

Università degli Studi di Torino

Facoltà di Scienze Matematiche, Fisiche e Naturali

Dipartimento di Informatica

Tesi di laurea

**Strumenti per la costruzione di ontologie multilingua in
ambito giuridico**

Relatore: Prof. Guido Boella

Candidato: Marco Martin

Anno Accademico 2007/2008

Sommario

Capitolo 1 -Introduzione.....	1
1.1Scopo della tesi.....	1
1.2Ontologie.....	1
Capitolo 2 -Sistema esistente.....	3
2.1scopo del sistema.....	3
2.1.1Schema ontologia utilizzata.....	4
2.2Casi d'uso e tipologia di utenti.....	6
2.3Progettazione.....	8
2.3.1Architettura.....	8
2.3.2Architettura object oriented.....	9
2.3.3Package.....	14
2.3.4Progettazione del database.....	14
2.3.4.1Controlli aggiuntivi sull'integrità dei dati.....	17
2.3.5Design di dettaglio.....	17
2.3.5.1Flusso di esecuzione.....	17
2.3.5.2Autenticazione utenti.....	18
2.3.5.3Azioni valide.....	19
2.4Aspetti implementativi.....	22
2.4.1PHP.....	22
2.4.2Smarty template engine.....	22
2.4.3PostgreSQL.....	23
Capitolo 3 -Obiettivi giuridici.....	24
3.1Problema di interpretazione del diritto.....	24
3.2Cambiamento normativo.....	25
3.3Limitazioni del sistema delle citazioni.....	26
3.3.1Costruzione terminologie giuridiche contestualizzate.....	26
3.4Destinatari del diritto comparato.....	26
Capitolo 4 -Estensione del sistema esistente.....	28
4.1Diversi livelli di astrazione dei concetti.....	28
4.1.1Raggruppamento mediante relazioni ontologiche.....	28
4.1.1.1Casi speciali per la presentazione grafica.....	28
4.2Tecnologie NLP.....	29

4.2.1	Entità dei dati e bisogno di un indice.....	29
4.2.1.1	Indice invertito.....	30
4.2.1.2	Sistemi esistenti.....	30
4.2.1.3	Implementazione attraverso un database relazionale.....	32
4.2.2	Ricerca attraverso uno stemmer.....	34
4.2.2.1	Sistemi per lo stemming.....	34
	Linguaggio Snowball.....	38
4.2.2.2	Stemming e indice.....	40
4.2.2.3	Function words.....	40
4.2.3	Document similarity.....	42
4.2.3.1	Peso dei termini all'interno di documenti.....	42
4.2.3.2	Cosine document similarity.....	45
4.3	Gestione dell'evoluzione temporale in un'ontologia.....	45
4.3.1	Similitudini con un data warehouse.....	46
4.3.2	Evoluzione mediante relazioni ontologiche.....	48
4.4	Modifiche nell'interfaccia.....	50
4.4.1	Traduzione interfaccia.....	50
4.4.2	Comportamento della funzione di ricerca.....	50
4.4.2.1	Ricerca nei gruppi.....	51
4.4.2.2	Ricerca di un meaning scaduti.....	51
4.5	Esportazione ontologia.....	51
4.5.1	Semantic Web.....	51
4.5.2	RDF.....	52
4.5.3	OWL.....	53
4.6	Specifica formale dei requisiti.....	56
4.6.1	INTRODUCTION.....	56
4.6.1.1	Purpose.....	56
4.6.1.2	Definitions, Acronyms, Abbreviations.....	56
4.6.1.3	Notational Conventions.....	57
4.6.2	OVERALL DESCRIPTION.....	57
4.6.2.1	Product perspective.....	57
4.6.2.2	Product functions overview.....	57
4.6.2.3	User Characteristics.....	58
4.6.3	SPECIFIC REQUIREMENTS.....	59
4.6.3.1	Inputs and Outputs.....	59

Inserimento Umeaning.....	59
Inserimento e cancellazione testi giuridici.....	60
Inserimento e cancellazione riferimenti.....	60
Raggruppamento di umeaning.....	61
Cambiamento normativo e sostituzione umeaning.....	61
Ricerca umeaning.....	61
Ricerca testi legali.....	62
4.6.3.2Functional Requirements.....	62
Inserimento Umeaning.....	62
Inserimento e cancellazione testi giuridici.....	63
Inserimento e cancellazione riferimenti.....	63
Raggruppamento di umeaning.....	64
Cambiamento normativo e sostituzione umeaning.....	64
Ricerca umeaning.....	65
Ricerca testi legali.....	65
Esportazione Ontologia.....	66
4.6.3.3External Interface Requirements.....	66
4.6.3.4Design Constraints.....	66
Capitolo 5 -Progettazione.....	68
5.1Architettura attuale delle classi.....	68
5.2Design di dettaglio delle modifiche necessarie.....	69
5.2.1Gestione livelli di astrazione.....	69
5.2.2Gestione dell'evoluzione temporale.....	70
5.2.3Archivio di testi giuridici.....	70
5.2.3.1Flusso di esecuzione.....	71
5.2.4Esportazione dell'ontologia.....	72
5.2.5Riepilogo.....	76
5.3Modifiche al database.....	77
5.3.1Gestione livelli di astrazione.....	78
5.3.2Gestione dell'evoluzione temporale.....	78
5.3.3Archivio di testi giuridici.....	78
5.3.4Modifica della struttura dell'umeaning.....	79
5.3.5Schema entità-relazione.....	80
Capitolo 6 -Implementazione.....	82

6.1	Raggruppamento.....	82
6.2	Evoluzione temporale.....	82
6.2.1	Modifica della ricerca.....	83
6.3	Archivio testi giuridici.....	84
6.3.1	Stemmer.....	86
6.3.2	Indice invertito.....	86
6.3.3	Document similarity.....	87
6.3.4	Riferimenti.....	88
6.3.4.1	Ricerca e inserimento.....	89
6.4	Traduzione dell'interfaccia.....	89
6.4.1	Funzionamento di Gettext.....	90
Capitolo 7 -Sviluppi futuri.....		92
Capitolo 8 -Manuale utente.....		93
8.1	Panoramica interfaccia.....	93
8.2	Autenticazione.....	94
8.3	Funzioni del dizionario.....	94
8.3.1	Pagina di ricerca.....	94
8.3.2	Visualizzazione umeaning.....	96
8.3.3	Visualizzazione di un termine.....	100
8.3.4	Gestione termini e umeaning.....	102
8.3.4.1	Inserimento di una nuova coppia termine/umeaning.....	102
8.3.4.2	Allegati.....	103
8.3.4.3	Sinonimi.....	103
8.3.4.4	Significati multipli di un termine.....	103
8.3.4.5	Traduzioni.....	103
8.3.4.6	Traduzioni di umeaning nazionali.....	103
8.3.4.7	Associazioni fra umeaning nazionali ed europei.....	104
8.3.5	Gestione Ontologia.....	105
8.3.5.1	I tipi di relazione.....	105
8.3.5.2	Le relazioni fra due umeaning.....	106
8.3.5.3	Relazioni ontologiche speciali.....	107
8.3.5.4	Gruppi.....	108
8.3.5.5	Cambiamento normativo.....	108
8.4	Riferimenti.....	109

8.4.1	Inserzione testi.....	109
8.4.2	Ricerca testi.....	110
8.4.3	Inserzione riferimenti.....	111
8.5	Statistiche.....	112
8.6	Amministrazione del sistema.....	113
8.6.1	Gestione degli account utente.....	113
8.6.2	Gestione delle lingue giuridiche.....	115
8.6.3	Consultazione e gestione del log degli eventi.....	115
8.6.4	Funzioni di gestione del database.....	117

Capitolo 1 - Introduzione

1.1 *Scopo della tesi*

Lo scopo di questa tesi è l'analisi delle caratteristiche, delle scelte implementative ed architetturali del sistema per la gestione di ontologie giuridiche multilingue denominato Syllabus, descritto in [1], analizzandone le limitazioni in base al feedback ricevuto dai suoi utenti durante alcuni anni di utilizzo.

Verranno quindi progettate ed implementate le modifiche necessarie al sistema esistente e soprattutto verrà progettato un nuovo modulo che permetterà l'indicizzazione e la ricerca full-text (mediante strumenti di natural language processing) in un archivio di testi giuridici appartenenti all'ordinamento europeo e agli ordinamenti nazionali di cinque stati membri.

1.2 *Ontologie*

Uno dei concetti teorici più importanti su cui è basato il Syllabus è l'*ontologia*: uno dei metodi ad oggi più efficaci per rappresentare formalmente un insieme di concetti.

Ontologia è un termine che deriva dalla filosofia: esso appare per la prima volta negli scritti di Parmenide (circa 504 a. C) e deriva quindi dal greco *eon logos*, cioè “discorso sull'ente”. L'ontologia si occupa infatti dello studio dell'essere, ovvero di ciò che è e delle sue categorie fondamentali.

In informatica, una ontologia, usata in particolar modo in studi sull'intelligenza artificiale e nella classificazione dei dati, è il tentativo di formulare una classificazione di concetti, quindi una gerarchizzazione, nell'ambito di un dominio. Un'ontologia, tuttavia, non deve limitarsi ad una gerarchia di concetti con un'unica relazione di appartenenza, spesso chiamata *isa*, cioè “è un”, ma deve comprendere anche altre relazioni semantiche per descrivere in maniera migliore il modo in cui gli oggetti sono correlati, ad esempio parte-di e contrario-di.

T.R. Gruber definisce l'ontologia come “una specificazione di una concettualizzazione” [2].

Egli afferma che una rappresentazione formale di un insieme di conoscenze è una *concettualizzazione*: essa è un insieme di oggetti, concetti e relazioni fra di essi che esistono in una particolare area d'interesse. Una concettualizzazione è una rappresentazione astratta e semplificata del particolare campo di conoscenza che si vuole rappresentare per un qualsiasi scopo.

Un'ontologia è dunque una specificazione di una concettualizzazione. Quando la conoscenza di un dominio è rappresentata in un qualche formalismo, l'insieme degli oggetti che possono essere rappresentati è chiamato l'universo del discorso. Questo insieme di oggetti e le relazioni fra loro, sono “riflessi” nel vocabolario in cui essi sono rappresentati.

Si hanno tre livelli di ontologie [3]:

- **Domain ontologies:** descrivono le categorie di una certa disciplina e sono legate ad uno specifico dominio di applicazione (domain), non sono usate direttamente per costruire sistemi, quanto per costruire archivi di informazioni (knowledge base). Vengono costruite per aiutare il lavoro cooperativo e costruire un accordo sui termini di un dominio e del loro significato, comprensibili da membri del team con diverso background culturale.
- **Core ontologies:** definiscono quali sono gli argomenti di un campo. Un campo è una disciplina, settore industriale o una qualsiasi area della società che unifica molti domini di applicazione. Una disciplina ha quindi bisogno di diverse domain ontologies. Le core ontologies descrivono quindi le categorie di una certa disciplina ed indicizzano delle librerie di domain ontologies.
- **Top ontologies:** il ruolo di una top ontology è esplicitare il ruolo dei *commitment ontologici* in una qualche domain ontology. I commitments ontologici sono accordi sull'utilizzo di un vocabolario condiviso usato in una maniera coerente e consistente. Non è necessario che gli agenti che condividono un vocabolario condividano l'intera knowledge base; un agente che implementa una particolare ontologia non è tenuto a rispondere a tutte le query che possono essere formulate nel vocabolario condiviso, ma soltanto a quelle che rientrano nella propria area di competenza [4]. Un esempio di top-ontology è DOLCE [5].
- **Lightweight ontologies:** in un'ontologia lightweight l'estensione del concetto di un nodo figlio è un sottoinsieme dell'estensione del concetto del nodo padre [6]. Un'ontologia lightweight è una tripla $\langle N, E, C \rangle$ dove N è un numero finito di nodi, E è un insieme di archi in modo che $\langle N, E \rangle$ sia un albero radicato e C un insieme finito di concetti espressi in un linguaggio formale, in modo che per ogni nodo $n_i \in N$ esiste uno ed un solo concetto $c_i \in C$.

Capitolo 2 - Sistema esistente

In questo capitolo verranno analizzate le caratteristiche del Syllabus e le scelte progettuali che stanno dietro alla sua architettura. Questo permetterà di comprendere meglio le motivazioni delle modifiche attuate e di poterne analizzare i dettagli implementativi.

2.1 *scopo del sistema*

Il Syllabus è stato progettato per essere uno strumento atto a facilitare l'identificazione della terminologia ed aiutare il giurista l'identificazione di similitudini e differenze, principi comuni e contrastanti tra la legislazione Europea a livello comunitario espressa attraverso le direttive e le legislazioni nazionali, caratterizzate da ordinamenti e tradizioni eterogenee.

Il sistema offre un dizionario che associa termini di uso comune nel linguaggio giuridico ai loro significati, espressi da passi di testi legali quali leggi o direttive in cui essi compaiono e che ne offrono un contesto.

Per meglio definire i concetti presenti vengono gestite nel sistema anche le relazioni esistenti tra diversi concetti attraverso un'ontologia: ad esempio una "clausola" è una parte di un contratto, quindi un concetto più specializzato.

A differenza di altri sistemi esistenti in precedenza come Eurovoc [7] le relazioni non sono fra i termini del dizionario, perché questi possono presentare ambiguità, ma sono fra i concetti che essi indicano, detti *umeaning*, cioè *unitary meaning*.

Il sistema gestisce quindi un insieme di dizionari indipendenti fra loro: il dizionario del livello europeo legato alle direttive comunitarie e un dizionario nazionale per ogni stato membro di cui si voglia analizzarne la legislazione.

Concettualmente il dizionario europeo è uno solo perché le direttive vengono redatte in inglese e tradotte nelle lingue degli stati membri mantenendo la medesima struttura del documento, avendo così una corrispondenza a livello di paragrafo.

Al momento gli ordinamenti nazionali gestiti dal Syllabus sono: Italia, Francia, Germania, Gran Bretagna e Spagna.

Solitamente le ontologie sono costruite attraverso un approccio top-down, partendo da concetti generali specializzandoli in concetti più dettagliati, tuttavia in questo caso si rischia di non tenere conto della differenza di interpretazione tra ordinamenti differenti.

Viceversa Syllabus utilizza un approccio di tipo bottom-up: per prima cosa i termini vengono raccolti in un database assieme ai suoi significati (le fonti legali in cui appaiono), si ha quindi all'inizio uno schema di tipo piatto. In un secondo tempo i concetti vengono

analizzati e rappresentati in ontologie che possono essere differenti fra loro nelle differenti tradizioni legali.

2.1.1 Schema ontologia utilizzata

L'ontologia modellata in Syllabus è la rappresentazione di un dizionario e delle relazioni esistenti fra le definizioni contenute in esso.

Come ogni dizionario è un'insieme di coppie chiave-valore in cui la parola definita è un *termine* giuridico e la sua definizione, chiamata *umeaning* è rappresentata da una descrizione e dati aggiuntivi.

Definiamo come **termine** l'espressione giuridica che vogliamo definire, il quale può essere tanto una singola parola, come ad esempio “contratto” quanto un sintagma più complessa, come ad esempio “chiaro e comprensibile”.

Il concetto corrispondente al termine, è detto **umeaning**. Definiamo *umeaning* una struttura composta da due campi testuali, rispettivamente la *descrizione*, compilata dal giurista per spiegare il significato del termine, e i *riferimenti* che riportano un passo di una o più normative che citano quel termine, ad esempio “...tali clausole devono essere sempre redatte in modo *chiaro e comprensibile*.” e i loro estremi ad esempio “dir 93/13: art. 4.2,5; dir 99/44: art. 6.2”.

Un ultimo campo identifica gli allegati, che sono informazioni aggiuntive opzionali, come ad esempio il testo completo della direttiva. O documenti di qualsiasi tipo che possono portare ad una maggior comprensione del termine.

Un termine può avere più di un significato, che può variare ad esempio dipendentemente dal contesto utilizzato, ad esempio il termine “diritto di recesso” nella legislazione Italiana può essere inteso come il diritto di una delle parti di rescindere unilateralmente un contratto (questo inerente alla legislazione sulla tutela dei consumatori) oppure il recesso del contratto in accordo fra le due parti, in questo caso è una particolare modalità di conclusione del contratto.

Allo stesso modo più termini possono avere più di un significato, questa è una situazione molto rara: nel database attuale sono presenti 549 termini e 251 *umeaning*, mentre gli unici due termini polisemici sono il tedesco “Mängelanzeige” e lo spagnolo “resolución”

Le relazioni fra due *umeaning*, possono essere di diversi tipi, se ad esempio “diritto di recesso” è una “clausola”, una “clausola” sarà una parte di “contratto”, ontologicamente lo possiamo esprimere come “diritto di recesso IS_A clausola” e “clausola PART_OF contratto”.

I dizionari presenti nel sistema sono uno per ogni lingua (allo stato attuale italiano, inglese, tedesco, francese e spagnolo) e rappresentano gli ordinamenti giuridici nazionali dei diversi stati. Inoltre è presente un ulteriore dizionario che descrive l'ordinamento giuridico europeo.

Questo è unico per tutte le lingue, perché le direttive europee vengono tradotte nelle lingue degli stati membri mantenendo la stessa struttura del documento, avendo così una corrispondenza di significato uno a uno per ogni paragrafo.

Questo porta ad una struttura diversa per quanto riguarda le ontologie nazionali ed europee: nel livello europeo potremo avere un *umeaning* tradotto in più lingue, mentre nel livello nazionale la lingua giuridica sarà parte integrante della definizione univoca di un *umeaning*.

D'ora in poi il termine oppure l'*umeaning* verranno detti *di livello nazionale* se appartenenti al dizionario nazionale, *di livello europeo* se appartenenti al dizionario europeo.

Gli *umeaning* nazionali ed europei sono associati fra loro e si dice che un *umeaning* nazionale *implementa* un *umeaning* di livello europeo. Ad un *umeaning* nazionale corrisponderà uno o più *umeaning* di livello europeo, in particolare solamente le loro traduzioni nella medesima lingua.

In figura è rappresentato il class diagram dei concetti espressi in precedenza, dove si può notare che la classe Termine è stata suddivisa in TermineNazionale e TermineEuropeo, allo stesso modo, la classe Umeaning è stata suddivisa in UmeaningNazionale e UmeaningEuropeo, questo per sottolineare il fatto che vengono utilizzati due dizionari separati.

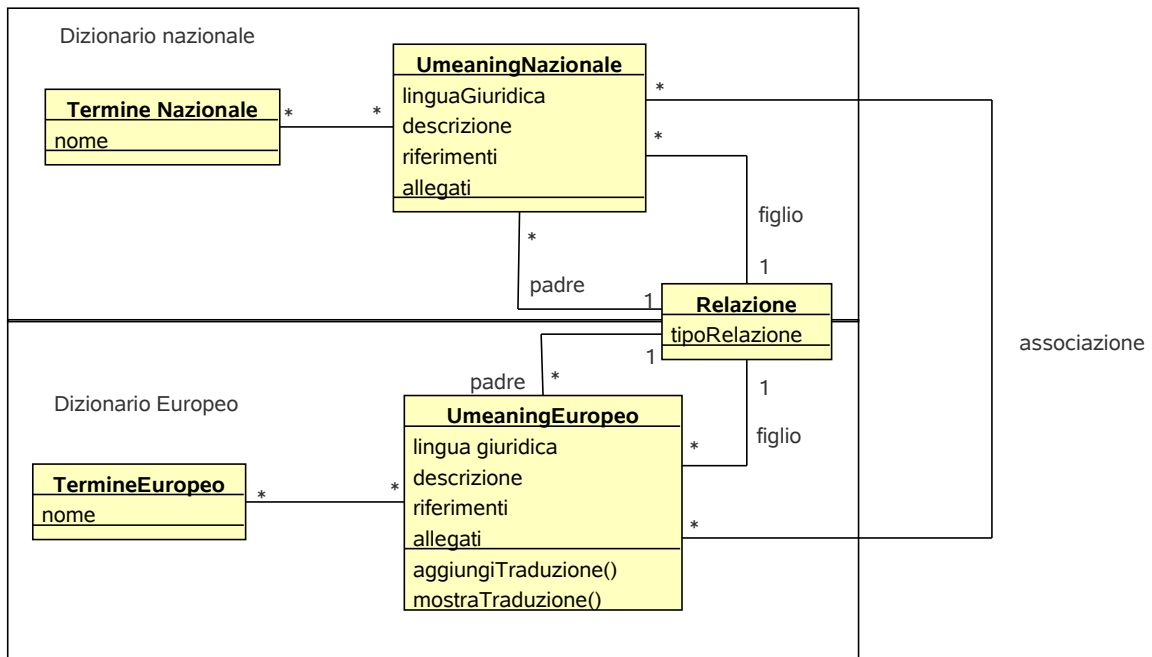


Figura 2.1 Class diagram che descrive il dominio del problema

Possiamo vedere la struttura simmetrica delle classi che modellano il problema: TermineNazionale e TermineEuropeo modellano i termini dei dizionari nazionale ed europeo rispettivamente e possiedono un'associazione molti a molti con i propri significati (umeaning) per poter consentire sia un termine con più significati che un significato associato a più termini.

Le classi Umeaning possiedono gli attributi linguaGiuridica, descrizione, riferimenti e allegati, che servono a definire il significato. Oltre a questo UmeaningEuropeo possiede due operazioni: aggiungiTraduzione e mostraTraduzione che esprimono la presenza di più traduzioni per un unico umeaning europeo.

Esiste un'associazione molti a molti tra gli umeaning dei due livelli e se stessi, per rappresentare le relazioni ontologiche fra di essi. L'esistenza di diversi tipi di relazioni è descritta dalla classe intermedia *Relazione* per cui passa la relazione tra umeaning, la quale possiede l'attributo *tipoRelazione*.

Infine, la classe UmeaningNazionale è collegata alla classe UmeaningEuropeo tramite un'associazione molti-a-molti, in quanto un umeaning nazionale può corrispondere a più umeaning europei e viceversa.

2.2 Casi d'uso e tipologia di utenti

Syllabus è una web-application rivolta ad un'utenza specializzata, in particolare ai giuristi che vogliono studiare e analizzare il diritto comparato tra la legislazione europea e le implementazioni negli ordinamenti nazionali delle direttive comunitarie.

Per schematizzare i casi di possibili utilizzi del sistema ricorriamo ad un grafico di use-case UML definito in [8], mostrato in figura 2.2.

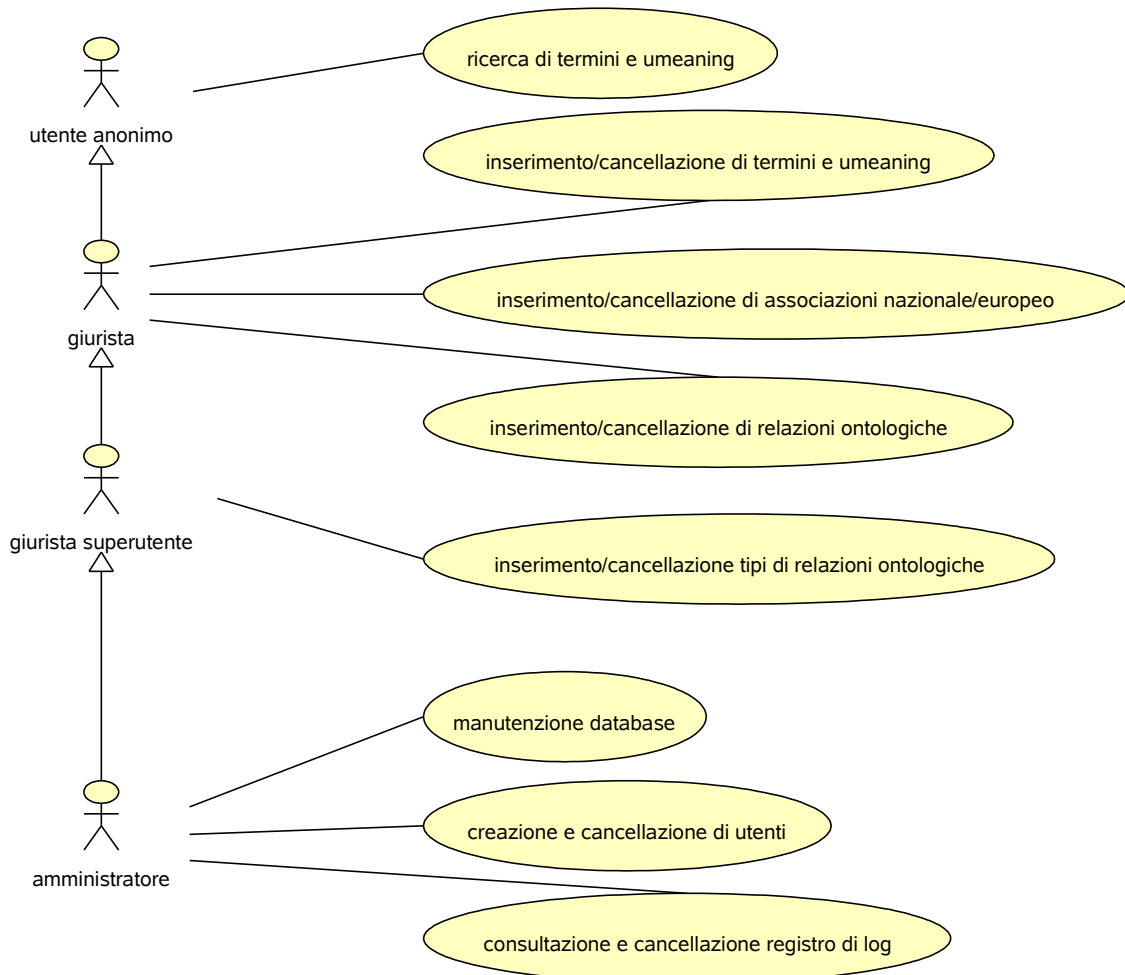


Figura 2.2 Diversi casi d'uso a seconda del tipo di utente

Il sistema è consultabile da tutti, quindi non occorre un'autenticazione per compiere semplici operazioni di ricerca e lettura, questo definito dalla tipologia di *utente anonimo*.

Gli utenti tipici del Syllabus sono giuristi, avvocati e magistrati, a cui non sono richieste particolari conoscenze informatiche ed a cui devono essere nascoste tutte le funzioni non strettamente inerenti la gestione e consultazione del dizionario. In particolare, essi devono poter:

- Effettuare una ricerca nel dizionario di termini e umeaning.
- Inserire, modificare e cancellare termini e umeaning

- Inserire e cancellare associazioni fra umeaning di livello nazionale e umeaning di livello europeo
- Inserire e cancellare relazioni ontologiche

Alcuni utenti inoltre hanno la possibilità di decidere la struttura dell'ontologia, quindi hanno anche la facoltà di aggiungere o rimuovere diversi tipi di relazioni ontologiche.

L'ultima categoria di utenti è quella degli amministratori, tipicamente personale qualificato in informatica a cui è richiesta una buona conoscenza del sistema, essi svolgono importanti operazioni non direttamente connesse alla gestione dei dizionari o dell'ontologia, in particolare:

- Eseguire una periodica manutenzione del database, come backup e deframmentazione delle tabelle
- Creare oppure cancellare utenti
- Consultare il log in cui sono conservate descrizioni di tutte le operazioni compiute sul sistema, utile ad esempio per diagnosticare eventuali problemi

2.3 Progettazione

Verrà qui esaminata la struttura del progetto e le scelte progettuali che hanno portato al sistema esistente.

2.3.1 Architettura

Il sistema è stato progettato seguendo una classica architettura three-tier molto comune delle web-application [7], illustrata in figura 2.3

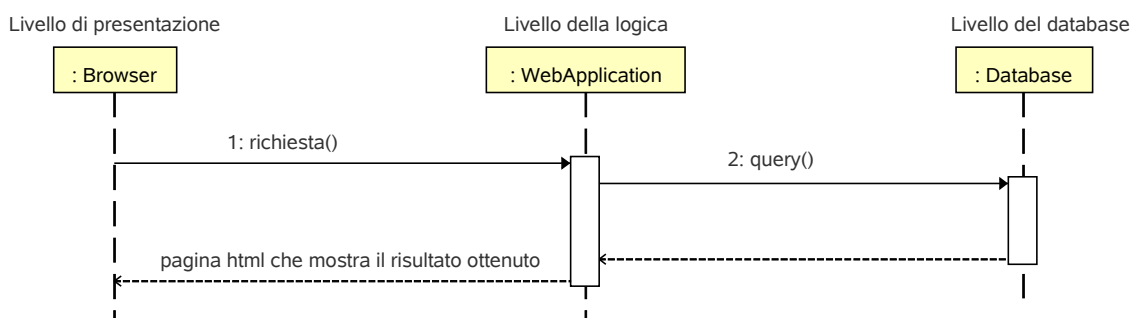


Figura 2.3 architettura di un'applicazione client/server three tier

In questo modello le funzionalità dell'applicazione sono ripartite in tre livelli:

- Livello di presentazione (presentation tier)
- Livello della logica (Logic tier)

- Livello del database (Data Tier)

Il livello del database è responsabile per l'immagazzinamento, il recupero, in mantenimento e l'integrità dei dati.

Il livello della logica si occupa delle elaborazioni dei dati che arrivano dal livello di presentazione in base alla cosiddetta *business logic*, cioè all'insieme delle regole per cui i dati sono considerati significativi rispetto all'attività che l'applicazione svolge e le loro relazioni consistenti. Spesso è a sua volta suddiviso in più unità logiche. Il nome business logic deriva dal fatto che primi software che hanno usato questa ripartizione erano per lo più software aziendali, progettati per automatizzare specifiche funzioni legati alle attività commerciali, al esempio il processamento degli ordini e la gestione dei bilanci.

Il livello di presentazione è rappresentato dall'interfaccia utente, che nel caso di una web application sarà costituita da pagine HTML ed eventualmente applet java mostrate dal browser. In questo progetto verrà utilizzato codice HTML con alcuni particolari dell'interfaccia realizzati in linguaggio Javascript.

