



Università degli Studi di Salerno

DIEII

Dipartimento di Ingegneria Elettronica e Ingegneria Informatica

Dottorato di Ricerca in Ingegneria dell'Informazione
IX Ciclo – Nuova Serie

TESI DI DOTTORATO

**Modelli e metodologie
innovative per una
soluzione di e-learning e
knowledge management**

CANDIDATO: **SERGIO MIRANDA**

TUTOR: **PROF. GIULIANO GARGIULO**

COORDINATORE: **PROF. ANGELO MARCELLI**

Anno Accademico 2009 – 2010

*Al Prof. Saverio Salerno e al Prof. Matteo Gaeta
che mi hanno dato l'opportunità
di svolgere il lavoro qui descritto.*

*Alla mia famiglia
a cui ho scelto di togliere tempo
per dedicarmi a questo.*

Sommario

1	Capitolo 1	7
1	1 Introduzione.....	7
2	Capitolo 2	10
2	2 Il contesto di riferimento: la piattaforma IWT	10
2.1	2.1 Web Semantico e IWT.....	13
2.2	2.2 Approccio semantico per la personalizzazione dei corsi	15
2.3	2.3 Servizi Web 2.0 e e-Learning 2.0 in IWT	17
2.4	2.4 User model	19
2.5	2.5 Knowledge Model	21
2.5.1	2.5.1 Le ontologie di IWT	22
2.6	2.6 Dai concetti alle learning activities.....	24
2.6.1	2.6.1 L'attività di verifica	25
2.6.2	2.6.2 Il modello dell'unità didattica	26
2.7	2.7 La creazione del corso personalizzato	27
2.7.1	2.7.1 Generazione del Learning Path.....	30
3	Capitolo 3	36
3	3 Il lavoro svolto.....	36
3.1	3.1 Come è stato modificato l'algoritmo di LIA di IWT	37
3.1.1	3.1.1 Determinazione delle Milestone.....	42
3.1.2	3.1.2 Generazione del Pre-Test.....	43
3.1.3	3.1.3 Generazione della Presentazione.....	44
3.1.4	3.1.4 La Complessità degli Algoritmi	55
3.2	3.2 La sperimentazione condotta per verificare la validità dell'approccio IWT nell'e-learning.....	56
3.2.1	3.2.1 Sperimentazione di IWT con gli studenti universitari.....	56
3.2.2	3.2.2 Sperimentazione di IWT nella grande distribuzione.....	62
3.3	3.3 Verifica dei contenuti.....	65

3.3.1	L’Aritmetica degli intervalli.....	66
3.3.2	La problematica.....	67
3.3.3	Il modello utente.....	67
3.3.4	Il modello contenuti.....	68
3.3.5	La funzione di tolleranza.....	68
3.3.6	La funzione di copertura	70
3.3.7	La verifica da effettuare	71
3.3.8	L’aritmetica degli intervalli per ottimizzare la verifica.....	71
3.4	Aggiornamento del profilo utente	73
3.4.1	Aggiornamento delle competenze di un discente durante la fruizione di un corso personalizzato.....	73
3.5	Aggiornamento delle preferenze	80
3.6	Estrazione dell’ontologia	81
3.6.1	Background e stato dell’arte su SCORM	82
3.6.2	e-Learning e personalizzazione	83
3.6.3	Ontology learning.....	84
3.6.4	Il modello proposto per l’estrazione di ontologie	85
3.7	Estrazione automatica di metadata	93
3.7.1	La problematica.....	94
3.7.2	I metadata.....	95
3.7.3	Stato dell’arte	96
3.7.4	Estrazione automatica dei metadata.....	97
3.7.5	Sperimentazione	105
4	Capitolo 4.....	109
4	Conclusioni e sviluppi futuri.....	109
5	Riferimenti	112

Capitolo 1

Introduzione

Le attività di ricerca svolte hanno focalizzato l'attenzione su **IWT** – Intelligent Web Teacher – un'innovativa piattaforma per l'apprendimento a distanza e, più in generale, per la rappresentazione e la gestione della conoscenza nata grazie alle esperienze nei progetti di ricerca nazionali ed internazionali del Polo di Eccellenza sulla Conoscenza dell'Università degli Studi di Salerno.

IWT offre, come principale caratteristica distintiva, la capacità di erogare corsi personalizzati che, tenendo conto delle caratteristiche degli utenti e garantendo flessibilità al livello dei contenuti e dei modelli di apprendimento, risultano più efficaci ed efficienti dei percorsi formativi statici di e-learning “classico”.

La capacità di erogare corsi personalizzati si basa fondamentalmente su una sofisticata modellazione sia della conoscenza dei domini di interesse e dei contenuti, effettuata grazie ad ontologie e metadata, sia delle competenze acquisite e delle preferenze didattiche dell'utente, per mezzo di un profilo utente che viene costantemente aggiornato.

La validità dell'approccio IWT è stata avvalorata da un attento confronto con soluzioni, metodologie e modelli allo stato dell'arte e dai risultati ottenuti in molteplici attività di sperimentazione condotte in ambito universitario e aziendale.

Il risultato principale è legato ad un confronto tra le performance ottenute in termini di competenze acquisite dagli utenti erogando a gruppi di utenti disgiunti, ma sui medesimi argomenti, sia corsi semplici che corsi personalizzati.

Sulla sperimentazione condotta sono state fatte riflessioni interessanti che sembrano convincere che l'approccio della personalizzazione di IWT porti a risultati migliori rispetto

all'approccio tipico dell'e-learning, ovvero all'erogazione di percorsi sequenziali statici di contenuti. Attraverso la personalizzazione si riescono a coinvolgere in modo efficace anche utenti che, per esempio, abbiano già un bagaglio di competenze medio-alto e che, tipicamente, non risultano essere propensi a seguire percorsi di apprendimento in e-learning.

Per contro, questo approccio pur se totalmente automatizzato, richiede un impegno in fase di start-up per definire correttamente i profili degli utenti, disegnare l'ontologia del dominio oggetto di formazione, intesa come una corretta rappresentazione semantica dei concetti e delle relazioni tra questi, definire i metadati delle risorse didattiche a disposizione, ovvero indicizzarle e descriverle in modo completo per poterle poi utilizzare nel modo più appropriato possibile.

Le attività di ricerca sono state orientate allo studio sullo stato dell'arte, allo studio di modelli esistenti, alla ricerca di soluzioni, alla definizione di possibili metodologie capaci di produrre miglioramenti su questi aspetti chiave legati sia alla rappresentazione della conoscenza in termini di ontologie e di metadati, che alla rappresentazione dei profili utente in termini di competenze sul dominio oggetto di apprendimento e di preferenze relative alle modalità di utilizzo e fruizione di contenuti.

Ovviamente l'approccio non è stato meramente applicativo, ma si è data attenzione alla ampia letteratura di settore per cercare di studiare problemi e soluzioni esistenti e definire qualcosa (anche dal punto di vista algoritmico) che sia in qualche modo valido e di interesse per la comunità scientifica.

In sintesi, avendo individuato dei modelli con cui delineare ed aggiornare i profili utente e delle metodologie computazionalmente efficienti attraverso cui comprendere se, nell'ambito di una specifica tematica, ci siano o meno contenuti sufficienti a rispondere alle esigenze degli utenti, l'idea è quella di concentrare l'attenzione su due aspetti fondamentali che sono l'estrazione automatica di ontologie e la definizione automatica di metadati per i contenuti.

La situazione maggiormente ricorrente è infatti quella in cui il materiale da impiegare per costruire i percorsi di apprendimento sia già disponibile, per cui, per poter sfruttare i benefici della personalizzazione, occorre costruire ontologie e metadati a posteriori (e generalmente a mano).

Per automatizzare queste operazioni sono stati definiti dei metodi attraverso i quali è possibile estrarre ontologie direttamente dai contenuti (diverse tipologie e standard) e si sta lavorando a come definire contestualmente i metadata, ovvero fornire una descrizione formale dei contenuti in termini di parametri sia tecnologici che pedagogici e indicizzarli in termini semantici attraverso i concetti estratti e riportati nelle ontologie.

In letteratura sono presenti molteplici approcci che riescono a caratterizzare in automatico gli aspetti tecnologici relativi al contenuto (formato, dimensioni, requisiti HW e SW per poterli visualizzare), rari ed incompleti sono invece i tentativi di descrizione degli aspetti pedagogici impiegati da standard riconosciuti come IMS Learning Object Metadata (tipologia di risorsa didattica, densità semantica, difficoltà, tempo per la comprensione, livello di interazione, etc.). Mettendo insieme elementi di teoria dell'informazione, con modelli di learning si è cercato di proporre un approccio alla risoluzione di questo problema. Approccio che, al momento, sembra funzionare in una sperimentazione su un archivio di circa 2000 learning object per l'insegnamento della matematica e dell'informatica e che, a breve, verrà esteso al di fuori dell'ambito e-learning e nell'ottica moderna della pubblicazione delle informazioni sul web secondo la logica dei Linked Data. Logica a cui stanno dando attenzione enti come IEEE, BBC, Governo USA, etc., che, ovviamente, hanno grandi esigenze di catalogazione automatica di documenti.

Capitolo 2

Il contesto di riferimento: la piattaforma IWT

La piattaforma IWT è un prodotto italiano nato dalle esperienze di ricerca del Polo di Eccellenza sulla Conoscenza dell'Università degli Studi di Salerno condotte in Europa e capaci di inglobare tecnologie all'avanguardia, idee innovative e standard emergenti nel settore dell'e-learning.

Il Polo di Eccellenza sulla Conoscenza è un'aggregazione virtuale costituita da quattro attori dalla genesi diversa, ovvero DIIMA, il Dipartimento di Ingegneria dell'Informazione e Matematica Applicata dell'Università di Salerno, due consorzi di ricerca pubblico-privati a prevalente partecipazione dell'Università di Salerno, CRMPA, Centro di Ricerca in Matematica Pura e Applicata, CEMSAC, Centro di Eccellenza su Metodi e Sistemi per Aziende Competitive ed infine MOMA, una società spin-off che, nata come società consortile, convertita in S.r.l. e, di recente, trasformata in S.p.A., è una società di proprietà degli stessi ricercatori che lavorano al Polo.

La sinergia degli attori indicati ha consentito da un lato di consolidare esperienze nella ricerca a livello nazionale e internazionale in diversi settori mentre dall'alto di confrontarsi con competitor di mercato e valutare nel vero senso della parola l'innovazione prodotta con la ricerca.

Con questa logica virtuosa è nata la piattaforma IWT, ideata, realizzata in forma prototipale, sperimentata in contesti accademici, scolastici ed aziendali, estesa con spunti sempre innovativi, integrata, ingegnerizzata ed industrializzata nell'arco degli ultimi dodici anni ed ancora oggetto di studi, sperimentazioni e continue evoluzioni grazie all'energia trainante del leader del Polo ed alle entusiasmi

esperienze dei progetti nazionali ed europei che rappresentano una sfida quotidiana costante.

La rete di conoscenze del Polo di eccellenza è punto di riferimento e guida per le evoluzioni della piattaforma IWT. Sono numerose le esperienze di ricerca e sviluppo che hanno determinato l'evoluzione di aspetti, che hanno portato alla realizzazione di moduli, che hanno sperimentato la validità dei modelli e delle tecnologie adottate. A queste sono seguite attività di ingegnerizzazione ed industrializzazione che hanno trasformato IWT in un prodotto per l'e-learning.

In particolare nel corso degli ultimi anni, il contributo è stato dato dalla sintesi di tecnologie e idee provenienti da importanti progetti nazionali e internazionali, tra cui citiamo:

- **DiSiMa** (MURST) – Area di ricerca: Learning & Knowledge – Il suo obiettivo è stato la realizzazione di un software prototipale di simulazione e tutoring intelligente che, in un ambiente totalmente integrato, permetta la formazione a distanza e la simulazione di scelte strategiche secondo modelli di business game centrati sulla gestione delle imprese commerciali complesse.

- **InTraSys** (ESPRIT) – Area di ricerca: Learning & Knowledge – Il suo obiettivo è stato la realizzazione di una piattaforma di comunicazione aperta ed integrata, avente come finalità la trasmissione e la condivisione di conoscenze specialistiche tra gli utilizzatori ed i professionisti di un determinato settore.

- **InTraServ** (IST Quinto Programma Quadro) – Area di ricerca: Learning & Knowledge – Il suo obiettivo è stato sperimentazione e valutazione di una soluzione web-based innovativa per la formazione intelligente a distanza, finalizzata all'aggiornamento professionale dei manager delle PMI.

- **Diogene** (IST Quinto Programma Quadro) – Area di ricerca: Learning & Knowledge – Il suo obiettivo è stato la creazione di un ambiente Web di brokering per la formazione individuale, adatto al supporto degli studenti, con strategie di apprendimento dinamiche, durante l'intero ciclo di formazione, dalla definizione degli obiettivi fino alla valutazione dei risultati. [<http://www.diogene.org>]

- **m-Learning** (IST Quinto Programma Quadro) – Area di ricerca: Learning & Knowledge – Il suo obiettivo è stato favorire la “formazione permanente”, mediante lo studio delle possibilità offerte

dai dispositivi mobili (cellulari, PC palmari e tascabili) e dalle reti di comunicazioni mobili ad essere veicoli di divulgazione di informazioni e di funzionalità didattiche. [<http://www.m-learning.org>]

- **Genesis** (IST Quinto Programma Quadro) – Area di ricerca: Web Technologies – Il suo obiettivo è stato supportare in modo non invasivo le esigenze di coordinamento e di cooperazione tipiche del Distributed Software Engineering, superando le barriere legate alla distribuzione geografica dei team di lavoro. [<http://www.genesis-ist.org>]

- **GeCoSan** (MIUR – ENEA) – Area di ricerca: Web Technology – Il suo obiettivo è stato fornire una soluzione operativamente efficace al problema della circolazione dell'informazione nel settore sanitario ovvero ai contesti dell'ECM (Educazione Continua in Medicina), con introduzione di tecniche di erogazione sul web di percorsi formativi personalizzati e dell'EBM (Medicina basata sull'Evidenza), con lo studio di un sistema di organizzazione e raccolta di dati provenienti dalla ricerca clinica internazionale e dall'esperienza operativa nazionale che ne consenta l'erogazione e la rappresentazione immediata e personalizzata. [<http://www.crimpa.it/gecosan>]

- **ELeGI** (IP Sesto Programma Quadro) – Area di ricerca: Grid, Learning & Knowledge – Il suo obiettivo consiste in studio, definizione e sperimentazione di una piattaforma software basata su tecnologie GRID per la realizzazione degli scenari di learning basati su modelli didattici di tipo esperienziale, collaborativo e contestualizzato. [<http://www.elegi.org>]

- **Sim Learning** (MIUR) – Area di ricerca: Learning & Knowledge – Il suo obiettivo è acquisire e validare conoscenze nei domini relativi alle metodologie ed alle tecnologie didattiche e simulative utili al successivo sviluppo di framework per il supporto di applicazioni e servizi di distance-learning secondo il modello didattico induttivo-esperienziale attraverso Esperimenti Scientifici Virtuali, simulazioni e Business-Game.

- **e-Science** (MIUR) – Area di ricerca: Grid, Learning & Knowledge – Il suo obiettivo è la creazione dell'infrastruttura di e-Science, basata su tecnologie GRID, per la cooperazione scientifica tra l'Università di Salerno ed altre strutture di ricerca italiane e straniere, per lo sviluppo di tematiche di ricerca nei settori dei Metodi

di Ottimizzazione e Simulazione, dell'Ingegneria del Software, dell'e-Learning e delle Architetture e Tecnologie GRID.

Il Polo di eccellenza, dunque ha svolto o svolge importanti attività di ricerca e sviluppo che hanno concorso e concorrono tuttora alle evoluzioni della piattaforma IWT, in termini di modelli, di tecnologie, di funzionalità semplici ed avanzate. I piani relativi ai rilasci delle nuove versioni della piattaforma sono infatti delineati proprio sotto la supervisione del Polo.

La validità scientifica degli aspetti realizzati è stata confermata da alcune decine di pubblicazioni e riconoscimenti di cui il Polo può attualmente vantarsi.

Di recente approvazione ci sono due progetti europei del Settimo Programma Quadro: **Aristotele** e **Alice** che articolano ancora innovative evoluzioni del medesimo framework tecnologico di IWT.

Grazie a questi progetti il Polo è risultato **primo in Europa** nell'Area TEL, Technology Enhanced Learning, sia nel 2005 che nel 2010.

La relativa efficacia didattica di IWT è stata analizzata "sul campo" attraverso programmi di valutazione e sperimentazione effettuati su gruppi di utenti accuratamente selezionati, nell'ambito di alcuni dei su menzionati progetti (InTraServ, m-learning, Diogene, e-Science) o nell'ambito delle attività didattiche svolte presso atenei o scuole come supporto alle lezioni frontali.

2.1 Web Semantico e IWT

Il Web Semantico appare offrire interessanti opportunità nell'ambito dei sistemi educativi per soddisfare il principio dello "AAAL: Anytime, Anywhere, Anybody Learning" (apprendimento per tutti, in qualunque momento e in ogni luogo). In questo contesto, IWT rappresenta una soluzione innovativa per l'e-learning in grado di supportare docenti e progettisti didattici nella modellazione di domini di conoscenza in contesti educativi, oltretutto le competenze e preferenze degli utenti tramite un approccio semantico al fine di creare attività di apprendimento personalizzate e contestualizzate e per consentire agli utenti di comunicare, cooperare, e creare dinamicamente nuovi contenuti da distribuire e informazioni da

condividere come anche per supportare piattaforme didattiche per l'e-learning 2.0.

L'evoluzione delle tecnologie del Semantic Web (SW) nel corso degli ultimi anni ha reso i sistemi basati su ontologie un'interessante prospettiva cui guardare per supportare i processi di apprendimento. Nonostante ciò, in accordo con Mizoguchi e Bourdeau (2000 cfr.[40]), diverse questioni restano ancora da affrontare, quali l'esistenza di un profondo gap concettuale esistente tra i sistemi autore e i modelli di progettazione didattica; il fatto che gli applicativi di authoring non siano spesso né intelligenti né particolarmente facili da usare; il fatto che la conoscenza e i componenti integrati negli Intelligent Web-Based Education Systems (IW BES) siano raramente condivisibili e riusabili; l'esistenza di un gap tra i modelli di progettazione per l'organizzazione dei domini di conoscenza e le strategie di tutoring per l'adattamento dinamico del comportamento degli IW BES.

Inoltre, un problema importante deriva dalla mancanza di una rappresentazione canonica delle strutture di conoscenza (un dominio di conoscenza può sempre essere visto da differenti prospettive mostranti strutture diversificate). Come evidenziato da Ohlsson (1987 cfr. [47]), questo fatto ha così rilevanti implicazioni per i sistemi autore che esso può essere definito come il "*Principio di non-equifinalità dell'apprendimento*", secondo cui "*lo stato di conoscenza di un determinato dominio non corrisponde a un singolo ben definito stato cognitivo [...] la conoscenza può sempre essere rappresentata in differenti modi e da diverse prospettive [...] dunque, anche il processo di acquisizione del dominio ha vari, differenti, egualmente validi, stati finali*". Dovrebbe essere chiaro che esiste un gran numero di rappresentazioni di possibili percorsi di apprendimento e che il problema consiste nel trovare i contenuti e le modalità di presentazione appropriati per lo specifico profilo degli studenti cui ci si rivolge e per lo specifico obiettivo educativo che è stato definito (Baker, 2000 cfr. [10]).

Altra questione che deve essere affrontata è relativa alla rappresentazione formale dei domini tramite gli standard esistenti per la modellazione della conoscenza. Con ciò si introduce il problema della incompatibilità esistente spesso tra le diverse fonti eterogenee di metadati relativi ai diversi domini, che potrebbe essere evitato

utilizzando le ontologie quale schema concettuale unificante in un contesto di e-learning (Stojanovic et al., 2001, cfr. [59]).

Vari sistemi sono stati sviluppati per la gestione di risorse per l'apprendimento tramite tecnologie semantiche; generalmente i loro componenti sono relativi principalmente al ruolo degli utenti (docenti, studenti, autori, gruppi e sviluppatori); alle risorse educative; all'ambiente di apprendimento; alla sua interfaccia e alle funzionalità che esso offre. Un SWBES dovrebbe pertanto prevedere: ontologie, agenti pedagogici e di tutoring, servizi e strumenti semanticamente definiti.

IWT introduce un innovativo approccio all'e-learning, sviluppato presso che consente di modellare domini di conoscenza educativi, oltretutto le competenze e preferenze degli utenti tramite un approccio semantico al fine di creare attività di apprendimento personalizzate e contestualizzate e di consentire agli utenti di comunicare, cooperare e creare dinamicamente nuovi contenuti da distribuire e informazioni da condividere, come anche per integrare/supportare piattaforme per l'e-learning 2.0.

2.2 Approccio semantico per la personalizzazione dei corsi

Gli agenti intelligenti di IWT (Albano et al., 2006 e 2007; Capuano et al., 2008 cfr. [6], [7] e [19]) operano principalmente su tre aspetti: la conoscenza didattica, il modello studente, le procedure di pianificazione.

La conoscenza didattica è rappresentata a diversi livelli di astrazione. Il più basso è dato direttamente dai Learning Object (LO), ovvero il materiale didattico, I LO devono essere indicizzati per consentire al sistema di sapere di cosa trattano e come possono essere impiegati al meglio nei percorsi formativi.

Questo avviene grazie ad un secondo livello di astrazione che consente appunto la rappresentazione, ovvero il livello dei Metadata. Un Metadata è una collezione di attributi su un LO che ne descrive le caratteristiche quali il tipo (testo, simulazione, slide, questionario, ...), il contesto formativo in cui sia più indicato adottarlo (scuola

superiore, università, ...), la lingua, il livello di interattività con l'utente e così via.

Infine, un terzo livello di astrazione e quello delle ontologie impiegato per rappresentare i concetti del dominio in termini semantici e le relazioni tra essi. Un concetto di un dominio è, in termini semantici, il senso di un argomento del dominio che possa essere spiegato da uno o più LO. Le relazioni tipiche tra i concetti sono: *Has Part*, *Is Required By*, *Suggested Order*, che indicano rispettivamente una relazione gerarchica di decomposizione, un vincolo di prerequisito oppure un ordine suggerito durante l'apprendimento.

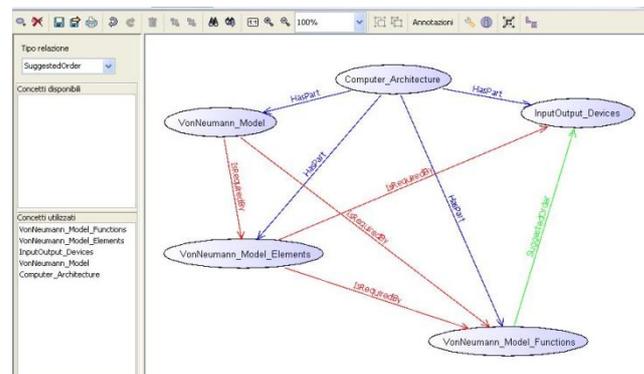


Figura 1. Tool di IWT per Dizionario e Ontologia di un dominio

Un'ontologia descrive la semantica sottostante ai contenuti attraverso una rappresentazione grafica dei concetti e delle relazioni tra essi (Gaeta et al., 2009 cfr. [26]). Infatti alcuni concetti sono richiesti da altri quando devono essere inclusi in un corso. Un corso è caratterizzato dal learning path, un percorso di concetti estratto dall'ontologia, ovvero da una serie di concetti del dizionario di dominio.

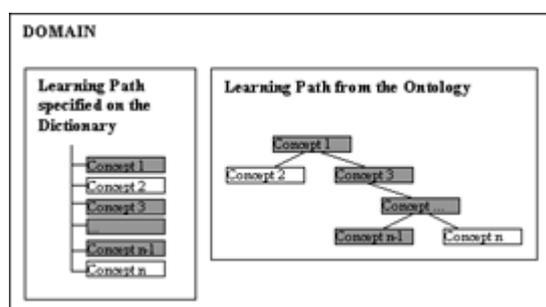


Figura 2. Il learning path è una sequenza di concetti del dominio

Quando la piattaforma IWT deve provvedere al delivery di un corso, individua una sequenza di contenuti che trattano i concetti presenti nel learning path.

Nel modello studente sono presenti tutte le informazioni sullo Stato Cognitivo e sulle Preferenze Didattiche. Le procedure di pianificazione di IWT sono capaci di creare automaticamente un corso in grado di soddisfare i requisiti di apprendimento di uno studente rispettando il suo stato cognitivo e le sue preferenze didattiche.

Dato un insieme di obiettivi didattici scelti dal docente direttamente sull'ontologia, IWT genera il miglior Learning Path per ogni studente. Studenti diversi si ritroveranno corsi diversi generati dal sistema.

2.3 Servizi Web 2.0 e e-Learning 2.0 in IWT

Come descritto in [21], IWT supporta l'integrazione di risorse, strumenti e servizi. Grazie a questa caratteristica, i principali aspetti del Web 2.0 e, quindi, dell'e-Learning 2.0 sono diventate presto estensioni di IWT. IWT è pertanto una soluzione e-Learning 2.0 completa poiché offre:

- **Approccio con l'utente al centro:** IWT dà attenzione all'utente nei processi di apprendimento e insegnamento;
- **Personalizzazione e contestualizzazione delle esperienze didattiche:** modello, processi e servizi di IWT servono a personalizzare le esperienze didattiche;
- Importanza alle teorie della didattica a distanza

- Importanza alla **semantica** e alla **conoscenza** nella formazione: la rappresentazione semantica è trasversale a tutti i servizi di IWT;

- **Vista orientata ai servizi.** Qui il modello di IWT è, ovviamente, basato sul concetto di Architettura Orientata ai Servizi (SOA)

IWT integra, ovviamente, funzionalità di LMS e LCMS e un ampio insieme di strumenti Web 2.0:

- **e-Portfolio.** Il portfolio memorizza in modo strutturato, informazioni personali, stili di apprendimento, stato cognitivo, traccia delle attività svolte o in corso, etc. Le informazioni possono essere a scelta pubbliche o private;

- **Blog.** In IWT gli studenti possono condividere idee su temi vari. I docenti possono rendere meglio esplicita la loro conoscenza su argomenti specifici;

- **Podcast.** Un modo semplice per catturare e far circolare contenuti audio/video, lezioni, seminari;

- **WIKI.** Un meccanismo molto diffuso per costruire conoscenza strutturate in modo collaborativo. Il WIKI di IWT può essere usato da gruppi di utenti per costruire contenuti da far valutare ai docenti o, addirittura, per costruire ontologie in modo collaborativo.

- **Social Network e Annotazioni.** Gli utenti possono tenersi in contatto in modo informale, creare gruppi in cui condividere gli stessi obiettivi o gli stessi interessi.

- **Forum semantici.** In cui gli utenti possono inviare domande e risposte e aggiungere annotazioni semantiche, commenti, giudizi in modo informale.

- **Aree condivise.** In cui gli utenti possono condividere contenuti, scaricare contenuti caricati da altri, aggiungere commenti, giudizi e feedback.

- **Feed RSS.** Per mezzo dei feed RSS, gli utenti possono pubblicare e raccogliere in modo semplice e veloce informazioni sullo stato, le attività in corso, gli interessi degli altri colleghi.

Tutti i contenuti generate dagli utenti grazie ai servizi Web 2.0 completano l'offerta di IWT (O'Really, 2005 cfr. [46]). Essi vengono

annotati semanticamente attraverso l'impiego di strutture semantiche al pari delle altre risorse. IWT ha così la possibilità di impiegare contenuti e servizi per creare il miglior percorso capace di far raggiungere ai discenti i loro obiettivi formativi.

