

Università degli Studi di Torino

Facoltà di Scienze Matematiche, Fisiche e Naturali

Dipartimento di Informatica

Relazione di tirocinio

**Sviluppo di uno strumento per la gestione di dizionari
giuridici basati sulle ontologie**

Relatore: Prof. Guido Boella

Candidato: Marco Martin

Anno Accademico 2004/2005

Sommario

Capitolo 1 - Introduzione.....	1
1.1 Scopo della tesi.....	1
1.2 Le ontologie.....	2
Capitolo 2 - Sistemi attualmente esistenti.....	5
2.1 Ontologie legali.....	5
2.1.1 Eurovoc.....	5
2.1.2 Altre ricerche riguardanti ontologie legali.....	9
2.1.2.1 Language of Legal Discourse e NORMA.....	9
2.1.2.2 Functional Ontology Of Law.....	10
2.1.2.3 Frame-based Ontology.....	11
2.1.2.4 Confronto delle quattro ontologie.....	13
2.2 Altre ontologie.....	17
2.2.1 Gene Ontology.....	17
2.2.1.1 AmiGO.....	19
2.2.1.2 DAG-Edit.....	22
2.2.1.3 Dettagli tecnici riguardanti la base di dati utilizzata da Gene Ontology...	23
2.2.2 Semantic Web.....	25
2.2.2.1 RDF.....	26
2.2.2.2 OWL.....	27
Capitolo 3 - Syllabus.....	29
3.1 Perché un sistema nuovo.....	29
3.2 Esempi di Ontologie.....	31
Capitolo 4 - Analisi dei requisiti.....	34
4.1 Analisi del problema.....	34
4.1.1 Notazione.....	34
4.1.2 Modellazione statica (Object oriented analysis).....	34
4.1.3 Casi d'uso.....	38
4.1.4 Modellazione dinamica.....	40
4.1.4.1 Consultazione dizionari.....	41
4.1.4.2 Inserimento.....	42
4.1.4.3 Cancellazione.....	44
4.1.4.4 Inserimento e cancellazione ontologia e associazioni.....	45
4.2 Specifica dei requisiti.....	45
4.2.1 INTRODUCTION.....	45
4.2.1.1 Purpose.....	45
4.2.1.2 Definitions, Acronyms, Abbreviations.....	46
4.2.1.3 Notational Conventions	46
4.2.2 OVERALL DESCRIPTION.....	46
4.2.2.1 Product Perspective.....	46
4.2.2.2 Product Functions Overview.....	46
4.2.2.3 User Characteristics.....	47
4.2.3 SPECIFIC REQUIREMENTS.....	48

4.2.3.1	Inputs and Outputs.....	48
4.2.3.2	Functional Requirements.....	51
4.2.3.3	External Interface Requirements.....	54
4.2.3.4	Design Contraints.....	54
Capitolo 5	- Progettazione.....	56
5.1	Architettura del sistema.....	56
5.2	Progettazione della web-application.....	57
5.2.1	Object oriented Design.....	57
5.2.2	Package.....	66
5.3	Progettazione del database.....	66
5.3.1	Modello Entità-relazione.....	67
5.3.2	Controlli aggiuntivi sull'integrità dei dati.....	71
5.4	Design di dettaglio.....	71
5.4.1	Flusso di esecuzione.....	71
5.4.2	Autenticazione utenti.....	72
5.4.3	Azioni valide.....	73
Capitolo 6	- Implementazione.....	76
6.1	Strumenti utilizzati.....	76
6.1.1	PHP.....	76
6.1.2	PostgreSQL.....	77
6.1.3	Web-server e sistema operativo.....	77
6.2	Implementazione della web-application.....	77
6.2.1	Le azioni e i gruppi di azioni: ActionGroupDispatcher.....	77
6.2.2	Implementazione delle capabilities.....	78
6.2.3	La generazione di codice HTML: Smarty.....	79
6.2.4	Considerazioni sulla sicurezza.....	80
6.3	Implementazione dei controlli di integrità del database.....	81
6.3.1	Trigger di controllo dei dati.....	81
6.3.2	Trigger di update.....	82
6.3.3	L'upload dei file.....	83
Capitolo 7	- Possibili sviluppi futuri.....	84
Capitolo 8	- Manuale d'uso di Syllabus.....	85
8.1	Panoramica interfaccia.....	85
8.2	Autenticazione.....	86
8.3	Funzioni del dizionario.....	86
8.3.1	Pagina di ricerca.....	86
8.3.2	Visualizzazione umeaning.....	88
8.3.3	Visualizzazione di un termine.....	91
8.3.4	Gestione termini e umeaning.....	92
8.3.4.1	Inserimento di una nuova coppia termine/umeaning.....	92
8.3.4.2	Allegati.....	94
8.3.4.3	Sinonimi.....	94
8.3.4.4	Significati multipli di un termine.....	94

8.3.4.5 Traduzioni.....	94
8.3.4.6 Associazioni fra umeaning nazionali ed europei.....	94
8.3.5 Gestione Ontologia.....	95
8.3.5.1 I tipi di relazione.....	95
8.3.5.2 Le relazioni fra due umeaning.....	97
8.4 Amministrazione del sistema.....	98
8.4.1 Gestione degli account utente.....	98
8.4.2 Gestione delle lingue giuridiche.....	101
8.4.3 Consultazione e gestione del log degli eventi.....	101
8.4.4 Funzioni di gestione del database.....	102

Capitolo 1 - Introduzione

L'unione europea produce un alto numero di direttive, le quali vengono tradotte in ognuna delle lingue dei paesi membri. Le direttive sono un insieme di norme che devono venire recepite nella legislazione nazionale dei paesi membri, per portare così ad una maggiore convergenza fra le normative nazionali.

Il recepimento di una direttiva spesso non è una trasposizione alla lettera nella legislazione nazionale, ma è soggetta ad interpretazione, questo perché essa deve rimanere coerente con la tradizione giuridica nazionale: ad esempio il termine “ragionevolmente” è tradotto nella legislazione nazionale come “con ordinaria diligenza”. Un altro problema è rappresentato dal fatto che alcune tradizioni legislative preferiscono utilizzare termini differenti per concetti differenti, altre invece no, ad esempio il termine tedesco “Klar und verstarndlich” si può tradurre nella legislazione italiana con tre concetti distinti, identificati rispettivamente dai termini “chiaramente”, “chiaro e comprensibile” e “inequivoco”.

Ad oggi, esistono alcuni strumenti che possono aiutare il giurista nel compito di cercare similitudini e differenze fra la legislazione comunitaria e degli stati membri, come Eurovoc (vedere il paragrafo 2.1.1), tuttavia sono orientati solo a livello dei termini, non facendo distinzione fra un termine ed il suo significato, portando così ad ambiguità.

1.1 Scopo della tesi

Lo scopo di questa tesi è l'analisi e l'implementazione di uno strumento che faciliti l'identificazione di similitudini, differenze e principi comuni nelle legislazioni nazionali e comunitaria, orientato inizialmente sulle normative inerenti la difesa dei consumatori.

Lo strumento, che è stato chiamato *Syllabus*, offrirà un database di termini giuridici legati ai concetti che essi denotano in contesti differenti, sarà inoltre possibile esprimere le relazioni fra concetti differenti per mezzo di ontologie.

La gestione delle ontologie in Syllabus rispetto a sistemi attualmente esistenti presenta alcune peculiarità. Ogni legislazione deve venire descritta da una differente ontologia, siccome in ogni ordinamento i concetti possono essere organizzati in modi differenti. Inoltre una direttiva europea può usare concetti già presenti in alcune legislazioni ma non in tutti. Ad esempio il termine “diritto di recesso” nelle direttive comunitarie inerenti la protezione dei consumatori non corrisponde ad un concetto legale esistente nella legislazione italiana.

In questo contesto quindi l'approccio tradizionale top-down (vedere il paragrafo 1.2) di sviluppo delle ontologie mostra alcuni limiti. Di solito le ontologie sono costruite partendo da concetti generali, specializzandoli successivamente in concetti più dettagliati. In questo processo tuttavia si rischia di non tener conto dell'interpretazione tra un ordinamento e l'altro.

L'approccio seguito in Syllabus, viceversa, è di tipo bottom-up. Per prima cosa i termini vengono raccolti in un database assieme alle fonti legali in cui appaiono e vengono identificati i concetti ad essi associati. In una seconda fase i concetti vengono organizzati in ontologie che possono essere differenti nelle diverse tradizioni legali. La funzione di queste ontologie sarà comparare la tassonomia dei concetti nelle legislazioni diverse.

Nei prossimi paragrafi verranno analizzati strumenti attualmente esistenti per la gestione sia di ontologie legali che ontologie con altri scopi.

La seconda parte verterà sull'analisi, la progettazione e l'implementazione di Syllabus, mentre nell'ultimo capitolo verrà presentato il manuale utente per l'utilizzo del sistema.

1.2 Le ontologie

Uno dei metodi ad oggi più efficienti per rappresentare formalmente un insieme di concetti è la rappresentazione mediante *ontologie*.

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” [1].

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 [2]:

- **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 [3]. Un esempio di top-ontology è DOLCE [4].

Capitolo 2 - Sistemi attualmente esistenti

2.1 Ontologie legali

Nei prossimi paragrafi esamineremo le caratteristiche ed i limiti di Eurovoc [5]: un tesoro di termini giuridici inerenti l'ordinamento giuridico europeo, oltre a diversi modelli di ontologie legali che sono stati proposti.

2.1.1 Eurovoc

Eurovoc è un progetto finanziato dalla comunità europea: esso consiste in un dizionario (più precisamente un thesauro) basato sulle direttive, leggi e trattati raccolti nel database Celex [6] liberamente consultabile online tramite una web-application (vedi figura 2.1) tradotto in 19 lingue di altrettanti Paesi europei.

Sebbene non sia un'ontologia, è ad oggi probabilmente il miglior esempio di dizionario concettuale con supporto alla gerarchizzazione di termini inerente l'ordinamento giuridico comunitario.

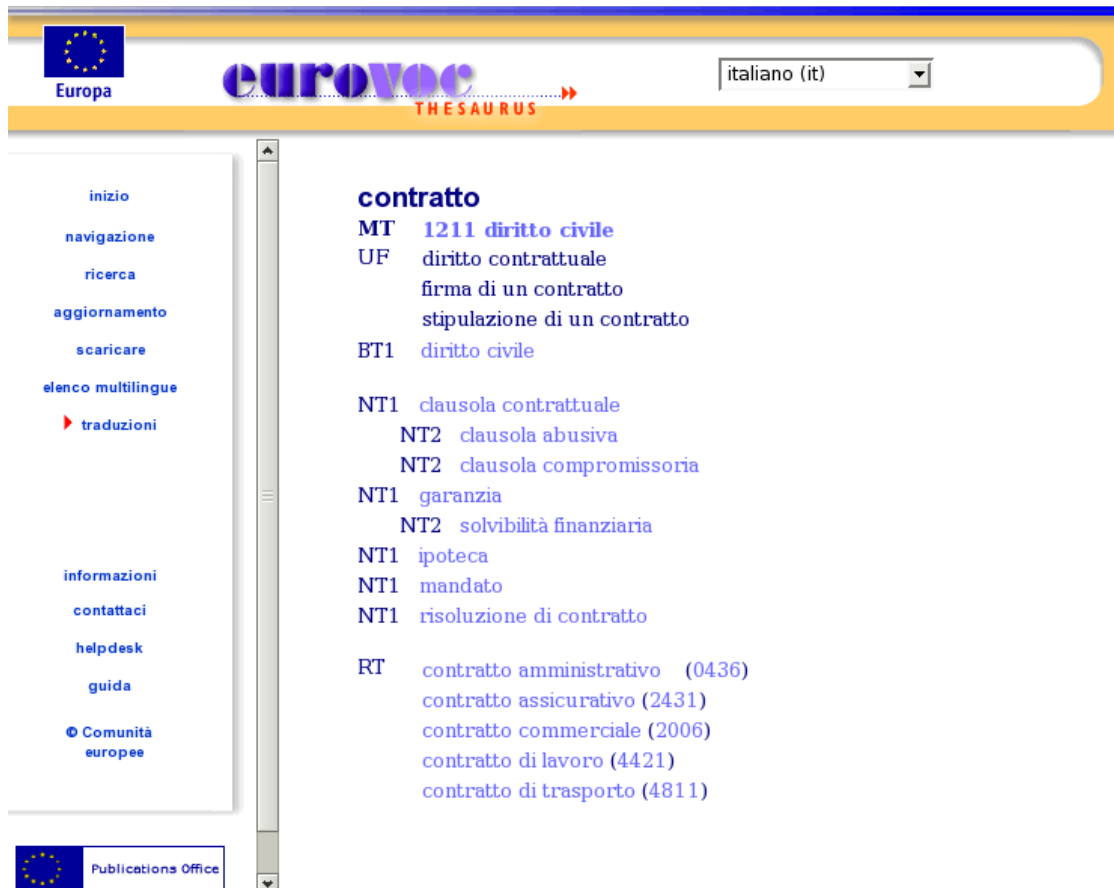


Figura 2.1 L'interfaccia web di Eurovoc

La struttura di Eurovoc è conforme agli standard ISO 5964 [7] e ISO 2778 [8] inerenti alla costruzione di thesaurus rispettivamente multilingue e monolingue.

I termini del thesaurus Eurovoc sono catalogati secondo una gerarchia di due livelli:

- I *settori* che sono rappresentati da un numero a due cifre più una stringa, ad esempio *10 Comunità Europee*.
- I microthesauri, identificati da un numero a quattro cifre, le cui prime due identificano il settore di appartenenza più una stringa, ad esempio 1011 Diritto Comunitario fa parte del settore 10 Comunità Europee.

I thesauri (in Eurovoc ogni microthesauro può essere considerato un thesauro logicamente separato) sono delle gerarchie di termini le cui relazioni collegano concetti più specifici a concetti più generali in tassonomie concettuali. Ad esempio *scuola* può essere visto come un concetto “ristretto” di *educazione*.

Un concetto più specifico di un altro viene definito un suo *narrower term* ed è indicato con la sigla NT, mentre un concetto più generale di un altro viene definito un suo *broader term*, indicato con la sigla BT.

Quindi ad esempio *BT scuola NT istruzione* indica che scuola è un narrower term di istruzione. Nell'esempio portato dalla figura 2.1, *diritto civile* è un broader term di *contratto*, mentre *clausola contrattuale* è un suo narrower term.

I termini che non possiedono alcun broader term sono detti *top term*, indicati con la sigla TT, e possono essere considerati le radici dell'albero, sebbene nell'implementazione di Eurovoc non siano espressamente indicati.

I termini del thesauro si dividono in termini *descrittori* e *non descrittori*: i primi sono parole o frasi che esprimono un concetto in maniera non ambigua e vengono utilizzati per definire la gerarchia dei broader term/narrower term; i *non descrittori* invece, sono espressioni che designano il medesimo concetto: sono quindi da intendersi come sinonimi di descrittori precedentemente esistenti.

Per poter ottenere una corrispondenza completa fra i termini in ogni lingua in cui Eurovoc è tradotto, ad ogni descrittore di una lingua corrisponde obbligatoriamente un descrittore in ciascuna delle altre lingue, mentre non ci sono corrispondenze fra i non descrittori di lingue diverse.

Se un descrittore è in relazione con un non descrittore si dice che è *used for* il non descrittore e si indica con UF, viceversa la relazione fra non descrittore e descrittore è *use*, indicata con USE, ad esempio si indicherà *contratto UF diritto contrattuale* e *diritto contrattuale USE contratto*.

Questa relazione di equivalenza comprenderà in realtà diversi tipi di relazione, può infatti indicare:

- un'equivalenza esatta
- un'equivalenza parziale
- l'opposizione di significato

- l'inclusione, nel caso un descrittore comprenda più concetti specifici, i quali diventano non descrittori a causa di un utilizzo meno frequente.

Un'ultima relazione fra termini, è la relazione associativa *related term*: essa serve per legare fra loro termini correlati che tuttavia non possiedono una relazione gerarchica di tipo *broader term/narrower term* e viene indicata mediante la sigla RT. Ad esempio *contratto RT contratto assicurativo*.

Infine, un termine può essere accompagnato da una più estesa definizione, detta *scope note*.

Questo approccio presenta alcune limitazioni che dovranno essere risolte in Syllabus:

- L'espressività delle relazioni gerarchiche è più limitata rispetto ad un'ontologia, perché sia le relazioni fra dei termini ed i loro NT, sia la relazione fra descrittori e non descrittori può essere soltanto di un tipo, quindi il vero tipo di relazione che c'è fra due termini collegati (ad esempio il narrower term può essere considerato parte del corrispondente broader term oppure potrebbe essere considerato il contrario) è lasciato all'interpretazione di chi legge.
- Ogni termine può essere tradotto in un singolo termine nelle altre lingue: si ha così una corrispondenza uno-a-uno. Questo può essere accettabile in Eurovoc, in quanto mappa solo concetti appartenenti all'ordinamento europeo. Nel sistema qui progettato, sarà tuttavia necessario mappare i termini dell'ordinamento europeo verso termini di ordinamenti nazionali, i quali possono essere anche molto diversi, ed un termine potrà essere mappato a più termini o anche a nessuno.
- Non viene fatta una distinzione fra termine e significato oltre la distinzione fra descrittori e non descrittori: ciò vuol dire che ogni termine possiede un'unica gerarchia, cioè si assume che tutti i descrittori non siano ambigui. Un termine, tuttavia, può assumere significati diversi dipendentemente dal contesto. In ciascun contesto, un termine può possedere determinati narrow term, ma non è detto che gli stessi narrow term appartengano al termine anche in contesti diversi.

- I diversi settori non seguono un'organizzazione concettuale, ma si basano sull'organizzazione burocratica degli uffici amministrativi dell'Unione Europea. Oltre a non essere un'organizzazione rigorosa dal punto di vista legale, può creare difficoltà di consultazione per chi non conosce l'organizzazione interna delle Istituzioni Europee.

2.1.2 Altre ricerche riguardanti ontologie legali

Sono stati proposti numerosi approcci per la costruzione di ontologie legali, le quali partono da concetti generali per specializzarsi successivamente in concetti più dettagliati. Di seguito ne confronteremo quattro fra i più significativi; per un confronto più dettagliato vedere [9].

2.1.2.1 *Language of Legal Discourse* e NORMA

Uno dei primi formalismi applicati al sistema legale, sebbene non propriamente un'ontologia è *Language of Legal Discourse* proposto da McCarty [10]. Esso applica l'uso di formule della logica tradizionale per costruire un linguaggio di rappresentazione della conoscenza legale. I suoi componenti principali sono: *formule atomiche, regole e modalità*.

Le *formule atomiche* sono predicati utilizzati per esprimere asserzioni, ad esempio "O1 è posseduto dall'attore A il quale ha la proprietà P". Le *regole* sono formate connettendo diverse formule atomiche con connettivi logici. Infine, le *modalità* descrivono le formule atomiche e le regole, e possono essere di tempo, eventi (cambiamenti tra due stati) e deontologia (ad esempio consentito oppure vietato).

Stamper [11] ha criticato l'uso della logica tradizionale per la rappresentazione della conoscenza giuridica, perché presenta alcuni problemi semantici. In breve, la logica tradizionale utilizza una rappresentazione simbolica che ha una connessione molto debole con i concetti del mondo reale che descrivono. Per risolvere questi problemi ha proposto il sistema NORMA (che significa logic of norms and affordances). Il sistema si basa su due assunzioni: non c'è conoscenza senza qualcuno che conosce e la conoscenza di uno che conosce dipende dal suo comportamento. Le entità del mondo sono descritte attraverso il loro comportamento anziché attraverso valori di verità assegnati arbitrariamente. I principali concetti ontologici sono: *agenti, invarianti di comportamento e realizzazioni*.

- Agenti: un agente è un'entità che sta all'interno della realtà descritta. Esso acquisisce conoscenza e modifica il mondo circostante con le sue azioni. Un agente può essere un individuo, un software oppure un'intera società.
- Invarianti di comportamento: le entità del mondo sono descritte da alcune caratteristiche di comportamento che rimangono costanti per qualche tempo. Ad esempio un bicchiere è descritto dalla sua capacità di contenere liquidi.
- Realizzazione: gli agenti realizzano delle situazioni compiendo azioni. La realizzazione è specificata come la combinazione di un agente e un invariante di comportamento. Sono descritte da un linguaggio particolare. Ad esempio se A è un agente e x è un'invariante di comportamento, Ax denota l'invariante di comportamento x realizzata dall'agente A . Esistono diversi tipi di realizzazioni: ad esempio Ax^* è l'abilità dell'agente A di realizzare x , $Ax@$ denota che A ha l'autorità per poter realizzare x , $Ax+$ e $Ax-$ denotano rispettivamente che A ha iniziato e finito di realizzare x . Combinando gli invarianti possono essere costruite realizzazioni composite, dando origine a costrutti anche molto complessi.

2.1.2.2 Functional Ontology Of Law

J.Breuker e A.Valente [2] hanno proposto un sistema basato su una prospettiva funzionale del sistema legale. Il sistema legale è considerato come uno strumento per cambiare o influenzare la società in direzioni specifiche. La sua funzione principale è quella di reagire ai comportamenti sociali e può essere scomposta in sei funzioni primitive, ciascuna corrispondente ad un particolare tipo di conoscenza legale:

- Normative knowledge: definisce i comportamenti sociali delle persone nella società.
- World knowledge (chiamato anche Legal Abstract Model): descrive il mondo che è regolamentato. Delinea i comportamenti possibili dei soggetti (persone e istituzioni), quale comportamento è consentito e quale comportamento non lo è. È suddivisa in *definitional knowledge* e *causal knowledge*; la prima è la parte statica, che consiste di definizioni di concetti legali, relazioni fra i concetti, casi (scenari reali in cui il concetto legale è applicato), circostanze e casi generici. La *causal knowledge* è la parte dinamica, che descrive il comportamento delle persone nella società secondo i termini della *definitional knowledge*.