2.3.2 Architettura object oriented

Il livello della logica è suddiviso in due moduli, *User Interaction Handler* e *Business Logic Layer* [9] (Figura 2.4)

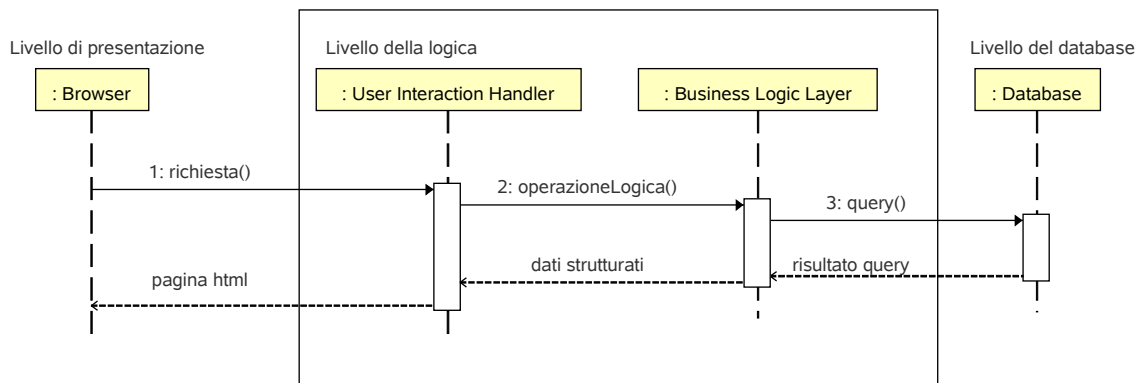


Figura 2.4 Il livello della logica suddiviso in *User Interaction Handler* e *Business Logic*

L'interazione con l'interfaccia utente è gestita dallo *User Interaction Handler*: quando l'utente compie una particolare azione, viene chiamato l'handler corretto, il quale si occupa di validare e fare controlli di sicurezza sulle richieste HTTP GET e POST, a questo punto l'handler instancia il modulo del business logic layer deputato a gestire l'azione specifica, dopo di che lo *User Interaction Handler* passa i dati al modulo del business logic layer, ed elabora la pagina HTML di risposta a partire dai dati forniti come risposta dal modulo del business logic layer.

Il livello di presentazione è gestito dalla libreria PHP Smarty [10] con cui si interagisce utilizzando le funzioni fornite da una singola classe, viene anche utilizzata un'interfaccia PHP al pacchetto software GraphViz [11], allo scopo di visualizzare graficamente la struttura gerarchica dell'ontologia.

Il modulo del business logic layer accederà al database tramite un livello di astrazione fornito dal linguaggio e fornirà il suo output allo User Interaction Handler.

In Syllabus, lo User Interaction Handler è implementato dalle classi di tipo ActionDispatcher, le quali sono specializzate per ogni tipo di funzione offerta all'utente, in particolare:

- Gestione dei dizionari: TermActionDispatcher.
- Eseguire l'upload di un file associato ad un umeaning, oppure cancellare un file caricato in precedenza: UploadActionDispatcher.
- Gestione delle lingue giuridiche: GiuridicLanguageActionDispatcher.
- Gestione dei tipi di relazione: RelationTypeActionDispatcher.
- Gestione degli utenti: UserActionDispatcher.
- Gestione del log delle operazioni: LogActionDispatcher.
- Manutenzione del database: DbMaintenanceActionDispatcher.

La classe centrale che smista le operazioni è ActionGroupDispatcher, la quale compie i seguenti passi:

- Valida l'utente attraverso il meccanismo delle sessioni PHP (attraverso cookie e dati salvati nel lato server) [1]
- Decisione dell'azione da compiere valutando il parametro url-encoded *action*, caricando la classe di tipo *ActionDispatcher* appropriata delegando il controllo ad essa.

Ogni metodo di ogni ActionDispatcher funziona secondo il medesimo schema:

- Alla propria creazione, istanzia la classe atta a gestire la business logic del gruppo di funzioni adatto (ad esempio TermActionDispatcher userà Term).
- Controlla se l'utente ha il permesso di eseguire l'azione data.
- Valida ogni parametro di input: se viene trovato un errore nell'input si mostra il messaggio di errore opportuno.

- Vengono eseguiti i metodi opportuni dell'oggetto che è stato creato dalla classe per la gestione della business logic, a seconda della loro condizione di ritorno viene mostrata la pagina opportuna, dopo di che il programma termina.

Le classi per la gestione della business logic sono le seguenti:

- Gestione dei dizionari: Term.
- Eseguire l'upload di un file associato ad un umeaning, oppure cancellare un file caricato in precedenza: Upload.
- Gestione delle lingue giuridiche: GiuridicLanguage.
- Gestione dei tipi di relazione: RelationType.
- Gestione degli utenti: UserAction.
- Gestione del log delle operazioni: Log.
- Manutenzione del database: DbMaintenance.

La figura 2.5 mostra una versione più completa del diagramma delle classi. Per semplicità in ambedue i diagrammi qui presentati le classi Smarty e GraphViz sono state omesse.

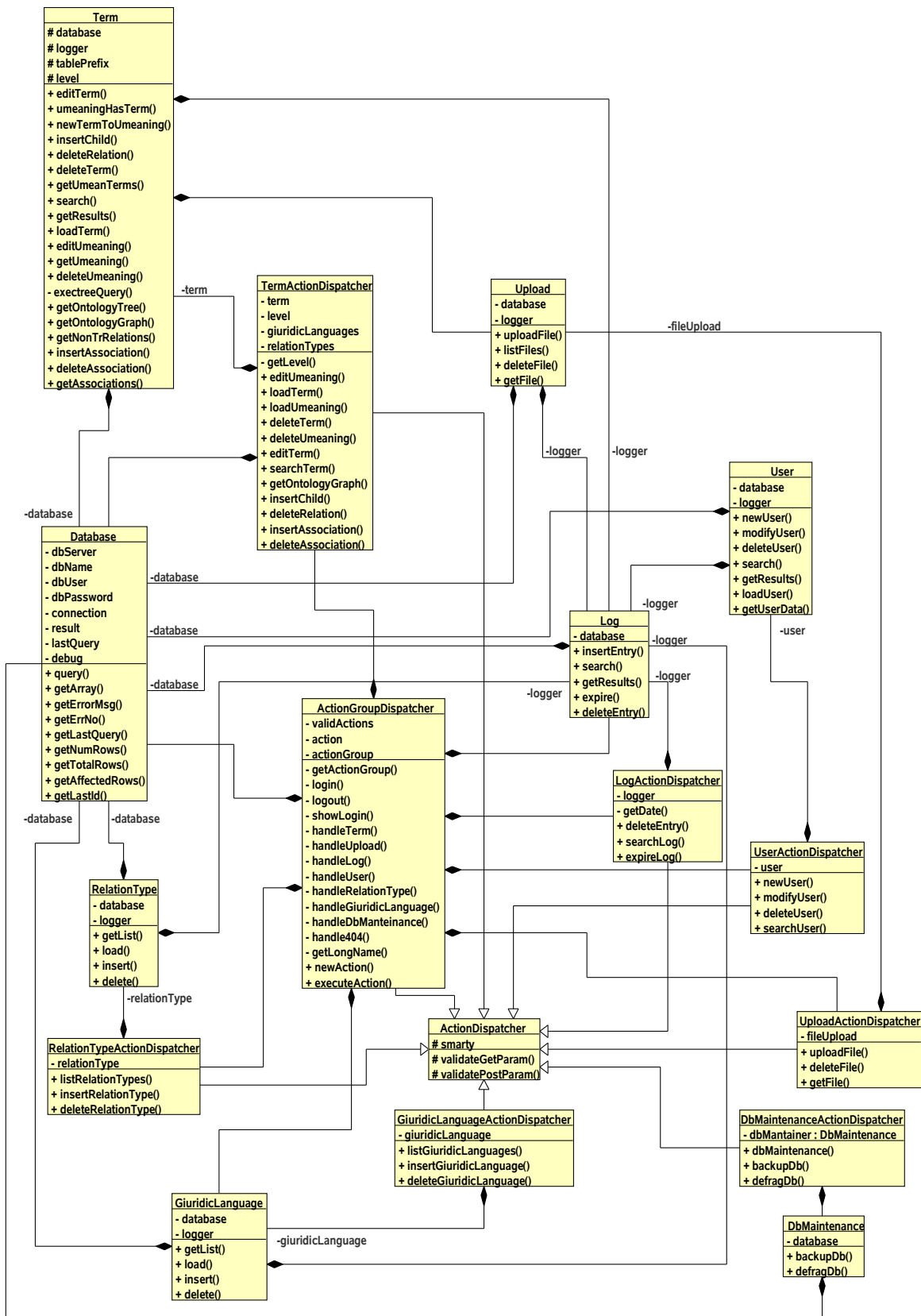


Figura 2.5 Diagramma delle classi della web-application

I metodi di tali classi, offrono funzionalità più semplici rispetto alle azioni supportate dal sistema, quindi non sempre ci sarà una corrispondenza uno-a-uno fra i metodi dell'ActionDispatcher e la corrispondente business logic: ad esempio la creazione o modifica di un umeaning è supportata dal metodo *editUmeaning* della classe *TermActionDispatcher* che avrà bisogno dei tre seguenti metodi appartenenti alla classe più di basso livello *Term*:

- *editTerm*: crea un nuovo termine oppure ne modifica uno esistente.
- *editUmeaning*: crea un nuovo umeaning oppure ne modifica uno esistente.
- *newTermToUmeaning*: crea un'associazione fra il termine e l'umeaning appena creati.

L'utilizzo di più metodi della classe di basso livello *Term* è racchiuso in una transazione SQL gestita in modo trasparente.

Seguendo il tipico approccio della programmazione object oriented i metodi delle classi della business logic si dividono in metodi per la scrittura e per la lettura del database, non esistono metodi che fanno le due cose assieme. Ognuno accetta come parametri unicamente tipi di dati primitivi, come stringhe ed interi. Quando i metodi di scrittura implementano operazioni di cancellazione ritornano valori booleani che indicano il successo oppure il fallimento dell'operazione di cancellazione. Quando essi implementano un'operazione di inserimento o di modifica su una tabella la cui chiave primaria è un intero auto-generato dal database, essi ritornano tale intero auto-generato in caso di successo, NULL in caso di fallimento.

I metodi che si occupano del recupero di dati dal database, ritornano un intero che rappresenta il numero di righe trovate dalla query, zero se la query non ha dato nessun risultato.

Ogni classe della business logic, possiede un metodo chiamato *getResults*, che serve ad ottenere i record risultanti dall'ultima query convertiti in un formato strutturato di tipo *array associativo* PHP (simile alla struttura *map* di C++ o Java), manipolabile dal codice PHP e soprattutto comprensibile al motore di template che si occuperà di presentarli in formato HTML comprensibili all'utente.

La classe *Database*, serve per connettere il livello della business logic al livello del database. Essa mantiene lo stato della connessione, e lo stato dell'esecuzione dell'ultima query, cioè il testo della query stessa, l'ultimo id numerico auto-generato e l'ultimo eventuale messaggio di errore prodotto dal database. Tale classe serve anche da “wrapper” attorno alle funzioni di connessione ed interrogazione a PostgreSQL, per rendere più semplice un'eventuale migrazione verso un diverso database server in versioni future.

2.3.3 Package

Le classi di cui l'applicazione è composta sono ordinate all'interno della gerarchia di package descritta nel diagramma delle classi di figura 2.6.

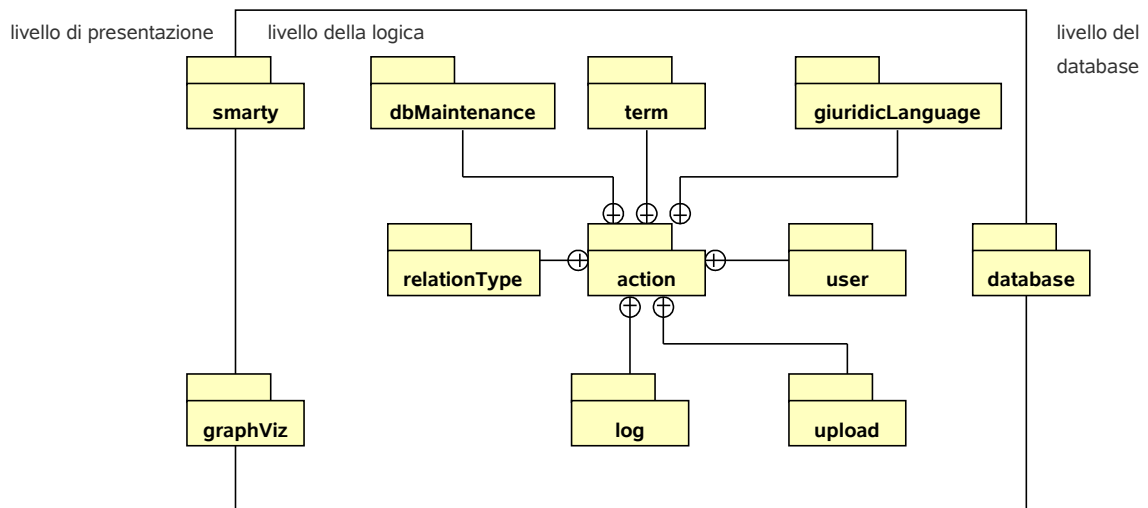


Figura 2.6 Gerarchia dei package del sistema

La prima suddivisione è fatta fra le classi appartenenti al livello della logica vero e proprio, che fanno parte del package “action” e le classi che servono per connettere tale livello con il livello del database (la classe Database) e il livello dell'interfaccia utente (le classi Smarty e GraphViz), le quali saranno contenute ognuna in un package proprio.

Il package action ha questo nome perché contiene le classi atte a gestire le diverse azioni che l'utente compie sul sistema: esso è suddiviso in un package per ogni tipo di azione.

Ogni package appartenente ad action contiene la classe dello User Interaction Handler e la classe della business logic associata, deputate a svolgere quel particolare gruppo di azioni.

2.3.4 Progettazione del database

La struttura del database ruota attorno alla definizione del concetto di *umeaning* descritto nel paragrafo 2.1.1. I dati che descrivono i livelli giuridici nazionali ed europeo sono separati in due cassi di relazioni il cui nome porta il prefisso *national_* oppure *european_* per rispecchiare la differenza strutturale dei concetti in questi due casi.

Rilevanti sono le relazioni *_term_to_umeaning* che collegano i termini ai loro significati, le relazioni *_umean_to_umean* che descrivono la struttura ontologica dei concetti e infine la relazione *european_national_umeaning* che rappresenta la trasposizione di un concetto dalla legislazione europea dalla sua *implementazione* a livello nazionale. La figura 2.7 rappresenta la struttura completa del database.

La principale differenza fra gli umeaning di livello nazionale ed europeo è l'estensione della chiave primaria: nel livello nazionale un concetto appartiene ad unico stato e di conseguenza ad un'unica lingua, mentre nel livello europeo come descritto nel paragrafo 2.1.1 una singola direttiva è tradotta nelle lingue di tutti gli stati membri.

La lingua giuridica verrà identificata dal countrycode come definito in [12]: la lingua viene identificata da due lettere minuscole, separate con un underscore da due lettere maiuscole che identificano la nazionalità: gli identificatori di lingua e nazionalità sono quelli che si trovano in ISO-639 [13]. Questo metodo consente di distinguere gli ordinamenti giuridici di Paesi che hanno la stessa lingua, come ad esempio Germania e Austria. Le lingue giuridiche disponibili sono conservate nella tabella *giuridic_language*.

La gestione delle ontologie per quanto riguarda l'implementazione attraverso un database relazionale si rifà all'approccio seguito dal progetto Gene Ontology [14] con la principale differenza nel fatto che i nodi dell'ontologia non sono rappresentati dai termini ma dai loro significati. Le tipologie di relazione ontologica possono essere di tipo transitivo (ad esempio IS_A) oppure non transitivo (ad esempio INVERSE_OF): per facilitare la fase di consultazione e interrogazione del database, come in Gene Ontology le intere gerarchie di relazioni transitive vengono sviluppate nelle tabelle *_graph_path* che rappresentano la loro chiusura transitiva.

I file allegati agli umeaning (o meglio le informazioni su tali file) sono conservati nelle tabelle *european_attachments* e *national_attachments*.

Per una maggior tracciabilità sia i termini che gli umeaning sono associati alla data di creazione, data di modifica e autore, il quale è una chiave esterna alla tabella degli utenti.

Gli utenti sono conservati nella tabella *user_account*, che contiene le seguenti informazioni: login, l'hash MD5 [15]:348 della sua password, nome, cognome, indirizzo e-mail e quattro valori booleani che specificano quali operazioni l'utente può compiere: navigare nel sito, modificare il dizionario, aggiungere o rimuovere tipi di relazione, o se sono amministratori. Il controllo di queste *capabilities* tuttavia non può venire fatto direttamente dal database, ma deve venire fatto dalla web-application, cioè dal livello della logica.

Nella tabella *operations_log* vengono salvate informazioni sulle modifiche fatte sul database, in particolare la data, l'autore della modifica, il tipo di modifica nel campo operation (ad esempio cancellazione umeaning) ed eventualmente i dati serializzati di record aggiunti o modificati nel campo message. Anche le operazioni su questa tabella sono gestite interamente dal livello della logica.

2.3.4.1 Controlli aggiuntivi sull'integrità dei dati

Affinché i dati rimangano integri è necessario eseguire alcuni controlli impossibili con comuni vincoli di chiave esterna, in particolare:

- Ad ogni *umeaning* deve corrispondere almeno un *termine*.
- Ad ogni *termine* deve corrispondere almeno un *umeaning*.
- I termini possono essere associati soltanto con *umeaning* della medesima lingua giuridica.
- Gli *Umeaning* nazionali possono essere associati soltanto ad *umeaning* europei della medesima lingua giuridica.
- Nelle tabelle della chiusura transitiva non possono comparire relazioni non transitive.
- La chiusura transitiva deve venire aggiornata ogni volta che si aggiunge o rimuove una relazione ontologica.
- I nomi dei tipi di relazioni ontologiche non possono contenere cifre o spazi, in quanto renderebbe più difficile il parsing del campo *path* delle tabelle *graph_path*.

2.3.5 Design di dettaglio

Di seguito descriviamo attraverso pseudo codice in *Process Design Language* (PDL) [16]:337-340 alcune attività compiute dalla web-application per meglio comprenderne il funzionamento..

2.3.5.1 Flusso di esecuzione

Qui di seguito è riportato l'algorithmo che il programma segue ogni volta che l'utente carica una pagina sul browser, cioè esegue una qualsiasi azione:

```

1 Connettiti al database
2 IF la connessione fallisce THEN
3   print(impossibile connettersi al database)
4   EXIT
5 ELSE
6   esegui il login dell'utente
7   IF il nome utente o la password sono errati THEN
8     print(nome utente o password errati)
9     EXIT
10  IF action non è definita THEN
11    action = searchTerm
12  IF action NOT IN validActions THEN
13    print(errore 404: pagina inesistente)
14    EXIT
15  ELSE
16    actionDispatcher = new(classe gestisce action)
17    chiama il metodo actionDispatcher.action()

```

Queste saranno tutte operazioni eseguite dalla classe ActionGroupDispatcher, istanziata da un main dalla struttura molto semplice:

```

1 esegui il parsing del file di configurazione
2 actionGroupDispatcher = new(ActionGroupdispatcher)
3 actionGroupDispatcher.newAction("login")
4 actionGroupDispatcher.executeAction()
5 actionGroupDispatcher.newAction($_GET["action"])
6 actionGroupDispatcher.executeAction()

```

Viene prima fissata l'azione a "login" ed eseguita, dopo di che viene eseguita l'azione specificata nei dati GET cioè codificata nell'URL, ad esempio index.php?action=searchTerm indicherà di eseguire l'azione searchTerm, nello pseudo codice è stata seguita la convenzione di PHP in cui ogni variabile urlencoded è mappata nella variabile globale \$_GET, la quale è un array associativo.

2.3.5.2 Autenticazione utenti

La procedura dell'autenticazione dell'utente avviene secondo i seguenti passi:

```

1 inizia o riprendi una sessione
2 cerca login e password in _POST
3 IF login e password non sono stati specificati THEN
4   cerca login e password nella sessione
5   IF login e password non sono nella sessione THEN
6     login = "anonymous"
7     password = "anonymous"
8 ELSE
9   salva login e password nella sessione
10 cerca nel database un utente con il login specificato e
    con l'hash MD5 della password specificata
11 IF non è stato trovato THEN
12   print(nome utente o password errati)
13 EXIT
14 ELSE
15   carica i dati dell'utente e imposta le sue
    capabilities

```

2.3.5.3 Azioni valide

Tutte le possibili azioni valide sono definite nella classe ActionGroupDispatcher, la quale controlla se l'azione specificata è valida ed istanzia la classe adatta a gestire il gruppo di azioni di cui l'azione richiesta fa parte. Le azioni valide catalogate per classe sono:

- ActionGroupDispatcher: login, logout, showLogin, show404; gestiscono rispettivamente il login, il logout, la generazione della pagina di login e la generazione della pagina di errore 404. Come detto in precedenza esse sono gestite direttamente dalla classe ActionGroupDispatcher per ragioni di prestazioni.
- TermActionDispatcher:
 - editUmeaning: Inserisce o modifica un dato umeaning; se specificato inserisce anche un nuovo termine associato con il nuovo umeaning e un'eventuale relazione ontologica se è stato specificato un umeaning padre e il tipo di relazione.
 - editTerm: inserisce o modifica un termine per un dato umeaning.
 - loadTerm: mostra all'utente una pagina contenente un elenco di umeaning per un dato termine.
 - loadUmeaning: mostra all'utente i dati su un umeaning: descrizione, ontologia, termini ad esso associati, umeaning dell'altro livello a cui corrisponde ed eventualmente le sue traduzioni.
 - deleteTerm: cancella un dato termine.

- deleteUmeaning: cancella un dato umeaning.
- searchTerm: esegue la ricerca nel database di termini umeaning secondo i parametri di ricerca specificati dall'utente.
- getOntologyGraph: mostra una rappresentazione grafica dell'ontologia generata mediante GraphViz [11].
- insertChild: Crea una relazione padre-figlio tra due umeaning precedentemente esistenti. Mostra una pagina di ricerca finché un id valido non viene selezionato come figlio.
- deleteRelation: cancella una relazione ontologica.
- insertAssociation: Inserisce un' associazione tra un umeaning di livello nazionale e un umeaning di livello europeo. Mostra una pagina di ricerca finché un id valido non viene selezionato dall'altro livello.
- deleteAssociation: cancella un'associazione tra un umeaning di livello nazionale e un umeaning di livello europeo.

UploadActionDispatcher:

- uploadFile: esegue l'upload di un file (che verrà associato ad un umeaning) sul server.
- deleteFile: cancella un file dal server.
- getFile: esegue il download di un file allegato.
- LogActionDispatcher:
 - searchLog: esegue una ricerca nella tabella di log secondo i parametri specificati dall'utente.
 - ExpireLog: cancella tutti gli elementi del log più vecchi della data specificata dall'utente.
 - deleteEntry: cancella un'entry dal log.
- UserActionDispatcher:
 - newUser: crea un nuovo utente.

- modifyUser: modifica un utente esistente.
- deleteUser: cancella un utente.
- searchUser: esegue una ricerca nella tabella degli utenti.
- RelationTypeActionDispatcher:
 - listRelationTypes: mostra una pagina contenente un elenco dei tipi di relazioni ontologiche disponibili.
 - insertRelationType: inserisce un nuovo tipo di relazione.
 - deleteRelationType: cancella un tipo di relazione.
- GiuridicLanguageActionDispatcher:
 - listGiuridicLanguages: mostra una pagina contenente un elenco delle lingue giuridiche disponibili.
 - insertGiuridicLanguage: inserisce una nuova lingua giuridica.
 - deleteGiuridicLanguage: cancella una lingua giuridica.
- DbMaintenanceActionDispatcher:
 - dbMaintenance: mostra una pagina contenente l'elenco delle operazioni di manutenzione al database.
 - backupDb: viene eseguito il download di un backup completo del database.
 - defragDb: viene eseguita la deframmentazione del database.

2.4 Aspetti implementativi

2.4.1 PHP

La web-application è stata implementata utilizzando il linguaggio PHP.

PHP, abbreviazione di Hypertext Preprocessor, è un linguaggio interpretato progettato ed utilizzato soprattutto per la realizzazione di applicazioni web. Come altri linguaggi dello stesso tipo, al momento esiste un'unica implementazione dell'interprete, che è open-source e disponibile sul sito <http://www.php.net>. L'integrazione con il web-server può avvenire in due diversi modi:

- Può venire configurato per funzionare come applicazione CGI (Common Gateway Interface): ad ogni richiesta viene eseguito l'eseguibile dell'interprete il cui output verrà restituito dal browser come pagina HTML. Questa modalità funziona su ogni web-server che supporta CGI, tuttavia soffre di gravi problemi di prestazioni in casi di alto traffico, in quanto ad ogni richiesta deve venire caricato un nuovo eseguibile.
- PHP offre anche delle estensioni per i web-server più comuni come Apache e Microsoft IIS: in questo caso l'interprete viene caricato come libreria a link dinamico all'interno dell'eseguibile del web-server, in questo caso c'è un'occupazione di memoria molto più contenuta anche nei momenti di alto traffico. Questo metodo, che è il più usato ha il vantaggio di avere prestazioni molto più elevate, ma tuttavia un crash dell'interprete può portare al crash dell'intero server.

La versione 5 di PHP, quella che è stata utilizzata in Syllabus fin dall'inizio del suo sviluppo introduce un migliore supporto alla programmazione object-oriented [17], rendendo possibile il design altamente modulare del sistema.

2.4.2 Smarty template engine

per mantenere meglio la separazione tra i livelli, La generazione di codice HTML non è realizzata direttamente in PHP, ma attraverso il motore di template Smarty [18], realizzato anch'esso in PHP ed è un interprete di un linguaggio di template intermedio.

I template utilizzati da smarty sono delle pagine HTML che contengono alcune direttive particolari racchiuse tra parentesi graffe. La più semplice è quella per mostrare il contenuto di una variabile, ad esempio `{$name}` mostrerà il contenuto della variabile *name* ([18]: paragrafo 2.3). Sono inoltre presenti direttive più complesse come definizione di variabili, inclusione di file, costrutti if-then-else e semplici cicli per l'iterazione di array associativi (utili ad esempio per mostrare tabelle).

Dal lato del PHP Smarty fornisce una classe i cui metodi principali sono *assign*, che assegna un valore ad una variabile che sia accessibile dal template e *display*, il quale mostra in output un file di template.

2.4.3 PostgreSQL

Per la gestione del database è stato scelto PostgreSQL per le sue caratteristiche piuttosto complete per quanto riguarda la gestione dell'integrità referenziale, caratteristica indispensabile per poter gestire correttamente ed in modo affidabile l'integrità della struttura dell'ontologia.

Esso offre supporto completo alle chiavi esterne con azioni di integrità referenziale IGNORE, RESTRICT oppure CASCADE ed il supporto ai Trigger definiti in SQL99 [19] utilizzati per mantenere l'integrità di strutture oltre la semplice integrità referenziale, come ad esempio assicurarsi l'assenza di cicli nell'ontologia.

Capitolo 3 - Obiettivi giuridici

3.1 Problema di interpretazione del diritto

Il Syllabus collega fra loro termini giuridici e concetti, utilizzando come definizione estratti di documenti legali come ad esempio direttive europee, o note interpretative dottrinali.

Le direttive europee contengono definizioni per i termini principali utilizzati all'interno di esse, ad esempio nella direttiva 93/13/CEE [20] riporta la seguente definizione di consumatore:

“[...] b) “consumatore”: qualsiasi persona fisica che, nei contratti oggetto della presente direttiva, agisce per fini che non rientrano nel quadro della sua attività professionale; [...]”

Altre direttive, che si rifanno ad argomenti diversi oppure emesse in tempi successivi possono tuttavia riportare definizioni diverse dello stesso termine, ad esempio la direttiva 2002/65/CE [21] riporta invece la seguente definizione di consumatore:

“[...] d) “consumatore”: qualunque persona fisica che, nei contratti a distanza, agisca per fini che non rientrano nel quadro della propria attività commerciale o professionale; [...]”

La differenza più importante sta nell'aggiunta del termine attività commerciale, in quanto questa definizione racchiude una categoria più ampia di persone giuridiche rispetto alla definizione della direttiva 93/13/CEE. Nel diritto europeo *professionale* indica chiunque svolge attività imprenditoriale, mentre nel diritto italiano indica solo chi è effettivamente un libero professionista, in altri ordinamenti giuridici invece le due frasi non hanno differenze di significato.

Queste differenti definizioni di termini analoghi possono capitare in quanto l'oggetto delle direttive europee è sempre molto settoriale, quindi normative riguardanti settori diversi possono recare diverse definizioni.

Il giurista esegue quindi una ricostruzione più ampia che va oltre un singolo settore per avere una visione di insieme sulla definizione di un concetto all'interno dell'ordinamento giuridico.

Nell'ordinamento italiano, in precedenza il recepimento di una direttiva europea avveniva mediante l'emanazione di un decreto legislativo per ogni singola direttiva. Questo ha portato tuttavia ad problema di ambiguità di interpretazione dove si verificava una sovrapposizione tra i concetti indicati in una direttiva ed i concetti indicati in un'altra quando essi appartenendo a settori diversi potevano entrare in parziale contraddizione.

Per rimediare a questo problema, i decreti di recepimento precedenti sono stati abrogati e accorpati nel decreto legislativo 206/2005 [22], in cui si trovano definizioni più generali dei termini in modo da racchiudere le definizioni non sempre uguali che si trovano nelle singole direttive, ad esempio la definizione di consumatore diventa:

“[...] a) consumatore o utente: la persona fisica che agisce per scopi estranei all'attività imprenditoriale o professionale eventualmente svolta;[...] ”.

