

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Specialistica in Informatica

**Integrazione di standard di
interoperabilità
dei test nelle piattaforme e-learning
dell'Ateneo**

Tesi di Laurea in Sistemi Multimediali

Relatore:

Chiar.ma Prof.ssa
Paola Salomoni

Presentata da:

Matteo Ricci

Correlatore:

Dott.ssa
Catia Prandi

**Sessione III
Anno Accademico 2011-2012**

Indice

Introduzione	v
1 Strumenti e standard e-learning	1
1.1 Evoluzione degli strumenti e-learning	2
1.2 Le piattaforme e-learning	4
1.3 Learning Object	6
1.4 Necessità di uno standard	8
1.5 Advanced Distributed Learning: SCORM	10
1.5.1 Content Aggregation Model	10
1.5.2 Run-time Environment	13
1.6 IMS Content Packaging	14
1.7 IMS Common Cartridge	15
1.7.1 Struttura del package	16
1.7.2 Risorse	16
2 IMS Question Test Interoperability	21
2.1 Storia e obiettivi dello standard	22
2.2 IMS QTI 1.2.1	23
2.2.1 Struttura del Modello ASI	23
2.2.2 Modello QTI Result Reporting	32
2.3 IMS QTI 2.1	34
3 Piattaforme e-learning di Ateneo	37
3.1 Moodle	38

3.1.1	Tipologie di esercizi	39
3.2	Piattaforme Moodle di Ateneo	43
3.2.1	Alfacert	44
3.2.2	DALIA	45
3.3	ATutor	48
3.3.1	Piattaforma A ³	49
3.3.2	AContent	50
4	Progetto	53
4.1	AContent BEAT	54
4.2	Analisi delle tipologie di esercizi esistenti	55
4.2.1	Likert	58
4.2.2	Open Ended	59
4.2.3	Matching graphical e simple	61
4.2.4	Multiple choice e multiple answer	62
4.2.5	Ordering	63
4.2.6	Riepilogo delle osservazioni emerse	64
4.3	Progettazione di nuovi esercizi	66
4.3.1	Crosswords	67
4.3.2	Text entry	68
4.3.3	In-line choice	69
4.3.4	Ordering Words	70
4.3.5	Ordering Sentences	71
4.3.6	Category Matching	72
4.3.7	Drag & Drop Matching	73
5	Implementazione	75
5.1	Struttura del codice AContent	75
5.2	Creazione di un esercizio su AContent	76
5.2.1	Maschera di creazione/modifica	79
5.2.2	Maschera di preview	81
5.2.3	Import/export	81

5.3	Altri esercizi implementati	83
5.3.1	In line choice	83
5.3.2	Crosswords	84
	Conclusioni	88
	Bibliografia	89

Introduzione

L'8 giugno del 1951 all'altro capo del mondo, ad Alice Springs, la prima lezione della *School of the Air* [SoA51] veniva trasmessa via radio a studenti distanti tra loro migliaia di chilometri, disseminati nell'*Outback* australiano. Si trattava della prima scuola di questo tipo, che applicava le tecnologie “moderne” all'insegnamento a distanza. Oggi, trascorsi più di sessant'anni, la stessa *School of the Air* è ancora operativa, sfruttando le infrastrutture di Rete e il Web per gli stessi scopi. Questo è solo uno dei tanti esempi di come l'evoluzione dei sistemi di comunicazione avvenuta negli ultimi decenni ha consentito lo sviluppo di sistemi di apprendimento a distanza, di cui l'*e-learning* è la più recente applicazione. Da un lato, la diffusione dell'*e-learning* è stata accompagnata da studi delle metodologie di insegnamento e di apprendimento, dall'altro è seguito un importante sviluppo tecnologico, che ha visto nascere piattaforme orientate al Web per l'insegnamento (LMS - *Learning Management System* o LCMS - *Learning Content Management System*) e classi virtuali (*Virtual Classroom*).

Tipicamente una piattaforma *e-learning* fornisce servizi a diversi tipi di utenti che ad essa si interfacciano. Al docente che deve rendere disponibili i contenuti didattici e le prove per la verifica dell'apprendimento vengono messi a disposizione degli “editor” per la creazione contenuti. Lo studente può avvalersi di un'interfaccia Web accessibile tramite Browser previa autenticazione, per accedere al materiale didattico e ai test di verifica, nonché agli strumenti collaborativi per ricevere assistenza durante l'apprendimento e confrontarsi con i suoi pari o il docente. L'utilizzo del Web consente un

semplice accesso e una grande fruibilità dei materiali didattici, senza limiti di orari e senza vincoli di presenza fisica. La responsabilità delle piattaforme è tuttavia affidata a personale tecnico specializzato, con competenze di programmazione, conoscenze delle basi di dati e sistemistiche e di gestione delle politiche di accesso.

Il cuore di ogni sistema *e-learning* è la disponibilità dei contenuti didattici. Per regolamentarne la diffusione su questi ambienti di apprendimento orientati al Web è sorta l'esigenza di progettare delle entità per la loro rappresentazione, che ne garantiscano il riutilizzo e la fruizione anche *offline*. Dal punto di vista tecnico questo obiettivo è stato raggiunto con l'introduzione dei *Learning Objects* (LO), definiti come i più piccoli elementi digitali interscambiabili che contengono materiale didattico. La versatilità e la portabilità dei *Learning Objects* non può prescindere dalla definizione di alcuni standard, definiti da enti quali *IMS Global Consortium* e *Advanced Distributed Learning*, che hanno individuato le specifiche per la gestione e lo scambio di contenuti nonché per la creazione di prove di verifica.

I due standard principali, *IMS CC (Common Cartridge)* e *SCORM (Shareable Content Object Reference Model)* si impegnano a curare tutti gli aspetti legati alla fruizione di contenuti e all'importazione ed esportazione di *Learning Objects* tra sistemi differenti. Le specifiche relative alle prove, le linee guida per la loro creazione, per la gestione dei risultati e per la distribuzione sono invece affidate allo standard *IMS QTI Question Test Interoperability*, progettato da *IMS Global Consortium*.

Per i suoi progetti *e-learning*, l'Ateneo di Bologna si avvale della piattaforma LMS Moodle a supporto agli insegnamenti e allo studio delle lingue (DALIA e Alfacert). La gestione della piattaforma, la formazione e l'assistenza ai docenti è attualmente competenza del Settore *e-learning* di Ateneo. Al fine di rendere i docenti o i tutor più indipendenti sulla creazione dei contenuti da pubblicare sulla piattaforma è nato il progetto AContent BEAT, frutto della collaborazione di pedagogisti e informatici dei Corsi di Studio dell'Ateneo, con la supervisione del Settore *e-learning*. Questo progetto ha fornito

metodologie e suggerimenti per la produzione di materiale didattico compatibile con gli standard più conosciuti attraverso la creazione di strutture e procedure guidate.

Questo progetto di tesi si colloca in una necessaria estensione del progetto AContent BEAT, con lo scopo di ampliare l'offerta di esercizi presenti sull'editor di contenuti, per renderlo uno strumento valido e completo per la creazione di prove, con particolare riguardo all'esportabilità sulle piattaforme più diffuse. Il punto di partenza è stato uno studio approfondito delle specifiche dello standard IMS QTI 1.2.1 e delle tipologie di esercizi che possono essere implementate, confrontando le specifiche con la nuova versione dello standard. Un'analisi comparativa delle tipologie di esercizi di base presenti su Moodle e di quelle più utilizzate all'interno delle piattaforme degli insegnamenti ha fatto emergere delle carenze degli esercizi di base in AContent. In seguito ad una serie di incontri con il Settore *e-learning* di Ateneo, sono quindi stati individuati gli esercizi da implementare e le interfacce da migliorare, al fine di rendere più comprensibile l'utilizzo della piattaforma a utenti che non hanno particolari competenze informatiche.