- Responsibility knowledge: estende o restringe la responsabilità di un soggetto per il suo comportamento.
- Reactive knowledge: specifica quali azioni devono essere intraprese e come se un soggetto contravviene ad una norma.
- Meta-legal knowledge: è la conoscenza legale riguardo alla conoscenza legale. Sono informazioni riguardo alle norme, al loro contesto di applicazione, al loro scopo e alla loro posizione nella gerarchia delle norme.
- Creative knowledge: è la conoscenza legale che consente la creazione di nuove entità legali.

2.1.2.3 Frame-based Ontology

V.Kraligen [12] hanno proposto un modello che separa l'ontologia legale dalle ontologie specifiche di ogni legge; viene quindi fatta una distinzione fra due ontologie: *generic legal ontology* e *statute-specific ontology*.

La *generic legal ontology* è la parte generica e riusabile e divide la conoscenza legale in tre entità distinte: *norme*, *atti* e *descrizione di concetti*. Per ognuna l'ontologia definisce un template (chiamato frame structure) che elenca tutti gli attributi rilevanti per l'entità.

Le *norme* sono regole generali e principi di comportamento che i soggetti della legge sono costretti a seguire; una norma è composta di otto elementi:

1. Identificatore: usato come riferimento alla norma.
2. Tipo: norma di comportamento oppure norma di competenza.
3. Promulgazione: la sorgente della norma, ad esempio *Decreto legislativo 185/99 art.3.2*.
4. Contesto: il contesto di applicazione della norma.
5. Condizioni di applicazione: le circostanze in cui la norma è applicabile.
6. Soggetto: la persona o le persone alle quali la norma è diretta.
7. Modalità legale: specifica se la norma descrive qualcosa di consentito, obbligatorio oppure vietato.
8. Identificatore dell'atto: usato come riferimento ad un atto descritto separatamente.

Gli *atti* rappresentano gli aspetti dinamici descritti nelle norme; una prima distinzione tra gli atti è fatta tra gli *eventi* intesi come un cambio istantaneo tra due stati e i *processi* intesi come un cambio graduale. Una seconda distinzione è tra *atti fisici* e *atti istituzionali*, che sono una qualificazione legale di un atto fisico. Tutti gli atti possiedono quattordici elementi:

1. Identificatore dell'atto: usato come identificatore univoco.
2. Promulgazione: sorgenti legali che descrivono l'atto.
3. Contesto: il range di applicazione della descrizione dell'atto.
4. Agente: colui che compie l'atto, che può essere un individuo, un gruppo o un'istituzione.
5. Tipo: se è un tipo di atto primitivo o derivato da atti descritti altrove.
6. Modalità di significato: gli oggetti utilizzati per compiere l'atto.
7. Modalità di maniera: il modo in cui gli atti sono stati eseguiti (ad esempio in modo minaccioso).
8. Aspetti temporali: specifiche di tempo assolute, cioè data e ora.
9. Aspetti spaziali: specifica dove l'atto si verifica.
10. Circostanze: specifica in quali circostanze l'atto si verifica.
11. Causa: la causa dell'atto; specifica quali sono le ragioni per eseguire quella determinata azione.
12. Scopo: lo scopo dell'azione (l'obiettivo di chi compie l'azione).
13. Intenzionalità: l'intenzionalità o la non intenzionalità dell'azione.
14. Stato finale: Il risultato e le conseguenze dell'azione.

Le *descrizioni di concetti* definiscono i significati dei concetti del dominio.

I concetti possono essere di quattro tipi:

1. Definizioni legali.

2. Deeming Provisions: consentono di trattare provvisoriamente concetti che non sono veri come se lo fossero e viceversa.
3. Fattori: usato quando non ci sono sufficienti condizioni per l'applicabilità di un concetto un fattore contribuisce alla sua applicabilità con un "peso" positivo oppure negativo.
4. Meta-concetti: gestiscono provvedimenti speciali ed eccezioni ad altri concetti.

Ogni descrizione possiede sette elementi:

1. Il concetto da descrivere.
2. Tipo del concetto (definizione legale, deeming provisions, fattori, meta-concetti).
3. La priorità assegnata ad un fattore (rilevante soltanto per concetti di tipo "fattore").
4. Promulgazione: le sorgenti legali da cui la descrizione è stata tratta.
5. Contesto: il contesto di applicazione della descrizione.
6. Condizioni sotto cui il concetto è applicabile.
7. Lista delle istanze del concetto.

La *statute-specific ontology* consiste in predicati usati per completare la terminologia per le norme, atti e descrizioni dei concetti. Non può essere riutilizzata in sotto-domini legali diversi e deve sempre essere creata per ogni sotto-dominio. Definisce il vocabolario con cui la *knowledge base* è costruita.

2.1.2.4 Confronto delle quattro ontologie

Visser [9] confronta e valuta queste quattro ontologie rispetto ad alcuni criteri di adeguatezza, funzionalità e riusabilità.

- Adeguatezza epistemologica: cioè quanto l'ontologia è vicina al modo di pensare della persona umana che risolve dei problemi.

- Chiarezza: *Language of Legal Discourse* (d'ora in avanti LLD) data la sua natura di riferimento ai testi legali ha poche ambiguità, il linguaggio utilizzato da NORMA invece può presentare formule ambigue, ad esempio l'espressione $A(x, y)$ denota che A realizza sia x che y ma non è chiaro se x e y siano legati fra loro da qualche relazione. *Functional Ontology of Law* (d'ora in avanti LFO) è definito chiaramente anche se le definizioni di *meta-legal knowledge* e *creative knowledge* sono più vaghe delle altre definizioni, ad esempio sia la *responsibility knowledge* che la *creative knowledge* potrebbero essere considerate meta-legal, in quanto trattando di violazione di norme sono anch'esse norme riguardo a norme. Un problema simile si verifica nella *Frame Based Ontology* (FBO) in cui l'elemento *circostanze* del frame dell'atto è molto generico perché gli altri elementi dell'atto potrebbero anch'essi venire considerati circostanze.
- Intuitività: LLD non è molto intuitivo perché le categorie principali provengono da costrutti logici e non dalla teoria legale. NORMA adotta la teoria legale ma i suoi costrutti si prestano facilmente a interpretazioni errate, ad esempio John(car, dog) significa chiaramente che John ha un'automobile ed un cane, mentre John(car, broken) può interpretarsi come John è rotto ed ha un'automobile oppure più correttamente come l'automobile di John è rotta; bisogna quindi affidarsi al contesto perché la formula è insufficiente. In LFU la categorizzazione non è molto intuitiva, ad esempio alcune norme appartenengono alla *meta-legal knowledge* verrebbero più intuitivamente classificate come appartenenti alla *normative knowledge*. FBO invece non presenta particolari problemi.
- Rilevanza: cioè il grado con cui le categorie sono rilevanti per tutti i compiti e sotto-domini. Le categorie di LLD sono rilevanti in quanto tutti i costrutti esprimono concetti distinguibili fra loro. Questo è anche vero per LFU e FBO, anche se in FBO la *statute-specific ontology* è rilevante solo per alcuni domini legali. In NORMA, la distinzione fra alcuni costrutti come $A(x \text{ whenever } y)$ e $A(y \text{ then } x)$ non è molto chiara ma soprattutto la distinzione fra questi due concetti è rilevante solo in alcuni casi.

- Completezza: capire la completezza di un'ontologia non è semplice e per il momento è possibile solamente coprire alcuni aspetti, qui ci limiteremo alla completezza riguardante le tipologie di norme e procedure legali. Per quanto concerne le tipologie di norme LLD è molto completo, in quanto fa la distinzione fra norme che permettono, obbligano, proibiscono e abilitano, inoltre ognuno di questi tipi può venire negato. LFU distingue le norme rispetto alla funzione che hanno nel sistema legale, in modo simile FBO distingue le norme sia secondo la loro funzione che secondo la loro modalità (ad esempio permesso o proibito). In NORMA le norme sono espresse come invarianti comportamentali e questo consente un'elevata ricchezza di tipi di norme. Per quanto riguarda il secondo aspetto, nessuna delle ontologie sembra aver trovato una soluzione adeguata per le norme che descrivono una procedura legale, forse perché non è facile trovare un linguaggio che esprima le procedure in una semplice forma dichiarativa.
- Potere discriminativo: come per la completezza, è difficile quantificare esattamente quanto potere discriminativo offrano le diverse ontologie. FBO effettua la distinzione tra la *statute-specific ontology* e una generica ontologia legale; attraverso la distinzione fra le due ontologie si ottiene una elevata capacità di discriminazione, anche se questo avviene a spese della riusabilità della *statute-specific ontology*.
- Funzionalità: cioè il costo richiesto per implementare i concetti e le relazioni ontologiche in un linguaggio di rappresentazione.
- Indipendenza dalla codifica: per ottenere un sistema il più flessibile possibile, un'ontologia dovrebbe essere indipendente da un particolare formalismo per la rappresentazione, anche se in alcuni linguaggi formali risulterà più semplice implementare i commitment ontologici. LFU separa chiaramente la definizione dell'ontologia dalla sua rappresentazione, mentre FBO lega all'utilizzo dei frame, tuttavia non sono strutture complesse e non usano formalismi particolari come ereditarietà oppure gerarchie. NORMA ha un alto livello di astrazione ed è legato solo a delle espressioni logiche e non necessariamente ad un formalismo di rappresentazione. Infine LLD è legato al formalismo da esso definito.

- Coerenza: tutte e quattro le ontologie non presentano particolari problemi sotto questo punto di vista.
- Computabilità: cioè il grado in cui un'ontologia fornisce una solida base per una rappresentazione facilmente implementabile. Ad Eccezione di NORMA, sono state tutte utilizzate per l'implementazione di problemi reali e non sono emersi particolari problemi dal punto di vista computazionale.
- Riusabilità. La riusabilità di un'ontologia è il grado in cui essa può essere riusata ed espansa per concettualizzare nuovi aspetti legali ed espanderne il suo dominio di definizione. Un'ontologia più astratta avrà un grado di riusabilità più elevato di un'ontologia con un maggiore livello di dettaglio.
- Riusabilità dei compiti e metodi: analizzeremo il grado in cui queste ontologie sono adatte ad implementare due diversi tipi di compiti diversi da quelli per i quali sono state progettate: esprimere delle probabilità e svolgere dei compiti di pianificazione. La capacità di acquisire e pesare fattori pro oppure contro di una certa decisione richiede la capacità di esprimere informazioni probabilistiche. FBO supporta una forma di informazioni probabilistiche (i concetti di tipo *factor*). LFU non supporta esplicitamente l'espressione di informazioni probabilistiche, ma potrebbero essere descritte sotto la *definitional knowledge*. Al momento LLD e NORMA non supportano informazioni di tipo probabilistico. La pianificazione di compiti richiede la capacità di specificare degli stati e delle transizioni tra essi. LLD e FBO possiedono costrutti espliciti per il cambiamento di stato, mentre NORMA se da un lato supporta espressioni temporali relative (ad esempio A accade prima di B), tuttavia non sono possibili espressioni assolute di tempo. In LFU i cambiamenti di stato sono considerati parte del *Legal Abstract Model* ma non sono stati elaborati in dettaglio.

- Riusabilità del dominio: cioè il grado con cui le ontologie possono essere riusate per la descrizione di sotto-domini legali differenti da quello per cui sono state progettate. Nessuna di queste ontologie è dipendente da qualche sotto-dominio legale, tuttavia sono presenti alcune assunzioni che le rendono più adatte ad alcuni domini anziché altri. Ad esempio in FBO i campi di riferimento temporale probabilmente provengono dalla normativa olandese sui sussidi ai disoccupati, che è stata usata come riferimento nel design e nella valutazione dell'ontologia, inoltre la *statute-specific ontology* è propria di ogni sotto-dominio e deve venire riscritta per ogni dominio di applicazione.

2.2 Altre ontologie

La gestione delle ontologie è un problema che si presenta spesso nei più svariati campi, ogni volta che ci si trova davanti al problema dell'archiviazione e catalogazione di grandi quantità di dati.

Di seguito analizzeremo il progetto Gene Ontology [13], che verrà utilizzato come punto di riferimento nella progettazione di Syllabus e alcuni progetti legati al progetto Semantic Web [14] che si occupa dello sviluppo di nuove metodologie per un'organizzazione più rigorosa delle informazioni presenti nei siti web, in particolare RDF [15] e OWL [16].

2.2.1 Gene Ontology

Esistono parecchi database che raccolgono informazioni sui prodotti genetici inerenti molte piccole aree di ricerca. Una ricerca comparata in molti database è quindi molto dispendiosa soprattutto per il fatto che diversi database utilizzano convenzioni diverse per assegnare i nomi ai termini che descrivono i diversi prodotti genetici (cioè ogni cosa codificata all'interno di geni: possono essere ad esempio componenti cellulari oppure proteine).

Il consorzio Gene Ontology [13] (spesso riferito come GO) è nato con lo scopo di creare una terminologia consistente fra database diversi. Il progetto è cominciato con la collaborazione fra i curatori di tre database diversi: *FlyBase* (<http://flybase.bio.indiana.edu/>), *Saccharomyces Genome Database* (<http://www.yeastgenome.org/>) e *Mouse Genome Database* (<http://www.informatics.jax.org/>). Sono stati resi liberamente disponibili una serie di database e di strumenti per la loro gestione.

Gene Ontology non è un database di sequenze genetiche né un catalogo di prodotti genetici, ma descrive come i prodotti genetici si comportano nel contesto della cellula.

Attualmente Gene Ontology è composto da tre ontologie che descrivono i prodotti genetici sotto tre distinti aspetti:

- Funzione molecolare: descrive le attività a livello molecolare, come “catalytic activity”, “transporter activity” e “adenylate cyclase activity”.
- Processi biologici: un processo biologico è una serie di eventi effettuati da uno o più gruppi di funzioni molecolari. Alcuni esempi sono “cellular physiological process” e “pyrimidine metabolism”.
- Componenti cellulari: un componente cellulare è un qualche componente anatomico della cellula che è parte di un qualche oggetto più generale, come il “nucleo” o un “ribosoma”

I termini di GO sono organizzati in grafi orientati aciclici [17]:84, quindi a differenza di un albero i nodi possono avere più di un padre, ad esempio “hexose biosynthesis” è figlio sia di “hexose metabolism” che di “monosaccharide biosynthesis” siccome “biosynthesis” è un tipo di metabolismo e “hexose” è un tipo di monosaccaride.

Ad ogni termine vengono associati i prodotti genetici da esso descritto provenienti dai database che collaborano al consorzio GO, queste associazioni, nella terminologia usata da Gene Ontology vengono chiamate *annotazioni*.

Gene Ontology offre diversi strumenti: Go database, il quale è il database vero e proprio, il cui schema è pubblico, AmiGO (figura 2.2), una web-application per la consultazione delle ontologie e DAG-Edit (figura 2.5).

2.2.1.1 AmiGO

AmiGO è una web-application scritta in linguaggio Perl, una sua installazione liberamente consultabile è disponibile all'indirizzo <http://www.godatabase.org>. Essa si presenta all'utente come una pagina suddivisa in due frame: una barra di navigazione a sinistra e una pagina centrale in cui è rappresentata l'ontologia. La barra di navigazione si compone in una casella di testo per la ricerca in con cui si può effettuare una ricerca basata sui termini oppure sui prodotti genetici e tre listbox che servono per filtrare i risultati di ricerca secondo particolari parametri: la prima presenta una lista delle specie di esseri viventi a cui appartengono i prodotti genetici del database, la seconda contiene un elenco delle varie sorgenti di dati che collaborano a Gene Ontology da cui il termine proviene (ad esempio FlyBase), infine la terza contiene l'elenco degli *evidence code*: essi sono il tipo di prova a sostegno di tale termine: ad esempio da quale lavoro di ricerca proviene la deduzione che un termine sia veramente part_of di un altro. Alcuni esempi sono: “IC: inferred by curator” quando non ci sono prove dirette oppure “IDA: Inferred from Direct Assay” quando c'è un'analisi scientifica per la funzione, processo o componente indicato dal termine GO.



Figura 2.2 AmiGO durante la navigazione dell'ontologia

Quando si effettua una ricerca viene presentata una pagina contenente i risultati trovati, da cui si può scegliere se caricare l'ontologia di uno di essi oppure se visualizzare l'unione delle ontologie di più risultati.

La navigazione dell'ontologia si presenta come una struttura ad albero le cui relazioni padre/figlio sono rappresentate mediante una maggiore indentazione (è anche possibile scaricare l'ontologia in formato immagine realizzata da GraphViz di AT&T [18]). Il termine al momento selezionato verrà presentato in grassetto.

Ogni elemento dell'albero è composto da sei elementi. Partendo da sinistra, il primo è un'icona che rappresenta un +, un – o un punto e il suo comportamento è simile alla visualizzazione dell'albero del filesystem in un filemanager. Un + significa che il termine ha dei figli e cliccandoci sopra la vista verrà “espansa” mostrando i figli del termine; un – significa che la vista è espansa e cliccandoci sopra i figli vengono nascosti, infine un punto significa che il termine è foglia.

La seconda icona rappresenta il tipo di relazione del termine con il padre: Gene Ontology utilizza le due relazioni ISA (è un) e PART_OF (è parte di). È seguita dall'id numerico che identifica univocamente il termine del database e il nome del termine (ad esempio GO:0008150: biological_process). Syllabus seguirà un approccio simile, tuttavia, siccome distinguerà fra termini e concetti, indicherà l'identificatore del concetto anziché del termine.

Gli ultimi due dati sono il numero di prodotti genetici che sono stati annotati a quel termine e un'icona che è un link alla rappresentazione grafica mediante un grafico a torta che rappresenta quali percentuali del precedente numero di prodotti genetici sono associate ad ognuno dei suoi discendenti nell'ontologia.

Cliccando sul nome di un termine viene aperta una nuova finestra (figura 2.3) che presenta una vista più dettagliata del termine, composta nella parte superiore dal nome del termine, un elenco di eventuali sinonimi, la definizione e un riepilogo dell'ontologia. La sezione successiva presenta un elenco di riferimenti esterni, cioè dove il termine è citato in altri database.

Il resto della pagina è dedicato alle associazioni con i prodotti genetici. Ci sono alcuni controlli per filtrare i prodotti genetici e la tabella che ne presenta l'elenco. La prima colonna è il nome del prodotto genetico, che è un link ad una pagina di dettagli sul prodotto. La seconda colonna è un link al sito della sorgente dei dati, la terza colonna

nutrient reservoir activity

Accession: GO:0045735
Synonyms: None
Definition:
 Functions in the storage of nutritious substrates.
Term Lineage [Graphical View](#)
[GO:0003673 : Gene_Ontology \(20041 \)](#)
 Ⓢ [GO:0003674 : molecular_function \(6411 \)](#)
 Ⓢ [GO:0045735 : nutrient_reservoir_activity \(3 \)](#)

External References

- ⊕ InterPro (10)
- ⊕ MIPS_funcat (1)
- ⊕ Pfam (4)
- ⊕ PRINTS (7)
- ⊕ ProDom (1)
- ⊕ PROSITE (1)
- ⊕ SP_KW (2)

Direct Gene Product Associations

Direct Associations

Filter Associations

Datasource	Evidence Code	Species
TAIR	All Curator Approved	A. thaliana
Wormbase	IMP	A. tumefaciens str.
Ensembl	IGI	B. anthracis str. Am
RGD	IPI	B. aphidicola (Acyr

Gene Symbol	Datasource	Evidence	Full Name
<input type="checkbox"/> CRA1	TAIR	ISS	CRUCIFERINA
<input type="checkbox"/> CRU3	TAIR	ISS	CRUCIFERIN 3
<input type="checkbox"/> RD22	TAIR	TAS	None

Figura 2.3 Visualizzazione dei dettagli di un termine

rappresenta il tipo di prova e l'ultima colonna presenta il nome completo (se esiste) del prodotto genetico.

La pagina dei dettagli dei prodotti genetici (figura 2.4) presenta il nome, eventuali sinonimi, i termini a cui il prodotto genetico è associato e la completa sequenza peptidica del prodotto genetico (quando disponibile).