La struttura precedente del Syllabus non permetteva di definire questo concetto di diversi livelli di astrazione per mezzo dell'ontologia, si vuole quindi progettare un ampliamento delle funzionalità della gestione dell'ontologia che permetta di operare questo tipo di distinzione.

3.2 Cambiamento normativo

Con cambiamento normativo si indica l'uscita di nuovo atto che cambia aspetti di altri atti precedenti, il cambiamento può avvenire in modo esplicito oppure implicito.

Nel primo caso la nuova norma riporta esplicitamente quale paragrafo è stato cambiato del testo precedente, ad esempio tornando alla direttiva 2002/65/CE essa riporta la seguente frase:

“[...] (33) In vista dell'adozione della presente direttiva, è opportuno procedere all'adeguamento del campo di applicazione della direttiva 97/7/CE e della direttiva 98/27/CE del Parlamento europeo e del Consiglio, del 19 maggio 1998, relativa a provvedimenti inibitori a tutela degli interessi dei consumatori(7), nonché del campo di applicazione del termine di risoluzione previsto nella direttiva 90/619/CEE del Consiglio, dell'8 novembre 1990, che coordina le disposizioni legislative, regolamentari e amministrative riguardanti l'assicurazione diretta sulla vita, fissa le disposizioni destinate a facilitare l'esercizio effettivo della libera prestazione di servizi e modifica la direttiva 79/267/CEE(8).[...] ”

Nel caso di cambiamento implicito invece la situazione è più ambigua: un testo normativo esprime un concetto in contraddizione con quanto affermato da un testo precedente senza indicarlo esplicitamente, in questo caso si assumerà che quanto indicato dal testo più vecchio sia stato abrogato e sostituito dalla norma contenuta nel testo più recente.

In entrambi i casi si hanno differenti concetti e quindi differenti gerarchie concettuali nel tempo. L'architettura precedente del sistema può rappresentare solamente la situazione vera ad un dato istante di tempo, senza tenere traccia della storia dell'evoluzione dei concetti giuridici dovuti al cambiamento normativo. Serve quindi un meccanismo che possa rappresentare l'evoluzione e la sostituzione dei concetti all'interno dell'ontologia.

3.3 Limitazioni del sistema delle citazioni

Gli umeaning all'interno del Syllabus vengono definiti per mezzo di estratti da testi giuridici, come ad esempio le direttive europee.

Tuttavia il contesto di una frase può non essere sufficiente. In caso di un problema interpretativo non è sufficiente vedere solamente la descrizione del termine, ma è necessario vedere il contesto della normativa, per avere un'idea del contesto in cui il termine è definito.

Ad esempio se si ha un problema interpretativo potrebbe non essere sufficiente vedere solo la definizione di consumatore, ma potrebbe essere necessario consultare gli articoli successivi in cui vengono ad esempio elencate le norme a difesa del consumatore e viene definita la controparte del consumatore.

L'interpretazione quindi non può essere limitata ad una singola frase, ma deve essere sempre di tipo intratestuale ed intertestuale, quindi è necessario analizzare il resto del testo in cui la definizione appare nonché altri testi che vertono sullo stesso argomento.

3.3.1 Costruzione terminologie giuridiche contestualizzate

Il Syllabus deve quindi disporre di un archivio delle versioni complete dei testi giuridici, i quali devono essere semanticamente collegati con gli umeaning contenuti nell'ontologia.

Deve quindi essere possibile partendo dalla definizione visualizzare il testo completo in cui saranno indicati tutti i termini collegati anche ad altri umeaning, per rendere così più semplice esplorare diversi concetti definiti in un determinato ambito.

3.4 Destinatari del diritto comparato

Il dizionario ontologico gestito dal Syllabus è destinato a diverse categorie di persone:

- Traduttori istituzionali: chi traduce le direttive europee nelle lingue ufficiali degli stati membri. Essi usano le cosiddette memorie di traduzione: analizzando il modo in cui un termine è stato tradotto in passato in vari contesti è possibile tradurre un nuovo testo in modo coerente rispetto ai testi precedenti. Tuttavia le memorie di traduzione possono rivelarsi insufficienti qualora emergano problemi o ambiguità di natura giuridica, in questo caso un'ontologia multilingua può essere di aiuto al lavoro di traduzione.
- Avvocato che operi a livello nazionale: può vedere come il diritto in un determinato ambito ad esempio il diritto del consumatore si applica in un altro stato europeo, passando attraverso il livello comunitario.

- Giudici: possono vedere come in un altro stato europeo è stato risolto lo stesso problema con cui si stanno confrontando.
- Giuristi: utile per lo studio del diritto comparato a livello teorico, specialmente negli studi di lemmatizzazione, cioè la standardizzazione a livello teorico della terminologia da utilizzare in un dato ambito.

Proprio per la grande eterogeneità delle tipologie dell'utente finale è necessario che il sistema abbia un'interfaccia accessibile ed usabile, ogni messaggio dell'interfaccia sia tradotto in tutte le lingue presenti nel dizionario.

Se l'utente richiede l'interfaccia tradotta in una determinata lingua ed effettua una ricerca nel livello europeo, deve venire selezionata in maniera predefinita la traduzione dell'umeaning corrispondente alla lingua selezionata. Più difficile il discorso per quanto riguarda gli umeaning che descrivono la legislazione di livello nazionale, in quanto tradurre i termini oppure i riferimenti ad un testo giuridico non è d'aiuto perché produce inesattezze, in quanto stessi termini possono avere significati diversi in diversi ordinamenti giuridici.

L'unica cosa che verrà quindi mostrata tradotta sarà una descrizione del termine in linguaggio naturale.

Capitolo 4 - Estensione del sistema esistente

In questo capitolo verranno esaminati i problemi posti nel capitolo 3 da un punto di vista tecnico e verranno posti in relazione all'architettura del Syllabus descritta nel capitolo 2, individuando così le modifiche strutturali da compiere nel sistema.

Alcuni aspetti del problema verranno modellati attraverso la tecnica dell'analisi orientata agli oggetti, utilizzando dei class diagram definiti in [8] e degli schemi entità-relazione, definiti in [23].

4.1 Diversi livelli di astrazione dei concetti

Come descritto nel paragrafo 3.1, diversi testi giuridici possono presentare diverse definizioni di stessi termini.

La differenza tra diverse definizioni è importante per quanto riguarda l'interpretazione di casi molto settoriali, tuttavia è anche necessaria una visione di insieme che permetta di astrarre un concetto oltre il settore di applicazione delle singole direttive.

4.1.1 Raggruppamento mediante relazioni ontologiche

La struttura gerarchica dei diversi livelli di astrazione di un concetto giuridico suggerisce l'utilizzo del sistema esistente di ontologia per rappresentare il raggruppamento.

Verrà utilizzato un tipo di relazione ad hoc che diventerà un nome riservato con un comportamento particolare: introduciamo la relazione GROUPED.

La relazione GROUPED è una relazione non transitiva, un umeaning padre è detto *group leader* e può avere un numero arbitrario di figli, mentre un umeaning figlio può essere in relazione GROUPED con un unico umeaning padre e il sistema non deve permettere altrimenti.

Il *group leader* verrà automaticamente creato con i campi descrizione, note e riferimenti vuoti (verranno compilati in un secondo momento) quando due umeaning *figli* vengono raggruppati. Se tutti gli umeaning figli oppure tutte le relazioni padre-figlio tra un group leader e i suoi figli vengono cancellate il *group leader* deve venire automaticamente cancellato.

4.1.1.1 Casi speciali per la presentazione grafica

Un group leader non è connesso direttamente a dei termini, inoltre i suoi campi descrizione note e riferimenti sono opzionali.

Ragion per cui quando viene visualizzato un *umeaning group leader* devono venire mostrati come termini tutti i termini collegati ai suoi *umeaning figli*, come anche ogni riferimento collegato agli *umeaning figli*, oltre agli eventuali riferimenti eventualmente collegati all'*umeaning padre*.

Quando l'utente chiede di visualizzare un termine il sistema attuale mostra ogni *umeaning* collegato a tale termine. Tuttavia questo approccio deve venire modificato per gli *umeaning* raggruppati siccome i *group leader* non possiedono termini propri.

Quando un termine appartenente ad un gruppo viene visualizzato deve venire caricato non l'*umeaning* direttamente collegato al termine ma il suo *group leader*, nascondendo così i membri del gruppo che verranno visualizzati solo quando esplicitamente richiesto.

Allo stesso modo quando un *umeaning* viene ricercato i componenti di un gruppo verranno esclusi dai risultati, mostrando quindi solamente i *group leader*, a meno che l'utente chieda esplicitamente di mostrare i componenti del gruppo.

4.2 Tecnologie NLP

4.2.1 Entità dei dati e bisogno di un indice

Un archivio di testi giuridici porta ad una grande mole di dati: la lunghezza media di una direttiva europea tra quelle analizzate è di circa 3000 parole equivalente a circa 20 kilobyte di dati. La parte del diritto analizzata nel sistema, le norme contrattuali sul commercio elettronico e la difesa del consumatore presenta alcune centinaia di normative europee e altrettante per ciascun ordinamento nazionale.

Il sistema deve avere un metodo efficiente per poter effettuare una ricerca di termini all'interno dei testi delle normative, ma la ricerca in un campo di testo molto grosso è piuttosto lenta in un database, inoltre come vedremo nel paragrafo 4.2.2 sorge il bisogno di eliminare dalla ricerca parole comuni che non producono risultati rilevanti e si vuole poter ricercare le parole per radice: ad esempio se ricerco “contratto” il sistema deve trovare anche le occorrenze di “contratti”.

Si dimostra quindi utile la classica tecnica dell'*indice invertito*.

4.2.1.1 *Indice invertito*

Implementazioni efficienti di algoritmi di free-text search richiedono un indice invertito. Esso è una struttura dati che mappa una chiave di ricerca (ad esempio una parola) ad una lista che elenca i documenti che contengono la chiave, oltre alla posizione della chiave stessa in ogni documento. L'insieme delle mappature fra parole e documenti è detta *posting list*.

Quest'approccio dà buoni risultati se le seguenti assunzioni sono valide:

- Le query sono molto frequenti
- Le cancellazioni saranno rare
- Le modifiche e le aggiunte di documenti saranno anch'esse rare

Nel nostro caso idealmente modifiche e cancellazioni non si verificheranno mai, perché i testi giuridici non cambiano mai nel tempo.

4.2.1.2 *Sistemi esistenti*

Esistono ad oggi molti pacchetti software che implementano efficientemente indici invertiti come ad esempio Apache Lucene [24].

Lucene è una libreria open source scritta in Java per l'information retrieval e il full-text indexing. Fornisce un insieme di classi per l'indicizzazione di file di testo quali *Document* e *IndexWriter* [25]. Ha anche la possibilità di indicizzare metadati di file non di testo, come immagini e file sonori o documenti rich-text che sono stati precedentemente convertiti in testo semplice.

Le principali classi fornite sono:

- *IndexWriter*: il componente centrale. Questa classe crea un nuovo indice e aggiunge documenti ad un indice esistente.
- *Directory*: rappresenta la locazione su disco di un indice Lucene. È una classe astratta e sono disponibili le implementazioni *FSDirectory* (indice su disco) e *RAMDirectory* (indice su memoria, per indici più piccoli ma più veloci)
- *Analyzer*: Prima di essere indicizzato il testo è passato attraverso un *Analyzer*. L'*Analyzer*, specificato nel costruttore di *IndexWriter* è la classe che estrae i token

dal testo da indicizzare eliminando il resto. Se il contenuto da indicizzare non è testo semplice, deve prima venire convertito.

- Document: rappresenta una collezione di campi. Si può pensare ad esso come ad un documento virtuale che racchiude il documento reale come un e-mail, un file di testo o una pagina web e i suoi eventuali metadati aggiuntivi, come autore, titolo e data. I documenti possono solo essere file di testo perché internamente Lucene tratta solo stringhe di tipo `java.lang.String`, quindi documenti differenti devono essere prima convertiti.
- Field: Ogni *Document* in un indice contiene uno o più campi, racchiusi in una classe chiamate *Field* consistente in una coppia chiave-valore. Ogni campo corrisponde ad un pezzo di dati su cui si può fare una query. Lucene fornisce quattro differenti tipi di Field:
 - Keyword: non analizzata ma indicizzata e salvata non modificata
 - UnIndexed: campo non analizzato e non indicizzato, ma salvato comunque nell'indice, sono valori che non vengono ricercati direttamente
 - UnStored: L'opposto di UnIndexed. Questo campo è analizzato e indicizzato ma non salvato nell'indice. Adatto per indicizzare una grande quantità di testo che non ha bisogno di essere recuperata nella sua forma originale.
 - Text: Analizzato e indicizzato. Campi di questo tipo compaiono nei risultati, ma bisogna stare attenti alla dimensione che può raggiungere.
- IndexSearcher: è il collegamento centrale con l'indice che espone diversi metodi di ricerca. Si può pensare come ad una classe che apre un indice in modalità di sola lettura.
- Term: un Term è l'unità di base per la ricerca. È simile alla classe field: è composto da una coppia di stringhe: il nome del campo e il suo valore. Vengono usati all'interno delle Query.
- Query: una classe astratta implementata da alcune sottoclassi, come TermQuery, PhraseQuery, PrefixQuery e alcune altre.
- TermQuery: è il tipo più semplice di sottoclasse di Query. È usata per il match di documenti che contengono campi con valori specifici.

- Hits: è un container di puntatori a risultati di ricerca, documenti che corrispondono ad una data query. Per motivi di prestazioni non vengono caricati tutti i documenti del risultato ma solo una piccola porzione di essi per volta.

4.2.1.3 Implementazione attraverso un database relazionale

Sistemi indipendenti come Lucene sono stati scartati in quanto sono rivolti prevalentemente all'indicizzazione di file e poco si prestano all'integrazione con un sistema relazionale precedentemente esistente, è stato invece preferito adottare tecniche di costruzione di indici mediante database come descritto in [26].

La classica implementazione di un indice invertito è attraverso una struttura a B-albero e ad uno heap. Viene utilizzato un B-albero in cui ogni entry può essere considerata come la coppia (parola, documento), l'inserimento richiede un accesso al disco per ogni istanza della parola contenuta nel nuovo documento. Come mostrato da [27], indicizzare n parole di un nuovo testo richiede $n(\log_b N - \log_b C)$ accessi al disco, dove b è il branching factor del b-albero [28]:366, N la sua profondità e C il numero di entry conservate in memoria ram. La rimozione di documenti ha un costo simile, mentre il costo di ricerca di un nodo è $\log_b N - \log_b C$ accessi al disco.

La *posting list* può essere conservata come righe in tabelle di un database relazionale invece di usare un B-albero e uno heap come precedentemente descritto. Con un indice appropriato creato sulla tabella il server SQL fornisce le stesse capacità e prestazioni (usando effettivamente un B-albero per implementare l'indice).

Un indice invertito può essere implementato in una singola tabella relazionale che associa la parola al documento in cui occorre unito alla posizione. Rispetto a [26] dove le occorrenze di una parola vengono raggruppate in un campo chiamato *block* opaco qua per semplicità è stato scelto di adottare una struttura normalizzata dove ogni riga rappresenta una singola occorrenza, i record presentano quindi la seguente struttura:

Nome campo	Tipo di dati
fulltext_id	integer
Word	String
Position	integer

In questo modo si ha una maggiore occupazione di spazio, tuttavia rende più efficiente il recupero della lista di occorrenze siccome non sono presenti campi multivalore a cui si dovrebbe fare una successiva operazione di parsing, rendendo più semplice anche alcuni

calcoli di tipo statistico utili per il calcolo della document similarity presentata nel paragrafo . Presenta una struttura simile all'architettura del sistema SIRE presentato in [29], a cui si rifarà anche la computazione della document similarity.

La chiave primaria della tabella è la coppia (fulltext_id, position), inoltre esiste un indice di tipo B-albero sul campo word per velocizzare la ricerca in quanto solitamente le ricerche verranno compiute per parola, ad esempio se si vuole recuperare la lista di documenti in cui compare la parola “contratto” potrà essere utilizzata questa semplice query:

```
16SELECT distinct fulltext_id
17from fulltext_index
18where word = “contratto”;
```

Un'altra misura che tornerà utile in seguito è il numero di occorrenze di ogni parola in ogni documento, ottenibile con la seguente query:

```
19SELECT fulltext_id, word, count(*) AS frequency
20FROM fulltext_index
21GROUP BY fulltext_id, word
22ORDER BY fulltext_id, frequency DESC
```

In figura 4.1 è presentato lo schema entità-relazione che descrive l'archivio di testi giuridici e il suo rapporto con l'indice invertito.

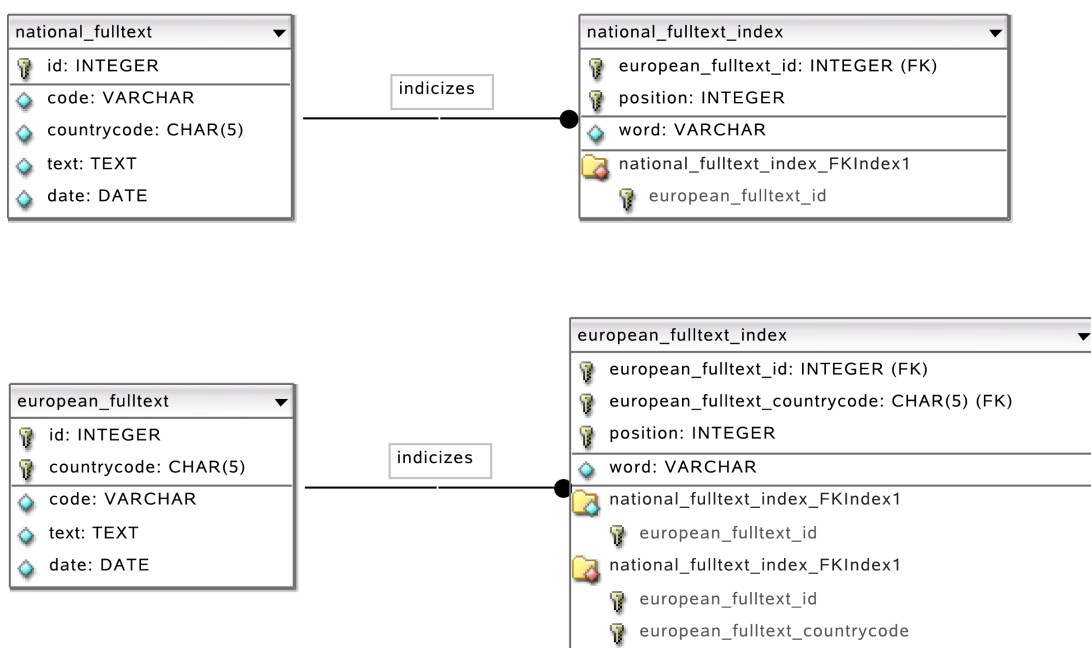


Figura 4.1: Struttura delle tabelle degli indici nazionale ed europeo

Come ricordato nel paragrafo la struttura del database è simmetrica per quanto riguarda gli ordinamenti nazionali ed europeo, in quanto un unico *umeaning* europeo viene tradotto in tutte le lingue degli stati membri cosa che non avviene per gli *umeaning* nazionali.

Questo succede in quanto viene rispecchiata la struttura dei testi legali, quindi nell'archivio testi avremo per quanto riguarda *european_fulltext* la coppia (*id*, *countrycode*) come chiave primaria, permettendo così ad un solo testo di apparire più volte in diverse lingue, mentre per quanto riguarda *national_fulltext* l'unica chiave è l'identificatore univoco, per riflettere il fatto che un testo legale nazionale può essere in una ed una sola lingua. In entrambi i casi il campo *text* contiene il testo completo della normativa, è un campo di tipo TEXT adatto per contenere testi molto lunghi ma non per effettuare direttamente una ricerca.

L'indice invertito è collegato alla tabella contenente il testo completo attraverso la chiave esterna collegata all'identificativo univoco del testo.

4.2.2 Ricerca attraverso uno stemmer

La ricerca di parole in un grande archivio testuale può portare a risultati non soddisfacenti nel caso la ricerca venga compiuta su parole esatte, ad esempio una ricerca sul termine “contratto” escluderebbe dai risultati tutti i testi contenenti il termine “contratti”. È necessario quindi cercare non tanto il termine quanto la sua radice grammaticale.

L' estrazione delle radici grammaticali in maniera algoritmica è chiamata *stemming* e sono disponibili diversi approcci e diversi pacchetti software per compiere questa operazione in testi di diverse lingue.

4.2.2.1 Sistemi per lo stemming

I sistemi per lo *stemming* prodotti negli anni si basano su alcune assunzioni: essi operano sul linguaggio scritto e non parlato, in quanto ora si tende a considerare la lingua parlata e la lingua scritta come due linguaggi diversi anche se profondamente correlati, per il momento tuttavia gli *stemmer* utilizzano unicamente la forma scritta per la differenza di *overhead* computazionale necessario a processare un linguaggio orale.

L'idea dello *stemming* è di migliorare le performance dell'information retrieval portando sotto un'unica forma di parola diverse altre che condividono un significato comune: in diversi test [30] pare che ci sia un effettivo miglioramento anche se non sempre consistente.

Lo *stemming* è un concetto che non si può applicare a tutte le lingue, non è per esempio applicabile al cinese, ma per i linguaggi del gruppo Indo-europeo emerge una struttura comune. Assumendo che le parole siano scritte da sinistra a destra lo *stem* o radice di una

parola è alla sinistra e possono essere presenti a destra zero o più suffissi, allo stesso modo prefissi possono essere aggiunti a sinistra.

Ad esempio *unhappiness* ha come prefisso *un* e come suffisso *ness*, e *y* di *happy* diventa *i* con l'aggiunta del suffisso. Di solito i prefissi alterano il significato della parola, quindi i sistemi di stemming di solito non li rimuovono (ad esempio *un* inverte il significato). Mentre per quanto riguarda i suffissi ad esempio *happy* e *happiness* hanno significati correlati, quindi lo stemmer le ridurrà entrambe alla forma *happy* oppure *happi*.

Il sistema di stemming usato nel progetto è in realtà un gruppo di stemmer linguistici in diverse lingue scritti in linguaggio *Snowball* [31], un linguaggio di gestione di stringhe, realizzato nel 2001 da Martin Porter per facilitare la creazione di stemmer utilizzando l'omonimo algoritmo.

L'algoritmo di Porter [32] è descritto nel seguente paragrafo.

Alcune definizioni: una *consonante* è una lettera diversa da A, E, I, O, oppure U e diversa da Y preceduta da un'altra consonante. Se una lettera non è una consonante è una *vocale*.

Una consonante verrà denotata con *c*, una vocale con *v*, una lista di consonanti più lunga di zero *C* e una lista di vocali *V*. una parola quindi può assumere le forme CVCV ... C, CVCV ... V, VCVC ... C oppure VCVC ... V.

Parentesi quadre denotano la presenza opzionale del loro contenuto, parentesi graffe denotano ripetizione, quindi le quattro forme precedenti possono essere espresse con [C] (VC){m}[V], m verrà chiamata la *misura* della parola.

Le regole per la rimozione di un prefisso vengono denotate con (condizione) S1 -> S2: se una parola finisce con il suffisso S1 e la radice soddisfa la condizione, S1 È sostituita con S2. Ad esempio ($m > 1$) *EMENT* -> modifica *replacement* in *replace*, siccome *replac* è una parola con $m=2$.

La condizione può essere sulla presenza di vocali o consonanti ed essere un insieme di regole composte con connettivi logici, ad esempio ($m > 1$ and (*S or *T)).

L'algoritmo in versione inglese applica le seguenti regole:

Il passo 1 gestisce i plurali e i participi passati.

Step 1a

1	SSES	->	SS	caresses	->	caress
2	IES	->	I	ponies	->	poni
3	SS	->	SS	caress	->	caress
4	S	->		cats	->	cat

Step 1b

5	(m>o) EED -> EE	feed	->	feed
6	(*v*) ED ->	plastered	->	plaster
7	(*v*) ING ->	motoring	->	motor

Se la seconda o terza regola nello step 1b ha successo vengono applicate le seguenti:

8	AT -> ATE	conflat(ed)	->	conflate
9	BL -> BLE	troubl(ed)	->	trouble
10	IZ -> IZE	siz(ed)	->	size
11	(*d and not (*L or *S or *Z)) -> singola lettera			
		hopp(ing)	->	hop
		tann(ed)	->	tan
12	(m=1 and *o) -> E	fil(ing)	->	file

La rimozione di una singola lettera causa la rimozione di una doppia, la E è messa di nuovo dopo -AT, -BL, -IZ in modo che i suffissi -ATE, -BLE e -IZE possano essere riconosciuti dopo.

Step 1c

13	(*v*) Y -> I	happy	->	happi
		sky	->	sky

Step 2

14	(m>o) ATIONAL -> ATE	relational	->	relate
15	(m>o) TIONAL -> TION	conditional	->	condition
		rational	->	rational
16	(m>o) ENCI -> ENCE	valenci	->	valence
17	(m>o) ANCI -> ANCE	hesitanci	->	hesitance
18	(m>o) IZER -> IZE	digitizer	->	digitize
19	(m>o) ABLI -> ABLE	conformabli	->	conformable
20	(m>o) ALLI -> AL	radicalli	->	radical
21	(m>o) ENTLI -> ENT	differentli	->	different
22	(m>o) ELI -> E	vileli	->	vile
23	(m>o) OUSLI -> OUS	analogousli	->	analogous
24	(m>o) IZATION -> IZE	vietnamization	->	vietnamize
25	(m>o) ATION -> ATE	predication	->	predicate
26	(m>o) ATOR -> ATE	operator	->	operate

Capitolo 4 -Estensione del sistema esistente

27	(m>o)	ALISM	->	AL	feudalism	->	feudal
28	(m>o)	IVENESS	->	IVE	decisiveness	->	decisive
29	(m>o)	FULNESS	->	FUL	hopefulness	->	hopeful
30	(m>o)	OUSNESS	->	OUS	callousness	->	callous
31	(m>o)	ALITI	->	AL	formaliti	->	formal
32	(m>o)	IVITI	->	IV	sensitiviti	->	sensitive
33	(m>o)	BILITI	->	BLE	sensibiliti	->	sensible

Step 3

34	(m>o)	ICATE	->	IC	triplicate	->	triplic
35	(m>o)	ATIVE	->		formative	->	form
36	(m>o)	ALIZE	->	AL	formalize	->	formal
37	(m>o)	ICITI	->	IC	electriciti	->	electric
38	(m>o)	ICAL	->	IC	electrical	->	electric
39	(m>o)	FUL	->		hopeful	->	hope
40	(m>o)	NESS	->		goodness	->	good

Step 4

41	(m>1)	AL	->		revival	->	reviv
42	(m>1)	ANCE	->		allowance	->	allow
43	(m>1)	ENCE	->		inference	->	infer
44	(m>1)	ER	->		airliner	->	airlin
45	(m>1)	IC	->		gyroscopic	->	gyroscop
46	(m>1)	ABLE	->		adjustable	->	adjust
47	(m>1)	IBLE	->		defensible	->	defens
48	(m>1)	ANT	->		irritant	->	irrit
49	(m>1)	EMENT	->		replacement	->	replac
50	(m>1)	MENT	->		adjustment	->	adjust
51	(m>1)	ENT	->		dependent	->	depend
52	(m>1 and (*S or *T))	ION	->		adoption	->	adopt
53	(m>1)	OU	->		homologou	->	homolog
54	(m>1)	ISM	->		communism	->	commun
55	(m>1)	ATE	->		activate	->	activ
56	(m>1)	ITI	->		angulariti	->	angular
57	(m>1)	OUS	->		homologous	->	homolog
58	(m>1)	IVE	->		effective	->	effect
59	(m>1)	IZE	->		bowdlerize	->	bowdler

I suffissi sono rimossi, vengono ancora effettuati alcuni controlli.

Step 5a

```
60 (m>1) E      ->          probate      -> probat
                                rate        -> rate
61 (m=1 and not *o) E ->          cease     -> ceas
```

Step 5b

```
62 (m > 1 and *d and *L) -> singola lettera
                                controll    -> control
                                roll         -> roll
```

Linguaggio Snowball

Lo Snowball è un linguaggio di programmazione incentrato alla manipolazione e al matching di stringhe, finalizzato alla produzione di stemmer. Un programma Snowball viene compilato una forma intermedia come programma C per poi venire infine compilato in codice macchina.

Di seguito due frammenti dell'implementazione in Snowball dello stemmer in italiano fornita con il pacchetto software: *attached_pronoun* ricerca all'interno della parola un pronome, in questo caso la parola è un verbo riflessivo che può essere scomposto (esempio *mettermeli e mettiela*).

La funzione *standard_suffix* cerca altri suffissi che possono essere rimossi dalla parola, come *trovabile* e *liberamente*.

```
define attached_pronoun as (
  [substring] among(
    'ci' 'gli' 'la' 'le' 'li' 'lo'
    'mi' 'ne' 'si' 'ti' 'vi'
    // the compound forms are:
    'sene' 'gliela' 'gliela' 'glieli' 'glielo' 'gliene'
    'mela' 'mele' 'meli' 'melo' 'mene'
    'tela' 'tele' 'teli' 'telo' 'tene'
    'cela' 'cele' 'celi' 'celo' 'cene'
    'vela' 'vele' 'veli' 'velo' 'vene'
  )
  among( (RV)
```

```

'ando' 'endo' (delete)
'ar' 'er' 'ir' (<- 'e')
)

define standard_suffix as (
  [substring] among(
    'anza' 'anze' 'ico' 'ici' 'ica' 'ice' 'iche' 'ichi' 'ismo'
    'ismi' 'abile' 'abili' 'ibile' 'ibili' 'ista' 'iste' 'isti'
    'ist{a`}' 'ist{e`}' 'ist{i`}' 'oso' 'osi' 'osa' 'ose' 'mente'
    'atrice' 'atrici'
    'ante' 'anti' // Note 1
    ( R2 delete )
    'azione' 'azioni' 'atore' 'atori'
    ( R2 delete
      try ( ['ic'] R2 delete )
    )
    'logia' 'logie'
    ( R2 <- 'log' )
    'uzione' 'uzioni' 'usione' 'usioni'
    ( R2 <- 'u' )
    'enza' 'enze'
    ( R2 <- 'ente' )
    'amento' 'amenti' 'imento' 'imenti'
    ( RV delete )
    'amente' (
      R1 delete
      try (
        [substring] R2 delete among(
          'iv' ( ['at'] R2 delete )
          'os' 'ic' 'abil'
        )
      )
    )
  )
  'it{a`}' (
    R2 delete
    try (
      [substring] among(
        'abil' 'ic' 'iv' (R2 delete)

```

```

        )
    )
)
'ivo' 'ivi' 'iva' 'ive' (
    R2 delete
    try ( ['at'] R2 delete ['ic'] R2 delete )
)
)
)

```

4.2.2.2 *Stemming e indice*

Per implementare quindi una funzionalità di ricerca soddisfacente è quindi necessario ricercare non i termini esatti forniti dall'utente, quanto le loro radici computate attraverso lo stemmer, dovremo quindi indicizzare le radici delle parole presenti nei documenti quando l'indice descritto nel paragrafo 4.2.1.1 viene compilato, aumentando così il numero di entry dell'indice che corrispondono ad una data query.

