenti.

- *OWL Full*, consente di combinare OWL con RDF e RDF Schema. E' destinato agli utenti che vogliono la massima espressività e la libertà sintattica di RDF senza alcuna garanzia computazionale. Il suo vantaggio è che è pienamente compatibile con RDF, sia sintatticamente che semanticamente, cioè qualsiasi documento RDF legale è anche un documento OWL Full legale.

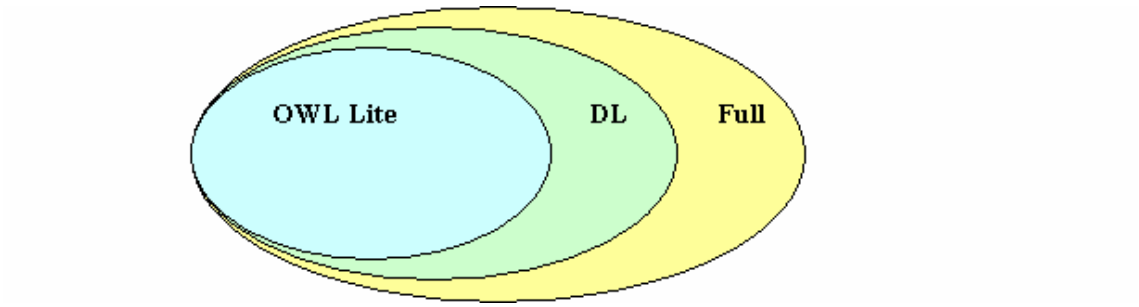


Figura 4. 11-Sottolinguaggi OWL

4.8.4.1 **Descrizioni di classi**

Nel linguaggio OWL i termini sono denominati *descrizioni di classi*, gli operatori per la definizione di termini sono denominati *costruttori di classi* e i ruoli sono denominati *proprietà*. OWL prevede sei tipi di descrizioni di classi:

- identificatore
- enumerazione
- restrizione di proprietà
- intersezione
- unione
- complemento.

Identificatore

Ogni descrizione di classe descrive una risorsa di tipo `owl:Class`. Nel caso più semplice la descrizione consta di un *identificatore* della classe (un URI). OWL prevede due identificatori predefiniti per la *classe universale* e la *classe vuota*:

- `owl:Thing`, identifica la radice, ogni classe definita in OWL è sottoclasse di Thing;

- `owl:Nothing`, definisce una classe vuota.

Ogni individuo (a_1, ... , a_n, ...) è istanza della classe `owl:Thing`.

Enumerazione

Una classe finita può essere descritta dall'*enumerazione* di (`owl:oneOf`) tutti gli individui che le appartengono:

$\{a_1,...,a_n\}$.

Ogni nominale (a_1, ... , a_n, ...) va interpretato come un URI.

Restrizioni di proprietà

In OWL sono dette *restrizioni di proprietà* le descrizioni di classi corrispondenti ai termini $\exists R.C$, $\forall R.C$, $\forall R.\{a\}$, $\leq nR$, $\geq nR$ e $=nR$. Più precisamente:

- la restrizione $\exists R.C$ è denominata `owl:someValuesFrom`;
- la restrizione $\forall R.C$ è denominata `owl:allValuesFrom`;
- la restrizione $\forall R.\{a\}$ è denominata `owl:hasValue`;
- la restrizione $\leq nR$ è denominata `owl:maxCardinality`;
- la restrizione $\geq nR$ è denominata `owl:minCardinality`;
- la restrizione $=nR$ è denominata `owl:cardinality`;

Intersezione

Una classe può essere descritta come *intersezione* di (`owl:intersectionOf`) un numero finito di classi:

$C_1 \cap ... \cap C_n$.

Unione

Una classe può essere descritta come *unione* di (`owl:unionOf`) un numero finito di classi:

$C_1 \cup ... \cup C_n$.

Complemento

Una classe può essere descritta come *complemento* di (`owl:complementOf`) un'altra classe:

$\neg C$.

4.8.4.2 Assiomi di classe

OWL prevede tre tipi distinti di assiomi di classe:

- assiomi di sottoclasse
- assiomi di equivalenza
- assiomi di disgiunzione.

Sottoclassi

Fra due descrizioni di classi, *C* e *D*, si può definire una relazione di sottoclasse (`rdfs:subClassOf`). Questa relazione, direttamente importata da RDFS, corrisponde alla sussunzione terminologica nelle DL:

$C \subseteq D$.

Equivalenza

Fra due descrizioni di classi, *C* e *D*, si può definire una relazione di equivalenza (`owl:equivalentClass`), che corrisponde all'equivalenza terminologica nelle DL:

$C \equiv D$.

Disgiunzione

Fra due descrizioni di classi, *C* e *D*, si può dichiarare una relazione di disgiunzione (`owl:disjointWith`), che corrisponde all'equivalenza:

$C \cap D \equiv \bot$.

4.8.4.3 Le proprietà

Coerentemente con quanto previsto da RDFS, in OWL anche le *proprietà* possono essere visti come particolari classi. Come tali le proprietà possono avere sottoproprietà ed essere combinate con vari costruttori.

Così come ogni classe di individui è una risorsa di tipo `owl:Class`, tutte le proprietà sono risorse di tipo `rdf:Property`. In OWL le proprietà possono essere risorse di tipo `owl:ObjectProperty` (*proprietà di individui*, cioè fra elementi di classi OWL)

oppure `owl:DatatypeProperty` (*proprietà di dati* appartenenti a tipi di dati RDFS).

In OWL si possono specificare i seguenti aspetti relativi alle proprietà:

- identificatore di proprietà
- dominio e codominio (*range*)
- sottoproprietà
- proprietà equivalente
- proprietà inversa.

Identificatore di proprietà

Un *identificatore di proprietà* corrisponde a un ruolo e sarà quindi indicato con *R* o *S*.

Dominio e codominio (range)

Di una proprietà possono essere specificati il *dominio* (`rdfs:domain`) e il *codominio* (`rdfs:range`).

Sottoproprietà

Una proprietà può essere definita come *sottoproprietà* di un'altra proprietà (`rdfs:subPropertyOf`).

Proprietà equivalente

Una proprietà può essere definita come *equivalente* a un'altra proprietà (`owl:equivalentProperty`).

Proprietà inversa

Data una proprietà *R* si può definire la *proprietà inversa* (`owl:inverseOf`).

Sulle proprietà è possibile definire altre due proprietà, ovvero:

- funzionalità
- funzionalità inversa.

Proprietà Funzionale

Se una proprietà P è definita *funzionale* (`owl:FunctionalProperty`) allora:

$$\forall x, y, z \in D : P(x, y), P(x, z) \Rightarrow x = z$$

Proprietà di Funzionalità inversa

Se una proprietà P è definita *funzionale inversa* (`owl:InverseFunctionalProperty`) allora:

$$\forall x, y, z \in D : P(x, y), P(z, y) \Rightarrow x = z$$

Infine, è possibile specificare alcune caratteristiche formali delle proprietà, e in particolare:

- simmetria
- transitività.

Simmetria

Una proprietà P è *simmetrica* se

$$\forall x, y \in D : P(x, y) \Rightarrow P(y, x)$$

In OWL è possibile dichiarare direttamente che una proprietà è simmetrica (`owl:SymmetricProperty`).

Transitività

Una proprietà P è *transitiva* se

$$\forall x, y, z \in D : P(x, y), P(y, z) \Rightarrow P(x, z)$$

In OWL è possibile dichiarare direttamente che una proprietà è transitiva (`owl:TransitiveProperty`).

4.8.4.4 Identità

Il linguaggio OWL non assume che gli individui abbiano nome unico. Quindi è possibile asserire che due nomi fanno riferimento allo stesso individuo (`owl:sameAs`):

$a = b$.

Analogamente è possibile asserire che due nomi fanno riferimento ad individui distinti (`owl:differentFrom`):

$a \neq b$.

È anche possibile asserire che n individui sono tutti distinti fra loro (`owl:AllDifferent`).

Capitolo V

5 Un caso applicativo: modellazione del dominio “Protocollo Informatico”

La scelta del dominio è un aspetto delicato in quanto per poter verificare che le assunzioni siano valide sarà importante avere a disposizione sia una base di contenuti grezzi, sia degli esperti del dominio (subject matter expert) per definire un modello del dominio verosimile (anche se ridotto) ed una mappa delle competenze, sia degli utenti sperimentatori.
La scelta del Protocollo Informatico è dovuta al fatto che Italdata ha sviluppato una soluzione di protocollo informatico da cui sarà possibile estrapolare i contenuti di base e avvalersi di suggerimenti di un esperto del dominio..

5.1 Il protocollo informatico

Per protocollo informatico s’intende “l’insieme delle risorse di calcolo, degli apparati, delle reti di comunicazione e delle procedure informatiche utilizzati dalle amministrazioni per la gestione dei documenti”, ovvero, tutte le risorse tecnologiche necessarie alla realizzazione di un sistema automatico per la gestione elettronica dei flussi documentali.
Ogni sistema di protocollo informatico, che si intende adottare o realizzare, deve ottemperare a specifiche indicazioni, riportate nel Testo Unico (DPR 445/2000).
Per una corretta gestione dei documenti, è necessario che il personale impiegato nella Pubblica Amministrazione che adotta il protocollo, conosca le fasi operative del sistema dalla tenuta del protocollo informatico, alla gestione dei flussi documentali e degli archivi.
Questo ha spinto a pensare di utilizzare il dominio del protocollo informatico come caso d’uso, e costruire l’ontologia per tale dominio didattico seguendo i passi fondamentali

della metodologia per la costruzione dell’ontologia. Uno dei passi della metodologia è la codifica dell’ontologia nel linguaggio scelto, in questa fase è stato utilizzato un tool per modellare l’ontologia che permette di costruire facilmente l’ontologia e codificarla automaticamente nel linguaggio ontologico scelto OWL. Vengono presentati in seguito alcuni tool esistenti e i motivi che hanno spinto alla scelta di Protégé.

5.2 Tool per modellare ontologie

Il processo di modellazione delle ontologie non è un processo molto semplice, spesso è conveniente farsi aiutare da un tool. Vediamo allora quali sono i principali tool esistenti, e quali sono stati i motivi che hanno spinto ad utilizzare Protégé per modellare la nostra ontologia di dominio.

5.2.1 Protégé

Protégé è un tool sviluppato presso l’Università di Stanford (Protégé 2000), che si preoccupa di fornire un editor grafico e interattivo, per l’ontology design e per l’acquisizione della conoscenza. Esso vuole essere di aiuto agli ingegneri della conoscenza ed agli esperti di dominio per realizzare obiettivi di knowldge-management. I progettisti di ontologie possono accedere alle informazioni velocemente ogni qual volta ne hanno bisogno e possono modificare ed interagire con l’ontologia direttamente navigando l’albero che la rappresenta. Il modello di conoscenza utilizzato da Protégé è compatibile con il protocollo OKBC (Open Knowledge Base Connectivity) per l’accesso alla base di conoscenza, ed uno dei maggiori vantaggi derivanti dalla sua architettura, è che garantisce uno strumento aperto e modulare. I programmatori, grazie a ciò, possono aggiungere nuove funzionalità semplicemente creando gli appropriati plug-ins. Le librerie di Protégé contengono i plug-ins per la visualizzazione grafica, per i motori d’inferenza basati sulla conoscenza per verificare i vincoli della logica del primo ordine e per l’acquisizione di informazione da risorse remote come UMLS, WordNet e molti altri.

Uno dei plug-in sicuramente più interessanti è quello su OWL. Protégé 2000, infatti offre la possibilità di creare una struttura ontologica in formato OWL, che attualmente è il linguaggio standard per la codifica di ontologie.

Visivamente il tool è organizzato secondo dei pannelli etichettati che distinguono le differenti funzionalità che sono svolte dal programma.

5.2.2 OntoEdit

OntoEdit è un tool grafico per lo sviluppo ed il mantenimento di ontologie, realizzato dalla ontopriseGmbH e realizzato a partire dall’architettura progettata da Siegfried Handschuh all’Università di Karlsrhue. Il tool si basa su un potente modello interno di ontologia. Questo paradigma supporta la modellazione della rappresentazione del linguaggio in modo più neutrale possibile per concetti relazioni e assiomi. Il tool permette la definizione di una gerarchia di concetti o classi, i quali possono essere astratti o concreti, inoltre, per ogni concetto, si possono definire dei sinonimi. Anche con questo tool le funzionalità vengono gestite attraverso un pannello etichettato in cui vengono definite le varie funzionalità.