Nel primo capitolo di questa tesi viene introdotto il concetto di *e-learning*, a partire dall'evoluzione degli strumenti disponibili grazie alle tecnologie del Web 2.0. Le piattaforme *e-learning* e le principali caratteristiche vengono descritte, includendo il percorso che ha portato alla nascita dei *Learning Objects*. Vengono infine illustrati gli standard SCORM, IMS CP e IMS CC.

Nel secondo capitolo sono illustrate in dettaglio le specifiche dello standard IMS QTI 1.2.1. Le strutture dei modelli *assessment section item* e *result reporting* vengono discusse a confronto con le nuove specifiche IMS QTI 2.1.

Nel terzo capitolo viene offerta una panoramica delle piattaforme *e-learning* e i loro ambiti di utilizzo nell'Ateneo di Bologna, soffermandosi sulle modalità di esercizio implementate. Dopo aver presentato la piattaforma Moodle, vengono descritte DALIA e Alfacert, le piattaforme dedicate all'ambito linguistico, e le piattaforme gestite dal Settore *e-learning*. Il capitolo si conclude con la descrizione delle caratteristiche principali di ATutor, una piattaforma

alternativa a Moodle che offre maggiori criteri di accessibilità e che è stata sviluppata da un gruppo di ricerca dell'università di Toronto in collaborazione l' *Adaptive Technology Resource Centre* con il quale l'Ateneo di Bologna, collabora in diversi progetti.

Nel quarto capitolo viene presentato il progetto di tesi e descritta la piattaforma *e-learning* di partenza, proseguendo con un'analisi dettagliata delle modalità di esercizio già presenti e le conseguenti motivazioni che hanno portato all'implementazione di nuovi esercizi.

Il quinto capitolo tratta dettagliatamente gli aspetti implementativi legati alla creazione di un esercizio, comprendenti importazione ed esportazione. Le interfacce progettate durante questo lavoro vengono infine illustrate.

Capitolo 1

Strumenti e standard e-learning

Esistono attualmente svariate definizioni di *e-learning*. Una delle più comuni è quella dell'Associazione Italiana per la Formazione Manageriale (ASFOR), che definisce con questo termine una metodologia didattica che offre la possibilità di erogare elettronicamente contenuti formativi attraverso Internet o reti intranet. L'*e-learning* rappresenta per l'utente una soluzione di apprendimento flessibile, in quanto fortemente personalizzabile e facilmente accessibile [Asf03].

L'*e-learning* è il risultato dell'applicazione delle tecnologie moderne alle metodologie di Formazione a Distanza (FAD), diffuse fin dalla metà del XIX secolo attraverso la posta. Conseguentemente all'evolversi dei canali di comunicazione anche le modalità di FAD si sono aggiornate, passando dall'utilizzo della corrispondenza (FAD di prima generazione), all'introduzione di sussidi multimediali quali videocassette, CD e DVD (seconda generazione) e infine all'utilizzo del personal computer e della rete Internet e intranet (terza generazione).

L'evoluzione degli strumenti di *e-learning* è brevemente presentata nella prima parte di questo capitolo. In particolare, viene discusso il contributo del Web 2.0 alle nuove metodologie di formazione *online*. Le diverse piattaforme di *e-learning* sono introdotte nella seconda sezione, dove si mette in luce la principale problematica ad esse legate, ovvero la mancanza di adeguati

meccanismi di interoperabilità. La soluzione a questo problema è rappresentata dai *Learning Objects* (LO) e dagli standard che sono nati in seguito per garantire il riutilizzo dei contenuti che vengono trattati nell'ultima parte del capitolo.

1.1 Evoluzione degli strumenti e-learning

In letteratura il concetto di *e-learning* è stato discusso sia dal punto di vista pedagogico che dal punto di vista tecnico ovvero della ricerca di nuovi strumenti tecnologici che servono allo scopo. Sebbene ci sia un generale consenso sul fatto che l'*e-learning* sia un processo educativo e formativo che presuppone l'utilizzo di dispositivi elettronici, non si è raggiunto un accordo sulla terminologia [Kar07]. Alcuni autori considerano già il *Computer Based Training* (CBT), e il *Web Based Training* (WBT) come prime forme di *e-learning*. Con la diffusione dei personal computer a partire dagli anni '90, infatti, il pc veniva usato come strumento di autoistruzione per mezzo della distribuzione di corsi multimediali o enciclopedie su CD-ROM (CBT). Tuttavia questi mezzi non permettevano interazione tra l'utilizzatore e chi offriva il corso e l'approccio era di tipo monodirezionale, non consentendo all'utente di potersi confrontare durante l'apprendimento o di verificare le conoscenze acquisite. Non appena l'utilizzo della rete Internet è divenuto di più larga fruibilità, le stesse informazioni sono diventate reperibili attraverso questo canale e si è passati dal CBT al WBT, introducendo anche strumenti collaborativi tra l'autore del contenuto e l'utente, come *email*, *forum*, *wiki*, *chat*, *newsgroup*, ecc...

Ciò che si intende per *e-learning* in questo lavoro è un'evoluzione ulteriore del WBT che include nuovi strumenti per una gestione più flessibile dei contenuti e la tracciabilità del percorso formativo dell'utilizzatore.

La novità che ha favorito lo sviluppo dell'*e-learning* in queste direzioni è l'utilizzo di *Learning Content Management System*(LCMS) e *Learning Management System*(LMS). Gli LMS sono piattaforme che consentono di svi-

luppare, gestire e tracciare l'interazione tra studenti e contenuti e tra studenti e docenti. I LCMS sono un'evoluzione dei *Content Management System* (CMS) nel contesto dell'*e-learning*. I CMS sono sistemi per la gestione di contenuti e forniscono servizi di gestione degli aggiornamenti e delle versioni, il processo di creazione e la pubblicazione dei contenuti. Sebbene i sistemi CMS siano utilizzati all'interno di progetti di *e-learning*, essi non vengono incontro a tutte le esigenze del processo formativo. Fondendo le funzionalità specifiche dei LMS con i CMS nascono gli LCMS. Gli aspetti tecnici che caratterizzano entrambi i sistemi sono approfonditi nel paragrafo successivo.

La nascita delle piattaforme di *e-learning* è una conseguenza dell'evoluzione del Web a Web 2.0, dove ci si ritrova un contesto più interattivo e dinamico nel quale non è la tecnologia ma l'utilizzatore ad aggiungere valore ai servizi offerti [Bon06]. Grazie alla separazione del contenuto dalla presentazione non è più necessario possedere particolari competenze informatiche per la creazione e gestione dei contenuti. L'introduzione innovativa di strumenti collaborativi e il *social networking* consentono di creare reti di relazioni e di contatti in cui la collaborazione assume il ruolo chiave per la crescita e lo sviluppo. Questi strumenti che in realtà sono stati creati con finalità diverse dall'*e-learning* hanno migliorato la diffusione e la condivisione dei materiali sul Web. Il Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA)(attualmente per decreto legislativo 177 del 1 dicembre 2009, diventato DigitPA [Dig09]) dichiara che gli strumenti di supporto all'apprendimento cooperativo su Internet come *wiki blog* e *chat*, strumenti per la ricerca delle informazioni e la gestione delle fonti su Internet come *feed* RSS [RSS13] e *social bookmarking*, e gli strumenti per il supporto all'apprendimento per utenti mobili sono tutte tecnologie del Web 2.0 che hanno una funzione specifica in un contesto *e-learning*.

In tempi recenti si è cominciato a parlare della prossima evoluzione del Web 2.0 ovvero il Web 3.0 caratterizzato da un approccio semantico dove le applicazioni sono in grado di rispondere a richieste più complesse, con la creazione di metodi di interazione sempre più simili a quello umano. Tuttavia

ancora non ci sono segnali che fanno capire se questi nuovi strumenti possono essere applicati all'*e-learning* [Lam09].