Gene Product: CAT1_CAEEL

Full Name: Peroxisomal catalase 1
Synonyms: None
Datasource: UniProt

Associated to Terms:

Evidence	Term
IMP	<input type="checkbox"/> catalase activity
IMP	<input type="checkbox"/> determination of adult life span
IMP	<input type="checkbox"/> peroxidase activity
IDA	<input type="checkbox"/> peroxisome
NAS	<input type="checkbox"/> response to oxidative stress

Peptide Sequence

```
>UNIPROT|Q27487 symbol:CAT1_CAEEL pfam:PF00199 Inte
MPNDPSDNQLKTYKETYPVSLEFKGTDAFSTTTTIYSNLFQKQVITTS
NGAPIYSKTAVLTAGRRGPMMLQDVVYMDEMAHFDRERIPERVVHAKGAG
AHGYFEVTHDISKYCKADIFNKVKGQTPLLIRFSTVGGESGSADTARDPR
GFAIKFYTEEGNWDLVGNNTPIFFIRDP IHFPNFIHTQKRNPQTHLKDPN
MIFDFWLRPEALHQVMFLFSDRGLPDGYRHMNGYGSHTFKMVNKDGKAI
YVKPHFKPTQGVKNLTVEKAGQLASSDPDYSIRDLFNAIEKGDFFVWKM
IQVMTFEQAEKWEFNFDFVTKVWPHGDYPLIEVGKMLNRRNPRNYFAEVE
QSAFCPAHIVPGIEFSPDKMLQGRIFSYTDTHPHRLGPNYIQLPVNCPYR
SRAHNTQRDGAMAYDNQQHAPNFFPNSFNYGKTRPDVKDTFFPATGDVDR
YESGDDNNYDQPRQFWEKVLDTGARERMCFNAGPLGECHDFI IKGMIDH
FSKVHPDFGARVKALIQKQARSHI
```

Check/Uncheck All Draw New Tree Submit

Figura 2.4 Dettagli di un prodotto genetico

2.2.1.2 DAG-Edit

AmiGO è limitato dal fatto che consente solo la consultazione del database in lettura, per costruire i dati dell'ontologia è necessario usare un software aggiuntivo dal nome DAG-Edit, un programma grafico scritto in linguaggio Java, mostrato in figura 2.5.

Mediante questo software è possibile definire i termini e le relazioni ontologiche fra essi e la varie tipologie di relazioni. Il dizionario viene poi salvato in un particolare formato di file testuale: l'*OBO flat file format*, il quale può essere convertito da uno script Perl in formato SQL per poi effettuare l'inserimento nel database GO.

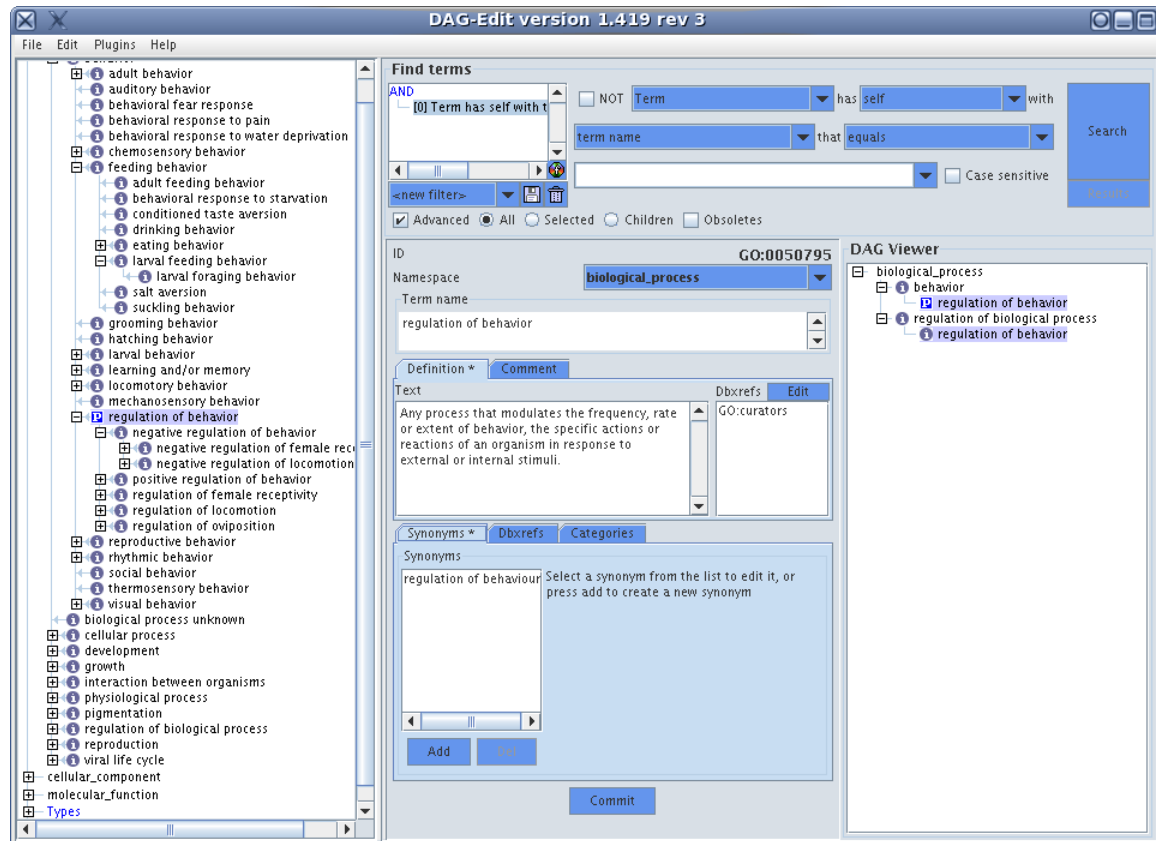


Figura 2.5 DAG-Edit mentre mostra il termine “regulation of behaviour” dell'ontologia di Gene Ontology

2.2.1.3 Dettagli tecnici riguardanti la base di dati utilizzata da Gene Ontology

La base dati usata internamente dal GO database [19] offre alcuni spunti interessanti riguardanti la gestione di ontologie mediante basi di dati relazionali. Il database di Gene Ontology si compone di 39 tabelle: in questa sede verranno analizzate soprattutto le relazioni inerenti al problema trattato, cioè quelle coinvolte nella gestione dell'ontologia, trascurando le tabelle volte alla rappresentazione dei prodotti genetici, delle sequenze peptidiche, delle prove e dei riferimenti esterni.

Il concetto centrale gestito dal database è il *grafo* dell'ontologia. I termini GO sono visti come *nodi* del grafo e le relazioni tra essi sono gli *archi*, i quali possiedono un'etichetta che individua quale sia il tipo di relazione. I nodi e gli archi sono implementati rispettivamente dalle relazioni **term** e **term2term**, mentre i tipi di relazione vengono immagazzinati come termini, sono differenziati dai termini appartenenti alle tre principali ontologie perché appartengono ad una speciale ontologia per la catalogazione dei tipi di relazione. Syllabus gestirà le ontologie in maniera simile, ma con due piccole differenze: le relazioni non saranno fra i termini, ma fra i loro significati; inoltre, per semplicità, i tipi di relazione non saranno termini, ma saranno considerati concetti separati, quindi conservati in una tabella apposita (vedere il paragrafo 4.1.2).

Quando vengono fatte query complesse riguardanti l'ontologia che richiedono, tuttavia, se si dispone solo delle tabelle precedenti potrebbe essere necessario l'utilizzo di un ciclo di parecchie query SQL.

Per rendere più efficiente tale tipo di query, viene quindi precomputato il percorso da ogni nodo verso tutti i suoi antenati, cioè viene calcolata la *chiusura transitiva riflessiva* [20]:228 delle relazioni: riflessiva perché viene aggiunta la relazione di ogni nodo con se stesso a distanza 0; essa viene implementata dalla tabella **graph_path**.

Più in dettaglio, la tabella **term** è composta da 6 colonne:

- **id**: intero chiave primaria che identifica univocamente i termini nel database, viene usata per effettuare il join delle tabelle.
- **name**: il nome del termine.
- **term_type**: l'ontologia a cui il termine appartiene.
- **acc**: chiave surrogata della tabella termine. Sarà l'id numerico pubblico presentato all'utente.
- **is_obsolete**: booleano (implementato mediante integer) se è settato a uno il termine è da considerarsi obsoleto.
- **is_root**: se settato ad uno il termine è radice.

La tabella **term2term** è composta da 5 colonne:

- id: intero chiave primaria che identifica univocamente la relazione.
- relationship_type_id: chiave esterna che punta alla colonna id della tabella term. Il termine a cui fa riferimento dovrà far parte dell'ontologia volta a catalogare i tipi di relazione.
- term1_id: chiave esterna che punta al termine “padre”.
- term2_id: chiave esterna che punta al termine “figlio”.
- complete: booleano che se settato ad uno la relazione è parte di una “definizione completa” del termine.

La tabella di ottimizzazione **graph_path** è composta di 4 colonne:

- id: intero chiave primaria che identifica univocamente il percorso.
- term1_id: chiave esterna che punta al termine “antenato”.
- term2_id: chiave esterna che punta al termine “discendente”.
- distance: la distanza fra i due termini, cioè quanti termini si trovano nel percorso fra il termine antenato e il termine discendente.

Siccome Gene Ontology utilizza MySQL come database (sebbene sia possibile installare il sistema anche su altri tipi di database) il quale non supporta ancora check e trigger la tabella graph_path deve venire aggiornata dal programma che si collega come client al database.

2.2.2 Semantic Web

Il Semantic Web è un progetto guidato dal consorzio W3C (World Wide Web Consortium) [14] con la partecipazione di molti ricercatori e partner industriali.

Tim Berners Lee [21] ha definito il Semantic Web come “l'estensione del web di oggi nel quale l'informazione è fornita in una maniera ben definita, consentendo una migliore interazione fra uomini e computer”. Infatti, i documenti pubblicati nel web come si presenta oggi sono progettati per essere letti e compresi da esseri umani e mal si prestano all'elaborazione automatica da parte di un computer.

Lo scopo del Semantic Web è quello di creare una serie di strumenti che rendano possibile assegnare un significato ai documenti e alle relazioni tra essi, cioè mappare i concetti espressi da essi in ontologie facilmente interrogabili ed elaborabili automaticamente.

Vengono già utilizzate due importanti tecnologie per lo sviluppo del Semantic Web: l'eXtensible Markup Language, il quale consente di descrivere semanticamente le diverse parti di un documento e il Resource Description Framework (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, un linguaggio per l'interrogazione di ontologie RDF simile a SQL e OWL (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.

In una fase futura dello sviluppo di Syllabus, verrà implementato il supporto all'importazione ed esportazione dapprima di ontologie OWL.

2.2.2.1 RDF

RDF [15] 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 Identifiers (URI) [22]. 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 [15], 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 [23] 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.”

2.2.2.2 OWL

OWL [16] 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.

Alcune proprietà aggiuntive definite da OWL sono:

- 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.

Capitolo 3 - Syllabus

3.1 *Perchè un sistema nuovo*

Come abbiamo visto, la maggior parte dei sistemi esistenti per la gestione di ontologie si basano sulla gerarchizzazione dei termini, anziché dei concetti che essi descrivono. Se questo può andare bene per sistemi di ontologie scientifiche come Gene Ontology, dove ogni componente cellulare è descritto rigorosamente da un unico termine, al contrario un sistema per la gestione di ontologie legali dovrebbe gestire le relazioni che intercorrono fra i concetti definiti dai termini anziché fra i termini stessi; in questo modo è possibile gestire eventuali ambiguità che si incontrano qualora si abbia a che fare con termini che possiedono più di un significato a seconda del contesto in cui ci si trova.

Questo problema di ambiguità di significato emerge soprattutto nel contesto di Eurovoc e Syllabus, quando cioè si ha a che fare con delle ontologie multilingue. Il tradizionale approccio top-down nello sviluppo di ontologie legali, come esaminato nel paragrafo 2.1.2, è adatto per descrivere ontologie orientate ad una singola tradizione nazionale. Queste ontologie, quindi, modellano semplicemente il codice legislativo, ma non la *dottrina* legale, che è il lavoro di interpretazione e rielaborazione del codice legislativo, che è fondamentale per trasporre una direttiva europea nelle leggi nazionali.

Partendo invece dalla legislazione reale, cioè seguendo un metodo di tipo bottom-up, si ottiene un approccio il più neutrale possibile, in quanto riflette solo termini e concetti esistenti.

In un prima fase verranno raccolti i termini assieme ai testi in cui appaiono e verranno quindi salvati in un database, oltre al termine, anche il passaggio della direttiva o della legge da cui il termine proviene e l'eventuale testo completo della legge: questo definisce il contesto del termine, da cui si può inferire il concetto identificato dal termine in tale contesto.

Una volta identificato un insieme di concetti, esso viene organizzato in ontologie, le quali sono differenti ed indipendenti per ogni lingua comunitaria. Inoltre, sono anche indipendenti le ontologie costruite a partire da concetti presi dalla legislazione di livello europeo rispetto alle ontologie costruite a partire da concetti presi dalle legislazioni di livello nazionale. Questo lavoro di ricostruzione dovrà essere fatto non da esperti di ontologie quanto da esperti in materia legale.

In questa fase verranno quindi prodotte ontologie molto semplificate, che saranno quindi senza vincoli di inferenza, come offerti da strumenti più avanzati come OWL.

In una seconda fase, degli esperti di ontologie potranno riorganizzarne la struttura ed integrarla in un'ontologia top-level (vedere il paragrafo 1.2) come DOLCE [4], a questo punto potrà essere aggiunta al sistema la capacità di importare ed esportare le ontologie in linguaggi di rappresentazione come OWL.

Siccome all'inizio il programma per la gestione delle ontologie dovrà essere usato da esperti legali a cui non sono richieste particolari conoscenze informatiche, il sistema deve offrire una buona semplicità d'uso e, siccome sarà un lavoro coordinato fra diversi gruppi, dovrà essere facilmente accessibile da computer diversi e non avere quindi vincoli di sistema operativo e non richiedere software particolari installati.

Questi due vincoli portano ad escludere applicazioni per la gestione di ontologie come DAG-Edit (vedere il paragrafo 2.2.1.2) o Protégé [24], che sebbene molto completi, presentano un'elevata complessità ed inoltre sono fatti per lavorare su file locali rendendo così difficile la collaborazione fra più soggetti.

Si è quindi deciso di realizzare un'interfaccia basata sul web e di gestire le ontologie mediante un database server, che offre la scalabilità, concorrenza e controllo di integrità richiesti. Il framework per l'ontologia è stato ispirato da Gene Ontology, dal quale eredita la rappresentazione grafica delle ontologie. A differenza di AmiGO, inoltre, l'interfaccia web di Syllabus dovrà essere in grado non solo di visualizzare i dizionari e le ontologie, ma dovrà poter gestire anche la creazione e la cancellazione dei termini, delle definizioni e delle ontologie.

Nei prossimi capitoli verrà presentata l'analisi, la progettazione e l'implementazione dell'interfaccia web e della base dati a cui essa si appoggia.

3.2 Esempi di Ontologie

Per illustrare meglio quanto detto, ora verrà illustrata la struttura di alcune ontologie inerenti concetti appartenenti al settore legale che verrà coperto da Syllabus, in particolare riguardanti i termini “diritto di recesso” e “contratto”.

Le relazioni utilizzate in questi esempi sono di due tipi: *IS_A* e *PURPOSE*. La prima indica un'inclusione insiemistica: ad esempio l'espressione “contratto *IS_A* accordo” indica che un significato del termine “contratto” è un tipo di accordo. *PURPOSE* indica che il termine “padre” è un principio giuridico che il termine “figlio” vuole realizzare, ad esempio “diritto di recesso *PURPOSE* tutela del consumatore” vuol dire che la tutela del consumatore è uno scopo di una delle accezioni del diritto di recesso.

I significati dei singoli termini cambiano nei diversi ordinamenti e fra il livello nazionale ed europeo, ad esempio in italiano e nella legislazione europea il termine “contratto” è unicamente un accordo fra due parti con un effettivo valore legale, mentre nella legislazione britannica, il corrispondente termine “contract” è sia un “agreement”, in questo caso assume il medesimo significato di “contratto” in Italiano, sia un “bargain”, cioè una negoziazione informale fra due parti che porta ad un accordo privo di valore legale. Il termine di livello europeo rappresentante il contratto viene associato con i vari termini di livello nazionale corrispondenti; tuttavia per quanto riguarda la legislazione inglese, bisognerà tener conto che il termine “contract” di livello europeo ha unicamente il significato di “agreement” e non “bargain”.

Nel diritto Europeo, “diritto di recesso” tradotto in inglese come “withdrawal” è un rimedio autonomo volto a permettere che il consumatore abbia la possibilità di essere sufficientemente informato del contratto, pena la possibilità di “withdrawing from the contract”.

Nella legislazione Inglese, il termine “withdrawal” rappresenta l'intenzione di una delle parti di uscire dall'impegno preso, che tuttavia si deve sempre risolvere in una delle due pratiche di “cancellation” oppure “termination”. Nel caso di “cancellation” sono previsti dei rimedi relativi al risarcimento del danno e alle restituzioni che “termination” non ha, per cui il primo è rimedio più grave.

Anche in Italiano il termine “diritto di recesso” possiede due significati, che tuttavia sono differenti dai concetti della legislazione Inglese: può essere inteso come il diritto di una delle parti di rescindere unilateralmente un contratto qualora l'altra non abbia rispettato alcune clausole e accade solamente in contratti di tipo elettronico o vendite per corrispondenza, oppure la rescissione del contratto in accordo fra le due parti, in questo caso è una particolare modalità di conclusione del contratto, quest'ultima non corrisponde con il concetto di diritto di recesso del livello europeo.

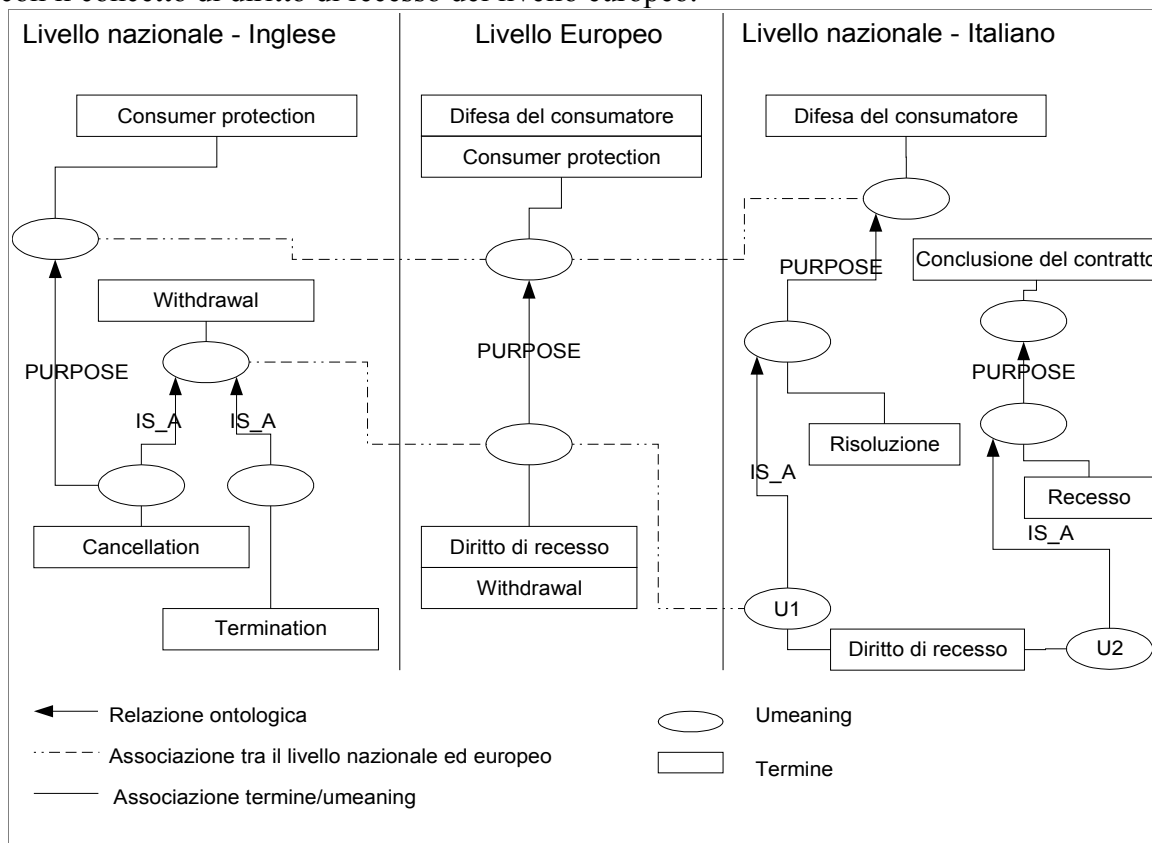


Figura 3.1 Struttura dell'ontologia dei termini legati al diritto di recesso

La figura 3.1 illustra l'ontologia dei termini legati al concetto di diritto di recesso nel livello europeo e nei livelli nazionali Italiano ed Inglese. In Italiano “diritto di recesso” è una “risoluzione” oppure un “recesso” a seconda dei casi e possiede quindi due significati che d'ora in poi chiameremo umeaning. Il primo IS_A “risoluzione” e l'altro IS_A “recesso” e solo il primo corrisponde all'umeaning “diritto di recesso” del livello europeo. Il PURPOSE di “recesso” è “conclusione del contratto” mentre il PURPOSE di “risoluzione” è “difesa del consumatore”. In inglese, invece sia “cancellation” che “termination” sono un tipo di “withdrawal” e il PURPOSE di “cancellation” è “consumer protection”.

Nel livello europeo il termine “withdrawal” tradotto in italiano come “diritto di recesso” è indipendente non essendo in relazione IS_A con altri termini e il suo PURPOSE è “difesa del consumatore”. Le due traduzioni di “diritto di recesso” e “difesa del consumatore” sono in relazione con i rispettivi meaning di livello nazionale.