5.2.3 OilEd

OilEd è un semplice editor che permette ai suoi utenti di progettare ontologie utilizzando il linguaggio DAML+OIL. Il tool è stato progettato ed implementato presso l’Università di Manchester. Il linguaggio utilizzato rappresenta la prima grossa differenza rispetto ai tool descritti in precedenza anche se il formato DAML racchiude al suo interno alcuni tag di RDF. Una particolarità che lo accomuna con gli altri due tool è l’utilizzo lo stesso parser per aprire e scrivere file, quello realizzato da Sergey Melnik e chiamato RDF API.

Come tutti i tool visti fino ad ora anche questo, graficamente, si basa su un pannello etichettato che serve per separare le diverse funzionalità

5.2.4 DOE

DOE (Differential Ontology Editor) è un editor di ontologie che si basa sulla metodologia proposta da Bruno Bachimont. Questa metodologia è basata su tre fasi. Nella prima gli utenti devono creare la tassonomia e le relazioni che intercorrono tra i concetti per giustificare la loro posizione nella gerarchia, dopo di che deve definire in similitudini e differenze tra una relazione e quelle che derivano da essa. Durante la seconda fase le due tassonomie prodotte vengono importate dopo di che l’utente può aggiungere vincoli al dominio delle relazioni. Infine, all’ultimo passo, l’utente può tradurre l’ontologia nel linguaggio KR utilizzando un foglio di stile XSLT. Questo tool spesso non viene utilizzato come un ambiente autonomo per la produzione di ontologie ma come integrazione di altri.

5.2.5 ODE

ODE (Ontological Design Environment) è un tool per la costruzione di ontologie, sviluppato presso l’Univerità di Madrid, che interagisce con l’utente ad un livello concettuale. L’ontologia viene costruita completando delle tabelle e generando del codice che attualmente segue il formato di Ontolingua e F-Logic. Un vantaggio introdotto dall’utilizzo di tabelle è che l’utente non necessita nessuna informazione sul linguaggio di rappresentazione della conoscenza sottostante. Il tool permette di definire un modello concettuale esplicito dell’ontologia partendo dalla stessa. Le finestre di ODE prendono in considerazione tutte le parti del ciclo di vita di un’ontologia.

5.2.6 OntoLingua

Ontolingua Server è un insieme di tool e sevizi, sviluppati presso l’Università di Stanford, che supportano la costruzione di ontologie condivise tra gruppo distribuiti. L’architettura del server di ontologie permette l’accesso a librerie di ontologie, traduttori di linguaggio ed a un editor per creare ontologie condivise. Gli editor remoti possono condividere o visualizzare le ontologie e per applicazioni remote o locali possono accedere a qualsiasi ontologia della libreria usando il protocollo OKBC (Open Knowledge Based Connectivity).

5.2.7 WebOnto

Questo tool è stato progettato per realizzare l’esplorazione, la creazione e la visualizzazione di ontologie eliminando i problemi dovuti alle interfacce. WebOnto è stato realizzato in modo da essere di facile utilizzo e da supportare un tool per la “discussione” di ontologie chiamato Tadzebao. Il tool è costituito da un applet java e da un server che permette di esplorare e visualizzare i modelli condivisi sul web.

5.2.8 La scelta di Protégé

Per realizzare l’ontologia di dominio, sviluppata in questa tesi, si è scelto l’aiuto del tool grafico Protégé. I motivi che hanno spinto verso tale scelta sono molti.

Innanzi tutto, Protégé è sicuramente il tool più diffuso e in questo campo è quasi diventato uno standard di fatto. Esso gode di una ricca gamma di plug-in che ne espandono molto le funzionalità. Particolarmente interessante è il plug-in OWL che permette di creare una struttura ontologica in tale formato, ciò è un innegabile vantaggio

in quanto, come sappiamo, OWL è attualmente il linguaggio standard definito dal W3C per la codifica di ontologie.
Un altro aspetto interessante è che Protégé è scritto interamente in Java e fornisce delle API Java per accedere alla base di conoscenza descritta dall’ontologia per interrogarla e manipolarla. Ciò permette di creare dei propri componenti Java che accedono al modello del dominio e includerli come plug-in in Protégé.
Protégé è sostenuto da una vasta comunità di utenti e ciò che ha sicuramente favorito la sua diffusione e ha inciso sulla nostra scelta, è che esso è open-source.

5.3 Costruzione dell’ontologia OntoInfoProt

Per costruire l’ontologia del protocollo informatico (OntoInfoProt) sono stati seguiti i passi principali della metodologia di costruzione dell’ontologia descritta nel Cap.4. e si è fatto uso del tool Protégé per modellare l’ontologia utilizzando il plugin sull’OWL avendo scelto tale linguaggio per codificare l’ontologia.

5.3.1 Determinare lo scopo dell’ontologia

Passo fondamentale per costruire una corretta ontologia è determinarne lo scopo. Nel caso in esame l’ontologia viene costruita per poter rappresentare tutte le fasi di gestione di un protocollo informatico, dei suoi flussi documentali e degli archivi per permettere al personale interessato di poter apprendere i concetti necessari per poter assolvere in maniera corretta alle proprie mansioni e nel corretto ordine consigliato da un esperto del protocollo stesso.

5.3.2 Costruzione dell’ontologia

La fase di costruzione dell’ontologia rappresenta il cuore del processo di sviluppo dell’ontologia ed è costituita da sotto fasi.

5.3.2.1 Cattura dell’ontologia

In questa fase, dove bisogna determinare i concetti e le relazioni del dominio e definirli, è stato chiesto l’aiuto di un esperto del dominio che ci ha fornito le informazioni utili per poter individuare i concetti fondamentali e le relazioni che dovevano esistere tra questi concetti.

Tra i concetti che non possono sicuramente mancare perché costituiscono la versione minima del protocollo abbiamo:

- Registrazione documenti in Ingresso/Uscita
- Segnatura di Protocollo
- Gestione del Titolario d’Archivio;
- Classificazione e fascicolazione dei documenti
- Gestione del piano di conservazione dei documenti o massimario di scarto;
- Gestione della organizzazione dell’Ente (AOO, Utenti, etc);
- Annullamento delle registrazioni di protocollo
- Protocollo Particolare;
- Gestione del registro di emergenza
- Gestione dei fascicoli in formato elettronico;
- Gestione dei flussi documentali.

Le relazioni che possiamo individuare sono quelle caratteristiche di un dominio didattico, oltre al concetto di classe e sottoclasse, abbiamo la relazione di propedeuticità tra i concetti e la relazione che lega i concetti al materiale didattico.

5.3.2.2 Realizzazione della meta-ontologia

L’intervista all’esperto del dominio ci ha portato alla realizzazione della *meta-ontologia* che riguarda la scelta dei termini basilari (classi e relazioni) che saranno utilizzati per specificare l’ontologia.

Classi:

- Archivio;
- FlussiDoc;
- GestioneProtocollo;
- ProtocolloInformaticoIntro;
- Sicurezza.

Sottoclassi

- ArchivioCorrente; ArchivioDiDeposito; ArchivioStorico.
- Classificazione; FlussoInUscita; FlussoInIngresso; FlussoInterno; Fascicolazione; Assegnazione; StatoDeiDocumenti; PresaInCarico; Registrazione.
- DefinizioneTitolario; GestioneDizionari; StrutturaOrganizzativa.
- Normativa; Descrizione.
- RegistroEmergenza; GestioneSicurezzaIntro; DirittiDiAccesso.

Le sottoclassi conterranno a loro volta altre sottoclassi su uno o più livelli.
La gerarchia delle classi si presenterà in Protégé come in Fig 5.1

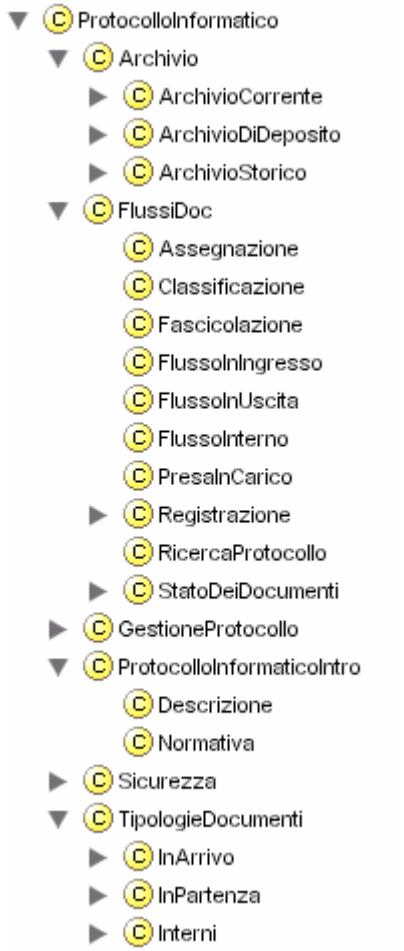


Figura 5. 1- Gerarchia delle classi

Relazioni:

- *IsPartOf* relazione che esprime il concetto di classe e sottoclasse;
- *Requires* esprime una relazione d’ordine tra i concetti;
- *IsRequiresBy* indica per ogni concetto quali sono gli altri concetti del dominio per cui è propedeutico.
- *IsExplainedBy* lega il concetto al Learning Object che lo spiega.
- *SuggestedOrder* suggerisce un ordine che è consigliato per apprendere gli argomenti.

Protégé mette a disposizione la possibilità di creare le relazioni tra oggetti e relazioni di tipo corrispondenti rispettivamente alle proprietà OWL owl:ObjectProperty, owl:DatatypeProperty.

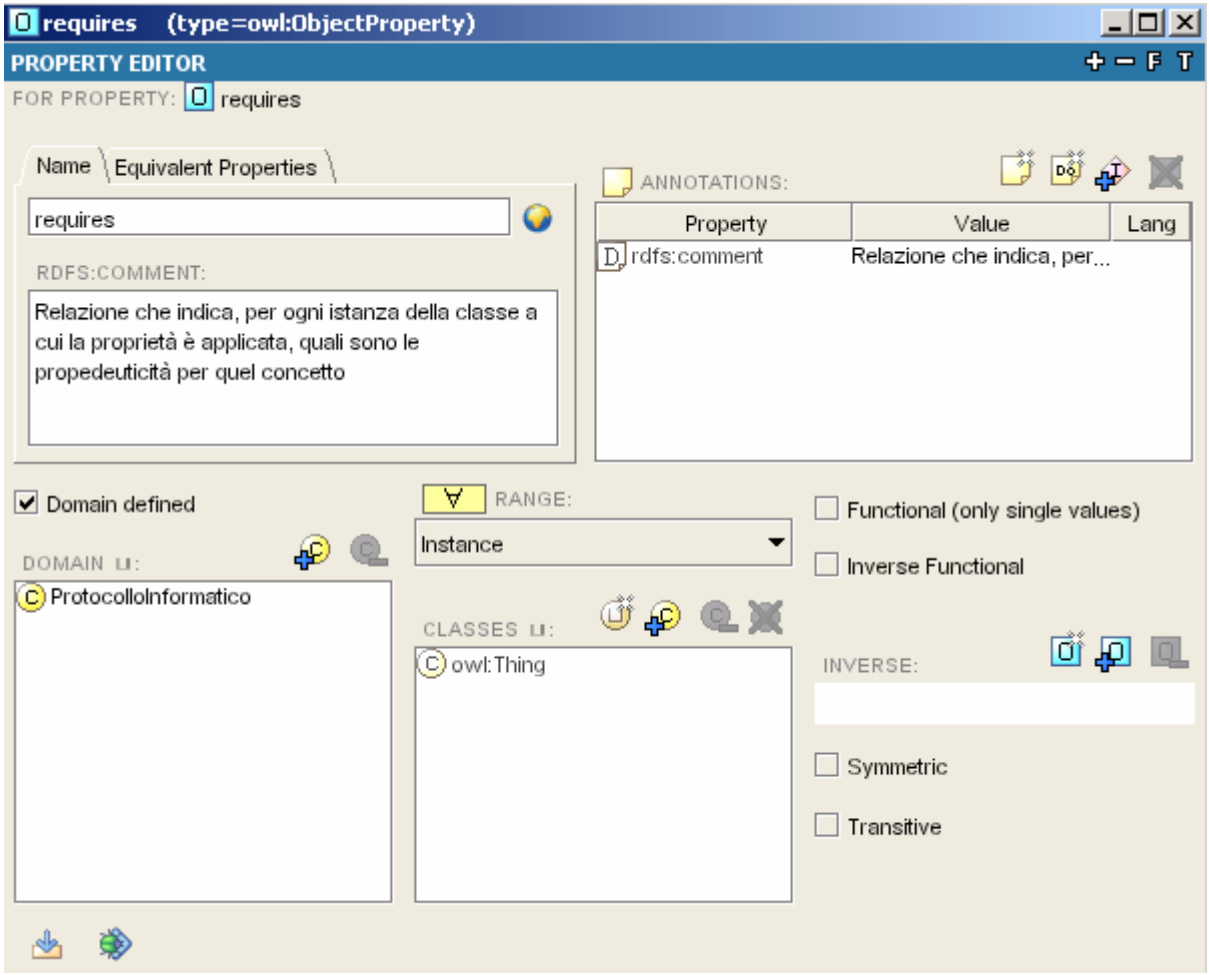


Figura 5. 2-Definizione delle proprietà in Protégé

Nella definizione delle proprietà bisogna indicare il dominio e il range. La proprietà collegherà individui del dominio con individui del range.