1.2 Le piattaforme e-learning

È possibile definire le piattaforme *e-learning* come ambienti per la formazione sul Web che mettono a disposizione strumenti per l'erogazione di contenuti di apprendimento opportunamente strutturati.

Attraverso una moltitudine di servizi questi ambienti offrono forme di interazione sia tra gli studenti e il corso che tra gli studenti e i tutor/docenti. I docenti sono in grado di creare e diffondere contenuti, gestire le iscrizioni ai corsi, supervisionare la partecipazione degli studenti e verificare le attività svolte. Gli studenti a loro volta possono effettuare un download dei contenuti, la visualizzazione e l'utilizzo dei materiali didattici e partecipare a discussioni tramite *forum*, *chat*, videoconferenza ecc..

Per creare questi ambienti di formazione, spesso denominati Virtual Learning Environment (VLE) ci si può avvalere di gestori dei contenuti orientati all'apprendimento come i LCMS o LMS. La fascia di intersezione tra questi due strumenti è ampia pertanto si tende a confondere quali siano le funzionalità dell'uno rispetto all'altro.

Un LMS è un sistema per l'erogazione dei corsi *e-learning* che oltre alla distribuzione dei corsi online, offre un supporto a tutte le funzioni e attività necessarie alla gestione di un corso. I servizi principali che contraddistinguono un LMS sono:

- *scheduling*: consente di effettuare una programmazione dei corsi e definire i piani di studio anche a seconda delle esigenze individuali dello studente;
- *distribuzione*: la distribuzione di corsi online che si avvalgono anche di strumenti per la valutazione;

- *tracking*, permette di tenere traccia dei progressi degli studenti e crea *report* utili al monitoraggio da parte dei docenti o a fini statistici;
- *comunicazione*: avviene tramite la messa a disposizione di *chat* e *forum*, l'organizzazione di seminari e la possibilità di scambio di email al fine di promuovere la discussione per rendere disponibili *chat*, *forum*, seminari, sistemi di scambio di email al fine di far dialogare su aspetti inerenti al corso e non gli studenti;
- *test*: permettono ai docenti di rendersi conto del livello finale di apprendimento dei propri studenti.

I LCMS consentono di soddisfare molteplici richieste:

- *registration*: registrazione utenti, pagamenti *online*, iscrizioni ad attività, importazione utenti;
- *scheduling*: organizzazione corsi, pianificazione di percorsi didattici , impostare curriculum utente, pianificare risorse;
- *delivery*: informazioni relative ai corsi e agli eventi, accesso ai corsi e alle risorse multimediali;
- *tracking*: tracciamento delle attività svolte dagli utenti tramite report e tracciamento sulla partecipazione agli eventi;
- *communications*: *email* , *chat*, *forum* e seminari *online*;
- *testing*: somministrazione di prove, creazione di test;
- *content*: creazione e modifica dei contenuti, importazione ed esportazione di contenuti, *versioning* dei contenuti e pubblicazione dei contenuti;
- *repository* archivio digitale per la gestione e la ricerca di contenuti didattici.

Per sfruttare un'analogia fornita da Jacobsen, i sistemi LMS tratterebbero in questo caso tutto quanto accade "fuori dall'aula", come sistemi amministrativi, biblioteche e strutture generali dell'ateneo, mentre i LCMS gestirebbero anche l'attività "dentro l'aula", ovvero la preparazione e lo svolgimento dei corsi incluso un registro dettagliato delle attività didattiche.[Jac03].

È possibile trovare piattaforme commerciali, offerte da fornitori legati ai principali marchi di software, ottenibili solo attraverso un dispendio economico; oppure piattaforme *Open Source* reperibili gratuitamente in Rete. Gli LMS e LCMS sono basati su diverse piattaforme di sviluppo, quali Java EE, Microsoft.NET, PHP. Generalmente per il salvataggio dei dati si ricorre a un DBMS come MySQL, Postgree, ORACLE ecc..

1.3 Learning Object

La proliferazione di piattaforme *e-learning* ha contribuito alla necessità di poter "trasportare" contenuti da un sistema a un altro. Questa esigenza ha cambiato il concetto di contenuto didattico con la conseguenza che unico corso prima monolitico è divenuto costituito da "blocchi" autonomi denominati *Learning Objects* LO.

La prima similitudine adottata per far comprendere la dinamica dei LO è stata quella di paragonarli ai mattoncini LEGO [HodWay02] per far risaltare la possibilità di scomporli, combinarli e ricomporli oltre a non richiedere particolari competenze tecniche. In un secondo momento si è capito che questo paragone non era adeguato alla reale struttura degli LO pertanto si è deciso di prendere come esempio gli atomi che hanno caratteristiche che si avvicinano di più a quelle dei LO. A differenza dei mattoncini LEGO gli atomi non possono combinarsi tutti tra loro ma possono combinarsi in strutture che dipendono dalla loro organizzazione interna. Inoltre per combinare gli atomi sono richieste particolari competenze tecniche [Wil02].

L'*Institute of Electrical and Electronics Engineers* (IEEE) definisce i LO come “una qualsiasi entità digitale e non che può essere utilizzata e riutilizzata per l'apprendimento l'insegnamento o l'esercitazione” [IEEE06]. Si può facilmente notare come questa definizione comprenda ogni tipo di contenuto. Le definizioni di *e-learning* e di *Learning Objects* adottate in questo capitolo, aderiscono alla visione di Stephen Downes [Dow02] che dichiara che un LO per definirsi tale deve possedere le seguenti caratteristiche:

- *riusabilità*: la possibilità riutilizzare un contenuto didattico pensato per un determinato corso, in un certo contesto, in un diverso corso e contesto è una delle ragioni primarie che stanno alla base della creazione del concetto di LO. Avere accesso a un contenuto e condividerlo con più utilizzatori. La riusabilità è l'elemento cardine LO, in quanto consente di non reinventare contenuti;
- *digitalizzazione*: sebbene una delle definizioni ufficiali più accreditate (IEEE, data in precedenza) includa nel concetto di LO anche risorse non digitali, come libri o qualunque altra risorsa didattica, una definizione operativa legata all'*e-learning* e basata su piattaforme tecnologiche limita il campo delle risorse a quelle digitali;
- *modularità*: gli oggetti di apprendimento non sono interi corsi monolitici, con un inizio ed una fine e senza possibilità di scomposizione, ma piuttosto unità di contenuti più piccole, utilizzabili in diversi contesti. Il singolo LO deve costituire un'entità autonoma dotato di un obiettivo didattico;
- *interoperabilità*: la possibilità di poter utilizzare lo stesso contenuto su piattaforme differenti attraverso un meccanismo di importazione dovuto a una compatibilità di formati fra sistemi differenti;
- *accessibilità*: un LO deve essere di facile reperibilità, in modo da poter utilizzare un determinato contenuto, venire a conoscenza della sua esistenza e poterlo ricercare.

Per una classificazione e ricerca di un contenuto pertanto ai LO vengono aggiunti un insieme di metadati. I metadati non si riferiscono direttamente ai contenuti concreti di un LO bensì li classificano, attraverso una serie di informazioni che descrivono elementi di interesse contenuti all'interno dei LO. La definizione data da *Learning Technology Standards Committee*(LTSC) li classifica come informazioni riguardanti un oggetto.[LTSC05]

Un' analogia utile per capire la funzione di questi descrittori è il prendere come esempio la catalogazione di un libro in una biblioteca sulla base di un insieme di elementi descrittivi come il titolo l'autore , anno di pubblicazione, genere etc.. Allo stesso modo un LO una volta introdotti i metadati diventa facilmente reperibile anche da persone diverse dall'autore che l'ha creato. Un LO metadatato può essere inserito all'interno di appositi *repository* rendendosi accessibile e reperibile a qualsiasi utente lo ricerchi in qualsiasi luogo e momento.