2.4 User model

Il discente rappresenta naturalmente l'attore principale nella formazione a distanza e, quindi, è l'oggetto di principale attenzione da parte della piattaforma IWT. Il suo modello rappresentativo è costituito da uno stato cognitivo e da un insieme di preferenze di apprendimento.

Lo **stato cognitivo** rappresenta il livello di conoscenza raggiunto dal discente in un determinato momento ed è costituito dalla lista di concetti analizzati dal discente, a ciascuno dei quali è associato il rispettivo livello di apprendimento. Tale livello di apprendimento è rappresentato da un valore compreso nell'intervallo 0 (completa ignoranza del concetto), 1 (ottimo apprendimento del concetto) e viene valutato nella fase di testing. Un concetto verrà considerato appreso da un discente quando il livello di apprendimento supera un fissato valore di soglia θ .

E' necessario considerare nello stato cognitivo, per ogni concetto, non soltanto il livello di apprendimento raggiunto dal discente, ma anche il numero di volte in cui quel concetto gli è stato riproposto. Questo perché il metodo con cui un concetto viene proposto, viene scelto in base alle preferenze di apprendimento del discente ed alle preferenze di insegnamento del concetto, così come definite dal docente. Dopo ogni fase di test, vengono riproposti al discente i concetti il cui livello di apprendimento non è risultato sufficiente. Dopo che uno stesso concetto viene riproposto con risultati non positivi per n volte (con n opportunamente valutato), si sceglierà di cambiare la tipologia con cui quel concetto viene proposto. Naturalmente questo verrà fatto considerando la lista di preferenze di apprendimento del discente e le preferenze di insegnamento del concetto, come descritto nel capitolo successivo.

Occorre infine osservare che il livello di apprendimento di un concetto appreso dal discente non è costante nel tempo ma può essere soggetto a cambiamenti. Per cercare di correggere eventuali errori di valutazione, infatti, ogni volta che una fase di test registra un insuccesso per un determinato concetto proposto, vengono cercati tutti i concetti propedeutici appartenenti al patrimonio cognitivo del discente e si abbassa per ciascuno di essi di un ε il livello di apprendimento. Questo perché si può supporre che il mancato apprendimento di un concetto possa dipendere non soltanto dal modo in cui questo concetto viene proposto e studiato dal discente ma anche da eventuali lacune dovute e cattive valutazioni su concetti propedeutici. Quando, a seguito di tali modifiche, il livello di conoscenza di un concetto considerato già appreso, scende al di sotto del livello di soglia θ , il concetto viene riproposto al discente e sarà di nuovo soggetto a valutazione. Al fine di evitare che lo studente si veda ripresentati concetti “distanti” da quelli attualmente in fase di apprendimento, si può pensare di parametrizzare il valore di ε in funzione della distanza del concetto propedeutico da quello attuale.

Lo stato delle **preferenze di apprendimento** del discente è costituito da una lista in cui per ciascuna proprietà dell’attività didattica e per ciascun possibile valore di tale proprietà, viene indicato il livello di preferenza da parte del discente. Le proprietà rappresentano le caratteristiche dell’offerta didattica ovvero delle attività didattiche proposte analoghe a quelle utilizzate per le preferenze di insegnamento e per ciascuna di esse è definito un insieme di valori ammissibili. In fase di inizializzazione del modello, per ogni coppia (proprietà, valore) il discente dovrà indicare il suo grado di preferenza rappresentato ad esempio da un valore compreso nell’intervallo $[1, 10]$.

Le due liste (proprietà, valore) utilizzate per le preferenze di apprendimento del modello del discente e per le preferenze di insegnamento nell’ontologia devono essere sovrapponibili in maniera tale che in fase di presentazione, per ogni concetto sarà scelta l’attività didattica che meglio soddisfa sia le une che le altre.

I valori espressi dal discente non verranno considerati costanti ma potranno essere modificati nel corso della fase di apprendimento. A intervalli regolari, il sistema analizzerà l’attività svolta dal discente ed i risultati ottenuti. A seguito di ciò potrebbe risultare che i concetti

proposti secondo determinate coppie (*proprietà, valore*) vengono più facilmente appresi dal discente mentre le attività didattiche caratterizzate da altre coppie risultino più difficili. In questo caso si provvederà ad incrementare il livello di preferenza delle prime e ad abbassare quello delle seconde.

Occorre inoltre tener conto del fatto che il discente non sempre è in grado di definire o scegliere le proprie preferenze di apprendimento (ad esempio nel caso di studenti della scuola dell'obbligo o di corsi di aggiornamento per lavoratori poco scolarizzati). Per tali casi si dà la possibilità al docente di ridefinire o modificare in ogni momento lo stato delle preferenze di apprendimento di ogni discente. Inoltre, prima di effettuare in automatico, come descritto in precedenza, una modifica dei valori di tale stato, vengono fornite al docente tutte le informazioni relative allo stato attuale, al percorso di apprendimento seguito dal discente e alle modifiche consigliate dal motore. Sarà dunque il docente a scegliere se effettuare tale modifica, lasciare tutto inalterato oppure effettuare ulteriori modifiche. Naturalmente si applica una procedura di silenzio assenso alla modifica nel caso il docente, dopo un tempo prefissato, non sia riuscito ad analizzare le informazioni pervenutegli.

Infine, si può scegliere di definire le preferenze di apprendimento di ciascun discente, prima dell'inizio del corso, sulla base dei risultati forniti da un pre-test opportunamente predisposto dal docente o anche da un pool di esperti, che sulla base delle risposte fornite dal discente sia in grado di valutare le sue caratteristiche e tradurle in un specifiche preferenze di apprendimento. Tale pre-test può essere utilizzato anche indipendentemente dal contenuto del corso e quindi può essere utilizzato da più docenti per raggruppamenti di argomenti (ad esempio umanistici, scientifici, di aggiornamento ecc.).

2.5 Knowledge Model

Il Modello della Conoscenza è in grado di rappresentare in maniera intelligibile alla macchina le informazioni associate al materiale didattico disponibile. In Particolare IWT utilizza gli standard per la definizione dei metadata relativi ai Moduli Didattici e per la gestione delle esercitazioni e dei test.

IWT offre, inoltre, il supporto opzionale alla gestione della conoscenza ad un livello più alto di astrazione orientato ai concetti piuttosto che ai Moduli Didattici attraverso le ontologie. Le ontologie consentono di formalizzare domini cognitivi tramite la definizione di concetti e relazioni tra concetti e costituiscono, in sinergia con i metadata, la base per la valutazione automatica del discente e la personalizzazione dei percorsi didattici.

2.5.1 Le ontologie di IWT

Il modello di ontologie adottato in IWT è un modello astratto utilizzato per la rappresentazione del dominio della conoscenza. Esso identifica i concetti del dominio ed indica le relazioni esistenti tra i vari concetti.

- Le **relazioni** tra concetti adottate da IWT sono tre:
 - *HasPart(x,y)*: il concetto y è una componente del concetto x ;
 - *IsRequiredBy(x,y)*: la comprensione del concetto y ha come prerequisito quella del concetto x ;
 - *SuggestedOrder(x,y)*: per favorire l'apprendimento è preferibile studiare prima il concetto x e poi il concetto y .

La relazione *HasPart* è una relazione gerarchica, *IsRequiredBy* indica un vincolo nell'ordine in cui devono essere appresi due concetti, infine, la relazione *SuggestedOrder* suggerisce un ordine nello sviluppo del processo di apprendimento.

Le tre relazioni indicate sono solo un esempio di tutte le relazioni che è possibile definire (quelle generalmente più utilizzate nella descrizione di un dominio) ma domini particolari potrebbero anche richiedere la definizione di altre relazioni per la loro descrizione.

In generale, le relazioni tra concetti possono essere classificate in due **categorie**:

- *Relazioni di scomposizione*, sono relazioni usate per suddividere un concetto di alto livello in concetti a più basso livello;
- *Relazioni d'ordine*, sono relazioni che stabiliscono ordini parziali nell'insieme dei concetti.

Per quanto riguarda le precedenze tra le varie relazioni, si può assumere che se un concetto c_1 precede un concetto c_2 (cioè esiste una relazione d'ordine tra i due) allora c_1 precede anche tutti i concetti ottenuti da una scomposizione di c_2 .

Dal punto di vista formale una ontologia è rappresentata da un multigrafo $G(V, A_1, \dots, A_n)$ con un insieme di nodi V che rappresentano i singoli concetti, ed un insieme di archi A_i per ogni relazione i .

Nel caso delle tre relazioni indicate in precedenza, un esempio di ontologia è mostrato nella figura seguente (dove HP indica *HasPart*, IR indica *IsRequiredBy* e SO indica *SuggestedOrder*).

L'ontologia così definita non dà informazioni circa le migliori modalità di trasferimento della conoscenza sui concetti. Diciamo che essa non è “contestualizzata” dove per contesto si intendono intuitivamente le caratteristiche dell'insieme dei discenti che devono apprendere il dominio in oggetto, le modalità di interazione con il materiale didattico relativo all'ontologia e gli obiettivi dell'esperienza educativa basata sull'ontologia.

Al fine di “contestualizzare” un'ontologia, ovvero di renderla utilizzabile in uno specifico contesto, è possibile associare ad ogni concetto (e, in presenza di una scomposizione, ad ogni concetto foglia) una lista di coppie (proprietà, valore) che prende il nome di **preferenze di insegnamento**. In tale lista le proprietà specificano, attraverso appositi valori, le modalità ottimali di trasferimento dei singoli concetti. Tali proprietà possono essere ad esempio: il *metodo didattico*, la *tipologia dell'attività*, il *livello di interattività*, ecc.

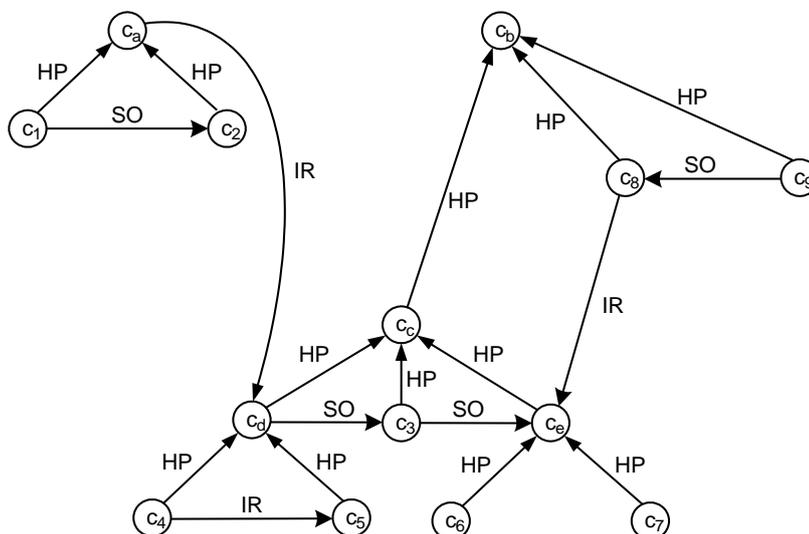


Figura 3. Un esempio di ontologia.

Per ciascuna proprietà è inoltre definito un insieme di valori ammissibili. Ad esempio, la proprietà *metodo didattico* avrà come valori ammissibili *induttivo*, *deduttivo*, ecc. mentre la proprietà *tipologia dell'attività* presenterà i valori *lettura di un testo*, *fruizione di un clip video*, *interazione con una simulazione*, *discussione con un pari*, *discussione con il docente*, ecc.

Per ogni coppia (proprietà, valore) ammissibile e per ciascun concetto, il docente può associare un livello di preferenza compreso nell'intervallo [1, 10]. In questo modo il docente indica quale secondo lui dovrebbe essere il modo più appropriato per presentare quel concetto ai discenti.

2.6 Dai concetti alle learning activities

L'attività didattica indica il modo col quale un determinato concetto viene proposto al discente. Essa può essere delle più svariate tipologie. Può ad esempio essere un testo esplicativo del concetto, una presentazione fatta con delle slide oppure un esperimento. Può inoltre essere un servizio quale ad esempio una chat, un forum, una

discussione diretta col docente ecc. Dal punto di vista degli algoritmi proposti, tutte queste attività costituiscono semplicemente elementi con cui è possibile presentare un concetto ad un discente.

Il modello dell'attività didattica è costituito dall'attività stessa e da un metadata ad essa associato. Il **metadata** include:

- la lista di concetti spiegati dall'attività;
- una lista di coppie (proprietà, valore);
- una lista di attributi.

Le proprietà considerate sono esattamente le stesse previste nel modello del discente e nel modello dell'ontologia. Per ogni proprietà il metadata indica il valore assunto dall'attività. Ad esempio, se l'attività è semplicemente un testo, alla proprietà *tipologia dell'attività* sarà associato il valore *lettura di un testo*.

Gli attributi costituiscono proprietà dell'attività non previste nel modello del discente e nel modello dell'ontologia ma di cui si deve tener conto nella fase di costruzione della presentazione. Tali attributi potrebbero riguardare ad esempio la durata dell'attività e il suo costo nel caso di corsi a pagamento,

Tali valori saranno utilizzati per scegliere l'insieme più opportuno di attività da utilizzare per presentare i concetti del learning path al discente.

2.6.1 L'attività di verifica

L'attività di verifica è una particolare attività didattica. Così come le altre attività didattiche, essa è relativa a una lista di concetti C di una data ontologia. In particolare, un test incluso in una unità didattica, riguarda uno o più concetti del learning path.

Dopo che un discente ha eseguito un test, quest'ultimo restituisce una valutazione caratterizzata da un numero appartenente all'intervallo $[0,1]$ per ogni concetto associato. Tale valutazione sarà utilizzata per aggiornare lo stato cognitivo nel modello del discente.

2.6.2 Il modello dell'unità didattica

Per unità didattica si intende l'insieme delle attività didattiche necessarie al raggiungimento di determinati obiettivi cognitivi. Il modello dell'unità didattica è costituito da:

- un insieme di concetti obiettivo (*TC* o *target concepts*) che devono essere appresi dal discente;
- un *learning path* (*LP*) ovvero una sequenza ordinata di concetti atomici che è necessario trasferire a un particolare discente affinché egli possa apprendere i concetti obiettivo e rappresentabile da un percorso orientato definito sul grafo rappresentante l'ontologia;
- un metadata costituito da un insieme di *preferenze di corso* e da un insieme di *attributi*;
- una presentazione (*PR*) ovvero una lista di attività didattiche che vengono fornite al discente affinché questi possa apprendere i concetti del *learning path*.
- Le *preferenze di corso* sono definite dalla stessa lista di coppie (proprietà, valore) utilizzata per descrivere le preferenze di insegnamento (nel modello dell'ontologia) e le preferenze di apprendimento (nel modello del discente). In fase di definizione dell'unità didattica, il docente può scegliere se assegnare o meno un valore a tali coppie. Se lo fa questo valore viene a sostituire, per ogni concetto del *learning path* il valore definito nelle preferenze di insegnamento e sarà utilizzato per la definizione della presentazione.
- Gli *attributi*, a differenza delle preferenze di corso si riferiscono all'intera presentazione e possono definire, ad esempio, la *durata* indicativa della presentazione o il suo *costo*. L'insieme di attributi considerati coincide con quelli previsti nel modello dell'attività didattica e vengono usati in fase di generazione della presentazione. Tenendo conto del fatto che uno stesso concetto può essere riproposto più volte ad uno stesso studente, tali valori sono soltanto indicativi e si riferiscono al caso ideale in cui il

discente non fallisca alcun test e riesca ad apprendere tutti i concetti utilizzando la prima attività didattica che gli viene proposta.

Una volta scelti i concetti obiettivo e definito il metadata, gli algoritmi di cui al capitolo successivo calcoleranno il learning path e successivamente, dato il learning path, costruiranno la presentazione. Nella presentazione non saranno presenti i concetti già conosciuti dal discente.

Una presentazione non viene mostrata interamente al discente. Essa viene suddivisa in segmenti ciascuno dei quali termina con una fase di testing (*milestone*). In ogni istante viene mostrata al discente la parte di presentazione che precede la successiva fase di testing. Alla fine di ogni test, vengono riproposti ai discenti i concetti precedenti per i quali non si ha un livello di conoscenza sufficiente oppure, se tutti i concetti sono stati appresi, si passa al segmento di presentazione successivo.

2.7 La creazione del corso personalizzato

Il processo di creazione di un corso personalizzato ovvero di insegnamento/apprendimento consta di una fase di preparazione, una fase di avvio, una o più fasi di esecuzione, una o più fasi di valutazione ed una fase di chiusura.

- Nella fase di **preparazione** il docente definisce/sceglie l'ontologia, le attività didattiche e le attività di verifica mentre il discente può inizializzare le proprie preferenze di apprendimento.
- Nella fase di **avvio** il docente sceglie un discente (o un gruppo di discenti) e inizializza un'unità didattica stabilendone i concetti obiettivo, ed eventualmente definendo le preferenze e gli attributi di corso. Ciò fatto il sistema genera il learning path.
- Nella fase di **esecuzione** il sistema genera il frammento di presentazione da sottoporre al discente ed il discente ne

intraprende le varie attività componenti fino a raggiungere un blocco di attività di verifica.

- Nella fase di **valutazione** (che si attiva al termine di un blocco di attività di verifica) si aggiorna il modello del discente sulla base dei risultati della attività di verifica per poi passare ad una successiva fase di esecuzione.
- Nella fase di **chiusura** (che si attiva quando tutti i concetti dell'unità didattica sono stati appresi) ci si occupa di raccogliere informazioni statistiche sul processo al fine di proporre migliorie all'ontologia o al materiale didattico.

La figura seguente mostra le varie fasi del processo di apprendimento/insegnamento e vi posiziona le relative operazioni che, laddove svolte da attori umani, sono rappresentati in box tratteggiati mentre, dove eseguiti automaticamente dal sistema attraverso l'applicazione di appositi algoritmi, sono rappresentati in box continui.

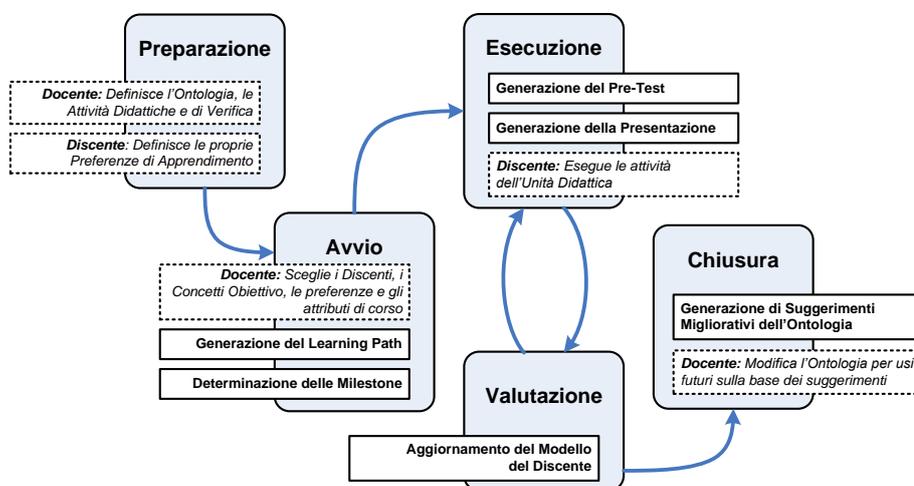


Figura 4. Il processo di insegnamento/apprendimento

Scopo di questo capitolo è quello di fornire una descrizione dettagliata degli algoritmi da attivare nelle varie fasi del suddetto processo di insegnamento/apprendimento.

Nella figura seguente, lo stesso processo viene descritto dal punto di vista algoritmico.

Figura 5. Algoritmo del processo di insegnamento/apprendimento

A partire dall'ontologia e dai concetti obiettivo si calcola il learning path e successivamente si individuano le milestone ovvero si posiziona e si definisce la successiva attività di testing. Si genera quindi la presentazione fino al successivo test e si esegue la verifica dopo la fruizione da parte del discente. Nel caso sia necessario includere un pre-test, questo viene generato e, dopo la fruizione, si passa direttamente alla verifica.

I risultati del test servono per aggiornare il modello del discente, in particolare il suo stato cognitivo e le preferenze di apprendimento. Viene quindi definita la nuova presentazione che sarà costituita da un nuovo segmento di unità didattica se tutti i precedenti concetti sono stati appresi, altrimenti sarà costituita dai concetti già proposti il cui livello di conoscenza non risulta sufficiente.

Il ciclo termina quando, per ogni concetto obiettivo, è stato raggiunto un livello di apprendimento sufficiente. I risultati della fase di valutazione vengono inoltre utilizzati per effettuare una raccolta statistica dei dati relativi ad un determinato dominio. Periodicamente, tali dati saranno analizzati per proporre al docente modifiche all'ontologia con particolare riferimento ai livelli di preferenza associati a ciascuna coppia (proprietà, valore) di ogni concetto.

Nel seguito saranno definiti e descritti gli algoritmi qui menzionati.

2.7.1 Generazione del Learning Path

In questo paragrafo (in accordo con [18]) vengono descritti i passi che portano LIA, il Learner Intelligent Advisor di IWT, alla creazione automatica del Learning Path ovvero del percorso didattico che si articola in quanto indicato come *Presentation*. La *Presentation* è una lista di learning object (LO) che vengono erogati all'utente discente e che rispondono ai suoi requisiti di apprendimento. Lo studente in modo diretto o, come è naturale che accada, per tramite del docente, sottomette a LIA una lista di concetti obiettivo (*TargetC*). Un *target concept* è un concetto del dominio su cui formare un discente che appartiene, appunto, ai suoi obiettivi di apprendimento.

Dato *TargetC* come input, LIA costruisce la *Presentation*, scegliendo una lista di LO tali da soddisfare sia i concetti obiettivo scelti, sia lo stato cognitivo del discente che le sue preferenze di apprendimento.

Più precisamente, una *Presentation PR* è una lista ordinata di LO ($PR = \{l_1, \dots, l_n\}$) con le seguenti proprietà:

1. L'unione dei LO ($\bigcup_{i=1, \dots, n} l_i$) di *PR* è sufficiente a spiegare allo studente tutti i concetti obiettivo dell'insieme *TargetC*.
2. For each $l_i, l_j \in PR$, **if** : *Explained_by*(d_1, l_i) **and** *Explained_by*(d_2, l_j) **and** $d_1 \prec d_2$, **then** $i < j$, dove la relazione d'ordine parziale \prec tra i concetti è ricorsivamente definita come segue:
 - **if** *IsRequiredBy*(x, y) **then** $x \prec y$
 - **if** *SuggestedOrder* (x, y) **then** $x \prec y$
 - **if** *IsTested_by* (x, y) **then** $x \prec y$
 - **if** *HasPart* (z, x) **and** *HasPart* (w, y) **and** $z \prec w$ **then** $x \prec y \wedge x \prec w \wedge z \prec y$.
3. *PR* rispetta le *Learning Preferences* dello studente, ovvero le sue preferenze di apprendimento e non tratta concetti già noti dallo studente ovvero presenti nel suo Stato Cognitivo.

Il punto 1 indica che l'unione dei LO della *Presentation* debba essere sufficiente a spiegare al discente tutti i concetti obiettivo, ovvero appartenenti all'insieme *TargetC*, il punto 3 indica che la *Presentation* deve rispettare le preferenze dell'utente e non deve comprendere concetti già noti. Il punto 2 richiede un po' più di attenzione. La relazione tra i concetti del dominio pone in ordine parziale gli elementi del dominio didattico. Come conseguenza di ciò, i LO appartenenti alla stessa *Presentation* devono rispettare questo ordine parziale. Se, per esempio, una *Presentation* contiene l_i e l_j , che spiegano rispettivamente i concetti *Derivate* e *Limiti*, l_j deve precedere l_i . Lo stesso accade quando l_i e l_j spiegano concetti non direttamente collegati tra loro da una relazione d'ordine (*IsRequiredBy*, *SuggestedOrder* or *IsTested_by*), ma sono componenti di altri concetti collegati da una relazione d'ordine.

Qui di seguito vengono mostrati gli algoritmi necessari alla generazione di una *Presentation* parzialmente ordinata e capace di soddisfare i punti 1 e 3 e la procedura di linearizzazione proposta per ottenere una *Presentation* totalmente ordinata in accordo con il punto 2.

Generazione automatica della *Presentation*

L'algoritmo di generazione della *Presentation* colleziona in una lista di concetti denominata *AtomicList* tutti i concetti atomici che possono essere raggiunti partendo dai concetti boettivo e seguendo le relazioni *HasPart*, *IsRequiredBy* e *IsTested_by*. Quindi la *AtomicList* è quindi linea rizzata. Infine, per ogni concetto della *AtomicList*, l'algoritmo cerca i LO i cui Metadata più si avvicinano alle preferenze di apprendimento dello studente.

Algoritmo di generazione della *Presentation*

Input: *TargetC*: DC list, *CS*: Cognitive State, *LP*: Learning Preferences. **Output:** LO list.

1. Verifica la consistenza dell'ontologia.
2. $Q := TargetC$, $AtomicList := Nil$, $PR := Nil$.
3. For each $x \in Q$ s.t. $\neg Known(x, CS)$, do:
 - a. If [$IsPart_of(y,x)$ or $Requires(x,y)$ or $IsTested_by(x,y)$] and $y \notin Q$, then insert y in Q .
 - b. If $\neg \exists y$ s.t. $IsPart_of(y,x)$, then insert x in $AtomicList$.
4. $AtomicList := Sort(AtomicList, TargetC)$.
5. For each $x \in AtomicList$ do:
 - a. Let $LOList$ be the list of all the LOs l s.t. $Explained_by(x, l)$ and $Consistent(l, LP)$.
 - b. $BLO := Choose_the_best_of(LOList, LP)$.
 - c. Insert BLO in PR .
 - d. For each $x' \in AtomicList$ s.t. $Explained_by(x', BLO)$, delete x' from $AtomicList$.
6. Return PR .

Il passo 1 dell'algoritmo verifica se l'ontologia è consistente. Ovvero verifica che non siano presenti dei cicli che di fatto devono essere evitati dai passi 3 e 4 dell'algoritmo. Un'ontologia è inconsistente quando, per esempio, $x \prec y$ e $y \prec z$ e $z \prec x$. La verifica è facilmente realizzabile cercando eventuali cicli nel grafo orientato che rappresenta l'ontologia. Per ogni concetto obiettivo, il passo 3.a

ricorsivamente colleziona tutti i concetti da apprendere. Al passo 3.b un concetto è aggiunto alla lista dei concetti atomici se non ha componenti (nodi foglia nell'ontologia). Il passo 4 linearizza la lista dei concetti atomici appena ottenuta.

Al passo 5, per ogni concetto x della lista di concetti atomici, viene scelto un LO appartenente all'insieme delle risorse capaci di spiegare x ($LOList(x)$). È importante notare che la scelta è locale a $LOList(x)$.

Infatti, un LO può essere collegato a due o più concetti. Per esempio, LO_3 spiega *Derivatives* e *Integrals*, pertanto, se al passo 5.a funzione *Choose_the_best_of* ($LOList(Derivatives), LP$) individua LO_3 , al passo 5.d *Integrals* sarà escluso da Q . Tuttavia, al passo 5.b si ha $BLO = LO_2$ poiché la funzione *Choose_the_best_of* dipende soltanto da $LOList(Derivatives)$ e da LP . In altre parole, LO_2 potrebbe essere giudicato migliore per lo studente ovvero rispettare di più le sue preferenze di apprendimento rispetto a LO_3 , senza prendere in considerazione alcuna ottimizzazione globale o mitizzazione della lista di LO. La ragione di questa scelta deriva dal fatto che un'ottimizzazione globale potrebbe portare ad un'esplosione combinatoria, mentre generalmente abbiamo soltanto pochi LO collegati a più di un concetto del dominio. Infine, le funzioni *Known*, *Consistent* e *Choose_the_best_of* sono facilmente realizzabili confrontando i concetti e LO con stato cognitivo e preferenze didattiche del discente.