5.3.2.3 Costruzione della tassonomia

La terza sotto-fase della fase di costruzione dell’ontologia consiste nel tracciare le relazioni *is-a* tra i vari termini del dominio, in base alle loro definizioni, ottenendo la tassonomia. che nel nostro caso si presenterà come in Fig 5.3.

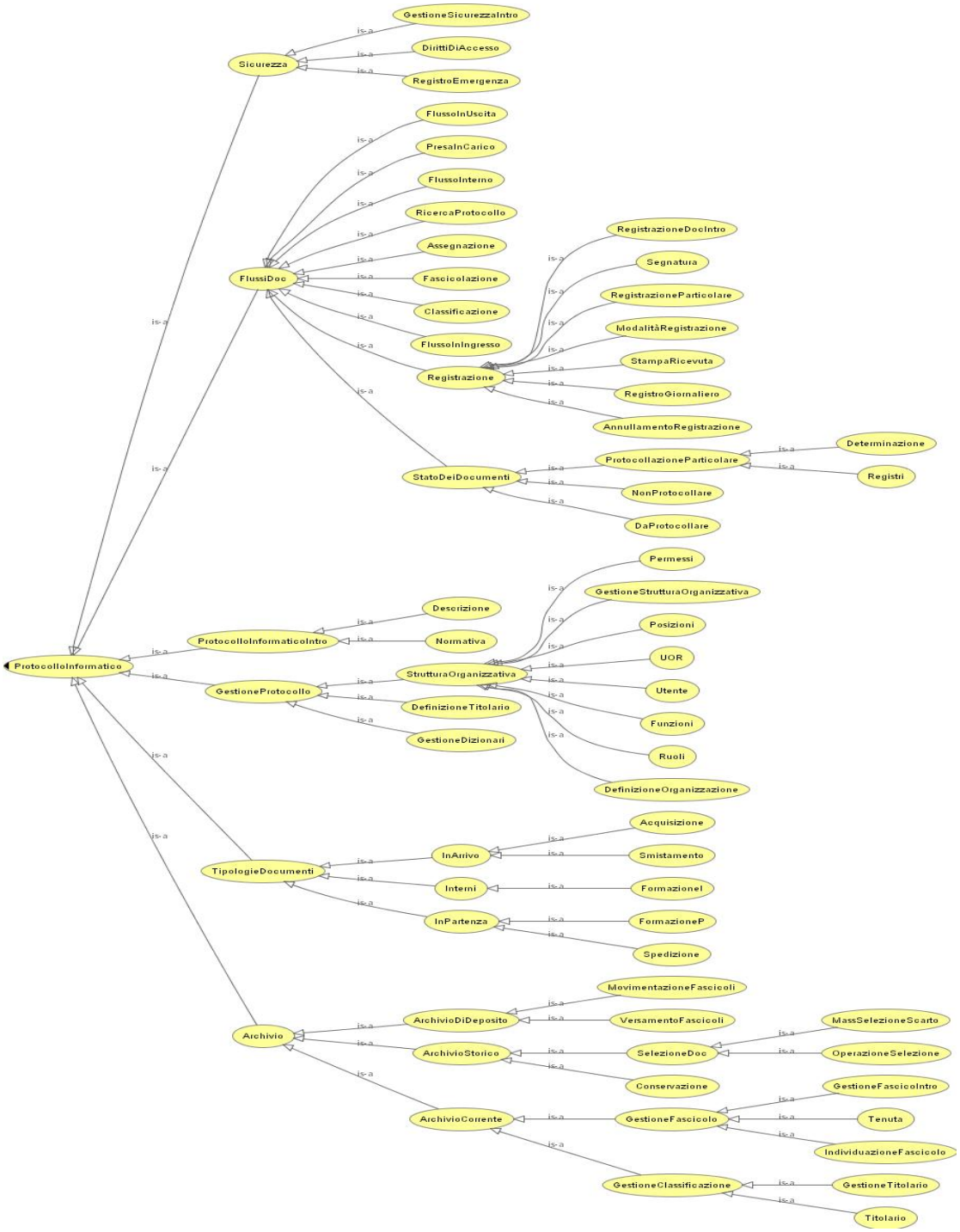


Figura 5. 3- Tassonomia dell’OntoProtInfo

5.4 Creazione delle istanze

Una volta che l’ontologia è stata costruita definendo classi e relazioni, bisogna creare le istanze.

Nel caso dell’ontologia del protocollo informatico è stata creata un’istanza per ogni classe in modo tale che potesse essere possibile agganciare le proprietà alle classi.
Le relazioni Requires, IsRequiresBy e SuggestedOrder sono popolate automaticamente dal sistema per ogni istanza creata, avendo definito delle restrizioni sui valori che tali proprietà devono assumere per ogni specifica classe.

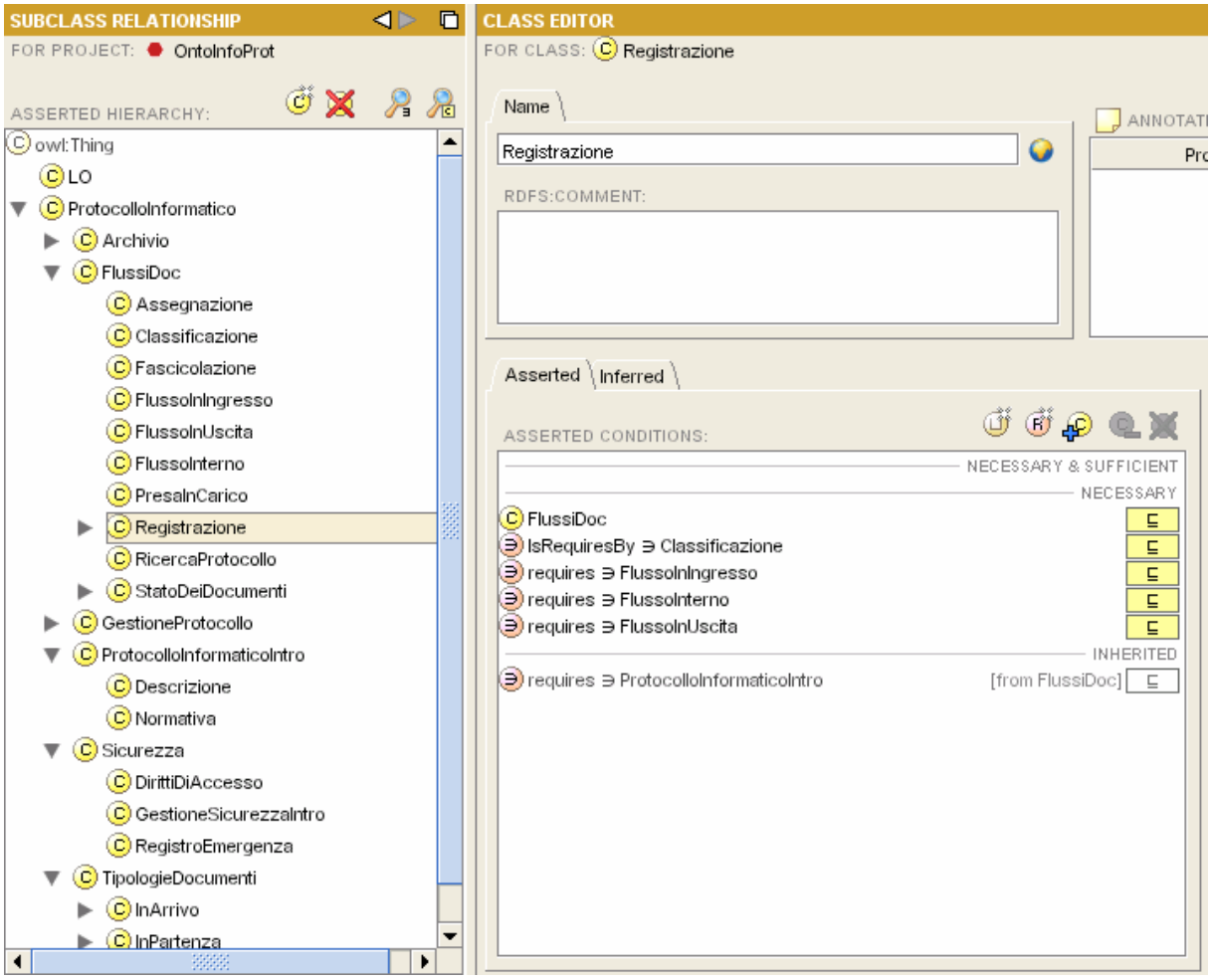


Figura 5. 4-Restrizione sui valori delle proprietà

Per ogni istanza bisogna indicarne il NOME con un identificativo unico, e il valore per il campo IsExplainedBy che associa al concetto il materiale didattico che lo spiega.

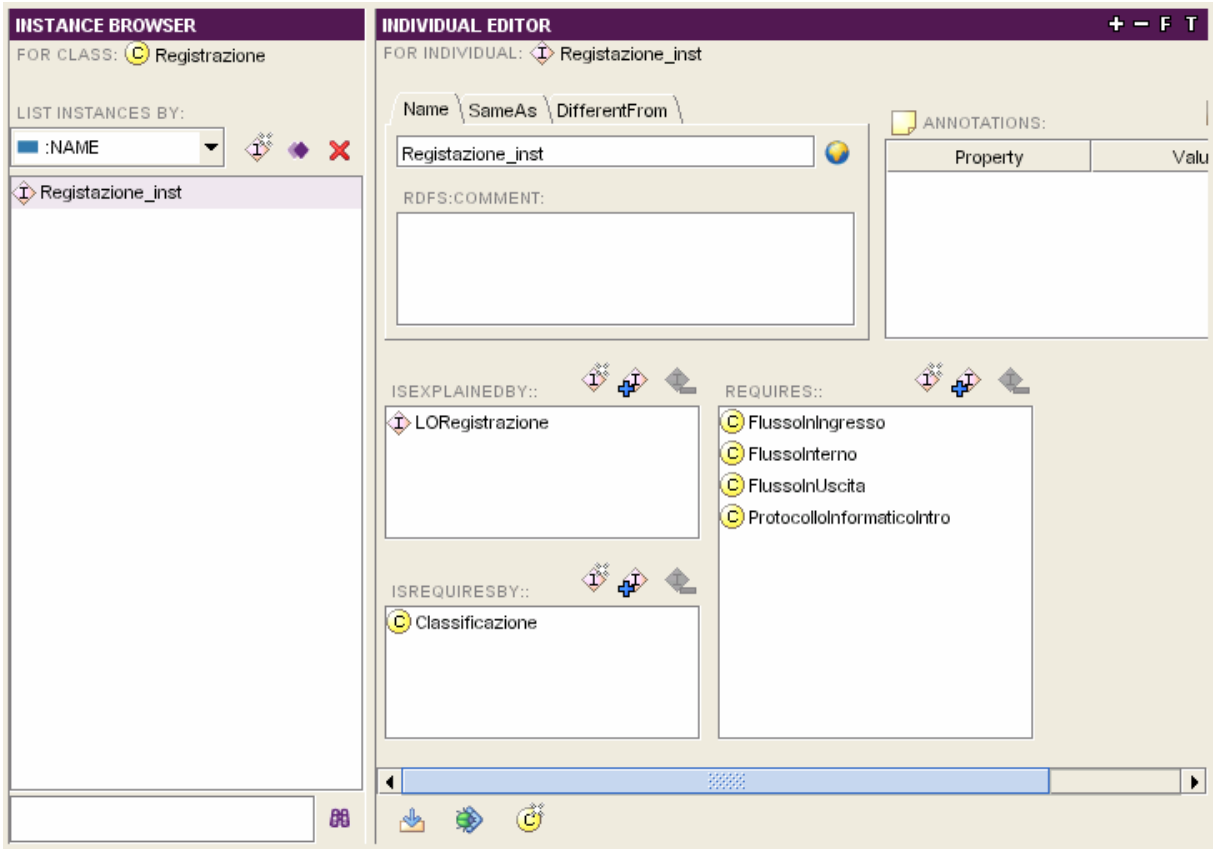


Figura 5. 5 Creazione delle istanze

5.4.1 Scelta del linguaggio di codifica dell’ontologia

Come linguaggio di codifica dell’ontologia è stato scelto OWL che rappresenta il linguaggio standard definito dal W3C per la definizione delle ontologie.