Capitolo 4 - Analisi dei requisiti

4.1 *Analisi del problema*

Il problema, come detto in precedenza, prevede la costruzione di un dizionario giuridico che faciliti il compito di recepimento delle direttive europee nei singoli ordinamenti giuridici nazionali, ciò vuol dire un'efficiente mappatura dei concetti presenti nei testi delle direttive comunitarie, quando possibile, verso concetti precedentemente esistenti nei testi delle leggi degli ordinamenti nazionali. Inoltre è importante che questi concetti vengano presentati in un'ontologia per poter meglio comprendere le differenze che ci sono tra il livello comunitario e il livello nazionale: ad esempio il termine “diritto di recesso” assume significati diversi nei vari ordinamenti nazionali, di conseguenza anche la struttura dell'ontologia sarà diversa.

4.1.1 **Notazione**

In questo capitolo viene analizzata la natura del problema da risolvere, usando la tecnica dell'analisi orientata agli oggetti, producendo e analizzando all'inizio un class diagram UML, che descrive l'aspetto statico e le entità coinvolte nel problema, dopo di che, per analizzare gli aspetti dinamici del problema, verranno utilizzati diagrammi delle attività e diagrammi di cambiamento di stato.

Tutti i class diagram, diagrammi di attività, diagrammi di stato, diagrammi di collaborazione e deployment diagram utilizzati qui di seguito saranno compatibili con lo standard UML definito in [25].

4.1.2 **Modellazione statica (Object oriented analysis)**

Il sistema da modellare è un dizionario con alcune particolarità aggiuntive. Come ogni dizionario contiene un'insieme di coppie chiave-valore, le quali sono termini e le loro definizioni.

Definiamo come **termine** l'espressione giuridica che vogliamo definire, il quale può essere tanto una singola parola, come ad esempio “contratto” quanto una proposizione 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 definizione, che riporta un passo di una normativa che cita quel termine, ad esempio “...tali clausole devono essere sempre redatte in modo *chiaro e comprensibile*.” e i riferimenti, che riportano gli estremi della legge o della direttiva da cui la definizione è stata tratta, 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.

La prima cosa che si nota è che un termine può avere più di un significato, ciò dipende dal contesto in cui viene utilizzato, ad esempio come precedentemente visto, 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 la rescissione 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 lo stesso significato, in questo caso vengono detti **sinonimi**, ad esempio “chiaro e comprensibile” e “comprensibile e accessibile”.

Da ciò si deduce che gli oggetti finora identificati, cioè il termine e l'umeaning avranno una corrispondenza fra loro di tipo molti-a-molti.

Per quanto riguarda l'ontologia, saranno in relazione fra loro non tanto i termini, i quali possono avere più di un significato, ma i loro umeaning, in quanto un termine è in relazione con un altro in una delle sue accezioni, ma non è detto che tale relazione sussista negli altri significati.

Le relazioni fra due umeaning, inoltre 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”.

Prima di vedere come sono definite le associazioni fra umeaning provenienti da direttive europee ed umeaning provenienti da leggi di ordinamenti nazionali, è necessario chiedersi in *quanti* dizionari deve essere rappresentato il problema.

Il numero totale di dizionari che verranno caricati nel sistema dipenderà dal numero di lingue che esso supporterà. Supponiamo ad esempio che all'inizio vengano supportati italiano, inglese, francese, tedesco e spagnolo; saranno presenti 5 dizionari contenenti i termini tratti da leggi nazionali di rispettivamente Italia, Inghilterra, Francia, Germania e Spagna e 5 dizionari contenenti i termini tratti da direttive tradotte nelle 5 rispettive lingue.

Tuttavia i 5 dizionari “nazionali” sono concettualmente equivalenti, e possono essere espressi da un unico dizionario, che chiameremo dizionario nazionale: basterà aggiungere l'attributo “lingua giuridica” al termine e all'umeaning appartenenti a questo dizionario.

Per quanto riguarda i termini e gli umeaning provenienti dal testo di direttive europee, invece, la situazione è differente: mentre gli umeaning del dizionario nazionale provengono da ordinamenti giuridici differenti fra loro, quindi concetti simili saranno comunque umeaning diversi, i testi delle direttive europee sono tutti la traduzione di un unico testo in Inglese. Questo vuol dire che un concetto espresso da un testo di direttiva sarà unico per tutte le lingue ed avrà differenti traduzioni, quindi è necessario utilizzare un dizionario dalla struttura diversa rispetto a quello utilizzato per gli umeaning provenienti da leggi nazionali. Saranno quindi necessari due rappresentazioni separate dei termini, una per il livello nazionale ed una per il livello europeo.

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.

Definiti i dizionari, ora è possibile definire l'associazione fra umeaning di livello nazionale ed europeo. Anche in questo caso l'associazione verrà fatta fra umeaning e non fra termini. Inoltre un umeaning di livello nazionale verrà associato con una sola delle traduzioni di un umeaning di livello europeo, in particolare quella corrispondente alla stessa lingua. Da ciò si deduce la necessità di trattare il caso particolare in cui una lingua viene utilizzata in più Stati, cioè di distinguere ad esempio fra il Tedesco utilizzato in Germania ed il Tedesco utilizzato in Austria, per non confondere ordinamenti giuridici differenti che tuttavia utilizzano la stessa lingua. Come si vedrà nella fase di design, questo verrà risolto identificando la lingua giuridica mediante un countrycode definito in [26].

Riepilogando, in figura 4.1 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.

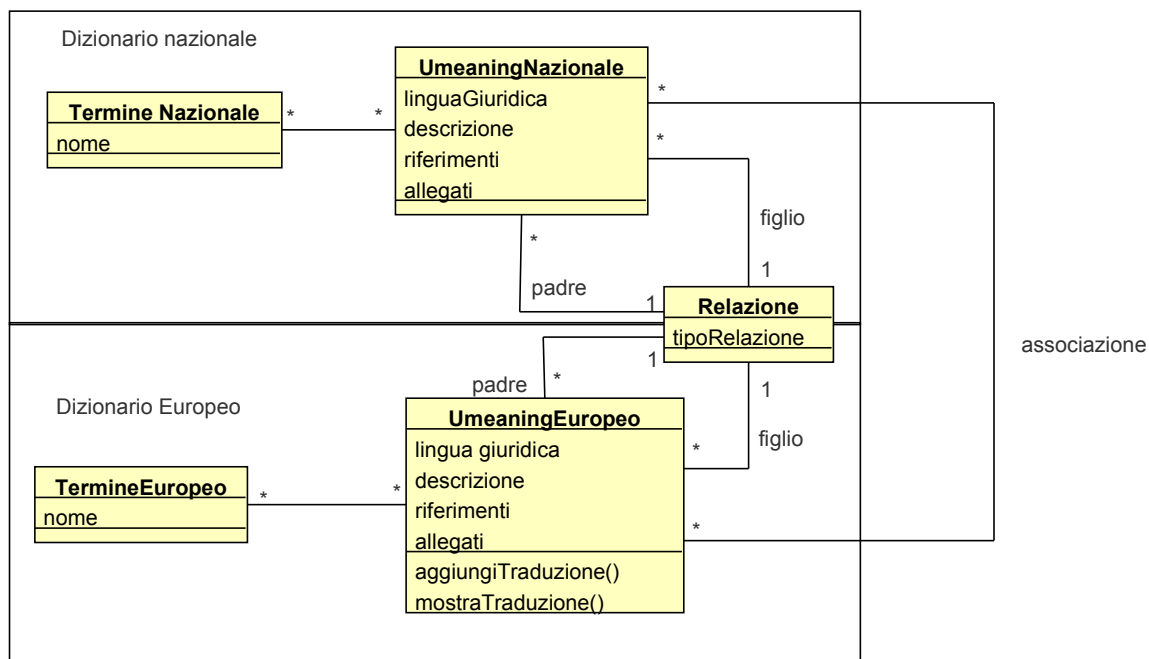


Figura 4.1 Class diagram che descrive il dominio del problema

Le classi Termine possiedono entrambe l'attributo nome, che sarà il nome del termine, ad esempio “contratto” oppure “chiaro e comprensibile”, mentre le classi Umeaning possiedono gli attributi linguaGiuridica, descrizione, riferimenti e allegati, che servono a definire l'umeaning. Oltre a questo, la classe UmeaningEuropeo possiede due operazioni, aggiungiTraduzione e mostraTraduzione che esprimono la presenza di più traduzioni per un unico umeaning europeo.

Le classi Umeaning sono collegate alle rispettive classi Termine tramite un'associazione multi-a-molti, questo per consentire sia la presenza di termini considerati sinonimi che la presenza di termini con più significati.

Per esprimere l'ontologia fra gli umeaning, le classi Umeaning hanno anche un'associazione multi-a-molti riflessiva con se stessa, tuttavia passando per una classe intermediaria Relazione, la quale ha un unico attributo tipoRelazione, questo per indicare la presenza di tipi diversi di relazioni. Siccome i tipi di relazione sono gli stessi sia per il livello europeo che per il livello nazionale, la classe Relazione è condivisa fra i due dizionari.

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

4.1.3 Casi d'uso

Per tracciare una prima mappa delle funzionalità richieste dal sistema, è importante identificare la tipologia di utenti a cui sarà destinato. La relazione fra le diverse tipologie di utente e le operazioni che potranno compiere è rappresentata in figura 4.2.

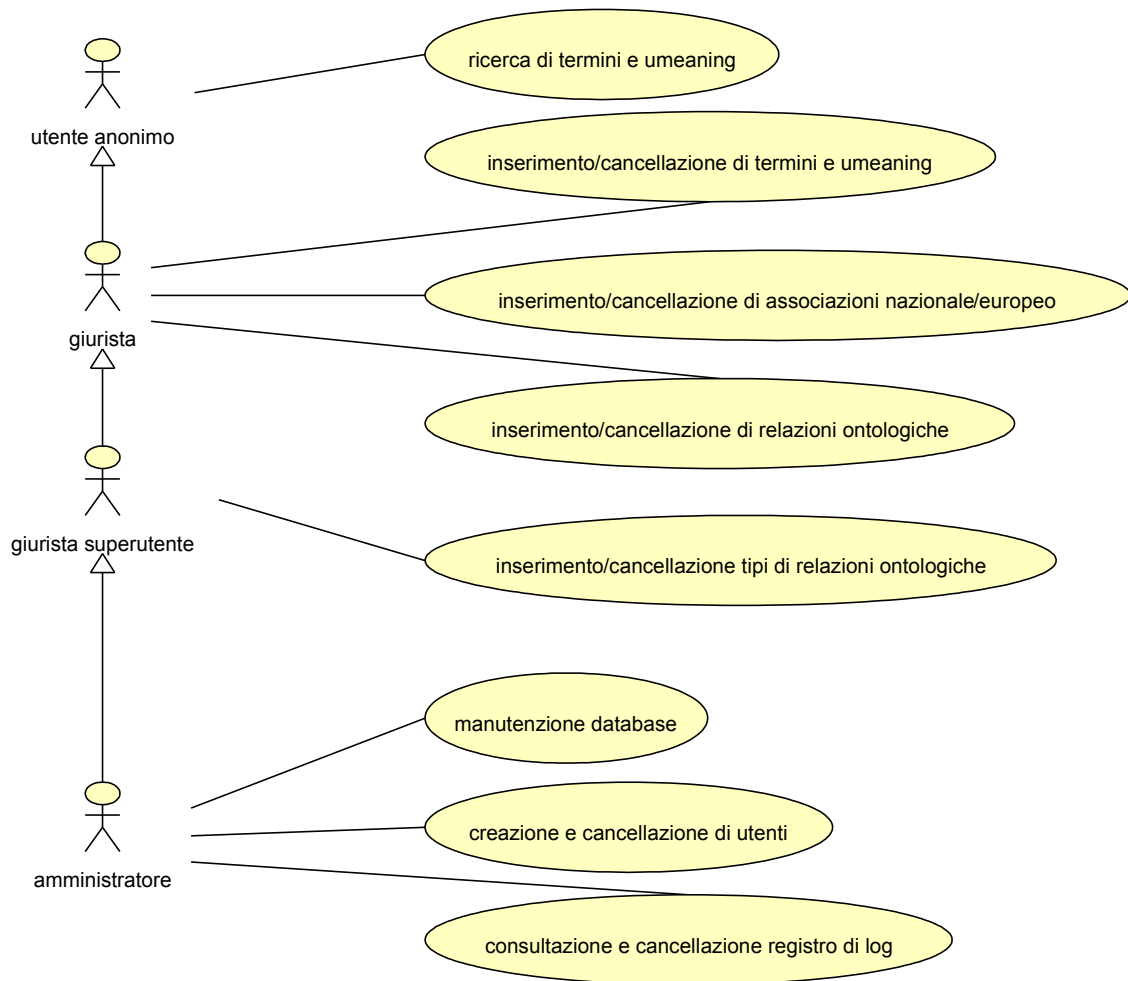


Figura 4.2 Diversi casi d'uso a seconda del tipo di utente

Il dizionario sarà consultabile via web da chiunque, quindi per una semplice consultazione non sarà necessario autenticarsi.

Gli utenti principali a cui il prodotto è destinato sono giuristi, avvocati e magistrati, a cui non devono essere 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, che saranno deputati a decidere più in dettaglio la struttura delle ontologie, oltre alle funzioni precedentemente elencate, dovranno avere la facoltà aggiuntiva di poter aggiungere e rimuovere dei tipi di relazione ontologica.

La terza categoria di utenti è quella degli amministratori, ai quali è richiesta una buona conoscenza del funzionamento del sistema, i quali devono poter gestire aspetti importanti del sistema sebbene non direttamente connessi alla gestione dei dizionari, in particolare:

- Eseguire una periodica manutenzione del database, ad esempio backup dei dati e deframmentazione
- Creare nuovi utenti, modificarli oppure cancellarli
- Consultare una tabella di log che registrerà le operazioni compiute sul dizionario.

Le quattro categorie di utenti saranno una estensione dell'altra: un utente giurista potrà fare ciò che fa un utente anonimo, un utente giurista superutente potrà fare ciò che fa un utente giurista e così via.

4.1.4 Modellazione dinamica

Per definire meglio le operazioni per la visualizzazione delle informazioni contenute in un'istanza delle classi sopra descritte, nei paragrafi successivi si farà uso di un diagramma di attività che descrive uno scenario di utilizzo di un ipotetico utente che compie operazioni legate alla consultazione.

Inoltre, il modello precedentemente descritto per essere valido ha bisogno che alcune proprietà vengano sempre soddisfatte, quindi è necessario che lo stato del sistema venga sempre controllato dopo che sono state eseguite operazioni che potrebbero potenzialmente portare ad uno stato incoerente. In particolare:

- Ogni termine deve essere definito sempre da almeno un umeaning, quindi non devono esistere termini senza umeaning.
- Ogni umeaning deve essere associato sempre ad almeno un termine, quindi non devono esistere umeaning senza termine.

Per quanto riguarda il dizionario europeo, deve essere soddisfatta una proprietà aggiuntiva, cioè ogni traduzione di ogni umeaning deve essere associata ad almeno un termine della stessa lingua.

4.1.4.1 Consultazione dizionari

In figura 4.3 viene mostrato il diagramma di attività che descrive un'operazione di consultazione del dizionario nazionale: la prima operazione è sempre un'operazione di ricerca, che può essere eseguita sia sul nome del termine che sui campi dell'umeaning. Se non è stato trovato nulla la consultazione termina, se invece è stato trovato qualcosa, i risultati saranno delle coppie termine/umeaning. A questo punto l'utente può decidere di visualizzare i dettagli dei termini trovati oppure degli umeaning trovati. Nel primo caso, l'utente potrà scegliere soltanto di visualizzare l'elenco degli umeaning associati al termine selezionato e la consultazione termina. Se invece l'utente decide di visualizzare i dati di uno degli umeaning trovati, successivamente potrà scegliere se visualizzare tutti i termini associati a quell'umeaning, la sua ontologia oppure le sue associazioni con umeaning di livello europeo.

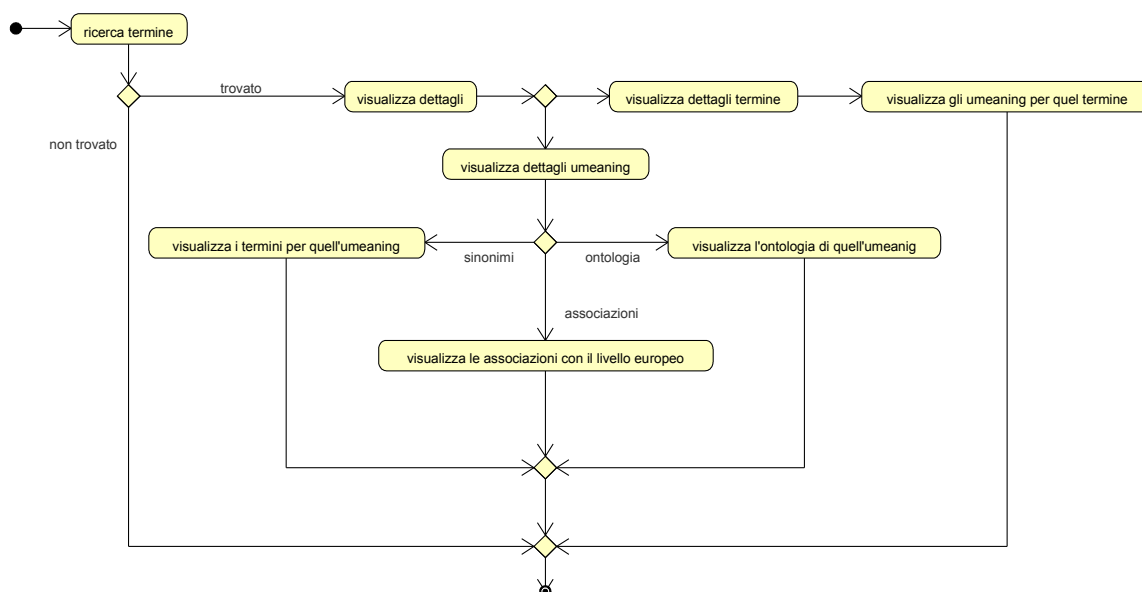


Figura 4.3 Consultazione del dizionario nazionale

La figura 4.4 mostra la stessa operazione compiuta sul dizionario europeo: l'unica differenza è rappresentata dalla disponibilità di un'operazione aggiuntiva nella consultazione degli umeaning, cioè la visualizzazione di tutte le traduzioni inserite per l'umeaning selezionato.

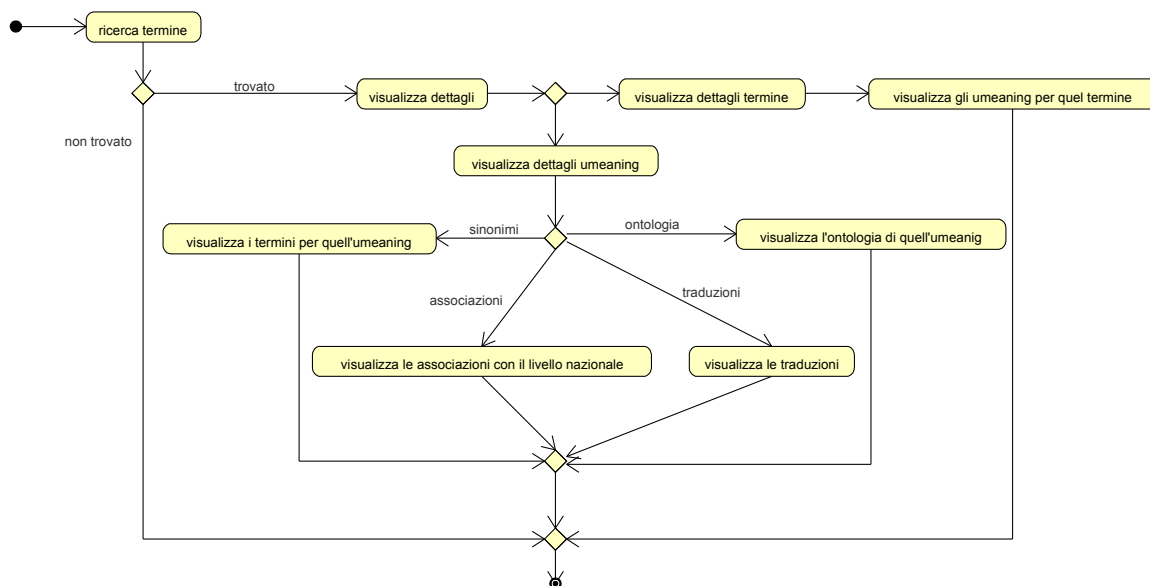


Figura 4.4 Consultazione del dizionario europeo

4.1.4.2 Inserimento

Dopo che è stato compiuto un inserimento in uno dei due dizionari, devono ancora essere valide le condizioni citate nel paragrafo 4.1.4.

La figura 4.5 descrive il controllo che è necessario eseguire: se viene inserito un nuovo termine si controlla che esista almeno un umeaning ad esso associato; allo stesso modo, se viene inserito un nuovo umeaning, si controlla che esista almeno un termine ad esso associato: ciò vuol dire che si sta inserendo un nuovo umeaning per un termine già esistente (o viceversa) oppure che il termine e l'umeaning sono stati inseriti nello stesso momento (cioè, come verrà specificato nella fase di progettazione del database, inseriti durante la stessa transazione).

Se la risposta è affermativa si procede con l'inserimento, altrimenti questo viene annullato e si produce un messaggio di errore.