A seconda della funzione, viene preso in considerazione il grado (*Degree*) oppure l'affidabilità (*Reliability*).

Se le liste utilizzate (Q , *AtomicList*, PR , ...) sono realizzate con uno stack (oppure una coda) sia ha che le operazioni di cancellazione o di inserimento di elementi hanno un costo computazionale costante. Inoltre se si associa ad ogni concetto e LO un flag booleano, è possibile marcare gli elementi visitati la prima volta. Così facendo è possibile essere sicuri che nessun concetto o LO sia visitato due volte durante l'esecuzione dell'algoritmo. Di conseguenza, il costo computazionale dei passi 1,2,3 e 5 è $O(n)$, dove n indica il numero di concetti e di LO nel dominio didattico considerato.

La fase di linearizzazione

L'idea presente dietro l'algoritmo di linearizzazione è la seguente. Partendo dalla lista dei concetti obiettivo, si visita tutto il grafo che rappresenta l'ontologia navigando tutte le relazioni che pongono i vincoli di ordine. Mentre nell'algoritmo di generazione della *Presentation* viene marcato ogni nodo quando viene visitato per la prima volta, ora non viene impiegato alcun flag booleano. Un nodo viene visitato dall'algoritmo di linearizzazione tutte le volte che è raggiungibile da una catena di relazioni tra i nodi. La ragione di questa scelta è la seguente. Si immagini di avere le seguenti relazioni tra concetti: $SO(x,y)$, $SO(x,z)$, $SO(y,w)$ e $SO(z,w)$ (dove SO è la relazione *SuggestedOrder*). Supponendo ora di visitare l'ontologia iniziando dal nodo x . Eseguendo una visita in profondità dei nodi, da x si va ad y (mettendo z nello stack), quindi da y si va a w , quindi da z (estratto dallo stack) si prova a raggiungere di nuovo w . Se non si permettesse di visitare nuovamente w , l'ordine risultante sarebbe $\{x,y,w,z\}$, che però viola il vincolo $SO(z,w)$. Viceversa, se si hanno le seguenti relazioni: $SO(x,y)$, $SO(x,z)$, $SO(y,w)$ e $SO(w,z)$ e una visita in ampiezza, si ottiene per esempio $\{x,y,z,w\}$, che viola la relazione $SO(w,z)$. Nell'algoritmo proposto si adotta una visita prima in profondità permettendo che w sia visitato ancora anche dopo l'estrazione di z , ottenendo così: $\{x,y,z,w\}$, che è la corretta linearizzazione che rispetta i vincoli posti.

Algoritmo di ordinamento

Input: *AtomicList*, *TargetC* (DC lists). **Output:** DC list.

1. $max := 1$, $Q := TargetC$.
2. Until Q is not empty do:
 - a. $x := Pop(Q)$, $order(x) := max$, $max := max + 1$.
 - b. For each y s.t. $IsPart_of(y, x)$, $Push(Q, y)$.
 - c. For each y s.t. $Requires(x, y)$ or $SuggestedOrder(y, x)$ or $IsTested_by(y, x)$, $Push(Q, y)$.
3. Ordina tutti gli elementi x di *AtomicList* in base ai valori $order(x)$ e rispettando un ordine decrescente.
4. Return *AtomicList*.

Se Q è realizzato attraverso uno stack, questo assicura che al passo 3.b, le componenti di un concetto x sono visitati prima di ogni altro concetto y possibilmente in relazione con x con una relazione d'ordine (Passo 3.c). Infatti, se sono presenti le relazioni *IsRequiredBy* (y,x) e *HasPart*(x,z), si dovrebbe avere: $z < y$.

Il passo 2 ordina totalmente la porzione di ontologia connessa ai concetti obiettivo (*Ontology(TargetC)*). Da qui, ogni concetto x appartenente all'ontologia *Ontology(TargetC)* al passo 3 è legato ad una etichetta di tipo intero (*order(x)*) che dà la sua posizione nell'insieme totalmente ordinato. Maggiore è *order(x)* rispetto a *order(y)*, minore sarà la posizione di x rispetto ad y nell'ordine. Quindi occorre selezionare nell'ontologia *Ontology(TargetC)*, i concetti scelti precedentemente dall'algoritmo di generazione della *Presentation* (è memorizzata in *AtomicList*). Infatti, al passo 3 è usato un comune algoritmo di ordinamento che ordina gli elementi di *AtomicList* prendendo in considerazione la loro etichetta *order* in ordine decrescente così da ottenere la linearizzazione richiesta.

Se n è il numero di concetti dell'ontologia *Ontology(TargetC)*, la complessità computazionale del passo 3 è $O(n \log(n))$; mentre l'intera procedura presenta una complessità di tipo polinomiale $O(n^4 + m^4)$.

Questo algoritmo presenta evidenti limiti di applicabilità poiché la sua complessità intrinseca fa lievitare i tempi di esecuzione quando si ha a che fare con un numero elevato di concetti e di learning object e quindi, di fatto, i tempi di risposta del sistema.

Capitolo 3

Il lavoro svolto

Il lavoro è stato orientato a studiare modelli e approcci allo stato dell'arte per individuare soluzioni metodologiche che possano apportare miglioramenti ad IWT.

Il lavoro svolto ha concentrato l'attenzione in una prima fase a quali aspetti potessero essere migliorati nell'algoritmo di LIA ovvero, ferma restando l'efficacia didattica di un approccio che presti attenzione al singolo utente finale, sono stati trattati aspetti di efficienza computazionale degli algoritmi di generazione del percorso personalizzati che, di fatto, sono stati realizzati seguendo tecniche risolutive differenti.

IWT con i nuovi algoritmi è stato poi sperimentato in vari contesti facendo confronti tra l'approccio tradizionale dell'e-learning statico e l'approccio IWT dell'e-learning personalizzato.

Da considerazioni legate principalmente alla fase sperimentale, sono stati individuati altri ambiti nei quali le metodologie adottate nella piattaforma IWT possono essere migliorate. In particolare, i casi più complessi derivano da una presenza di un elevato numero di utenti e di un elevato numero di contenuti. Sia utenti che contenuti vengono modellati con diversi parametri, ma non esistono metodologie che a priori, ovvero prima di intraprendere qualsiasi percorso di formazione a distanza, siano in grado di verificare se ci siano o meno i contenuti sufficienti a soddisfare le esigenze degli utenti. Per questo è stato studiato un approccio risolutivo che fa impiego di tecniche basate sull'aritmetica degli intervalli e che risultano essere decisamente efficaci.

Altro aspetto chiave su cui è stato necessario effettuare studi e definire delle metodologie appropriate è stato l'aggiornamento del profilo utente. IWT ha un modello utente basato su stato cognitivo e

preferenze che, perché abbiano senso, devono venire costantemente aggiornati durante i processi di formazione a distanza. Le metodologie di aggiornamento sono state studiate e sono stati disegnati algoritmi che, anche grazie ai suggerimenti di esperti della didattica, possano fornire maggiori garanzie sulle valutazioni delle performance degli utenti.

Ultimo aspetto trattato è legato alla modellazione del dominio richiesta per poter applicare l'approccio della personalizzazione di IWT ovvero, in particolare, allo sforzo necessario per disegnare ontologie e descrivere i contenuti didattici da utilizzare attraverso i metadata. Per questa complessa problematica è stata proposta una soluzione di estrazione automatica di ontologie direttamente dai contenuti e di definizione automatica di metadata.

3.1 Come è stato modificato l'algoritmo di LIA di IWT

La fase di generazione di learning path LP prevede i seguenti dati di ingresso:

- una ontologia $O(C, (R_1, \dots, R_n))$ in cui C è l'insieme dei concetti ed R_1, \dots, R_n sono le relazioni esistenti tra i concetti;
- un insieme di concetti obiettivo TC ;
- una relazione di scomposizione D (ad esempio la relazione *HasPart*);
- una lista ordinata A di relazioni d'ordine utilizzate per definire gli elementi del learning path (ad esempio la relazione *IsRequiredBy*);
- una lista ordinata di relazioni d'ordine utilizzate per definire l'ordine degli elementi presenti nel learning path (ad esempio le relazioni *IsRequiredBy* e *SuggestedOrder*);
- lo stato cognitivo di un discente CS .

A partire da tali dati di input, l'algoritmo può essere descritto utilizzando il seguente diagramma di flusso in cui $A' = A \cup D$ e $B' = B \cup D$.

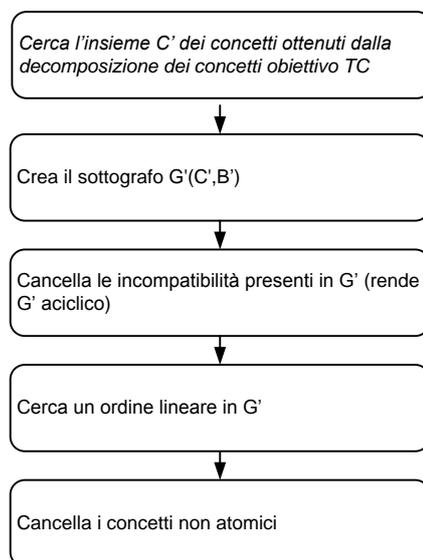


Figura 6. Generazione del learning path

Nel seguito del paragrafo verranno descritti i singoli passi dell'algoritmo.

Step 1: Cerca l'Insieme C' dei Concetti Ottenuti dalla Scomposizione di TC

Per individuare l'insieme dei concetti C' in cui sono scomposti i concetti obiettivo TC , è sufficiente verificare sul grafo $G(C,A')$ quali sono i nodi a partire dai quali si può raggiungere un nodo di TC con un percorso orientato. Tali nodi, insieme ai concetti obiettivo stessi TC , sono gli elementi dell'insieme C' .

Se $A' = (HP, IR)$, a partire dall'esempio di ontologia di figura 1, possiamo considerare il grafo seguente. Se c_c è il concetto obiettivo abbiamo: $C' = \{c_d, c_1, c_2, c_3, c_4, c_d, c_5, c_6, c_7, c_8, c_e, c_c\}$.

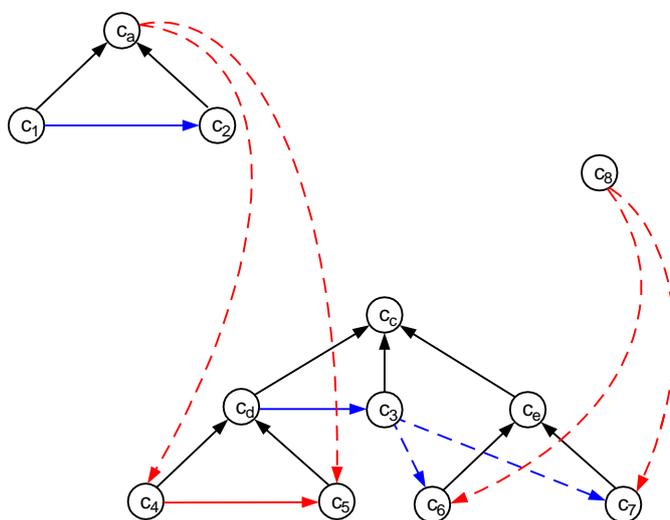


Figura 8. Esempio di creazione del sottografo G'

Step 3: Rende G' Aciclico

G' è un grafo orientato. A questo punto è necessario trovare un ordinamento lineare di questo grafo ma non sappiamo se esso è aciclico o meno. Se esistono cicli, questi devono essere eliminati seguendo l'ordine fornito dall'insieme B' (che è costituito da relazioni d'ordine).

L'algoritmo per l'identificazione dei cicli opera nel seguente modo. Esso elimina le radici del grafo finché è possibile. Se ad un certo punto il grafo risultante non contiene radici ma non è vuoto allora vuol dire che esiste un ciclo. Per eliminare tale ciclo, possiamo eliminare gli archi uscenti dal nodo con il minimo numero di archi uscenti. La procedura viene iterata finché non si arriva ad un sottografo vuoto.

Usando la notazione $\delta^+(v)$ per indicare l'insieme degli archi uscenti dal nodo v e $\delta^-(v)$ per indicare l'insieme degli archi entranti in v , lo schema formale dell'algoritmo è il seguente:

- **Step 3.0.** $i = 0$, $\bar{B} = B'$, $\bar{C} = C'$, $\hat{B} = \emptyset$.

- **Step 3.1.** Cerca un nodo $v \in \bar{C}$ tale che $|\delta^+(v)| = \min_{u \in \bar{C}} |\delta^+(u)|$. Se il nodo v è una radice, non è stato ancora individuato alcun ciclo, altrimenti (esistono archi uscenti) il ciclo è stato trovato.
- **Step 3.2.** Cerca il nodo h tale che $\delta^+(h)$ contiene archi a minima priorità.
- **Step 3.3.** Pone $\bar{C} = \bar{C} \setminus \{h\}$, $\bar{B} = \bar{B} \setminus (\delta^+(h) \cup \delta^-(h))$, $\hat{B} = \hat{B} \cup \delta^+(h)$. Se v non è una radice, allora gli archi uscenti creano qualche ciclo e li memorizza in \hat{B} . Quindi elimina questo nodo e ripete l'operazione.
- **Step 3.4.** Se $\bar{C} = 0$ allora va allo **Step 3.5** altrimenti va allo **Step 3.1**.
- **Step 3.5.** $G(C', B' \setminus \hat{B})$ è un grafo aciclico.

Step 4: Cerca l'Ordine Lineare nel Grafo Aciclico

Usando l'ordinamento topologico, possiamo ordinare i nodi di C' . Questo problema può essere facilmente risolto utilizzando il metodo di ricerca depth-first. Tale metodo visita i nodi del grafo secondo un percorso che predilige i nodi più profondi cioè dei livelli più bassi.

Un possibile ordine lineare per l'esempio precedente è rappresentato nella figura seguente.

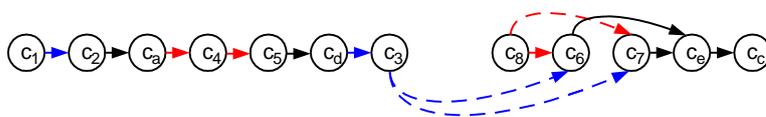


Figura 9. Esempio di ordinamento lineare

All'interno dell'ordinamento topologico costruiamo una chiusura transitiva del grafo G' . A tale scopo aggiungiamo un arco tra ogni coppia di nodi a e b se esiste un percorso orientato da a a b .

3.1.1 Determinazione delle Milestone

Le milestone rappresentano i punti del learning path in cui inserire le attività di testing. L'output di un algoritmo per la determinazione delle milestone è quindi costituito da un nuovo learning path LP' costituito non soltanto dai concetti dell'ontologia ma anche da concetti fittizi che dovranno essere proposti al discente tramite attività di testing.

La determinazione delle milestone può essere fatta sostanzialmente in due modi:

1. possono essere settate direttamente dal docente in una posizione precisa del learning path;
2. possono essere definite in percentuale sulla lunghezza del learning path dal docente; ad esempio una lista di percentuali del tipo [0.2, 0.5, 0.7, 1] significa che il docente ha deciso di inserire nel corso 4 milestone posizionate rispettivamente al 20%, 50%, 70% e alla fine del corso.

Quando viene scelto il secondo metodo può capitare che una milestone si presenti tra concetti ottenuti dalla scomposizione di un unico concetto padre. In altri termini potrebbe capitare che si inizi la presentazione di un tale concetto e che si presenti la fase di testing prima che il concetto sia terminato. Per evitare questo inconveniente ed imporre che le milestone non dividano alcun concetto soggetto a scomposizione si può operare nel modo seguente:

1. a partire dal learning path originario, si individuano gli insiemi di concetti ottenuti dalla scomposizione dello stesso concetto padre;
2. si sostituiscono tali concetti con il concetto padre posizionato in corrispondenza dell'ultimo concetto figlio;
3. sul learning path così ottenuto si calcolano le milestone;
4. si ritorna al learning path originario con l'aggiunta delle milestone.

In questo modo si avrà un posizionamento delle milestone compatibile con la relazione di scomposizione. Per evitare che, in seguito all'aggregazione (step 2) vengano alterate in maniera

eccessiva le percentuali indicate dal docente per il posizionamento delle milestone, ai concetti aggregati verrà attribuito un peso pari al numero di concetti costituenti.

3.1.2 Generazione del Pre-Test

Come indicato nella introduzione, alcuni corsi possono prevedere dei pre-test di diverse tipologie: pre-test sui requisiti, pre-test sui contenuti, pre-test integrati e pre-test per la definizione delle preferenze di apprendimento. I primi si riferiscono ai concetti del learning path considerati già noti al discente, i secondi ai concetti del learning path non ancora appresi dal discente, i terzi sono costituiti da entrambe le tipologie precedenti. Infine, i pre-test per la definizione delle preferenze di apprendimento possono essere utilizzati prima dell'inizio del corso per definire le preferenze di apprendimento del discente. Questi pre-test possono essere definiti indipendentemente dallo specifico argomento del corso per valutare le caratteristiche del discente e tradurle in un specifiche preferenze di apprendimento. Quest'ultima tipologia di pre-test, dunque, non deve essere generata automaticamente ma deve soltanto essere posta all'inizio del corso, prima della generazione della presentazione. Le risposte fornite dal discente saranno poi utilizzate per la definizione delle preferenze di apprendimento così come indicato dal pre-test stesso.

Nel caso delle altre tipologie, scopo dell'algoritmo per la generazione dei pre-test è di individuare i concetti da sottoporre a test e, sulla base di questi, definire il pre-test ovvero sequenzializzare un insieme di attività di verifica congruenti con i concetti da testare.

- Nel caso di **pre-test sui requisiti**, l'algoritmo percorrere il learning path e inserisce nella lista dei concetti da sottoporre a test tutti quelli già appresi dallo studente.
- Nel caso di **pre-test sui contenuti**, l'algoritmo inserisce nella lista dei concetti da valutare tutti quelli non ancora appresi dallo studente.
- Infine nel caso dei **pre-test integrati**, l'algoritmo inserisce nella lista dei concetti da valutare tutti quelli appartenenti al learning path.

A seguito della fase di pre-testing, verrà modificata la valutazione attribuita al discente in relazione a tutti i concetti valutati. Questo significa che alcuni concetti già appresi (in seguito a pre-test sui requisiti) potranno essere riproposti allo studente, mentre altri non ancora proposti (in seguito a pre-test sui contenuti) potranno essere considerati già conosciuti dal discente e quindi non gli saranno proposti in fase di presentazione.

3.1.3 Generazione della Presentazione

La prima operazione da effettuare per generare una presentazione di una unità didattica consiste nella scelta dei concetti da presentare al discente.

Per effettuare tale scelta, si percorre il learning path fino al raggiungimento della prima milestone preceduta da almeno un concetto non ancora appreso dal discente. Tali concetti sono quelli che dovranno essere opportunamente presentati.

Presentare un insieme di concetti significa scegliere l'insieme delle attività didattiche più opportuno sulla base delle preferenze di apprendimento del discente, delle preferenze di insegnamento legate ai concetti proposti, delle preferenze e degli attributi di corso.

Come è stato descritto nel capitolo precedente, le preferenze di apprendimento del discente e le preferenze di insegnamento del singolo concetto (o le preferenze di corso nel caso queste siano state definite) sono rappresentate entrambe mediante una stessa lista (proprietà, valore). Tale lista presenta, in corrispondenza di ciascuna coppia, un grado che, nel modello dello discente, rappresenta il grado di preferenza da parte del discente, nell'ontologia (o nel modello dell'unità didattica) invece indica quanto quella caratteristica sia ritenuta appropriata dal docente nel presentare il concetto corrispondente.

Dal momento che, come descritto nel paragrafo 2.5, le preferenze di corso, se definite, vengono semplicemente a sostituire le preferenze di insegnamento, nel seguito faremo riferimento per semplicità soltanto alle preferenze di insegnamento senza indicare se esse sono state modificate o meno dalle preferenze di corso.

Un'attività didattica è a sua volta caratterizzata da una lista di coppie (proprietà, valore) che esprime il valore assunto da ogni singola proprietà.

Per ogni coppia (*attività didattica, concetto descritto*), si può definire una funzione distanza (*DI*) rispetto alle preferenze di insegnamento ottimali, definita nel modo seguente:

- per ogni proprietà si considera il valore assunto dall'attività didattica;
- si considera il grado assegnato dal docente alla coppia corrispondente (proprietà, valore);
- si incrementa il valore della distanza di una quantità pari alla differenza tra il grado ottimale (10) e il grado assegnato dal docente.

Infine se si vuole tenere conto della diversa importanza di ciascuna proprietà, si può pesare la somma precedente moltiplicando ciascun termine per un valore di priorità compreso nell'intervallo $[0,1]$. Più tale valore si avvicina ad 1, più la corrispondente proprietà è ritenuta prioritaria rispetto alle altre.

La stessa procedura può inoltre essere utilizzata per definire, per ogni coppia (*attività didattica, concetto descritto*), una funzione distanza (*DA*) rispetto alle preferenze di apprendimento.

Nella fase di generazione della presentazione, oltre alla distanza tra attività didattica e concetto descritto, occorre tener conto, per ogni attività didattica, del valor assunto dagli attributi di corso introdotti nel paragrafo 2.5. Tali attributi sono caratterizzati da grandezze sommabili. In fase di definizione dell'unità didattica, il docente definisce le preferenze di corso (oppure decide di utilizzare le preferenze di insegnamento presenti nell'ontologia) e definisce i valori di alcuni attributi quali ad esempio la durata dell'unità didattica e il suo costo. Infine decide dove posizionare le milestone (al esempio al 50% e alla fine del corso). Per ciascuna milestone si può calcolare la percentuale del valore degli attributi che riguardano la parte di unità didattica che si sta presentando (ad esempio la metà della durata totale e la metà del costo totale). Nella definizione della presentazione, si scelgono dunque le unità didattiche in maniera tale che, per ciascun attributo, la somma dei valori assunti dalle attività didattiche utilizzate

si avvicini il più possibile al valore previsto dal docente per quella porzione di unità didattica.

Nel seguito il problema di scelta ottimale di un insieme di attività didattiche per la copertura di un assegnato insieme di concetti viene modellato come un problema di Programmazione Lineare Intera. Successivamente sono descritti un algoritmo greedy e un algoritmo di ricerca locale per la sua soluzione.

Formulazione del problema

Il problema di scegliere un insieme ottimale di attività didattiche per la copertura di un assegnato insieme di concetti può essere opportunamente formalizzato come un problema di localizzazione di impianti su un grafo bipartito come indicato nella figura seguente.

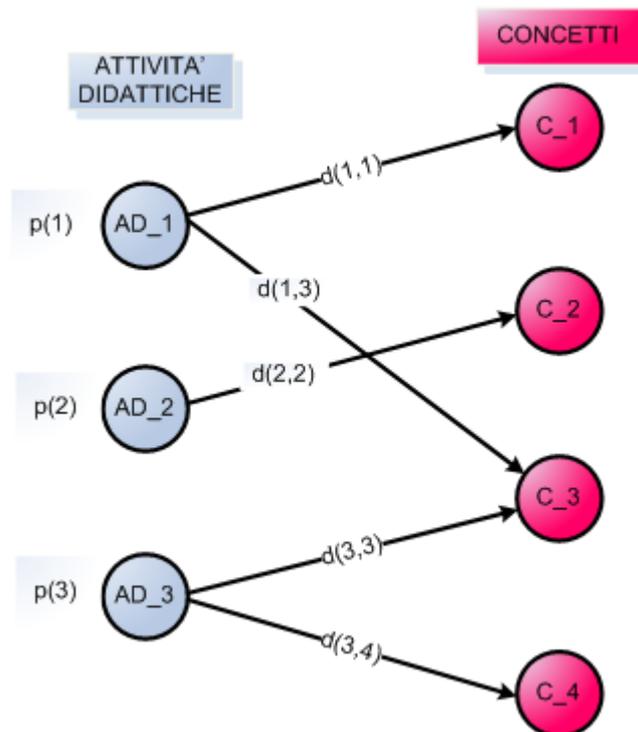


Figura 10. Schematizzazione del problema di copertura

In tale figura si è supposto di avere a disposizione tre attività didattiche per coprire un insieme di 4 concetti. Ad ogni coppia *attività*

didattica – *concetto* è associato un valore di distanza $d(i,j)$ che rappresenta la qualità di copertura del concetto j -esimo da parte dell'attività i -esima. Inoltre ad ogni attività didattica è associato un peso $p(i)$ che rappresenta il costo dell'attività didattica i .

I concetti possono essere considerati come *clienti* che richiedono una *spiegazione* da parte di uno dei potenziali impianti (*attività didattiche*). Se viene utilizzata l'attività didattica i -esima per fornire la spiegazione del concetto j -esimo, si genera un costo $d(i,j)$ che rappresenta il costo di trasporto dell'oggetto *spiegazione* da i a j . Inoltre l'apertura dell'impianto i , ovvero l'utilizzo dell' i -esima attività didattica genera un costo fisso di utilizzo pari a $p(i)$. Tale peso può dipendere dal tempo richiesto per il suo apprendimento, oppure può essere lo stesso per ogni attività didattica. In questo secondo caso si ipotizza che la qualità di un insieme di attività didattiche, dipende oltre che dalla qualità di copertura anche dal numero di attività didattiche utilizzate.

Il costo di un insieme di attività didattiche è dato dalla somma dei costi di trasporto ovvero delle distanze $d(i,j)$ tra attività didattica e concetto che esso deve spiegare e dei costi fissi ovvero dei pesi $p(i)$ delle attività didattiche utilizzate. L'obiettivo è quello di scegliere l'insieme delle attività didattiche che copra tutti i concetti e che abbia costo minimo.

Tale problema è un classico problema di localizzazione (*unsplittable plant location*) che può essere facilmente formalizzato come problema in Programmazione Lineare Intera. A tale scopo, sia m il numero di attività didattiche disponibili e siano n i concetti del learning path da coprire. Sia inoltre y_i $i=1,\dots,m$ una variabile binaria che assume valore 1 se si decide di utilizzare l'attività didattica i , 0 altrimenti, e sia x_{ij} $i=1,\dots,m$ $j=1,\dots,n$ una variabile binaria che assume valore 1 se il concetto j è coperto dall'attività didattica i , 0 altrimenti. Il modello in programmazione lineare intera che formalizza il problema descritto è il seguente:

$$\min \sum_{i=1}^m p(i) y_i + \sum_{i=1}^m \sum_{j=1}^n d(i, j) x_{ij}$$

s.t.

$$\sum_{i=1}^n x_{ij} \geq 1 \quad i = 1, \dots, n \quad (1)$$

$$x_{ij} \leq y_i \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (2)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (3)$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, m \quad (4)$$

I vincoli (1) impongono che ciascun concetto debba essere coperto da almeno una attività didattica, i vincoli (2) impongono che un concetto possa essere spiegato soltanto da una attività didattica presente nell'insieme delle attività utilizzati (*impianti aperti*).

Per tenere in conto anche degli attributi di corso, al modello precedente può essere aggiunto un insieme di vincoli che per ogni attributo imponga un valore complessivo massimo: $\sum_{i=1}^m a_i^l y_i \leq ub^l$ per ogni attributo l dove a_i^l è il valore dell'attributo l associato all'attività didattica i e ub^l è il valore complessivo massimo che l'attributo l può assumere su tutto l'insieme di attività didattiche che coprono i concetti assegnati.

Nel caso in cui vengano forniti anche dei valori minimi lb^l per ogni attributo, si possono aggiungere vincoli analoghi che impongano un valore complessivo minimo di ogni attributo.

Una volta definiti opportunamente i valori dei pesi associati alle attività didattiche $p(i)$ e delle distanze $d(i, j)$, la soluzione ottima del modello (1)-(4) fornisce l'insieme ottimale delle attività didattiche da utilizzare e, per ogni concetto, l'attività didattica corrispondente.