1.4 Necessità di uno standard

La varietà di piattaforme di *e-learning* presenti oggi giorno sul mercato ha evidenziato l'esigenza di standardizzare i contenuti formativi per poter utilizzare quest'ultimi all'interno di più piattaforme. Il riutilizzo di contenuti formativi attraverso la creazione di LO non solo garantisce interoperabilità ma porta vantaggi anche dal punto di vista della manutenzione e sugli investimenti che possono essere compiuti a livello di contenuti e servizi e non essere legati alla scelta architettonica [CNIP04].

Le prime soluzioni di standardizzazione furono implementate per le FAD agli inizi anni '80, nel contesto dell'industria militare americana dove venivano richieste conformità ad una visione modulare e procedurale della formazione per il soggetto che apprende in autonomia e una semplicità di identificazione orientata al riutilizzo. Con il passare degli anni e le trasformazioni delle possibilità di apprendimento a distanza sempre più enti si sono occupati di cercare soluzioni comuni per creare interoperabilità. L'elemento principa-

le da standardizzare in questa generazione di *e-learning* risulta chiaramente il LO e infatti il 25 luglio 2002 l'IEEE-SA (Standards Association) ha approvato, con i contributi di altre organizzazioni, tra cui ARIADNE [ARI05], *Dublin Core Metadata Initiative* [Dub05]. e IMS *Global Learning Consortium* [IMS05], il protocollo IEEE 1484.12.1 *Standards for Learning Object Metadata* [IEEE02] che fornisce una linea guida su ciclo di vita, metadati, caratteristiche educative, tecniche, legislative (relative ai diritti d'autore), relazionali (eventuali collegamenti o relazioni con altri contenuti), annotazioni e classificazioni in categorie. In particolare la parte relativa ai metadati suddivide questi descrittori in 9 categorie che contengono gruppi strutturati di sottocategorie di standard. Segue un elenco schematico delle categorie:

- *general*: caratteristiche generali, indipendenti dal contesto del contenuto dell'oggetto didattico descritto;
- *lifecycle*: caratteristiche relative al ciclo di vita della risorsa;
- *meta-metadata*: caratteristiche della descrizione stessa e non dell'oggetto descritto;
- *technical*: caratteristiche tecniche della risorsa;
- *educational*: caratteristiche educative e pedagogiche della risorsa;
- *rights*: informazioni che hanno a che fare con le condizioni di utilizzo della risorsa;
- *relation*: caratteristiche della risorsa che la collegano ad altre;
- *annotation*: caratteristiche che permettono la descrizione di note sull'utilizzo educativo della risorsa;
- *classification*: caratteristiche utili per classificare i contenuti della risorsa.

Tutti gli standard promossi dalle maggiori organizzazioni o associazioni riguardanti la creazione di piattaforme *e-learning* utilizzano o integrano il

LOM per descrivere i contenuti dei LO. Per questo lavoro di tesi sono presi in considerazione i lavori di standardizzazione svolti da enti quali *IMS Global Consortium* e *Advanced Distributed Learning* [ADL05] in quanto sono gli standard attualmente utilizzati sulle piattaforme di Ateneo di Bologna.

1.5 Advanced Distributed Learning: SCORM

Lo standard oggi giorno più diffuso in circolazione è sicuramente lo *Shareable Content Object Reference Model* (SCORM) nato nel 1997 da un'iniziativa promossa dal governo Statunitense, ed in particolare dal Department of Defense (DoD)[Dod97] per convogliare in un unico standard quelli allora esistenti.

SCORM, utilizza difatti integralmente sottoparti delle specifiche IMS relative ai contenuti *e-learning*, per l'interoperabilità delle risorse didattiche tra differenti piattaforme. Attualmente le specifiche di SCORM sono alla versione 1.3 denominato SCORM 2004 anche se il più diffuso rimane la versione 1.2. Questo standard si basa sul linguaggio XML e comprende specifiche che forniscono istruzioni su come impacchettare, catalogare e consegnare all'utente finale i materiali formativi. Pertanto l'idea di SCORM rientra nella filosofia di LO introdotta precedentemente secondo la quale ogni singolo oggetto di apprendimento deve essere accessibile, riusabile, interoperabile, e affidabile. I fornitori di contenuto sono a conoscenza del fatto che i LO su cui hanno richiesto ed ottenuto la certificazione SCORM potranno essere utilizzati da tutte le piattaforme certificate SCORM garantendone l'interoperabilità [DalMar06]. SCORM definisce un *Content Aggregation Model*(CAM) utilizzato per l'aggregazione dei contenuti e un ambiente di elaborazione per i LO denominato *Run-Time Environment*.

1.5.1 Content Aggregation Model

Il *Content Aggregation Model* (CAM) è uno strumento per l'aggregazione delle risorse di apprendimento in un unico strutturato contenuto didattico.

Dalla parola stessa *aggregation* il CAM è composto da tre elementi quali:

- *Content Model*: che contiene definizione e nomenclatura dei componenti dell'oggetto didattico;
- *metadata*: un meccanismo di descrizione dei componenti di specifici *Content Model*;
- *Content Packaging*: definisce la rappresentazione della struttura dei contenuti; didattici e la modalità di impacchettamento delle risorse per l'interscambio tra ambienti differenti.

Content Model

Il *Content Model* (CM), descrive i “componenti SCORM” utilizzati per costruire un'esperienza di apprendimento a partire dal riutilizzo delle risorse disponibili. Il CM definisce come queste risorse “semplici”, condivisibili e riutilizzabili, sono aggregate per comporre unità formative di livello superiore. Il CM si compone di tre parti fondamentali: *Asset*, *Sharable Content Objects* e *Content Aggregations*. L'apprendimento dei contenuti, nella sua forma più elementare, è reso possibile grazie alla presenza di *Asset*, ovvero tutte quelle risorse di base che sono rappresentazioni digitali di testi, immagini, suoni, pagine Web e che, come tali, possono essere inviate ad un *Client Web*. Un *Asset* può essere descritto attraverso metadata che ne garantiscono la ricerca all'interno di *repository online*, rafforzandone, in tal modo, le possibilità di riutilizzo.

Un *Sharable Content Object* (SCO) si compone di una raccolta di *Asset* tra le quali vi è uno specifico *launchable Asset* che utilizza *SCORM Run Time Environment* per comunicare con la piattaforma. Uno SCO rappresenta il più basso livello di granularità delle risorse per l'apprendimento che possono essere monitorati da un LMS o LCMS utilizzando lo *SCORM Run Time Environment*. Per essere riutilizzabile, uno SCO deve essere indipendente dal contesto di apprendimento nel quale è situato. Ad esempio, uno SCO può essere riutilizzato in diverse esperienze di apprendimento per raggiungere

differenti obiettivi formativi. Inoltre, uno o più SCO possono essere aggregati per formare un'unità di insegnamento o di formazione di livello superiore che soddisfi obiettivi di apprendimento di alto livello.

Gli SCO sono destinati a essere piccole unità, in modo da rendere possibile il loro potenziale riutilizzo per raggiungere più obiettivi formativi anche se lo standard non impone alcun vincolo implementativo sull'esatta dimensione di uno SCO.

Lo SCO deve aderire allo SCORM *Run-Time Environment* attraverso un metodo che gli permetta di individuare un *API adapter* del LMS e, contenere le principali funzioni di chiamata *Application Program Interface* API ovvero (*LMSInitialize()* e *LMSFinish()*). Le altre chiamate di funzioni non sono necessarie poiché opzionali e dipendenti dalla natura del contenuto. L'esigenza d'iterazione tra uno SCO e lo SCORM *Run Time Environment* porta i seguenti vantaggi:

- qualsiasi LMS che supporta lo SCORM *Run Time Environment* può lanciare gli SCO e tenerne traccia, indipendentemente da chi li ha generati;
- può monitorare qualsiasi SCO e sapere quando è stato avviato e terminato;
- può avviare qualsiasi SCO allo stesso modo.