Avendo utilizzato il tool Protégé-OWL Plugin, l’ontologia viene automaticamente salva in formato OWL e il codice può essere automaticamente visualizzato mediante il comando Show Source Code.

5.4.2 Valutazione dell’ontologia

L’ontologia una volta creata deve essere attentamente valutata per verificarne soprattutto la chiarezza e la coerenza.

Navigando l’ontologia deve essere possibile estrarre tutta la conoscenza sul dominio d’interesse, in tal modo, l’ontologia rappresenta chiaramente tutte la conoscenza.

Essa deve essere internamente consistente con i vincoli imposti e con gli assiomi definiti.

Nel nostro caso per verificare la consistenza bisogna verificare che siano sempre validi i seguenti assiomi:

1. $\forall x, y \in D : IsPartOf(x, y) \Rightarrow \neg\exists x, y : IsPartOf(y, x)$

   Poiché l’ontologia deve rappresentare un grafo aciclico che collega i concetti del dominio, il rispetto di questo vincolo ci garantisce l’aciclicità.

2. $\forall x, y, : \mathrm{Re}\,quires(x, y) \Leftrightarrow Is\,\mathrm{Re}\,quireBy(y, x)$

   Se un concetto x richiede un concetto y allora, y è richiesto da x e viceversa.

5.4.3 Mantenimento dell’ontologia

L’ontologia del protocollo informatico dovrà essere mantenuta sempre aggiornata con i cambiamenti che possono interessare il protocollo. Dovrà allora essere modificata se si inseriscono nuovi concetti o se cambiano le relazioni tra essi, o se si modificano i vincoli sul dominio.

5.5 Ontologia per l’utente e ontologia per i Learning Objects

Al fine di testare l’ontologia di dominio con un prototipo che acceda alla base di conoscenza ed estragga i giusti L.O. che formeranno il percorso d’apprendimento dell’utente, abbiamo rappresentato il profilo dello studente e l’insieme dei L.O. attraverso due ontologie.

L’ontologia del profilo utente contiene la definizione di tre profili professionali *Amministratore, Operatore, Responsabile* per ogni profilo si definiscono mediante la relazione “hasCompetence” le competenze che l’utente deve acquisire per raggiungere l’obiettivo professionale scelto.

L’ontologia per i Learning Objects conterrà per ogni L.O. un identificativo, il titolo e l’indirizzo fisico in cui la risorsa risiede.

5.6 Un’applicazione per la pianificazione automatica delle attività di learning

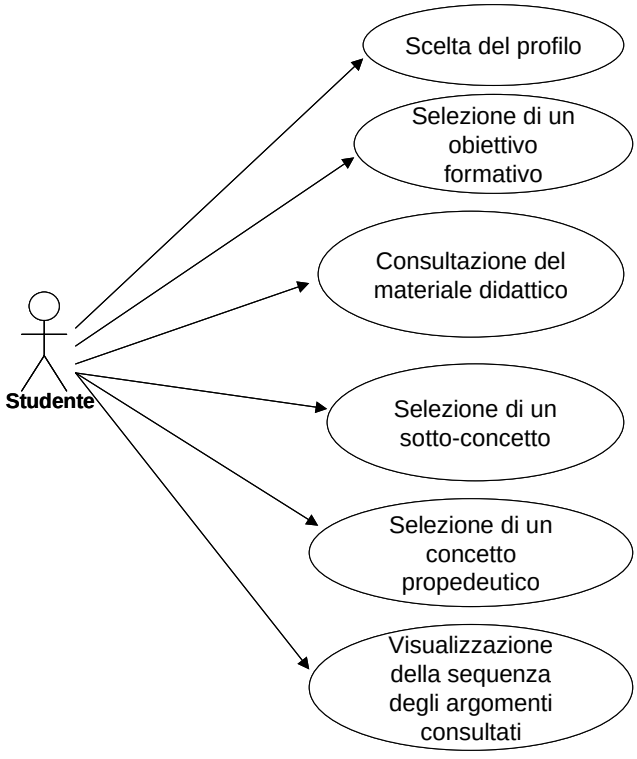
Il prototipo realizzato mira ad utilizzare l’ontologia del protocollo informatico costruita, per effettuare una pianificazione automatica e interattiva dei contenuti didattici.

L’utente che accede al sistema è guidato nella scelta del materiale di apprendimento per acquisire una competenza professionale da lui scelta.

Di seguito si elencano le funzionalità messe a disposizione dal prototipo per l’utente finale:

- Scelta di un “profilo formativo” associato al “protocollo informatico”
- Selezione di un obiettivo formativo o competenza da acquisire associato al profilo
- Consultazione del materiale didattico associato all’obiettivo/concetto selezionato
- A partire da un dato concetto, selezione di uno dei sotto-concetti associati
- A partire da un dato concetto, selezione di uno dei concetti propedeutici
- Visualizzazione della sequenza degli argomenti consultati

Nella seguenta figura vengono illustrati gli scenari d’uso relativi allo studente.



A partire dai profili disponibili per l’apprendimento del protocollo informatico (*Amministratore, Operatore, Responsabile*), l’utente sceglie il profilo che vuole acquisire accedendo agli obietti formativi associati a quel profilo che rappresentano dei macroconcetti su cui l’utente si deve formare e che costituiscono argomenti che l’utente deve apprendere nell’ordine consigliato.

Navigando attraverso gli obiettivi formativi l’utente accede all’ontologia del dominio e ai concetti che deve apprendere e per ogni concetto visualizza le seguenti informazioni:

- l’insieme ordinato dei sotto-concetti
- l’insieme dei concetti propedeutici
- la sequenza dei concetti appresi
- il contenuto didattico che spiega il concetto.

L’utente viene guidato nel suo percorso didattico dal sistema che per ogni concetto da apprendere gli indica quali sono i concetti propedeutici e di quali sottoconcetti si compone, e qual’è il corretto ordine con cui i concetti devono essere appresi.
Nella figura 5.6 vengono illustrati i legami esistenti tra i profili di apprendimento, gli obiettivi formativi o competenze da acquisire, i concetti dell’ontologia di dominio e le risorse didattiche che spiegano appunto tali concetti.

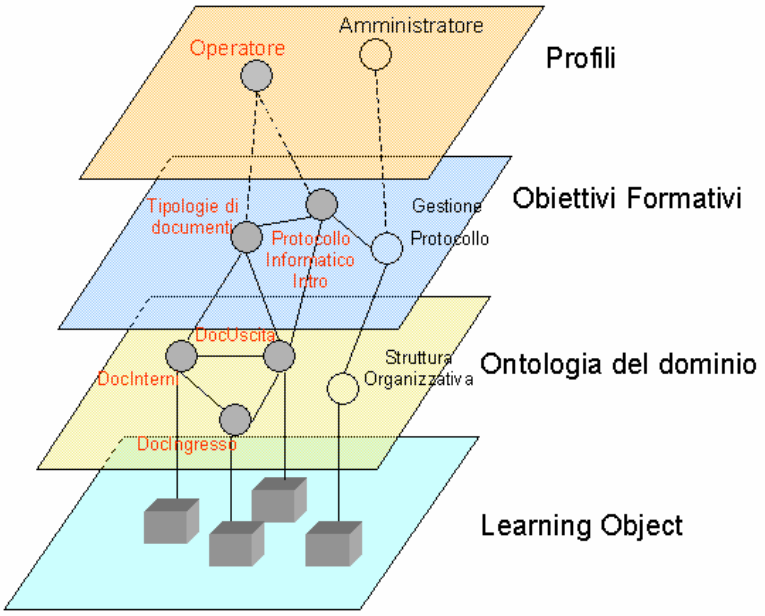


Figura 5. 6 – Relazioni tra profili, competenze e concetti del dominio didattico

Dalla figura precedente si evince che un utente per acquisire le competenze relative a un profilo di “Operatore” (si intende operatore del protocollo informatico) deve apprendere , tra i vari concetti di cui si compone la realtà del protocollo informatico, sicuramente il “ProtocolloInformaticoIntro” e “TipologieDocumenti”.

5.6.1 Architettura del prototipo

L’architettura del prototipo può essere schematizzata come in Fig. 5.7 dove si possono individuare i seguenti livelli:

- Livello Client - costituito dal web browser che consente all’utente di consultare da remoto l’applicazione.
- Livello Web Application Server - si compone dell’insieme di pagine web sia statiche che dinamiche e dalla logica applicativa
- Livello risorse o dati - costituito dall’insieme delle risorse utilizzate dal sistema, come la knowledge base accessibile attraverso l’ambiente costituito da Algernon e Protégé

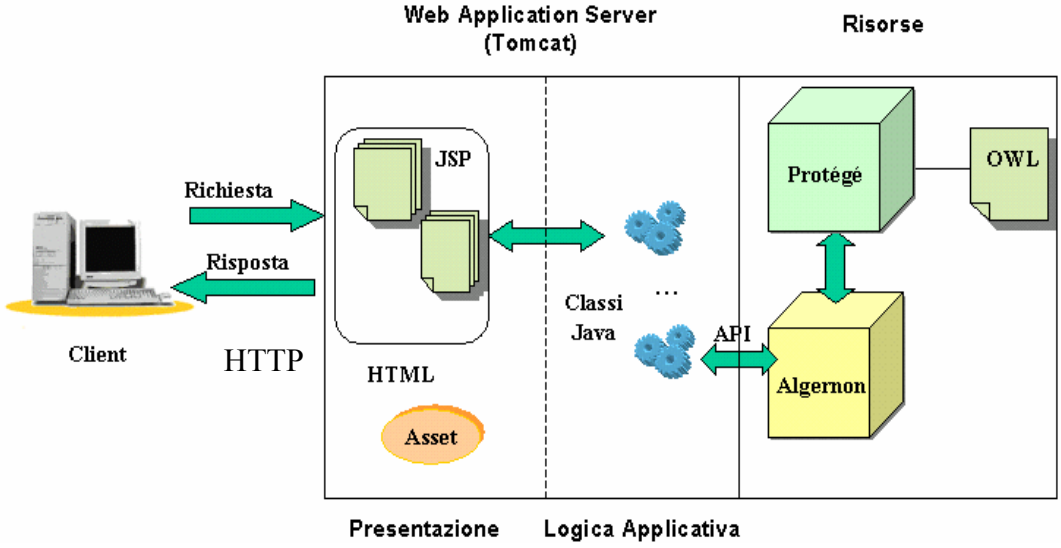


Figura 5. 7-Architettura del prototipo

5.6.2 Client

Il client richiede al web application server Tomcat una pagina dinamica JSP, attraverso il protocollo HTTP. Il server verifica, innanzitutto, se tale pagina è già stata compilata, quindi carica ed esegue il codice Java della pagina JSP, producendo in output la pagina HTML da inviare al browser. Se la pagina dinamica non è stata ancora compilata, ovvero viene richiesta per la prima volta, allora Tomcat provvederà a tradurla e compilarla in linguaggio Java. Il browser (lato client) riceverà la pagina in formato HTML generata dal server e quindi non necessita di alcun applicazione esterna per visualizzare le informazioni rivecute.

5.6.3 Web Application Server - Tomcat

Il web application server è l’ambiente che consente di servire le richieste dell’utente, di processare le pagine dinamiche per la presentazione dell’’informazione al client e di eseguire le classi Java. Le classi implementate realizzano l’applicazione per la pianificazione dei contenuti interagendo con la base di conoscenza attraverso il terzo livello risorse.

Nel prototipo in questione viene utilizzato l’ambiente open source Tomcat.

Tomcat contiene al suo interno tutte le funzionalità tipiche di un web server, ovvero ha la capacità di interpretare una richiesta di una risorsa veicolata su protocollo HTTP, indirizzarla ad un opportuno gestore (o prenderla dal filesystem) e restituire poi il risultato (codice HTML o contenuto multimediale che sia).

Supponiamo che un client acceda a una risorsa su di un server su cui gira Tomcat. Innanzi tutto viene instanziato un oggetto *Request*, `Req`, in Tomcat nel quale vengono inserite tutte le informazioni ricevute tramite HTTP (l’url richiesta sino ai cookies) e "al suo fianco" viene instanziato un oggetto risposta (*Response*, `Res`) nel quale costruire pian piano la risposta da restituire al client. Leggendo l’url all'interno della richiesta Tomcat è quindi in grado di comprendere a quale dei contesti residenti in memoria debbano essere consegnati questi due oggetti `Req` e `Res`.

Il contesto invocato da Tomcat provvederà a vedere se esiste già una sessione attiva per quel tipo di client leggendone la "firma" nella richiesta (sotto forma di cookie o nell’url, a seconda del metodo usato). Se la sessione, `Ses`, esiste già tra quelle immagazzinate nel *context*, il contesto la riprende e la affianca agli oggetti `Req` e `Res`. A questo punto l'ambiente è stato completamente definito e il web server può passare il controllo all'applicazione associata alla risorsa in questione.