Una soluzione ammissibile per il problema (1)-(4) può contenere una attività didattica capace di coprire più concetti ma utilizzata per spiegarne solo un sottoinsieme. Ad esempio può capitare, nel caso rappresentato dalla figura 1, che vengano utilizzati contemporaneamente sia l'attività didattica 1 che la 2. Questo significa che la prima, anche se potenzialmente potrebbe coprire sia il concetto 1 che il concetto 2, viene utilizzata solo per spiegare il

concetto 1 mentre per spiegare il concetto 2 si utilizza la seconda attività didattica.

Una tale eventualità potrebbe non essere ammissibile perché porta ad una lista di attività didattiche che spiegano più di una volta lo stesso concetto. Per eliminare soluzioni di questo tipo è sufficiente sostituire i vincoli (1) con vincoli di uguaglianza che impongono che ogni concetto debba essere coperto da esattamente una attività didattica.

In conclusione, tenendo conto delle differenti funzioni di distanze e degli attributi di corso, il modello in Programmazione Lineare Intera da risolvere diventa:

$$\begin{aligned} \min \quad & \sum_{i=1}^m p(i)y_i + \sum_{i=1}^m \sum_{j=1}^n (\alpha_1 \cdot DI(i,j) + \alpha_2 \cdot DA(i,j)) x_{ij} \\ \text{s.t.} \quad & \\ & \sum_{i=1}^m x_{ij} = 1 \quad j = 1, \dots, n \\ & x_{ij} \leq y_i \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (5) \\ & \sum_{i=1}^m a_i^l y_i \leq ub^l \quad l = 1, \dots, na \\ & \sum_{i=1}^m a_i^l y_i \geq lb^l \quad l = 1, \dots, na \\ & x_{ij} \in \{0,1\} \quad i = 1, \dots, m \quad j = 1, \dots, n \\ & y_i \in \{0,1\} \quad i = 1, \dots, m \end{aligned}$$

dove α_1 e α_2 sono due parametri di priorità tra le preferenze di insegnamento e le preferenze di corso il cui valore può essere definito nel corso della sperimentazione (inizialmente si può porre $\alpha_1 = 0.7$ e $\alpha_2 = 0.3$ per tentare di dare maggiore importanza alle preferenze di insegnamento).

Infine occorre osservare che i valori massimi e minimi di ciascun attributo, inizialmente possono essere posti uguali a quelli definiti dal

docente per poi aggiornarli nel caso in cui, a seguito di una fase di test, la presentazione debba essere rigenerata con attività didattiche diverse. In questi casi può diventare molto complicato rispettare i valori previsti dal docente per cui è possibile che il modello precedente non presenti alcuna soluzione ammissibile. Può risultare dunque più utile spostare questi due insiemi di vincoli in funzione obiettivo richiedendo la minimizzazione della violazione rispetto ai tempi previsti dal docente. Il modello (5) diventa dunque:

$$\begin{aligned} \min \quad & \sum_{i=1}^m p(i)y_i + \sum_{i=1}^m \sum_{j=1}^n (\alpha_1 \cdot DI(i,j) + \alpha_2 \cdot DP(i,j)) x_{ij} + \sum_{l=1}^{na} \lambda_l \left(ub^l - \sum_{i=1}^m a_i^l y_i \right) + \sum_{l=1}^{na} \lambda_l \left(\sum_{i=1}^m a_i^l y_i - lb^l \right) \\ \text{s.t.} \quad & \\ & \sum_{i=1}^m x_{ij} = 1 \quad j = 1, \dots, n \\ & x_{ij} \leq y_i \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (6) \\ & x_{ij} \in \{0,1\} \quad i = 1, \dots, m \quad j = 1, \dots, n \\ & y_i \in \{0,1\} \quad i = 1, \dots, m \end{aligned}$$

In cui la funzione obiettivo può essere ricondotta, a meno di costanti additive ad una espressione del tipo

$$\sum_{i=1}^m f(i)y_i + \sum_{i=1}^m \sum_{j=1}^n d(i,j)x_{ij} .$$

Per risolvere in maniera euristica sia il problema rappresentato dal modello (6), è possibile utilizzare una euristica *greedy* per ricavare una prima soluzione ammissibile e, successivamente, utilizzare un algoritmo di ricerca locale per cercare di migliorare tale soluzione iniziale.

Algoritmo *greedy*

I problemi di localizzazione sono caratterizzati dal fatto che ogni soluzione ammissibile è completamente caratterizzata dall'insieme dei centri di servizio (impianti) attivati. In altre parole, una volta definito l'insieme L delle attività didattiche utilizzate (sottoinsieme dell'insieme di tutte le attività didattiche LT), è banale definire una funzione $D(L)$ che ad ogni concetto j associa l'attività didattica i appartenente all'insieme L che lo copre meglio ovvero tale che

$d(i, j) = \min_{l \in L} (d(l, j))$. Di conseguenza la funzione obiettivo del problema è esprimibile come funzione dell'insieme dei learning object utilizzati, nella forma:

$$z(L) = \sum_{i \in L} f(i) + \sum_{j \in C} \min_{i \in L} d(i, j)$$

dove C è l'insieme di concetti da coprire e si pone $d(i, j) = \infty$ nel caso in cui l'attività i non copra il concetto j .

L'algoritmo greedy costruisce l'insieme delle attività didattiche utilizzate in passi successivi a partire dall'insieme vuoto. Ad ogni passo, l'algoritmo sceglie tra le attività didattiche non utilizzate quella che, con la sua aggiunta, causa la *massima diminuzione* della funzione obiettivo $z(L)$.

Se ad un certo passo, l'aggiunta di una qualsiasi attività didattica non ancora utilizzata non causa una diminuzione della funzione $z(L)$ allora l'algoritmo si ferma e l'insieme corrente delle attività didattiche utilizzate definisce la soluzione ammissibile prodotta dall'algoritmo (*soluzione greedy*). Più formalmente i passi dell'algoritmo sono i seguenti:

Inizializzazione: $i = 0$, $L_0 = \emptyset$, $z(L_0) = \infty$

Iterazione i-esima

$$l_i = \arg \min_{l \in LT - L_{i-1}} z(L_{i-1} \cup \{l\})$$

$$\text{Se } z(L_{i-1} \cup \{l_i\}) \geq z(L_{i-1})$$

Allora STOP: L_{i-1} è la soluzione greedy.

$$\text{Altrimenti poni } L_i = L_{i-1} \cup \{l_i\}$$

$$\text{Se } L_i = LT$$

Allora STOP: L_i è la soluzione greedy

Altrimenti poni } i := i + 1 e vai all'iterazione i

Fine dell'iterazione i-esima

Algoritmo di ricerca locale

L'algoritmo greedy consente di ottenere molto velocemente una soluzione ammissibile. La qualità di tale soluzione è condizionata, però, dal fatto che l'algoritmo greedy non consente, nella generica iterazione, di rivedere le decisioni prese alle iterazioni precedenti. Si può però cercare di migliorare la qualità di tale soluzione utilizzando un metodo di ricerca locale.

Il primo passo da fare nella definizione di un algoritmo di ricerca locale consiste nel definire, data una soluzione ammissibile, un intorno di tale soluzione ovvero un insieme di soluzioni ammissibili vicine alla soluzione corrente. Se L è la soluzione corrente, chiamiamo $N(L)$ il suo intorno.

Nel caso del problema (6) un intorno di una soluzione corrente L può essere definito facilmente nel seguente modo:

$$N(L) = L \cup \{L \cup \{u\} : u \in LT - L\} \cup \{L - \{u\} : u \in L\} \cup \{L - \{w\} \cup \{u\} : w \in L, u \in LT - l\}$$

In altri termini un intorno della soluzione L è costituito:

- 1) dalla stessa soluzione L ,
- 2) da tutte le soluzioni che si ottengono da L eliminando un'attività didattica,
- 3) da tutte le soluzioni che si ottengono da L aggiungendo un'attività,
- 4) da tutte le soluzioni che si ottengono da L sostituendo un'attività appartenente a L con una non ancora utilizzata.

Una volta definito un intorno di una generica soluzione ammissibile, un algoritmo di ricerca locale si basa sui seguenti passi:

Inizializzazione: $i = 1$, $L_0 =$ soluzione ammissibile iniziale (soluzione greedy)

Iterazione i -esima

$$L_i = \arg \min \{z(L) : L \in N(L_{i-1})\}$$

$$\text{Se } z(L_{i-1}) \leq z(L_i)$$

Allora STOP: L_{i-1} è un ottimo locale.

Altrimenti poni $i = i + 1$ e vai all'iterazione i

Fine dell'iterazione i -esima

Nel caso del problema (6) la definizione di un intorno di una soluzione corrente pone qualche piccolo problema in più. In questo caso infatti se si elimina un'attività da una soluzione ammissibile non si ottiene mai una nuova soluzione ammissibile. Allo stesso modo, se si aggiunge un'attività didattica ad una soluzione ammissibile si ottiene sempre una soluzione inammissibile. Occorre dunque individuare un nuovo metodo per costruirsi un intorno di una soluzione qualsiasi.

A tale scopo indichiamo con $C(L,l)$ l'insieme di concetti che non sono coperti se si elimina l'attività l dalla soluzione L . Indichiamo inoltre con $gr(C(L,l))$ l'insieme di attività che copre soltanto i concetti in $C(L,l)$. Tale insieme può essere ottenuto applicando l'euristica greedy descritta in precedenza al sottoinsieme di concetti scoperti. Inoltre sia $E(i)$ l'insieme di concetti coperti dalla generica attività didattica $i \notin L$ e sia $EL(i) \subseteq L$ il sottoinsieme delle attività che coprono solo i concetti in $E(i)$. Un intorno di una generica soluzione ammissibile L si può definire come:

$$N(L) = L \cup \{L - \{l\} \cup gr(C(L,l)) : l \in L\} \cup \{L \cup \{i\} - EL(i) : i \in LT - L\}$$

Sulla base di tale definizione di intorno è possibile definire un algoritmo di ricerca locale in grado di ricavare un ottimo locale del problema (6).

Generazione della presentazione a seguito di fallimenti

La formulazione del problema di scelta di un insieme ottimale di attività didattiche per la copertura di un assegnato insieme di concetti e gli algoritmi proposti per la sua soluzione restano validi in ogni fase del processo di fruizione del corso da parte di un discente.

Come indicato in precedenza, a seguito di una fase di verifica, viene scelto un sottoinsieme dei concetti appartenenti al learning path da presentare al discente. Questo sottoinsieme deve contenere, tra gli altri, tutti i concetti per i quali la fase di test ha dato esito negativo. Nel riproporre tali concetti al discente, sarebbe opportuno che le

attività didattiche scelte per la loro descrizione fossero diverse da quelle già utilizzate in precedenza. D'altro canto potrebbe capitare che un concetto sia descritto soltanto da una attività didattica e in quel caso è giocoforza riproporre il concetto utilizzando la stessa attività didattica.

Per tenere in conto di questa esigenza si può semplicemente assegnare ad ogni coppia *attività didattica - concetto* già proposta con esito negativo, una distanza $d(i,j)$ molto grande rispetto ai valori di distanza associati a tutte le altre coppie (per esempio si può porre

$d(i, j) = \sum_{\substack{u=1 \\ u \neq i}}^m d(u, j)$). In questo modo, l'algoritmo proposto non

riutilizzerà mai la stessa attività didattica per spiegare lo stesso concetto a meno che non sia obbligato dalla mancanza di altre attività didattiche utilizzabili per quel concetto.

3.1.4 La Complessità degli Algoritmi

Gli algoritmi proposti presentano un tempo di calcolo trascurabile anche nel caso di istanze di grandi dimensioni. L'unico algoritmo la cui implementazione può essere critica in termini di tempo di calcolo è l'algoritmo di generazione del *learning path*. Tale algoritmo è definito su di un grafo orientato G con n nodi ed m archi.

La complessità di ciascuna fase dell'algoritmo è riportata in tabella 1, in cui la colonna *Algoritmo* elenca le diverse fasi dell'algoritmo, *Descrizione* riporta una descrizione dell'implementazione di ogni fase, le colonne n ed m indicano rispettivamente il numero di nodi e di archi del grafo su cui, in quella fase, l'algoritmo lavora. La colonna *Comp.* indica la complessità computazionale di ogni fase.

Occorre osservare che la complessità computazionale è calcolata considerando un grafo come una lista di archi orientati. Questa struttura dati è preferibile perché i grafi su cui si trova ad operare l'algoritmo sono molto sparsi, cioè il numero di archi è molto più piccolo di n^2 .

Nella tabella, infine, non si tiene conto dell'ordinamento degli archi. In questo algoritmo viene impiegata la procedura *quick sort* che presenta una complessità nel caso peggiore pari a $O(m \log(m))$.

Algoritmo	Descrizione	n	M	Comp.
Step 1	Il grafo viene esploso usando l'algoritmo di ricerca depth first.	$ C $	$ A $	$O(m)$
Step 2	L'ispezione di ogni arco viene fatto utilizzando l'algoritmo di ricerca depth first.	$ C $	$ B $	$O(m^2)$
Step 3	Per ogni nodo devono essere considerati tutti gli archi.	$ C' $	$ B $	$O(mn)$
Step A5	Il grafo viene ispezionato utilizzando l'algoritmo di ricerca depth first.	$ C' $	$ B $	$O(m)$

Tabella 1. Complessità computazionale

3.2 La sperimentazione condotta per verificare la validità dell'approccio IWT nell'e-learning

3.2.1 Sperimentazione di IWT con gli studenti universitari

IWT stata sperimentata nel corso di tre anni accademici su una platea di utenti alquanto variegata sui seguenti corsi universitari:

- Fondamenti di informatica
- Algoritmi e strutture dati (Informatica II)
- Matematica I
- Fondamenti sui database
- Reti di calcolatori
- Programmazione in linguaggio C.

Sono stati coinvolti studenti afferenti a due atenei diversi, Università degli Studi di Salerno e Università degli Studi Roma 3, per un totale di 1385 studenti.

Complessivamente i corsi sono stati costruiti trattando un numero pari a quasi 600 concetti.

Gli utenti per ciascun corso sono stati suddivisi in due gruppi e a loro insaputa sono stati erogati percorsi formativi apparentemente personalizzati per tutti. In realtà una metà ha fruito percorsi di e-learning convenzionali statici e l'altra metà percorsi personalizzati. I percorsi iniziavano tutti con dei test di ingresso per appurare il livello di conoscenze iniziale sugli argomenti del corso e quindi, a valle del test, sono stati erogati percorsi formativi differenti per ciascun utente. Alcuni hanno ricevuto, nonostante le risposte ai test, dei percorsi formativi completi, altri hanno ricevuto dei percorsi formativi costruiti su misura e quindi confezionati alla luce dei risultati dei test prendendo in considerazione i concetti non noti.

A valle della formazione, tutti gli studenti hanno sostenuto una prova finale di cui sono stati raccolti i risultati e quindi le performance dei singoli sui vari corsi.

I risultati presentati nelle tabelle e nelle figure riportate qui di seguito riassumono quanto detto, ovvero riportano il risultato medio per singolo corso degli studenti che hanno studiato corsi statici (barre più chiare) e corsi personalizzate (barre più scure). Il risultato evidente è che la personalizzazione ha portato gli studenti a raggiungere, mediamente, risultati migliori.

Course	Without personalized courses			With personalized courses		
	Students	Passed	%	Students	Passed	%
Foundation of Informatics	186	81	43,55%	152	97	63,82%
Informatics II	69	12	17,39%	81	32	39,51%
Principles of Databases	125	57	45,60%	110	60	54,55%
C Programming	68	17	25,00%	81	30	37,04%
Computer Nets I	95	62	65,26%	98	65	66,33%
Mathematics I	171	46	26,90%	149	55	36,91%
TOT	714	275	38,52%	671	339	50,52%

Tabella 2. Riepilogo risultati al test finale degli studenti a valle dei corsi senza personalizzazione o con la personalizzazione di IWT sulle varie materie

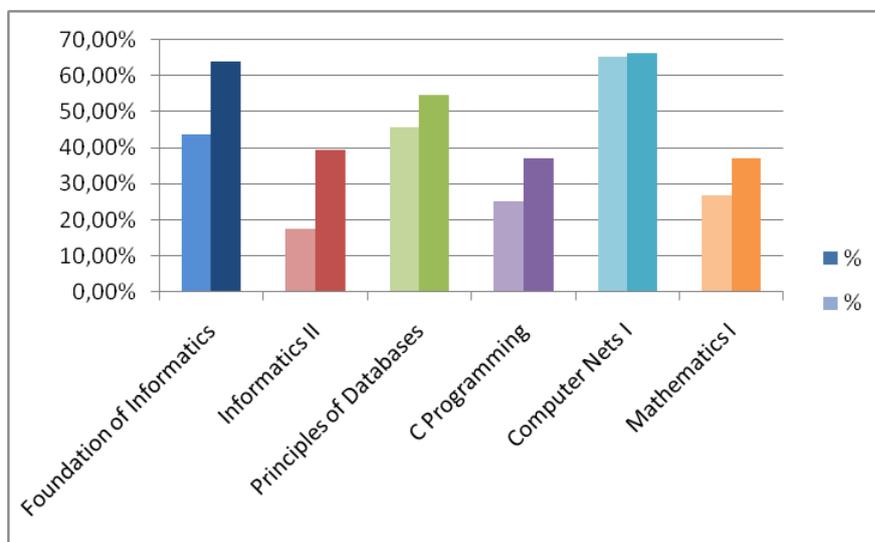


Figura 11. Risultati sui corsi universitari con e-learning statico (barre più chiare davanti) e e-learning personalizzato (barre più scure dietro); tutti i corsi sono stati erogati attraverso IWT

Course	Before e-learning			Without personalized courses			With personalized courses		
	Low level competences	Medium level competences	High level competences	Low level competences	Medium level competences	High level competences	Low level competences	Medium level competences	High level competences
Foundation of Informatics	40%	40%	20%	20%	60%	20%	2%	68%	30%
Informatics II	20%	50%	30%	5%	65%	30%	5%	60%	35%
Principles of Databases	50%	40%	10%	30%	60%	10%	8%	63%	29%
C Programming	40%	40%	20%	20%	60%	20%	4%	64%	32%
Computer Nets I	84%	15%	1%	23%	72%	5%	2%	68%	30%
Mathematics I	50%	35%	15%	33%	52%	15%	5%	59%	35%
Average	47%	37%	16%	22%	62%	17%	4%	64%	32%

Tabella 3. Distribuzione delle competenze (livello basso, medio e alto) sulle varie materie prima e dopo la formazione a distanza, senza e con corsi personalizzati

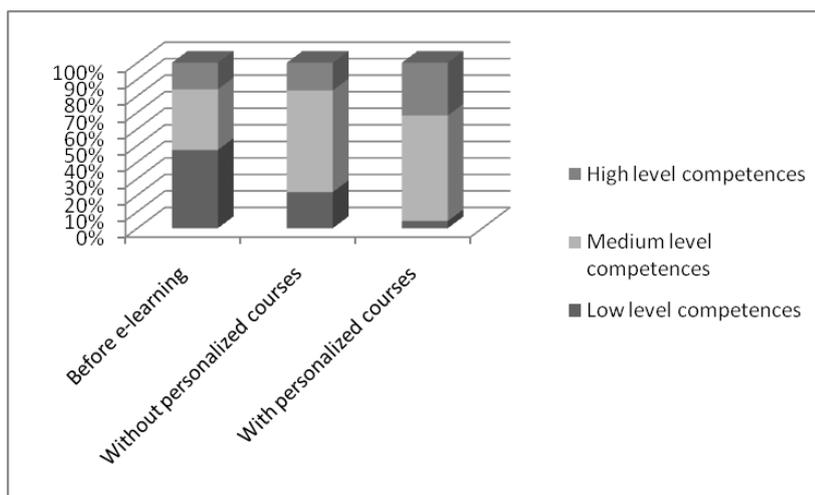


Figura 12. Distribuzione media delle competenze (livello basso, medio e alto) sulle varie materie prima e dopo la formazione a distanza, senza e con corsi personalizzati

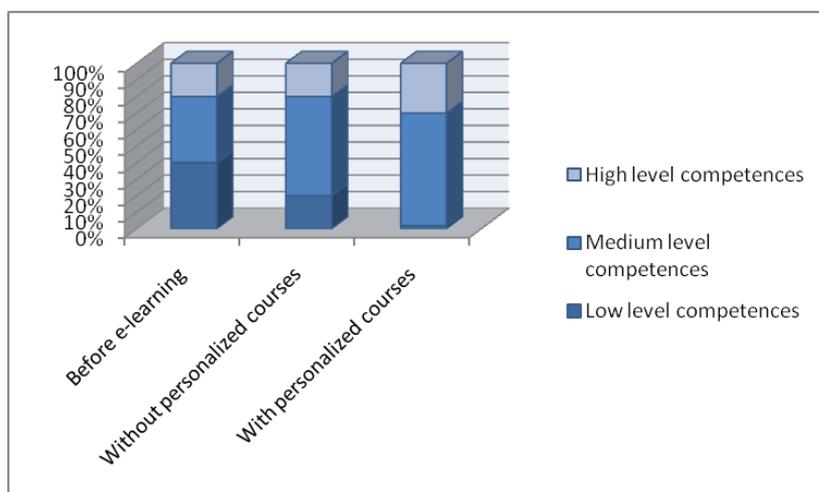


Figura 13. Distribuzione delle competenze (livello basso, medio e alto) relativa alla materia “Fondamenti di Informatica” (*Foundation of Informatics*) prima e dopo la formazione a distanza, senza e con corsi personalizzati

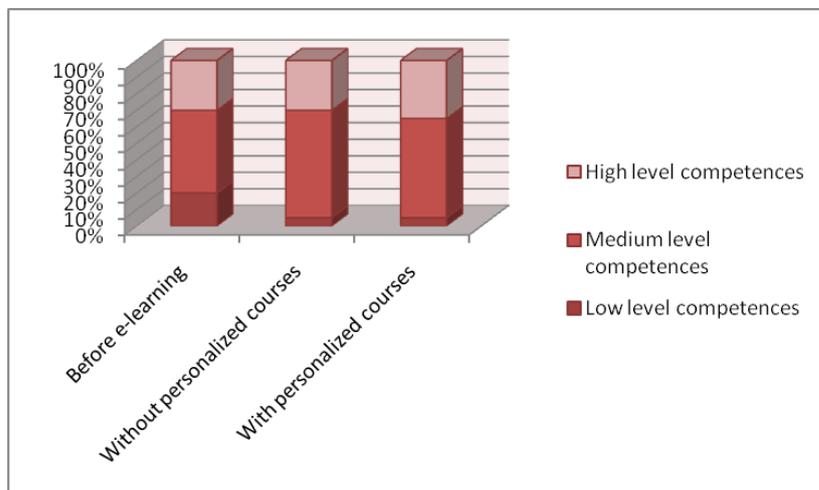


Figura 14. Distribuzione delle competenze (livello basso, medio e alto) relativa alla materia “Informatica II” (*Informatics II*) prima e dopo la formazione a distanza, senza e con corsi personalizzati

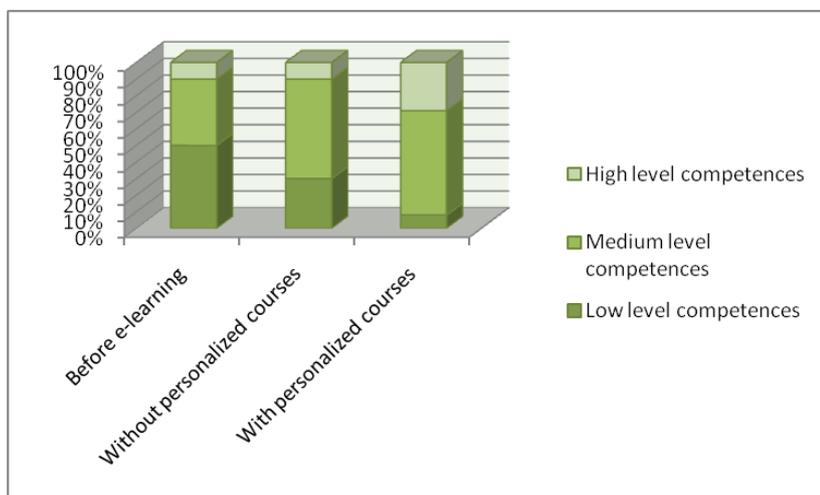


Figura 15. Distribuzione delle competenze (livello basso, medio e alto) relativa alla materia “Fondamenti di Database” (*Principles of Database*) prima e dopo la formazione a distanza, senza e con corsi personalizzati

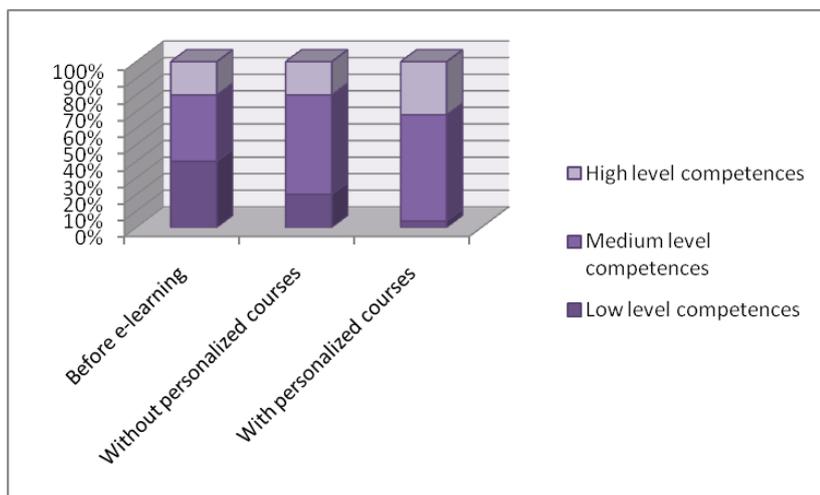


Figura 16. Distribuzione delle competenze (livello basso, medio e alto) relativa alla materia “Programmazione in Linguaggio C” (*C Programming*) prima e dopo la formazione a distanza, senza e con corsi personalizzati

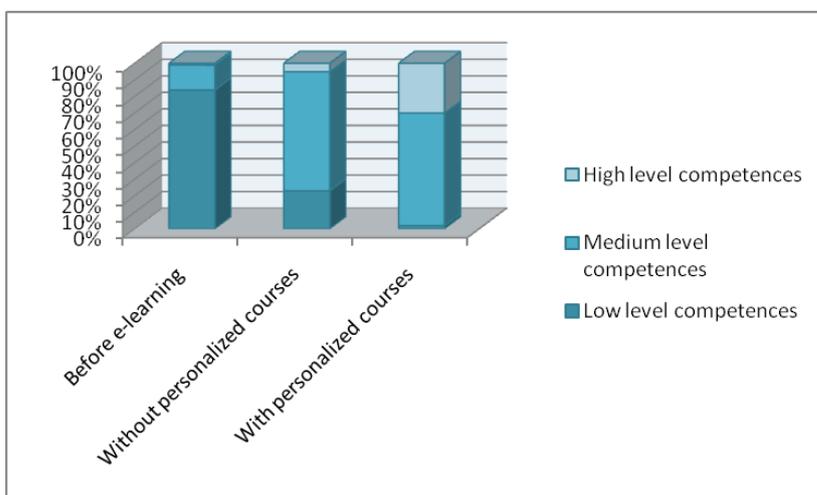


Figura 17. Distribuzione delle competenze (livello basso, medio e alto) relativa alla materia “Reti di calcolatori” (*Computer Nets I*) prima e dopo la formazione a distanza, senza e con corsi personalizzati

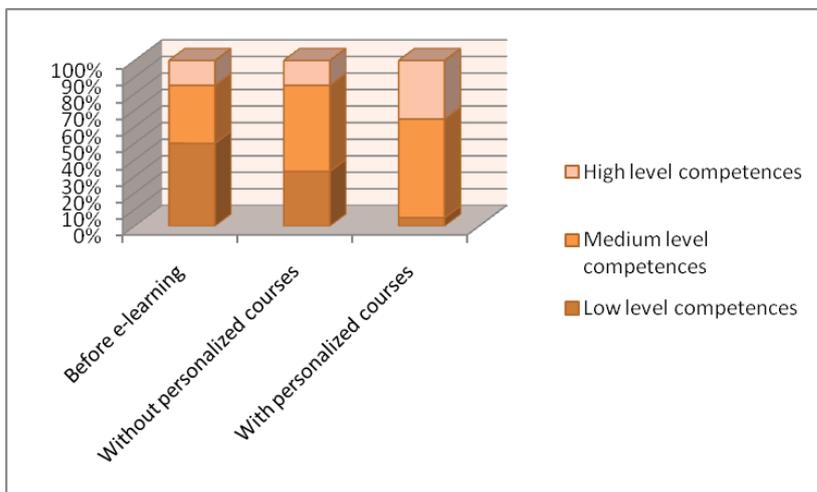


Figura 18. Distribuzione delle competenze (livello basso, medio e alto) relativa alla materia “Matematica I” (*Mathematics I*) prima e dopo la formazione a distanza, senza e con corsi personalizzati

3.2.2 Sperimentazione di IWT nella grande distribuzione

La sperimentazione ha avuto l'intento di verificare l'approccio innovativo della piattaforma IWT nell'erogazione dei percorsi personalizzati nella formazione al di fuori dai contesti scolastici e universitari. In particolare è stata impiegata per l'aggiornamento del personale dell'area commerciale di un grande gruppo della Grande Distribuzione organizzata.