4.2.2.3 *Function words*

In un testo di qualsiasi lingua la maggioranza delle parole che vi compaiono hanno uno scarso valore lessicale e vengono usate per lo più per connettere le altre parole in una frase. Esse vengono chiamate *function words* oppure *stop words*. Esse possono essere preposizioni, pronomi, verbi ausiliari, articoli e congiunzioni. Sono una classe chiusa (non presentano una grande evoluzione negli anni) contrapposte alle *content words* che fanno parte di una classe aperta: ne nascono molte negli anni mentre altre cadono in disuso.

In inglese le function words sono circa 500, simili numeri si hanno anche per le lingue neolatine e germaniche.

Quando si effettua una ricerca in un insieme di documenti l'eventuale inclusione di function-words nella ricerca produce una grande quantità di risultati spuri che non riflettono la query, inoltre siccome l'occorrenza delle function words rappresenta la maggior parte delle occorrenze totali di parole in un documento l'indice invertito descritto nel paragrafo 4.2.1.1 cresce molto velocemente, producendo un'eccessiva occupazione di memoria fisica e degradando le prestazioni delle query.

Nel progetto è stata utilizzata la lista di function words compilata dal dipartimento di informatica dell'università di Neuchatel [33], le liste di function words per le lingue usate dal Syllabus sono così composte:

- Inglese: 571 parole
- Francese: 463 parole
- Tedesco: 603 parole
- Italiano: 430 parole
- Spagnolo: 307 parole

Le liste sono state compilate secondo il metodo descritto in [34]: è stato preso un insieme di documenti per ogni lingua che rappresentassero un gruppo statisticamente rilevante per quanto riguarda la distribuzione della frequenza delle parole contenute e ne sono state estratte le 200 parole con più occorrenze.

Sono stati rimossi dalla lista tutti i numeri e i nomi che potessero essere collegati con l'argomento principale del documento e altri nomi comuni ma che possono essere comunque utili nell'indicizzazione (ad esempio "Italia" in italiano).

Infine sono state aggiunte altre parole povere di informazione anche se non erano nella lista originale, come pronomi personali o possessivi, ad esempio "mio".

Facendo qualche confronto nel sistema finale, non indicizzando le parole contenute nella lista delle stop-word la grandezza dell'indice si riduce a circa la metà, ad esempio per quanto riguarda la traduzione italiana della direttiva europea 90/314/CEE [35] l'indice presenta 2336 record comprese le function words (numeri esclusi) e 1152 esclusi numeri e function-words.

4.2.3 Document similarity

Come descritto nel paragrafo 2.1.1 l'ontologia gestita dal Syllabus ha come elemento centrale il concetto di *umeaning*, il significato di un termine e ciò che descrive il significato è un passo di un testo giuridico.

Quando il giurista compila il database ontologico deve quindi cercare le parti di interesse in un archivio di documenti grande e potenzialmente molto eterogeneo. Un grande aiuto può venire dalla computazione automatica delle similitudini fra i documenti. Se visualizzando un documento viene presentata la lista dei documenti maggiormente simili a quello corrente, quando alcuni riferimenti sono già stati associati ad un *umeaning* sarà più facile trovarne eventualmente dei nuovi sfogliando prima i documenti che il sistema ha identificato come simili (clustering).

Esistono alcune tecniche basate sull'analisi statistica nella co-occorrenza di termini. Nel sistema è stata implementata la *cosine document similarity* [36], analizzata nel seguente paragrafo.

4.2.3.1 *Peso dei termini all'interno di documenti*

È comune a diverse tecniche di *document similarity* trattare i documenti come vettori [37] di termini. Il punto centrale è associare diversi pesi ai termini, esistono svariate tecniche di computazione e assegnamento di pesi ai termini e per il momento nessuna si è dimostrata nettamente migliore delle altre. Siccome differenti termini hanno differente importanza in un testo, un indicatore di importanza viene associato a ciascun termine del vettore che rappresenta il documento.

I tre fattori che influenzano maggiormente l'importanza di un termine in un testo sono la sua frequenza (*tf*) l'*inverse document frequency factor (idf)* e la normalizzazione della lunghezza del documento.

Esistono alcune misure che dipendono solo dalle frequenze in un documento e non dalle frequenze inter-documento. Esse sono:

- Binaria: ogni parola che appare ha la stessa rilevanza, u se appare, 0 altrimenti., denotata con $\chi(f_{ij})$
- Frequenza: numero di occorrenze di una parola nel documento. Usata da sola favorisce parole comuni e documenti lunghi.
- Augmented normalized term frequency: $K * \chi(f_{ij}) + (1 - K) * \frac{1}{\max k_{kj}(f_{kj})}$ da credito ad ogni parola che appare e un po' più credito alle parole che appaiono di

frequente, di solito il valore di K è 0.5 per documenti corti e 0.3 per documenti più lunghi. Questa formula tende a favorire risultati in documenti più lunghi

- Frequenza logaritmica: $\log(f_{ij} + 1)$ diminuisce l'effetto di grandi differenze nella frequenza di termini.

La misura dell'importanza di un termine non è molto accurata nel contesto globale se l'analisi è fatta solamente sul documento stesso, esistono quindi altre misure basate sulla distribuzione di un termine su tutto il range di documenti. Il peso globale ha di solito un discreto successo nella corretta interpretazione dei valori. In teoria potrebbe eliminare il bisogno della rimozione delle function words ma in pratica si hanno migliori risultati sia in accuratezza che in prestazioni se esse vengono rimosse in un'operazione di preprocessing.

Alcune comuni formule per la computazione dei pesi globali sono:

- Inverse Document frequency (IDF): $\log\left(\frac{n}{\sum_{k=1}^n \chi(f_{ik})}\right)$ è la misura più usata: è

definita come il logaritmo tra il rapporto del numero di documenti in una collezione e il numero di documenti che contengono una parola data. Parole rare hanno un IDF alto e parole comuni un IDF basso. Ad esempio una parole che compare in ogni documento ha un valore IDF di 0. Il peso aumenta al calare della percentuale dei documenti che contengono quella parola. Una variante usata ogni tanto è la Square Inverse Document frequency, dove l'argomento del logaritmo è elevato al quadrato.

- Probabilistic Inverse Document frequency: $\log\left(\frac{n - \sum_{k=1}^n \chi(f_{ik})}{\sum_{k=1}^n \chi(f_{ik})}\right)$ assegna pesi nel

range $-\infty$ per termini che appaiono in un solo documento a $\log(n-1)$ per termini che appaiono in un unico documento. Differisce dalla IDF nel fatto che da' sempre un peso negativo a documenti che appaiono in più di metà dei documenti.

- GFIDF: $\frac{\sum_{k=1}^n f_{ik}}{\sum_{k=1}^n \chi(f_{ik})}$ calcola il rapporto tra il numero totale di volte che il termine

appare nella collezione al numero di documenti in cui appare. Se un termine appare una volta in ogni documento o una volta in un documento viene assegnato il peso uno, il più piccolo possibile. Questo metodo funziona bene solo quando è combinato con un altro.

- Entropia: $1 + \sum_{j=1}^n \frac{p_{ij} * \log(p_{ij})}{\log(n)}$, $p_{ij} = \frac{f_{ij}}{\sum_{k=1}^n f_{ik}}$ basato sulla teoria dell'informazione

[38] ed è il più sofisticato. Assegna pesi tra 0 e 1, se appare in ogni documento ha peso 0. Se appare in un solo documento ha peso 1.

Nel sistema i risultati secondo ad una query verranno ordinati secondo la rilevanza calcolata attraverso il prodotto tra la frequenza di un termine e la sua IDF.

La terza componente è la normalizzazione usata per correggere le discrepanze nelle lunghezze dei documenti. È utile normalizzare i vettori dei documenti in modo che i documenti facciano il matching indipendentemente dalla loro lunghezza, altrimenti i valori dei pesi possono essere ingannevoli, in quanto documenti lunghi di solito usano ripetutamente gli stessi termini, quindi hanno una frequenza dei termini più alta. Inoltre documenti lunghi hanno anche un numero differente di termini. Questo aumenta il numero di match tra una query su di un documento lungo rispetto ad una su di un documento corto.

Esistono alcune formule di normalizzazione:

- Nessun cambiamento: usata quando si vogliono enfatizzare i documenti lunghi
- Normalizzazione coseno: $\sqrt{\sum_{k=1}^n (g_k * t_{ki})^2}$ è la più usata. In questa formula g_k

rappresenta il peso globale del termine k e t_{ki} il peso del termine k nel documento i . Frequenze più alte aumentano la penalità sui pesi. Inoltre se un documento ha più termini, il numero di pesi individuali nel coseno aumenta, portando un fattore di normalizzazione più alto. Quindi documenti più lunghi hanno pesi individuali più piccoli favorendo i documenti corti.

- Esistono due varianti raramente usate: *somma di pesi* $\sum_{k=1}^n (g_k * t_{ki})^2$ e *quarta*

normalizzazione $\sum_{k=1}^n (g_k * t_{ki})^4$

- Max weight normalization: $\max_{k=1}^n (g_k * t_{ki})^4$ assegna pesi tra 0 e 1, ma non

considera la distribuzione dei termini all'interno dei documenti

- Pivoted unique normalization: $\frac{1}{(1 - slope) * pivot + (slope * l_i)}$ nella formula, l_i è il numero di termini distinti nel documento j slope è solitamente impostato a 0.2 e pivot il numero medio di termini distinti nell'intera collezione. Il suo obiettivo è

correggere le discrepanze derivanti dal rapporto della lunghezza del documento e la probabilità che il documento sia rilevante.

4.2.3.2 Cosine document similarity

Avendo una misura di rilevanza dei termini, possiamo misurare la similarità fra due documenti, basandoci sulla rilevanza dei termini contenuti in essi.

Se su un piano abbiamo due punti come in figura 4.2 rappresentati da due vettori (x_1, y_1) e (x_2, y_2) [36] la loro distanza può essere calcolata come $\cos \theta$, cioè

$$\frac{A \cdot B}{|A||B|} = \frac{x_1 * x_2 + y_1 * y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}}$$

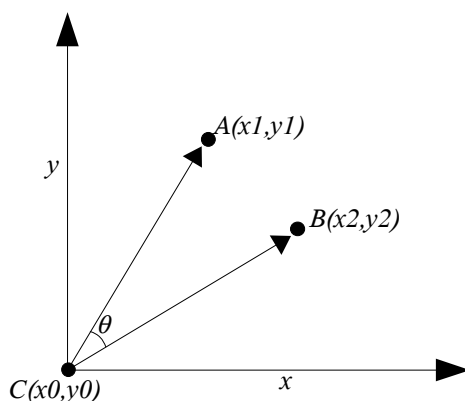


Figura 4.2: Distanza tra i vettori A e B

Se consideriamo i vettori A e B (o un numero arbitrario di vettori) come documenti, le cui componenti rappresentano i pesi dei termini contenuti nel documento possiamo definire la distanza fra due arbitrari documenti come [39]:

$$\cos(q, d) = \frac{\sum_{(t \in q) \wedge (t \in d)} f_{d,t} f_{q,t}}{\sqrt{\left(\sum_{t \in q} f_{q,t}^2\right) \left(\sum_{t \in d} f_{d,t}^2\right)}} \quad \text{dove } f_{x,t} \text{ è la frequenza del termine } t \text{ nel documento } x,$$

calcolata come numero occorrenze*IDF (vedere paragrafo 4.2.3.1).

Nel sistema la formula precedente è applicata al prodotto cartesiano di tutti i documenti, si avrà quindi la distanza tra ogni coppia possibile. I dettagli sono descritti nel paragrafo 6.3.3.

4.3 Gestione dell'evoluzione temporale in un'ontologia

Come descritto nel paragrafo 3.2 l'evoluzione delle normative può portare all'obsolescenza di alcuni concetti in favore di altri che appaiono in testi giuridici più recenti.

Succede così che un meaning può venire sostituito da un altro. Ciò che viene richiesto al sistema non è tanto una semplice cancellazione dell'meaning vecchio. Quello che si vuole ottenere è dare al giurista la possibilità di visualizzare la linea temporale dei cambiamenti e consultare tanto la situazione attuale dell'ontologia di concetti quanto le situazioni passate, per poter capire meglio l'evoluzione normativa.

Questo è un problema simile a quello cui si trovano davanti i progettisti di data warehouse, verrà quindi risolto in una maniera simile, seppur utilizzando la struttura di relazioni ontologiche già presente.

4.3.1 Similitudini con un data warehouse

Un data warehouse è una particolare classe di database aziendali contenenti prevalentemente informazioni storiche, contrapposte al sistema utilizzato per le operazioni giornaliere chiamato database operativo [40].

Esso ha dimensioni solitamente superiori ad un database operativo ed integra dati provenienti da più fonti eterogenee, analizza i fatti e le informazioni da essi generati. I dati sono strutturati secondo il modello multidimensionale, che permette un'analisi dei dati da diverse prospettive, chiamate dimensioni. Nell'esempio della tabella 1 si può vedere la misura valore venduto analizzata nell'ambito delle dimensioni punto vendita e tipo prodotto in un ipotetico data warehouse che tiene traccia delle attività di una catena di negozi [41].

Tabella 1: descrizione bidimensionale del fenomeno vendita

		Punto vendita		
		Asti	Milano	Torino
Tipo prodotto	matite	0	1500	700
	quaderni	0	1400	1000
	zainetti	400	8000	5000

Le dimensioni possono facilmente salire oltre a due, i dati fanno quindi concettualmente parte di un ipercubo ad n dimensioni le cui coordinate sono le diverse dimensioni.

Il tempo nei data warehouse è solitamente una dimensione: in questo esempio si potrebbe avere un'ulteriore dimensione che descrive i mesi: potrei quindi sapere ad esempio quante

matite sono state vendute ad Asti a gennaio 2006, ma anche le gerarchie delle misure sebbene in misura minore sono dinamiche: in questo esempio i negozi, descritti nel database in tutti i loro dettagli possono cambiare alcuni loro attributi, come ad esempio il nome o il gestore oppure possono essere chiusi e nascerne di nuovi.

I data warehouse possono tenere traccia della dinamicità delle gerarchie temporali con diverse granularità, siccome può essere un'operazione costosa sia in tempo che in spazio, si possono avere alcuni scenari [42][43]:

- Oggi o ieri: ciascun evento è riferito al valore delle gerarchie all'istante di tempo in cui si è verificato, è il più completo.
- Ieri per oggi: Ciascun evento è riferito al valore iniziale delle gerarchie.
- Oggi per ieri: Gli eventi sono riferiti alla situazione attuale delle gerarchie, di solito è l'opzione di default.
- Oggi e ieri: Vengono considerati solo gli eventi che si riferiscono a valori immutati nella gerarchia.

Esaminiamo lo scenario *oggi o ieri* mediante un esempio:

Situazione della dimensione negozio il primo gennaio 2001:

Tabella 2: Situazione della dimensione negozio il primo gennaio 2001

ChiaveN	Negozi	Responsabile
1	Di tutto un po'	Bianchi
2	Non solo pane	Rossi

Tabella 3: Situazione il primo gennaio 2002, il responsabile di un negozio è cambiato

ChiaveN	Negozi	Responsabile
1	Di tutto un po'	Bianchi
2	Non solo pane	Verdi

Questa situazione può venire gestita attraverso una tavola relazionale fatta in questo modo:

Tabella 4: Tabella relazionale che gestisce l'evoluzione

ChiaveN	Negozio	Responsabile	da	a	master
1	Di tutto un po'	Bianchi	01/05/1999	--	1
2	Non solo pane	Rossi	02/04/2000	31/12/2001	2
3	Non solo pane	Verdi	01/01/2002	--	2

La chiave della tabella identifica non tanto un negozio quanto la sua situazione in un dato momento e sono presenti due campi data che specificano la validità della situazione: i record attualmente validi hanno il campo di fine validità vuoto. Il campo master permette di sapere quali campi si riferiscono effettivamente allo stesso negozio (potrebbe cambiare anche il nome).

Tornando all'implementazione dell'evoluzione temporale nel Syllabus si può notare che la sostituzione di un umeaning con un altro dovuta ad un cambiamento normativo presenta un problema ed una possibile soluzione allo scenario *Oggi o ieri*: si vuole conservare un riferimento alla situazione passata, quindi deve essere possibile visualizzare un umeaning scaduto e recuperare la struttura dell'ontologia com'era quando esso era valido, ma avere anche l'informazione che questo è scaduto e sapere da chi è stato sostituito.

4.3.2 Evoluzione mediante relazioni ontologiche

Per poter conservare una situazione storica nello scenario oggi o ieri sono necessarie alcune informazioni per poter ricostruire l'evoluzione temporale: le informazioni relative alla descrizione della gerarchia in un dato momento, la data di inizio validità, la data di fine validità ed una relazione verso la situazione che la nuova sostituisce.

Esaminando la struttura del database utilizzato dal Syllabus riportata in figura 2.7 possiamo notare che ogni umeaning presenta una data di creazione ed essi sono collegati fra di loro mediante l'ontologia (nel database rappresentata dalle tabelle *european_umean_to_umean* e *national_umean_to_umean*).

È stato quindi deciso di implementare la sostituzione di umeaning mediante una relazione ontologica, che verrà chiamata *REPLACED_BY*: l'introduzione di nuove relazioni di questo tipo causerà l'invalidazione dell'umeaning figlio, il cui identificativo verrà quindi

inserito in una nuova tabella dove viene associato con la data di scadenza: abbiamo quindi tutti i dati necessari per poter conoscere la situazione in un dato momento, ad esempio recuperare la lista di tutti gli umeaning che erano validi una certa data.

In figura 4.3 è riportata la struttura delle tabelle che gestiscono gli umeaning e la loro sostituzione dovuta al cambiamento normativo.

Si può notare come anche in questo caso si mantiene la struttura simmetrica tra la struttura di umeaning di livello nazionale ed europeo.

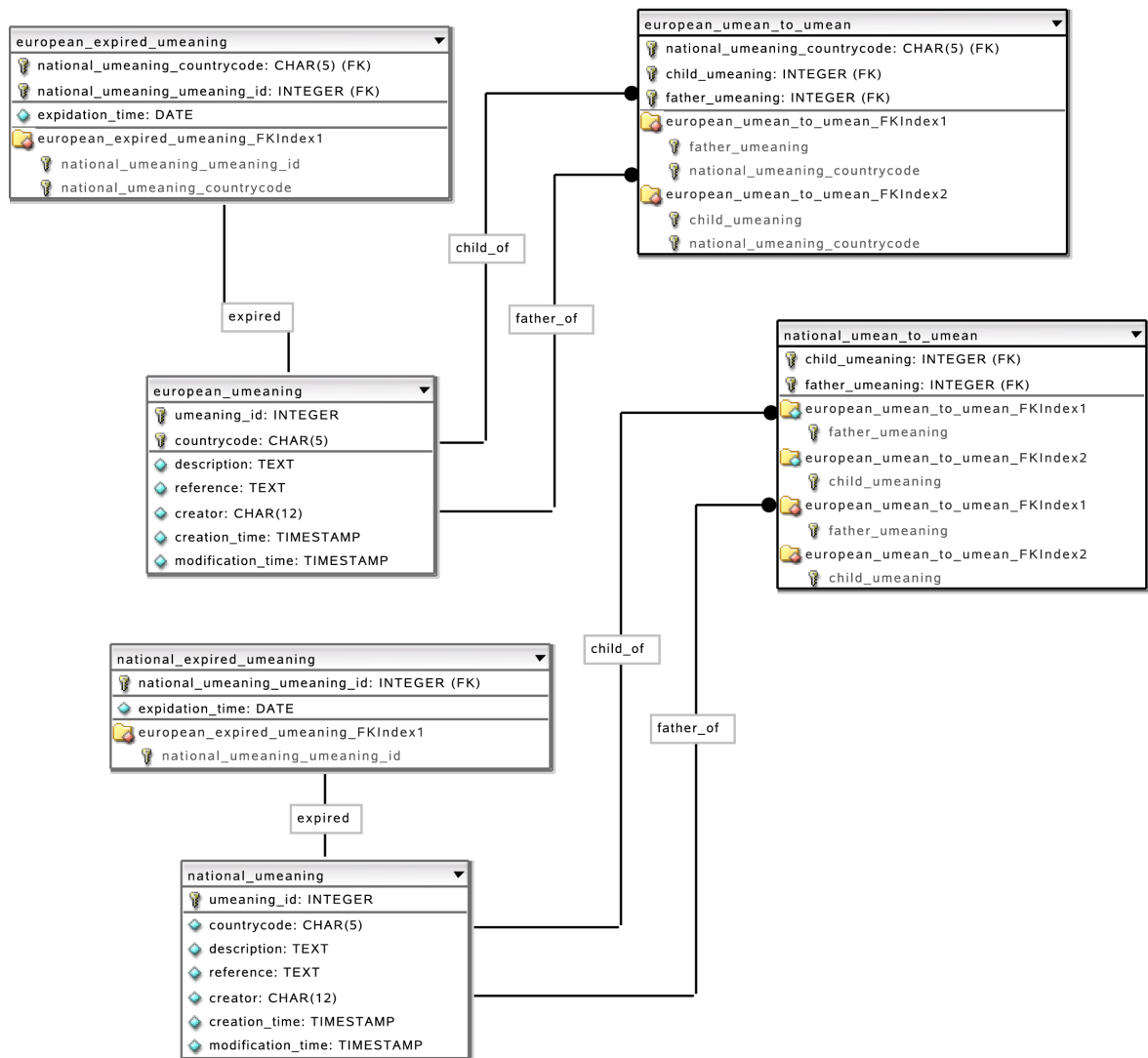


Figura 4.3: Struttura della relazione umeaning con supporto alla sostituzione

4.4 Modifiche nell'interfaccia

L'espansione e la modifica del sistema richiede alcune modifiche nel comportamento dell'interfaccia utente, inoltre altre modifiche sono state fatte dopo colloqui con giuristi per meglio adattarsi alle esigenze dell'utente finale e per dare una rappresentazione il più corretta possibile dei concetti giuridici.

4.4.1 Traduzione interfaccia

Data la natura multilingue del database del Syllabus, deve essere adatto ad un utenza internazionale, quindi l'interfaccia deve essere tradotta nella lingua naturale dell'utente tutto dove possibile, in particolare:

- Gli elementi dell'interfaccia utente non direttamente connessi con i dati come i link di navigazione, titoli delle tabelle e i campi delle form devono essere tradotti completamente.
- Durante la visualizzazione degli *umeaning* di livello europeo devono venire mostrate automaticamente le versioni nella medesima lingua dell'interfaccia quando disponibili.
- Gli *umeaning* di livello nazionale non possono venire tradotti, in quanto gran parte del significato di un testo giuridico è dato dal suo contesto linguistico e dai particolari termini utilizzati. Tuttavia il generico campo di descrizione dell'*umeaning* può essere tradotto nelle diverse lingue e deve venire automaticamente mostrata la traduzione nella medesima lingua dell'interfaccia utente (se disponibile).

Nella precedente architettura del Syllabus la traduzione dell'interfaccia era ottenuta attraverso una copia di tutti i template smarty per ogni lingua, ma si è subito rivelato di difficile mantenimento al crescere delle dimensioni dell'interfaccia grafica.

Si è scelto di reimplementare il sistema di traduzioni attraverso Gettext [44] come descritto nel paragrafo 6.4.

4.4.2 Comportamento della funzione di ricerca

Sono stati introdotti due concetti che cambiano il funzionamento della funzione di ricerca di termini e *umeaning*: il raggruppamento descritto al paragrafo 4.1.1 e il cambiamento normativo descritto al paragrafo 3.2.

4.4.2.1 Ricerca nei gruppi

Quando viene immessa una query di ricerca i risultati devono rispecchiare il livello di astrazione più elevato, devono quindi venire nascosti tutti gli *umeaning* che presentano una relazione di tipo GROUPEd con un altro *umeaning*, detto *group leader*.

Solo i *group leader* devono venire mostrati nei risultati di ricerca a meno che non venga specificato altrimenti dall'utente.

4.4.2.2 Ricerca di *umeaning* scaduti

Gli *umeaning* rappresentanti normative non più valide devono venire esclusi dai risultati di una ricerca, la quale deve rispecchiare unicamente la situazione attuale. Tuttavia l'utente può specificare una data di validità: in questo modo verranno restituiti anche gli *umeaning* validi fino e non oltre quella data.

4.5 Esportazione ontologia

Per meglio comprendere ed analizzare l'ontologia verrà implementata una funzionalità di esportazione nel formato standard OWL definito in [45] questo permetterà l'interoperabilità con altre applicazioni sia tradizionali che web-based.

4.5.1 Semantic Web

Le risorse presenti attualmente sul web sono rappresentate mediante linguaggio naturale, grafica, video elementi multimediali e anche il layout stesso della presentazione delle pagine è parte dell'informazione veicolata dalle pagine.

Questo tipo di presentazione dell'informazione è fatto in modo da essere processata e compresa facilmente dalle persone, ma sono di molto difficile comprensione per programmi software e sistemi automatici.

Questo può essere un problema per varie ragioni: persone con disabilità possono trovare molto difficile accedere ai contenuti web nella loro forma attuale, specialmente per quanto riguarda pagine ricche di contenuti multimediali. Inoltre i dati di ogni sito internet sono isolati gli uni dagli altri: è difficile combinare le informazioni provenienti da più siti se non si ha un linguaggio comune comprensibile ad un software.

Per risolvere questi problemi, è nato il progetto Semantic Web, guidato dal consorzio W3C [46] con la partecipazione di alcuni partner industriali.

Il Semantic Web vuole portare struttura al contenuto delle pagine web, in modo che agenti software possano automaticamente svolgere dei compiti di aggregazione dei dati ed utilizzo di servizi per conto degli utenti. Un esempio fatto spesso [47] è il seguente scenario: Lucy deve prenotare una visita da un medico specialista, formula quindi una query ad un agente software specificando di volere una visita da un particolare tipo di specialista: l'agente cerca i medici che possiedono la specializzazione voluta in quella zona e confronta gli orari di ricevimento di quelli che hanno ricevuto un rating più elevato con l'agenda degli impegni di Lucy e fissa un appuntamento in un orario libero oppure sposta gli appuntamenti precedenti che erano specificati come meno importanti.

Per rendere possibile uno scenario del genere è necessario che ogni sito web che fornisce un dato servizio presenti i dati attraverso un formato standard in modo che un software possa estrarre da essi parte della semantica del contenuto. A questo proposito il gruppo del Semantic Web ha definito una serie di formati aperti basati su XML [48] (eXtensible Markup Language) il quale consente di descrivere semanticamente le diverse parti di un documento.

Su di esso sono stati definiti il Resource Description Framework [49] (RDF), un particolare DTD XML che standardizza la definizione di relazioni tra informazioni utilizzando i principi della logica dei predicati. Sono inoltre in fase di studio estensioni a RDF e strumenti aggiuntivi come SPARQL [50], un linguaggio per l'interrogazione di ontologie RDF simile a SQL e OWL [45] (Web Ontology language), un'estensione a RDF che permette la definizione di ontologie più complete provviste di regole di inferenza dedotte dalle caratteristiche dei tipi di relazione.

4.5.2 RDF

RDF definito in [51] è un linguaggio basato su XML per rappresentare risorse nel web. Può anche essere utilizzato per rappresentare entità che possono essere identificate tramite il web, anche se non possono essere direttamente “contenute” in esso, ad esempio informazioni su beni materiali disponibili su negozi on-line.

RDF utilizza la logica dei predicati per definire le informazioni, cioè mediante *asserzioni* nella forma *soggetto, predicato e valore*, mentre i singoli oggetti sono definiti mediante Uniform Resource Identificators (URI) [52]. Il soggetto di un'asserzione RDF è una *resource*, identificata da una URI, il predicato è un'altra risorsa identificata da un'altra URI e il valore è una stringa unicode, una risorsa identificata da un'altra URI oppure un nodo vuoto. RDF è quindi un metodo di rappresentazione di grafi orientati che rappresentano semplici ontologie senza proprietà di inferenza.

Riportiamo un esempio contenuto in [51], che identifica le informazioni sui recapiti di uno degli autori:

```

1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
3     xmlns:contact="http://www.w3.org/2000/10/swap/pim/
  contact#">
4
5   <contact:Person
6     rdf:about="http://www.w3.org/People/EM/contact#me">
7     <contact:fullName>Eric Miller</contact:fullName>
8     <contact:mailbox rdf:resource="mailto:em@w3.org"/>
9     <contact:personalTitle>Dr.</contact:personalTitle>
10  </contact:Person>
11</rdf:RDF>

```

Le righe 2 e 3 definiscono i namespace XML [48] utilizzati nel resto del documento: d'ora in poi ogni tag “rdf” identificherà l'URI nella riga 2, quindi ad esempio “<rdf:about>” identificherà “http://www.w3.org/1999/02/22-rdf-syntax-ns#about”, allo stesso modo in riga 3 è definito il namespace “contact”.