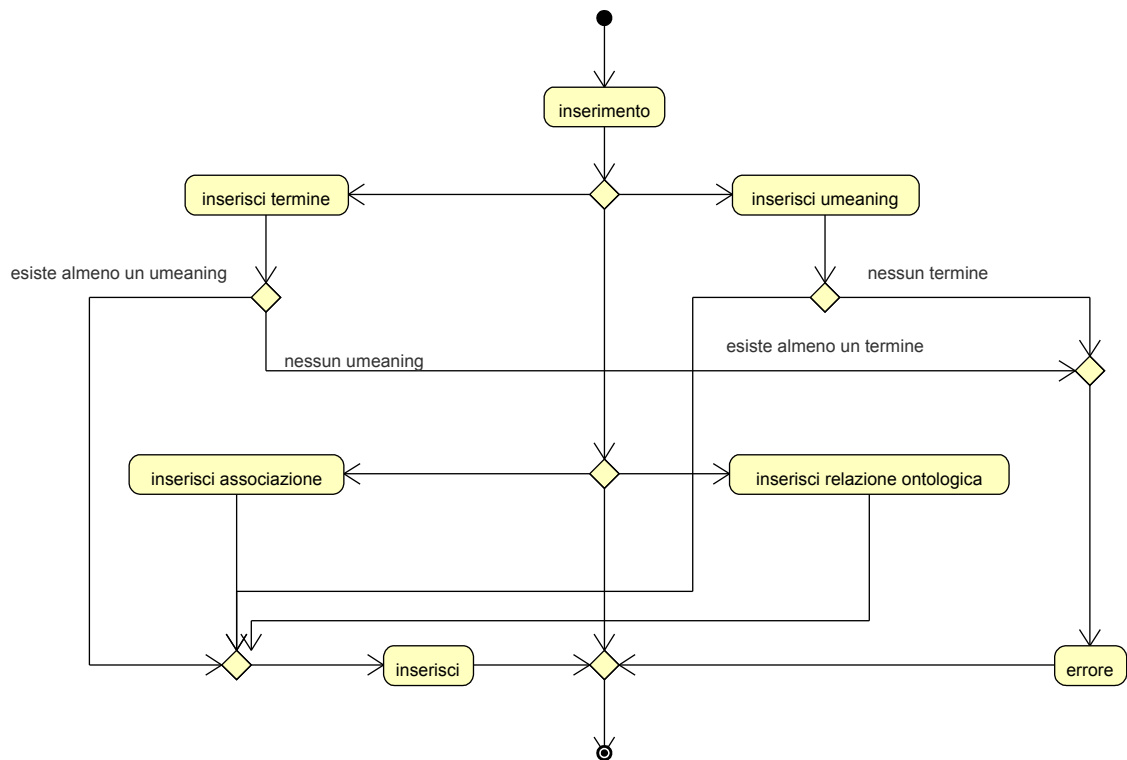


Figura 4.5 Inserimento di un nuovo termine oppure di un nuovo umeaning

La figura 4.6 descrive il controllo aggiuntivo che viene fatto prima dell'inserimento di una nuova traduzione: essa asserisce che la traduzione di un umeaning viene effettivamente inserita solo se a questa corrisponde almeno un termine della stessa lingua giuridica (anche questo dovrà venire inserito all'interno della stessa transazione).

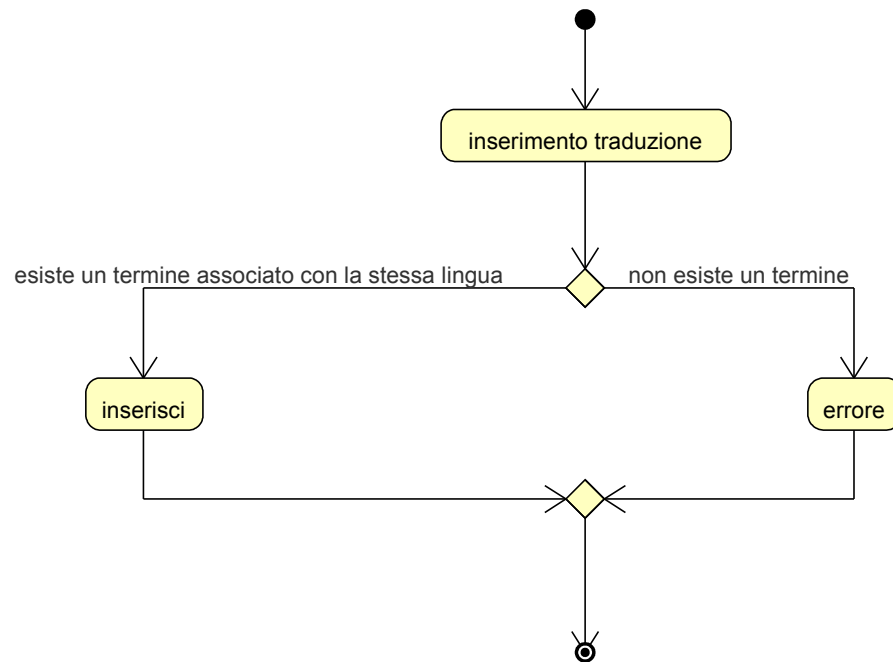


Figura 4.6 Inserimento di una nuova traduzione su un umeaning europeo

4.1.4.3 Cancellazione

Per poter mantenere uno stato coerente, è necessario controllare se, dopo che è stato cancellato un termine, gli umeaning ad esso associati siano ancora effettivamente associati almeno ad un altro termine e, se così non è, occorre cancellare tutti quelli che non sono più associati ad alcun termine. Allo stesso modo, dopo la cancellazione di un umeaning occorre cancellare tutti i termini che non sono più associati ad alcun umeaning. Nel dizionario europeo, l'integrità delle traduzioni è gestita allo stesso modo.

La figura 4.7 rappresenta un diagramma di attività che mostra il controllo che viene fatto; allo stesso modo, la figura 4.8 rappresenta un diagramma di stato che mostra come da uno stato non valido di debba sempre tornare ad uno valido.

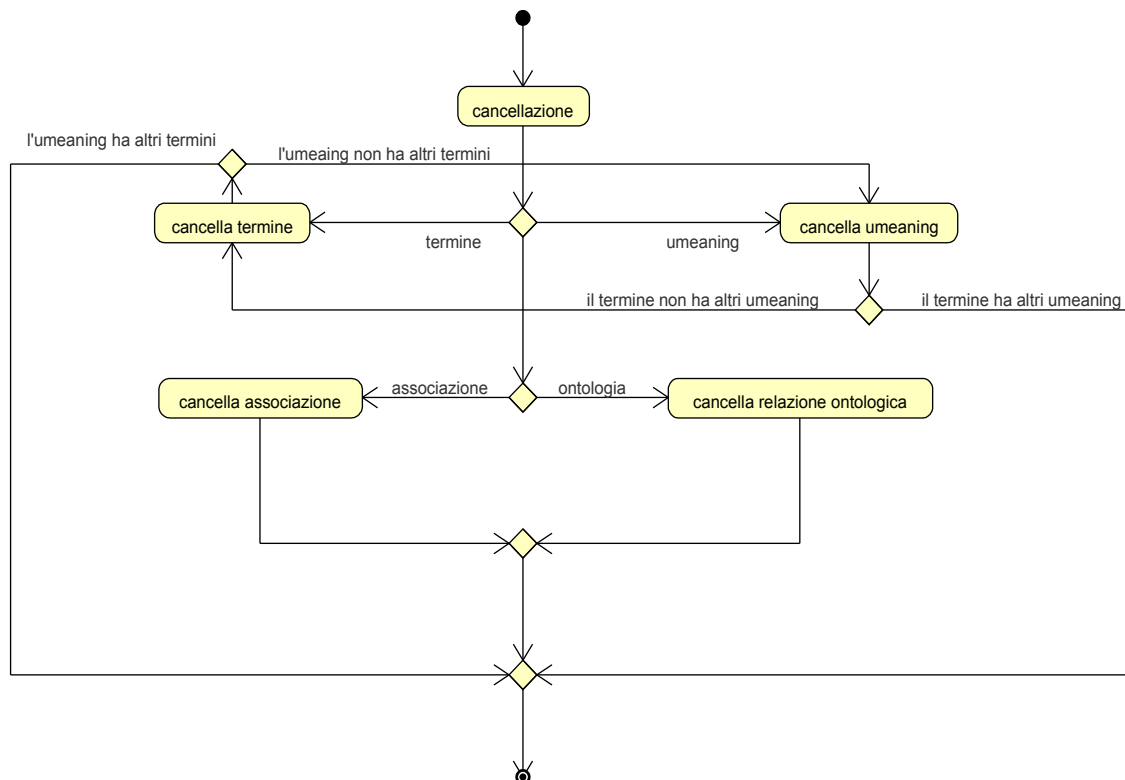


Figura 4.7 Cancellazione di un termine oppure di un umeaing

4.1.4.4 Inserimento e cancellazione ontologia e associazioni

Per quanto riguarda l'inserimento e cancellazione delle relazioni ontologiche e delle associazioni fra umeaing nazionali ed europei, in entrambi i casi l'unico controllo che è necessario eseguire è che i due umeaing che si desidera associare esistano effettivamente e che abbiano la stessa lingua giuridica.

4.2 Specifica dei requisiti

4.2.1 INTRODUCTION

Seguiremo lo standard IEEE per un documento di specifica dei requisiti come definito da [27].

4.2.1.1 Purpose

Lo scopo di questo documento è di descrivere i requisiti di un sistema per la gestione di un dizionario concettuale giuridico, comparativo tra l'ordinamento europeo e gli ordinamenti nazionali.

4.2.1.2 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.

4.2.1.3 Notational Conventions

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

4.2.2 OVERALL DESCRIPTION

4.2.2.1 Product Perspective

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

4.2.2.2 Product Functions Overview

Dopo essere state approvate dal Parlamento, le direttive Europee vengono tradotte in cinque lingue (inglese, italiano, francese, tedesco e spagnolo), tuttavia i significati delle singole parole variano da paese a paese, e possono eventualmente anche non essere presenti in ogni ordinamento. Un giurista, quindi, per interpretare correttamente il testo di una direttiva europea, deve fare un lungo lavoro di comparazione con l'ordinamento nazionale, esaminando come sono state recepite direttive precedenti.

Il sistema serve quindi per creare un dizionario comparativo tra i concetti espressi nel sistema giuridico europeo con i corrispondenti concetti espressi nel sistema giuridico nazionale, ovvero come sono espressi nel testo di leggi esistenti dello Stato.

Alcuni *umeaning* sono in relazione anche con altri dello stesso livello, ad esempio l'*umeaning* che definisce “clausola” sarà definito come facente parte dell'*umeaning* che definisce “contratto”, in quanto un contratto può essere formato da diverse clausole. Quindi, un'importante funzionalità che il sistema dovrà offrire, sarà la rappresentazione delle relazioni ontologiche fra gli *umeaning*.

L'interfaccia web dovrà offrire un semplice sistema per la navigazione degli *umeaning* basata sulla ricerca, per consentire la gestione di una quantità di dati anche molto grande.

Attraverso l'interfaccia web dovrà essere possibile inoltre inserire, modificare e cancellare nuovi *umeaning*, così come relazioni fra *umeaning* nazionali ed europei e relazioni ontologiche fra *umeaning* dello stesso livello. Ogni modifica effettuata dovrà essere registrata in un file di log, per tenere traccia delle modifiche e soprattutto da chi sono state fatte.

L'interfaccia dovrà essere tradotta in più lingue, e l'utente dovrà poter scegliere in ogni momento in quale lingua effettuare la navigazione.

Per accedere all'interfaccia web sarà necessario autenticarsi, il sistema dovrà quindi aver una propria gestione interna degli account utente; inoltre diverse tipologie di utente avranno diversi privilegi: alcuni utenti potranno solo consultare il database in sola lettura, altri potranno inserire soltanto *umeaning*, altri potranno creare utenti e visualizzare il file di log e così via.

4.2.2.3 User Characteristics

Sono previsti principalmente 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.

4.2.3 SPECIFIC REQUIREMENTS

4.2.3.1 *Inputs and Outputs*

Il sistema riceve tutti i suoi input unicamente attraverso il metodo http post [28] (quindi presentato all'utente sotto forma di form html) oppure codificate nell'URL [22].

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 di tutti i campi delle form utilizzate per l'input.

Ricerca

- 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.

Inserimento/modifica Umeaning

- 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.
- Riferimenti: casella di testo in cui inserire riferimenti alle leggi (se il livello è nazionale) oppure alle direttive (se il livello è europeo) in cui il termine appare.

Inserimento di un nuovo allegato

- Id Umeaning: l'identificativo numerico dell'umeaning.
- Lingua giuridica: la lingua giuridica dell'umeaning, questo è necessario solo nel livello europeo.
- Livello: il livello nazionale oppure europeo, codificato nell'url.
- File: il file di cui eseguire l'upload, presentato all'utente mediante un campo input di tipo file definito in [29].

Modifica ontologia

L'inserimento di una nuova relazione fra due umeaning sarà presentato all'utente con una pagina di ricerca simile alla pagina definita nel paragrafo 4.2.3.1, da cui sarà però possibile specificare i seguenti parametri aggiuntivi:

- Umeaning padre: id numerico che identifica l'umeaning padre.
- Umeaning figlio: id numerico che identifica l'umeaning figlio.
- Lingua giuridica: questo parametro sarà richiesto solo quando si modifica un'ontologia di umeaning di livello europeo, in quanto solo nel livello europeo uno stesso umeaning verrà tradotto in più lingue e ogni lingua avrà un'ontologia diversa.
- Tipo di relazione: la relazione che lega l'umeaning padre all'umeaning figlio, ad esempio "IS_A" oppure "PART_OF".

Creazione e modifica degli utenti

- Login: casella di testo.
- Password: casella di testo.
- Nome: casella di testo.
- Cognome: casella di testo.
- Indirizzo e-mail: casella di testo.
- Amministratore: checkbox che definisce se l'utente creato potrà creare a sua volta utenti e visualizzare i log.
- Navigazione sito: checkbox che definisce se l'utente potrà visualizzare il sito.
- Inserimento umeaning: checkbox che definisce se l'utente potrà inserire umeaning.
- Inserimento tipi di relazione: checkbox che definisce se l'utente potrà inserire nuovi tipi di relazioni ontologiche.

Visualizzazione del log

La visualizzazione del log sarà presentata all'utente mediante una pagina di ricerca, in cui sarà possibile specificare i seguenti input:

- Elementi più recenti di: cerca solo gli elementi più recenti di una certa data, specificata dall'utente mediante tre combobox in cui si inseriranno rispettivamente giorno, mese e anno.
- Elementi più vecchi di: cerca solo gli elementi più vecchi di una certa data, specificata dall'utente nella stessa maniera del punto precedente.
- Utente: textbox per restringere la ricerca solo alle entry generate da un certo utente
- Operazione: textbox per restringere la ricerca solo alle entry che si riferiscono ad un certo tipo di operazione, ad esempio “inserimento umeaning”
- Messaggio: per cercare un particolare messaggio inserito nella tabella di log.

Cancellazione di una entry dal log

l'unico input di questa funzione è un id numerico che si riferisce univocamente ad una riga della tabella contenente il log.

4.2.3.2 Functional Requirements

Ricerca dei termini inseriti

Il sistema deve poter supportare la ricerca dei termini inseriti, i loro umeaning e le loro associazioni tra livello nazionale ed europeo. Quest'operazione solitamente è disponibile sia per utenti autenticati che per utenti anonimi, tuttavia l'amministratore deve anche poter 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 o riferimenti degli umeaning ad esso associati.

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.

Inserimento di nuovi termini e umeaning

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

- Per ogni id termine, il termine associato è unico: i sinonimi vanno salvati in una tabella a parte e condividono il medesimo id.
- L'utente deve riempire tutti i campi definiti come obbligatori
- Per poter inserire nuovi umeaning è necessario essersi autenticati ed essere un utente a cui è stata esplicitamente data la possibilità di inserire umeaning

Ad ogni termine deve corrispondere uno o più umeaning, allo stesso modo ad ogni umeaning deve corrispondere uno o più termini; se ad un umeaning corrispondono più termini, questi sono considerati sinonimi.

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, questo non è possibile.

L'inserimento di nuovo umeaning può avvenire in tre 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.

Input: termine (obbligatorio solo nel caso 1), lingua giuridica (obbligatorio), livello (obbligatorio), descrizione (obbligatorio), riferimenti (obbligatorio) definiti nel paragrafo 4.2.3.1.

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 nuovi allegati

Per poter eseguire l'upload di un file che verrà associato ad un umeaning è necessario che le seguenti condizioni siano soddisfatte:

- L'utente deve essere autenticato ed avere il permesso di inserire nuovi umeaning.
- L'umeaning a cui l'allegato fa riferimento deve essere esistente.
- Il file non deve essere più grande della dimensione specificata nel file di configurazione di apache dalla direttiva LimitRequestBody.

Input: l'id dell'umeaning, il livello, la lingua giuridica se il livello è europeo, inoltre il nome e il contenuto del file codificati in una richiesta HTTP POST, di tipo multipart/form-data come definito in [29].

Output: verrà mostrata nuovamente la pagina di riepilogo dell'umeaning, oppure un messaggio di errore appropriato tra cui “accesso negato” e “impossibile effettuare l'upload”.

Cancellazione di umeaning e termini

Affinché l'utente possa cancellare dal database un umeaning oppure un termine è necessario che sia autenticato ed abbia il permesso di inserire nuovi umeaning.

Inoltre ad avvenuta cancellazione, è importante che le seguenti condizioni siano soddisfatte:

- Ogni umeaning deve essere associato ad almeno un termine della medesima lingua giuridica.
- Ogni termine deve essere associato ad almeno un umeaning della medesima lingua giuridica.

Se una delle due condizione non dovesse essere soddisfatta, gli umeaning privi di termine oppure i termini privi di umeaning dovranno venire automaticamente cancellati.

Modifica ontologia

Per poter inserire relazioni ontologiche tra diversi umeaning i seguenti vincoli devono essere soddisfatti:

- Per poter inserire nuove relazioni ontologiche, è necessario essersi autenticati ed essere un utente a cui è stata esplicitamente data la possibilità di inserire umenaning.
- Gli umeaning padre e figlio devono essere preesistenti nel database.
- Il tipo di relazione ontologica specificata deve essere esistente nel database
- Una relazione ontologica fra due umeaning è possibile solo nel caso in cui siano entrambi dello stesso livello e abbiano la stessa lingua giuridica.

Input: gli id numerici degli umeaning padre e figlio, il tipo di relazione ontologica, il livello e, nel caso quest'ultimo sia europeo, la lingua giuridica degli umeaning.

Output: una rappresentazione ontologica ad albero se è una relazione transitiva, una rappresentazione “piatta” se invece la relazione non è transitiva, oppure il messaggio di errore “accesso negato”.

Inserimento di nuove associazioni europeo/nazionale

Anche per poter inserire associazioni tra umeaning di livello europeo e umeaning di livello nazionale è necessario essersi autenticati ed avere il permesso di inserire nuovi umeaning.

Input: gli id numerici degli umeaning di livello nazionale ed europeo e la lingua giuridica.

Output: una rappresentazione grafica dell'associazione, oppure un adatto messaggio di errore tra cui “impossibile eseguire l'associazione” e “accesso negato”.

Gestione utenti

Per poter creare o modificare nuovi utenti è necessario essersi autenticati come utenti amministratori, inoltre il login non può essere duplicato per due utenti.

Input: login, password, seconda password (devono coincidere, questo per evitare errori durante l'inserimento), nome, cognome, e-mail e permessi dell'utente

Output: riepilogo sui dati dell'utente appena inserito oppure un messaggio di errore, tra cui: “accesso negato”, “non tutti i campi obbligatori sono stati compilati”, “le due password non coincidono” e “login duplicato”.

4.2.3.3 External Interface Requirements

L'interfaccia web deve seguire la specifica w3c xhtml 1.0 strict [30] e deve venire visualizzata correttamente sui principali browser web, in particolare:

- Internet Explorer versione 5.5 o superiore
- Mozilla Firefox versione 1.0 o superiore
- Netscape versione 7.0 o superiore
- Opera versione 8.0 o superiore.

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

4.2.3.4 Design Constraints

Software 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.
- 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.

Hardware constraints

Il sistema deve poter girare su piattaforme ia32 oppure sparc con almeno 500mhz e 256 megabyte di ram.

Capitolo 5 - Progettazione

La progettazione avverrà in sue distinte fasi. Verrà prima progettato il database contenente i dizionari, oltre a dati aggiuntivi indispensabili per il corretto funzionamento come ad esempio gli account utente. Successivamente verrà progettata l'applicazione di gestione del database deputata ad interagire con l'utente finale.

5.1 Architettura del sistema

Le funzionalità richieste al sistema verranno quindi in parte gestite dal database stesso ed in parte gestite dalla web-application, la quale è l'entità deputata alla comunicazione fra l'utente finale e il database, dove vengono fisicamente conservati i dati.

Il design del sistema segue la classica architettura three-tier, sviluppata per ottenere un migliore utilizzo delle risorse nelle applicazioni client/server [31]: l'interazione fra i componenti del sistema è illustrata nella figura 5.1.