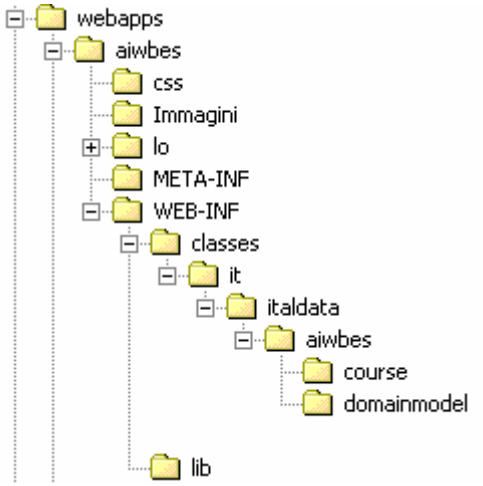
Struttura del repository del prototipo:

Nel repository è organizzato l’insieme delle risorse web, delle applicazione e delle librerie che realizzano il prototipo.

Le risorse Web sono:

- pagine html, stile di presentazione (css) e asset (immagini, testo, audio, video)
- pagine dinamiche JSP

Di seguito viene mostrata la struttura del repository:



Nella directory “aiwbes” sono stati posizionati i seguenti file:

- i file JSP: index.jsp, competence.jsp, frame.jsp, testata.jsp e topics.jsp.
- il foglio di stile per la presentazione “/css/ stili_me.css”
- le immagini nella cartella Immagini
- i contenuti in formato HTML nella directory “/lo/”

Nella sottodirectory classes, invece, si trovano le classi dell’applicazione, suddivise in funzione del loro package.

Nella sottocartella “lib”, infine, sono state posizionate le librerie esterne che sono utilizzate dall’applicazione, come la libreria “algernon.jar”, “protege.jar” e “protege-owl.jar” che consentono l’utilizzo delle funzionalità per l’interfacciamento con gli ambienti algernen e protege.

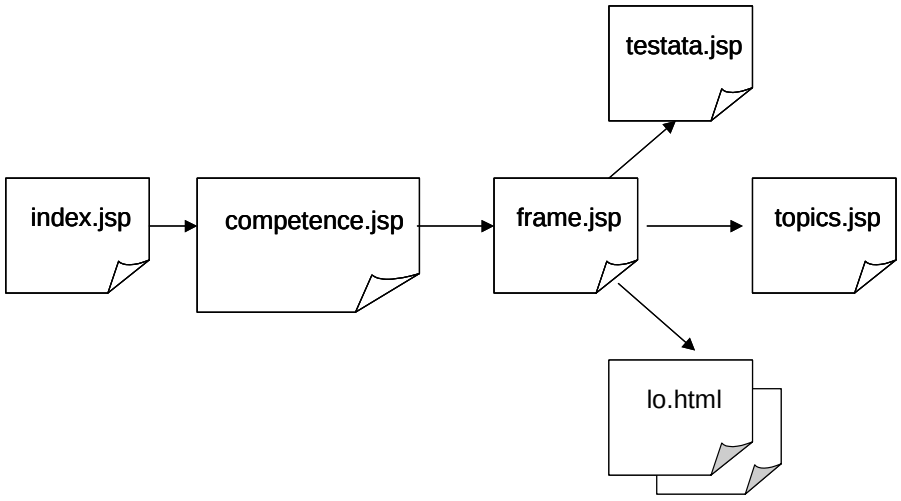
5.6.3.1 JSP (Java Server Pages)

Una pagina JSP è un semplice file di testo che fonde codice html a codice Java a formare una pagina dai contenuti dinamici. La possibilità di fondere codice html con codice Java senza che nessuno interferisca con l’altro consente di isolare la rappresentazione dei contenuti dinamici dalle logiche di presentazione. Il disegnatore potrà concentrarsi solo sulla impaginazione dei contenuti che saranno inseriti dal programmatore che non dovrà preoccuparsi dell’aspetto puramente grafico.

Da sole, Java Server Pages consentono di realizzare applicazioni web dinamiche accedendo a componenti Java contenenti logiche di business o alla base dati del sistema. In questo modello il browser accede direttamente ad una pagina JSP che riceve i dati di

input, li processa utilizzando eventualmente oggetti Java, si connette alla base dati effettuando le operazioni necessarie e ritorna al client la pagina html prodotta come risultato della processazione dei dati.
Se dal punto di vista del programmatore una pagina JSP è un documento di testo contenente tag html e codice Java, dal punto di vista del server una pagina JSP è utilizzata allo stesso modo di una servlet. Di fatto, nel momento del primo accesso da parte dell’utente, la pagina JSP richiesta viene trasformata in un file Java e compilata dal compilatore interno della virtual machine. Come prodotto della compilazione otterremo una classe Java che rappresenta una servlet di tipo HttpServlet che crea una pagina html e la invia al client.
Tipicamente il web server memorizza su disco tutte le definizioni di classe ottenute dal processo di compilazione appena descritto per poter riutilizzare il codice già compilato. L’unica volta che una pagina JSP viene compilata è al momento del suo primo accesso da parte di un client o dopo modifiche apportate dal programmatore affinché il client acceda sempre alla ultima versione prodotta.
Le pagine JSP non possono venire visualizzate mediante l'utilizzo di un normale server web, necessitano infatti di un particolare software in grado di interpretarle.

Le JSP realizzate per il prototipo:
Di seguito vengono mostrate le JSP realizzate e le relative associazioni:

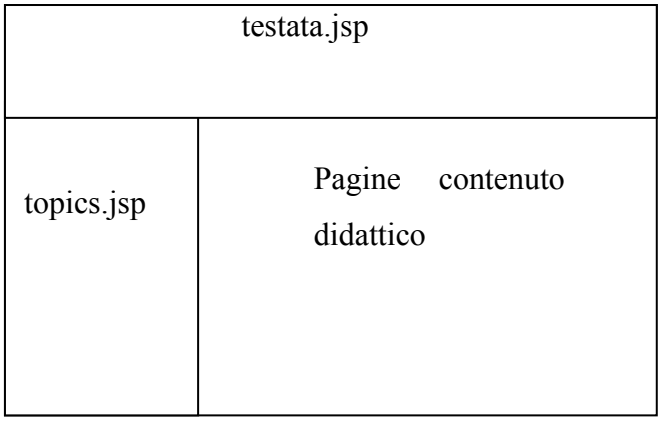


La pagine index.jsp presenta all’utente l’insieme dei profili legati al dominio del protocollo informatico e cioè “amministratore”, “operatore” e “responsabile”. A ciascun

profilo corrisponde un insieme di obiettivi da raggiungere che vengono mostrati dalla pagina *competence.jsp*. In seguito alla selezione di un obiettivo vengono presentati dalla pagina *frame.jsp* il concetto da apprendere, gli altri concetti di cui si compone e quelli che sono propedeutici per l’apprendimento di tale concetto. In particolare, la pagina frame.jsp si compone di altre tre pagine:

- testata.jsp che si occupa della testata della pagina dove vengono riportati i loghi e mostrata la sequenza degli argomenti studiati
- topics.jsp che mostra il concetto da apprendere, i concetti associati e i concetti propedeutici.
- le pagine per i contenuti associati ai concetti

Di seguito si riporta la struttura della pagina frame.jsp:



Interazioni tra le pagine jsp:

La jsp *competence.jsp* riceve da *index.jsp* il parametro “_profile” che indica il tipo di profilo di cui si vogliono visualizzare gli obiettivi da raggiungere o se si vuole le competenze da acquisire. La pagina *frame.jsp*, invece, riceve da *competence.jsp* il parametro “_concept” che indica il concetto da apprendere. Questo parametro viene comunicato alla pagina topics.jsp che si occupa di richiamare la classe CourseControl per la navigazione dell’ontologia di dominio e di mostrare il concetto da apprendere e i concetti ad esso associati.

5.6.3.2 Classi Java

Le classi Java, invocate all’interno delle pagine JSP, produrranno dinamicamente i risultati da inviare al client. Utilizzando le API Algernon, esse accedono e interrogano la base di conoscenza di Protégé che rappresenta la nostra ontologia di dominio.

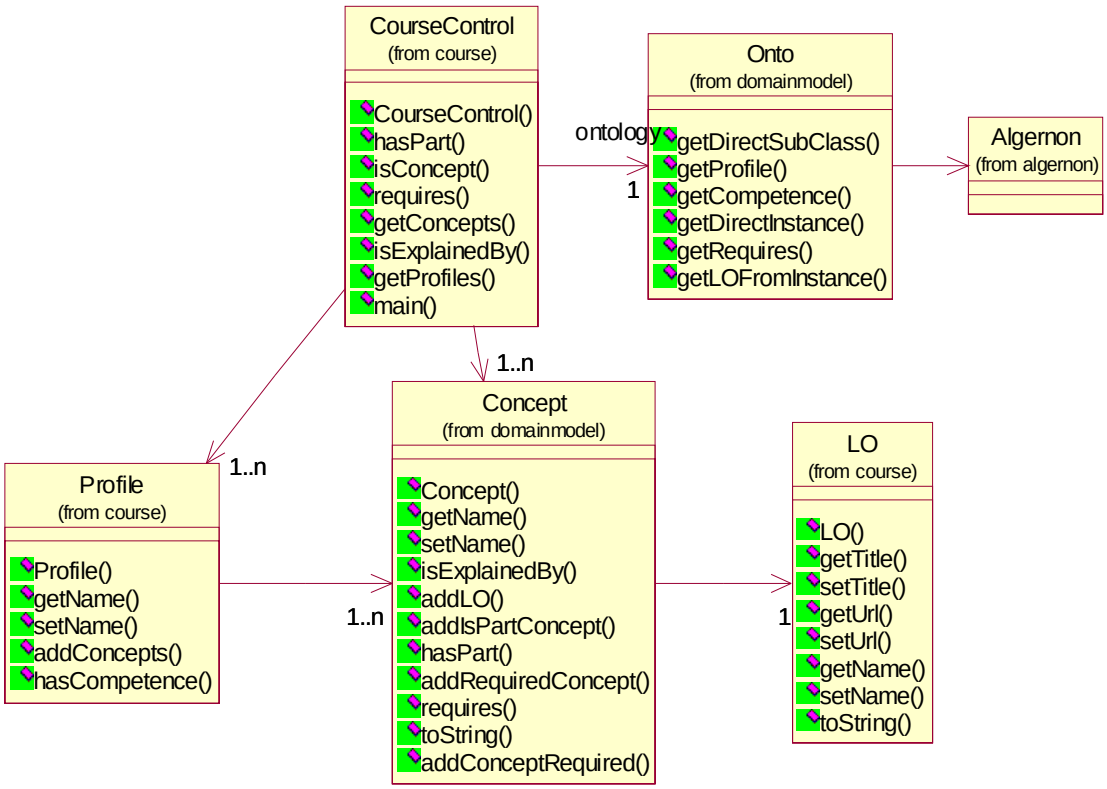


Figura 5. 8-Diagramma delle classi

5.6.4 CourseControl

La classe CourseControl gestisce la logica con cui vengono associati i concetti del corso. Inizializza la connessione con l’ontologia

5.6.5 Profile

La classe Profile definisce il profilo dell’utente e le sue competenze, ossia i concetti da apprendere associati alla specifica competenza.

5.6.6 LO

La classe LO definisce i Learning Object attraverso il nome, il titolo e URL.

5.6.7 Concept

La classe Concept gestisce i concetti associati al corso e il collegamento dei concetti con i Learning Object.

5.6.8 Onto

Gestisce la connessione con Algernon e Protégé e mette a disposizione una serie di metodi per la navigazione dell’ontologia di dominio.
Utilizzando le API di Protégé e Algernon carica la base di conoscenza dell’ontologia e la interroga attraverso Algernon. Definisce le seguenti funzioni per le interrogazioni attraverso le query in Algernon:

FUNZIONE	QUERY ALGERNON
getDirectSubClass() restituisce le sottoclassi dirette associate al concetto	((:TRACE :VERBOSE)(:DIRECT-SUBCLASS " + _concept +" ?concetto))
getProfile() restituisce le sottoclassi dirette del profilo	((:TRACE :VERBOSE)(:direct-subclass "+_profile+" ?profile))
getCompetence() restituisce le competenze associate al profilo	((:TRACE :VERBOSE)(:direct-instance "+_profile+" ?instance)(hasCompetence ?instance ?concetto))"
getDirectInstance() restituisce le istanze dirette del concetto	((:TRACE :VERBOSE)(:DIRECT-INSTANCE " + _concept +" ?instance))";
getRequires() restituisce per l’istanza del concetto associato i concetti che lo richiedono	((:TRACE :VERBOSE)(requires " + _instance + " ?requires))";
getLOFromInstance() restituisce il LO associato al concetto, titolo e URL	((:TRACE :VERBOSE)(isExplainedBy " + _instance + " ?resource)(title ?resource ?LOtitle)(url ?resource ?LOurl)

5.6.9 Algernon / Protégé

Algernon è un motore inferenziale che supporta regole di forward e backward e permette di interrogare la Knowledge Based di Protégé. Una regola di forward è automaticamente attivata, quando un nuovo dato è memorizzato nella base di conoscenza. Una regola di backward è automaticamente attivata, quando viene

interrogata la base di conoscenza. Esso è implementato in Java e contiene una serie di API che consentono di accedere alle sue funzionalità.