Le unità di personale coinvolte sono state 140 a vari livelli erogando loro percorsi formativi personalizzati su tematiche di gestione fornitori, di gestione listini e condizioni di acquisto. Complessivamente sono stati erogati 13 percorsi formativi per un numero complessivo di concetti pari a circa 300.

La sperimentazione è durata circa 3 mesi articolandosi in varie fasi:

- Individuazione delle tematiche e raccolta contenuti (documenti, presentazioni, audio, video)
- Definizione del dominio ovvero del dizionario di concetti oggetto di formazione
- Disegno delle ontologie ovvero del modello di rappresentazione della conoscenza in cui vengono delineate le relazioni tra i concetti del dominio
- Definizione degli obiettivi degli interventi formativi da erogare
- Caratterizzazione dei profili degli utenti in termini di preferenze e competenze iniziali
- Riorganizzazione e indicizzazione tramite metadata dei contenuti da utilizzare
- Creazione dei test di valutazione
- Caricamento nella piattaforma IWT.

Quando è iniziata l'erogazione, direttamente la piattaforma IWT ha creato per ogni singolo utente un percorso personalizzato, somministrando un test di ingresso e selezionando i contenuti più adatti per rispondere alle esigenze di ognuno.

Il test di ingresso ha consentito anche di delineare un quadro delle competenze in possesso delle unità di personale presenti in azienda.

Tutta l'attività di formazione ha impegnato le persone per circa 5 ore complessive on-line.

Alla fine delle attività sono stati raccolti i dati tracciati dalla piattaforma in merito alle risorse viste, alle risposte date ai test, alle competenze effettivamente acquisite.

Tutto il personale coinvolto è stato valutato prima e dopo il progetto pilota ed i risultati sono decisamente positivi.

Laddove il livello medio iniziale era già positivo si è segnato un netto miglioramento. Dove invece erano state riscontrate delle lacune, i vuoti sono stati colmati con risultati più che soddisfacenti.

La piattaforma si è così dimostrata particolarmente efficace anche in questo contesto formativo sicuramente diverso dalle scuole e le università dove abitualmente viene utilizzata.

I grafici seguenti sono ottenuti confrontando i risultati reali dei test di ingresso e dei test finali degli utenti su ogni singolo percorso erogato.

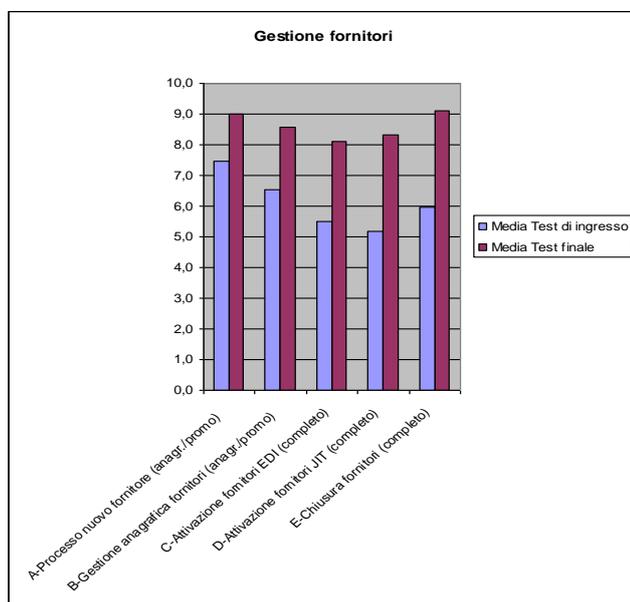


Figura 19. Risultati prima e dopo l'e-learning sui corsi di gestione fornitori

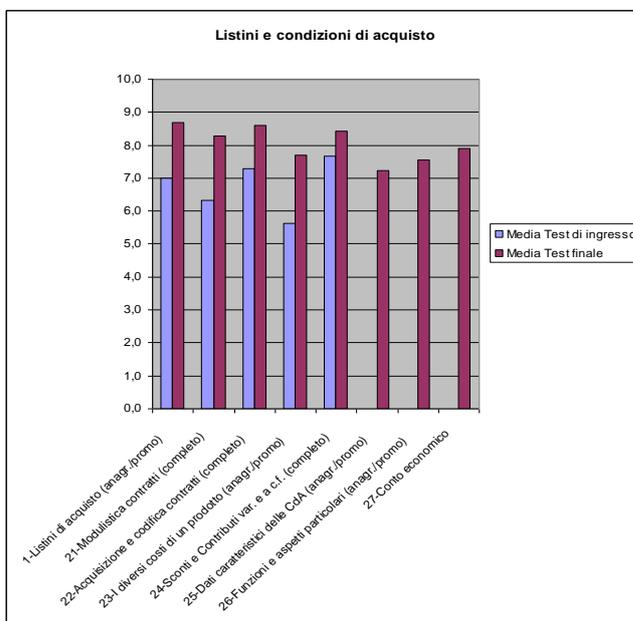


Figura 20. Risultati prima e dopo l'e-learning sui corsi di listini e condizioni di acquisto

I dati puntuali sui vari percorsi sono stati analizzati anche in funzione delle competenze già esistenti in azienda e su come queste sono state modificate a seguito dell'erogazione dei percorsi formativi.

Questo ha infatti consentito di fare ulteriori analisi gli utenti fruitori in due gruppi ed erogando loro due tipologie di approccio: approccio e-learning convenzionale (in altri termini, corsi statici di IWT) e approccio e-learning personalizzato (in altri termini, corsi personalizzati di IWT).

La personalizzazione oltre, di fatto, ad abbattere radicalmente le lacune sulle tematiche trattate, è riuscita anche a coinvolgere un'insieme di utenti che, per la natura dell'approccio e-learning convenzionale, non si riesce coinvolgere. Generalmente le persone con un livello elevato di competenze su una tematica, molto probabilmente si annoiano a ricevere un percorso statico completo all'interno del quale dover individuare gli argomenti per i quali hanno necessità di approfondimenti.

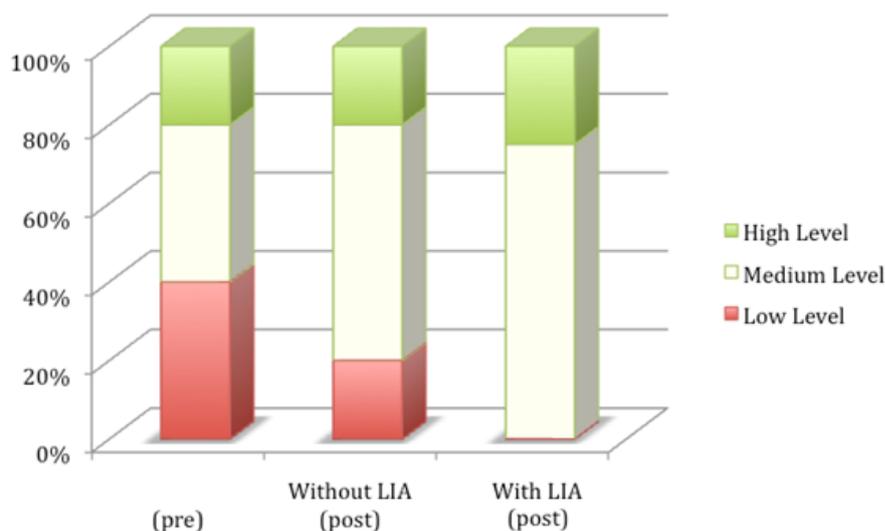


Figura 21. Prima e dopo l'e-learning in IWT senza e con LIA, ovvero con corsi statici e personalizzati

Grazie a questi risultati, l'azienda ha acquisito una licenza di IWT a tempo indeterminato dimostrando così la sua ferma convinzione che l'approccio innovativo di IWT effettivamente porti buoni risultati. L'esperienza condotta all'interno di questa area ha destato l'interesse anche di altri settori dove l'esigenza di aggiornamento e formazione è da ritenersi cruciale, per cui non è escluso che presto l'utilizzo di questa piattaforma si estenda a tutto il personale, dagli uffici ai punti vendita.

3.3 Verifica dei contenuti

Poiché la piattaforma di e-learning IWT è capace di creare percorsi formativi personalizzati selezionando risorse in funzione di un modello di rappresentazione della conoscenza e dei profili degli utenti, può risultare utile verificare, nell'ambito di una determinata tematica, se i contenuti disponibili siano adatti a rispondere alle esigenze degli utenti.

I parametri didattici del contenuto e le preferenze didattiche dell'utente appartengono ad uno stesso spazio di caratteristiche,

pertanto è possibile verificare quanto un determinato contenuto sia vicino alle preferenze di un determinato utente.

In situazioni in cui ci sono molti utenti e molti contenuti, non è detto che si abbia un numero sufficiente di contenuti tale da rispettare al meglio le preferenze degli utenti e, inoltre, questa verifica richiederebbe un numero di calcoli elevato. Un metodo efficace e computazionalmente molto efficiente per eseguire questa operazione di controllo è dato dall'aritmetica degli intervalli, una tecnica largamente impiegata in elettrotecnica nel dimensionamento dei componenti nei circuiti.

3.3.1 L'Aritmetica degli intervalli

L'aritmetica degli intervalli è un metodo sviluppato dai matematici tra il 1950 e il 1960 come approccio da impiegare nella matematica computazionale e nella stima degli errori di arrotondamento.

Mentre l'aritmetica classica definisce operazioni sui singoli numeri, l'aritmetica degli intervalli definisce un insieme di operazioni sugli intervalli (cfr. [41] e [65]).

Invece di lavorare con un reale x (anche nei casi in cui possa essere soggetto ad incertezza), grazie all'aritmetica degli intervalli è possibile lavorare con gli estremi dell'intervallo $[a,b]$ che contiene x .

Avendo una funzione f reale, se applicata ad x produce un risultato $f(x)$ reale (e se x è soggetta ad incertezza, anche $f(x)$ sarà soggetto ad incertezza). Mentre, nell'aritmetica degli intervalli f applicata ad un intervallo $[a,b]$ produce un nuovo intervallo $[c,d]$ i cui estremi sono certi che è dato da tutti i possibili valori di $f(x)$ per ogni $x \in [a,b]$.

Questo approccio è particolarmente indicato nella risoluzione di problemi di ottimizzazione oppure nella stima di parametri tecnici (naturalmente soggetta ad errore). Ad esempio quando vengono dimensionati dei componenti elettronici è ricorrente approcciare la stima di tolleranza in questo modo, poiché consente operazioni di verifica più veloci di quelle analitiche che indicano in modo preciso gli estremi limite superiore e limite inferiore oltre i quali non è possibile andare (cfr. [32] e [34]).

3.3.2 La problematica

Per la creazione di percorsi formativi personalizzati può essere utile verificare, nell'ambito di una determinata tematica, se i contenuti disponibili siano sufficienti a soddisfare le esigenze degli utenti.

Ogni utente è caratterizzato da un insieme di competenze e un insieme di preferenze didattiche. Ogni contenuto è caratterizzato da un argomento a cui fa riferimento e da un insieme di parametri didattici.

I parametri didattici del contenuto e le preferenze didattiche dell'utente appartengono allo stesso spazio di caratteristiche, pertanto è possibile verificare quanto un determinato contenuto sia vicino alle preferenze di un determinato utente.

Quando il numero di utenti e di contenuti è particolarmente elevato, questa operazione può essere particolarmente onerosa e non è detto che si abbia un numero sufficiente di contenuti tale da rispettare al meglio le preferenze degli utenti.

3.3.3 Il modello utente

Ogni utente è caratterizzato da un insieme di competenze e un insieme di preferenze didattiche. Trascurando le competenze, prendiamo in esame l'insieme delle preferenze didattiche (cfr. paragrafo 2.4).

Relativamente ad una particolare tematica (ad esempio una materia oggetto di studio), un utente u è identificato da un vettore di preferenze caratteristiche:

$$\bar{p}_u = (l_u, cd_u, fe_u, tlo_u, d_u, ds_u, mds_u, trd_u, ti_u, li_u)$$

I parametri del vettore corrispondono rispettivamente a: Lingua, Contesto Didattico, Fascia d'Età, Tempo impiegato su un Learning Object, Difficoltà, Densità Semantica, Massima Dimensione Supportata, Tipologia Risorsa Didattica, Tipo di Interazione, Livello di Interazione.

3.3.4 Il modello contenuti

Ogni contenuto è caratterizzato da un argomento a cui fa riferimento e da un insieme di parametri didattici. Trascurando l'argomento e la tematica a cui fa riferimento, prendiamo in esame l'insieme di parametri didattici.

Un contenuto c è identificato da un vettore di parametri caratteristici:

$$\bar{r}_c = (l_c, cd_c, fe_c, tlo_c, d_c, ds_c, mds_c, trd_c, ti_c, li_c)$$

I parametri del vettore corrispondono rispettivamente a: Lingua, Contesto Didattico, Fascia d'Età, Tempo di fruizione del Learning Object, Difficoltà, Densità Semantica, Dimensione, Tipologia Risorsa Didattica, Tipo di Interazione, Livello di Interazione.

Il vettore di parametri caratteristici dei contenuti appartiene allo stesso spazio del vettore di preferenze caratteristiche degli utenti.

3.3.5 La funzione di tolleranza

Dato uno specifico utente u , ci poniamo come primo obiettivo quello di definire una funzione indice di tolleranza in grado di esprimere quanto un utente sia capace di fruire contenuti che non siano esattamente corrispondenti al proprio profilo. Essendo gli aspetti del profilo molto diversi tra loro, facciamo delle congetture per riuscire a raggiungere l'obiettivo preposto.

Il parametro l_u dell'utente, ovvero la lingua conosciuta, è in realtà un vettore di più lingue per ciascuna delle quali è indicato il relativo livello di conoscenza espresso secondo una scala di valori (ad esempio, da livello elementare a eccellente/madre-lingua).

Questi valori vanno analizzati rispetto alle caratteristiche difficoltà d_u e densità semantica ds_u . Infatti, un contenuto in una lingua non ben conosciuta, ma di difficoltà bassa e densità semantica bassa, potrà essere più facilmente comprensibile di un contenuto nella stessa lingua ma di difficoltà maggiore. Questo implica che più questi

parametri d_u e ds_u sono distanti, più l'utente avrà difficoltà a recepire contenuti.

Il parametro cd_u , ovvero il contesto didattico, può assumere secondo il dizionario definito da IMS (IMS Learning Object Metadata...) dei valori di base, ovvero: Scuole inferiori, Scuole superiori, Training, Altro.

E' ragionevole analizzare il parametro cd_u in relazione alla fascia di età fe_u (che a sua volta, da indicazioni IMS, può assumere valori quali: 12-, 12-18, 19-30, 31-50, 50+). Più contesto e fascia di età sono distanti, più l'utente avrà difficoltà a recepire contenuti. Ad esempio, per un utente di 12 anni sarà complicato approcciare tematiche connesse alla formazione professionale.

Il parametro tlo_u – tempo di fruizione del learning object – indica una soglia di tempo entro la quale il livello di attenzione dell'utente si mantiene elevato. Ovviamente sarà maggiormente capace di dedicare attenzione a contenuti che non superino questa soglia.

Il parametro trd_u – tipologia risorsa didattica – può assumere, in base alle specifiche IMS, uno o più dei seguenti valori: *exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, lecture*. In primo luogo, questi valori andrebbero ordinati secondo una logica che vede agli antipodi le tipologie di contenuti di tipo teorico (come narrative text, slide, lecture, ...) e quelle di tipo pratico (come exercise, simulation, experiment, ...). Una volta definito un criterio di ordinamento "topologico", la capacità di un utente sarà espressa, relativamente al parametro trd_u , in termini di distanza geometrica, analizzando, nel contempo, altri due parametri che sono ti_u e li_u , ovvero, rispettivamente, tipo di interazione e livello di interazione. Più i contenuti sono di tipo teorico, più l'interazione dovrà essere limitata perché l'utente sia in grado di tollerarli, viceversa, più i contenuti sono di tipo pratico, più l'interazione dovrà essere elevata.

Infine, il parametro mds_u , massima dimensione supportata, è più di natura tecnologica che pedagogica e viene analizzato esclusivamente quando l'utente ha capacità di connessione o di computazione limitati. In questo lavoro non è stato tenuto presente questo parametro.

Alla luce delle considerazioni appena esposte, la funzione di tolleranza per l'utente sarà la risultante ottenuta attraverso una

combinazione pesata delle seguenti funzioni, opportunamente normalizzate:

$$\begin{aligned} g_1(\bar{p}_u) &= |l_u - d_u| + |l_u - ds_u| + |d_u - ds_u| \\ g_2(\bar{p}_u) &= |cd_u - fe_u| \\ g_3(\bar{p}_u) &= |trd_u - ti_u| + |trd_u - li_u| + |ti_u - li_u| \end{aligned}$$

e del parametro tempo tlo_u .

L'indice di tolleranza dell'utente esprime, in altri termini, la soglia di capacità dell'utente di fruire contenuti diversi da quanto indicato nel proprio profilo.

3.3.6 La funzione di copertura

Analogamente alla funzione di tolleranza per un utente, dato uno specifico contenuto, possiamo definire una funzione indice di copertura attraverso cui esprimere una sorta di capacità del contenuto stesso.

Ad esempio, un contenuto di difficoltà elevata e di densità semantica elevata, avrà una capacità di copertura molto bassa (i.e. pochi utenti saranno in grado di comprenderne correttamente il senso). Viceversa, un contenuto di difficoltà bassa e densità semantica bassa avrà una capacità di copertura molto alta (i.e. sarà comprensibile a molti utenti).

La funzione di copertura di uno specifico contenuto c sarà data dalla combinazione pesata delle seguenti funzioni, opportunamente normalizzate:

$$\begin{aligned} f_1(\bar{r}_c) &= |l_c - d_c| + |l_c - ds_c| + |d_c - ds_c| \\ f_2(\bar{r}_c) &= |cd_c - fe_c| \\ f_3(\bar{r}_c) &= |trd_c - ti_c| + |trd_c - li_c| + |ti_c - li_c| \end{aligned}$$

e del parametro tempo tlo_c .

E' immediato osservare come tutte le componenti della funzione di copertura dei contenuti siano confrontabili con le componenti della funzione di tolleranza degli utenti.

3.3.7 La verifica da effettuare

Le due funzioni sopra definite, funzione di tolleranza degli utenti e funzione di copertura dei contenuti, in base a quanto appena osservato, hanno componenti sicuramente confrontabili, pertanto, possiamo supporre che per verificare se l'insieme di contenuti disponibile sia adeguato per l'insieme di utenti da addestrare su una specifica tematica, occorre verificare che per ogni utente esista almeno un contenuto la cui funzione di copertura sia compatibile con la soglia di tolleranza dell'utente stesso.

Ovvero, sia U l'insieme di utenti u e sia C l'insieme di contenuti c , occorre verificare che:

$$\forall u \in U \quad \exists c \in C : tlo_u \geq tlo_c \quad e \quad g_i(\bar{p}_u) \geq f_i(\bar{r}_c) \quad \forall i = 1..3$$

Questo vuol dire cioè verificare che, dato un argomento, per ogni utente ci sia un contenuto adatto a spiegarlo.

3.3.8 L'aritmetica degli intervalli per ottimizzare la verifica

Secondo l'aritmetica degli intervalli è possibile applicare le funzioni f_i ad un intervallo che descrive lo spazio dei contenuti (il dominio), piuttosto che al singolo contenuto, ottenendo così degli intervalli (il codominio).

Per fare ciò definiamo dei contenuti fittizi \bar{c}^{\max} e \bar{c}^{\min} . Questi contenuti si ottengono prendendo per ciascuna delle componenti, rispettivamente il massimo e il minimo delle relative componenti di tutti vettori \bar{r}_c dei contenuti.

A questo punto definiamo l'intervallo dei contenuti I nel modo seguente:

$$I = [\bar{c}^{\min}, \bar{c}^{\max}]$$

Per cui la funzione di copertura applicata all'intervallo I ci darà degli intervalli H_i :

$$H_i = f_i(I)$$

Una volta determinati gli intervalli H_i la verifica diventerà quindi la seguente:

$$g_i(\bar{p}_u) \in H_i \quad \forall u \in U \quad \forall i = 1, \dots, 3$$

Che consiste cioè nel verificare che per ogni utente esista un contenuto adatto.

Questo è ovviamente vero nell'ipotesi in cui lo spazio dei contenuti si possa correttamente circoscrivere all'intervallo I .

Questo approccio è paragonabile agli approcci utilizzati in geometria computazionale (cfr. [25] e [48]). e costituisce un'alternativa efficace ed efficiente.

In questo modo il numero di confronti da effettuare è linearmente confrontabile con la dimensione dell'insieme U , ovvero con il numero degli utenti. La medesima verifica, senza l'approccio descritto, richiederebbe un numero di confronti pari al prodotto delle dimensioni degli insiemi U e C , ovvero il numero di utenti moltiplicato per il numero dei contenuti disponibili.

L'approccio proposto provato su casi reali sia di digital library consistenti che di utenze variegata dovrà essere ulteriormente verificato, poiché è, inoltre, da approfondire e validare la soluzione proposta per la valutazione della tolleranza degli utenti e della copertura per i contenuti. Nel proporre tali funzioni infatti sono state introdotte delle congetture che vanno provate almeno su base statistica.

3.4 Aggiornamento del profilo utente

Il modello del discente viene aggiornato dopo ogni fase di testing. In particolare, in seguito ad una sequenza di test, si aggiorna la lista dei concetti già presentati al discente e la valutazione dei concetti oggetto del test.

Sia in caso di valutazione positiva del grado di conoscenza di un concetto che in caso di valutazione negativa, si aggiorna non soltanto il grado di conoscenza di quel concetto, ma si abbassa di un valore pari a ε/d il grado di conoscenza di tutti i concetti già appresi e propedeutici a quello corrente. Ovviamente un concetto sarà considerato propedeutico ad un altro se esiste un percorso definito sull'albero delle relazioni d'ordine che collega il primo al secondo. Il valore di d è proprio uguale al numero di archi costituenti tale percorso. Tale parametrizzazione serve ad evitare che la modifica del grado di apprendimento si propaghi molto risalendo l'albero delle propedeuticità.

Infine, occorre osservare che in seguito ad una fase di testing si possono modificare anche le preferenze fornite dal discente. Nel caso in cui una stessa attività didattica è stata proposta n volte (per esempio 2 o 3) in maniera fallimentare al discente, si abbassa di un determinato valore δ il grado di preferenza associato a tutte le coppie (*proprietà, valore*) caratterizzanti la prima attività. Prima di effettuare in automatico tale variazione, si forniscono al docente tutte le informazioni relative allo stato cognitivo del discente insieme ai risultati ottenuti nei vari test. Il docente può decidere di effettuare il cambiamento consigliato dal motore oppure può decidere di fare autonomamente altre modifiche che ritiene più opportune.

3.4.1 Aggiornamento delle competenze di un discente durante la fruizione di un corso personalizzato

E' stato analizzato l'approccio adottato da IWT nell'aggiornamento delle competenze degli utenti durante la fruizione di un corso

personalizzato ed è stato proposto un miglioramento all'algoritmo di valutazione.

Durante la fruizione di un test, quando il discente clicca sul tasto "Consegna", il sistema valuta le risposte ed inserisce i risultati all'interno di un file XML.

All'interno di questo file XML il sistema memorizza il punteggio ottenuto, per ogni domanda del test, nella forma `<voto_ottenuto>/<voto_massimo>`.

Il `<voto_ottenuto>` ed il `<voto_massimo>` possono essere facilmente calcolati in quanto ad ogni domanda è possibile associare un voto per ognuna delle possibili risposte:

- voto per risposta corretta;
- voto per risposta errata;
- voto per risposta non data.

Alla consegna dell'ultimo test di una batteria, viene invocato il servizio che si occupa dell'aggiornamento delle competenze del discente.

Tale aggiornamento si compone di una serie di macro fasi, indicate di seguito:

- fase1: caricamento dello stato cognitivo del discente (prima dell'aggiornamento);
- fase2: calcolo dei risultati di tutti i test della batteria;
- fase3: fusione dei risultati delle prime due fasi;
- fase4: aggiornamento dei prerequisiti;
- fase5: assegnazione della sufficienza ai concetti non testati (fase opzionale).

FASE1: caricamento dello stato cognitivo del discente

Questa fase consiste nel selezionare dal DB lo stato cognitivo del discente che sta fruendo il corso. Lo stato cognitivo si compone di una serie di coppie (`<nome_concetto>`,`<valore>`), dove `<nome_concetto>` rappresenta il nome di un concetto che il discente ha incontrato in un corso e per il quale ha eseguito un test e `<valore>` è il voto, rappresentato in un intervallo 0..10, che il discente stesso ha ottenuto dalla fruizione di quel test.

Inizialmente, ovvero quando il discente non ha mai fruito un corso, lo stato cognitivo è vuoto.

FASE2: calcolo dei risultati di tutti i test della batteria

I corsi personalizzati, oggetto del documento, sono costituiti da sequenze alternate di risorse di spiegazione e risorse di test; ogni sequenza può contenere una o più risorse. Ad ogni sequenza di test contigui faremo riferimento con il termine batteria di test.

Alla consegna dell'ultimo test di una batteria vengono convertiti i punteggi ottenuti per ogni test (quelli della forma $\langle \text{voto_ottenuto} \rangle / \langle \text{voto_massimo} \rangle$) in voti, compresi tra 0 e 10, da associare ai concetti che questi test coprono.

Preso uno dei test, la conversione avviene secondo la seguente procedura:

```

∇ concetto  $C_i$  coperto dal test
  recupera i punteggi delle domande che coprono  $C_i$  e li inserisce in
  punteggi_per_concetto
  contatore ← 1
  votofinale $C_i$  ← 0
  ∇ punteggio in punteggi_per_concetto
    score ← punteggio( $\langle \text{voto\_ottenuto} \rangle$ )
    votomax ← punteggio( $\langle \text{voto\_massimo} \rangle$ )
    vototemp ← (int) ((float)score / (float)votomax * 10)
    se vototemp < 0 allora vototemp ← 0
    votofinale $C_i$  ← votofinale $C_i$  + vototemp
    contatore ← contatore + 1
  votofinale $C_i$  ← votofinale $C_i$  / contatore

```

Immaginiamo ad esempio di fare un test con 3 domande sullo stesso concetto con punti -1 per risposta errata, 0 per risposta non data e 1 per risposta corretta. Immaginiamo che io risponda bene a due domande e male ad una. Il punteggio dovrebbe essere 1 (=1+1-1).