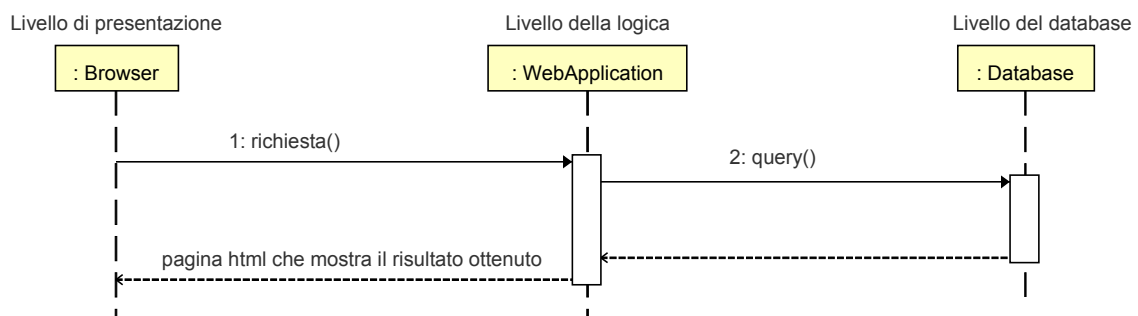


Figura 5.1 architettura di un'applicazione client/server three tier

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.

5.2 Progettazione della web-application

5.2.1 Object oriented Design

Spesso, nelle web-application, il livello della logica è suddiviso in due moduli, *User Interaction Handler* e *Business Logic Layer* [32] (vedi figura 5.2).

Lo User Interaction Handler e il Business Logic Layer saranno entrambi suddivisi in più moduli, ognuno progettato per compiere una specifica funzionalità del sistema.

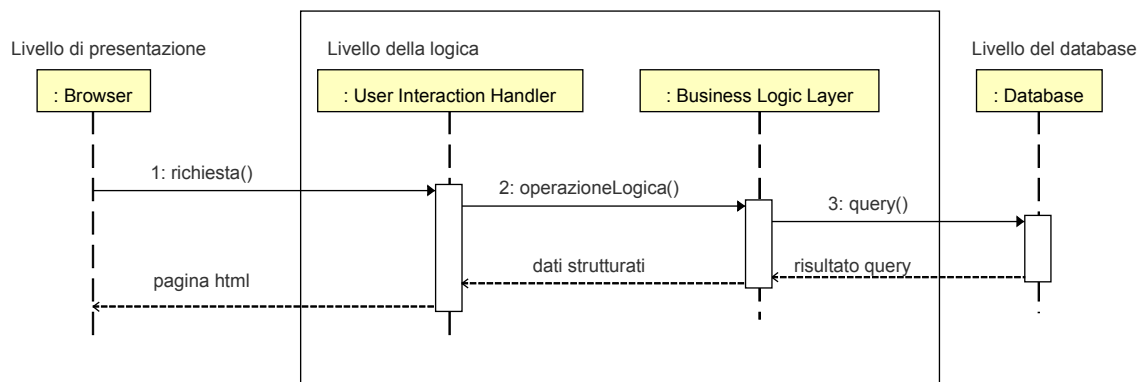


Figura 5.2 Il livello della logica suddiviso in User Interaction Handler e Business Logic

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 istanzia 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 modulo del business logic layer accederà al database tramite un livello di astrazione fornito dal linguaggio e fornirà in output allo User Interaction Handler i dati del database convertiti in un formato strutturato adatto alla manipolazione nell'applicazione, come ad esempio un'array associativo: in Syllabus useremo le funzioni di accesso a PostgreSQL fornite dal PHP come `pg_connect()`, `pg_query()` e `pg_fetch_array()`.

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 meaning, 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.

Ognuna di queste classi avrà un metodo pubblico per ogni funzione che offre. Un primo diagramma delle classi è presentato in figura 5.3, in cui per semplicità sono stati omessi gli attributi e i metodi.

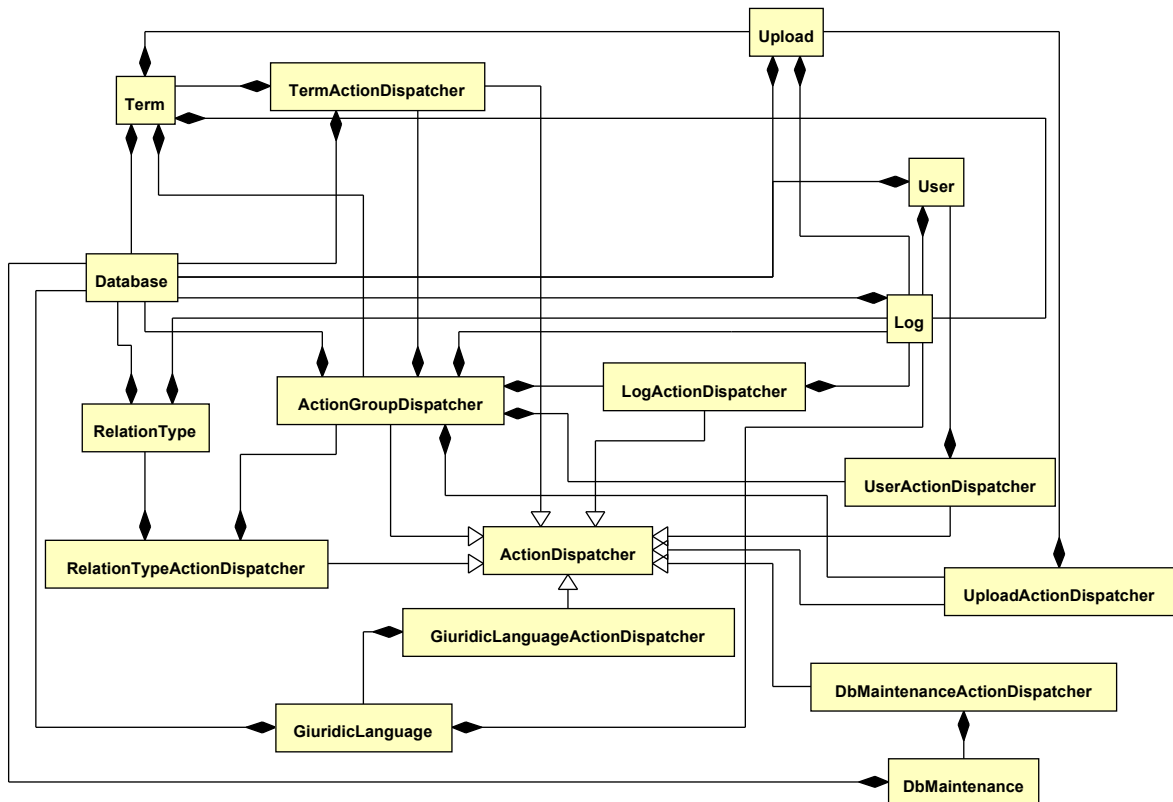


Figura 5.3 Diagramma delle classi della web application

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

1. Valida l'utente iniziando una nuova sessione oppure riprendendone una precedente (PHP conserva sul server le variabili di sessione e accede ad esse mediante un numero random che il browser passa mediante un cookie).

2. Valida il parametro URL-encoded [22] *action*, accessibile dal codice PHP mediante l'array associativo `$_GET`. Prima di tutto controlla se è un'operazione supportata dal server, se così è, viene istanziata la classe `ActionDispatcher` di cui l'azione fa parte ed eseguito il metodo atto a gestirla, il quale d'ora in poi prenderà il controllo. La figura 5.4 mostra il flusso di esecuzione dell'operazione `searchTerm`, cioè la ricerca di termini e umening mediante parte del loro testo.

A differenza di tutte le altre operazioni, queste due sono gestite direttamente dalla classe `ActionGroupDispatcher` per ragioni di prestazioni. Siccome è un'operazione compiuta al caricamento di ogni pagina, viene così evitata la creazione e la distruzione di due ulteriori oggetti ad ogni pagina visitata.

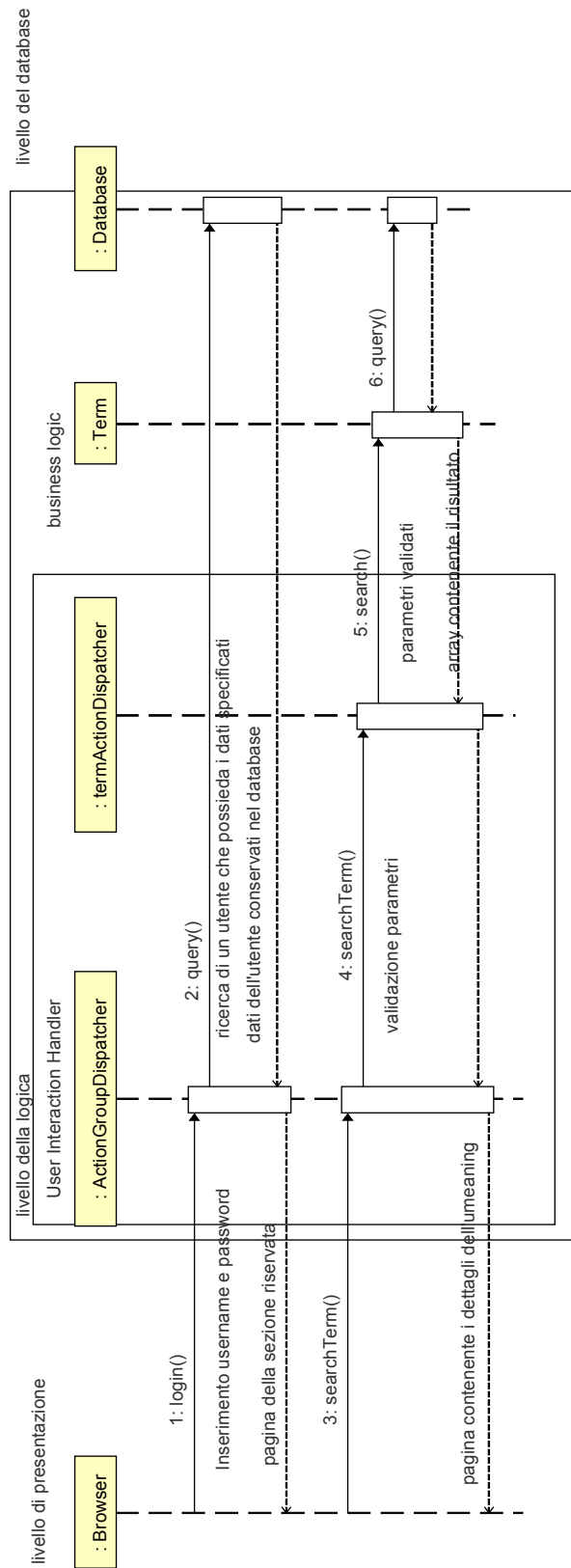


Figura 5.4 Lo scambio di messaggi fra i moduli del sistema durante la ricerca di un termine

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 5.5 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.

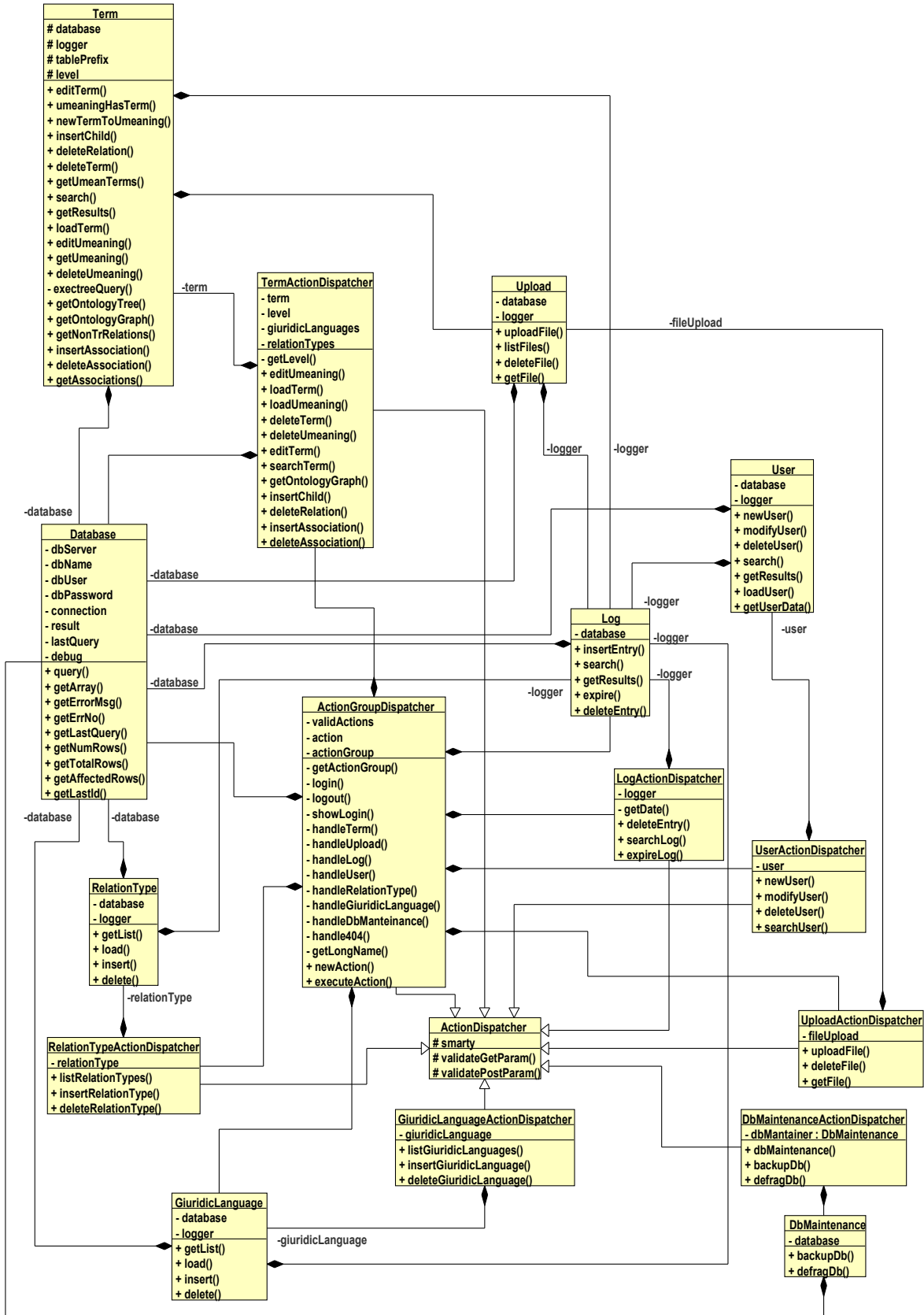


Figura 5.5 Diagramma delle classi della web-application

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, se l'utente vuole creare un nuovo termine, gli verrà mostrata una form in cui egli può inserire il nome del termine e i dati dell'umeaning ad esso associati, e ciò è supportato dal metodo *editUmeaning* della classe TermActionDispatcher, che quindi avrà bisogno di chiamare tre distinti metodi della classe 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.

Se una classe della business logic deve supportare operazioni sul database mediante più metodi, cosa che per il momento avviene solamente per la classe Term, provvederà a racchiudere tutte le sue operazioni all'interno di una transazione SQL.

I metodi delle classi della business logic si dividono in metodi per la scrittura e per la lettura dal 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, manipolabile dal codice PHP e soprattutto comprensibile al motore di template che si occuperà di presentarli in formato HTML comprensibili all'utente. Tale metodo ritorna una struttura dati più complessa: essa è una rappresentazione del risultato della query, cioè una tabella, la quale è implementata nel seguente modo: è un array con tanti elementi quante sono le righe del risultato ed ogni elemento è un *array associativo* (che in PHP è l'equivalente delle *map* di C++ e Java) i cui dati sono gli elementi della riga del risultato indicizzati per il nome della colonna. Ad esempio se l'array si chiama *result*, `$result[2]['description']` sarà il campo "description" della terza riga del risultato.

Ogni classe della business logic, contiene anche la classe *Log*, la quale si occupa di inserire una voce nella tabella di log dopo ogni operazione di modifica al database andata a buon fine.

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.

Le classi *Smarty* e *GraphViz* (per una descrizione più dettagliata di *Smarty* vedere il paragrafo 6.2.3) sono due progetti esterni e vengono utilizzate per connettere il livello dello *User Interaction Handler* al livello dell'interfaccia utente vera e propria. La classe *GraphViz* gestisce la generazione di un'immagine rappresentante il grafo dell'ontologia di un dato umeaning attraverso l'omonimo strumento di AT&T [18]. La classe *Smarty* è un motore per l'interpretazione di template HTML (anche se non è legata all'uso di HTML in particolare) ed offre un buon livello di astrazione, rendendo molto semplice effettuare modifiche anche molto consistenti all'interfaccia utente senza per questo dover intaccare la struttura dell'applicazione.

5.2.2 Package

Le classi di cui l'applicazione è composta sono ordinate all'interno della gerarchia di package descritta nel diagramma delle classi di figura 5.6.

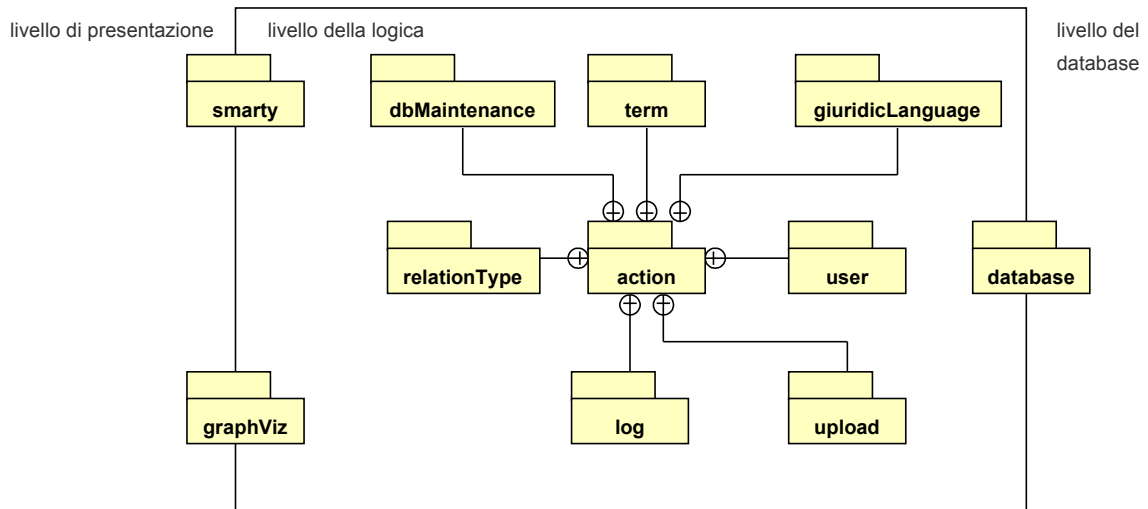


Figura 5.6 Gerarchia dei package del sistema

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.

5.3 Progettazione del database

Nei prossimi paragrafi verrà presentata l'architettura del database: verranno utilizzati dei diagrammi delle classi adattati alla modellazione entità-relazione, come definito in [33]:96 e utilizzati in alcuni software per il design dei database come DB designer 4 [34], che è il software utilizzato per disegnare lo schema di figura 5.7.

5.3.1 Modello Entità-relazione

La struttura del database segue da vicino la struttura delle classi ottenuta dalla fase di analisi nel paragrafo 4.1.2: le relazioni *european_term* e *national_term*¹ rappresentano i termini europeo e nazionale e sono connesse ai rispettivi *umeaning* contenuti rispettivamente nelle relazioni *european_umeaning* e *national_umeaning* attraverso le relazioni *european_term_to_umeaning* e *national_term_to_umeaning*, le quali implementano un'associazione M-N, in quanto ad un termine possono corrispondere più *umeaning* (molteplicità di significato) e viceversa ad un *umeaning* possono corrispondere più termini (nel caso di termini sinonimi in un dato contesto). La figura 5.7 rappresenta la struttura completa del database.