La riga 5, assegnando un URI alla proprietà about, identifica il soggetto di un'asserzione RDF, a cui sono associati 3 predicati e quindi 3 valori, definiti nelle righe 6, 7 e 8.

L'asserzione di questo esempio quindi si può leggere come: “La persona identificata da http://www.w3.org/People/EM/contact#me *ha nome* Eric Miller, il suo indirizzo di posta elettronica è em@w3.org e il suo titolo è Dr.”

Possono anche essere definiti gruppi di oggetti, attraverso le TAG <rdf:Bag>, <rdf:Seq> e <rdf:Alt>. Una bag rappresenta un gruppo di oggetti con possibilità di duplicati e quando l'ordine dei membri non ha significato. Una sequenza (Seq) rappresenta un gruppo con possibilità di duplicati in cui l'ordine è fondamentale al significato. Un'alternativa (Alt) rappresenta un gruppo di oggetti che sono da considerarsi alternativi fra di essi, ad esempio potrebbe essere utilizzato per descrivere il titolo di un libro tradotto in diverse lingue.

Un limite dei gruppi precedentemente descritti è che non è possibile definire un gruppo chiuso, cioè dire “questi sono *tutti* i membri del gruppo”. Per ottenere questo si utilizzano le *collezioni*, identificate dall'attributo rdf:parseType="Collection" in una qualsiasi tag che contiene un gruppo di elementi che rappresentano i membri di quella collezione.

4.5.3 OWL

OWL definito in [45] è un linguaggio basato su RDF il cui intento è di fornire proprietà semantiche più avanzate alle *risorse* RDF soprattutto che siano processabili

automaticamente, di dare quindi un diverso significato semantico ai diversi tipi di relazioni ontologiche fra le differenti risorse. OWL è stato progettato per soddisfare quindici obiettivi:

1. Il linguaggio deve supportare differenti livelli di complessità per la definizione di ontologie.
2. Il linguaggio deve supportare la specifica di valori predefiniti per le proprietà, le modalità con cui trattare i valori di default non è specificata dal linguaggio.
3. Data la dimensione e la velocità di cambiamento del Web, l'assunzione cosiddetta del *mondo chiuso* è inappropriata (dove qualsiasi cosa che non può venire inferita dai dati esistenti è assunta come falsa). Tuttavia in alcuni casi l'informazione che una particolare ontologia sia un mondo chiuso può essere utile. Il linguaggio deve poter quindi essere in grado di definire una particolare ontologia come completa oppure no.
4. Il linguaggio deve essere in grado di definire dei range di valori per le proprietà.
5. Il linguaggio deve supportare la composizione di proprietà negli statement riguardo classi e proprietà. Ad esempio una proprietà chiamata *uncleOf* deve essere equivalente alla composizione di *fatherOf* e *brotherOf*.
6. Il linguaggio deve essere computazionalmente decidibile.
7. Il linguaggio deve supportare la possibilità di mostrare anche solo una parte di un'ontologia
8. Il linguaggio deve avere la possibilità di creare visualizzazioni di un'ontologia in cui è possibile assegnare nomi diversi sottoinsiemi dell'ontologia.
9. Deve essere possibile firmare digitalmente ontologie attraverso la W3C XML Digital Signature specification
10. Il linguaggio deve supportare l'utilizzo di funzioni primitive aritmetiche.
11. Il linguaggio deve supportare la concatenazione di stringhe ed un semplice pattern matching.
12. Il linguaggio deve poter supportare la possibilità di aggregare informazioni in un modo simile alla clausola GROUP BY di SQL, devono essere possibili conti, somme, medie aritmetiche e altre operazioni.
13. Il linguaggio deve poter supportare l'invocazione di codice eseguibile, ad esempio Java per valutare criteri complessi.

14. Non vengono fatte assunzioni di unicità dei nomi: identificatori distinti non è detto che si riferiscano a differenti individui.
15. Il linguaggio deve supportare la definizione e l'uso di tipi di dati complessi e strutturati.

Rispetto alle relazioni soggetto-predicato-valore di RDF, OWL deve supportare alcuni vincoli aggiuntivi su tali relazioni:

- Vincoli di cardinalità: definire ad esempio che una persona può avere un solo padre biologico.
- Specificare che una data proprietà sia transitiva: ad esempio se B è antenato di A e C è antenato di B, allora C è antenato di A.
- Specificare che una data proprietà dia un identificatore univoco, ad esempio la chiave primaria di una tabella.
- Specificare che due risorse identificate da URI differenti rappresentano in realtà la medesima risorsa.
- Specificare che la cardinalità di una proprietà dipende dalla classe di risorse a cui la proprietà è applicata: ad esempio per la risorsa *squadraDiCalcio* la proprietà *giocatori* avrà esattamente 11 valori, mentre per la risorsa *squadraDiBasket* avrà soltanto 5 valori.
- La possibilità di descrivere nuove classi in termini di combinazioni di altre classi, come unioni ed intersezioni, oppure dire che due classi sono separate, cioè che nessuna risorsa è istanza di entrambe le classi.

OWL offre tre sottolinguaggi con differenti livelli di espressività, per far fronte a diversi vincoli di completezza e risorse di calcolo:

- OWL Lite: utilizzato quando occorre solo una gerarchia di classificazione e vincoli semplici, ad esempio supporta soltanto vincoli di cardinalità 0 e 1.
- OWL DL (Description Logic): Offre la massima espressività che si può ottenere mantenendo la completezza computazionale, cioè tutti i vincoli sono sicuramente computabili in un tempo finito. Si possono utilizzare tutti i costrutti ma sotto certi limiti, ad esempio una classe non può essere istanza di un'altra classe.
- OWL Full: Offre la massima espressività ma non ci sono garanzie di computabilità.

4.6 Specifica formale dei requisiti

4.6.1 INTRODUCTION

Seguiremo lo standard IEEE per un documento di specifica dei requisiti come definito da [53].

4.6.1.1 Purpose

Lo scopo di questo documento è di descrivere i requisiti delle modifiche al sistema per la gestione di un dizionario ontologico giuridico denominato SSyllabus, descritto in [1].

4.6.1.2 Definitions, Acronyms, Abbreviations

- Dizionario Concettuale: dizionario in grado di definire non solo singoli termini, quanto piuttosto una serie di concetti complessi e articolati.
- Umeaning: unità volta a definire il significato di un termine giuridico.
- Termine: un termine del dizionario, cioè una parola oppure una frase, ad esempio “contratto”.
- Lingua giuridica: lingua e nazionalità a cui appartiene il testo giuridico.
- Livello: i termini e gli umeaning sono definiti di livello europeo se provengono dal testo di una direttiva comunitaria, di livello nazionale se invece provengono dal un testo giuridico di uno stato membro.
- Ontologia: specifica formale di una concettualizzazione, cioè una visione astratta e semplificata del mondo che si vuole rappresentare. Essa è vista come un insieme di oggetti e le relazioni fra essi.
- Cambiamento normativo: situazione in cui concetti definiti in vecchie normative vengono resi obsoleti e sostituiti da concetti presenti in normative più recenti
- Umeaning scaduto: un umeaning reso non più valido da una nuova norma: viene sostituito da un altro.
- Astrazione dei concetti: diverso livello di dettaglio in cui i concetti possono essere visti.
- Umeaning Group Leader: Un umeaning di tipo group leader rappresenta un concetto ad un livello maggiore di astrazione, collegato mediante una relazione ontologica ai membri del gruppo che rappresentano concetti più dettagliati, di solito nascosti.

- Normativa: il testo completo di una normativa giuridica: può essere una direttiva europea oppure una legge nazionale.

4.6.1.3 Notational Conventions

I requisiti verranno specificati in questo documento utilizzando il linguaggio naturale.

4.6.2 OVERALL DESCRIPTION

4.6.2.1 Product perspective

Syllabus è un sistema basato su tecnologia web, l'utente interagisce con esso mediante un qualsiasi web-browser, supportare l'accesso e modifica dei dati in maniera altamente concorrente, mantenendone quindi l'integrità.

L'espansione del sistema comprenderà modifiche all'ontologia per gestire livelli diversi di astrazione e il cambiamento normativo. Quindi verrà implementato un archivio di testi giuridici con funzionalità di ricerca, avendo la possibilità di collegare passi delle normative agli umeaning.

4.6.2.2 Product functions overview

Il sistema è un dizionario giuridico che collega termini ai loro significati, denominati "umeaning". Un umeaning è attualmente composto da una descrizione, un campo testuale che riporta passi di normative in cui il termine appare ed eventuali file allegati.

Gli umeaning sono separati in due livelli: il livello europeo e il livello nazionale. Siccome le direttive europee vengono tradotte nelle lingue degli stati membri gli umeaning associati ai concetti definiti da esse sono anch'essi tradotti nelle diverse lingue, mentre gli umeaning di livello nazionale sono definiti unicamente nella lingua utilizzata dalla nazione in questione.

Il recepimento di direttive europee porta alla creazione di normative che definiscono il concetto nell'ambito dell'ordinamento giuridico nazionale, si ha quindi una corrispondenza tra gli umeaning nazionali e quelli europei. Si dice in questo caso che l'umeaning nazionale *implementa* l'umeaning europeo. Il sistema fornisce un'interfaccia di ricerca per definire facilmente questo tipo di relazioni tra umeaning nazionali ed europei.

Senza alterare significativamente l'architettura del sistema dovrà essere aggiunto un archivio di testi giuridici completi di ambedue i livelli.

Dato un nuovo testo giuridico che ridefinisce concetti definiti in testi precedenti deve essere possibile definire nuovi *umeaning* che sostituiscono i significati definiti nel testo vecchio. Questa sostituzione non deve portare alla perdita dei vecchi dati e devono essere chiare nell'interfaccia le relazioni di sostituzione tra i diversi *umeaning*.

Nei risultati di una ricerca non devono comparire gli *umeaning* scaduti ma solo le versioni correnti, a meno che l'utente non specifichi una data di validità, in questo caso il sistema restituirà la situazione valida nella data specificata, includendo nei risultati gli *umeaning* scaduti dopo la data specificata.

L'utente deve inoltre poter raggruppare più *umeaning* sotto un *umeaning* creato per l'occasione, detto *group leader*, il quale rappresenta un concetto più astratto rispetto agli *umeaning* che fanno parte di quel gruppo che sono concetti più dettagliati.

Nei risultati di una ricerca gli *umeaning* che fanno parte di un gruppo vengono esclusi: solo il *group leader* è mostrato (se rilevante rispetto alla query) a meno che l'utente non specifichi esplicitamente di includere i membri del gruppo nei risultati.

Data l'utenza internazionale del Syllabus, il sistema per la traduzione dell'interfaccia deve essere rifatto per permettere una traduzione completa dell'interfaccia e una più veloce propagazione a tutte le lingue quando un elemento di essa o un messaggio vengono cambiati.

4.6.2.3 User Characteristics

Il sistema è utilizzato da due tipi di utenti.

Gli utenti generali del sistema saranno persone con conoscenze di giurisprudenza come giuristi e avvocati, ai quali non deve essere richiesta nessuna conoscenza informatica specifica.

Saranno anche presenti amministratori di sistema, delegati alla creazione di utenti, manutenzione e backup del sistema; a questa categoria è richiesta una più dettagliata conoscenza sul funzionamento del sistema.

4.6.3 SPECIFIC REQUIREMENTS

4.6.3.1 *Inputs and Outputs*

Il sistema riceve tutti i suoi input unicamente attraverso il metodo http post [54] (quindi presentato all'utente sotto forma di form html) oppure codificate nell'URL [52].

L'output del sistema è invece rappresentato da pagine html visualizzabili mediante il browser oppure mediante file da scaricarsi sul computer per poter essere visualizzati, ad esempio file pdf e immagini. È anche da considerarsi output il file di log di tutte le operazioni eseguite dall'utente, implementato in una tabella del database e visibile solo agli utenti amministratori.

Qui di seguito sono riportate le descrizioni dei campi delle form che devono venire aggiunte al sistema.

Inserimento Umeaning

Attualmente il sistema presenta una form di inserimento degli umeaning in cui è possibile inserire il nome del termine, la descrizione e i riferimenti. Eventuali allegati verranno inseriti in un secondo tempo. Allo stesso modo relazioni ontologiche e associazione tra umeaning nazionali ed europei verranno inserite in un secondo momento mediante un interfaccia di ricerca.

L'input in questa fase dovrà essere modificato in questo modo:

- Termine: casella di testo in cui inserire il termine descritto dall'umeaning che si sta creando.
- Lingua giuridica: combobox usata per selezionare la lingua giuridica a cui l'umeaning appartiene (es. “italiano”, “tedesco” oppure “tutte”).
- Livello: combobox da cui è possibile scegliere “livello nazionale” oppure “livello europeo”.
- Descrizione: casella di testo in cui inserire la descrizione del termine; sarà solitamente un breve passo di una legge in cui il termine appare.
- Note: nuovo campo che riporta eventuali sentenze riguardanti il concetto espresso nell'umeaning.

I riferimenti non verranno più inseriti da questa form, ma sarà presente un'interfaccia di ricerca apposita descritta nel paragrafo 4.6.3.1.

Per quanto riguarda gli umeaning di livello nazionale deve essere possibile inserire una traduzione in un secondo momento nella medesima lingua dell'interfaccia utente.

L'unico input sarà il testo della descrizione dell'umeaning, la lingua e l'identificativo dell'umeaning saranno input impliciti.

Inserimento e cancellazione testi giuridici

- Lingua giuridica: combobox usata per selezionare la lingua giuridica a cui l'umeaning appartiene (es. “italiano”, “tedesco” oppure “tutte”).
- Livello: combobox da cui è possibile scegliere “livello nazionale” oppure “livello europeo”.
- Codice: casella di testo. Identificativo univoco per il testo giuridico: ogni testo giuridico presenta una numerazione univoca che varia da paese a paese. Per quanto riguarda l'ordinamento europeo l'identificativo è composto da anno/numero progressivo/nome corrente dell'unione europea(negli anni è passato da CEE a CE ed infine a UE) ad esempio 93/13/CEE è la tredicesima direttiva emessa nel 1993.
- Data emissione: data inserita attraverso tre combobox per l'inserimento di giorno, mese ed anno. Rappresenta la data in cui il testo è stato emesso dal parlamento, sarà quindi solitamente precedente alla data di inserimento nel database
- Testo: casella di testo multi linea in cui viene inserito il testo completo della normativa

Inserimento e cancellazione riferimenti

L'inserimento di nuovi riferimenti deve avvenire attraverso un'interfaccia di ricerca, in cui l'utente può effettuare una query sul database dei testi giuridici, dopo di che selezionare con il mouse il pezzo del testo che interessa. L'input e output dell'operazione si svolge in due fasi:

Prima fase:

Input: Codice del testo oppure uno o più termini che appaiono nel testo. Il livello, la lingua giuridica e l'identificativo dell'umeaning sono input impliciti.

Output: Elenco di normative che soddisfano la query immessa.

Seconda fase:

Input: pezzo di testo della normativa evidenziato con il mouse, l'identificativo dell'umeaning è un input implicito.

Output: conferma che l'operazione è andata a buon fine.

Raggruppamento di umeaning

L'operazione di raggruppamento di umeaning avviene in modo simile alla modifica dell'ontologia: partendo da un umeaning se ne seleziona uno nuovo attraverso un'interfaccia di ricerca simile alla classica ricerca di umeaning.

Input: gli identificativi di due umeaning

Output: la visualizzazione dell'umeaning *group leader*, che viene creato se non precedentemente esistente.

Cambiamento normativo e sostituzione umeaning

Nell'ontologia il cambiamento normativo si riflette attraverso due umeaning fra loro in relazione REPLACED_BY, l'umeaning figlio rappresenta il concetto indicato dalla vecchia normativa, l'umeaning padre il concetto indicato dalla normativa più recente, che quindi ridefinisce il concetto.

Ricerca umeaning

Input:

- Id umeaning: per ricercare un umeaning corrispondente ad un dato id: la numerazione degli ueaning effettuata dal database verrà quindi esposta all'utente.
- Termine: casella di testo in cui inserire il termine da cercare oppure parte di esso (es. "diritto di recesso" oppure "recesso").
- Lingua giuridica: combobox usata per selezionare la lingua giuridica a cui limitare la ricerca (es. "italiano", "tedesco" oppure "qualsiasi").
- Livello: combobox da cui è possibile scegliere "livello nazionale", "livello europeo" oppure "qualsiasi".
- Descrizione: casella di testo per ricercare i termini in base alla loro descrizione.
- Riferimenti: casella di testo per ricercare i termini in base ai loro riferimenti giuridici.

- Data di validità: data immessa mediante tre combobox (giorno, mese e anno) che richiede l'inclusione nei risultati degli umeaning che erano ancora validi fino alla data indicata (anche se attualmente non più validi)
- Mostra umeaning raggruppati: checkbox. Se selezionata i componenti di un gruppo vengono mostrati nei risultati, altrimenti solamente il group leader,

Output: la lista degli umeaning che corrispondono ai parametri di ricerca.

Ricerca testi legali

Input:

- Lingua giuridica: combobox usata per selezionare la lingua giuridica a cui limitare la ricerca (es. "italiano", "tedesco" oppure "qualsiasi").
- Livello: combobox da cui è possibile scegliere "livello nazionale", "livello europeo" oppure "qualsiasi".
- Codice: casella di testo per specificare l'identificativo univoco per il testo giuridico.
- Testo: un o più termini che si vogliono cercare all'interno del testo.

Output: la lista dei testi che corrispondono ai parametri di ricerca.

4.6.3.2 Functional Requirements

Inserimento Umeaning

Per poter inserire nuovi termini e umeaning i seguenti vincoli devono essere soddisfatti:

- L'utente deve riempire tutti i campi definiti come obbligatori: lingua giuridica, livello, termine e descrizione
- per poter inserire umeaning è necessario autenticarsi come un utente che possiede esplicitamente questa capacità

Gli umeaning di livello europeo possono per un unico id avere più traduzioni nelle varie lingue giuridiche disponibili, mentre per gli umeaning di livello nazionale unicamente il campo descrizione viene tradotto.

L'inserimento di nuovo umeaning può avvenire in quattro differenti scenari:

1. L'utente crea una nuova coppia termine/umeaning.
2. Viene aggiunto un nuovo umeaning per un termine esistente.

3. Viene aggiunta una nuova traduzione di un termine europeo.

4. Viene creato un nuovo gruppo. Il nuovo umeaning auto generato sarà il group leader.

Input: termine (obbligatorio solo nel caso 1), lingua giuridica (obbligatorio), livello (obbligatorio), descrizione (obbligatorio), note definiti nel paragrafo 4.6.3.1.

Output: tutti i campi dell'umeaning appena inserito oppure un messaggio di errore, tra cui: “accesso negato” e “non tutti i campi obbligatori sono stati compilati”

Inserimento e cancellazione testi giuridici

Per potere inserire o cancellare testi giuridici le seguenti condizioni devono essere soddisfatte:

- l'utente deve essere autenticato e avere esplicitamente il permesso di inserimento testi
- Tutti i campi specificati come obbligatori devono essere compilati

Input (inserimento): Lingua giuridica (obbligatorio), livello (obbligatorio), Codice (obbligatorio), data emissione (obbligatorio), testo completo (obbligatorio). Sono tutti campi che verranno spediti tramite una POST Http.

Output (inserimento): Il testo completo viene mostrato.

Input (cancellazione): Codice (obbligatorio), lingua giuridica (obbligatorio), livello (obbligatorio)

Output (cancellazione): il browser verrà rediretto alla pagina di ricerca dei testi, verrà mostrato un messaggio di successo oppure un messaggio di accesso negato.

Quando un testo viene cancellato ogni eventuale riferimento ad esso negli umeaning viene anch'esso cancellato.

Inserimento e cancellazione riferimenti

L'inserimento di riferimenti può essere fatto unicamente da un utente autenticato con il permesso di inserzione umeaning.

L'inserimento è fatto in due passi:

1. Da un umeaning si seleziona la funzione di inserimento riferimenti, il browser viene diretto sulla pagina di ricerca di testi completi (vedere paragrafo 4.6.3.2) a differenza di una ricerca normale il livello e la lingua giuridica sono fissi, come anche l'id dell'umeaning.
2. Scelto un testo esso viene visualizzato, l'utente seleziona con il mouse il pezzo di interesse che verrà inserito nei riferimenti.

Raggruppamento di umeaning

Per poter creare, modificare o cancellare un gruppo l'utente deve avere il permesso di modifica dell'ontologia.

La creazione dei gruppi avviene mediante un'interfaccia di ricerca: partendo da un componente del gruppo se ne ricerca un altro con la medesima interfaccia di ricerca degli umeaning, in cui lingua giuridica e livello sono fissati. Scelto il secondo umeaning del gruppo le seguenti azioni devono essere compiute dal sistema:

- Se nessuno dei due umeaning faceva parte di un gruppo crea un nuovo group leader e entrambi gli umeaning vengono messi in relazione GROUPED con il padre appena creato
- Se uno dei due umeaning fa già parte di un gruppo l'altro umeaning viene messo in relazione con il group leader, in questo modo si possono avere gruppi con più di due membri
- Se i due umeaning fanno già parte dello stesso gruppo non viene compiuta nessuna operazione
- Se i due umeaning fanno parte di due gruppi diversi non viene compiuta nessuna operazione e viene restituito un messaggio di errore

Cambiamento normativo e sostituzione umeaning

Per poter creare, modificare o cancellare relazioni di sostituzione derivanti dal cambiamento normativo l'utente deve avere il permesso di modifica dell'ontologia.

La sostituzione di umeaning derivante dal cambiamento normativo viene gestita attraverso un'interfaccia di ricerca simile alla ricerca degli umeaning e modifica dell'ontologia, si parte dall'umeaning che ne sostituisce uno vecchio (precedentemente creato) e si cerca

l'umeaning sostituito da questo; quando questo viene scelto si crea una nuova relazione di tipo REPLACED_BY tra l'umeaning padre e il figlio.

Input: umeaning padre, umeaning figlio

Output: viene mostrato l'umeaning padre ed eventualmente un errore di accesso negato.

Ricerca umeaning

Il sistema supporta la ricerca dei termini e umeaning inseriti. Quest'operazione solitamente è disponibile sia per utenti autenticati che per utenti anonimi, tuttavia l'amministratore può configurare il sistema in modo che consenta la navigazione solo più ad utenti autenticati.

Input: il termine da ricercare o parte di esso, oppure parte della sua descrizione. La ricerca inoltre deve poter essere eseguita su descrizione, riferimenti e note degli umeaning ad esso associati. Si può specificare se includere gli umeaning facenti parte di un gruppo o gli umeaning scaduti. È possibile specificare una data di validità per gli umeaning che verranno inclusi nei risultati.

L'input fornito dall'utente nei campi termine, descrizione, note e riferimenti deve venire processato da uno stemmer nella lingua corretta, in modo da effettuare una ricerca non sulle parole ma sulle loro radici: ad esempio la query “contratto” deve poter

Output: elenco dei termini e umeaning che soddisfano i criteri di ricerca, un appropriato messaggio di errore se non sono stati trovati termini che soddisfano i criteri, oppure un messaggio di accesso negato se la navigazione anonima è stata disabilitata.

Gli umeaning sostituiti da altri a causa di un cambiamento normativo devono venire esclusi dalla ricerca, a meno che sia stata specificata una data di validità che include tali umeaning scaduti.

Il comportamento predefinito della funzione di ricerca deve essere l'esclusione di umeaning raggruppati; solo i group leader devono venire mostrati.

Ricerca testi legali

Per poter effettuare una ricerca nel database di testi legali l'utente deve avere il permesso di navigazione nel sistema.

Input: Lingua giuridica, livello, codice e testo: tutti i campi sono opzionali, deve esserne presente almeno uno per effettuare la ricerca.

L'input fornito dall'utente nel campo testo deve venire pre-processato da uno stemmer e il match deve venire fatto con i termini conservati nell'indice su cui viene effettuato lo stemming anche su essi.

Output: lista dei testi giuridici che soddisfano la query.

Quando un testo legale è mostrato all'utente i termini ricercati devono essere evidenziati.

Esportazione Ontologia

Deve essere possibile esportare l'intera ontologia in formato OWL, In questo modo sarà possibile esaminare l'ontologia in altri tool quali Protegé [55].

4.6.3.3 External Interface Requirements

L'interfaccia web del sistema esistente come ogni parte che verrà modificata deve seguire la specifica w3c xhtml 1.0 strict [56] e deve venire visualizzata correttamente sui principali browser web, in particolare:

- Internet Explorer versione 6.0 o superiore
- Mozilla Firefox versione 2.0 o superiore
- Apple Safari versione 2.0 o superiore
- Opera versione 9.0 o superiore.

Inoltre la dimensione delle pagine html deve essere contenuta per rendere possibile la navigazione anche attraverso connessioni internet a bassa velocità.

L'ontologia esportata deve essere un documento sintatticamente valido in formato OWL definito in [45].

4.6.3.4 Design Constraints

Il sistema, richiederà la seguente piattaforma software:

- Apache versione 1.3 o superiore.
- PHP versione 5.0 o superiore.
- PostgreSQL versione 8.0 o superiore.
- AT&T GraphViz versione 2.2 o superiore.

- Estensione Php Image/GraphViz versione 1.1 o superiore.
- Estensione Php per lo stemming multilingua [57].
- Pacchetto software per Php Smarty versione 2.6 o superiore.
- Sistema operativo unix (in particolare Linux e Sun Solaris) oppure Microsoft Windows 2000 e successivi. Attualmente il sistema è in produzione in un ambiente Windows 2000.

Rispetto al sistema esistente è stata introdotta la dipendenza rispetto all'estensione Php per lo stemming.

Capitolo 5 - Progettazione

La progettazione avverrà in due distinte fasi: verranno prima analizzati i cambiamenti da effettuare all'architettura delle classi appartenenti al livello della logica per rispecchiare le funzionalità richieste esaminate nel capitolo 4, verrà poi analizzato il modo in cui i cambiamenti al livello della logica si riflettono sul livello del database.

5.1 Architettura attuale delle classi

La struttura attuale delle classi è mostrata in figura 2.5 a pagina 12, come detto il sistema segue la classica architettura three-tier che fornisce un buon livello di astrazione ed ottimizzazione della gestione delle risorse nelle applicazioni client-server [58], come illustrato in figura 5.1.

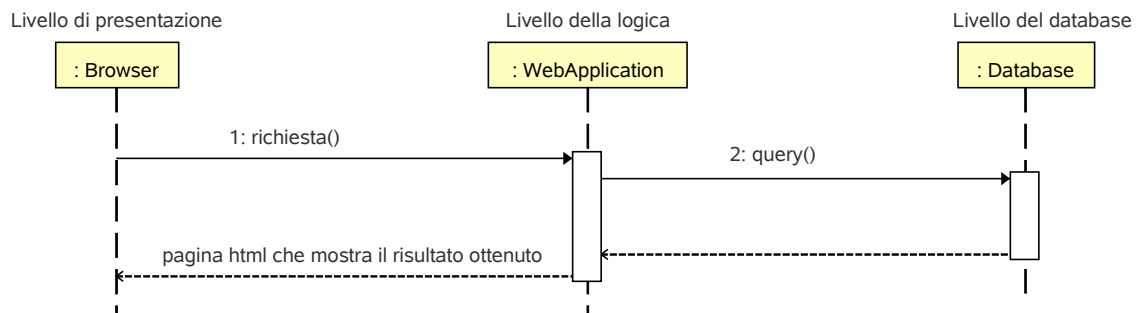


Figura 5.1 architettura di un'applicazione client/server three tier

Nello schema di figura 2.5 il livello della logica è connesso al livello di presentazione mediante la classi di tipo *ActionDispatcher* (che fanno parte del sotto-livello *Interaction Handler*) utilizzando la classe *Smarty* per delegare la costruzione della presentazione finale ed al livello del database mediante la classe *Database*.

In figura 5.2 è mostrata la nuova atchitettura dei package del sistema, in cui è stato aggiunto il package *fullText*.

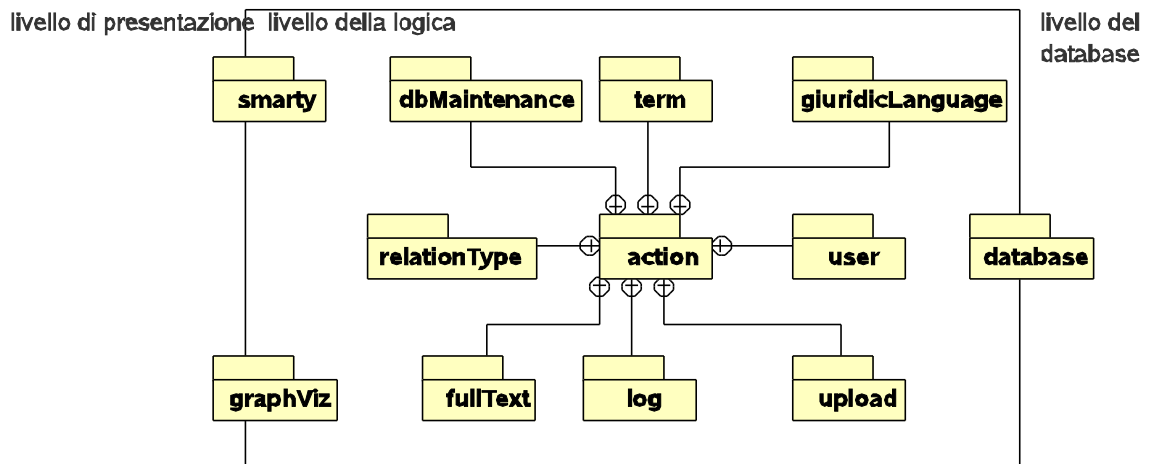


Figura 5.2: Package presenti nel sistema

5.2 Design di dettaglio delle modifiche necessarie

Verranno analizzate le modifiche alle classi esistenti e le nuove classi che dovranno essere progettate in maniera separata per ognuno dei requisiti definiti nel paragrafo 4.6.