Il massimo che avrei potuto prendere è 3 (=1+1+1). Dunque ho preso 1 su 3, ovvero in decimi: 3,333.

Usando l'algoritmo attuale si avrebbe:

prima domanda:

$\text{vototemp} = 1/1 * 10$

$\text{votofinale}_{C_i} = 10$

seconda domanda:

$vototemp = 1/1 * 10$
 $votofinaleCi = 10 + 10 = 20$

terza domanda:

$vototemp = -1/1 * 10 = -10$ ma siccome è minore di 0 ho 0
 $votofinaleCi = 20 - 0 = 20$

contatore varrà 3 e quindi voto finale $Ci = 20/3 = 6.6666$:
NON CORRETTO AI FINI DELLA VALUTAZIONE!

Per risolvere la problematica individuata, è stata proposta la seguente alternativa:

```

score ← 0
votomax ← 0
∀ punteggio in punteggi_per_concetto
    score ← score + punteggio(<voto_ottenuto>)
    votomax ← votomax + punteggio(<voto_massimo>)
se votomax <> 0 allora
    votofinaleCi ← (float) score / (float) votomax * 10
    se votofinaleCi < 0 allora votofinaleCi ← 0
altrimenti se score < 0 votofinaleCi ← 0
altrimenti votofinaleCi ← 10

```

FASE3: fusione dei risultati delle prime due fasi

In questa fase viene verificato se qualcuno dei concetti testati è già presente nello stato cognitivo del discente (recuperato nella fase 1). Durante questa analisi, fissato un concetto Ci , si possono verificare tre casi:

- Il concetto NON è già presente nello stato cognitivo, allora:
 $votoCi \leftarrow (double)votofinaleCi$
dove $votofinaleCi$ è il voto calcolato sul concetto Ci al termine della fase 2.
- Il concetto è già presente nello stato cognitivo. In questo caso il voto del concetto Ci viene calcolato nel seguente modo:
 $votoCi \leftarrow (votoprofiloCi + (double)votofinaleCi * 3) / 4$
dove $votofinaleCi$ è il voto calcolato sul concetto Ci al termine della fase 2 e per il calcolo del $votoCi$ assume un peso 3 volte superiore a quello già presente nel profilo (ovvero $votoprofiloCi$), a parità di concetto.

- Il concetto NON è stato testato e si trova già nello stato cognitivo, allora:
 $voto_{Ci} \leftarrow votoprofilo_{Ci}$

FASE4: aggiornamento dei prerequisiti

Questa fase consiste nell'aggiornamento del voto dei concetti prerequisito di ognuno dei concetti del percorso di apprendimento, testati durante la fruizione del corso. I prerequisiti di un dato concetto vengono calcolati a partire dalle relazioni stabilite nell'ontologia, utilizzata per la creazione del corso personalizzato.

Dal momento che il valore con il quale aggiornare il voto di un prerequisito dipende dalla sua distanza dal concetto testato, c'è la necessità di calcolare le distanze fra i vari nodi dell'ontologia. Tale distanza è intesa come il numero minimo di archi "IsRequired" che bisogna attraversare per andare da un nodo ad un altro.

L'aggiornamento dei voti dei concetti prerequisito di un dato concetto C_i , viene fatto solo se il voto ottenuto su C_i è inferiore ad un voto soglia presente nel file di configurazione di IWT, alla voce "LIAVOTE". Inoltre tale aggiornamento, se fatto, interessa i prerequisiti che distano al più un certo valore " $max_distanza$ " dal concetto C_i . Il valore " $max_distanza$ " si trova nel file di configurazione di IWT alla voce " $UpdateRequiredRelationLevel$ ".

Di seguito viene mostrato come viene fatto variare il voto dei prerequisiti:

\forall concetto testato C_i
 $double\ delta \leftarrow (LIAVOTE - voto_{Ci}) / LIAVOTE$
 se $delta > 0$
 \forall prerequisito P_j di C_i
 se $distanza\ di\ P_j\ da\ C_i \leq max_distanza$
 $voto_{P_j} \leftarrow voto_{P_j} - (delta / distanza\ di\ P_j\ da\ C_i)$
 se $voto_{P_j} < 0$ allora $voto_{P_j} \leftarrow 0$

L'ordine di fruizione delle risorse di un corso avanzato può essere vincolato dalla relazione "IsRequiredBy". Un'ontologia che presenta due nodi C_1 e C_2 legati da tale relazione:

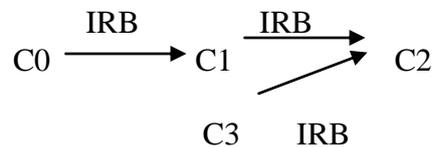


e spiegati da due risorse rispettivamente R1 ed R2, visualizzerà in fase di fruizione del corso prima R1 e poi R2. Lo stesso discorso vale per eventuali test T1 e T2 che coprono tali concetti.

Allo stato attuale il risultato ottenuto durante lo svolgimento dei test influisce solo sui concetti coperti dai test stessi e non tiene conto di eventuali relazioni stabilite tra i concetti.

Con la presente modifica l'aggiornamento del grado di conoscenza di un concetto dovrà tenere conto delle relazioni *IsRequiredBy* presenti nell'ontologia. Nello specifico, utilizzando l'esempio sopra riportato, supponiamo di aver fruito T1 e di aver ottenuto un voto V1, poi fruiamo T2 e otteniamo un voto V2. Il voto finale sul concetto C1 non sarà V1, ma sarà maggiore o minore di V1 in dipendenza del voto V2 ottenuto su C2.

Di seguito viene riportato un esempio per comprendere la modalità di aggiornamento delle competenze tenendo conto delle relazioni *IsRequiredBy*. Supponiamo di avere i seguenti concetti legati da relazioni *IsRequiredBy* come illustrato in figura.



Ad ogni concetto è associato un test e supponiamo che il discente, svolgendo i test, abbia ottenuto i seguenti risultati: C0 voto V0, C1 voto V1, C2 voto V2, C3 voto V3. Al momento di aggiornare le competenze nel profilo verrà utilizzata la seguente procedura:

C2 assegno V2
 C1 assegno $V1 + (V1 - V2) * \beta$
 C3 assegno $V3 + (V3 - V2) * \beta$
 C0 assegno $V0 + (V0 - V1) * \beta$
 Con β costante definita nel file di configurazione di IWT.

Alla luce di queste considerazioni l'algoritmo di aggiornamento viene modificato nel seguente:

\forall concetto testato C_i

$$\text{double } \delta \leftarrow (LIAVOTE - \text{voto}_{C_i}) / LIAVOTE$$

$$\delta \leftarrow \text{round}(\delta, 1)$$

\forall prerequisito P_j di C_i

se $\text{distanza di } P_j \text{ da } C_i \leq \text{max_distanza}$

$$\text{voto}_{P_j} \leftarrow \text{voto}_{P_j} - (\delta / \text{distanza di } P_j \text{ da } C_i)$$

se $\text{voto}_{P_j} < 0$ allora $\text{voto}_{P_j} \leftarrow 0$

In questa fase di aggiornamento del voto dei prerequisiti, può capitare che un prerequisito P_j venga aggiornato più volte. Per meglio comprendere quanto detto, supponiamo che il nostro percorso di apprendimento sia costituito da:

C_1 , C_2 , C_3 e che C_1 è un prerequisito di C_2 che è un prerequisito di C_3 .

Il voto di C_1 verrà aggiornato due volte. La prima in quanto prerequisito diretto di C_2 , la seconda in quanto prerequisito indiretto di C_3 . Naturalmente questo presuppone che max_distanza sia almeno 2 e che il voto ottenuto sui test per C_2 e C_3 sia minore di $LIAVOTE$.

In particolare la prima volta il voto di C_1 viene aggiornato come segue:

$$\text{double } \delta_{C_2} \leftarrow (LIAVOTE - \text{voto}_{C_2}) / LIAVOTE$$

$$\text{voto}_{C_1} \leftarrow \text{voto}_{C_1} - (\delta_{C_2} / 1). \text{ Dove } 1 \text{ è la distanza di } C_1 \text{ da } C_2.$$

La seconda volta il voto di C_1 viene aggiornato come segue:

$$\text{double } \delta_{C_3} \leftarrow (LIAVOTE - \text{voto}_{C_3}) / LIAVOTE$$

$$\text{voto}_{C_1} \leftarrow \text{voto}_{C_1} - (\delta_{C_3} / 2). \text{ Dove } 2 \text{ è la distanza di } C_1 \text{ da } C_3.$$

FASE5: assegnazione della sufficienza ai concetti non testati

Questa fase viene eseguita solo se il docente durante la creazione del corso personalizzato ha abilitato l'opzione "Assegna la sufficienza ai concetti non testati". Essa consiste nell'assegnazione della sufficienza ai concetti, del percorso di apprendimento, che non sono stati testati. Il valore della sufficienza è espresso dal parametro $LIAVOTE$ presente nel file di configurazione di IWT.

Un concetto risulta "non testato" se durante la selezione del materiale didattico esiste una risorsa che lo spiega ma non esiste la risorsa che lo testa.

3.5 Aggiornamento delle preferenze

L'altro aspetto chiave che riguarda la personalizzazione dei corsi è legato all'aggiornamento delle preferenze didattiche del discente. Queste devono essere consistenti con le performance durante la fase di apprendimento. In altri termini occorre tenere presente i seguenti principi:

1. **L'approccio ha portato a risultati positivi? IWT "rinforza" l'approccio usato**
2. **L'approccio ha portato a risultati negativi? IWT "indebolisce" l'approccio usato**

Per fare ciò si considerano da un lato il profilo utente che, in termini di preferenze è descritto dal seguente vettore:

$$\bar{p}_u = (l_u, cd_u, fe_u, tlo_u, d_u, ds_u, mds_u, trd_u, ti_u, li_u)$$

Dall'altro le caratteristiche dei materiali didattici adottati che, per singolo contenuto è identificato dal vettore relativo al metadata, ovvero:

$$\bar{r}_c = (l_c, cd_c, fe_c, tlo_c, d_c, ds_c, mds_c, trd_c, ti_c, li_c)$$

Entrambi i vettori, come già detto in precedenza appartengono allo stesso spazio caratteristico, ovvero ciascuna delle dimensioni componenti può assumere esattamente gli stessi valori. Molto in questi campi sono, di fatto, ordinabili, su altri occorre fare delle congetture.

In ogni caso il comportamento ottenuto è il seguente: il vettore p si avvicina al, o si allontana dal vettore r a seconda delle performance dell'utente u sul contenuto c .

L'avvicinamento o l'allontanamento viene "pesato" attraverso l'impiego di un opportuno parametro che può essere configurato e che viene deciso dal docente o da chi organizza i processi di formazione a distanza e ha contezza del materiale disponibile. In molti casi, infatti, l'assenza di materiale alternativo che spiega i medesimi concetti,

rende questo aspetto praticamente trascurabile. In altri termini, se ho più modi di spiegare uno stesso concetto all'utente, può aver senso vedere con quali contenuti ottengo risultati migliori e ne tengo traccia. Se non ci sono alternative tra i contenuti per un insieme di concetti, questa valutazione risulta essere del tutto inutile.

3.6 Estrazione dell'ontologia

Come ampiamente descritto in precedenza, l'uso delle ontologie è effettivamente un punto di riferimento per molte soluzioni di knowledge management e e-learning. In particolare la piattaforma IWT basa sull'impiego delle ontologie tutti gli aspetti di indicizzazione, di ricerca semantica, di preparazione di percorsi formativi personalizzati.

Ciò nonostante, il processo di costruzione dell'ontologia (cfr. [31]) è un'attività dispendiosa che richiede l'intervento non solo di "tecnici" delle metodologie di rappresentazione, ma di esperti della materia e dei legami sottostanti i concetti che la caratterizzano.

Questo impegno è pressoché giustificato ed affrontabile quando occorre realizzare un progetto di formazione partendo da zero, ovvero dovendo creare anche i contenuti da inserire in un percorso formativo.

Nella maggior parte delle situazioni, però, si tende ad impiegare materiale già disponibile rendendo, pertanto, qualsiasi sforzo di rappresentazione a posteriori del materiale didattico, difficile da comprendere e giustificare.

Da questa evidente difficoltà è nata l'idea di progettare strumenti che siano in grado di estrarre ontologie direttamente dal materiale didattico. Come primo passo è stato affrontato il problema dell'estrazione semi-automatica di conoscenza da pacchetti SCORM, ovvero di riorganizzazione dei componenti di un corso SCORM in un insieme di oggetti didattici a grana più fine e tali da consentire ad IWT la creazione di percorsi di apprendimento personalizzati.

In uno scenario in cui l'e-learning sempre più sta uscendo al di fuori di contesti scolastici e universitari incentivando un mercato di contenuti digitali e di ambienti di apprendimento virtuali, la necessità della condivisione e del riuso di materiali è sempre più sentita. Questa necessità ha quindi giustificato la nascita ed il consolidamento di

specifiche e di standard. Tra questi SCORM è attualmente quello più ampiamente adottato dai Learning Management System (LMS).

Dal punto di vista della capacità di adattarsi alle esigenze degli utenti, SCORM presenta notevoli limitazioni. Il suo modello di navigazione tra i contenuti non considera la necessità di personalizzazione: è fissato per un particolare insieme di *learning experience*.

La scelta di trattare i pacchetti SCORM è stata dettata dunque da due aspetti caratteristici: il suo ampio utilizzo e la sua sostanziale staticità nella fruizione da parte degli utenti. L'obiettivo preposto è stato infatti quello di riuscire a trattare i pacchetti SCORM con IWT senza impegni aggiuntivi e riuscendo cioè ad estrarre un'ontologia direttamente dalla struttura del pacchetto, a decomporre il pacchetto nei suoi componenti chiave e a rimodulare, grazie ad IWT, percorsi di apprendimento personalizzati ed in grado cioè di rispettare esigenze e competenze anche di un insieme variegato di utenti.

3.6.1 Background e stato dell'arte su SCORM

SCORM (Sharable Content Object Reference Model) è un insieme di standard e specifiche realizzato grazie ad un'iniziativa Advanced Distributed Learning (ADL).

È un modello di riferimento per i learning objects, per il confezionamento, l'aggregazione e, in particolare, il riutilizzo in tutti gli LMS che lo supportano. Il nucleo di SCORM comprende tre documenti:

- *Content Aggregation Model (CAM)*
- *Run-Time Environment (RTE)*
- *Sequencing and Navigation (SN)*.

Gli elementi base all'interno di un pacchetto SCORM sono:

- *Asset*: contenuto nella forma più elementare (testi, immagini, suoni, dati in generale che possano essere erogati all'utente via web)
- *SCO* (Shareable Content Object) una collezione di uno o più assets che rappresentano il livello più basso di granularità delle

risorse didattiche che possono essere tracciate da un LMS attraverso il RTE di SCORM.

Questi elementi primari rappresentano i learning object che compongono la struttura gerarchica di un corso, un mezzo pedagogicamente “neutrale” per aggregare risorse e farne il *delivery* all’utente finale.

Il *sequencing* e le regole di navigazione in SCORM sono basate sulle specifiche *IMS Simple Sequencing* (IMS-SS) e sono intese come il mezzo per instradare l’utente da un learning object ad un altro in funzione della fruizione, delle intenzioni o del raggiungimento di determinati obiettivi valutabili oggettivamente (ad es. il superamento di alcuni test).

Molti autori (cfr. [1], [57] e [56]) hanno discusso delle limitazioni di SCORM e del suo modello di metadata.

Dal punto di vista di Abdullah et al. [1] la versione 2004 di SCORM segue la proposta di IMS ma è troppo semplicistico poiché non fornisce meccanismi per l’effettiva implementazione di learning object *adattivi*. Gomes et al. [29] mostrano invece le limitazioni degli standard in generale e di SCORM in particolare. Simoes et al. [57] propongono una estensione di SCORM tale da supportare informazioni trasversali ai learning object come le regole di valutazione, curriculum o bibliografia. Silva et al. [56] presentano un’architettura alternativa per l’impiego di SCORM al fine di fornire una rappresentazione effettiva delle relazioni concettuali tra i learning object.

3.6.2 e-Learning e personalizzazione

I sistemi di e-learning adattivi, anche se ancora pochi sul mercato, mirano a rivoluzionare la formazione a distanza fornendo esperienze di apprendimento personalizzate per ogni discente.

KGTutor è un sistema di tutoring intelligente basato su reti di conoscenza. In [63] Zhuge e Li propongono un modello per la costruzione delle esperienze formative in un modo efficace e gradevole per l’utente.

Questo sfrutta le caratteristiche degli utenti quali la conoscenza pregressa sui temi oggetto di formazione per scegliere, organizzare ed erogare il materiale didattico. Durante la formazione, il sistema può fornire valutazioni oggettive e suggerimenti personalizzati per ogni studente rispettando le loro performance di apprendimento.

Un altro sistema altrettanto rappresentativo che usa tecniche di *Semantic Web* in un ambiente di e-learning è Courseware WatchDog [61]. WatchDog è completamente basata sulle ontologie e usa tecniche di *clustering* per creare viste personalizzate dei learning object. Inoltre, questo ha qualche tecnica per la gestione delle evoluzioni delle ontologie in funzione del contenuto da erogare

In questa panoramica è altresì importante segnalare il progetto PADLR (*Personalised Access to Distributed Learning Repositories*) [45] di cui fa parte una delle sopra citate soluzioni, basato su un approccio *peer-to-peer* per supportare l'accesso personalizzato alle risorse didattiche. Lo scopo di questo framework è quello di creare un'architettura software in grado di aiutare docenti e studenti nella ricerca e nella gestione dei learning object.

La modellazione della conoscenza in questi lavori è legata al processo di creazione ed evoluzione delle ontologie.

3.6.3 Ontology learning

In letteratura l'ontology learning indica il processo di estrazione delle rappresentazioni ontologiche dal corpo non strutturato di documenti di testo. Questo processo è ovviamente un processo di acquisizione della conoscenza dal testo ed ha molto in comune con le tecniche e le metodologie di *Natural Language Processing* (NLP), *Artificial Intelligence* (AI), e *Machine Learning* (cfr. [36] e [62]).

I sistemi basati su queste tecniche estraggono termini direttamente dal testo: documenti, siti web, data warehouse, scambi di dati tra utenti all'interno di virtual community. Quindi filtrano i termini estratti usando algoritmi di NLP e tecniche statistiche che eseguono analisi comparativa tra differenti domini. I concetti vengono collegati tra loro in accordo a relazioni tassonomiche o ad altre relazioni semantiche. Le relazioni sono estratte usando basi di conoscenza come WordNet (cfr. [72]).

Esempi di estrazioni di ontologie con queste tecniche sono descritte in [44] e [62].

3.6.4 Il modello proposto per l'estrazione di ontologie

Come già ampiamente descritto nel precedente capitolo, l'ontologia adottata in IWT è un grafo con un numero limitato di relazioni (*HasPart*, *IsRequiredBy*, *SuggestedOrder*).

Supponiamo di voler modellare un dominio in cui un concetto *C* sia suddiviso in tre sottoconcetti *C1*, *C2* e *C3*, ciascuno trattato da 3 differenti LO: *LO1*, *LO2* e *LO3*. Supponiamo inoltre che i concetti siano legati da relazioni di prerequisito, ovvero *C1* sia prerequisito di *C2* e *C2*, a sua volta, sia prerequisito di *C3*.

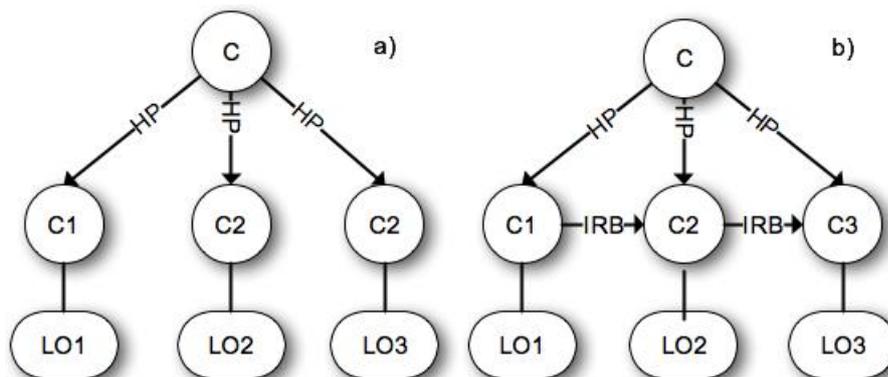


Figura 22. Ontologie e relazioni HP (*HasPart*) e IRB (*IsRequiredBy*)

L'insieme dei contenuti *LO1*, *LO2* e *LO3* spiega di fatto i concetti *C1*, *C2* e *C3*, ovvero l'intero concetto *C*.

Costruire un'ontologia richiede l'identificazione dei concetti e quindi la definizione delle relazioni tra queste nel rispetto delle tipologie delle relazioni consentite e dei vincoli a cui esse sono soggette.

Nonostante la definizione di standard come OWL (cfr. [73]) e la disponibilità di tool di authoring di ontologie alquanto facili e popolari come Protégé (cfr. [74]), creare un'ontologia di dominio è

un'operazione generalmente noiosa, dispendiosa in termini di tempo e facilmente soggetta ad errori poiché gli utenti possono non aver competenze tecniche nella rappresentazione o di dominio per la concettualizzazione.

Anche IWT è "allineato" a queste considerazioni: offre tool di authoring facili da usare, offre il vantaggio della personalizzazione dei corsi, ma soffre della complessità della modellazione del dominio attraverso le ontologie.

L'approccio proposto mira a risolvere in maniera quasi del tutto automatica questo problema partendo da pacchetti SCORM e soprattutto senza l'ausilio di utenti che abbiano competenze da "ingegneri della conoscenza".

Gli elementi coinvolti sono i seguenti:

- L'ontologia di dominio
- Un insieme di learning object ottenuti spezzettando il pacchetto SCORM
- I metadata per questi learning object ottenuti.

Il processo è semiautomatico in quanto l'ontologia frutto dell'estrazione automatica viene presentata all'utente, il quale può decidere se vuole prenderla così com'è o apportare modifiche ad eventuali anomalie o incogruenze.

L'approccio proposto non si basa su tecniche di NLP o *machine learning* per due motivi.

Primo, si lavora ad un numero generalmente piccolo di documenti, mentre le tecniche di NLP richiedono test corposi da elaborare e quindi molti tra gli algoritmi più efficaci in questo, non possono essere applicati.

Secondo, eventuali elementi legati alla struttura implicita o esplicita che sia, come ad esempio la formattazione, la suddivisione in capitoli o in moduli (i.e. gli SCO nel pacchetto) non verrebbero presi in considerazione.

L'idea dietro l'approccio proposto è quella di trarre vantaggio sia dalla conoscenza esplicita che da quella implicita al pacchetto SCORM.

Nel pacchetto SCORM l'aggregazione e le strutture di navigazione sono pianificate con un preciso scopo: il delivery di una determinata

esperienza didattica. Pertanto, al primo livello il processo di estrazione fa affidamento all'abilità di chi ha progettato il pacchetto.

In uno scenario classico, il corso SCORM è logicamente diviso in moduli e ogni modulo è diviso in lezioni; moduli e lezioni sono realizzati attraverso *SCO* e *Asset*. È possibile assumere che una lezione sia disegnata per spiegare un concetto specifico mentre un modulo, che di fatto è un'aggregazione di lezioni, ha l'obiettivo di spiegare un macro-concetto (i.e. un concetto che possa essere suddiviso in concetti più elementari).

Il progettista del pacchetto SCORM definisce anche le regole di navigazione per consentire al LMS di erogare una sequenza corretta di moduli e lezioni, cioè la giusta sequenza di *SCO* e *Asset*.

Queste informazioni vengono tenute in considerazione dall'approccio proposto nel disegno della struttura ontologica.

L'intero processo può essere sintetizzato come segue:

1. Creazione dell'ontologia *draft* che riproduce la struttura del pacchetto SCORM
2. Analisi delle risorse per estrarre sub-ontologie da integrare all'ontologia *draft*
3. Suddivisione delle risorse e definizione dei metadata
4. Risoluzione di eventuali anomalie attraverso un'interazione con l'utente.

Creazione dell'ontologia *draft*

In questa fase si lavora sulle informazioni dello SCORM CAM.

Il contenuto del pacchetto SCORM è infatti strutturato in *organization*, degli elementi che contengono SCO e identificano una sequenza di regole da applicare in funzione delle tipologie di utenti. Se il pacchetto contiene più *organization* si otterranno più ontologie, una per ogni *organization*.

D'ora in avanti saranno presi in considerazione soltanto pacchetti con una sola *organization*.

Per creare l'ontologia vengono applicate le seguenti regole:

1. Il concetto radice dell'ontologia corrisponde alla radice dell'*organization*
2. Per ogni elemento nella gerarchia viene creato un concetto descritto con il titolo dell'elemento stesso

3. Per ogni coppia di elementi in una relazione padre-figlio nella gerarchia del pacchetto SCORM viene creata una relazione *HasPart* tra i corrispondenti concetti nell'ontologia *draft*.

Quindi viene assegnata una semantica alle regole di navigazione presenti nel pacchetto.

No.	Name	Description	Value Space	Default Value
1	Sequencing Control Choice	Indicates that a <i>Choice</i> navigation request is permitted (True or False) to target the children of the activity.	boolean	True
2	Sequencing Control Choice Exit	Indicates whether the activity is permitted to terminate (True or False) if a <i>Choice</i> navigation request is processed.	boolean	True
3	Sequencing Control Flow	Indicates the <i>Flow Subprocess</i> may be applied (True or False) to the children of this activity.	boolean	False
4	Sequencing Control Forward Only	Indicates that backward targets (in terms of Activity Tree traversal) are not permitted (True or False) from the children of this activity.	boolean	False
5	Use Current Attempt Objective Information	Indicates that the Objective Progress Information for the children of the activity will only be used (True or False) in rollup if that information was recorded during the current attempt on the activity.	boolean	True
6	Use Current Attempt Progress Information	Indicates that the Attempt Progress Information for the children of the activity	boolean	True

Tabella 4. Controlli di *Sequencing* in SCORM

Ad ogni regola di navigazione viene fatta corrispondere una relazione d'ordine. Per esempio si aggiungono relazioni *SuggestedOrder* tra concetti corrispondenti a elementi se Sequencing Control Choice = true. Viene creata una relazione *IsRequiredBy* se Sequencing Control Choice = false.