¹ Se nel database i nomi dei campi sono composti da più parole, esse sono separate da un underscore anziché mediante maiuscole, in quanto in molti database server (tra cui PostgreSQL qui utilizzato) i nomi dei campi e delle tabelle sono case-insensitive

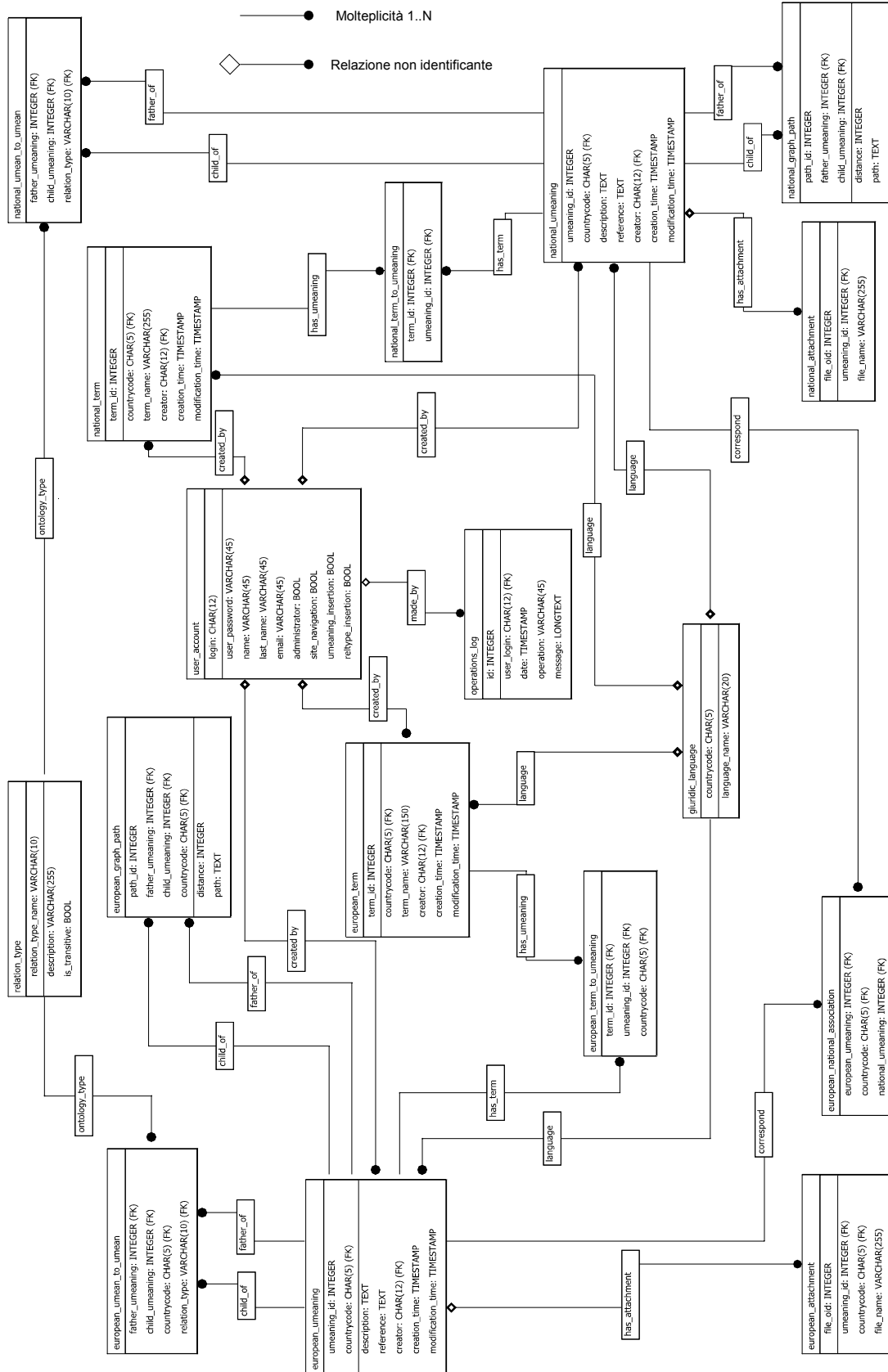


Figura 5.7 Schema relazionale del database

Come specificato nel paragrafo 4.1.2 i dizionari nazionale ed europeo sono separati, poiché, se per quanto riguarda l'ordinamento europeo uno stesso *umeaning* verrà tradotto in più lingue giuridiche, perché tratto dal testo di un'unica direttiva europea che è stata tradotta in più lingue, viceversa negli ordinamenti nazionali gli *umeaning* derivano da leggi nazionali, quindi da un'unica lingua giuridica.

Questo è implementato nel seguente modo: la chiave primaria della tabella *national_umeaning* è unicamente *umeaning_id*, un identificatore numerico univoco, mentre la chiave primaria della tabella *european_umeaning* è la coppia *umeaning_id* e *countrycode*; ciò fa sì che per un unico *umeaning_id* siano possibili più traduzioni. Per i due livelli si segue una numerazione indipendente, questo rende possibile che un *umeaning* nazionale sia definito dallo stesso identificatore di un *umeaning* europeo, quindi quando tale identificatore verrà presentato all'utente, sarà necessario identificare anche il livello, ad esempio *eu15* identificherà l'*umeaning* 15 del livello europeo, mentre *na15* identificherà l'*umeaning* 15 del livello nazionale.

La lingua giuridica verrà identificata dal *countrycode* come definito in [26]: 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 [35]. 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 (le ontologie europee e nazionali anche in questo caso sono separate in due tabelle) è tratta da Gene Ontology (vedere il paragrafo 2.2.1) con alcune importanti differenze: sono ontologie non fra termini ma fra *umeaning*, perché le ontologie dipendono dal significato che i termini assumono; inoltre per semplicità di comprensione i tipi di relazione non sono, come in Gene Ontology, essi stessi termini, ma sono conservati in una tabella a parte *relation_type*, la quale contiene un breve identificatore della relazione (ad esempio *IS_A*), la descrizione ed un valore booleano che identifica se la relazione è transitiva o no (ad esempio *IS_A* è transitiva mentre *INVERSE_OF* non lo è).

Sono simili a Gene Ontology anche le tabelle di denormalizzazione *european_graph_path* e *national_graph_path*, le quali conservano la chiusura transitiva delle relazioni transitive, anche se in questo caso è una semplice chiusura transitiva non riflessiva. È importante in queste tabelle il campo *path*, che contiene l'intero percorso fra un nodo dell'ontologia ed il suo discendente, secondo la seguente struttura: l' id di un *umeaning*, seguito da uno spazio, seguito dal nome del tipo di relazione, seguito da uno spazio, seguito dall'id di un *umeaning*; ad esempio l'*umeaning* 6 è antenato dell'*umeaning* 9 con distanza due e il loro percorso è “6 IS_A 8 PART_OF 9”, vuol dire che l'*umeaning* 9 è parte dell'*umeaning* 8, il quale è un tipo di *umeaning* 6. Queste tabelle non sono necessarie per un corretto funzionamento del sistema, perché salvano dati ridondanti, tuttavia servono per rendere più efficienti alcune query che richiedono di trovare relazioni antenato/discendente fra *umeaning*, inoltre servono ad alcuni trigger.

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 [36]: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.

5.3.2 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`.

5.4 Design di dettaglio

Di seguito sono specificati gli algoritmi concettuali che seguono le principali funzioni della web-application. Gli algoritmi sono dati in Process Design Language (PDL) [37]:337-340.

5.4.1 Flusso di esecuzione

Qui di seguito è riportato l'algoritmo che il programma segue ogni volta che l'utente carica una pagina sul browser, cioè esegue un'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 GET² cioè codificata nell'URL, ad esempio `index.php?action=searchTerm` indicherà di eseguire l'azione `searchTerm`.

5.4.2 Autenticazione utenti

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

² È stata usata la notazione utilizzata da PHP: le variabili codificate nell'URL sono contenute nell'array associativo `_GET`

```

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

```

5.4.3 Azioni valide

Come accennato prima, ActionGroupDispatcher controllerà se l'azione specificata dall'utente è un'azione valida e solo in questo caso proverà ad eseguirla: 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 [18].
- 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.

Capitolo 6 - Implementazione

6.1 Strumenti utilizzati

6.1.1 PHP

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

PHP (che sta per 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.

Per lo sviluppo di Syllabus è stata utilizzata la recente versione 5 di PHP, perché offre alcune importanti migliorie per quanto riguarda la programmazione orientata agli oggetti [38].

Nella versione 4 di PHP, gli oggetti venivano sempre passati per copia e questo poteva causare problemi di prestazioni, soprattutto in termini di utilizzo di memoria nel caso si lavorasse su istanze di classi molto complesse.

In PHP 5, invece gli oggetti sono passati per riferimento evitando tali problemi; inoltre il supporto alla programmazione orientata agli oggetti è più completo: ora è possibile la definizione di membri pubblici e privati (in PHP4 i membri erano sempre pubblici) ed è stato aggiunto il supporto all'ereditarietà.

6.1.2 PostgreSQL

Si è voluto scegliere un database server che da una parte fosse disponibile gratuitamente senza costi di licenza aggiuntivi, dall'altra deve supportare un buon insieme di funzioni tipiche di database commerciali, come vincoli di chiave esterna e transazioni.

PostgreSQL offre una buona serie di feature, tra cui check complessi, vincoli di chiave esterna, trigger e stored procedure scritte in una variante di PL/SQL e il supporto alle transazioni con lock a livello di riga.

6.1.3 Web-server e sistema operativo

Il sistema è stato sviluppato per funzionare sul web-server Apache ed è stato testato sulle versioni 1.3 e 2.0 in ambienti Microsoft Windows e Linux. Sebbene non sia ancora stato testato su altri sistemi, portarlo su altri unix supportati da PHP, Apache e PostgreSQL quali Sun Solaris e FreeBSD non dovrebbe presentare particolari problemi.

6.2 Implementazione della web-application

6.2.1 Le azioni e i gruppi di azioni: ActionGroupDispatcher

La classe ActionGroupDispatcher funziona nel seguente modo.

Le azioni valide sono conservate nel membro privato validActions: esso è un array associativo le cui chiavi sono dei nomi mnemonici per i gruppi di azioni. Ogni gruppo corrisponde ad una classe deputata a gestire tale gruppo di azioni, il valore corrispondente ad un gruppo è a sua volta un array contenete la lista di tutte le azioni valide per quel gruppo. I gruppi sono login, handleTerm, handleUpload, handleLog, handleUser, handleRelationType, handleGiuridicLanguage, handleDbMaintenance e handle404.

Quando l'utente specifica un'azione per prima cosa si controlla se essa è contenuta nell'array `validActions`, dopo di che viene eseguito il metodo privato di `ActionGroupDispatcher` deputato a gestire quel gruppo di azioni. Tale metodo ha il medesimo gruppo di azioni e viene chiamato mediante la tecnica della *funzione variabile* ([39]: capitolo 17) che il PHP offre: se ad esempio la variabile `$action` contiene la stringa "handleTerm", una chiamata con la sintassi `$action()` sarà equivalente ad una chiamata con la sintassi `handleTerm()`. Per chiamare i metodi di un oggetto si utilizza una sintassi simile. Questa tecnica evita il bisogno di ricorrere a lunghi costrutti *switch* offrendo così una migliore leggibilità.

I metodi privati gestori dei gruppi di azioni eseguono tutti i medesimi passi: si include il file contenente la classe che gestisce quel determinato gruppo di azioni e se ne crea una nuova istanza, dopo di che eseguono il metodo deputato a gestire l'azione specificata sempre mediante la tecnica della funzione variabile. Ad esempio, il metodo `handleTerm` per la gestione delle operazioni inerenti il dizionario (termini, umeaning e ontologia) utilizza il seguente codice:

```

1   require
2   './components/action/term/TermActionDispatcher.php';
3   $method = $this->action;
4   $actionDispatcher =
5       new TermActionDispatcher($this-
6   >getLongName($this->action));
7   /*questa condizione non dovrebbe mai verificarsi*/
8   if( !method_exists($actionDispatcher, $method) )
9       throw new Exception('invalid action');
10
11  $actionDispatcher->$method();

```

6.2.2 Implementazione delle capabilities

Le capabilities sono i permessi che ogni utente ha e servono ad identificare le diverse categorie di utenti. Sono dei valori booleani conservati nella tabella `user_account` nei campi `administrator`, `site_navigation`, `umeaning_insertion`, `retype_insertion`, che definiscono rispettivamente se l'utente è amministratore, se può navigare nel sito, se può modificare il dizionario e se può inserire/cancellare tipi di relazioni ontologiche.

Il sistema, dopo aver eseguito l'autenticazione dell'utente, legge dal database questi valori e li salva in altrettante costanti:

- `__CAP_ADMINISTRATOR`
- `__CAP_SITE_NAVIGATION`
- `__CAP_UMEANING_INSERTION`
- `__CAP_RELTYPE_INSERTION`

Vengono utilizzate delle costanti in primo luogo perché l'accesso da parte del codice è molto veloce rispetto all'utilizzo di strutture dati più complesse ed occupa meno memoria, inoltre le costanti in PHP sono sempre globali, quindi in qualsiasi parte del codice per controllare se l'utente può eseguire una determinata azione sarà sufficiente un semplice *if* sul valore della costante appropriata. Infine, essendo costanti non c'è il rischio di modificarle in parti del codice che non siano la loro definizione, causando così problemi di sicurezza.

6.2.3 La generazione di codice HTML: Smarty

Come detto in precedenza, è stata mantenuta una chiara distinzione tra il livello della logica e il livello dell'interfaccia. Per fare questo è stato utilizzato il motore di template Smarty [40]. I template utilizzati da smarty sono delle pagine HTML³ 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* ([40]: 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).

Smarty si presenta come una classe PHP da usare liberamente all'interno del programma, i suoi metodi principali sono:

- `assign(string varname, mixed variabile)`: assegna una variabile di nome *varname* in modo che sia accessibile dal template.

³ I template Smarty non sono limitati all'utilizzo di HTML in particolare, basta che sia un formato plaintext. Si potrebbe ad esempio costruire un template RDF.

- `display(string templatename)`: mostra in output il template contenuto nel file `templatename`: da chiamare solo dopo che sono state assegnate tutte le variabili contenenti i dati che si vogliono mostrare in output.

L'utilizzo dei template ha buone prestazioni, perché Smarty esegue il parsing solo la prima volta, in cui traduce il template in un file PHP valido che viene salvato in una directory di cache specificata nel membro `compile_dir`. Nelle esecuzioni successive verrà solamente eseguito il file PHP generato.

6.2.4 Considerazioni sulla sicurezza

La sicurezza è un aspetto importante per quanto riguarda le applicazioni distribuite, in questo caso sono stati presi alcuni accorgimenti:

- È stato vietato l'accesso al client HTTP a tutti i file che servono al sistema ma non devono poter essere scaricabili direttamente mediante l'utilizzo dei file di configurazione locali di apache `.htaccess`. Se si volesse utilizzare il sistema su altri web-server, potrebbe essere necessaria una configurazione apposita.
- Il sistema è progettato per funzionare con la direttiva del file di configurazione `php.ini` `register_globals`⁴ disattivata: è importante tenerla disattivata per impedire attacchi basati sulla sovrascrittura di variabili utilizzate dal programma.
- Gli id di sessione sono legati all'indirizzo IP della macchina da cui l'utente effettua il login: questo può risultare scomodo nel caso di accesso da linee dial-up in cui l'indirizzo IP assegnato è differente ogni volta, ma impedisce attacchi di tipo *Session Fixation* [41].
- Nella pagina di visualizzazione degli `umeaning` è presente un link all'indirizzo email dell'utente che ha creato l'`umeaning`, tuttavia è solo visibile per utenti autenticati e non per l'utente anonimo: questo impedisce a programmi automatici di recuperare gli indirizzi email degli utenti a scopo di *spam*.

⁴ La direttiva `register_globals` nel file di configurazione `php.ini` fa sì che ogni variabile mandata mediante l'encoding nell'URL o mediante post data venga automaticamente registrata come variabile globale. Ad esempio una URL contenente `index.php?action=searchTerm` causerebbe la creazione della variabile globale `action` contenente la stringa `searchTerm`

6.3 Implementazione dei controlli di integrità del database

I controlli nell'integrità del database sono stati realizzati attraverso una serie di trigger.

I trigger in PostgreSQL hanno una struttura particolare ([42]:capitolo 32). Il body della procedura deve essere posizionato in una stored procedure particolare, chiamata da PostgreSQL *trigger function*, la cui particolarità è di avere il tipo di dati *trigger* come valore di ritorno, mentre nella definizione del trigger viene chiamata tale stored procedure e non si può quindi mettere codice PL/SQL direttamente nella sua dichiarazione.

La dichiarazione di un trigger ha quindi la seguente struttura:

```

1 create function function_name() returns trigger as
  $BODY$
2 declare
3   --variabili utilizzate;
4 begin
5   --corpo della procedura
6 end
7 $BODY$ language plpgsql;
8
9 create trigger trigger_name
10before insert on table_name
11for each row execute procedure function_name();

```

Le righe da 1 a 7 definiscono la trigger function (la riga 7 specifica che il linguaggio utilizzato è la variante di PL/SQL che utilizza PostgreSQL, infatti sono supportate anche stored procedure in linguaggi diversi, come Perl, Python e Tcl), le righe da 9 a 11 definiscono il trigger.

6.3.1 Trigger di controllo dei dati

Lo schema di Syllabus possiede 16 trigger function, di cui 6 sono deputate alla validazione dei dati inseriti e sono riconoscibili dal prefisso *check_* nel loro nome, mentre le altre 10 eseguono update sul database dopo alcune modifiche sui dati, per mantenere l'integrità dei dati e sono riconoscibili dal prefisso *update_* oppure *delete_*.

I trigger di controllo dei dati eseguono particolari controlli impossibili da realizzare mediante check in quanto si basano sul risultato di query select; se il controllo fallisce l'inserimento viene impedito. Le trigger function sono le seguenti:

- `check_association_countrycode`: controlla che gli `umeaning` nazionali siano associati ad `umeaning` europei della stessa lingua giuridica
- `check_european_cycle` e `check_national_cycle`: controllano che i grafi delle ontologie nazionali ed europee non contengano cicli
- `check_european_term_to_umean_countrycode`,
`check_national_term_to_umean_countrycode`: controllano che i termini nazionali ed europei siano associati ad `umeaning` della medesima lingua giuridica.
- `check_national_ontology_countrycode`: controlla che gli `umeaning` di un'ontologia nazionale siano tutti della medesima lingua giuridica. Questo controllo è necessario solo nel caso nazionale perché date le diverse proprietà degli `umeaning` europei la tabella `european_umeaning_to_umeaning` contiene anche la lingua giuridica ed è quindi sufficiente un semplice controllo di chiave esterna.

6.3.2 Trigger di update

I trigger di update causano azioni referenziali innescate volte a preservare l'integrità dei dati, in casi in cui il modello relazionale non è sufficiente a descriverli. Ricordiamo dal paragrafo 4.1.4 che un termine deve sempre essere associato ad almeno un `umeaning` e viceversa un `umeaning` deve sempre essere associato ad almeno un termine; inoltre le due tabelle `graph_path` devono sempre in ogni momento rappresentare delle chiusure transitive corrette.

In particolare le trigger function di update sono:

- `delete_empty_european_term` e `delete_empty_national_term`: cancellano i termini che non sono più associati ad alcun `umeaning` e gli `umeaning` che non sono più associati ad alcun termine.
- `delete_european_graph_nodes` e `delete_national_graph_nodes`: cancellano dalla chiusura transitiva tutti i percorsi contenenti `umeaning` che sono stati cancellati.
- `delete_european_graph_path` e `delete_national_graph_path`: cancellano dalla chiusura transitiva tutti i percorsi contenenti relazioni che sono state cancellate dalle tabelle `umean_to_umean`

- `delete_largeobject`: cancella un file di cui si era fatto l'upload (vedere il paragrafo 6.3.3)
- `update_european_graph_path` e `update_national_graph_path`: aggiornano la chiusura transitiva dopo l'inserimento di nuove relazioni in `european_umean_to_umean` e `national_umean_to_umean`
- `update_relations_transitivity`: questa funzione viene eseguita dopo un update della tabella `relation_type`. Si occupa dell'inserimento nella chiusura transitiva di tutte le relazioni che sono diventate transitive e della rimozione di tutte le relazioni che non sono più transitive.

6.3.3 L'upload dei file

Per poter gestire meglio i file che vengono caricati sul server come “allegati degli umeaning”, invece di salvarli direttamente nel filesystem del server si è scelto di caricarli nel database come *large_objects*, principalmente per due motivi. In questo modo è possibile gestire automaticamente la cancellazione dei file per gli umeaning che sono stati cancellati, inoltre siccome non si deve scrivere sul filesystem del server si ha una maggior sicurezza.

PostgreSQL offre un meccanismo per salvare all'interno del database file di tipo arbitrario di grandi dimensioni ([42]:capitolo 41.19). Essi vengono salvati nella pseudotabella *lo_largeobject*, i cui campi sono *loid*, *pageno* e *data*. Il campo *loid* è un oid di PostgreSQL [42]: capitolo 8.12⁵, il campo *pageno* è il numero di pagina e il campo *data* sono i dati di quella pagina: i file vengono infatti suddivisi in più pagine (di dimensione 1/4 del blocco del disco, solitamente 2 kilobyte), conservate una per riga.

I *pg_largeobject* possono venire manipolati attraverso una serie di stored procedure, quali *lo_creat*, *lo_import*, *lo_write* e *lo_unlink*, accessibili dal PHP mediante opportune funzioni di accesso a PostgreSQL.

In Syllabus, le tabelle *european_attachment* e *national_attachment* contengono infatti non i file ma solo dei riferimenti ai loro oid. Quando un umeaning viene cancellato, la trigger function `delete_largeobject` cancella da *pg_largeobject* i file con l'oid associato mediante le tabelle *attachment* all'id dell'umeaning cancellato.

⁵ PostgreSQL identifica tutti i suoi oggetti interni, quali database, tabelle, lle righe delle tabelle, stored procedure e *largeobject* mediante un'unica numerazione, detta OID (Object Identifier)

Capitolo 7 - Possibili sviluppi futuri

Lo sviluppo di questo software rappresenta soltanto il primo passo di un progetto più ampio. La prima fase comprende la progettazione e la realizzazione di questo software, per rendere possibile la creazione di una semplice ontologia per ogni lingua comunitaria. All'inizio saranno semplici ontologie senza vincoli (mancano vincoli aggiuntivi supportati ad esempio da OWL, trattato nel paragrafo 2.2.2.2) utilizzando come sorgenti giuridiche solamente le normative riguardanti la protezione dei consumatori. Questo lavoro sarà fatto da giuristi anziché da esperti informatici di ontologie. La funzione di queste ontologie sarà di confrontare la struttura tassonomica delle differenti legislazioni e soprattutto comprendere meglio le differenze fra la legislazione europea e le varie legislazioni nazionali.

In una seconda fase esperti di ontologie riorganizzeranno il sistema in una top-ontology la quale potrebbe ad esempio essere DOLCE [4] e verrà aggiunta al sistema la gestione di vincoli tra le relazioni, come ad esempio relazioni riflessive e transitive, oppure vincoli di cardinalità. Eventualmente il dizionario sarà ampliato includendo anche altre aree della legislazione, come ad esempio tutte le norme in materia contrattuale. Allo stato attuale il programma implementa la distinzione fra relazioni transitive e non transitive. Una volta implementata la gestione dei vincoli, potrà essere realizzata l'importazione ed esportazione di ontologie in OWL, per permettere l'interoperabilità con strumenti grafici come Protégé [24].