5.2.1 Gestione livelli di astrazione

L'introduzione di diversi livelli di astrazione rende necessaria la modifica della coppia di classi Term e TermActionDispatcher atte alla gestione di Termini e umeaning.

TermActionDispatcher deve fornire una nuova azione denominata group() la quale avendo in input due umeaning crea un nuovo group leader e mette i due umeaning in relazione GROUPED. Se uno dei due umeaning possiede un group leader aggiunge l'altro a tale gruppo, se i due umeaning hanno group leader diversi restituisce un messaggio di errore.

L'astrazione group() utilizza per effettuare l'operazione di raggruppamento una nuova funzione in Term denominata insertGroup che riceve come parametri i due umeaning.

La funzione di ricerca in Term viene modificata in modo da non più includere gli umeaning in relazione GROUPED con un group leader a meno che il parametro booleano aggiuntivo *showGrouped* non sia vero.

La funzione loadTerm di TermActionDispatcher deve mostrare il group leader collegato tutti i termini associati ai suoi figli come anche tutti i riferimenti dei figli. Per fare questo verranno utilizzate alcune funzioni di utilità di Term come getGroupLeader() che dato un

umeaning oppure un termine restituisce il group leader se questo fa parte di un gruppo e getGroupMembers(), che dato un group leader restituisce la lista dei membri del gruppo.

5.2.2 Gestione dell'evoluzione temporale

TermActionDispatcher dovrà fornire la nuova azione insertReplacement() che gestirà l'inserimento di nuove relazioni ontologiche di tipo REPLACED_BY, il comportamento è del tutto simile alla funzione insertChild().

La funzione di ricerca in dipende da un nuovo parametro: la data di validità. Il valore predefinito di tale data sarà la data odierna, quindi il suo comportamento sarà mostrare la situazione giuridica attuale. Se l'input fornito dall'utente è una data passata il sistema dovrà includere nei risultati anche gli umeaning validi nella data immessa ma non più in data odierna.

La cancellazione di relazioni REPLACED_BY è compiuta dalla funzione deleteRelation(), non ci sono differenze di comportamento rispetto alla cancellazione di un qualsiasi tipo di relazione ontologica.

5.2.3 Archivio di testi giuridici

Il database di testi giuridici è una funzionalità del tutto nuova nel sistema: verrà implementata attraverso una nuova coppia di classi: FullTextActionDispatcher che gestirà il livello di *User Interaction Handler* e FullText che gestirà la *Business Logic* dell'archivio.

L'User Interaction Handler espone le seguenti funzionalità:

- insertFullText(): inserisce un nuovo testo giuridico. Ha bisogno (oltre al testo stesso) della lingua giuridica, del codice univoco del testo giuridico (ad esempio 90/314/CEE) e del livello (nazionale oppure europeo)
- deleteFullText(): cancella un testo giuridico inserito. Ha bisogno del codice, del livello e nel caso che il livello sia europeo della lingua giuridica.
- LoadFullText(): recupera un testo giuridico dal database e lo mostra all'utente. Ha bisogno del codice, del livello e anche in questo caso se il livello è europeo della lingua giuridica. Inoltre può evidenziare alcuni termini specificati dall'utente all'interno del testo, questo viene utilizzato nella funzione di ricerca, in cui i termini ricercati verranno evidenziati quando viene caricato uno dei risultati.
- searchFullText(): effettua una ricerca all'interno del database. L'utente specifica il livello, la lingua giuridica, il codice oppure una parte del testo, sono tutti campi opzionali, ma deve essere presente un campo fra il codice e il testo. Se vengono trovati dei risultati viene presentata all'utente una tabella riassuntiva dei risultati i

cui campi sono il codice identificativo, il livello, la lingua giuridica ed un estratto del testo.

- `insertReference()`: inserisce il riferimento ad un testo giuridico all'interno di un `umeaning`. Ha bisogno dell'identificativo dell'`umeaning`, del livello, della lingua giuridica, del codice del testo giuridico, del carattere in cui il riferimento inizia all'interno del testo e della sua lunghezza.
- `DeleteReference()`: cancella un riferimento dal database. Ha bisogno dell'`umeaning`, del livello, della lingua giuridica, del codice del testo giuridico, del carattere in cui il riferimento inizia all'interno del testo.
- `SearchReference()`: condivide lo stesso comportamento con `searchFullText()`. Richiede l'identificativo di un `umeaning` come parametro aggiuntivo, in quanto questa funzione rappresenta la ricerca di un nuovo riferimento da aggiungere ad un `umeaning`.

La ricerca è fatta mediante l'ausilio di un indice invertito descritto nel paragrafo 4.2.1.1 e di uno stemmer, descritto nel paragrafo 4.2.2.

Non è prevista una funzione di modifica dei testi giuridici in quanto quando essi entrano in vigore (ad esempio nell'ordinamento italiano dopo la pubblicazione sulla Gazzetta Ufficiale) non verranno più modificati. Successivi testi giuridici possono modificare i concetti descritti dai testi precedenti, e questo viene gestito mediante l'evoluzione temporale attraverso le ontologie, descritta nel paragrafo 3.2.

5.2.3.1 Flusso di esecuzione

Qui di seguito è riportato l'algoritmo in pseudo codice che il programma esegue quando un nuovo documento viene inserito:

```

1 IF (codice, lingua, livello, data oppure testo== NULL) THEN
2   print(dati mancanti)
3 ELSE
4   inserisci nel database il testo, codice e lingua
5   nella tabella del livello appropriato
6   FOR EACH parola IN testo DO
7     IF parola NOT IN function_word_list THEN
8       inserisci stem(parola, lingua),
9       identificativo documento e posizione
10      nella tabella dell'indice
11     END IF
12   DONE
13END IF

```

L'inserimento di un nuovo riferimento avviene secondo il seguente algoritmo:

```

1 IF (codice, lingua, livello, umeaningId,
2   testoSelezionato == NULL) THEN
3   print (campi mancanti)
4 ELSE
5   identificativoDocumento = cercaIdentificativo(codice)
6   IF (identificativoDocumento non è valido)
7     print(codice non valido)
8   ELSE
9     carattereIniziale = trova l'inizio di testoSelezionato
10      selezionato nel testo completo
11     lunghezza = strlen(testoSelezionato)
12     inserisci umeaningId, identificativoDocumento,
13     lingua, testoSelezionato carattereIniziale
14     e lunghezza nella tabella dei riferimenti
15     del livello opportuno.
16   END IF
17END IF

```

5.2.4 Esportazione dell'ontologia

TermActionDispatcher espone la nuova funzione getOntologyOWL(), non ha parametri in input e produce l'intera ontologia di Syllabus in formato OWL.

L'ontologia OWL definita da Syllabus è composta da 3 classi: Umeaning è una sottoclasse della classe standard “Thing”, European_umeaning e National_umeaning sono sottoclassi di Umeaning.

Le relazioni ontologiche sono definite come <owl::ObjectProperty> , le relazioni transitive come ad esempio IS_A hanno la proprietà aggiuntiva

```
<rdf:type rdf:resource="owl::TransitiveProperty"/>
```

Esistono due ulteriori relazioni nella versione OWL dell'ontologia che definiscono le associazioni tra umeaning nazionali e umeaning europei: la relazione tra umeaning europeo e nazionale è chiamata “implements” ed è definita come inversa alla relazione tra umeaning nazionale ed europeo chiamata “implementation”.

Gli umeaning presenti nell'ontologia sono “individuals”, cioè istanze delle classi European_umeaning oppure National_umeaning.

Qui di seguito riportiamo un estratto dell'ontologia OWL con due soli termini:

```
18<?xml version="1.0"?>
19
20
21<!DOCTYPE rdf:RDF [
22     <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
23     <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
24     <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
25     <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#"
26     >
27     <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
28     <!ENTITY ontology
29     "http://localhost/Syllabus/ontology.owl#" >
30     <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-
31     ns#" >
32]>
33<rdf:RDF xmlns="http://localhost/Syllabus/ontology.owl#"
34     xml:base="http://localhost/Syllabus/ontology.owl"
35     xmlns:dc="http://purl.org/dc/elements/1.1/"
36     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
37     >
```

```

36   xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
37   xmlns:owl="http://www.w3.org/2002/07/owl#"
38   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
39   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
    ns#"
40   xmlns:ontology="http://www.eulawtaxonomy.org/inex.php
    ?action=getOntologyOwl#">
41   <owl:Ontology rdf:about=""/>
42
43   <owl:AnnotationProperty rdf:about="&dc;creator"/>
44
45   <owl:ObjectProperty rdf:about="#GROUPED">
46     <rdfs:comment rdf:datatype="&xsd;string"
47       >L'umeaning fa parte di un gruppo il cui capo è
    l' umeaning padre</rdfs:comment>
48   </owl:ObjectProperty>
49
50   <owl:ObjectProperty rdf:about="#HAS-VALUE">
51     <rdfs:comment
52       rdf:datatype="&xsd;string">..</rdfs:comment>
53   </owl:ObjectProperty>
54
55   <owl:ObjectProperty rdf:about="#INVERSE_OF">
56     <rdfs:comment rdf:datatype="&xsd;string"
57       >L'umeaning è il contrario del termine
    padre</rdfs:comment>
58   </owl:ObjectProperty>
59
60   <owl:ObjectProperty rdf:about="#IS_A">
61     <rdfs:comment rdf:datatype="&xsd;string"
62       >Inclusione insiemistica dell'umeaning figlio
    nell'umeaning padre</rdfs:comment>
63   </owl:ObjectProperty>
64
65   <owl:ObjectProperty rdf:about="#PART_OF">
66     <rdfs:comment rdf:datatype="&xsd;string"
67

```

```

68         >L'umeaning è parte della definizione del
        padre</rdfs:comment>
69     </owl:ObjectProperty>
70
71     <owl:ObjectProperty rdf:about="#PURPOSE">
72         <rdfs:comment rdf:datatype="&xsd:string"
73             >Principio giuridico che il termine vuole
        realizzare</rdfs:comment>
74     </owl:ObjectProperty>
75
76     <owl:ObjectProperty rdf:about="#REPLACED_BY">
77         <rdfs:comment rdf:datatype="&xsd:string"
78             >l'umeaning padre sostituisce e rende obsoleto
        l'umeaning figlio</rdfs:comment>
79     </owl:ObjectProperty>
80
81     <owl:ObjectProperty rdf:about="#quasiSyn">
82         <rdfs:comment rdf:datatype="&xsd:string"
83             >Quasi-Synonymous</rdfs:comment>
84     </owl:ObjectProperty>
85
86     <owl:Class rdf:about="#Giuridic_concept">
87         <rdfs:subClassOf rdf:resource="&owl;Thing"/>
88     </owl:Class>
89
90     <owl:Class rdf:about="&owl;Thing"/>
91
92     <European_umeaning rdf:about="#eu68">
93         <rdfs:label rdf:datatype="&xsd:string"
        xml:lang="it-IT">venditore</rdfs:label>
94         <IS_A rdf:resource="#eu70"/>
95         <dc:creator>chr1 </dc:creator>
96     </European_umeaning>
97
98     <National_umeaning rdf:about="#na86" xml:lang="it-IT">
99         <rdfs:label
        rdf:datatype="&xsd:string">ordine</rdfs:label>
100        <rdfs:comment rdf:datatype="&xsd:string">

```

101Nota d'ordine, ordinazione (Decreto Legislativo 6 settembre 2005, n. 206 Codice del consumo, a norma dell'articolo 7 della legge 29 luglio 2003, n. 229), ordine (Decreto Legislativo 9 aprile 2003, n. 70 “Attuazione della direttiva 2000/31/CE relativa a taluni aspetti giuridici dei servizi della società dell'informazione nel mercato interno, con particolare riferimento al commercio elettronico”).

```
102         </rdfs:comment>
103         <dc:creator>chr1</dc:creator>
104         <implements rdf:resource="#na123"/>
105     </National_umeaning>
106</rdf:RDF>
```

5.2.5 Riepilogo

In figura 5.3 è mostrata la nuova architettura delle classi del sistema.

5.3.1 Gestione livelli di astrazione

All'interno del database i gruppi di umeaning sono semplicemente gestiti con relazioni di tipo GROUPED, quindi non richiede modifiche nella struttura del database, tuttavia sono necessarie alcune modifiche ai vincoli di integrità per poter soddisfare i requisiti:

- Deve essere possibile creare un umeaning non collegato a nessun termine, questo è il caso dei group leader.
- Non deve essere possibile per un umeaning “figlio” essere in relazione GROUPED con più di un group leader.
- Quando un group leader non è più in relazione GROUPED con altri umeaning, perché le relazioni oppure gli umeaning figli sono stati cancellati va automaticamente cancellato anch'esso.

Gli ultimi due vincoli sono concettualmente oltre i classici vincoli di integrità referenziale, verranno quindi implementati attraverso trigger.

5.3.2 Gestione dell'evoluzione temporale

L'evoluzione temporale è gestita attraverso relazioni di tipo REPLACED_BY nell'ontologia, tuttavia per poter gestire le query di ricerca di termini scaduti e poter quindi ricreare nei risultati la situazione dell'ontologia valida in una qualsiasi data è necessario un nuovo dato: la data in cui il vecchio umeaning è stato sostituito da uno nuovo. È quindi presente una nuova tabella duplicata nei livelli europeo e nazionale: `european_expired_umeaning` e `national_expired_umeaning`, che associa gli identificativi degli umeaning scaduti alle date in cui sono stati sostituiti da un altro umeaning.

5.3.3 Archivio di testi giuridici

L'archivio di testi giuridici è gestito dalle due nuove tabelle `european_fulltext` e `national_fulltext`.

Esse contengono un identificativo auto generato, il codice del testo legale, la lingua giuridica e un lungo campo text che contiene il testo completo. La chiave è un identificativo numerico anziché l'identificativo del testo legale in quanto i diversi ordinamenti giuridici utilizzano stringhe in formati diversi che avranno quindi lunghezze diverse, un campo intero è quindi più efficiente come chiave rispetto ad un campo varchar.

Nell'ordinamento nazionale il codice del testo giuridico è unico, mentre nell'ordinamento europeo è unica la coppia codice/countrycode.

La ricerca nei testi giuridici non viene fatta attraverso una ricerca fulltext nel campo contenente il testo completo, ma attraverso la tecnica dell'indice invertito, descritta nel paragrafo 4.2.1.1.

L'indice invertito è implementato attraverso le tabelle `europa_fulltext_index` e `national_fulltext_index`, i cui campi sono l'identificativo del testo legale, la lingua giuridica la parola e la posizione nel testo, indicata in numero di parole. Anche in questo caso la differenza tra il livello nazionale ed europeo è l'unicità delle coppie identificativo del testo/posizione per quanto riguarda il livello nazionale. Per quanto riguarda il livello nazionale si avrà l'unicità della tripletta identificativo/lingua giuridica e posizione.

Un ricerca all'interno del database sarà fatta ricercando parole all'interno delle tabelle `fulltext_index`, recuperando così il testo di cui fanno parte e la loro posizione all'interno del testo.

5.3.4 Modifica della struttura dell'umeaning

Le tabelle che descrivono gli umeaning nazionali ed europei acquistano l'attributo note descritto nel paragrafo 4.6.3.1 e perdono l'attributo riferimenti.

I riferimenti giuridici quindi non sono più un semplice campo ma diventano un'associazione di tipo M-n tra l'umeaning e l'archivio di testi, rappresentati dalla tabella `europa_reference` che collega `europa_umeaning` con `europa_fulltext` e `national_reference` che collega `national_umeaning` a `national_fulltext`.

Sono composte dall'identificativo dell'umeaning, dall'identificativo del testo legale, dalla lingua giuridica (in `europa_reference`) dal numero del carattere all'interno del testo in cui il riferimento inizia, dalla lunghezza del riferimento e dal pezzo di testo selezionato come riferimento.

L'ultimo campo non è strettamente necessario, ma è utilizzato per velocizzare il caricamento degli umeaning, in questo modo si potrà evitare un accesso alle tabelle dei testi completi che sono di dimensioni maggiori.

5.3.5 Schema entità-relazione

In figura 5.4 è riportato il nuovo schema entità-relazione che implementa i cambiamenti discussi nei paragrafi precedenti.

Capitolo 6 - Implementazione

6.1 Raggruppamento

Come detto in precedenza il raggruppamento è realizzato attraverso relazioni di tipo GROUPED, per soddisfare i vincoli di integrità aggiuntivi sono stati introdotti i seguenti trigger:

- `check_european_group` e `check_national_group`: controlla prima di un inserimento che due `umeaning` non facciano parte di gruppi diversi, in questo caso blocca l'inserimento
- `delete_european_froup` e `delete_national_group`: cancella un `group leader` quando non è più in relazione GROUPED con nessun altro `umeaning`, a causa di una cancellazione di una relazione o di uno dei componenti del gruppo

Per quanto riguarda il livello di *business logic* è stato necessario aggiungere alcuni metodi nella classe `Term` per la gestione dei gruppi, in particolare:

- `insertGroup`: crea un nuovo gruppo se possibile.
- `getTermsForGroup`: in `input` richiede l'identificativo di un termine oppure di un `umeaning` e restituisce l'elenco dei termini associati a tutti gli `umeaning` che fanno parte dello stesso gruppo dell'oggetto in `input`, oppure una lista vuota se l'`umeaning` o termine in `input` non fa parte di un gruppo.
- `getGroupLeader`: prende in `input` l'identificativo di un termine oppure di un `umeaning` e restituisce l'identificativo del leader del gruppo di cui l'`umeaning` fa parte o il `group leader` di cui fanno parte gli `umeaning` collegati al termine in `input`.
- `getGroupMembers`: in `input` richiede l'identificativo di un termine oppure di un `umeaning` e se l'oggetto in `input` fa parte di un gruppo restituisce la lista di tutti gli `umeaning` che fanno parte di tale gruppo.

6.2 Evoluzione temporale

L'evoluzione temporale dei concetti giuridici, la quale porta alla sostituzione di `umeaning` con altri più recenti è implementata attraverso la presenza di relazioni del tipo `REPLACED_BY` tra gli `umeaning`.

Per quanto riguarda la struttura del database sono state aggiunte le tabelle `european_expired_umeaning` e `national_expired_umeaning`, gestite unicamente da due trigger ciascuna:

- `expire_european_umeaning` e `expire_national_umeaning`: quando una nuova relazione `REPLACED_BY` viene inserita nelle tabelle `_umeanto_umean`, inserisce l'`umeaning` figlio e la data odierna in `expired_umeaning`
- `restore_european_umeaning` e `restore_national_umeaning`: quando una relazione di tipo `REPLACED_BY` viene cancellata dalle tabelle `_umeanto_umean`, questi trigger cancellano automaticamente la corrispondente entry in `_expired_umeaning`, in questo modo l'`umeaning` scaduto diventa di nuovo valido.

Dal punto di vista della business logic si hanno dei cambiamenti nella funzione di ricerca come spiegato nel prossimo paragrafo.

6.2.1 Modifica della ricerca

La funzione di ricerca ha due nuovi parametri: `validityDate` e `showExpired`.

La query SQL di ricerca escluderà dai risultati tutti gli `umeaning` presenti nelle tabelle `_expired_umeaning` la cui data sia anteriore al valore del parametro `validityDate`, escluderà anche tutti gli `umeaning` in relazione `GROUPED` con un `group leader`.

La query seguente che torneremo ad analizzare nei prossimi paragrafi esegue una ricerca all'interno del database di `umeaning` nazionali che siano collegati a dei termini chiamati "contratto".

```

1 select t.term_id, umean.umeaning_id,
2     umean.countrycode,language_name,
3     term_name, upper(term_name) as termKey,
4     substr(description,1,30) as description,
5     'na' as level
6 from national_term t, national_umeaning umean,
7     national_term_to_umeaning t2u,
8     national_umeaning_translation ut,
9     giuridic_language gl
10where t.term_id=t2u.term_id
11and ut.umeaning_id=umean.umeaning_id
12and ut.countrycode=umean.countrycode
13and umean.umeaning_id = t2u.umeaning_id
14and gl.countrycode=umean.countrycode
15and umean.umeaning_id not in
16     (select child_umeaning
17     from national_umean_to_umean
18     where child_umeaning=umean.umeaning_id

```

```

19         and relation_type='GROUPED')
20and umean.umeaning_id not in
21     (select umeaning_id
22         from national_expired_umeaning exumean
23         where exumean.umeaning_id=umean.umeaning_id
24         and exumean.expiration_time < date '2008-10-22')
25and umean.countrycode=t.countrycode
26and (umean.countrycode='fr_FR'or umean.countrycode='it_IT')
27and ((( upper(term_name) like '%CONTRATTO%')
28 or ( upper(term_name) like '%CONTRATT%'))
29 or umean.umeaning_id in
30     (select inner_u2u.father_umeaning
31         from national_umean_to_umean inner_u2u,
32             national_term inner_t,
33             national_term_to_umeaning inner_t2u
34         where inner_t.term_id = inner_t2u.term_id
35             and inner_u2u.relation_type = 'GROUPED'
36             and inner_u2u.child_umeaning = inner_t2u.umeaning_id
37             and (( upper(inner_t.term_name) like '%CONTRATTO%')
38                 or ( upper(inner_t.term_name) like '%CONTRATT%'))))

```

La query annidata alla linea 16 esclude gli umeaning che fanno parte di un gruppo, la query alla linea 21 esclude tutti gli umeaning che sono stati sostituiti con una relazione di tipo REPLACED_BY dopo la data del 22 ottobre 2008.

La ricerca sul termine corrispondente a “contratto” è fatta alla linea 27 in cui vengono cercati gli umeaning collegati al termine contratto, ma questo non è sufficiente, perché i group leader non hanno termini propri, quindi la sottoquery alla linea 30 cerca corrispondenze tra il termine cercato e i termini collegati agli umeaning che fanno parte di un gruppo.

6.3 Archivio testi giuridici

L'archivio dei testi giuridici è conservato nel database all'interno delle tabelle european_fulltext e national_fulltext, esse hanno un indice auto generato oltre al codice univoco del testo legale in quanto un indice numerico è più efficiente di una stringa a lunghezza variabile per le operazioni di integrità referenziale.

La business logic di questo modulo è implementata dalla classe FullText e lo user interaction handler dalla classe FullTextActionDispatcher.

La classe FullText offre le seguenti funzioni:

- search(): esegue una ricerca all'interno del database, richiede, una lista di lingue giuridiche in cui fare la ricerca, il livello, il codice o una sua parte e alcune parole del testo. Tutti questi parametri sono opzionali, ma deve esserci almeno una lingua giuridica e un campo tra il codice e il testo.
- getFullText(): restituisce un testo giuridico a partire dal suo identificatore (es 93/13/CEE), dalla lingua giuridica (necessaria nel livello europeo) e dal livello.
- getSimilarities(): restituisce una lista di documenti simili a partire dai dati necessari per identificare correttamente un documento. Il funzionamento della document similarity è descritto al paragrafo 6.3.3.
- insertFullText(): inserisce un nuovo testo giuridico nel database a partire dai dati completi: codice, lingua giuridica, livello, testo e data di emissione del documento. Tutti i campi sono obbligatori.
- delete(): cancella un documento a partire dai dati necessari per identificarlo. Il database cancella automaticamente eventuali riferimenti a questo documento.
- insertReference(): inserisce un nuovo riferimento tra umeaning e testo, ha bisogno di tutti i dati per identificare sia l'umeaning che il documento: identificativo umeaning, codice del documento, lingua giuridica, livello, estratto del documento, il carattere di inizio del pezzo selezionato e la sua lunghezza.
- getReferences(): recupera una lista dei riferimenti ad un testo, richiede in input l'identificativo del testo.
- getReferredTerms(): recupera la lista di termini collegati agli umeaning che contengono un riferimento a questo testo, richiede in input i dati per identificare un testo.
- deleteReference(): cancella un riferimento dal database, richiede in input gli identificativi dell'umeaning e del documento.

Le operazioni in questo modulo sono permesse soltanto agli utenti che possiedono una nuova capability: `__CAP_FULLTEXT_INSERTION`, salvata nel campo `fulltext_insertion` della tabella `user_account`.

6.3.1 Stemmer

Per implementare lo stemming dei termini descritto nel paragrafo 4.2.2 è stato utilizzato un modulo PHP [57] che utilizza la libreria per lo stemming scritta in Snowball da Martin Porter [31], siccome le funzioni di stemming vengono utilizzate sia dal modulo Term che dal modulo FullText esse sono disponibili globalmente per tutte le classi di cui è composto il Syllabus.

Il modulo PHP rende disponibile un nuovo gruppo di funzioni dal nome stem_nomelingua dove nomelingua è il nome in inglese di una lingua, ad esempio la funzione stem_italian esegue lo stemming di parole italiane. Hanno un unico parametro che è la parola su cui eseguire lo stemming.

Sono state implementate due funzioni:

- stemCountryCode: esegue lo stemming di una parola, richiede due parametri: la parola stessa e il countrycode in formato ISO-639 [13]: verrà automaticamente invocato lo stemmer adatto per il countrycode in input.
- buildStemmedQuery: all'interno del database ora verranno ricercate solo le radici delle parole, quindi l'input di una query dell'utente deve essere processato dallo stemmer. Se l'utente ha specificato in input più di una lingua giuridica la query dovrà essere processata dagli stemmer di ogni lingua desiderata. Nella query di esempio nel paragrafo 6.2.1 alla riga 26 si può vedere un frammento di query generato da questa funzione: si ricerca la parola “contratto” in italiano ed in francese, quindi si applica su di essa sia lo stemmer in italiano che produce in output “contratt” che lo stemmer in francese che produce in output “contratto”, ricercando all'interno del database entrambe le stringhe.

6.3.2 Indice invertito

Quando un nuovo documento viene inserito all'interno del database, viene analizzato all'interno della funzione insertFullText() nella classe FullText. Ogni parola presente all'interno del documento viene processata dallo stemmer e la radice inserita nell'indice invertito se questa non è presente all'interno della tabella function_words (vedere paragrafo 4.2.2.3), che contiene una lista di parole comuni nelle cinque lingue supportate dal Syllabus.

La funzione di ricerca dei testi giuridici utilizzerà solo le tabelle european_fulltext_index e national_fulltext_index, ottenendo così prestazioni maggiori.

6.3.3 Document similarity

Quando un documento legale viene mostrato all'utente viene anche mostrata la lista dei dieci documenti giudicati più simili a questo, calcolati mediante la tecnica della *cosine document similarity* descritta nel paragrafo 4.2.3.

La document similarity è calcolata per mezzo di alcune viste, come al solito simmetriche tra il livello europeo e nazionale. La formula utilizzata per il calcolo della vicinanza tra i documenti q e d è:

$$\cos(q, d) = \frac{\sum_{(t \in q) \wedge (t \in d)} w_{d,t} w_{q,t}}{\sqrt{\left(\sum_{t \in q} w_{q,t}^2\right) \left(\sum_{t \in d} w_{d,t}^2\right)}}$$

dove $w_{x,t}$ è il peso del termine t all'interno del documento x , calcolato come $w=tf*IDF$, tf è la frequenza assoluta del termine nel documento (il numero di occorrenze), mentre IDF è la *inverse document frequency* descritta nel paragrafo 4.2.3.1.

Le frequenze, i pesi e le IDF sono calcolate nelle viste `european_terms_weights` e `national_term_weights`, eseguendo statistiche sulle tabelle `national_fulltext_index` e `european_fulltext_index` per mezzo della seguente query:

```

1 SELECT tftable.fulltext_id, tftable.word, tftable.tf,
2         idftable.idf, tftable.tf * idftable.idf AS weight
3 FROM ( SELECT european_fulltext_index.fulltext_id,
4           european_fulltext_index.word,
5           count(*) AS tf
6 FROM european_fulltext_index
7 GROUP BY european_fulltext_index.fulltext_id,
8           european_fulltext_index.word
9 ORDER BY european_fulltext_index.fulltext_id,
10          count(*) DESC) tftable,
11 ( SELECT partial.word,
12     log(((SELECT count(*) AS total
13           FROM european_fulltext)) / count(*)) AS idf
14 FROM ( SELECT DISTINCT european_fulltext_index.word,
15           european_fulltext_index.fulltext_id
16 FROM european_fulltext_index
17 ORDER BY european_fulltext_index.word,
18           european_fulltext_index.fulltext_id) AS partial
19 GROUP BY partial.word
```

```

20      ORDER BY idf DESC) idftable
21WHERE idftable.word = tftable.word
22ORDER BY tftable.tf * idftable.idf DESC;

```

La sottoquery alla riga 3 computa le frequenze assolute dei termini, quella alla riga 14 la

IDF mediante la formula $\log\left(\frac{n}{\sum_{k=1}^n \chi(f_{ik})}\right)$ dove $\chi(f_{ik})$ vale 1 se il termine è presente nel

documento, 0 altrimenti.

Una volta ottenuti questi dati la vista che computa la document similarity è la seguente:

```

1 SELECT t1.id AS id1, t2.id AS id2,
2      (( SELECT sum(tw1.weight * tw2.weight) AS sum
3         FROM european_term_weights tw1,
4              european_term_weights tw2
5         WHERE tw1.fulltext_id = t1.id
6              AND tw2.fulltext_id = t2.id
7              AND tw1.word = tw2.word)) /
8      sqrt((( SELECT sum(tw.weight * tw.weight) AS sum
9              FROM european_term_weights tw
10             WHERE tw.fulltext_id = t1.id)) *
11           (( SELECT sum(tw.weight*tw.weight) AS sum
12              FROM european_term_weights tw
13             WHERE tw.fulltext_id = t2.id))) AS similarity
14 FROM european_fulltext t1, european_fulltext t2
15 WHERE t1.id <> t2.id
16 ORDER BY t1.id, similarity DESC;

```

La sottoquery alla riga due computa il numeratore della formula della document similarity: la somma dei prodotti delle coppie di pesi di ogni termine dei due documenti. La sottoquery alla riga 8 computa il denominatore: la radice quadrata del prodotto delle somme dei quadrati dei pesi dei termini nei due documenti.