Il *Content Aggregation* è una mappa (struttura dei contenuti), che può essere utilizzata per aggregare risorse didattiche in una coerente unità d'insegnamento (corso, capitolo, modulo ecc...). Anche il *Content Aggregation*, esattamente come per gli *Asset* e per gli SCO, può avvalersi dei metadati, che ne garantiscono una corretta ricerca all'interno di *repository* online. Il *Content Aggregation* descrive quindi la struttura dei contenuti e i meccanismi per la definizione dell'ordine con il quale le risorse vengono presentate all'utente finale.

Content Packaging

Il *Content Packaging* (CP) fornisce una metodologia per lo scambio di risorse tra differenti sistemi ovvero ha lo scopo di garantire l'interoperabilità. Le specifiche del CP includono la definizione della struttura e il comportamento di una collezione di risorse didattiche. In particolare vengono definiti:

- il file *manifest*: file *XML-based* che contiene i metadati di riferimento del *package* e contiene una struttura su come organizzare i contenuti con un elenco di riferimenti alle risorse;
- le linee guida e le direttive su come comprimere il file manifest e le relative risorse.

I CP vengono utilizzati per la condivisione, il riutilizzo e quindi lo spostamento di risorse, o aggregati di risorse, tra LMS, LCMS, *authoring tools* e *content repositories*. Le specifiche del CP forniscono un formato standard per l' "import/export" che qualsiasi sistema può implementare.

1.5.2 Run-time Environment

L'obiettivo principale di SCORM è quello di rendere le risorse didattiche riutilizzabili e garantire l'interoperabilità attraverso diversi LMS o LCMS. Per rendere questo possibile, deve essere presente: un metodo comune per "avviare" una risorsa didattica di apprendimento, un meccanismo comune che permetta alla risorsa di dialogare con il LMS o LCMS e un linguaggio predefinito che costituisce la base della comunicazione. Il *Run-time Environment* si occupa di questo processo di comunicazione attraverso API, *Launch* e il *Data Model*.

Launch

Il meccanismo di *launch* definisce un metodo generale che permette alla piattaforma di avviare una risorsa di apprendimento sul Web. Le procedure di comunicazione sono definite per mezzo di API comuni.

API

L'interfaccia rappresenta quel meccanismo comunicativo che informa la piattaforma dello stato della risorsa (può essere inizializzata, terminata o in stato di errore). La configurazione dell'API permette la comunicazione tra SCO e piattaforma.

Data Model

Il *Data Model* è il modello standard per definire la modalità di comunicazione tra piattaforma e SCO. Entrambi sono tenuti a “conoscere” lo stato degli elementi sia per garantirne la fruibilità (piattaforma *e-learning*) sia per permetterne il riutilizzo attraverso diversi sistemi (SCO).

1.6 IMS Content Packaging

L'IMS *Content Packaging* [IMSCP09] analogamente al *Content Packaging* del CAM nello SCORM fornisce le specifiche per descrivere ed importare/esportare il materiale di apprendimento, quali ad esempio un singolo corso oppure una collezione di corsi. Un pacchetto IMS CP è formato da un *manifest* e dalle risorse contenute nel *package*. Per il loro utilizzo, i file sono organizzati in una singola *directory* che include il *manifest*, mentre, per essere esportati, vengono compressi in un singolo file denominato *Package Interchange File* (PIF). Il file XML *manifest* deve obbligatoriamente chiamarsi *imsmanifest.xml*. All'interno del file *manifest* vi sono:

- *metadata section*: per includere i metadati;

- *organizations section*: dove viene descritta la struttura dei contenuti;
- *resources*: sezione che contiene i riferimenti alle risorse contenute all'interno del *package*.

1.7 IMS Common Cartridge

L'IMS *GLC Common Cartridge* (IMS CC) [IMSCC08] è uno standard per la creazione e condivisione di contenuti digitali. Attualmente le specifiche di IMS CC sono alla versione 1.1, pubblicate in data 10 maggio 2011. L'IMS CC è il primo dei tre maggiori standard che compongono le specifiche IMS *Digital Learning Services* che hanno lo scopo di supportare le tecnologie per l'*e-learning*. Gli altri due standard sono il *Learning Tools Interoperability* (LTI)[IMSLT10] e il *Learning Information Services* (LIS) [IMSLI08].

Lo standard LTI si occupa dei sistemi di comunicazione tra piattaforme diverse mentre il LIS si occupa di standardizzare le informazioni sugli utenti della piattaforma come privilegi e risultati dei test, ecc.. Le specifiche IMS CC consentono grande flessibilità sulla tipologia dei contenuti e garantiscono interoperabilità. IMS CC contiene all'interno delle sue specifiche un formato per lo scambio di contenuti tra sistemi garantendo consistenza e organizzazione del materiale esportato.

In questo standard è presente inoltre un file *manifest* che descrive la struttura del contenuto e dove sono state archiviate le risorse (interne o esterne tramite URL) che possono essere scambiate anche singolarmente. IMS CC utilizza uno standard di metadati basato su LOM e inoltre ammette altri schemi di dati mostrandosi uno schema estensibile. IMS CC ammette l'integrazione con un altro standard IMS QTI [IMSQT02] nella versione 1.2.1 relativo ai test. Quest'ultimo permette di creare prove e importarle o esportarle su altre piattaforme, come viene discusso nel secondo capitolo di questo lavoro in quanto parte integrante del progetto.

1.7.1 Struttura del package

Una risorsa IMS CC è un *package* di contenuti e metadati integrato in un contesto *e-learning* ovvero tipicamente un LMS o LCMS. Ad alto livello può corrispondere direttamente a un corso in quanto non è specificata una relazione tra contenuto e dimensione del pacchetto. All'interno del *package* dati sono suddivisi in:

- *dati relativi all'esperienza del partecipante*: comprendono le risorse che vengono utilizzate direttamente dall'utente;
- *risorse supplementari*: sono risorse che possono essere aggiunte come ad esempio i test;
- *dati operazionali*: sono inseriti per controllare gli aspetti comportamentali del LMS relativi alla visualizzazione/interazione con la risorsa IMS CC come ad esempio le autorizzazioni;
- *metadati*: sono i descrittori definiti in IEEE LOM [IEEE02].

1.7.2 Risorse

All'interno di un package IMS CC sono contenute le risorse di cui i vincoli sono elencati nella tabella 1.1. Per vincoli si intende il numero di occorrenze dell'elemento che si possono avere all'interno di un package. Le risorse che possono essere incluse in un *package* IMS CC sono:

- *folder*: è un elemento strutturale utilizzato per organizzare il contenuto. Può contenere *subfolder*;
- *Web content*: risorse del Web intese come immagini, documenti e file multimediali ecc.. Possono includere anche riferimenti a file esterni al *package*;
- *discussion Topic*: è un *Learning Application Object* utilizzato per avviare un attività di discussione all'interno del VLE. Non rappresenta

il link a una discussione esistente ma dichiara che la risorsa conteneva una discussione e che il LMS o LCMS deve ricrearla;

- *Web link*: è un *Learning Application Object* che rappresenta un link standard HTTP e può contenere anche attributi legati alle opzioni di apertura;
- *assesment*: è un istanza di un *Assesment QTI*;
- *associated content*: sono file addizionali utilizzati da un singolo *Learning Application Object*;
- *intra-package reference*: sono utilizzati per referenziare i file all'interno del *package*;
- *IMS CC package meta-data*: descrittori relativi a concessione di licenze, organizzazione, ecc...
- *question bank*: rappresenta un'istanza di *QTI Object Bank*.