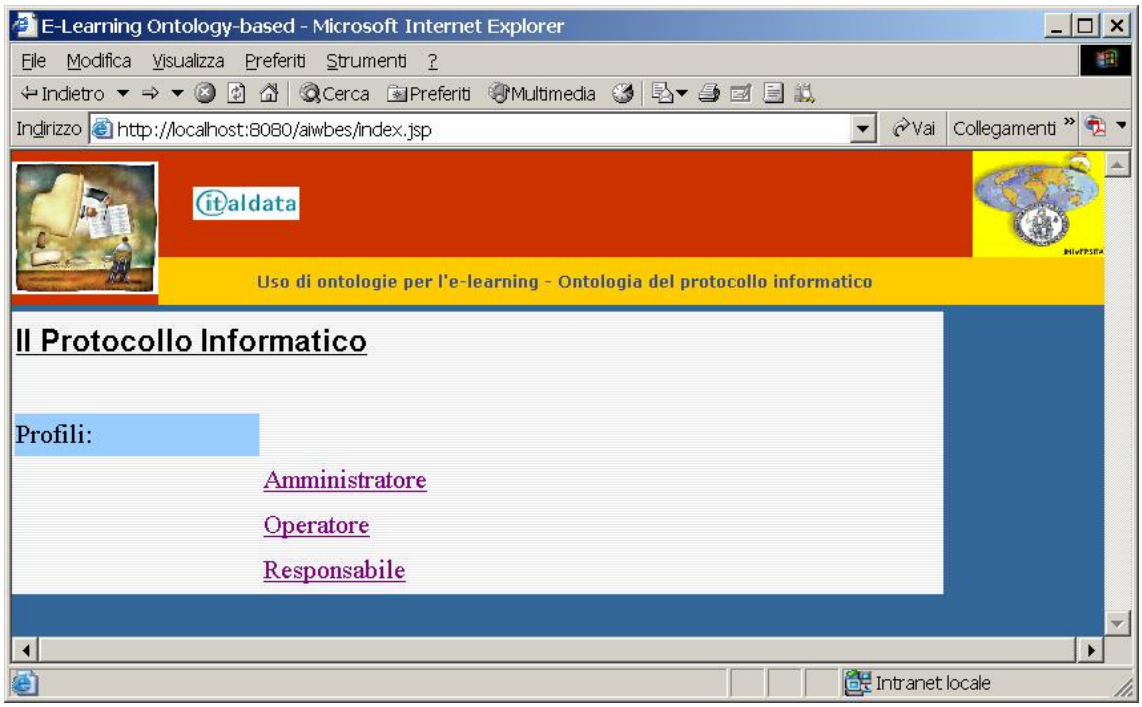
Utilizzando le API Algernon, è possibile interrogare da Java la base di conoscenza sfruttando il linguaggio di interrogazione che Algernon mette a disposizione. La base di conoscenza su cui esso opera è la nostra ontologia di dominio memorizzata nel file di Protégé OntoInfoProt.pprj a cui corrisponde l’ontologia in formato OWL OntoInfoProt.owl.

5.7 Interfacce utente del prototipo

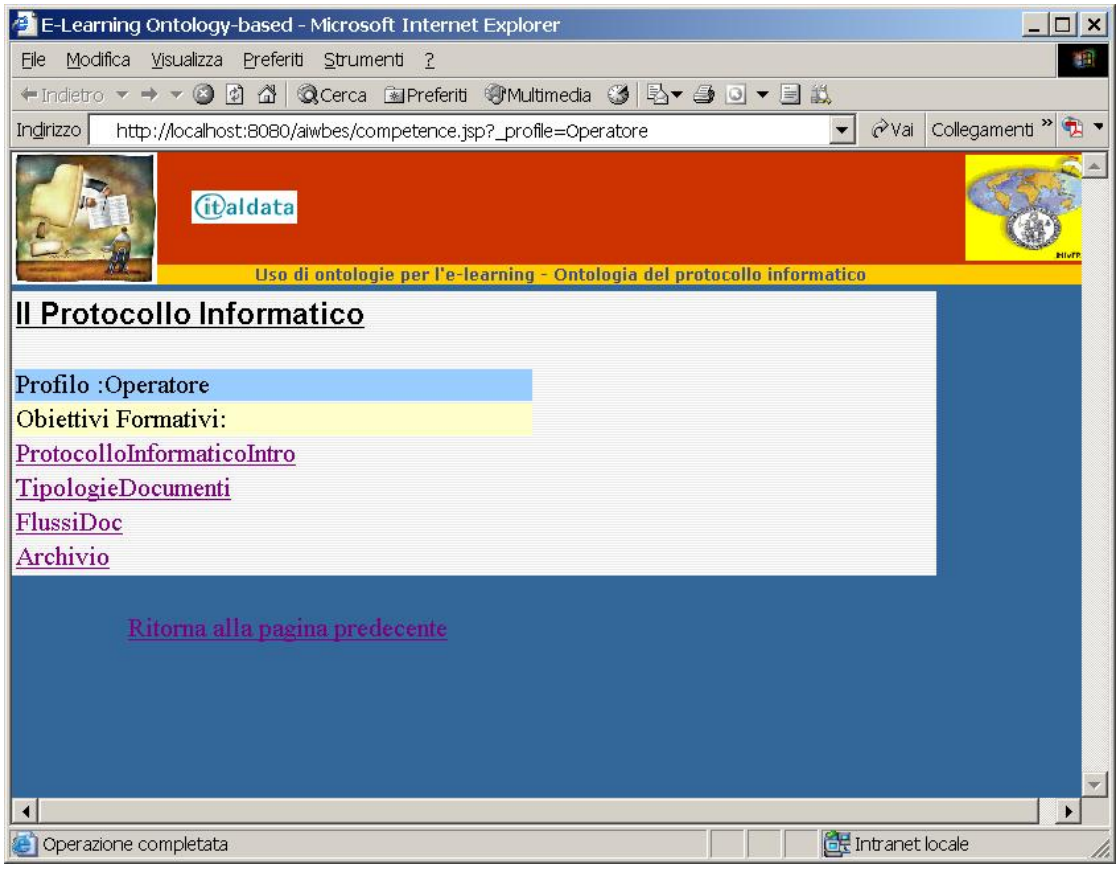
Come già detto in precedenza in questo capitolo, l’architettura del prototipo utilizza la tecnologia Web, per questo motivo le interfacce utente sono semplici pagine HTML che possono essere interpretate da un qualunque browser Web, come Internet Explorer e Netscape.

Nelle figure che seguono vengono mostrati alcuni screeshot dell’applicazione che mostrano come le informazioni vengono presentate all’utente finale.

Nel seguente screenshot viene mostrata l’interfaccia utente per la selazione del profilo d’apprendimento da parte dell’utente.

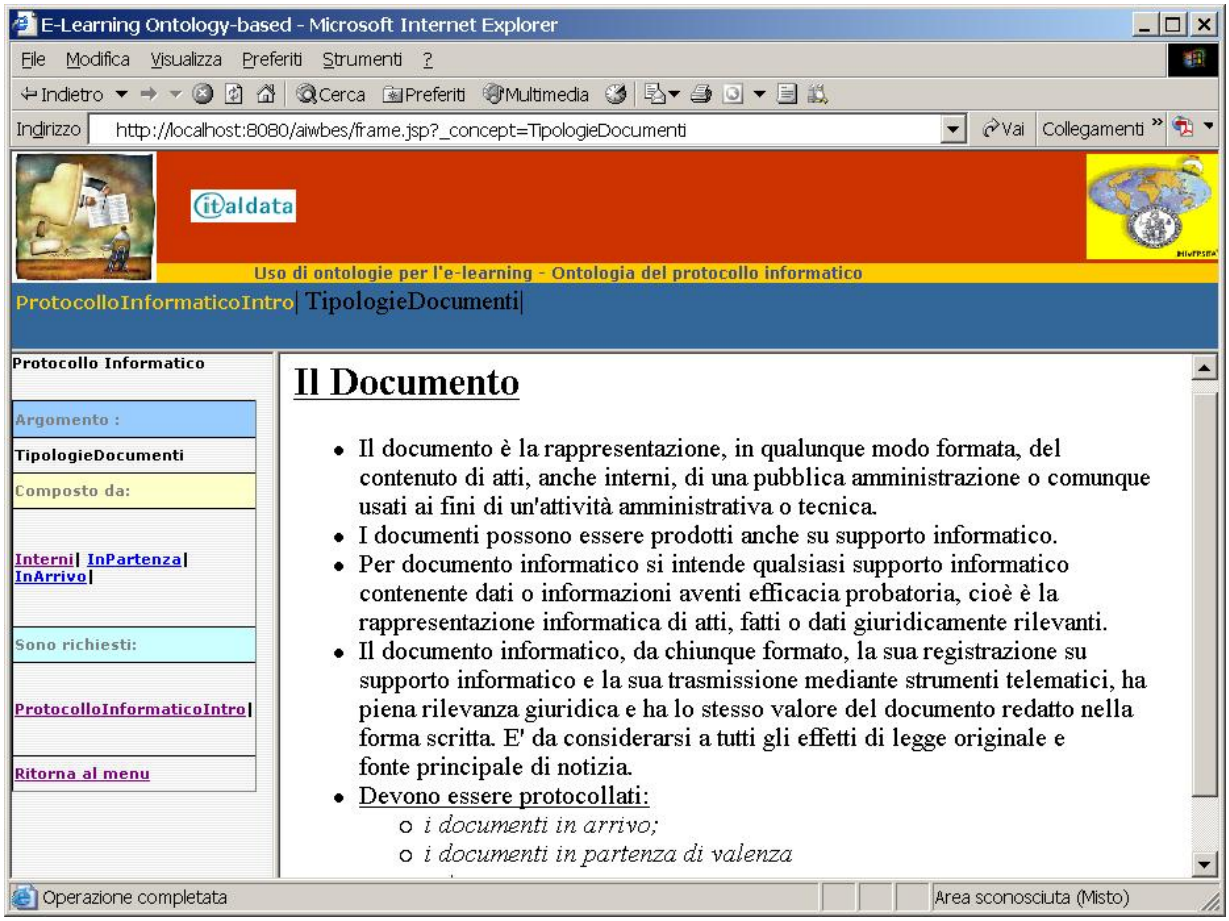


Nel seguente screenshot viene, invece, mostrata l’interfaccia utente per la selazione di un obiettivo formatico o competenza da acquisire.



Nel seguente screenshot viene, infine, mostrata l’interfaccia utente per la consultazione del materiale didattico associato al concetto selezionato. Tale interfaccia è suddivisa in tre parti:

- parte superiore - La testata che mostra, oltre al logo dell’applicazione, la sequenza degli argomenti consultati
- parte sinistra - Il menu che mostra il concetto corrente, i sotto-concetti associati (“composto da”), e i concetti propedeutici (“sono richiesti”)
- parte centrale – Il materiale didattico che spiega il concetto/argomento da apprendere


E-Learning Ontology-based - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://localhost:8080/aiwbes/frame.jsp?_concept=TipologieDocumenti
Vai Collegamenti
italdata
Uso di ontologie per l'e-learning - Ontologia del protocollo informatico
ProtocolloInformaticoIntro| TipologieDocumenti|
Protocollo Informatico
Argomento :
TipologieDocumenti
Composto da:
Interni| InPartenza| InArrivo|
Sono richiesti:
ProtocolloInformaticoIntro|
Ritorna al menu
Il Documento
Il documento è la rappresentazione, in qualunque modo formata, del contenuto di atti, anche interni, di una pubblica amministrazione o comunque usati ai fini di un'attività amministrativa o tecnica.
I documenti possono essere prodotti anche su supporto informatico.
Per documento informatico si intende qualsiasi supporto informatico contenente dati o informazioni aventi efficacia probatoria, cioè è la rappresentazione informatica di atti, fatti o dati giuridicamente rilevanti.
Il documento informatico, da chiunque formato, la sua registrazione su supporto informatico e la sua trasmissione mediante strumenti telematici, ha piena rilevanza giuridica e ha lo stesso valore del documento redatto nella forma scritta. E' da considerarsi a tutti gli effetti di legge originale e fonte principale di notizia.
Devono essere protocollati:
i documenti in arrivo;
i documenti in partenza di valenza
Operazione completata
Area sconosciuta (Misto)

Capitolo VI

6 Conclusioni e sviluppi futuri

Nell'ambito di questa tesi dopo aver descritto un'architettura di massima per il delivery intelligente e adattivo dei contenuti sono stati approfonditi gli aspetti fondamentali e gli approcci legati alla rappresentazione del dominio di conoscenza per arrivare a definire un modello di dominio specifico per la didattica. Il modello realizzato coniuga la rappresentazione esplicita della conoscenza con la descrizione esplicita dei learning object attraverso l'impiego di metadata standard. Per la formalizzazione del dominio di conoscenza è stato utilizzato il linguaggio OWL e un vocabolario di relazioni tra i concetti che è stato definito a partire dai metadata standard per l'e-learning (Dublin Core e LOM). Tale vocabolario potrà essere condiviso tra le applicazioni che eseguono ragionamenti automatici sul dominio della conoscenza.