6.3.4 Riferimenti

I riferimenti giuridici sono conservati nel database nelle tabelle `european_reference` and `national_reference`. Per ragioni di prestazioni l'estratto del testo è copiato dal documento ad un campo della tabella dei riferimenti.

6.3.4.1 Ricerca e inserimento

Dalla pagina di visualizzazione degli umeaning è possibile richiedere l'inserimento di un nuovo riferimento, in questo modo si andrà alla pagina di ricerca dei riferimenti. Questa pagina è del tutto simile alla pagina di ricerca dei testi giuridici, l'unica differenza sta nel fatto che è presente l'identificativo dell'umeaning come parametro aggiuntivo implicito in tutta la fase di ricerca e di inserimento.

Quando l'utente sceglie un documento tra i risultati della ricerca esso viene visualizzato. A differenza della normale visualizzazione, in questo caso è presente una form html aggiuntiva contenente 3 pulsanti: “torna all'umeaning”, “prendi selezione” e “invia”.

Per inserire un nuovo riferimento si seleziona il pezzo di interesse dal testo e si preme il pulsante “prendi selezione” questo copia il testo selezionato in una casella di testo della form attraverso funzioni Javascript. Per poter funzionare in browser diversi con implementazioni di Javascript leggermente differenti vengono utilizzate condizionalmente diverse funzioni: `window.getSelection()`, `document.getSelection()` oppure `document.selection.createRange().text`, testando l'esistenza di ognuna.

Il tasto invia inserisce il riferimento nel database, tuttavia il browser rimane alla pagina di visualizzazione del documento, rendendo così facile l'eventuale inserzione di più riferimenti al medesimo documento.

Quando un documento viene visualizzato i termini del testo precedentemente cercati nella pagina di ricerca vengono evidenziati all'interno del documento. Per fare questo il testo del documento viene processato dalla funzione della classe `FullTextActionDispatcher HighlightTerms`, la quale utilizza una serie di espressioni regolari per racchiudere in elementi `` i termini da evidenziare.

Inoltre vengono evidenziati con una tecnica simile i termini rilevanti contenuti in riferimenti già esistenti e trasformati in link agli umeaning: non viene evidenziata tutta la parte del riferimento ma soltanto i termini corrispondenti ai termini collegati all'umeaning di cui il riferimento fa parte.

6.4 Traduzione dell'interfaccia

Il sistema di traduzione dell'interfaccia è stato riscritto. Precedentemente veniva utilizzata una serie di template Smarty diversa per ogni traduzione. Questo portava a problemi di scalabilità in quanto ogni modifica o aggiunta all'interfaccia doveva venire replicata per ogni lingua giuridica.

Il template ora è uno solo per ogni lingua e viene utilizzata per la traduzione un'estensione a Smarty [10] che fa uso della libreria Gettext [44], il cui supporto è integrato in PHP a partire dalla versione 5.

L'estensione Smarty-gettext introduce la seguente sintassi aggiuntiva al linguaggio di template:

- `{t}stringa{/t}` traduce la stringa compresa all'interno del blocco `{t}` e `{/t}`: la stringa è all'origine in inglese.
- `{t 1='one' 2='two'}first paramenter id %1{/t}`: è possibile assegnare dei parametri al blocco tradotto, le stringhe del tipo `%1 %2` etc all'interno del blocco verranno sostituite all'interno del testo.
- Plurali: se una frase dipende da un numero di oggetti deciso a runtime è possibile definire le forme plurali che la frase assume al variare del parametro numerico, esempio:

```
{t count=$files|@count 1=$files|@count plural="%1 files"}One file{/t}
```

se la variabile `$files` vale uno verrà stampato “One file” altrimenti “%1 files” dove `%1` viene sostituito con il valore di `$files`.

6.4.1 Funzionamento di Gettext

Gettext si basa sull'analisi dei codici sorgenti di un programma, in questo caso di un file generato a partire dal template smarty. In questa fase viene prodotto un file testuale il cui formato è chiamato *Portable Object* composto di coppie

msgid “stringa in inglese”

msgstr “stringa tradotta”

msgid contiene la stringa in inglese estratta dal codice sorgente, *msgstr* indica la traduzione.

Nei codici sorgenti di Syllabus la directory *locale* contiene i file portable object per le quattro lingue supportate (oltre all'inglese), essi vengono estratti e sincronizzati alle eventuali modifiche automaticamente dallo script *translate.sh*, che genera anche i file *machine object* delle traduzioni, che sono quelli che verranno effettivamente utilizzati dal sistema e vengono posizionati nelle sottodirectory *countrycode/LC_MESSAGES*.

I file machine object sono un formato binario ottimizzato per garantire l'efficienza del lookup delle stringhe. La prima parte del file è occupata dalla lunghezza e offset di una tabella di hash contenuta in esso, seguita da lunghezza e offset delle stringhe, lunghezza e offset delle stringhe tradotte, la tabella di hash medesima, in cui sono contenute le corrispondenze tra le stringhe in inglese e i loro hash. La risoluzione delle collisioni è effettuata tramite doppio hash. L'ultima parte del file è composta dalle stringhe originali terminate da 0 seguita dalle stringhe tradotte anch'esse null-terminated.

Per gestire le traduzioni è disponibile un insieme di strumenti a linea di comando: `xgettext` estrae da un file di sorgenti i messaggi e prepara un file portable object, `msgfmt` compila un file portable object in un file machine object, `msgmerge` viene usato per aggiornare il file portable object dopo che i messaggi nei sorgenti originali sono stati modificati.

Capitolo 7 - Sviluppi futuri

La realizzazione di questa tesi ha portato alla progettazione ed implementazione di un database per l'archivio di testi giuridici ed ad un inizio di analisi automatica di essi per mezzo di tecniche di Natural Language Processing.

Questo porterà alla creazione di un corpus di testi legali potenzialmente molto vasto, il quale si presta a numerose applicazioni di recenti tecniche di Natural Language Processing per l'estrazione automatica di informazioni.

Come mostrato in [59] e [60] è possibile basandosi su costrutti grammaticali chiari e non ambigui dedurre la relazione ontologica tra diversi concetti riportati nel testo. Ad esempio dal costrutto T_1 **such as** T_2, T_3, \dots, T_n è possibile dedurre che i termini T_2, \dots, T_n definiscono il concetto indicato con il termine T_1 in maniera più specifica e saranno quindi in relazione di iponimo con esso.

Inoltre come descritto nel paragrafo 3.2 quando il cambiamento normativo è esplicito si possono incontrare frasi del tipo: *“1) l’articolo 10, paragrafo 4, è sostituito dal seguente:”* anche questo caso potrà essere possibile la deduzione automatica di una relazione di tipo REPLACED_BY.

Sarà quindi possibile dedurre e costruire automaticamente parte dell'ontologia, il grado di automazione e di intervento umano necessari dipenderanno dall'accuratezza del sistema nel caso reale e dovrà quindi essere oggetto di test sul campo.

Capitolo 8 - Manuale utente

In questo capitolo illustreremo brevemente le funzionalità offerte dal sistema.

8.1 Panoramica interfaccia



Figura 8.1 L'interfaccia di Syllabus: 1) fascia del titolo; 2) Barra di navigazione; 3) Contenuto principale.

L'interfaccia utente di Syllabus, illustrata in figura 8.1 si presenta come una semplice pagina web suddivisa in tre parti principali:

1. Fascia del titolo: in questa zona è presente una combobox che permette la selezione della lingua del titolo (non ha nulla a che vedere con la gestione delle lingue giuridiche).
2. Barra di navigazione: la colonna sulla sinistra contiene il nome utente con cui ci si è autenticati e i link alle funzioni principali del sistema; il numero di link visualizzati dipende dai diritti assegnati al nome utente con cui ci si è autenticati.
3. Contenuto: è la parte principale dell'interfaccia: in questa area vengono visualizzati tutti i dati e i risultati delle operazioni richieste dall'utente; il suo contenuto varia a seconda della funzione che si è scelta nella barra di navigazione.

8.2 Autenticazione

Appena collegati alla web-application, le uniche funzioni disponibili sono la ricerca nel dizionario (approfondita nel paragrafo 8.3.1) e l'autenticazione dell'utente. Per poter compiere una qualsiasi operazione di modifica è necessario quindi autenticarsi con il proprio nome utente e password, immettendoli in una form che si ottiene cliccando sul link "login" nella barra di navigazione (vedi figura 8.1).

8.3 Funzioni del dizionario

8.3.1 Pagina di ricerca

Ricerca termine

Inserisci i parametri di ricerca

Lingua giuridica: tutti

Livello: tutti

ID umeaning:

Termine:

Descrizione: chiaro

Riferimenti:

Note:

Data validità: novembre 03 2008

Mostra gli umeaning raggruppati

Trova

ID umeaning	Lingua giuridica	Livello	Termine	Descrizione
na235	Italiano	Nazionale	Buona fede	Considerazioni sulle scelte le
na235	Italiano	Nazionale	Correttezza	Considerazioni sulle scelte le
na235	Italiano	Nazionale	Lealtà	Considerazioni sulle scelte le
na81	Italiano	Nazionale	produttore	Decreto Legislativo 6 settembr

Figura 8.2 La pagina di ricerca che mostra alcuni risultati.

La pagina di ricerca illustrata in figura 8.2 presenta una form con alcuni campi compilabili mediante i quali si possono restringere i parametri della ricerca, rispetto alla lingua giuridica, al livello nazionale oppure europeo, id dell'umeaning, nome del termine, descrizione, note oppure riferimenti giuridici.

È possibile immettere una data di validità nella ricerca: in questo modo gli umeaning sostituiti da altri dopo la data immessa verranno inclusi nei risultati. L'ultima checkbox presente decide se includere o no nei risultati gli umeaning che fanno parte di un gruppo (vedere paragrafo 8.3.5.4).

In ognuno di questi campi non è necessario inserire il testo completo, è sufficiente inserire una parola o parte di essa, ad esempio inserendo “contratto” come nome termine verrà effettuata la ricerca di tutti i termini il cui nome contiene la parola contratto o una qualsiasi sua variante, ad esempio “contratti”.

Se si vuole invece ottenere un semplice elenco di tutti i termini e di tutti gli umeaning presenti nel dizionario, è sufficiente inserire un asterisco (*) in uno qualsiasi dei campi di ricerca.

Se vengono trovati dei risultati, essi vengono presentati in una tabella al di sotto della form di ricerca e se sono molti ne verranno mostrati soltanto dieci per volta e appariranno dei pulsanti nella form di ricerca per la navigazione dei risultati, per spostarsi nella pagina precedente, successiva, prima pagina e ultima pagina.

La tabella che mostra i risultati della ricerca presenta cinque colonne:

- **Id umeaning:** identifica univocamente un umeaning, è formato da due lettere che specificano il livello nazionale (na) oppure europeo (eu) seguite da un numero. È possibile che esistano umeaning nazionali ed europei che possiedano lo stesso numero, ad esempio “eu6” e “na6”, ma questo non implica una relazione fra i due umeaning. È presentato come un link che porta alla pagina di visualizzazione dell'umeaning (trattata nel paragrafo 8.3.2). Solo agli utenti autenticati è mostrata questa colonna.
- **Lingua giuridica:** specifica la lingua di appartenenza del termine e dell'umeaning presentati in quella riga.
- **Livello:** specifica se quel termine appartiene alla legislazione comunitaria oppure ad una legislazione nazionale; il suo valore è sempre “nazionale” oppure “europeo”.
- **Termine:** il nome del termine collegato a quell' id umeaning; è presentato come un link che porta alla pagina di visualizzazione del termine (trattata nel paragrafo 8.3.3).
- **Descrizione:** è il testo completo della descrizione dell'umeaning.

8.3.2 Visualizzazione umeaning

Collegato come: **admin**

- Logout
- Cerca termine
- Gestione umeaning
- Relazioni
- Cerca testi
- Inserisci testi giuridici
- Lingua giuridica
- Utenti
- Log
- Manutenzione database
- Manuale utente
- Statistiche
- Ontologia completa

Ontologia

Grafo dell'ontologia

```

na176
"pacchetto turistico"
  • IS_A
    na177
    "viaggio tutto compreso" (Rimuovi relazione)
  • IS_A
    na178
    "vacanza tutto compreso" (Rimuovi relazione)
  • IS_A
    na179
    "circuito tutto compreso" (Rimuovi relazione)
        
```

Livello nazionale

ID umeaning: 176

Azioni	Lingua giuridica	Termine	Descrizione	Riferimenti	Note	Allegati	Implementazioni
<ul style="list-style-type: none"> Modifica Inserisci un nuovo riferimento Cancella termine Inserisci sinonimo Inserisci allegato Inserisci implementazione Inserisci figlio Aggiungi iperonimo Aggiungi iponimo Aggiungi un quasi-sinonimo Raggruppa umeaning Aggiungi sostituto 	Italian	<ul style="list-style-type: none"> • pacchetto turistico 	Notes: [...]		Notes: [...]		<ul style="list-style-type: none"> • eu27 servizio tutto compreso (Cancella implementazione)

Umeaning creato da Christian Vogel Data 19 giugno 2006, h 12:46:23
 Ultima modifica: 19 giugno 2006, h 13:08:31

Figura 8.3 Visualizzazione dei dettagli di un umeaning

Cliccando sull'id dell'umeaning, viene caricata una pagina contenente tutti i dati di quell'umeaning, mostrata in figura 8.3. È composta di due parti: il grafo dell'ontologia (non sempre presente) ed una tabella contenente i dati dell'umeaning, eventualmente tutte le sue traduzioni.

Il primo link in alto a sinistra *mostra traduzioni* è disponibile soltanto per umeaning di livello europeo; esso ricarica la pagina nascondendo, se presente, il grafo dell'ontologia e caricando tutte le traduzioni nella tabella dell'umeaning.

Il grafo dell'ontologia ha una struttura ad albero ed ha una forma simile a quella usata in Gene Ontology dalla sua interfaccia AmiGO [61]. Ogni nodo dell'albero è una riga composta dai seguenti elementi:

- Un piccolo quadrato contenente un “-”, cliccandolo il sottoalbero di quel nodo viene nascosto e l'icona diventa un “+”, cliccando un'altra volta il sottoalbero viene mostrato nuovamente. Se il nodo è foglia quest'icona è sostituita da un pallino nero.
- Il tipo di relazione ontologica fra questo nodo e il suo nodo padre.
- L'id dell'umeaning nella forma definita nel paragrafo 8.3.1.
- Il nome del termine a cui l'umeaning è associato; se fosse collegato a più di un termine viene scelto il termine più vecchio.
- Un link dal testo *rimuovi relazione*: rimuove la relazione con il nodo padre. È disponibile soltanto se ci si è autenticati come un utente in grado di modificare il dizionario. Quando viene selezionato un qualsiasi link di cancellazione, appare una finestra di dialogo che ne chiede la conferma.

Cliccando sul link *grafo ontologia* viene caricata un'immagine contenente una rappresentazione grafica dell'ontologia come grafo orientato aciclico [28]:84, di cui un semplice esempio è riportato in figura 8.4.

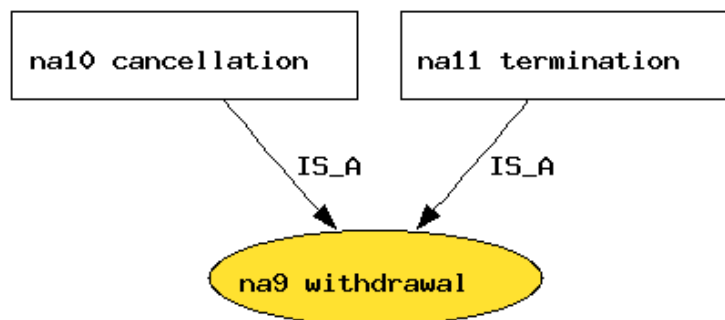


Figura 8.4 Grafo dell'ontologia corrispondente ad un umeaning del termine "withdrawal"

Se l'ontologia contiene anche relazioni non transitive, esse verranno visualizzate al di sotto del grafo dell'ontologia nella forma: id dell'umeaning padre, termine associato all'umeaning padre, tipo di relazione, id dell'umeaning figlio, termine associato all'umeaning figlio. Ad esempio *eu9 "diritto di recesso" PURPOSE eu17 "difesa del consumatore"*.

Se ci si è autenticati come un utente che ha il permesso di modificare il dizionario, è anche presente il link *rimuovi relazione*.

La tabella contenente i dati dell'umeaning presenta sette colonne:

1. Azioni: una serie di link inerenti ad azioni di modifica unicamente dell'umeaning contenuto in quella riga.

2. Lingua giuridica: la lingua dell'umeaning.
3. Termine: il termine oppure l'elenco di termini (nel caso ci siano sinonii) a cui è associato l'umeaning.
4. Descrizione: una descrizione in linguaggio naturale del significato dell'umeaning
5. Riferimenti: elenco di testi legali da cui l'umeaning è stato tratto accompagnati da estratti di essi in cui compaiono i termini che questo umeaning definisce.
6. Note: estratti di eventuali sentenze o altri testi giuridici riguardanti il concetto espresso dall'umeaning.
7. Allegati: per ogni umeaning è possibile effettuare l'upload sul server di uno o più file contenenti informazioni che consentono di piegare meglio il concetto espresso da esso, possono ad esempio essere dei file pdf contenenti il testo completo della legge da cui l'umeaning è stato tratto. Vengono qui presentati come un elenco di link da cui è possibile scaricare una copia del file.
8. Associazioni: Ad ogni umeaning di livello nazionale si può associare uno o più umeaning di livello europeo e viceversa. In questa colonna, quindi, viene visualizzato un elenco puntato di tutti gli umeaning dell'altro livello a cui questo è associato. Se ci si è autenticati come un utente che può modificare il dizionario, sarà presente un link *rimuovi associazione* utilizzato per rimuovere l'associazione tra i due umeaning nazionale ed europeo.

Le azioni specifiche per ogni umeaning sono:

- Mostra ontologia: Questo link appare soltanto quando sono mostrate le traduzioni oppure nella pagina di visualizzazione di un termine (paragrafo 8.3.3), anche durante la navigazione anonima. Viene ricaricata la pagina di visualizzazione dell'umeaning nascondendo le eventuali traduzioni e mostrando l'ontologia.
- Modifica: carica i campi dell'umeaning nella pagina di inserimento (paragrafo 8.3.4.1) in modo che si possano modificarne i dati.
- Inserisci nuovo riferimento: inserisce un nuovo riferimento ad un testo giuridico già esistente (paragrafo 8.4).
- Cancella: cancella l'umeaning e tutti i termini che erano associati unicamente a quell'umeaning.

- Inserisci sinonimo: link utilizzato per creare un nuovo termine collegato all'umeaning, che verrà considerato un sinonimo dei termini già presenti (paragrafo 8.3.4.3).
- Inserisci un'altra versione linguistica: è disponibile soltanto per umeaning europei. Porta alla pagina di creazione degli umeaning (paragrafo 8.3.4.5): l'umeaning appena creato verrà considerato una traduzione di quello di partenza.
- Inserisci allegato: porta alla pagina di upload dei file (paragrafo 8.3.4.2). Utilizzato per caricare dei file sul server che verranno considerati collegati al dato umeaning.
- Inserisci implementazione: porta ad una pagina di ricerca in cui si può selezionare l'umeaning dell'altro livello da venire associato (paragrafo 8.3.4.7).
- Inserisci figlio: porta ad una pagina di ricerca in cui si può selezionare un umeaning che diventerà figlio di questo nell'ontologia (paragrafo 8.3.5.2).
- Aggiungi iperonimo: crea una nuova relazione ontologica, di tipo hyperNym con un umeaning già esistente. Questo umeaning nella relazione sarà il figlio (paragrafo 8.3.5.3).
- Aggiungi iponimo: crea una nuova relazione ontologica, di tipo hypoNym con un umeaning già esistente. Questo umeaning nella relazione sarà il padre (paragrafo 8.3.5.3).
- Aggiungi quasi-sinonimo: crea una nuova relazione ontologica, di tipo quasiSyn con un umeaning già esistente. Questo umeaning nella relazione sarà il padre (paragrafo 8.3.5.3).
- Raggruppa umeaning: reaggruppa questo umeaning con un altro già esistente, creando se necessario un nuovo group leader, che sarà in relazione GROUPED con i due figli (paragrafo 8.3.5.4).
- Aggiungi sostituto: ricerca un umeaning già esistente. Questo umeaning sarà il sostituto di quello ricercato, indicato da una relazione ontologica di tipo REPLACED_BY (paragrafo 8.3.5.5).

8.3.3 Visualizzazione di un termine




	Azioni	ID umeaning	Lingua giuridica	Termine	Descrizione	Riferimenti	Note	Allegati	Implementazioni
<ul style="list-style-type: none"> Cerca termine Gestione umeaning Relazioni Cerca testi Inserisci testi giuridici Lingua giuridica Utenti Log Manutenzione database Manuale utente Statistiche Ontologia completa  	<ul style="list-style-type: none"> Mostra ontologia Modifica Inserisci un nuovo riferimento Cancella termine Inserisci sinonimo Inserisci allegato Inserisci implementazione Inserisci figlio Aggiungi iperonimo Aggiungi iponimo Aggiungi un quasi-sinonimo Flaggruppa umeaning Aggiungi sostituto 	na19	German	<ul style="list-style-type: none"> • Beanstandung • Mängelanzeige 	[..]		1. Bei der Umsetzung der Fernabsatzrichtlinie (Richtlinie 1997/7/EG) wurde der im Richtlinienext in Art. 5 verwendete Begriff "Beanstandung" in das deutsche Recht übernommen wurde. Der Begriff der Beschwerde, der sowohl in den Erwägungsgründen der Richtlinie als auch in ... 		<ul style="list-style-type: none"> • eu28 Beanstandung (Cancella implementazione)
	<ul style="list-style-type: none"> Mostra ontologia Modifica Inserisci un nuovo riferimento Cancella termine Inserisci sinonimo Inserisci allegato Inserisci implementazione Inserisci figlio Aggiungi iperonimo Aggiungi iponimo Aggiungi un quasi-sinonimo Flaggruppa umeaning Aggiungi sostituto 	na20	German	<ul style="list-style-type: none"> • Mängelanzeige 	[..]		1. Bei der Umsetzung der Fernabsatzrichtlinie (Richtlinie 1997/7/EG) wurde der im Richtlinienext in Art. 5 verwendete Begriff "Beanstandung" in das deutsche Recht übernommen wurde. Der Begriff der Beschwerde, der sowohl in den Erwägungsgründen der Richtlinie als auch in ... 		

Figura 8.5 Visualizzazione di un termine a cui corrispondono due umeaning

Cliccando sul nome del termine nella pagina di ricerca si accede alla pagina di visualizzazione del termine mostrata in figura 8.5. Questa pagina contiene l'elenco rappresentato in una tabella di tutti gli umeaning associati al termine scelto.

Se si è autenticati come un utente che ha la possibilità di modificare il dizionario, in alto appare una barra grigia contenente link che portano ad azioni di modifica inerenti il termine, mentre nella prima colonna della tabella contenente l'elenco degli umeaning sono presenti le azioni di modifica degli umeaning.

Le azioni contenute nella barra superiore sono:

- Mostra le altre versioni linguistiche: è disponibile solo per termini di livello europeo ed anche durante la navigazione anonima. Aggiunge all'elenco degli umeaning tutte le traduzioni di ogni umeaning dell'elenco.
- Mostra umeaning scaduti: mostra gli umeaning che sono stati sostituiti da altri a causa di un cambiamento normativo.
- Nuovo umeaning: porta ad una pagina per la creazione di un nuovo umeaning che sarà collegato al termine corrente ed apparirà quindi nell'elenco di umeaning collegati.
- Rinomina termine: porta ad una pagina in cui sarà possibile inserire un nuovo nome per il termine corrente.
- Cancella termine: cancella il termine corrente dal database, assieme a tutti gli umeaning che erano associati unicamente a quel termine.

La tabella degli umeaning ha lo stesso formato di quella nella pagina di visualizzazione degli umeaning (paragrafo 8.3.2), ma possiede una colonna in più contenete l'id dell'umeaning.

8.3.4 Gestione termini e umeaning

8.3.4.1 Inserimento di una nuova coppia termine/umeaning

The screenshot shows the 'Legal Taxonomy Syllabus' website interface. At the top, there is a header with the site title and a language selector set to 'Italian'. Below the header is a navigation bar with links: Home | Database | Commentary | Howto | Members | Intra | Links. On the left side, there is a sidebar menu with options like 'Logout', 'Cerca termine', 'Gestione umeaning', 'Relazioni', 'Cerca testi', 'Inserisci testi giuridici', 'Lingua giuridica', 'Utenti', 'Log', 'Manutenzione database', 'Manuale utente', 'Statistiche', and 'Ontologia completa'. The main content area displays a form titled 'Inserisci i dati dell'umeaning'. The form includes a dropdown for 'Lingua giuridica' (set to 'French'), a dropdown for 'Livello' (set to 'Nazionale'), a text input for 'Termine', a text area for 'Descrizione', and another text area for 'Note'. An 'Inserisci' button is located at the bottom right of the form.

Figura 8.6 Creazione di una nuova coppia termine/umeaning

Cliccando sul link *gestione umeaning* nella barra di navigazione, si ottiene una pagina contenente una form in cui si possono inserire i dati di una nuova coppia termine/umeaning, mostrata in figura 8.6.

I dati da inserire sono:

- Lingua giuridica, da selezionare tra quelle disponibili.
- Livello: è possibile selezionare solamente “nazionale” oppure “europeo”.
- Termine: il nome del termine.
- Descrizione: il passo della legge o della direttiva da cui l'umeaning è stato tratto.
- Note: estratti di eventuali sentenze o altri testi giuridici riguardanti il concetto espresso dall'umeaning.

Verrà creato un nuovo umeaning legato unicamente al termine specificato: per aggiungere sinonimi, allegati, relazioni ontologiche ed associazioni con l'altro livello è necessario utilizzare gli appositi link nella pagina di visualizzazione dell'umeaning.

8.3.4.2 Allegati

Cliccando sul link *inserisci allegato* nella pagina di visualizzazione dell'umeaning verrà caricata una form molto semplice che possiede un pulsante “Sfoggia...” che rende possibile selezionare dal proprio hard disk il file desiderato da caricare sul server ed associare all'umeaning. Ad ogni umeaning può venire associato un numero arbitrario di file.

8.3.4.3 Sinonimi

Cliccando sul link *inserisci sinonimo* nella pagina di visualizzazione dell'umeaning verrà caricata una form molto semplice che possiede una casella di testo in cui è possibile immettere il nome di un nuovo termine da creare ed associare all'umeaning. Ogni umeaning può essere associato ad un numero arbitrario di termini.

8.3.4.4 Significati multipli di un termine

Se un termine ha più di un significato, nella pagina di visualizzazione del termine (paragrafo 8.3.3) è sufficiente cliccare sul link *nuovo umeaning* e verrà presentata una form simile alla form per l'inserimento di una nuova coppia termine/umeaning (paragrafo 8.3.4.1). Tuttavia, in questo caso saranno presenti soltanto i campi “descrizione” e “riferimenti”, in quanto la lingua giuridica è fissata a quella degli umeaning già presenti, come anche il livello, mentre il termine è fissato al termine in precedenza caricato.

8.3.4.5 Traduzioni

Per inserire una nuova traduzione di un umeaning di livello europeo è necessario cliccare sul link *inserisci traduzione* nella pagina di visualizzazione dell'umeaning (paragrafo 8.3.2) e verrà presentata una form simile alla form per l'inserimento di una nuova coppia termine/umeaning (paragrafo 8.3.4.1). Tuttavia, in questo caso il campo “livello” non sarà presente, perché viene fissato al livello europeo.

8.3.4.6 Traduzioni di umeaning nazionali

È possibile effettuare traduzioni delle descrizioni degli umeaning nazionali, attraverso il link “inserisci traduzione” presente in testa alla colonna descrizione nella pagina di

visualizzazione dell'umeaning. Cliccando su questo link si aprirà una pagina con una semplice form in cui è presente solamente un campo per la descrizione tradotta.

Quando un umeaning nazionale viene visualizzato verrà sempre caricata la descrizione nella medesima lingua dell'interfaccia se disponibile.

8.3.4.7 Associazioni fra umeaning nazionali ed europei

Collegato come: **admin**

Logout

Cerca termine

Gestione umeaning

Relazioni

Cerca testi

Inserisci testi giuridici

Lingua giuridica

Utenti

Log

Manutenzione database

Manuale utente

Statistiche

Ontologia completa
W3C OWL

Inserimento implementazione

Inserisci i parametri di ricerca

ID umeaning

Termine

Descrizione

Riferimenti

Note

Data validità: novembre 03 2008

Mostra gli umeaning raggruppati

Vai alla pagina

Prima Precedente Prossima Ultima

Scegli l'umeaning che vuoi inserire

Risultati ricerca (52 Risultati trovati; pagina 1 di 6)						
Scegli ID umeaning	Lingua giuridica	Livello	Termine	Descrizione		
<input type="radio"/>	eu6	Italiano	Europeo	adeguato		
<input type="radio"/>	eu33	Italiano	Europeo	altra analoga facilitazione finanziaria		
<input type="radio"/>	eu22	Italiano	Europeo	autorizzazione		
<input type="radio"/>	eu47	Italiano	Europeo	buona fede		
<input type="radio"/>	eu74	Italiano	Europeo	commerciant		
<input type="radio"/>	eu52	Italiano	Europeo	comunicazione		
<input type="radio"/>	eu38	Italiano	Europeo	consumatore	...	
<input type="radio"/>	eu39	Italiano	Europeo	consumatore finale		
<input type="radio"/>	eu26	Italiano	Europeo	contratto di credito		
<input type="radio"/>	eu25	Italiano	Europeo	creditore		

Figura 8.7 Ricerca di un umeaning europeo da associare con uno nazionale
 Cliccando sul link *inserisci associazione* della pagina di visualizzazione dell'umeaning (paragrafo 8.3.2), viene presentata una pagina simile alla pagina di ricerca (paragrafo 8.3.1), mostrata in figura 8.7. In questo caso in ogni riga dei risultati è presente un

radiobutton: verrà selezionato quello corrispondente all'umeaning desiderato, dopo di che verrà schiacciato il pulsante *inserisci* posizionato al di sotto della tabella.