Tipo di risorsa	Vincoli
Web Content	0 o più
Associated Content	0 o più
QTI Assesment	0 o più
QTI Question Bank	0 o 1
Authorization Data	0 o 1
Discussion Topic	0 o più
Web Link	0 o più
Basic LTI	0 o più

Tabella 1.1: Tipi di risorse supportate con relativi vincoli [IMSCC08]

Learning Application Object

È un oggetto formato da vari tipi di risorse che richiedono ulteriore elaborazione ed interpretazione prima di essere importati e rappresentati nella piattaforma. Un *Learning Application Object* è costituito da una *directory* contenente un file descrittore e ulteriori file o sotto *directory* opzionali utilizzabili solo ed esclusivamente dal *Learning Application Object* stesso. Ogni *Learning Application Object* deve avere un elemento **resource** nel *manifest*. Esempi di *Learning Application Object* includono *QTI Assesment Discussion Topic* e *Web Link*.

L'elemento **resource** che rappresenta il *Learning Application Object* deve contenere un elemento *file* che fa riferimento al descrittore del *Learning Application Object* e non può contenere ulteriori elementi *file*. Se esistono *file* addizionali o *directory* oltre al descrittore del *Learning Application Object* nell'elemento **resource** deve essere presente l'elemento **dependency** per mappare i contenuti aggiuntivi. Inoltre all'interno dell'elemento **resource** non vi possono essere ulteriori elementi **dependency** di tipo *Associated Content* ma può solo contenere ulteriori elementi **dependency** di tipo *Web Content*.

Associated Content

Rappresenta un tipo di risorsa che comprende un insieme di file utilizzati da uno specifico *Learning Application Object*. All'interno del file *manifest* un *Associated Content* è rappresentato da un elemento **resource** con l'attributo **type** di tipo "*associatedcontent*".

L'elemento **resource** che rappresenta *Associated Content* deve contenere un elemento *file* per ogni file esistente nella *directory* del file descrittore del *Learning Application Object* e non può contenere elementi **dependency**.

Web Content

Comprende risorse Web statiche quali: pagine HTML, immagini, GIF JPEG PNG, documenti in formato PDF ecc.. A differenza dell'elemento

resource dell'*Associated Content* gli elementi **resource** di tipo *Web Content* possono fare riferimento anche a un numero indefinito di file. Inoltre le risorse di tipo *Web Content* possono comprendere elementi **dependency** che fanno riferimento a ad altre risorse di tipo *Web Content*.

L'elemento **resource** che rappresenta un *Web Content* può contenere un elemento file per ogni file presente nel *package* non posizionato nella directory del *Learning Application Object* o in una *subfolder*. Infine vi possono essere elementi **dependency** tipo *Web Content* ma non tipo *different*.

Capitolo 2

IMS Question Test Interoperability

Un aspetto fondamentale dell'*e-learning* è la verifica di ciò che si è appreso attraverso l'impiego di esercitazioni. Uno standard per la creazione di domande e test e rispettivi risultati presente anche in IMS *Common Cartridge* (CC) è IMS *Question Test Interoperability* (QTI). Lo standard prevede la possibilità di poter “scambiare” o “trasportare” il materiale prodotto da una piattaforma all'altra grazie all'utilizzo del linguaggio XML, da cui il termine *Interoperability*. Il QTI si presenta flessibile e personalizzabile così da essere adottato rapidamente e su prodotti proprietari già esistenti. Come tutte le specifiche IMS anche quelle relative ai test non limitano le interfacce e i paradigmi pedagogici, nè sono vincolati all'uso di una tecnologia rispetto ad un'altra.

In questo capitolo vengono analizzate le specifiche della versione 1.2.1 in quanto versione attualmente supportata nello standard IMS CC e utilizzata in questo progetto di tesi per l'implementazione delle funzioni di importazione ed esportazione delle tipologie di esercizi aggiunte. Nella parte finale del capitolo vengono inoltre analizzate le principali differenze con la più recente versione dello standard, ovvero la 2.1.

2.1 Storia e obiettivi dello standard

La versione 0.5 delle specifiche del IMS QTI è stata rilasciata a marzo del 1999, successivamente discussa e approvata a novembre dello stesso anno. Nel marzo del 2000 è stato pubblicato il documento relativo alla versione 1.0, poi aggiornata a marzo del 2001 e a gennaio del 2002. Nel febbraio del 2002 circa 6000 copie delle specifiche 1.2 sono state scaricate dal sito del consorzio IMS GLOBAL [IMSQT02]. Nell'addendum del documento relativo alla versione 1.2.1 del marzo del 2003 [QTIA03] vengono descritte le modifiche alle funzionalità segnalate dalla comunità degli sviluppatori e dal *QTI Project Team* incaricato della creazione delle specifiche. Tuttavia è nella versione 2.0 che le specifiche vengono riprogettate e significativamente estese per risolvere le problematiche emerse con la diffusione della versione 1.2.1.[QTI205]. Tra le modifiche principali vi sono l'accorpamento del processo di elaborazione dei risultati all'interno del corpo della domanda, l'estensione dei formati supportati e una maggiore integrazione con i nuovi standard IMS. Nella versione più recente la 2.1, approvata nell'agosto del 2012 sono state rivisitate funzionalità di aggregazione degli item mancanti nella versione 2.0 [QTI2112].

L'IMS QTI *Working Group* si pone come obiettivo quello di implementare uno standard che possa essere utilizzato attraverso i più comuni VLE sia in ambito scolastico (scuole medie inferiori, superiori e università), che in ambito aziendale e nelle pubbliche amministrazioni. L'IMS QTI *Working Group* si prodiga per migliorare innanzitutto la possibilità di utilizzare batterie di test indipendentemente dal VLE a disposizione dell'utente e diverse fonti per le batterie di test su uno stesso VLE. A questo scopo vengono definiti attributi standardizzati, ovvero metadati, associabili all'intero test o a ogni suo singolo elemento costituente. Ad esempio i metadati specificano i criteri di *feedback* ovvero il riscontro alla risposta dell'utente, e di *scoring*, ovvero relativi al punteggio per la valutazione. La definizione di modalità di *packaging* e distribuzione contribuisce ulteriormente a migliorare l'interoperabilità tra VLE differenti. Al fine di garantire la continuità di sviluppo dei software per le tipologie di test esistenti e nuove, nonché la modalità di recupero dei risul-

tati in modo consistente, vengono inoltre create nuove interfacce dinamiche (API) ed estesi gli schemi per i report dei risultati, per gli *assessment*, il tracciamento e la presentazione dei test.

2.2 IMS QTI 1.2.1

2.2.1 Struttura del Modello ASI

Diversamente da quanto si può pensare, le specifiche QTI non si limitano a definire regole esclusive sulla struttura di test e domande, ma comprendono delle linee guida per la creazione, lo svolgimento, la presentazione, la distribuzione e la valutazione dei test. Il nucleo principale dello standard è rappresentato dal modello *Assessment Section Item* (ASI). Quest'ultimo comprende gli elementi XML necessari a creare la struttura di ogni test, combinabili in maniera diversa a seconda della complessità (si veda figura 2.1). Gli elementi che caratterizzano questo modello sono i seguenti:

- *item*: è il più piccolo elemento interscambiabile che non può essere composto da altri elementi dello stesso tipo. Ogni *item* contiene le informazioni relative alla domanda includendone contenuto, modalità di visualizzazione e pesi delle risposte;
- *section*: ognuna di esse viene utilizzata per rappresentare gruppi di domande dello stesso argomento per limitare l'estensione della sequenza di istruzioni. Una sezione al suo interno può contenere *item* o altre sezioni;
- *assessment*: è il contenitore di tutte le istruzioni relative alla modalità di utilizzo del test e relative all'elaborazione del risultato finale di tutti gli *item* coinvolti. All'interno di un *assessment* è obbligatoria la presenza di almeno una *section*. Non è possibile definire relazioni tra *assessment*.