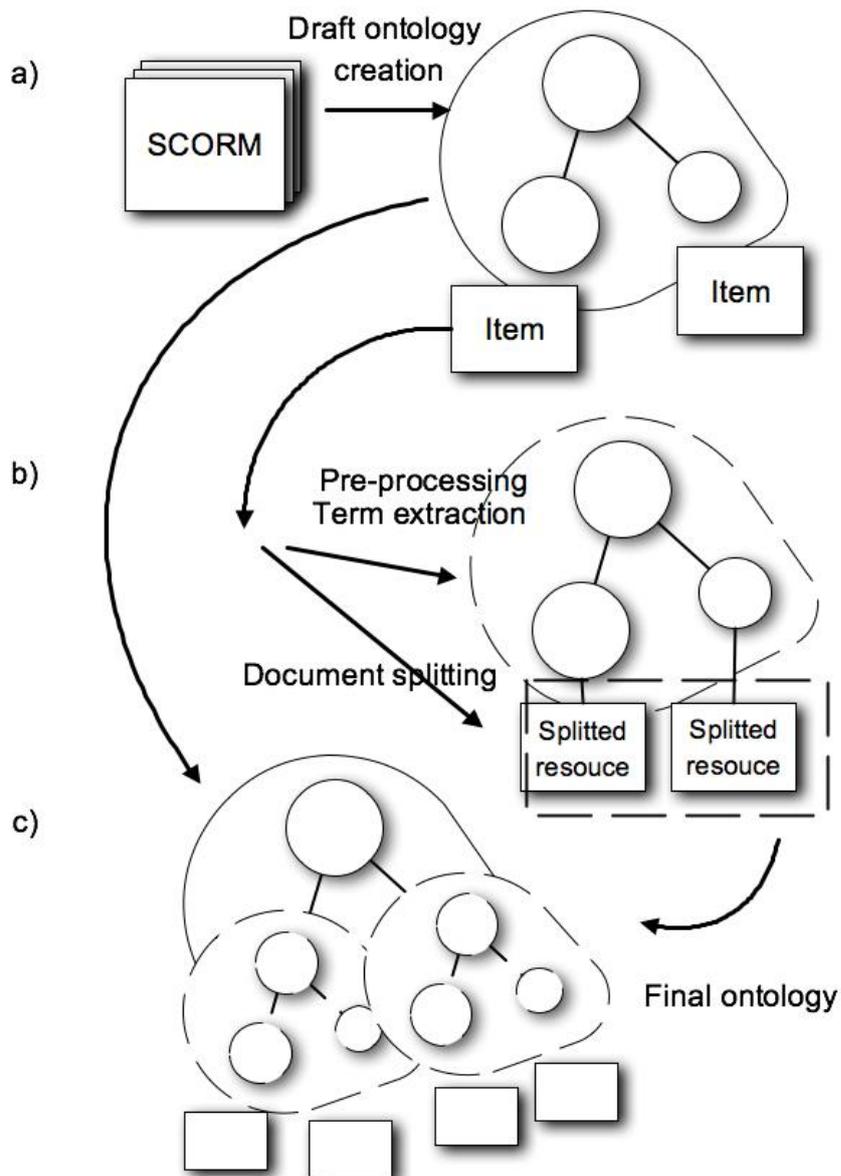


Figura 23. Processo di estrazione dell'ontologia: a) l'ontologia draft è creata analizzando la struttura del pacchetto SCORM; b) la sub-ontologia è creata a partire dagli elementi nella struttura; c) l'ontologia finale assemblata

Resources processing

In profondità il processo analizza gli elementi base (*Item/Assets*) che compongono il pacchetto.

È importante che questi elementi abbiano una struttura definita. La sperimentazione è stata condotta su file PPT e DOC, ma può essere facilmente estesa ad altri formati.

È possibile distinguere le seguenti fasi:

1. **Preprocessing**. I documenti vengono preparati per l'estrazione; è possibile distinguere diverse fasi
 - a. Conversione di formato: i documenti vengono convertiti in XML
 - b. Stemming: gli elementi sono ridotti al loro *stem* ovvero alla loro forma radice (ad es. *writing* > *write*)
 - c. Part-of-Speech tagging: i termini nel documento sono annotati come corrispondenti a particolari parti del testo (nomi, verbi, aggettivi, avverbi, etc.)
 - d. Stopword list: i termini che non portano informazioni utili (come gli articoli e le congiunzioni) vengono cancellati.
 - e. Identificazione dei sinonimi: viene impiegato il database lessico di WordNet per acquisire sinonimi per ogni termine
 - f. Estrazione dei termini: tutti questi termini estratti vengono usati per concettualizzare il dominio di conoscenza.
 - g. Interazione con l'utente: laddove necessario per anomalie o incongruenze.

2. **Creazione di concetti e relazioni**. Sono stati impiegati diversi algoritmi statistici e di data mining per identificare i concetti e le relazioni che li coinvolgono nell'ontologia. In prima battuta viene applicato un algoritmo che individua la frequenza con cui appaiono tutti i termini quindi vengono adottati degli algoritmi di clustering (cfr. [37]) che consentono di individuare concetti e gerarchia tra questi

(rendendola in termini di relazioni *HasPart*). Con un altro algoritmo si identificano le coppie di termini ricorrenti nel testo per identificare le relazioni tra i concetti a cui essi si riferiscono (cfr. [36]).

3. ***Document splitting***. Algoritmi di suddivisione dei documenti in funzione della posizione dei termini rilevanti o dei cluster di termini individuati al passo precedente. Nel caso più semplice, se un termine è rilevante per un insieme di slide, queste saranno raggruppate in un unico learning object che tratta il concetto corrispondente al termine rilevante identificato.

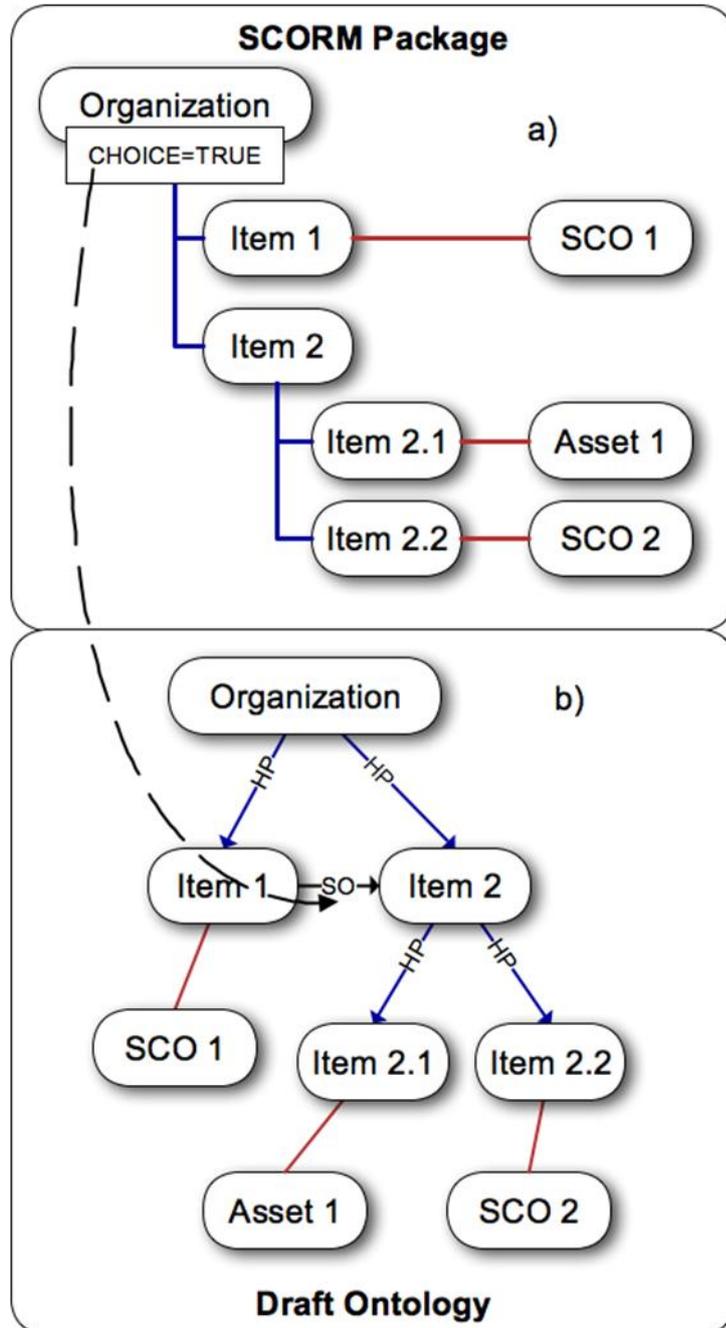


Figura 24. Mapping della struttura del pacchetto SCORM nella gerarchia tra i nodi dell'ontologia

Il prototipo è stato realizzato impiegando le librerie Jakarta POI (cfr. [75]), che forniscono un insieme di funzionalità di accesso a documenti Office per il sottosistema di gestione dei documenti, le librerie ADL RTE per l'analizzatore dei pacchetti SCORM, jGraph (cfr. [76]) e Graphviz (cfr [77]) per il sistema di visualizzazione. Infine per l'elaborazione del testo è stato impiegato un insieme di librerie (cfr. [78] e [79]).

3.7 Estrazione automatica di metadata

Per garantire il riuso dei learning object nel mondo della didattica a distanza, occorre una buona indicizzazione. Occorre cioè corredare ogni contenuto di una descrizione *machine-understandable* completa sia in termini tecnologici che pedagogici, che consenta, da un lato, di comprenderne requisiti e limiti per un corretto utilizzo, dall'altro, di rendere più efficienti tutte le eventuali operazioni di ricerca e fruizione.

La descrizione comprensibile alla macchina dei contenuti viene mantenuta in strutture dati alquanto consolidate che prendono il nome di metadata. La compilazione dei metadata è un'operazione noiosa e decisamente *time-consuming* ma estremamente importante dato che da essa dipende la scelta dei contenuti didattici che devono essere quanto più congeniali per gli studenti in un'ottica *learner-centred*.

L'obiettivo preposto è quello di automatizzare il più possibile questa operazione presentando un approccio metodologico in grado di estrarre i metadata direttamente dai file che costituiscono i learning object.

In letteratura sono presenti molteplici metodi che riescono a caratterizzare in automatico gli aspetti tecnologici relativi al contenuto (formato, dimensioni, requisiti HW e SW per poterli visualizzare), rari ed incompleti sono invece i tentativi di descrizione degli aspetti pedagogici (tipologia di risorsa didattica, densità semantica, difficoltà, tempo per la comprensione, livello di interazione, etc.). Mettendo insieme elementi di teoria dell'informazione, modelli di learning, analisi statistiche ed euristiche realizzate ad hoc, si è cercato di proporre un approccio efficace per l'estrazione di un ampio insieme di

campi dei metadata. I risultati della sperimentazione effettuata a partire da un prototipo implementato sono particolarmente incoraggianti da far pensare che questo approccio possa essere ingegnerizzato in una soluzione di mercato e applicato anche al di fuori dell'ambito e-learning.

3.7.1 La problematica

La problematica generale affrontata è quella di estrarre in automatico i metadata dai contenuti per i fini di catalogazione e riuso. Ci si confronta, inoltre, con una problematica più specifica collegata ad IWT, ovvero alla sua capacità di erogare corsi personalizzati che, tenendo conto delle caratteristiche degli utenti e garantendo flessibilità al livello dei contenuti e dei modelli di apprendimento, risultano più efficaci ed efficienti dei percorsi formativi statici di e-learning "classico".

La capacità di erogare corsi personalizzati si basa fondamentalmente su una sofisticata modellazione sia della conoscenza dei domini di interesse e dei contenuti effettuata grazie ad ontologie e metadata, sia delle competenze acquisite e delle preferenze didattiche dell'utente per mezzo di un profilo utente che viene costantemente aggiornato.

La validità dell'approccio IWT è stata avvalorata da un attento confronto con soluzioni, metodologie e modelli allo stato dell'arte e dai risultati ottenuti in molteplici attività di sperimentazione condotte in ambito universitario e aziendale descritte nel paragrafo 3.2.

Grazie alla sperimentazione sono state fatte riflessioni interessanti che sembrano convincere che l'approccio della personalizzazione di IWT porti a risultati migliori rispetto all'approccio tipico dell'e-learning, ovvero all'erogazione di percorsi sequenziali statici di contenuti. Per contro, questo approccio pur se totalmente automatizzato, richiede un impegno in fase di start-up in cui rientra lo sforzo per definire i metadata delle risorse didattiche a disposizione, cioè indicizzarle e descriverle in modo completo per poterle poi utilizzare nel modo più appropriato possibile.

3.7.2 I metadata

Esistono diversi standard e specifiche adottate per l'indicizzazione dei contenuti. Tra questi i più utilizzati sono il *Dublin Core* (cfr. [80]), *IMS Metadata* (definito dall'*IMS Global Learning Consortium* cfr. [81]) e *IEEE LOM* (acronimo di *Learning Object Metadata* definito dal *Learning Technology Standardization Committee, LTSC* dell'*IEEE* cfr. [83]). *Dublin Core* è uno standard generale piuttosto diffuso ed utilizzato da varie applicazioni, mentre gli altri due sono standard specifici per la didattica e sempre in competizione per conquistare una posizione dominante nel mercato dell'e-learning.

A differenza del *Dublin Core*, sia la specifica *IMS Metadata* che lo standard *LOM* hanno una struttura gerarchica. Ogni elemento può essere *obbligatorio*, *opzionale* o *condizionale*. Alcuni attributi possono avere, come valore, una lista di termini di uno specifico vocabolario, piuttosto che un singolo termine.

Dall'insieme alquanto ampio dei campi caratterizzanti i metadata, verranno presi in considerazione principalmente quelli riportati nella seguente tabella.

	Campi Metadata	Valori	Cardinalità
1	Lingua della risorsa	{ IT, EN, ES, FR, ... }	1-N
2	Dizionario/Concetti	Campo di testo	0-1/1<=>1-N
3	Tipo di file associato	MIME Type	0-N
4	Dimensione	Espresso in byte	0-1
5	Tipologia risorsa didattica	{ AUTOVALUTAZIONE, DIAGRAMMA, ESAME, ESERCIZIO, ESPERIMENTO, FIGURA, GRAFICO, INDICE, LEZIONE, PROBLEMA, QUESTIONARIO, SIMULAZIONE, SLIDE, TABELLA, TESTO }	0-N
6	Durata	Formato: HH MM SS	1
7	Tipo di interazione	{ Espositivo, Attivo, Misto }	0-1
8	Livello di interazione	{ Molto basso, Basso, Medio, Alto, Molto Alto }	0-1
9	Difficoltà	{ Molto facile, Facile, Medio, Difficile, Molto difficile }	0-1
10	Densità semantica	{ Molto bassa, Basso, Media, Alta, Molto Alta }	0-1
11	Tempo impiegato su un Learning Object	Formato: HH MM SS	0-1

Tabella 5. Campi del metadata con possibili valori e relativa cardinalità ammessa

La scelta ricade su questi poiché sono quelli principalmente usati dalla piattaforma IWT citata in precedenza e sui quali l'operazione di caricamento, sia attraverso una form che attraverso maschere visuali di inserimento, richiede comunque del tempo.

3.7.3 Stato dell'arte

Esistono diversi approcci e modelli in letteratura per l'estrazione automatica di metadata direttamente dai contenuti.

Tra questi, particolarmente efficaci sembrano essere quelli sviluppati da National Library della Nuova Zelanda (cfr. [84]), da HATII della University of Glasgow (cfr. [86]), o ancora altri modelli (cfr. [24], [87] e [88]). Tutti sono in grado di produrre informazioni a partire da un ampio insieme di contenuti: documenti PDF, immagini, documenti Microsoft Office, file audio, etc. ma producono principalmente informazioni di tipo tecnico su come è fatto il

contenuto o quali applicazioni siano necessarie alla sua fruizione (cfr. [30]), mentre manca buona parte dei dati utilizzabili ai fini dell'e-learning come, in particolare, i seguenti: *Tipologia risorsa didattica, Tempo impiegato su un Learning Object, Tipo di interazione, Livello di interazione, Difficoltà, Densità semantica*.

Su questo versante, per contro, esistono vari modelli come quello citato da Ronsivalle (cfr. [50]) quale Integrazione delle tassonomie di Bloom secondo Anderson, Marzano e Romiszowski (cfr. [8]), oppure altri (cfr. [52], [58] e [43]). Questi modelli, seppur efficaci dal punto di vista della caratterizzazione pedagogica dei contenuti lavorano principalmente a posteriori, ovvero sulla base di rilevazioni fatte su campioni di utenti sul tempo impiegato, piuttosto che su altri aspetti, deducendo vari parametri tra cui quelli relativi a complessità o densità semantica. Essi si basano però su un principio non ben definito di quantità di informazioni presenti nei contenuti. La loro validità in ogni caso è stata considerata un punto di riferimento per poter avvalorare le euristiche descritte utilizzandoli in combinazione con la rappresentazione semantica dei contenuti.

3.7.4 Estrazione automatica dei metadata

Il modello proposto vuole mettere insieme tutti questi aspetti e cercare di operare in maniera integrata, producendo metadata a partire da dettagli tecnici dei contenuti stessi e sfruttando poi considerazioni provenienti dalla teoria dell'informazione (cfr. [54]).

Il processo presuppone che avvenga a monte una caratterizzazione semantica del Learning Object (LO), ovvero che venga assegnato (con procedimento manuale o automatico) al campo indicato nella Tab.1 come 2 - *Dizionario/Concetti*, l'insieme di concetti del dizionario a cui sia legato il contenuto. In altre parole: relativamente ad una tematica, quali concetti spiega. Contestualmente a questa operazione può essere indicata anche la lingua del contenuto desumibile appunto dal dizionario.

Detto ciò, il procedimento inizia la sua elaborazione sul LO considerando l'insieme di file da cui è composto. L'idea da cui partire è particolarmente semplice, ovvero: ogni file ha una dimensione nota e un'estensione che ne caratterizza il tipo MIME.

Questo consente di determinare la dimensione totale del LO e di popolare il campo 3 - *Tipo di file associato*. Essendo un campo con cardinalità 0-N, è possibile caricare un insieme di elementi ottenuti determinando un MIME TYPE per ogni file.

Per individuare il MIME TYPE da caricare si fa riferimento a schemi reperibili in rete (ad esempio si veda [82]).

A partire dai MIME TYPE è possibile determinare anche la tipologia di risorsa didattica e quale tipo e livello di interazione siano previsti con l'utente.

Da un'attenta analisi sulle caratteristiche dei LO sono state definite le seguenti euristiche da utilizzare per ogni file che compone la risorsa:

Nome	MIMEType	Tipologia risorsa didattica	Tipo di interazione	Livello di interazione
Regola 1	application/excel	TABELLA	MISTO	MEDIO
Regola 2	application/msword	TESTO	ESPOSITIVO	MOLTO BASSO
Regola 3	application/mspowerpoint	SLIDE	ESPOSITIVO	BASSO
Regola 4	application/pdf	TESTO	ESPOSITIVO	MOLTO BASSO
Regola 5	application/wordperfect*	TESTO	ESPOSITIVO	MOLTO BASSO
Regola 6	application/* (tranne le precedenti)	SIMULAZIONE	ATTIVO	MOLTO ALTO
Regola 7	text/*	TESTO	ESPOSITIVO	MOLTO BASSO
Regola 8	audio/*	LEZIONE	ESPOSITIVO	MOLTO BASSO
Regola 9	video/*	LEZIONE	ESPOSITIVO	BASSO
Regola 10	image/*	FIGURA	ESPOSITIVO	MOLTO BASSO
Regola 11	*world/*	SIMULAZIONE	ATTIVO	MOLTO ALTO
Regola 12	*message/*	TESTO	ESPOSITIVO	MOLTO BASSO
Regola 13	*conference/*	LEZIONE	MISTO	MEDIO
Regola 14	drawing/*	GRAFICO	ESPOSITIVO	MOLTO BASSO
Regola 15	chemical/*	DIAGRAMMA	ESPOSITIVO	MOLTO BASSO
Regola 16	model/*	SIMULAZIONE	ATTIVO	MOLTO ALTO
Regola 17	paleovu/*	GRAFICO	ESPOSITIVO	MOLTO BASSO
Regola 18	*/*metafile*	INDICE	ESPOSITIVO	BASSO
Regola 19	www/*	INDICE	ESPOSITIVO	BASSO

Tabella 6. Regole euristiche per la determinazione dei campi Tipologia di risorsa didattica, Tipologia e Livello di Interazione del metadata a partire dal MIME Type

Il campo 5 - *Tipologia di risorsa didattica* ha cardinalità 0-N pertanto sarà aggiunta una tipologia per ogni file che faccia parte della risorsa.

Il campo 7 - *Tipo di interazione* ha cardinalità 0-1, quindi, in caso di più parti (i.e. più file) della stessa risorsa, occorre combinare insieme i valori (per semplicità, a due a due) secondo la tabella riportata di seguito.

	ESPOSITIVO	MISTO	ATTIVO
ESPOSITIVO	ESPOSITIVO	MISTO	MISTO
MISTO	MISTO	MISTO	ATTIVO
ATTIVO	MISTO	ATTIVO	ATTIVO

Tabella 7. Regole euristiche per la determinazione del Tipo di interazione di un LO composto da più file

Il campo 8 - *Livello di interazione* ha cardinalità 0-1, quindi, in caso di più parti (i.e. più file) della stessa risorsa, occorre combinare insieme i valori (per semplicità, a due a due) secondo la tabella riportata di seguito.

	MOLTO BASSO	BASSO	MEDIO	ALTO	MOLTO ALTO
MOLTO BASSO	MOLTO BASSO	BASSO	BASSO	MEDIO	ALTO
BASSO	BASSO	BASSO	MEDIO	MEDIO	ALTO
MEDIO	BASSO	MEDIO	MEDIO	ALTO	ALTO
ALTO	MEDIO	MEDIO	ALTO	ALTO	MOLTO ALTO
MOLTO ALTO	ALTO	ALTO	ALTO	MOLTO ALTO	MOLTO ALTO

Tabella 8. Regole euristiche per la determinazione del Tipo di interazione di un LO composto da più file

Per velocizzare le operazioni di analisi e di calcolo della durata, che tipicamente vengono effettuate accedendo al contenuto di ogni file, viene fatta una stima approssimativa per ciascuno degli oggetti (file) che compongono il LO.

Per i testi si adotta un approccio analogo adottato dai “*Writer Services*” (cfr. [88]). Secondo questo approccio è possibile infatti avere un ordine di grandezza del numero di parole presenti in un file di testo in funzione della sua dimensione, poiché la dimensione è un aspetto determinato totalmente in automatico.

In realtà questo approccio è stato adeguato con delle stime realistiche prendendo documenti in vari formati (txt, html, pdf, doc, rtf, docx) e con un numero di parole note (10, 100, 1000, 10000).

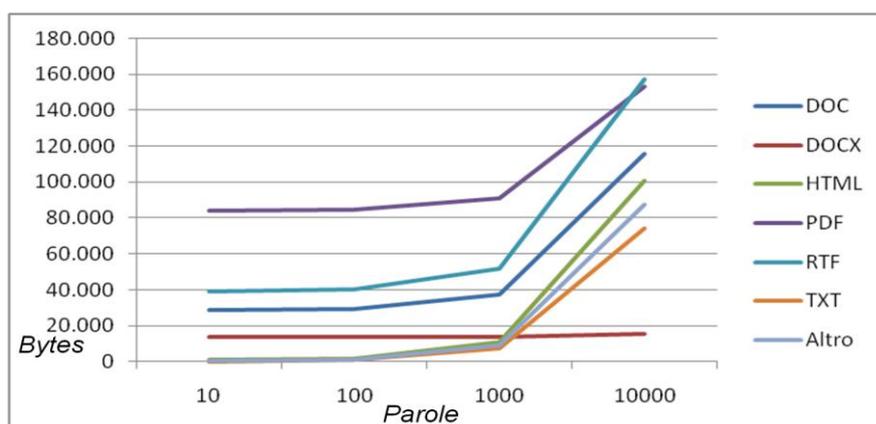


Figura 25. Funzioni per la stima delle parole in funzione della dimensione in Bytes

Da questo sono state ricavate delle funzioni di stima che consentono di determinare con una buona approssimazione il numero di parole presenti in un file e, quindi, l'effettiva durata intesa come il tempo necessario alla lettura, direttamente a partire dalla dimensione in Bytes.

Delle analisi analoghe sono state effettuate anche su altri formati, ovvero su altre tipologie di risorse quali ad esempio le immagini funzionali alla comprensione del testo (corrispondenti a tipologie di risorse quali *FIGURA*, *GRAFICO*, *DIAGRAMMA*), per cui,

analogamente a quanto individuato da ricerche della *Brandon Hall* (cfr. [22]), è possibile stimare il tempo dedicato dall'utente in fase di fruizione. Esso, mediamente, è banalmente correlato alla qualità dell'immagine e, dunque, a sua volta, normalizzando eventuali fattori di compressione, alla dimensione in Bytes.

Considerazioni più empiriche sono state fatte su oggetti ad elevato grado di interazione (risorse di tipologia *SIMULAZIONE*, *ESPERIMENTO*, *ESERCIZIO*) per i quali è stato ricondotto il parametro tempo alla dimensione in Bytes sulla base di analisi statistiche.

Per tutti gli altri file quali audio, video, eventi live sincroni (ad esempio come le lezioni in videoconferenza) appartenenti alla Tipologia di risorsa didattica *LEZIONE* la stima della durata è esattamente pari alla durata del file.

E' bene sottolineare che la durata non è uguale al tempo necessario per la comprensione e per poter determinare questo serve completare la descrizione del LO determinandone Densità semantica e Difficoltà. Ad esempio se un utente legge una pagina di testo, può impiegare un determinato tempo, ma se la densità semantica o la difficoltà sono elevate, sarà necessario più tempo per comprenderne il significato.

Il processo finora descritto parte dalla caratterizzazione semantica effettuata, ovvero un LO viene collegato ad un insieme di concetti (uno o più). In chiave Teoria dell'Informazione (cfr. [54]) è possibile ricondurre i concetti alla quantità di informazioni significative presenti nel LO, per cui è ragionevole che, a parità di concetti, più è alta la dimensione, minore sarà la densità semantica; più è bassa la dimensione, maggiore sarà la densità semantica.

Questo consente quindi di calcolare la densità semantica ds_c come il rapporto tra il numero di concetti n_c e la dimensione complessiva della risorsa (la sommatoria su tutti i file f delle rispettive dimensioni d_f), vista cioè come una quantità di informazioni impiegata per esprimere i concetti trattati:

$$ds_c = \frac{n_c}{\sum_f d_f}$$

Riprendendo infine quanto riportato nei modelli di Motelet et al. (cfr. [42]), Ronsivalle et al. (cfr. [50]) e Anderson et. Al (cfr. [9]), la

difficoltà può essere determinata correlando i concetti trattati alla durata del LO. A parità di concetti, minore è la durata, maggiore sarà la difficoltà e viceversa. Pertanto la difficoltà d_c può essere calcolata come un rapporto tra il numero di concetti n_c e la durata complessiva del LO (la sommatoria su tutti i file f delle rispettive durate t_f):

$$d_c = \frac{n_c}{\sum_f t_f}$$

I campi 9 - *Difficoltà* e 10 - *Densità semantica* dei Metadata, sono campi a cardinalità 0-1 che fanno riferimento ad un vocabolario di termini limitato e dato da:

Densità semantica \in {Molto basso, Basso, Medio, Alto, Molto Alto}
Difficoltà \in {Molto facile, Facile, Medio, Difficile, Molto difficile}

Queste considerazioni aprono un altro dibattito del tipo: Qual è la dimensione “giusta” per un concetto? Qual è il tempo di fruizione necessario per un concetto? In altri termini, rispettando la dicitura del vocabolario dei Metadata, *Densità semantica Media*, a cosa corrisponde? Con *Difficoltà Media* cosa si intende?