8.3.5 Gestione Ontologia

8.3.5.1 I tipi di relazione

Le relazioni ontologiche che possono intercorrere fra due umeaning possono essere di diversi tipi: ad esempio un *contratto elettronico è un contratto*, mentre una *clausola è parte di un contratto*. I tipi di relazione possono essere *transitivi* oppure *non transitivi*: le relazioni transitive (se A è in relazione con B e B è in relazione con C, allora A è in relazione con C) verranno mostrate nel grafo dell'ontologia, mentre le relazioni non transitive verranno mostrate in una parte apposita (paragrafo 8.3.2).

Gli utenti a cui è stato dato il permesso esplicito di poter creare e cancellare i tipi di relazione, potranno accedere mediante il link nella barra di navigazione *relazioni* ad una pagina in cui sarà possibile gestirli, mostrata in figura 8.8.

The screenshot shows the 'Legal Taxonomy Syllabus' interface. At the top, there is a header with the site name and a language dropdown set to 'Italian'. Below the header is a navigation bar with links: Home | Database | Commentary | Howto | Members | Intra | Links. On the left side, there is a sidebar menu with options like 'Logout', 'Cerca termine', 'Gestione umeaning', 'Relazioni', 'Cerca testi', 'Inserisci testi giuridici', 'Lingua giuridica', 'Utenti', 'Log', 'Manutenzione database', 'Manuale utente', 'Statistiche', and 'Ontologia completa'. The main content area features a form titled 'Inserisci un nuovo tipo di relazione' with fields for 'Nome della relazione' and 'Descrizione', a checkbox for 'E' una relazione transitiva', and an 'Inserisci' button. Below the form is a table titled 'Tipi di relazioni disponibili'.

Tipi di relazioni disponibili			
Azioni	Nome della relazione	Descrizione	Transitiva
Modifica Cancella	PART_OF	L'umeaning è parte della definizione del padre	Transitiva
Modifica Cancella	INVERSE_OF	L'umeaning è il contrario del termine padre	non transitiva
Modifica Cancella	PURPOSE	Principio giuridico che il termine vuole realizzare	non transitiva
Modifica Cancella	quasiSyn	Quasi-Synonymous	non transitiva
Modifica Cancella	IS_A	Inclusione insiemistica dell'umeaning figlio nell'umeaning padre	Transitiva

Figura 8.8 Pagina di gestione dei tipi di relazione.

La pagina per la modifica dei tipi di relazione presenta una form per l'inserimento e la modifica composta dai seguenti campi:

- Nome: il nome della relazione. Può essere lungo al massimo 10 caratteri e non può contenere spazi oppure numeri.
- Descrizione: una breve descrizione delle proprietà della relazione, ad esempio per la relazione IS_A, la descrizione potrebbe essere *L'inclusione insiemistica dell'umeaning figlio nell'umeaning padre*.
- È una relazione transitiva: è una checkbox mediante la quale si può specificare se la relazione sia transitiva.

Al di sotto della form per l'inserimento c'è una tabella contenente i dati di tutte le relazioni inserite, la cui prima colonna contiene dei link che servono per modificare e cancellare la relazione descritta in quella riga.

Non è possibile cancellare relazioni che sono al momento usate nelle ontologie.

8.3.5.2 Le relazioni fra due umeaning

Per mettere in relazione fra loro due diversi umeaning è necessario caricare un umeaning già esistente nella pagina di visualizzazione degli umeaning (paragrafo 8.3.2), a questo punto è possibile procedere in due modi distinti: è possibile creare una nuova coppia termine/umeaning il cui umeaning sarà collegato a quello caricato oppure è possibile effettuare una ricerca fra gli umeaning già esistenti nel database e selezionarne uno da inserire nell'ontologia.

Il primo metodo, raggiungibile dal link *crea figlio* nella tabella dell'umeaning, porta ad una form simile alla form per l'inserimento di una nuova coppia termine/umeaning (paragrafo 8.3.4.1). Tuttavia in questo caso il campo "livello" non sarà presente perché viene fissato al livello europeo, inoltre sarà presente una *combobox* da cui è possibile selezionare il tipo di relazione fra l'umeaning selezionato e quello appena creato. Al di sotto della form è inoltre presente una tabella riassuntiva sui tipi di relazione disponibili.

Cliccando invece sul link *aggiungi figlio* nella tabella dell'umeaning, viene presentata una pagina simile alla pagina di ricerca (paragrafo 8.3.1) mostrata in figura 8.9, tuttavia come la pagina di ricerca delle associazioni (paragrafo 8.3.4.7), la tabella dei risultati presenta una colonna aggiuntiva contenente un *radiobutton* mediante il quale è possibile selezionare l'umeaning desiderato. Al di sotto della tabella sono presenti una *combobox* con cui è

possibile selezionare il tipo di relazione ontologica fra i due umeaning e un pulsante *inserisci* con il quale si effettua l'inserimento della relazione ontologica dopo aver scelto l'umeaning e il tipo di relazione.

Anche in questo caso è presente una tabella di riepilogo descrittiva sui tipi di relazione disponibili.

Inserimento figlio

- Logout
- Cerca termine
- Gestione umeaning
- Relazioni
- Cerca testi
- Inserisci testi giuridici
- Lingua giuridica
- Utenti
- Log
- Manutenzione database
- Manuale utente
- Statistiche
- Ontologia completa
W3C OWL

Inserisci i parametri di ricerca

ID umeaning

Termine

De scrizione

Rife rimenti

Note

Data validità novembre 03 2008

Mostra gli umeaning raggruppati

Vai alla pagina

Prima Precedente Prossima Ultima

Scegli l'umeaning che vuoi inserire

Risultati ricerca (52 Risultati trovati; pagina 1 di 6)

Scegli ID umeaning	Lingua giuridica	Livello	Termine	Descrizione	
<input type="radio"/>	eu6	Italiano	Europeo	adeguato	
<input type="radio"/>	eu33	Italiano	Europeo	altra analoga facilitazione finanziaria	
<input type="radio"/>	eu22	Italiano	Europeo	autorizzazione	
<input type="radio"/>	eu47	Italiano	Europeo	buona fede	
<input type="radio"/>	eu74	Italiano	Europeo	commerciante	
<input type="radio"/>	eu52	Italiano	Europeo	comunicazione	
<input type="radio"/>	eu38	Italiano	Europeo	consumatore	...
<input type="radio"/>	eu39	Italiano	Europeo	consumatore finale	
<input type="radio"/>	eu26	Italiano	Europeo	contratto di credito	
<input type="radio"/>	eu25	Italiano	Europeo	creditore	

Tipo di relazione: PART_OF

Figura 8.9 Ricerca di un umeaning da aggiungere all'ontologia.

8.3.5.3 Relazioni ontologiche speciali

Per rendere più pratica l'interfaccia, la creazione di relazioni ontologiche di alcuni tipi usati frequentemente è stata facilitata aggiungendo un link nell'interfaccia di visualizzazione dell'umeaning, come descritto nel paragrafo 8.3.2, esse sono:

- hypoNym: L'umeaning figlio è un iponimo dell'umeaning padre, cioè ha un significato più specifico.
- HyperNym: L'umeaning padre è un iperonimo dell'umeaning figlio, cioè ha un significato più generale
- quasiSyn: L'umeaning padre e figlio sono fra loro quasi sinonimi.

Un ruolo importante è rivestito dalle relazioni di tipo GROUPED e REPLACED_BY, descritte nei prossimi paragrafi.

8.3.5.4 Gruppi

Un umeaning può essere un'aggregazione generalizzata di concetti, questi ultimi presentano un livello di dettaglio che si vuole solitamente nascondere, a meno che l'utente non chieda esplicitamente di esplorare i sotto-significati di un dato umeaning.

Questo è catturato dalla relazione GROUPED: essa può essere creata nel modo convenzionale come descritto nel paragrafo 8.3.5.2 oppure più semplicemente mediante il link “raggruppa umeaning” presente nella colonna delle azioni nella pagina di visualizzazione.

Si aprirà la classica pagina di ricerca degli umeaning, uguale alla pagina di ricerca per l'aggiunta di un figlio nell'ontologia. Ma in questo caso il nuovo umeaning selezionato sarà un “fratello” dell'umeaning corrente: se nessuno dei due umeaning correnti fa parte di un gruppo viene creato un nuovo umeaning, detto *group leader* che sarà padre nell'ontologia di entrambi.

Se uno dei due umeaning fa già parte di un gruppo allora l'altro verrà messo a far parte di tale gruppo, mentre non è possibile raggrupparli se fanno già parte di due gruppi diversi.

Effettuando una ricerca tra gli umeaning, i componenti di un gruppo verranno esclusi dai risultati, a meno che non vengano esplicitamente richiesti tramite l'apposita checkbox nella pagina di ricerca.

8.3.5.5 Cambiamento normativo

Un testo giuridico può rendere obsolete alcune parti di un testo precedente, si avrà così che un nuovo umeaning definito dal testo più recente può sostituire un altro umeaning definito dal testo precedente. Questo è modellato nel sistema mediante la relazione ontologica di

tipo REPLACED_BY, selezionabile mediante il link “aggiungi sostituto” nella pagina di visualizzazione dell'umeaning.

Si apre una pagina di ricerca in tutto e per tutto simile alla classica pagina di ricerca per l'aggiunta di elementi nell'ontologia, l'unica differenza (come nel caso dell'aggiunta di iponimi, iperonimi e quasi sinonimi) sta nel fatto che il tipo di relazione ontologica è fissato a REPLACED_BY. Quando la relazione ontologica fra due umeaning è stabilita questo significa che l'umeaning padre sostituisce l'umeaning figlio.

In questo modo l'umeaning figlio non comparirà più nei risultati di una ricerca, a meno che non venga specificata nella pagina di ricerca una data di validità anteriore alla data in cui è stata definita la sostituzione.

8.4 Riferimenti

Il Syllabus offre un sistema per archiviare la forma integrale di testi giuridici, effettuare delle ricerche su di essi e soprattutto collegare gli umeaning ad estratti di questi testi, per poter così dare un contesto al significato di un dato termine giuridico.

8.4.1 Inserzione testi

La figura 8.10 mostra la form di inserimento dei testi giuridici, la quale richiede i seguenti campi:

- Lingua giuridica: lingua in cui il testo è scritto.
- Livello: nazionale oppure europeo
- Codice: nome unico del testo giuridico: ad esempio 90/314/CEE
- Data emissione: la data in cui il testo è entrato in vigore
- Testo: il testo completo.

Figura 8.10: Form di inserimento di un testo giuridico

8.4.2 Ricerca testi

La ricerca di testi è effettuata tramite la form mostrata in figura 8.11. È possibile restringere la ricerca ad una lingua giuridica oppure ad un unico livello. Si può inoltre specificare un codice o parte di esso ed alcune parole contenute nel testo. Verrà effettuata la ricerca delle radici delle parole immesse anziché le prole complete, quindi ad esempio se si ricerca “contratto” verranno restituiti nei risultati anche i documenti contenenti “contratti”.

Collegato come: **admin**

Ricerca nei testi completi

Logout

Cerca termine

Gestione umeaning

Relazioni

Cerca testi

Inserisci testi giuridici

Lingua giuridica

Utenti

Log

Manutenzione database

Manuale utente

Statistiche

Ontologia completa
W3C OWL

Inserisci i parametri della ricerca

Lingua giuridica

livello

codice

testo

Risultati ricerca (2 risultati trovati; pagina 1 di 1)			
codice	Lingua giuridica	livello	testo
93/13/CEE	Italiano	Europeo	... 93/13/CEE del 5 aprile 1993 Clausole abusive nei contratti stipulati con i consumator...
90/314/CEE	Italiano	Europeo	... che vende o offre in vendita servizi tutto compreso proposti dall'organizzatore; 4) consumator...

Figura 8.11: Ricerca nei documenti giuridici

Quando un documento viene caricato dalla pagina dei risultati mostrata in figura 8.12 tutti i termini che erano stati immessi nella pagina di ricerca e che sono effettivamente presenti nel testo vengono evidenziati. Verranno inoltre sempre evidenziati i termini che fanno parte di riferimenti già esistenti. Essi saranno dei link ai corrispondenti umeaning.

Nella pagina di visualizzazione del documento è anche mostrata una lista dei dieci documenti dello stesso livello e dieci documenti dell'altro livello considerati maggiormente simili a questo mediante un'analisi statistica dei termini contenuti. Questo elenco è utile soprattutto nella fase di inserzione dei riferimenti.

8.4.3 Inserzione riferimenti

Dalla pagina di visualizzazione di un umeaning, scegliendo il link “aggiungi riferimenti” si va ad una pagina di ricerca simile a quella descritta nel paragrafo precedente, i termini ricercato nel testo sono precompilati con i termini collegati all'umeaning, ma possono venire cambiati in una seconda ricerca. Quando viene scelto un documento fra i risultati esso viene visualizzato come mostrato in figura 8.12, anche in questo caso i termini cercati vengono evidenziati.

Collegato come: **admin**

Logout

Cerca termine

Gestione umeaning

Relazioni

Cerca testi

Inserisci testi giuridici

Lingua giuridica

Utenti

Log

Manutenzione database

Manuale utente

Statistiche

Ontologia completa

W3C OWL

90/314/CEE

Livello Europeo

Lingua giuridica: Italiano

Data emissione: mer 13 giugno 1990

Documenti simili (Europeo): [93/13/CEE](#) [2001/58/CE](#)

Documenti simili (Nazionale): [titolo2](#) [titolo1](#)

Inserimento di riferimenti negli umeaning

Stai inserendo riferimenti nell'umeaning eu6 a questo testo. Evidenzia con il mouse dal testo qui sotto con il mouse e premi il pulsante "Prendi selezione" per avere un'anteprima e quindi premi il pulsante Invia

Direttiva 90/314/CEE del Consiglio, del 13 giugno 1990, concernente i viaggi, le vacanze

Direttiva 90/314/CEE del Consiglio, del 13 giugno 1990, concernente i viaggi, le vacanze ed i circuiti "tutto compreso" gazzetta ufficiale n. L 158 del 23/06/1990

DIRETTIVA DEL CONSIGLIO
del 13 giugno 1990
concernente i viaggi, le vacanze ed i circuiti « tutto compreso »
(90/314/CEE)

Figura 8.12: inserimento di un riferimento in un umeaning

In questo caso appare una form aggiuntiva in cui sono presenti tre pulsanti: torna all'umeaning, che carica nuovamente l'umeaning da cui si è partiti, prendi selezione e invia.

Per aggiungere all'umeaning un nuovo riferimento si seleziona con il mouse il pezzo di testo rilevante ai fini della definizione dell'umeaning, dopo di che si preme il pulsante "prendi selezione" : il testo selezionato con il mouse apparirà in un campo di anteprima sotto i tre pulsanti, per confermare l'inserimento si preme il pulsante invia. Il browser rimane alla medesima pagina per rendere più semplice l'inserimento di eventuali riferimenti multipli ad un medesimo documento. Finito l'inserimento si torna alla pagina di visualizzazione dell'umeaning mediante il pulsante "torna all'umeaning"

8.5 Statistiche

È disponibile una pagina di statistiche accessibile dalla barra laterale. È possibile alcuni elenchi di riepilogo sui dati contenuti nel database:

- Termini europei associati ad una trasposizione nazionale

- Termini ontologicamente collegati: mostra tutte le relazioni padre-figlio contenute nel database
- Polisemie: termini con più significati, cioè termini collegati a più di un umeaning
- Sinonimie: umeaning collegati a più di un termine. I termini collegati a questi umeaning sono quindi da considerarsi sinonimi
- Tutte le versioni linguistiche per tutti gli umeaning: per ogni umeaning europeo mostra tutte le traduzioni
- Statistiche generali: mostra il numero totale di termini e umeaning e il loro numero raggruppati per lingua e livello.

8.6 Amministrazione del sistema

Per poter utilizzare le seguenti funzioni di amministrazione del sistema è necessario essersi autenticati come utente di tipo *amministratore*.

8.6.1 Gestione degli account utente

Cliccando sul link *utenti* nella bara di navigazione, si accede ad una pagina mostrata in figura 8.13 in cui è possibile cercare gli utenti già inseriti nel database secondo i loro dati che sono: nome utente, nome, cognome ed e-mail. L' utilizzo della form di ricerca è simile all'utilizzo della form di ricerca degli umeaning (paragrafo 8.3.1): i risultati sono presentati in una tabella al di sotto della form di ricerca e se ci sono più di dieci risultati appaiono dei pulsanti di navigazione delle pagine nella form di ricerca.

The screenshot shows a web interface for user management. At the top, it indicates the user is logged in as 'admin' and provides a link to 'inserisci un nuovo utente'. A sidebar on the left contains navigation links: Logout, Cerca termine, Gestione umeaning, Relazioni, Cerca testi, Inserisci testi giuridici, and Lingua giuridica. The main content area features a search form titled 'Inserisci i parametri di ricerca' with input fields for Login (pre-filled with 'admin'), Nome, Cognome, and Email, and a 'Trova' button. Below the form, a table displays the search results for the user 'admin'.

Risultati ricerca (1 Risultati trovati: pagina 1 di 1)				
Azioni	Login	Nome	Cognome	Email
Modifica Cancella	admin	admin	admin	admin@admin

At the bottom of the sidebar, there are additional links: Utenti, Log, Manutenzione database, Manuale utente, Statistiche, and Ontologia completa (with W3C and OWL logos).

Figura 8.13 Ricerca di un utente.

La prima colonna della pagina di ricerca contiene due link per eseguire la modifica e la cancellazione dell'utente rappresentato nella riga.

Cliccando sul link *modifica* oppure sul link *crea nuovo utente* posizionato nella barra grigia in alto, viene caricata la pagina per l'inserimento di utenti mostrata in figura 8.14; nel primo caso nei campi della form vengono caricati i dati dell'utente selezionato e non è disponibile il campo *login*, mentre nel secondo caso viene creato un utente nuovo.

Figura 8.14 Creazione di un nuovo utente.

I campi della form sono:

- Login: il nome utente.
- Password: la password assegnata all'utente. I due campi *password* devono coincidere per assicurarsi che venga scritta correttamente.
- Nome e cognome: il nome e il cognome reali dell'utente.
- E-mail: l'indirizzo email dell'utente.
- Capabilities: sono cinque *checkbox* che definiscono il livello dell'utente e definiscono le operazioni che egli potrà compiere sul sistema e sono:

amministratore, navigazione sito, gestione umeaning, gestione dei tipi di relazione e gestione dell'archivio di testi.

8.6.2 Gestione delle lingue giuridiche

Lingue giuridiche disponibili		
Azioni	Countrycode	Nome lingua
Modifica Cancella	fr_FR	French
Modifica Cancella	it_IT	Italian
Modifica Cancella	en_GB	English
Modifica Cancella	es_ES	Spanish
Modifica Cancella	de_DE	German

Figura 8.15 Gestione delle lingue giuridiche.

Cliccando sul link *lingue giuridiche* nella barra di navigazione si accede ad una pagina mostrata in figura 8.15 contenente una form per l'inserimento di una nuova lingua giuridica ed una tabella contenente tutte le lingue giuridiche già inserite, provvista degli appositi link per la modifica e cancellazione delle singole lingue giuridiche.

Nel sistema, una lingua giuridica è definita come la coppia countrycode e nome. Il countrycode è un'identificatore univoco di una lingua definito da [13] e [12] ed è composto da esattamente cinque caratteri nella forma: due caratteri minuscoli che identificano la lingua, un underscore “_”, due caratteri maiuscoli che identificano lo stato. Ad esempio “de_DE” sarà il tedesco della germania mentre “de_AT” sarà il tedesco austriaco.

8.6.3 Consultazione e gestione del log degli eventi

Nel log vengono salvate informazioni sulle operazioni compiute dagli utenti, in particolare:

- Data dell'operazione.
- Nome dell'utente che ha eseguito l'operazione.
- Una breve descrizione del tipo di operazione: ad esempio “inserimento umeaning” o “cancellazione associazione”
- Messaggio: i campi del record che è stato inserito o modificato, ad esempio *inserted relation_type_name="contraente" description="il soggetto che stipula un contratto"*.

Anche la gestione del log degli eventi è implementata attraverso una pagina di ricerca, mostrata in figura 8.15. La ricerca può essere ristretta rispetto ad uno qualsiasi dei campi.

Collegato come: **admin** [Fai scadere le vecchie entry](#)

Logout

Cerca termine

Gestione umeaning

Relazioni

Cerca testi

Inserisci testi giuridici

Lingua giuridica

Utenti

Log

Manutenzione database

Manuale utente

Statistiche

Ontologia completa

Inserisci i parametri di ricerca

Entry più nuove di

Entry più vecchie di

Utente

Operazione

Messaggio

Vai alla pagina:

Prima Precedente Prossima Ultima

Risultati ricerca (14 Risultati trovati: pagina 1 di 2)				
Azioni	Utente	Data	Operazione	Messaggio
Cancella	admin	03 novembre 2008 12:07:04	european_national_association delete	european_umeaning=74, countrycode=it_IT, national_umeaning=91
Cancella	admin	03 novembre 2008 12:07:00	european_national_association insert	european_umeaning=74, european_countrycode=it_IT, national_umeaning=91
Cancella	admin	27 ottobre 2008 15:01:55	european full text delete	code prova50, countrycode fr_FR deleted
Cancella	admin	27 ottobre 2008 15:01:44	european full text delete	code prova50, countrycode en_GB deleted
Cancella	admin	27 ottobre 2008 15:01:33	european full text insert	code prova50, countrycode en_GB inserted
Cancella	admin	27 ottobre 2008 15:00:15	european full text insert	code prova50, countrycode fr_FR inserted
Cancella	admin	27 ottobre 2008 14:59:41	european full text delete	code prova50, countrycode fr_FR deleted

Figura 8.16 Ricerca nel Log degli eventi.

La tabella dei risultati contiene tutti i campi precedentemente descritti oltre ad un link per ogni riga per la cancellazione del record.

È inoltre possibile cancellare i vecchi elementi dalla tabella del log attraverso una pagina a cui si accede attraverso il link *cancella tutti gli elementi più vecchi di una certa data*.

8.6.4 Funzioni di gestione del database

Cliccando sul link *manutenzione database* sulla barra di navigazione si accede ad una semplice pagina contenente due link riguardanti alla manutenzione del database:

- Scarica il backup dei dati: si scarica un file compresso contenente una copia completa del database installabile su un qualsiasi server PostgreSQL versione 8.0 o superiore attraverso il comando *pg_restore*.
- Deframmenta il database: ricompatta la struttura fisica del database eliminando spazi vuoti e record marcati come cancellati; è consigliabile eseguirlo periodicamente, specialmente se si eseguono molte cancellazioni dal database.

Bibliografia

- 1: Marco Martin, Sviluppo di uno strumento per la gestione di dizionari giuridici basati sulle ontologie, 2005
- 2: T.R. Gruber, A Translation Approach to Portable Ontology Specification, 1993
- 3: J.Breuker, A.Valente, R.Winkels, Procs. of 1st LegOut Workshop on Legal Ontologies, 1997
- 4: T.R.Gruber, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, 1993
- 5: A.Gangemi, N.Guarino, C.Masolo, A.Oltramari, L.Schneider, Sweetening Ontologies with DOLCE. Procs of EKAW, 2002
- 6: Pietro Giunchiglia, Lightweight Ontologies, 2007
- 7: Eurovoc, 2008, <http://europa.eu.int/celex/eurovoc/>
- 8: UML working group, OMG Unified Modeling Language Specification Version 1.4.1, 2001, <http://www.omg.org/cgi-bin/apps/doc?formal/05-04-01.pdf>
- 9: S.Wheeler, Mature Design Theory in Web Development, 2005, <http://www.zend.com/php/design/mature-design.php>
- 10: M.Ohrt A.Zmievski, Smarty - the compiling PHP template engine, 2008, <http://smarty.php.net/manual/en/>
- 11: Graphviz - Graph Visualization Software, 2008, <http://www.research.att.com/sw/tools/graphviz/>
- 12: H.Alvestrand, Tags for the Identification of Languages, RFC 1766, 1995, <ftp://ftp.rfc-editor.org/in-notes/rfc1766.txt>
- 13: International Standardization Organization, ISO 639 Language Codes, 1988, <http://www.w3.org/WAI/ER/IG/ert/iso639.htm>
- 14: Chris Mungall, GO Database Schema, 2008, <http://www.godatabase.org/dev/sql/doc/>
- 15: W.Stallings, Cryptography and network security, Prentice Hall 2003
- 16: P.Jalote, An integrated approach to software engineering, Springer 1997
- 17: Z.Suraski, PHP's OO Evolution, 2004, <http://www.zend.com/php5/articles/engine2-php-oo.php>

- 18: M.Ohrt A.Zmievski, Smarty - the compiling PHP template engine, 2005,
<http://smarty.php.net/manual/en/>
- 19: International Committee for Information Technology Standards, Structured Query Language, 1999 revision, 1999
- 20: European parliament, DIRETTIVA 93/13/CEE DEL CONSIGLIO del 5 aprile 1993 concernente le clausole abusive nei contratti stipulati con i consumatori, 1993
- 21: European parliament, Direttiva 2002/65/CE del Parlamento europeo e del Consiglio concernente la commercializzazione a distanza di servizi finanziari ai consumatori, 2002
- 22: Parlamento italiano, Codice del consumo, a norma dell'articolo 7 della legge 29 luglio 2003, n. 229, 2005
- 23: R.A.Elmastri, S.B.Navathe, Sistemi di basi di dati - fondamenti, Addison-Wesley 2001
- 24: Apache Lucene, 2008, <http://lucene.apache.org>
- 25: Erik Hatcher, Lucene in action, Manning 2005
- 26: Steve Puts, Using a relational database for an inverted text index, 1991
- 27: D.Cutting, J. Pedersen, Optimizations for Dynamic inverted index maintenance, 1990
- 28: T.H.Cormen, Introduzione agli algoritmi, McGraw-Hill 1994
- 29: O. Frieder, A. Chowdhury, D. Grossman, and M. McCabe, On the Integration of Structured Data and Text: A Review of the SIRE architecture, 2000
- 30: B. Krovetz, Word sense disambiguation for large text databases, 1995
- 31: Snowball: a language for stemming algorithms, , snowball.tartarus.org
- 32: Martin Porter, An algorithm for suffix stripping, 1980
- 33: University of Neuchatel Computer Science dept, Stop word list, 2007,
<http://members.unine.ch/jacques.savoy/clef/>
- 34: C.Fox, A stop list for general text, 1990
- 35: European parliament, Direttiva 90/314/CEE del Consiglio concernente i viaggi e le vacanze "tutto compreso", 1990
- 36: E. Garcia, Cosine Similarity and Term Weight Tutorial, 2006,
<http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html>

- 37: Nicola Polettini, The Vector Space Model in Information Retrieval - Term Weighting Problem, 2004
- 38: C.E. Shannon, A Mathematical Theory of Communication, 1948
- 39: Jeffrey Isaacs, Investigating Measures for Pairwise Document Similarity, 1999
- 40: Surajit Chaudhuri, An Overview of Data Warehousing and OLAP Technology, 1997
- 41: Piercarlo Giolito, Sistemi informativi II, 2007
- 42: Johann Eder, Evolution of Dimension Data in Temporal Data Warehouses, Proc. of DaWak01 conference, 2001
- 43: Bartosz Bebel, Creation and management of versions in multiversion data warehouse, Proc. ACM SAC, 2004
- 44: Gnu Gettext, 2008, <http://www.gnu.org/software/gettext/>
- 45: D.L.McGuinness, F.vanHarmelen, OWL Web Ontology Language Overview, 2004, <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- 46: Semantic web project, 2008, <http://www.semanticweb.org>
- 47: T.B.Lee, J.Hendler, O.Lassila, The Semantic Web, 2001, <http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>
- 48: Tim Bray, Jean Paoli, C. M. Sperberg, Extensible Markup Language (XML) 1.0, 2006, <http://www.w3.org/TR/2006/REC-xml-20060816/>
- 49: Dave Beckett, RDF/XML Syntax Specification, 2004, <http://www.w3.org/TR/rdf-syntax-grammar/>
- 50: Eric Prud'hommeaux, Andy Seaborne, SPARQL Query Language for RDF, 2008, <http://www.w3.org/TR/rdf-sparql-query/>
- 51: F.Manola, E.Miller, RDF Primer, 2004, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>
- 52: T.Berners-Lee, R. Fielding, L. Masinter, Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, 1998, <ftp://ftp.rfc-editor.org/in-notes/rfc2396.txt>
- 53: IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, 1998

- 54: T. Berners-Lee, R. Fielding, H. Frystyk, Hypertext Transfer Protocol - HTTP/1.0, RFC 1945, 1996, <ftp://ftp.rfc-editor.org/in-notes/rfc1945.txt>
- 55: Protégé ontology editor and knowledge-base framework, 2008, <http://protege.stanford.edu>
- 56: W3C HTML Working Group, XHTML 1.0 The Extensible HyperText Markup Language, 2002, <http://www.w3.org/TR/2002/REC-xhtml1-20020801>
- 57: PHP Stem PECL module, 2008, <http://pecl.php.net/package/stem>
- 58: D.Sadoski, Three Tier Software Architectures, 2000, http://www.sei.cmu.edu/str/descriptions/threetier_body.html
- 59: L. Iwanska, N. Mata, K. Kruger, Fully automatic acquisition of taxonomic knowledge from large corpora of texts, Natural Language Processing and Knowledge Processing, 2000
- 60: M.A.Hearst, Automatic acquisition of hyponyms from large text corpora, 14th International Conference on Computational Linguistics, 1992
- 61: Gene Ontology Consortium, 2008, <http://www.geneontology.org>