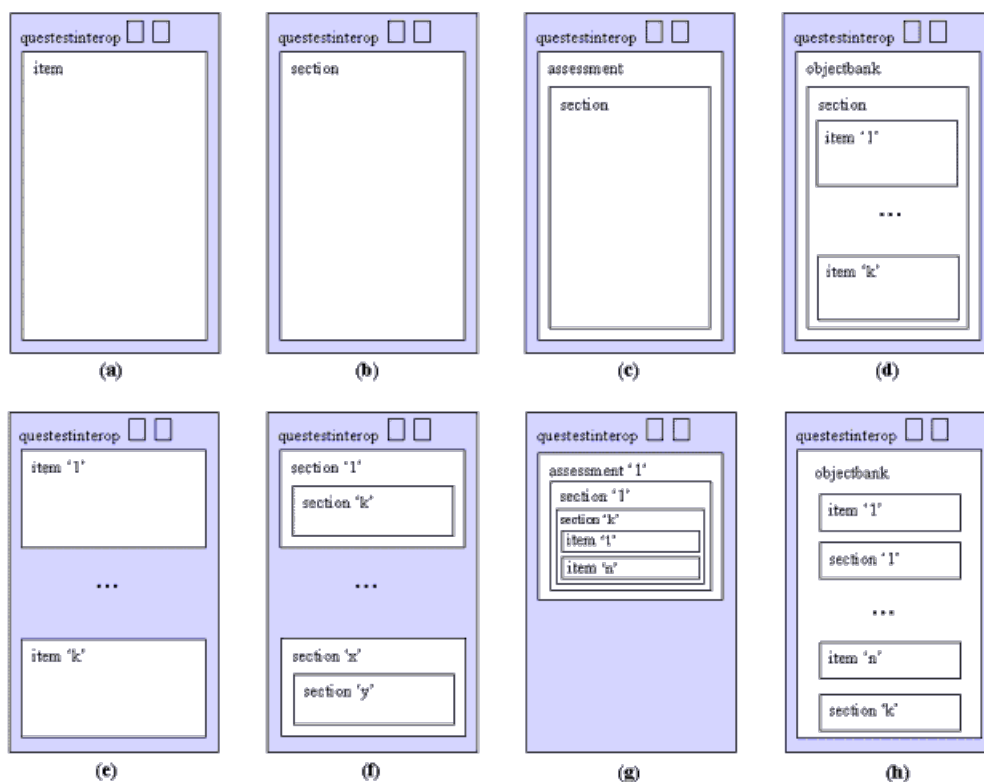


Figura 2.1: Combinazioni degli elementi nel modello ASI [QTIASI03]

- *object bank*: è una collezione di *item* e *section* con un identificatore univoco e metadati per facilitare la ricerca all'interno dei *repository*. Gli *object bank* vengono inseriti all'interno degli *assessment*.

Casi d'uso

I casi d'uso più significativi riguardano il processo di creazione di materiale per i test e la dei test che può avvenire secondo le modalità differenti di *High-Stakes* o *Low-Stakes*. Per *High Stakes Assessment* si intende quella tipologia di test che prevede una valutazione con archiviazione dei risultati. I *Low-Stakes Assesemnt* sono le tipologie di test adottate dai tutor per la preparazione. I *Low-Stakes Assesemnt* sono sprovvisti di sistemi di valuta-

zione ma comprendono suggerimenti e *feedback*. Le componenti chiave dei casi d'uso rappresentate in figura 2.2 presi in esame sono:

- *Sistema di Authoring*: processo che supporta la creazione e la modifica dei test;
- *Assessment Engine*: sistema incaricato dell'elaborazione dei risultati dei test attraverso la creazione di report con valutazioni e feedback;
- *LMS*: è il sistema responsabile dell'intero processo di creazione/esecuzione dei test;
- *Repository dei candidati*: database che contiene tutte le informazioni relative ai partecipanti;
- *Repository ASI*: database dei test già creati interno al sistema;
- *Repository ASI esterno*: database esterno al sistema che può essere importato attraverso le specifiche QTI.

Oltre ai ruoli fondamentali di tutor, autore e candidato negli scenari proposti intervengono anche il vigilante, ovvero colui che è addetto alla sorveglianza durante i test, il *psychometrician* incaricato di controllare l'efficacia dei test e lo *scorer* che ricopre il ruolo di persona che controlla le valutazioni.

Creazione del materiale per i test

Nella fase di creazione del materiale per i test l'autore avvia il sistema di *Authoring* e successivamente crea o modifica un *item* una *section* o un *assessment*. Il materiale creato viene esportato in un database esterno. Questo materiale può essere reimportato per creare nuove strutture ASI. La difficoltà nell'autore risiede nel determinare il tipo di risposta che può dipendere dall'obiettivo formativo in modo da poterla reindirizzare al tipo di visualizzazione più appropriata. L'autore deve anche fornire all'utente informazioni di supporto per facilitare la comprensione di come utilizzare al meglio il materiale creato.

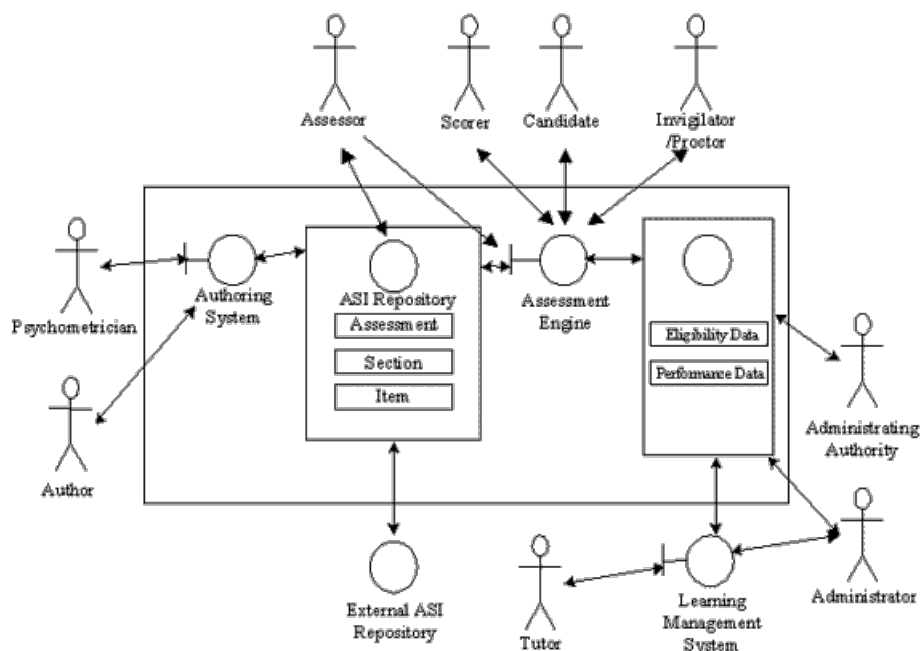


Figura 2.2: Rappresentazione delle componenti dell'*Assessment System* [QTIASI03]

I parametri di valutazione sono memorizzati all'interno dell'*assessment data* come dati di pre-test.