Oltre al modello di dominio didattico, altri risultati importanti della tesi sono l'ontologia "Protocollo Informatico" che è stata realizzata applicando, appunto, le regole definite nel modello, e un prototipo in grado di interrogare l'ontologia e guidare in maniera "intelligente" l'utente attraverso il materiale di apprendimento per il raggiungimento degli obiettivi legati al profilo professionale da lui scelto.

Il modello del dominio didattico rappresenta una parte, seppure fondamentale, dell'architettura presentata nella tesi. Per centrare l'obiettivo di costruire un percorso formativo individuale basato sulle reali esigenze e preferenze dell'utente bisognerà ovviamente completare l'architettura in tutte le sue parti. Per questo motivo gli sviluppi futuri riguarderanno la ricerca di una giusta rappresentazione del modello dello studente e delle mappe di competenza compatibilmente con il modello del dominio didattico definito in questo lavoro. In tale contesto si potrà logicamente utilizzare l'ontologia "Protocollo Informatico" ed estendere e migliorare il prototipo per la consultazione "intelligente" dei contenuti didattici.

Per quanto riguarda la rappresentazione del modello dello studente e delle mappe di competenza si potrebbe, ad esempio, valutare la possibilità di impiegare le ontologie, come è stato fatto in questa tesi per rappresentare il modello del dominio.

L'ontologia dello studente potrà essere conforme ai maggiori standard di rappresentazione per il profilo dell'utente, come IMS LIP eIEEE PAPI.

PAPI introduce categorie come *Personal* che contiene informazioni personali sull'utente quali nome, indirizzo, contatto, *Relations* specifica le relazioni con gli altri utenti, S*ecurity* contiene informazioni sulla sicurezza e la privace, *Porfolio* rappresenta l'esperienze dell'utente. Importanti ai fini del nostro sistema sono sicuramente le categorie *Preference,* che indicano le preferenze dell'utente sui device di fruizione e sui tipo di materiale da presentargli e sulla lingua, e *Performance* che rappresenta la parte dinamica del profilo utente da aggiornare ogni qual volta un utente ha acquisito nuova conoscenza, contiene informazioni sul materiale d'apprendimento che l'utente conosce.

LIP introduce una categoria molto importante *Goal,* che contiene tutti gli obiettivi che l'utente deve raggiungere e assegna loro una priorità. L'obiettivo con priorità più alta dovrà essere raggiunto per primo.

Un esempio di ontologia per il learner è schematizzata in Fig.6.1. ed e quella creata nel contesto di EU/IST project Elena, in essa si fa riferimento sia allo standard PAPI che a quello LIP.

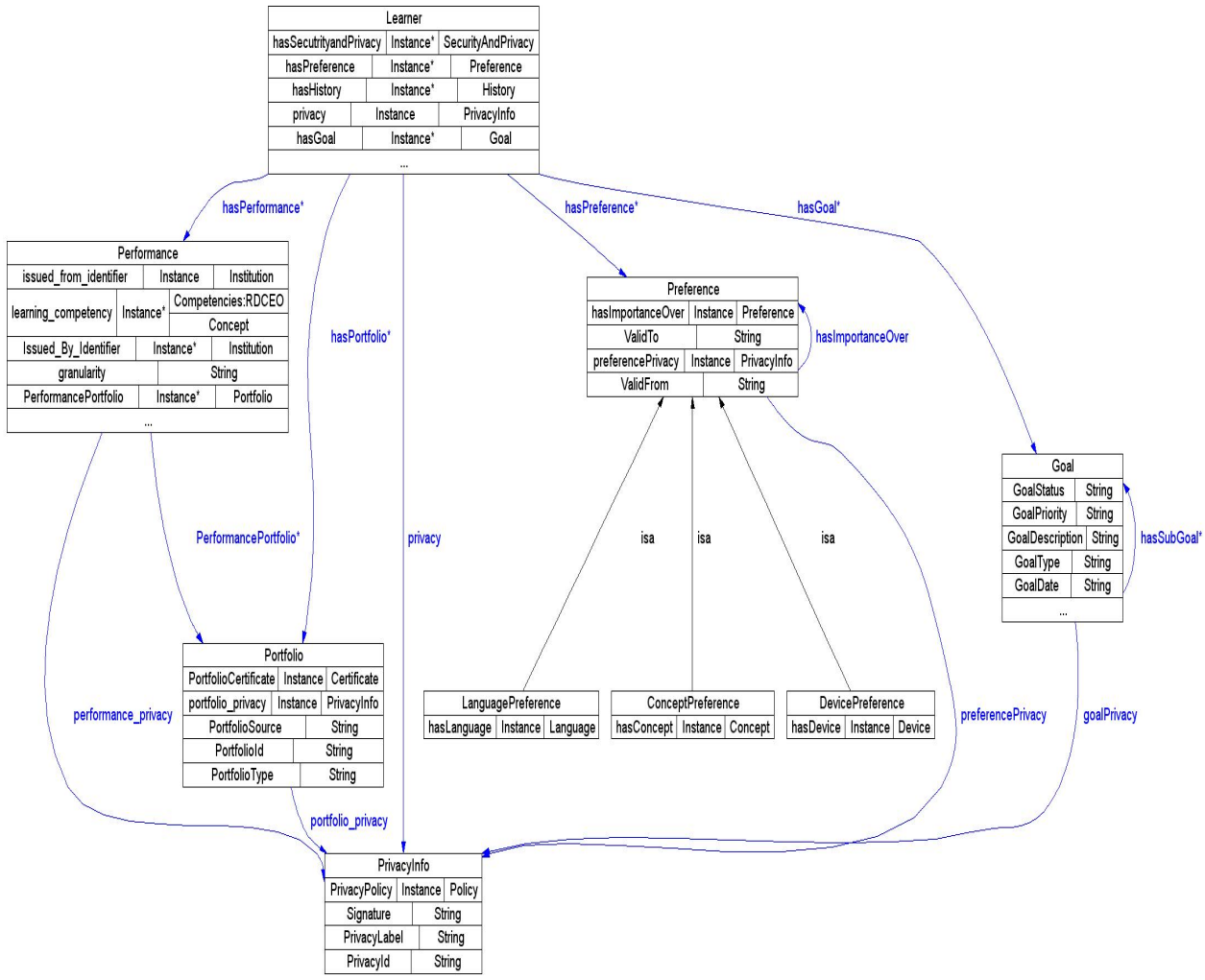


Figura 6. 1-Ontologia per il learner

Per quel che riguarda la rappresentazione delle mappe di competenza, l'ontologia che le rappresenta dovrà contenere tutte le informazioni per poter risalire, dati gli obiettivi, agli argomenti del dominio d'apprendere. L'ontologia rifletterà ovviamente la struttura organizzativa, ad esempio nel caso specifico della tesi è quella del protocollo informatico, dove si può pensare di strutturarla in base ai ruoli e funzioni che vengono assegnati agli utenti del protocollo. In questa ottica, quindi, è importante trovare una rappresentazione che integra le ontologie di dominio e le mappe di competenza al fine di effettuare una selezione sulle classi di Learning Object che meglio si adattano agli obiettivi formativi descritte in un insieme di mappe di competenza. Anch'esse, dunque, ontologie legate alle relazioni fra competenze associate a un ruolo e alle figure professionali e le classi di domini applicativi indispensabili per quel profilo

professionale. Attualmente, non esiste ancora uno standard per la rappresentazione delle gerarchie di competenze anche se esistono notevoli sforzi in tal senso. Un esempio sono le definizioni IMS Reusable Definition of a Competency or Educational Objective (RDCEO) e HR-XML Consortium Competencies Schema (HR-XML, 2003).

Per l'adattamento dinamico delle mappe di competenza in funzione delle necessità manifestate dagli utenti di consultare argomenti che non sono esplicitamente collegati alla competenza che dovrebbero acquisire.(dati, ad esempio, derivanti da attività di free-learning), potrebbe essere utile estendere il linguaggio di definizione di ontologie con la teoria della probabilità. Infatti, incorporare metodi per la rappresentazione di fenomeni probabilistici all'interno di linguaggi per la definizione di ontologie può fornire a tali linguaggi una maggiore potenza espressiva per quantificare il grado di sovrapposizione o di inclusione fra i concetti ed effettuare ricerche di similarità, ad esempio fra concetti simili. Se, ad esempio, per la rappresentazione delle mappe di competenze si utilizza il linguaggio OWL si potrebbe pensare di estendere tale linguaggio (Ding & Peng, 2004) con marcatori di tipo probabilistico che possono essere collegati ai concetti del dominio didattico e sviluppare regole di traduzione per convertire le ontologie con vere e proprie reti bayesiane.

Bibliografia

ALPAY L. (1989), *Developmental User Models*, Interactive Learning International, vol. 5, n. 2, pp 79-86.

ALPERT, S. R., SINGLEY, M. K., & FAIRWEATHER, P. G. (1999). Deploying Intelligent Tutors on the Web: An Architecture and an Example. *International Journal of Artificial Intelligence in Education*, 10**,** 183-197. Available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_10/alpert.html.

BACCHUS F.(1990), Representing and Reasoning with Probabilistic Knowledge, MIT Press 1990; J. Halpern, "An analysis of first order logics of probability", Artificial Intelligence 46:311-350.

BERNARAS A., I. LARESGOITI, AND J. CORERA . "*Building and Reusing Ontologies for Electrical Network Applications*". In Proceedings of the European Conference on Artificial Intelligence ECAI-96, 1996.

BOBROW, DANIEL G., AND TERRY WINOGRAD,(1977) *"An Overview of KRL, A Knowledge Representation language"*, Cognitive Science, V. 1, No. 1.

BOYLE, C., & ENCARNACION, A. O. (1994). MetaDoc: an adaptive hypertext reading system. *User Modeling and User-Adapted Interaction*, **4** (1), 1-19.

BRAILSFORD, T. J., STEWART, C. D., ZAKARIA, M. R., & MOORE, A. (2002). Autonavigation, links, and narrative in an adaptive Web-based integrated learning environment. *Proceedings of World Wide Web 2002 Conference,* May 7-11, 2002. Honolulu, HI.

BROWN, J. S., & BURTON, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2**,** 155-192.

BROWN, J. S.; BURTON, R. R.; BELL, A. G.(1975). *SOPHIE: a step towards a reactive learning environment.* International Journal of Man-Machine Studies, Vol. 7, pp. 675-696.

BRUSILOVSKY, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), 87-129.

BRUSILOVSKY, P. (1999). *Adaptive and Intelligent Technologies for Web-based Education.* Künstliche Intelligenz, (4), 19-25. Available online at http://www2.sis.pitt.edu/~peterb/papers/KIreview.html.

BRUSILOVSKY, P. (2000). Concept-based courseware engineering for large scale Web-basededucation. In G. Davies, & C. Owen (Eds.), *Proceedings of WebNet'2000, World Conference of the WWW and Internet*, Oct. 30 - Nov. 4, 2000. San Antonio, TX, AACE. pp. 69-74.

BRUSILOVSKY, P., & MILLER, P. (2001). Course Delivery Systems for the Virtual University. In T. Tschang, & T. Della Senta (Eds.) *Access to Knowledge: Information Technologies and the Emergence of the Virtual University New,* (pp. 167-206.). Amsterdam: Elsevier Science. Available online at http://www2.sis.pitt.edu/~peterb/papers/UNU.html.

BRUSILOVSKY, P., SCHWARZ, E., & WEBER, G. (1996). *ELM-ART: An intelligent tutoring system on Wold Wide Web*. I Frasson, C., Gauthier, G., & Lesgold, A. (Ed.), Intelligent Tutoring System (Lecture Notes in Computer Science, Vol. 1086). Berlin: Sprinter Verlag. 261-269.