Sono state condotte delle analisi statistiche finalizzate a fornire risposta ai quesiti sollevati e determinare dei *range* in funzione dei quali attribuire dei valori congrui a *Difficoltà* e *Densità semantica*. È stato individuato come riferimento che una risorsa relativa ad un concetto ha una durata media di 3 minuti e dimensione media pari a 150 Kbyte e sono stati determinati anche gli altri intervalli ragionevoli. Il risultato è sintetizzato nelle seguenti tabelle:

Durata in minuti per concetto		Difficoltà
Min	Max	
5	>	Molto facile
3,5	5	Facile
2,5	3,5	Media
1	2,5	Difficile
<	1	Molto difficile

Tabella 9. Range di Durata per la determinazione della Difficoltà

KBytes per concetto		Densità semantica
Min	Max	
1500	>	Molto bassa
200	1500	Bassa
100	200	Media
50	100	Alta
<	50	Molto alta

Tabella 10. Range di Dimensione in KBytes per la determinazione della Densità semantica

Resta infine da valutare il campo *11 - Tempo impiegato su un Learning Object* ovvero il tempo per la comprensione del LO. Ovviamente più un contenuto è complesso, più tempo sarà necessario per comprenderlo. Analogamente, maggiore è la densità semantica che lo caratterizza, maggiore sarà il tempo da dedicare alla comprensione. Alla luce di queste considerazioni sono stati determinati dei fattori moltiplicativi ragionevoli che sono riportati nella seguente tabella:

Difficoltà	Fattore Difficoltà
Molto bassa	1,2
Bassa	1,3
Media	1,5
Alta	1,8
Molto alta	2

Tabella 11. Fattore moltiplicativo per il calcolo del Tempo di comprensione di un LO in funzione della Difficoltà

Interviene, inoltre, un altro fattore che dipende dalla densità semantica secondo la tabella seguente:

Densità semantica	Fattore Densità Semantica
Molto bassa	1,2
Bassa	1,3
Media	1,5
Alta	1,8
Molto alta	2

Tabella 12. Fattore moltiplicativo per il calcolo del Tempo di comprensione di un LO in funzione della Densità semantica

Il Tempo impiegato su un Learning Object (tlo_c) può essere quindi calcolato attraverso un prodotto tra la durata (d_c), il fattore difficoltà (fd_c) e il fattore densità semantica (fds_c):

$$tlo_c = d_c \cdot fd_c \cdot fds_c$$

3.7.5 Sperimentazione

E' stato preso in esame un archivio di oltre 2000 LO per l'insegnamento della Matematica e dell'Informatica in ambito universitario e un archivio di oltre 500 LO per l'insegnamento di tematiche specifiche del settore della grande distribuzione.

Contenuto didattico	Numero di contenuti	Tipologia risorsa didattica	Commento	Tipo di interazione	Livello di interazione
File MS-Excel	27	TABELLA	Qualsiasi oggetto per cui sia necessaria l'applicazione Excel, viene caratterizzato come TABELLA. Il fine è sicuramente sia visualizzare qualcosa, sia consentire un po' di interazione quindi il tipo è MISTO e il livello è MEDIO	MISTO	MEDIO
File MS-Word	72	TESTO	Qualsiasi oggetto per cui sia necessaria l'applicazione msWord viene caratterizzato come TESTO da leggere, quindi ESPOSITIVO e con interazione minima (MOLTO BASSO)	ESPOSITIVO	MOLTO BASSO
File MS-PowerPoint	259	SLIDE	Qualsiasi oggetto per cui sia necessaria l'applicazione msPowerPoint viene caratterizzato come SLIDE di tipo ESPOSITIVO. Il livello di interazione è leggermente maggiore del documento da leggere, poiché minimo ci sono dei pulsanti AVANTI e INDIETRO	ESPOSITIVO	BASSO
File PDF	563	TESTO	Qualsiasi oggetto per cui sia necessaria l'applicazione per visualizzare i pdf viene caratterizzato come TESTO da leggere quindi di tipo ESPOSITIVO e livello di interazione MOLTO BASSO	ESPOSITIVO	MOLTO BASSO
File OpenOffice	11	TESTO	Qualsiasi oggetto per cui sia necessaria l'applicazione Wordperfect viene caratterizzato come TESTO, ESPOSITIVO e con livello di interazione minimo (MOLTO BASSO)	ESPOSITIVO	MOLTO BASSO
Applicazioni FLASH	78	SIMULAZIONE	Qualsiasi altro oggetto APPLICATION viene caratterizzato con il tipo SIMULAZIONE e come tale prevede interazione ATTIVO-MOLTO ALTO	ATTIVO	MOLTO ALTO
File di Testo	121	TESTO	Qualsiasi oggetto TEXT viene caratterizzato con il tipo TESTO, ESPOSITIVO e con livello di interazione minimo (MOLTO BASSO)	ESPOSITIVO	MOLTO BASSO
File audio	234	LEZIONE	Qualsiasi oggetto AUDIO viene caratterizzato con il tipo LEZIONE da ascoltare (ESPOSITIVO) e con livello di interazione minimo (MOLTO BASSO)	ESPOSITIVO	MOLTO BASSO
File video	86	LEZIONE	Qualsiasi oggetto VIDEO viene caratterizzato con il tipo LEZIONE da vedere (ESPOSITIVO) e con livello di interazione BASSO (Leggermente maggiore dell'audio)	ESPOSITIVO	BASSO
Immagini/Foto	658	FIGURA	Qualsiasi oggetto IMAGE viene caratterizzato con il tipo FIGURA, ESPOSITIVO e con livello di interazione minimo (MOLTO BASSO)	ESPOSITIVO	MOLTO BASSO
Oggetti 3D/VRML	5	SIMULAZIONE	Qualsiasi oggetto di realtà 3D viene indicizzato come SIMULAZIONE e come tale prevede interazione ATTIVO-MOLTO ALTO	ATTIVO	MOLTO ALTO
Messaggi	147	TESTO	Qualsiasi oggetto Messaggio viene indicizzato come TESTO, ESPOSITIVO e con livello di interazione minimo (MOLTO BASSO)	ESPOSITIVO	MOLTO BASSO
Videoconferenze	16	LEZIONE	Qualsiasi oggetto Conferenza viene indicizzato come LEZIONE. Il livello di interazione previsto è medio (strumenti disponibili). Il tipo di interazione MISTO	MISTO	MEDIO
HTML	68	INDICE	Qualsiasi oggetto WWW viene caratterizzato come INDICE è espositivo ma consente un minimo di interazione per accedere ad altri contenuti	ESPOSITIVO	BASSO
HTML	162	TESTO	Qualsiasi ipertesto fruibile via Web che si presenta come HTML con altri elementi collegati e che non sia semplicemente un indice di accesso ad altri contenuti	ESPOSITIVO	BASSO
Totale	2507				

Tabella 13. I contenuti adottati ai fini della verifica sperimentale

I contenuti di tali archivi sono stati fatti fruire nell'ambito di corsi semplici o personalizzati attraverso la piattaforma IWT ad un numero complessivo di oltre 2300 utenti coinvolti in periodi differenti.

I tempi di fruizione sono stati tracciati e la percezione sulla Difficoltà e sulla Densità semantica è stata rilevata attraverso dei sondaggi realizzati elaborando i principi di progettazione didattica di Ronsivalle (cfr. [50]). I sondaggi sono stati erogati direttamente on-line per raccogliere feedback sui contenuti collegati ai singoli concetti:

Dizionario/Concetto	Learning Object	Indicare il livello di Difficoltà percepito {<i>Molto facile, Facile, Medio, Difficile, Molto difficile</i>}	Indicare la Densità semantica riscontrata {<i>Molto bassa, Bassa, Media, Alta, Molto Alta</i>}
Fondamenti di Informatica/ Costrutto condizionale IF	Slide su costruito IF		
	Testo su costruito IF		
	Spiegazione video-lavagna IF		
Fondamenti di Informatica/ Ciclo FOR	Slide su ciclo FOR		
	Testo su ciclo FOR		
	Esercizio su ciclo FOR		
Etc.			

Tabella 14. Esempi di schede per la raccolta feedback sulla percezione degli utenti in merito a Difficoltà e Densità semantica

Considerando le seguenti interpretazioni:

Difficoltà	Interpretazione
Molto facile	Conoscenza sui concetti trattati: riuscire a ricordare quali concetti sono stati presentati
Facile	Comprensione sui concetti trattati: avere chiari i concetti e sentire di averli appresi
Medio	Applicazione dei concetti trattati: sentirsi nelle condizioni di applicare i concetti trattati a problemi pratici
Difficile	Analisi sui concetti trattati: essere nelle condizioni di poter analizzare le informazioni ricevute con spirito critico
Molto difficile	Sintesi sui concetti trattati: essere nelle condizioni di astrarre dalle informazioni e fare sintesi

Tabella 15. Interpretazione dei vari livelli di Difficoltà

Densità semantica	Interpretazione
Molto bassa	Quantità di informazioni NON sufficiente a giustificare l'unità di erogazione
Bassa	Quantità di informazioni minima per giustificare l'unità di erogazione
Media	Quantità di informazioni ottimale associata all'unità di erogazione
Alta	Quantità di informazioni massima per giustificare l'unità di erogazione
Molto alta	Quantità di informazioni sovradimensionata rispetto all'unità di erogazione

Tabella 16. Interpretazione dei vari livelli di Densità semantica

Questa raccolta di feedback a posteriori ha consentito di verificare e raffinare i modelli di stima definiti per poter creare in automatico i metadata.

Capitolo 4

Conclusioni e sviluppi futuri

Nel corso degli anni di lavoro per la preparazione di questa tesi di dottorato sono stati studiati modelli e approcci allo stato dell'arte per individuare soluzioni metodologiche che possano apportare miglioramenti significativi alla piattaforma di e-learning IWT.

Durante il lavoro l'attenzione si è concentrata in una prima fase a quali aspetti potessero essere migliorati nell'algoritmo di LIA, il Learner Intelligent Advisor di IWT che ha il compito di preparare i percorsi formativi personalizzati. Esso, sulla base di un obiettivo didattico definito su un'ontologia, di un archivio di contenuti caratterizzati da metadata e collegati ai nodi dell'ontologia, di un insieme di utenti caratterizzati da un profilo di conoscenze e di preferenze didattiche, è capace di generare per ciascun utente un corso diverso. Questo aveva, per contro, un peso di elaborazione non indifferente. Pertanto, sono stati trattati aspetti di efficienza computazionale degli algoritmi di generazione dei percorsi personalizzati che, di fatto, sono stati realizzati ex-novo seguendo tecniche risolutive differenti. La complessità computazionale dei nuovi algoritmi è decisamente inferiore a quella degli algoritmi originari (si è scesi da una complessità polinomiale del quarto ordine ad una quadratica).

La piattaforma IWT con i nuovi algoritmi è stata poi sperimentata in vari contesti facendo confronti tra l'approccio tradizionale dell'e-learning statico e l'approccio IWT dell'e-learning personalizzato.

La sperimentazione ha avuto un duplice intento: da un lato, verificare l'approccio innovativo di IWT nell'erogazione dei percorsi personalizzati nella formazione sia dentro che fuori dai contesti universitari, dall'altro, comprendere con maggiore contezza quali siano le metodologie adottate in IWT che possano essere ulteriormente migliorate. In particolare, i casi più complessi derivano

da una presenza di un elevato numero di utenti e di un elevato numero di contenuti.

In questi casi, infatti, sono particolarmente onerose tutte le operazioni che consentono di verificare se i contenuti disponibili siano adeguati a rispondere opportunamente alle esigenze degli utenti, che consentano di aggiornare correttamente i profili degli utenti durante la fruizione dei percorsi formativi e, soprattutto, che consentano di modellare in modo appropriato la base di conoscenza necessaria ad IWT per ottimizzare i percorsi personalizzati da erogare agli utenti.

E' stato proposto un approccio semplice, computazionalmente efficiente ed alquanto originale in ambito e-learning per verificare se una libreria digitale di contenuti possa essere impiegata propriamente in un contesto didattico. La complessità computazionale della soluzione realizzata è, infatti, paragonabile ad algoritmi di geometria computazionale lineare con le dimensioni dello spazio.

E' stato analizzato l'approccio adottato da IWT nell'aggiornamento dei profili degli utenti durante la fruizione di un corso personalizzato ed è stato proposto un miglioramento all'algoritmo di valutazione che risulta essere didatticamente più efficace.

Per ridurre i costi di progettazione della base di conoscenza è stato proposto un approccio che consente di estrarre in automatico l'ontologia da pacchetti SCORM e suddividerli in tante parti che possono poi essere usate come risorse "a grana più fine" per consentire ad IWT, anche in presenza di corsi già pronti e confezionati, di creare ed erogare a partire da questi dei percorsi formativi personalizzati e, quindi, di rispondere anche a differenziate esigenze di apprendimento di una platea variegata di utenti.

E' stato, infine, proposto un approccio semplice, efficiente ed anch'esso decisamente originale rispetto allo stato dell'arte in ambito e-learning, per catalogare contenuti di una libreria digitale affinché possa essere impiegata propriamente in un contesto didattico.

E' stato definito un modello di estrazione dei metadata direttamente dai learning object evitando "costose" operazioni di analisi del contenuto, ma elaborando caratteristiche oggettive ed immediatamente reperibili dai file che costituiscono i learning object.

Questo approccio dovrà essere sperimentato in altri contesti al di fuori di quello della didattica a distanza ed eventualmente raffinato al

fine proiettarlo nell'ottica moderna della pubblicazione delle informazioni sul web secondo la logica dei Linked Data. Logica a cui stanno dando attenzione enti come IEEE, BBC, Governo USA, etc., che, ovviamente, hanno grandi esigenze di catalogazione automatica di documenti.

Riferimenti

- [1] Abdullah N. A., Bailey C., and Davis H. *Synthetic hypertext and hyperfiction: Augmenting SCORM manifests with adaptive links*. Proceedings of the 15th ACM Conference on Hypertext and Hypermedia, Santa Cruz, EUA, 2004, pp. 183-184.
- [2] Acanfora G., Gaeta M., Loia V., Ritrovato P. and Salerno S. *Optimizing Learning Path Selection through Memetic Algorithms*. Proceedings of IEEE World Congress on Computational Intelligence, 1-6 June 2008 Hong Kong.
- [3] Adorni G., Brondo D. and Vivanet G. (2009), *A formal instructional model based on Concept Maps*, Journal of e-Learning and Knowledge Society, 5(3): 33-42.
- [4] Adorni G., Coccoli M. and Vivanet G. (2007), *Topic Maps e XTM per l'e-learning*, Journal of e-Learning and Knowledge Society, 3: 51-60.
- [5] Adorni G., Di Manzo M. and Frisiani A.L. (1981), *Evaluation of a Formal Approach to the Structuring of Subject Matter*, Journal of Computer-Based Instruction, 8(2): 35-42.
- [6] Albano G., Gaeta M. and Ritrovato P. *IWT: an innovative solution for AGS e-Learning model*. International Journal of Knowledge and Learning, vol. 3, no. 2/3 p. 209-224, 2007.
- [7] Albano G., Gaeta M., Salerno S. *E-learning: a model and process proposal*. International Journal of Knowledge and Learning, Inderscience Publisher, vol. 2, no. 1/2, pp. 73-88, 2006.
- [8] Anderson T. and Whitelock D. (2004), *The educational semantic web: visioning and practicing the future of education*, Journal of Interactive Media in Education, 7:1-15.
- [9] Anderson, Marzano and Romiszowski *Fun with learning taxonomies of Bloom* <http://gramconsulting.com/2009/02/fun-with-learning-taxonomies/>
- [10] Baker M. (2000), *The roles of models in Artificial Intelligence and Education research: a prospective view*, International Journal of Artificial Intelligence in Education, 11: 122-143.
- [11] Baldacci M. (2005). *Personalizzazione o Individualizzazione?*, Trento: Centro Studi Erickson.
- [12] Berners Lee, T., Hendler, J. and Lassila, O. (2001), *The Semantic Web*, Scientific American.

-
- [13] Berners-Lee T. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco, 1999.
- [14] Bittencourt I. I., Isotani S., Costa E., Mizoguchi R. *Research directions on Semantic Web and education*, Scientia Interdisciplinary Studies in Computer Science 19(1): 60-67, January/June 2008 © 2008 by Unisinos
- [15] Bolettieri P., Falchi F., Gennaro C., and Rabitti F. *Automatic metadata extraction and indexing for reusing e-learning multimedia objects*. In Workshop on Multimedia information Retrieval on the Many Faces of Multimedia Semantics (Augsburg, Bavaria, Germany, September 28 - 28, 2007). MS '07. ACM, New York, NY, 21-28. DOI=
<http://doi.acm.org/10.1145/1290067.1290072>
- [16] Brase J. and Nejdil W., *Ontologies and Metadata for eLearning*, Handbook on Ontologies, 2004, pp. 555-574.
- [17] Capuano N., Gaeta A., Orciuoli F., Paolozzi S. *Exploiting Tagging in Ontology-based e-Learning* – Proceedings of the 1st International Workshop on Ontology for e-Technologies (OET 2009) in conjunction with ICEIS 2009 - Milan, Italy, May 6-10, 2009.
- [18] Capuano N., Gaeta M., Micarelli A. and Sangineto E. *An Integrated Architecture for Automatic Course Generation*. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2002), September 9-12, 2002, Kazan, Russia. V. Petrushin, P. Kommers, Kinshuk, I. Galeev eds. p. 322-326, IEEE Computer Society, 2002.
- [19] Capuano N., Gaeta M., Miranda S., Orciuoli F. and Ritrovato P. *LIA: an Intelligent Advisor for e-Learning*. M. Lytras, J. Carroll, E. Damiani, R. Tennyson (Eds.), Emerging Technologies and Information Systems for the Knowledge Society - Proceedings of the World Summit on the Knowledge Society (WSKS 2008), September 24-26, 2008, Athens, Greece, Lecture Notes in Computer Science Series, vol. 5288, pp. 187-196, Springer-Verlag, 2008 – 2nd Best Paper Award
- [20] Capuano N., Gaeta M., Orciuoli F. and Ritrovato P. (2009), *On-Demand Construction of Personalized Learning Experiences Using Semantic Web and Web 2.0 Techniques*, 9th. IEEE

- International Conference on Advanced Learning Technologies (ICALT 2009), IEEE Computer Society.
- [21] Capuano N., Miranda S. and Orciuoli F. *IWT: A Semantic Web-based Educational System*. G. Adorni, M. Coccoli (Eds.) Proceedings of the IV Workshop of the AI*IA Working Group on Artificial Intelligence & e-Learning held in conjunction with the XI International Conference of the Italian Association for Artificial Intelligence (AI*IA 2009), December 12, 2009, Reggio Emilia, Italy, pp. 11-16, AI*IA, 2009
- [22] Clarey J.. *How do you estimate e-learning seat time?* October 17, 2008.
<http://www.brandon-hall.com/workplacelearningtoday/?p=767>
- [23] Crispiani P. (2004), *Didattica cognitivista*, Roma: Armando Editore.
- [24] Dobрева M., Kim Y. and Ross S.. *Automated Metadata Extraction*. <http://www.dcc.ac.uk/resources/curation-reference-manual/chapters-production/automated-metadata-extraction>
- [25] Forrest A.R. *Computational geometry*. Proc. Royal Society London, 321, series 4, 187-195 (1971)
- [26] Gaeta M., Orciuoli F. and Ritrovato P. (2009), *Advanced ontology management system for personalised e-Learning*, Knowledge-Based Systems, 22(4): 292-301.
- [27] Gagnè R.M. e Briggs L.J. (1990), *Fondamenti di Progettazione Didattica*, SEI, Milano.
- [28] Gardner, H. (1983), *Frames of mind: The theory of multiple intelligences*. New York: Basic books.
- [29] Gomes S. R., Gadelha B. F., Mendonca A. P., Amortti M. S. M. *Objetos de Aprendizagem Funcionais e as Limitacoes dos Metadados Atuais* in Anais do XVI Simpósio Brasileiro de Informatica na Educacdo - SBIE 2005, Juiz de Fora, Brasil, pp. 211-221
- [30] Greenberg, J. *Metadata Extraction and Harvesting: A Comparison of Two Automatic Metadata Generation Applications*. Journal of Internet Cataloguing: The International Quarterly of Digital Organization, Classification, and Access, 6(4), 58-82, 2004
- [31] Gruber T. *Towards principles for the design of ontologies used for knowledge sharing*. Int. J. of Human and Computer Studies, 43:907-928, 1994.

-
- [32] Hansen E, and Walster W. G. *Global Optimization using Interval Analysis*, Second Edition, Revised and Expanded, Marcel Dekker, New York, 2004, ISBN 0-8247-4059-9.
- [33] Hepp M., Bachlechner D. and Siorpaes K. *OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis*
- [34] Jaulin L., Kieffer M., Didrit O., and Walter É. *Applied Interval Analysis: With examples in parameter estimation robust control and robotics*. Springer, London, 2001, ISBN 1-85233-219-0.
- [35] Maedche A. and Staab S. *Ontology Learning for the Semantic Web*. IEEE Intelligent Systems, vol. 16, no. 2, Mar./Apr. 2001, pp. 7279.
- [36] Maedche E. and Staab S. *Mining non-taxonomic conceptual relations from text*. R. Dieng and O Corby. EKAW00 European Knowledge Acquisition Workshop. October 2-6, 2000, Juan-les-Pins. LNAI, Springer.
- [37] Manning Christopher D. and Schtze H. *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999.
- [38] McKenzie W. (2006), *Intelligenze multiple e tecnologie per la didattica*, Trento, Erickson.
- [39] Merrill M.D. (1999), *Instructional transaction theory (ITT): instructional design based of knowledge objects*, in C.M. Reigeluth (ed.), *Instructional design theories and models: a new paradigm of instructional theory*, Lawrence Erlbaum Associates, Hillsdale, NJ, pp.397-424.
- [40] Mizoguchi R. e Bordeau J. (2000), *Using Ontological Engineering to Overcome Common AI-ED Problems*, *International Journal of Artificial Intelligence in Education*, 11(2).
- [41] Moore R. E. *Interval Analysis*. Prentice-Hall, Englewood Cliff, NJ, 1966, ISBN 0-13-476853-1.
- [42] Motelet O., Baloian N.A., Pino J.A. *Hybrid System for Generating Learning Object Metadata*. In *JOURNAL OF COMPUTERS*, VOL. 2, NO. 3, MAY 2007
- [43] Motelet O., Baloian N.A., Pino J.A. *Taking advantage of metadata semantics: the case of learning-object-based lesson graphs*. *Knowl Inf Syst* (2009) 20:323–348. Springer Ed.

- [44] Navigli R., Velardi P, and Gangemi A. *Ontology learning and its application to automated terminology translation*. IEEE Intelligent Systems, 18(1):22-31, 2003.
- [45] Nejdil W., Wolf B., Staab S. and Tane J., *EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network*, Semantic Web Workshop, 2002.
- [46] O' Reilly T. *What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software*, Available at <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- [47] Ohlsson S. (1987) *Some principles of intelligent tutoring*, in: Lawler R.W., Masoud Yazdani M. (eds.), *Artificial Intelligence and Education: Learning Environments and Tutorial Systems v. 1*, Intellect Books.
- [48] Preparata F. P. and Shamos M.I. (1985). *Computational Geometry - An Introduction*. Springer-Verlag. 1st edition: ISBN 0-387-96131-3; 2nd printing, corrected and expanded, 1988: ISBN 3-540-96131-3.
- [49] Rawlings, A., van Rosmalen, P., Koper, R., Rodriguez-Artacho, M. and Lefrere, P. (2002), *Survey of Educational Modelling Languages (EMLs)*, CEN/ISSS WS Learning Technologies Workshop.
- [50] Ronsivalle G.B., Carta S., Metus V. (a cura di), *L'arte della progettazione didattica. Dall'analisi dei contenuti alla valutazione dell'efficacia* FrancoAngeli, Milano 2009, pp. 72-89
- [51] Roy D., Sarkar S., Ghose S. *Automatic Extraction of Pedagogic Metadata from Learning Content*. International Journal of Artificial Intelligence in Education, Volume 18 Issue 2, April 2008, Pages: 97-118
- [52] Sagi, E., Kaufmann, S., and Clark, B. *Semantic density analysis: comparing word meaning across time and phonetic space*. Proc. of the Workshop on Geometrical Models of Natural Language Semantics (Athens, Greece, March 31 - 31, 2009). M. Pennacchiotti, Ed. ANLP/NAACL Workshops. Association for Computational Linguistics, Morristown, NJ, 104-111.
- [53] Shadbolt N., Berners-Lee T., Hall W. (2006), *The Semantic Web Revisited*, IEEE Intelligent Systems, 21(3): 96-101.

-
- [54] Shannon C. E. *A Mathematical Theory of Communication*. Bell System Technical Journal, July and October 1948. Reprinted in Claude Elwood Shannon: Collected Papers. New York: IEEE Press, 1993
- [55] Shmoys D., Tardos E. and Aardal K.. *Approximation Algorithms for Facility Location Problems*. Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, Texas, United States, p. 265-274, 1997.
- [56] Silva, L. et al (2006) *Using Conceptual Lattices to Represent Fine Granular Learning Objects through SCORM Meta-Objects*. In: Williams, S. (Ed): The Electronic Journal of e-Learning, Volume 4 Issue 2, pp.141-148, available online at <http://www.ejel.org>
- [57] Simoes D., Luis R., and Horta N. *Enhancing the SCORM Metadata Model*. Proceedings of the 13th international World WiConference. New York, EUA, 2004, pp. 238-239.
- [58] Staelin C., Elad M., Greig D., Shmueli O., Vans M., *Biblio: automatic meta-data extraction*. IJDAR (2007) 10:113–126. Springer Ed.
- [59] Stojanovic L., Staab S., and Studer R. (2001) *eLearning based on the Semantic Web*, World Conference on the WWW and Internet, Orlando, Florida, USA, December 2001
- [60] Sure Y., Erdmann M., Angele J., Staab S., Studer R., Wenke D. *OntoEdit: Collaborative Ontology Development for the Semantic Web*. Proceedings of the 1st International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Italy.
- [61] Tane J., Schmitz C., Stumme G., Staab S., Studer R. *The courseware watchdog: an ontology-based tool for finding and organizing learning material*. Beitrge der Fachtagung an der Universitt, Kassel University Press, 2003, pp. 93-104.
- [62] Velardi P., Navigli R., Cucchiarelli A., and Neri F. *Evaluation of OntoLearn, a methodology for automatic population of domain ontologies*. Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press, 2006.
- [63] Zhuge H., Y. Li *KGTutor: a knowledge grid based intelligent tutoring system*, APWeb (2004) 473-478.
- [64] MOMA (2009), IWT – Intelligent Web Teacher, <http://www.momanet.it>.

- [65] http://en.wikipedia.org/wiki/Interval_arithmetic
- [66] <http://www.didatticaadistanza.com/>
- [67] <http://www.imsproject.org/learningdesign/>
- [68] <http://www.lamsfoundation.org/>
- [69] <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- [70] IEEE LTSC Personal Private Information Standard (PAPI) - <http://edutool.com/papi/> (verificato il 26/03/2010).
- [71] IMS Learner Information Package (LIP) - <http://www.imsproject.org/profiles/> (verificato il 26/03/2010).
- [72] <http://wordnet.princeton.edu/>
- [73] <http://www.w3.org/TR/2002/WD-owl-ref-20021112/>
- [74] <http://protege.stanford.edu/>
- [75] <http://poi.apache.org/>
- [76] <http://www.jgraph.com/>
- [77] <http://www.graphviz.org/>
- [78] General Architecture for Text Engineering - <http://gate.ac.uk/>
- [79] <http://snowball.tartarus.org/>
- [80] Dublin Core. <http://dublincore.org/>
- [81] IMS global learning consortium. <http://www.imsglobal.org>
- [82] Tavola con i tipi MIME.
<http://www.asciitable.it/mimetypes.asp>
- [83] IEEE LOM specification. <http://ltsc.ieee.org>
- [84] Metadata Extractor by National Library of New Zealand,
<http://meta-extractor.sourceforge.net/>
- [85] Digital Preservation Europe.
<http://www.digitalpreservationeurope.eu/>
- [86] Humanities Advanced Technology and Information Institute (HATII), University of Glasgow
<http://www.digitalpreservationeurope.eu/publications/briefs/semantic%20metatada.pdf>
- [87] Metadata Extraction,
http://wiki.alfresco.com/wiki/Metadata_Extraction
- [88] Word count to page,
http://www.writersservices.com/wps/p_word_count.htm
- [89] Apache Tika. <http://tika.apache.org/>