Per quanto riguarda il programma, è iniziata la compilazione del dizionario: questa sarà una fase di test molto importante in quanto in questa fase sarà possibile trovare e correggere eventuali bug rimasti e sentire il parere degli utenti finali riguardo all'interfaccia, per poter così migliorare l'usabilità del sistema.

Capitolo 8 - Manuale d'uso di Syllabus

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 sono presenti alcuni link che permettono di scegliere la lingua dell'interfaccia (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

The screenshot shows the Syllabus application interface. At the top, there is a navigation bar with the Syllabus logo and a language selection icon. On the left, a sidebar contains menu items: Collegato come admin, logout, Ricerca termine (highlighted), gestione umeaning, Relazioni, Lingue giuridiche, Utenti, Log, and Manutenzione database. The main content area features a search form titled 'Inserisci i parametri della ricerca' with fields for 'lingua giuridica' (dropdown), 'livello' (dropdown), 'id umeaning', 'nome termine', 'descrizione', and 'riferimenti'. A 'trova' button and a 'Vai alla pagina' section with navigation buttons (Prima, Precedente, Successiva, Ultima) are also present. Below the form is a table of search results.

risultati ricerca (19 entry trovate; pagina 1 di 2)				
id umeaning	lingua giuridica	livello	termine	descrizione
eu2	italiano	europeo	comprensibile e accessibile	...tali clausole devono essere sempre redatte in modo chiaro e comprensibile.
eu2	italiano	europeo	chiaro e comprensibile	...tali clausole devono essere sempre redatte in modo chiaro e comprensibile.
eu2	english	europeo	clear and comprehensible	"...shall be provided in clear and comprehensible manner"
eu2	deutsche	europeo	Klar und verständlich	"Die Verträge müssen in klarer und verständlicher Sprache abgefasst sein"
na4	italiano	nazionale	in modo chiaro, comprensibile ed inequivocabile	"...il prestatore, salvo diverso accordo tra le parti che non siano consumatori, deve fornire in modo chiaro, comprensibile ed inequivocabile"
eu4	deutsche	europeo	Klar und verständlich	"Die Verträge müssen in klarer und verständlicher Sprache abgefasst sein"-2
eu4	italiano	europeo	inequivoco	ᖝᖝ il cui fine commerciale deve risultare in maniera inequivoca .ᖝ
na5	deutsche	nazionale	Einfach und verständlich	"Eine unangemessene Benachteiligung kann sich auch daraus ergeben, dass die Bestimmung nicht klar un
eu5	deutsche	europeo	Klar und verständlich	"Die Verträge müssen in klarer und verständlicher Sprache abgefasst sein"-3
eu5	italiano	europeo	chiaramente	"... vanno chiaramente indicati sui documenti commerciali."

Copyright ©2005

Figura 8.2 La pagina di ricerca che mostra alcuni risultati.

La pagina di ricerca illustrata in figura 8.2 presenta una form con sei 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 oppure riferimenti giuridici.

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

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).
- **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

Syllabus
Lingua interfaccia

Collegato come admin

logout

Ricerca termine

gestione umeaning

Relazioni

Lingue giuridiche

Utenti

Log

Manutenzione database

Ontologia

Grafo ontologia

[-] na9 withdrawal

- IS_A na10 cancellation (rimuovi relazione)
- IS_A na11 termination (rimuovi relazione)

Umeaning di livello nazionale na9

Azioni	Lingua giuridica	Termine	Descrizione	Riferimenti Allegati	Associazioni
Modifica Cancella Inserisci sinonimo Inserisci allegato Inserisci associazione Aggiungi figlio Crea figlio	english	<ul style="list-style-type: none"> ▪ withdrawal 	the act of taking back or away something that has been granted or possessed	...	<ul style="list-style-type: none"> ▪ eu17 withdrawal (rimuovi associazione)

Umeaning creato da [Mario Rossi](#) in data 29 novembre 2005 alle ore 16:26:28
 Ultima modifica: 29 novembre 2005 alle ore 16:26:28

Copyright ©2005

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 (vedere paragrafo 2.2.1.1). 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 [17]:84, di cui un semplice esempio è riportato in figura 8.4.

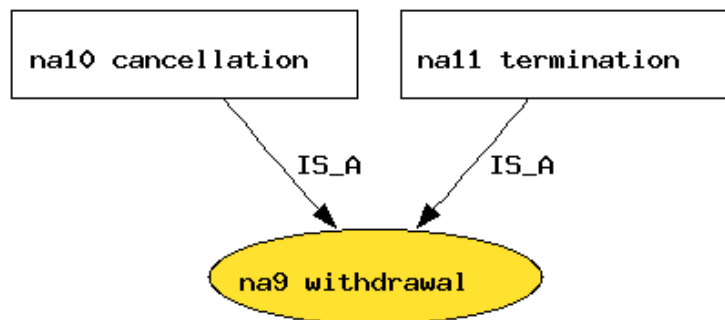


Figura 8.4 Grafo dell'ontologia corrispondente ad un umeaning del termine "withdrawal"

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: la descrizione dell'umeaning. Solitamente è il passo del testo della legge o direttiva da cui è stato tratto.
5. Riferimenti: il nome e il numero della direttiva o della legge da cui l'umeaning è stato tratto.
6. 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.
7. 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.
- 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 traduzione: 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 associazione: porta ad una pagina di ricerca in cui si può selezionare l'umeaning dell'altro livello da venire associato (paragrafo 8.3.4.6).
- Aggiungi 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).
- Crea figlio: crea una nuova coppia termine/umeaning e assegna l'umeaning appena creato come figlio di quello corrente nel grafo dell'ontologia (paragrafo 8.3.5.2).

8.3.3 Visualizzazione di un termine

The screenshot shows the Syllabus application interface. At the top, there's a navigation bar with 'Syllabus' and a 'Lingua interfaccia' dropdown menu. Below the navigation bar, there are buttons for 'Mostra traduzioni', 'Nuovo umeaning', 'Rinomina termine', and 'Cancella termine'. The main content area is titled 'Elenco degli umeaning del termine di livello europeo "chiaro e comprensibile"'. It contains a table with the following columns: Azioni, id umeaning, Lingua giuridica, Termine, Descrizione, Riferimenti, Allegati, and Associazioni. The table lists two umeaning entries for the term 'chiaro e comprensibile'. The first entry has id 'eu2' and refers to 'dir 93/13: art. 4, 2, 5, dir 99/44: art. 6, 2'. The second entry has id 'eu18' and refers to 'dir 99/44: art. 6, 2'. A sidebar on the left contains various navigation options like 'logout', 'Ricerca termine', 'gestione umeaning', etc. At the bottom, there's a footer with 'Copyright ©2005'.

Azioni	id umeaning	Lingua giuridica	Termine	Descrizione	Riferimenti	Allegati	Associazioni
Mostra ontologia Modifica Cancella Inserisci sinonimo Inserisci traduzione Inserisci allegato Inserisci associazione Aggiungi figlio Crea figlio	eu2	italiano	<ul style="list-style-type: none"> comprensibile e accessibile chiaro e comprensibile 	...tali clausole devono essere sempre redatte in modo chiaro e comprensibile.	dir 93/13: art. 4, 2, 5, dir 99/44: art. 6, 2		<ul style="list-style-type: none"> na4 in modo chiaro, comprensibile ed inequivocabile (rimuovi associazione)
Mostra ontologia Modifica Cancella Inserisci sinonimo Inserisci traduzione Inserisci allegato Inserisci associazione Aggiungi figlio Crea figlio	eu18	italiano	<ul style="list-style-type: none"> chiaro e comprensibile 	"...indicare in modo chiaro e comprensibile l'oggetto della garanzia e gli elementi essenziali"#\$230;"	dir 99/44: art. 6, 2		

Termine creato da Mario Rossi in data 14 novembre 2002 alle ore 21:31:35
Ultima modifica: 14 novembre 2002 alle ore 21:31:35

Copyright ©2005

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 grigia superiore sono:

- **Mostra traduzioni:** 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.
- **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

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.

The screenshot shows the Syllabus web application interface. At the top, there is a blue header with the 'Syllabus' logo on the left and 'Lingua interfaccia' with Italian and UK flags on the right. A sidebar on the left contains the following elements: 'Collegato come admin', 'logout', 'Ricerca termine', 'gestione umeaning' (highlighted), 'Relazioni', 'Lingue giuridiche', 'Utenti', 'Log', and 'Manutenzione database'. The main content area features a form titled 'Inserisci i dati dell' umeaning'. This form has a 'lingua giuridica' dropdown menu set to 'italiano', a 'livello' dropdown menu set to 'nazionale', a text input field for 'termine', and two text areas for 'descrizione' and 'riferimenti', each with a vertical scrollbar. An 'Inserisci' button is located at the bottom right of the form. At the bottom of the page, a blue footer contains the text 'Copyright ©2005'.

Figura 8.6 Creazione di una nuova coppia termine/umeaning

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.
- Riferimenti: il nome ed il numero della legge o della direttiva da cui l'umeaning è stato tratto.

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 Associazioni fra umeaning nazionali ed europei

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.

The screenshot shows the Syllabus application interface. At the top left, the word "Syllabus" is displayed in a blue header. At the top right, there is a logo for "Lingua interfaccia" with flags of Italy and the UK. A left sidebar contains navigation links: "Collegato come admin", "logout", "Ricerca termine", "gestione umeaning", "Relazioni", "Lingue giuridiche", "Utenti", "Log", and "Manutenzione database".

The main content area features a search form titled "Inserisci i parametri della ricerca" with the following fields: "id umeaning" (empty), "nome termine" (filled with "comprensibile"), "descrizione" (empty), and "riferimenti" (empty). A "trova" button is located at the bottom right of the form.

Below the search form is a section titled "Seleziona l'umeaning che vuoi inserire". It displays search results in a table:

risultati ricerca (2 entry trovate; pagina 1 di 1)					
seleziona	id umeaning	lingua giuridica	livello	termine	descrizione
<input checked="" type="radio"/>	eu2	italiano	europeo	comprensibile e accessibile	...tali clausole devono essere sempre redatte in modo chiaro e comprensibile.
<input type="radio"/>	eu2	italiano	europeo	chiaro e comprensibile	..tali clausole devono essere sempre redatte in modo chiaro e comprensibile.

Below the table is an "Inserisci" button. At the bottom of the page, a blue footer contains the text "Copyright ©2005".

Figura 8.7 Ricerca di un umeaning europeo da associare con uno nazionale

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.

Syllabus
Lingua interfaccia

Collegato come admin

[logout](#)

[Ricerca termine](#)

[gestione umeaning](#)

Relazioni

[Lingue giuridiche](#)

[Utenti](#)

[Log](#)

[Manutenzione database](#)

Inserisci un nuovo tipo di relazione

*nome

*descrizione

Inclusione insiemistica dell'umeaning figlio
 nell'umeaning padre

è una relazione transitiva

tipi di relazione inseriti			
azioni	nome	descrizione	transitiva
Modifica	IS_A	Inclusione insiemistica dell'umeaning figlio nell'umeaning padre	transitiva
Cancella			
Modifica	PART_OF	L'umeaning è parte della definizione del padre	transitiva
Cancella			
Modifica	INVERSE_OF	L'umeaning è il contrario del termine padre	non transitiva
Cancella			
Modifica	contraente	il soggetto che stipula un contratto	non transitiva
Cancella			

Copyright ©2005

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 8.3.4.6), 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.

The screenshot shows the Syllabus application interface. On the left is a navigation menu with items like 'Collegato come admin', 'logout', 'Ricerca termine', 'gestione umeaning', 'Relazioni', 'Lingue giuridiche', 'Utenti', 'Log', and 'Manutenzione database'. The main area contains a search form titled 'Inserisci i parametri della ricerca' with fields for 'id umeaning', 'nome termine' (containing 'contratto'), 'descrizione', and 'riferimenti', and a 'trova' button. Below the form is a section 'Seleziona l'umeaning che vuoi inserire' showing search results for 'contratto'. The results table has columns for 'selezione', 'id umeaning', 'lingua giuridica', 'livello', 'termine', and 'descrizione'. Below the table is a 'Tipo relazione:' dropdown set to 'IS_A' and an 'Inserisci' button. At the bottom, there is a 'riepilogo sui tipi di relazione disponibili' table.

selezione	id umeaning	lingua giuridica	livello	termine	descrizione
	eu6	italiano	europeo	contratto	contratti stipulati tra un professionista e un consumatore (Art. 1(1) 93/13/EEC) contratti a distanza (97/7/EC) contratti negoziati fuori dei locali commerciali (85/577/EEC) contratti di vendita (1999/44/EC) contratti conclusi per via elettronica (2000/31/EC) contratto di adesione (93/13/EEC) contratto di credito (87/102/EEC) contratti relativi all'acquisizione di un diritto di godimento a tempo parziale di beni immobili (94/47/EC)
	eu8	italiano	europeo	Contratto elettronico	contratti conclusi per via elettronica

nome	descrizione	transitiva
IS_A	L'umeaning è equivalente al padre	transitiva
PART_OF	L'umeaning è parte della definizione del padre	transitiva
INVERSE_OF	L'umeaning è il contrario del termine padre	non transitiva
contraente	il soggetto che stipula un contratto	non transitiva

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

8.4 Amministrazione del sistema

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

8.4.1 Gestione degli account utente

Cliccando sul link *utenti* nella bara di navigazione, si accede ad una pagina mostrata in figura 8.10 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.

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

crea un nuovo utente

Collegato come admin

logout

Ricerca termine

gestione umeaning

Relazioni

Lingue giuridiche

Utenti

Log

Manutenzione database

Parametri di ricerca

login

nome

cognome

e-mail

Trova

risultati ricerca (1 entry trovate; pagina 1 di 1)

azioni	login	nome	cognome	e-mail
Modifica Cancella	admin	Mario	Rossi	admin@sitodisyllabus.org

Copyright ©2005

Figura 8.10 Ricerca di un utente.

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 8.11; 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.

I campi della form sono:

Figura 8.11 Creazione di un nuovo utente.

- 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 quattro *checkbox* che definiscono il livello dell'utente e definiscono le operazioni che egli potrà compiere sul sistema e sono: amministratore, navigazione sito, gestione umeaning e gestione dei tipi di relazione.

8.4.2 Gestione delle lingue giuridiche

The screenshot shows the Syllabus interface. On the left is a navigation menu with items like 'Collegato come admin', 'logout', 'Ricerca termine', 'gestione umeaning', 'Relazioni', 'Lingue giuridiche', 'Utenti', 'Log', and 'Manutenzione data base'. The main content area features a form titled 'Inserisci una nuova lingua giuridica' with two input fields: '*iso639 Countrycode' and '*nome lingua', and an 'Inserisci' button. Below the form is a table titled 'lingue giuridiche inserite' with columns for 'azioni', 'countrycode', and 'nome'. The table lists five languages: italiano (it_IT), english (en_GB), française (fr_FR), deutsche (de_DE), and español (es_ES). Each row has 'Modifica' and 'Cancella' links. The footer contains 'Copyright ©2005'.

lingue giuridiche inserite		
azioni	countrycode	nome
Modifica Cancella	it_IT	italiano
Modifica Cancella	en_GB	english
Modifica Cancella	fr_FR	française
Modifica Cancella	de_DE	deutsche
Modifica Cancella	es_ES	español

Figura 8.12 Gestione delle lingue giuridiche.

Cliccando sul link *lingue giuridiche* nella barra di navigazione si accede ad una pagina mostrata in figura 8.12 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 [35] e [26] 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.

8.4.3 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 8.12. La ricerca può essere ristretta rispetto ad uno qualsiasi dei campi.

The screenshot shows the Syllabus application interface. At the top, there is a navigation bar with the Syllabus logo and a language selector (Lingua interfaccia) with flags for Italian and English. Below the navigation bar, there is a sidebar on the left with menu items: Collegato come admin, logout, Ricerca termine, gestione umeaning, Relazioni, Lingue giuridiche, Utenti, Log, and Manutenzione database. The main content area features a search form titled "Parametri di ricerca" with fields for "elementi piu recenti di" (07, gennaio, 2005), "elementi piu vecchi di" (07, novembre, 2005), "utente", "operazione", and "messaggio" (contratto). A "Trova" button is located at the bottom right of the form. Below the form, there is a table of search results with the following data:

risultati ricerca (1 entry trovate; pagina 1 di 1)				
azioni	utente	data	operazione	messaggio
Cancella	admin	28 settembre 2005 12:56:26	relation_type insert	inserted relation_type_name="contraente" description="il soggetto che stipula un contratto"

At the bottom of the page, there is a footer with the text "Copyright ©2005".

Figura 8.13 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*.

8.4.4 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] T.R. Gruber, A Translation Approach to Portable Ontology Specification, p.199-220, 1993
- [2] J.Breuker, A.Valente, R.Winkels, Procs. of 1st LegOut Workshop on Legal Ontologies, 1997
- [3] T.R.Gruber, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, p.907-928, 1993
- [4] A.Gangemi, N.Guarino, C.Masolo, A.Oltramari, L.Schneider, Sweetening Ontologies with DOLCE. Procs of EKAW, 2002
- [5] Eurovoc, <http://europa.eu.int/celex/eurovoc/>
- [6] Celex, <http://europa.eu.int/celex/>
- [7] International Standardization Organizaton, ISO 5964: Guidelines for the establishment and development of multilingual thesauri, 1985
- [8] International Stadardization Organization, ISO 2788: Guidelines for the establishment and development of monolingual thesauri, 1986
- [9] P.R.Visser, T.J.Bench-Capon, A comparison of four ontologies for the Design of Legal Knowledge Systems, p.27-57, 1998
- [10] L.T.McCarty, A Language for Legal Discourse: Basic Features. Procs of Second International Conference on Artificial Intelligence and Law, 1989
- [11] R.K.Stamper, The role of semantics in legal expert systems and legal reasoning, p.219-244, 1991
- [12] V.Kraligen, Frame-based conceptual models of statute law, p., 1995
- [13] Gene Ontology Consortium, <http://www.geneontology.org>
- [14] Semantic web project, <http://www.semanticweb.org>
- [15] F.Manola, E.Miller, RDF Primer, 2004, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>
- [16] D.L.McGuinness, F.vanHarmelen, OWL Web Ontology Language Overview, 2004, <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- [17] T.H.Cormen, Introduzione agli algoritmi, McGraw-Hill, 1994
- [18] Graphviz - Graph Visualization Software, <http://www.research.att.com/sw/tools/graphviz/>
- [19] Chris Mungall, GO Database Schema, 2004
- [20] G.Lolli, Logica Matematica, 2004, <http://www2.dm.unito.it/paginepersonali/lolli/dispense.htm>

- [21] T.B.Lee, J.Hendler, O.Lassila, The Semantic Web, 2001,
<http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>
- [22] 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>
- [23] T.Bray, J.Paoli, C. M. Sperberg-McQueen, E.Maler, F.Yergeau, Extensible Markup Language (XML) 1.0 (Third Edition), 2004, <http://www.w3.org/TR/2004/REC-xml-20040204/>
- [24] Protégé ontology editor and knowledge-base framework, <http://protege.stanford.edu>
- [25] 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>
- [26] H.Alvestrand, Tags for the Identification of Languages, RFC 1766, 1995, <ftp://ftp.rfc-editor.org/in-notes/rfc1766.txt>
- [27] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, 1998
- [28] 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>
- [29] E. Nebel, L. Masinter, Form-based File Upload in HTML, RFC 1867, 1995, <ftp://ftp.rfc-editor.org/in-notes/rfc1967.txt>
- [30] W3C HTML Working Group, XHTML 1.0 The Extensible HyperText Markup Language, 2002, <http://www.w3.org/TR/2002/REC-xhtml1-20020801>
- [31] D.Sadoski, Three Tier Software Architectures, 2000,
http://www.sei.cmu.edu/str/descriptions/threetier_body.html
- [32] S.Wheeler, Mature Design Theory in Web Development, 2005,
<http://www.zend.com/php/design/mature-design.php>
- [33] R.A.Elmasri, S.B.Navathe, Sistemi di basi di dati - fondamenti, Addison-Wesley, 2001
- [34] FabForce DB Designer 4, <http://www.fabforce.net/dbdesigner4/>
- [35] International Standardization Organizatton, ISO 639 Language Codes, 1988,
<http://www.w3.org/WAI/ER/IG/ert/iso639.htm>
- [36] W.Stallings, Cryptography and network security, Pretience Hall, 2003
- [37] P.Jalote, An integrated approach to software engineering, Springer, 1997
- [38] Z.Suraski, PHP's OO Evolution, 2004, <http://www.zend.com/php5/articles/engine2-php-oo.php>
- [39] The PHP Group, PHP manual, 2005, <http://www.php.net/manual/en/>

- [40] M.Ohrt A.Zmievski, Smarty - the compiling PHP template engine, 2005,
<http://smarty.php.net/manual/en/>
- [41] Mitja Kolsek, Session Fixation Vulnerability in Web-based Applications, 2002,
http://www.acros.si/papers/session_fixation.pdf
- [42] The PostgreSQL Global Development Group, PostgreSQL 8.0.4 Documentation,
2005, <http://www.postgresql.org/docs/8.0/static/index.html>