BRUSILOVSKY, P., SCHWARZ, E., & WEBER, G. (1996a). ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.) *Third International Conference on Intelligent Tutoring Systems, ITS-96* (Vol. 1086, pp. 261-269). Berlin: Springer Verlag, Available online at http://www.contrib.andrew.cmu.edu/~plb/ITS96.html

BRUSILOVSKY, P., SCHWARZ, E., & WEBER, G. (1996b). A tool for developing adaptive electronic textbooks on WWW. In H. Maurer (Ed.), *Proceedings of WebNet'96, World Conference of the Web Society*, (pp. 64-69). October 15-19, 1996. San Francisco, CA, AACE. Online at http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html.

BRUSILOVSKY, P., HENZE, N., & MILLÁN, E. (Ed.). (2002). *Proceedings of the workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*. Málaga, Spain: University of Malaga.

CALVANI A. (2002), E-learning: tipologie e criticità nel contesto universitario, Form@re per la formazione in rete, Marzo, Editoriale, http://www.formare.erickson.it/archivio/marzo_aprile/editoriale.html.

CAPUANO N., M. MARSELLA, S. SALERNO (2000). ABITS: An Agent Based Intelligent Tutoring System for Distance Learning. Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems. ITS 2000, Montreal, Canada, 2000.

CARBONELL J.R. (1970), *AI in CAI: An artificial Intelligence Approach to Computer Aided Instruction*, IEEE Transaction on Man Machine Systems, vol. 11, n. 4, pp 190-202.

CARR, B.AND GOLDSTEIN I. (1997). Overlay: a Theory of Modeling for Computer-aided Instruction, Teical Report, AI Lab Memo 406, MIT.

CLANCEY, W. J. (1979). Tutoring rules for guiding a case method dialog. *International Journal on the Man-Machine Studies*, 11**,** 25-49.

DE BRA, P. M. E. (1996). Teaching Hypertext and Hypermedia through the Web. *Journal of Universal Computer Science,* 2(12), 797-804. Available online at http://www.iicm.edu/jucs_2_12/teaching_hypertext_and_hypermedia.

DE BRA, P., & CALVI, L. (1998). AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*, 4 115-139.

DE BRA, P., & RUITER, J.-P. (2001). AHA! Adaptive hypermedia for all. In W. Fowler, & J.Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 262-268.

DING, PENG (2004) A probabilistic extension to ontology language OWL, *Proceedings of the 37th Hawaii international Conference on System Sciences, 2004.*

GIUGNO R., T. LUKASIEWICZ.(2002) "P-SHOQ(D): A Probabilistic Extension of SHOQ(D) for Probabilistic Ontologies in the Semantic Web"*, Lecture Notes in Artificial Intelligence*, Springer, 2002.

GOMEZ-PÉREZ A., N. JURISTO, AND J. PAZOS (1995). Evaluation and assessment of knowledge sharing technology. In N. J. Maers, editor, *Towards Very Large Knowlwdge Bases – Knowledge Building and Knowledge Sharing 1995*, pages 289-296. IOS Press, Amsterdam, 1995.

GONZALEZ C., M. A., SUTHERS, D., & ESCAMILLA DE LOS SANTOS, J. G. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, 13(2-4), 261-297

GRIGORIADOU, M., PAPANIKOLAOU, K., KORNILAKIS, H., & MAGOULAS, G. (2001). INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In P. D. Bra, P. Brusilovsky, & A. Kobsa (Eds.), *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia,* July 14, 2001. Sonthofen, Germany, Technical University Eindhoven. - pp. 13-24.

GRUBER T.R (1995): "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", International Journal of Human-Computer Studies, 43(5/6), pp. 907-928.

HEIFT, T., & NICHOLSON, D. (2001). *Web delivery of adaptive and interactive language tutoring*. International Journal of Artificial Intelligence in Education, 12(4), 310-324.

HENZE, N., & NEJDL, W. (2001). *Adaptation in open corpus hypermedia*. International Journal of Artificial Intelligence in Education, 12(4), 325-350. Available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/henze.html.

HR-XML, Raccomendation 2003 http://ns.hr-xml.org/2_0/HR-XML-2_0/CPO/Competencies.pdf

IEEE P1484.12 Learning Object Metadata Working Group. LOM Approved Working Draft 4.
(http://ltsc.ieee.org/doc/wg12/LOM_WD4.PDF).

IMS Learning Resource Meta-Data Best Practice and Implementation Guide, 2001, version 1.2.1,Final Specification (http://www.imsproject.org/specifications.html).

LAROUSSI, M., & BENAHMED, M. (1998). Providing an adaptive learning through the Web case of CAMELEON: Computer Aided MEdium for LEarning on Networks. In C. Alvegård (Eds.), *Proceedings of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering,* June 15-17, 1998. Göteborg, Sweden, - pp. 411-416.

LIN, D.-Y .M . (2003) Hypertext for the aged : effects of text topologies. *Computers in Human Behavior,* 19 (2003), 201-209.

MELIS, E., ANDRÈS, E., BÜDENBENDER, J., FRISHAUF, A., GOGUADSE, G., LIBBRECHT, P., POLLET, M., & ULLRICH, C. (2001). ActiveMath: A web-based learning environment. *International Journal of Artificial Intelligence in Education*, 12(4), 385-407

MERCERON, A., & YACEF, K. (2003). A Web-based tutoring tool with mining facilities to improve learning and teaching. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *AI-Ed'2003* (pp. 201-208). Amsterdam: IOS Press.

MITROVIC, A. (2003). *An Intelligent SQL Tutor on the Web*. International Journal of Artificial Intelligencein Education, 13(2-4), 171-195.

MITSUHARA, H., OCHI, Y., KANENISHI, K., & YANO, Y. (2002). *An adaptive Web-based learning system with a free-hyperlink environment.* In P. Brusilovsky, N. Henze, & E. Millán (Eds.), Proceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'2002 (pp. 81-91). May 28, 2002. Málaga, Spain.

MURRAY, T. (2003). MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. *International Journal of Artificial Intelligence in Education*, 13(2-4), 197-231.

NECHES, R., FICHES, R. E., FININ, T., GRUBER, T. R., PATIL, R., SENATOR, T. AND SWARTOUT, W. (1991). *Enabling Technology for Knowledge Sharing*. AI Magazine, 12:36-56.

NONAKA I. E TAKUECHI H., (1995). The Knowledge Creating Company. Oxford University Press, 1995.

ODA, T., SATOH, H., & WATANABE, S. (1998). Searching deadlocked Web learners by measuring similarity of learning activities. *Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98),* August 16-19, 1998. San Antonio, TX. Available online at http://www.sw.cas.uec.ac.jp/~watanabe/conference/its98workshop1.ps.

IEEE PAPI. IEEE P1484.2.5/D8,(2002). Draft standard for learning technology. public and private information (papi) for learners (papi learner). Available at: http://jtc1sc36.org/doc/36N0179.pdf.

KOLB DAVID A. (1975) Organizational Psychology, A Book of Readings.

KOPER E.J. (2001), Modeling units of study from a pedagogical perspective, http://eml.ou.nl/introduction/docs/ped-metamodel.pdf

PEYLO, C. (Ed.). (2000). *Proceedings of the International Workshop on Adaptive and Intelligent Webbased Education Systems held in conjunction with 5th*

International Conference on Intelligent Tutoring Systems (ITS'2000). Ösnabrück: Institute for Semantic Information Processing, University of Ösnabrück.

POOLE D.(1993), "Probabilistic Horn Abduction and Bayesian Networks", *Artificial Intelligence*, 64:81-129.

RDCEO, IMS Reusable Definition of a Competency or Educational Objective: www.imsproject.org

RITTER, S. (1997). Pat Online: A Model-tracing tutor on the World-wide Web. In P. Brusilovsky, K. Nakabayashi, & S. Ritter (Eds.), *Proceedings of Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education,* (pp. 11-17). 18 August 1997. Kobe, Japan, ISIR. Online http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Ritter/Ritter.html.

ROMERO, C., VENTURA, S., BRA, P. D., & CASTRO, C. D. (2003). Discovering prediction rules in AHA! courses. In P. Brusilovsky, A. Corbett, & F. d. Rosis (Eds.), *9th International User Modeling Conference* (Vol. 2702, pp. 25-34). Berlin: Springer Verlag.

SCHWARZ, E. (1998). Self-organized goal-oriented tutoring in adaptive hypermedia environments. In B. P. Goettl, H. M. Halff, C. L. Redfield, & V. J. Shute (Ed.), *4th International Conference, ITS-98* (pp. 294-303). Berlin: Springer-Verlag.

SMITH, A. S. G., & BLANDFORD, A. (2003). MLTutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System. *International Journal of Artificial Intelligence in Education*. 13(2-4), 233-260. Available online at http://www.cogs.susx.ac.uk/ijaied/abstracts/Vol_13/smith.html.

SOLLER, A., & LESGOLD, A. (2003). A computational approach to analysing online knowledge sharing interaction. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *AI-ED'2003* (pp. 253-260). Amsterdam: IOS Press.

SPECHT, M., & KLEMKE, R. (2001). ALE - Adaptive Learning Environment. In W. Fowler, & J. Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 1155-1160.

SPECHT, M., KRAVCIK, M., KLEMKE, R., PESIN, L., & HÜTTENHAIN, R. (2002). Adaptive Learning Environment (ALE) for Teaching and Learning in WINDS. In *Second*

International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002) (Vol. 2347, pp. 572-581). Berlin: Springer-Verlag.

STOCKLEY D. (2002), E-learning Definition and Explanation (Elearning, Online Training, Online Learning), http://derekstockley.com.au/elearning-definition.html.

STRACCIA U., (2001) "Reasoning with Fuzzy Description Logic", *Int. Journal of Artificial Intelligence Research,* 14:136-166, 2001.

SWARTOUT W.. *Ontologies*. IEEE Transactions on Intelligent Systems, pages 18--19, 1999. 34.

TSAI S., MACHADO P.(2002), E-Learning, Online Learning, Web-based Learning, or Distance Learning: Unveiling the Ambiguity in Current Terminology, InkiTiki Corporation, Island of Kauai, Hawaii, http://www.elearnmag.org/subpage/sub_page.cfm?section=3&list_item=6&page=1

USCHOLD M., M. KING. Toward a methodology for building ontologies, AIAI-TR-183, Giugno 1995.

USCHOLD M., M. KING. Building ontologies: Towards a unified method, AIAI-TR-197, Settembre 1996.

VARISCO B.M. (2002), Costruttivismo Socio-Culturale, Carocci, Roma.

W3C Recommendation 10 Feb 2004. OWL Web Ontology Language Guide. Smith, Welty, McGuinness, eds. Online: http://www.w3.org/TR/owl-guide/.

W3C Recommendation 10 Feb 2004. XML Extensible Markup Language. 1.0 (Third Edition). Online: http://www.w3.org/TR/2004/REC-xml-20040204/.

W3C Recommendation 2 Maggio 2001. XML Schema Part 1: Structures. Online: http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

W3C Resource Description Framework (RDF).Online:http://www.w3c.org/RDF.

WARENDORF, K., & TAN, C. (1997). ADIS - An animated data structure intelligent tutoring system or Putting an interactive tutor on the WWW. In P. Brusilovsky, K. Nakabayashi, & S. Ritter (Eds.), *Proceedings of Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education,* (pp. 54-60). 18 August 1997. Kobe, Japan, ISIR.Online: http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Warendorf/Warendorf.html.

WEBER, G., KUHL, H.-C., & WEIBELZAHL, S. (2001b). Developing adaptive internet based courses with the authoring system NetCoach. In P. D. Bra, P. Brusilovsky, & A. Kobsa (Eds.), *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia,* July 14, 2001. Sonthofen, Germany, Technical University Eindhoven. - pp. 35-48, online at http://wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf.

WEBER, G. AND MÖLLENBERG, A. (1994) *'ELM-PE: A knowledge-based programming environment for learning LISP'. Proceedings of ED-MEDIA '94*, Vancouver, Canada, pp. 557-562.

WEBER, G., & BRUSILOVSKY, P. (2001). *ELM-ART: An adaptive versatile system for Web-based instruction.* International Journal of Artificial Intelligence in Education. 12(4), 351-384. http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html.

WOOLF, B. (1992). AI in Education. *Encyclopedia of Artificial Intelligence*, Shapiro, S., ed., John Wiley & Sons, Inc., New York, pp. 434-444.

ZORAN JEREMIC, VLADAN DEVEDŽIC,(2004) Design Pattern ITS: Student Model Implementation August 30 - September 01, 2004, The 4th IEEE International Conference on Advanced Learning Technologies (ICALT'04) .