High-Stakes Assessment

L'*assessment* viene caricato all'interno del *repository* ASI pronto per essere utilizzato dal candidato, il quale una volta attivato il test, viene monitorato attraverso gli appositi strumenti del LMS da un vigilante. Tutte le risposte fornite dal candidato vengono elaborate andando a formare un set di risposte che vengono memorizzate nel *repository* ASI. In una modalità che può essere sia sincrona che asincrona, le risposte vengono elaborate dal *response processing* il cui risultato è contenuto nella struttura dell'*item*. Il risultato, richiede che siano impostate una serie di *evidence rules* che hanno il compito

di impostare i parametri utilizzati per la valutazione. Una volta che tutte le risposte alle domande sono state completate e quindi tutti gli *item* sono stati processati, vengono calcolati i pesi delle risposte e memorizzati come parte del *assessment record*. La parte finale del processo consiste nel mostrare al candidato il risultato, completo di feedback, se presente.

Low-Stakes Assessment

La tipologia di *assessment Low-Stakes Assessment* si presenta simile a quella *high stakes* ma con la particolarità che le singole domande ai test possono contenere suggerimenti o *feedback* in quanto pensata come una tipologia di preparazione a un test. Un tutor viene incaricato di selezionare il test da svolgere che viene memorizzato nel *repository* interno. Il candidato accede al test attivando la sessione di *tutoring*. Rispondendo alle domande il candidato innesca il meccanismo di elaborazione delle risposte che vengono processate attuando le politiche di *evidence rules* presenti nelle specifiche della prova. Come nel caso del *high-stakes assessment* i risultati vengono memorizzati nella struttura di *outcomes data*. Infine all'utente viene restituito il risultato con relativi *feedback* ed eventuali suggerimenti.

Creazione del contenuto

È possibile utilizzare i file QTI-XML prodotti per realizzare materiale di apprendimento non necessariamente legato a una forma di valutazione. Il processo prevede che si utilizzi un *authoring tool* per la creazione di contenuti attraverso l'ausilio di appositi *wizard*. Gli *item* creati non vengono raggruppati in un *assessment* e possono essere esportati singolarmente. Deve essere presente una corrispondenza tra modello di test nel LMS e specifiche dello standard per permettere la creazione adeguata del contenuto e per avvalersi del sistema di *response processing* per l'elaborazione senza alcun tipo di valutazione.

Question, Item e Response

La terminologia adottata nelle specifiche prevede che l'*item* sia definito come il blocco fondamentale che contiene una o più domande (*question*) e risposte (*response*). Per fare riferimento al tipo di risposta viene utilizzato l'identificativo **response type** e non *item-type* o *matter question type*. In figura 2.3 è mostrata la tassonomia dei **response type** disponibili legati alla creazione delle tipologie di domande. La prima suddivisione riguarda la creazione di un singolo tipo di **response type basic** o la possibilità di combinare diversi **response type composite** solitamente individuabili in *item* tematici. Per ogni **response type** sia *basic* che *composite* si può avere un'ulteriore suddivisione di tipo singolo, multiplo o ordinato. Nel caso ci sia un **response type** ordinato viene data importanza anche alla sequenza con la quale vengono date le risposte. Per ognuna di queste due tipologie esiste un'ulteriore suddivisione relativa al tempo di risposta.

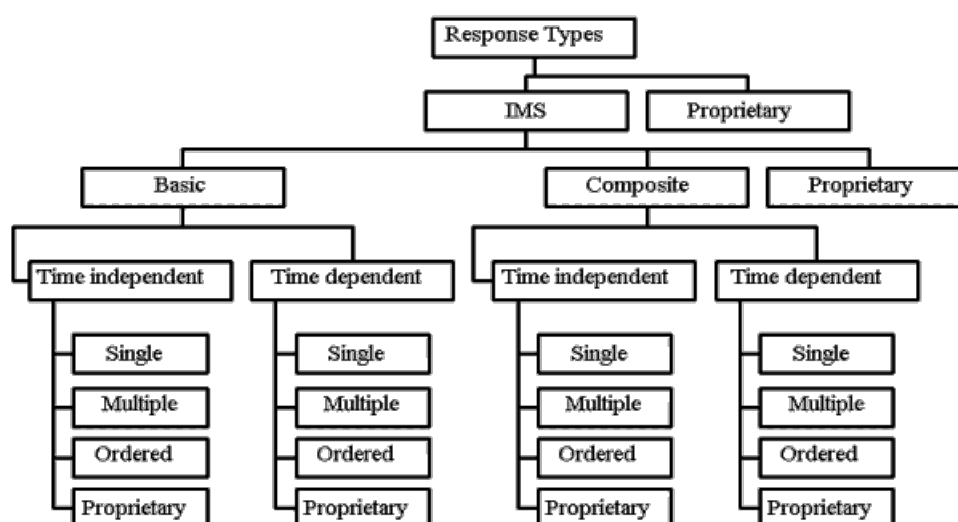


Figura 2.3: Classificazione dei response type [QTIASI03]

Nel **response type time dependent** è importante il tempo di risposta che viene memorizzato assieme ad essa. Al contrario, nel **response type time**

independent la componente temporale non è rilevante ai fini della valutazione.



Figura 2.4: *Rendering* differente per la stessa tipologia di **response type** [QTIASI03]

La relazione tra *rendering* della domanda e **response type** è visibile in figura 2.4. È possibile per una stessa domanda ottenere molteplici modalità di visualizzazione. In figura 2.4 si nota che sulla parte sinistra il quesito viene visualizzato tramite l'utilizzo di *radio button* mentre nella parte destra ci si avvale di un immagine dotata di *hot spot*. All'utente viene sottoposto lo stesso esercizio in modalità differenti, ma a livello di creazione della struttura la domanda rimane invariata. Questa proprietà permette di creare nuove visualizzazioni senza apportare modifiche alla struttura, in quanto *rendering* e struttura sono completamente indipendenti.

Tipologia di **response type**

Per ogni tipo di esercizio, sia *basic* che *composite* è possibile avere **response type** differenti. I **response type** supportati da questa versione delle specifiche sono i seguenti:

- *logical identifier LID*: è un **response type** di tipo identità ed è possibile creare domande di tipo *multiple choice*, *true o false*, *slider*, *multiple response*, *order objects*, *connect the points*, *match object*, *order object*, *drag object* e *drag target*;

- *X-Y coordinates* ; è una tipologia di **response type** basato sulle coordinate centrali di un oggetto è possibile creare esercizi di tipo *image hot spot, order objects, connect the point*;
- *string STR*: **response type** di tipo stringa non caratterizzato da tipologie comprendenti gruppi di utenti. Gli esercizi che appartengono a questa categoria sono di tipo *fill in the blanks, essay, short answer e select test*;
- *numerical NUM*: **response type** di tipo numerico caratterizzato da esercizi *fill in the blanks* o *slider*;
- *logical Group GRP*: è un **response type** di tipo identità o comprendente gruppi di identità sono disponibili per questa categoria esercizi di tipo *match objects, drag objects e drag target*.

Elementi

Il modello concettuale degli elementi presenti all'interno della struttura di un test è rappresentato in figura 2.5 dove è possibile anche capire quali sono le relazioni con gli elementi principali del modello ovvero *assessment, section, item*. Gli elementi raffigurati sono i seguenti:

- *activity selection*: elemento per la selezione dell'attività successiva;
- *outcome processing*: raccolta di tutte le valutazioni degli esercizi al fine di ottenere una valutazione del test intero;
- *scoring weights*: serie di pesi che sono assegnati ai risultati;
- *response processing*: elemento utilizzato per l'elaborazione delle valutazioni delle risposte dell'utente;
- *presentation*: rendering del contenuto con i possibili risultati;

- *examinee record*: elemento contenente insieme di istruzioni per la visualizzazione del risultato finale comprendente anche lo storico dell'utente relativo ad eventuali altri test;
- *outcome*: insieme di risultati da valutare dal sistema di response processing con l'applicazione delle metriche di giudizio stabilite ;
- *response*: elemento contenente le risposte fornite dall'utente;
- *flow*: flusso delle attività che vengono visualizzate ;
- *material*: elemento che contiene il contenuto da visualizzare.

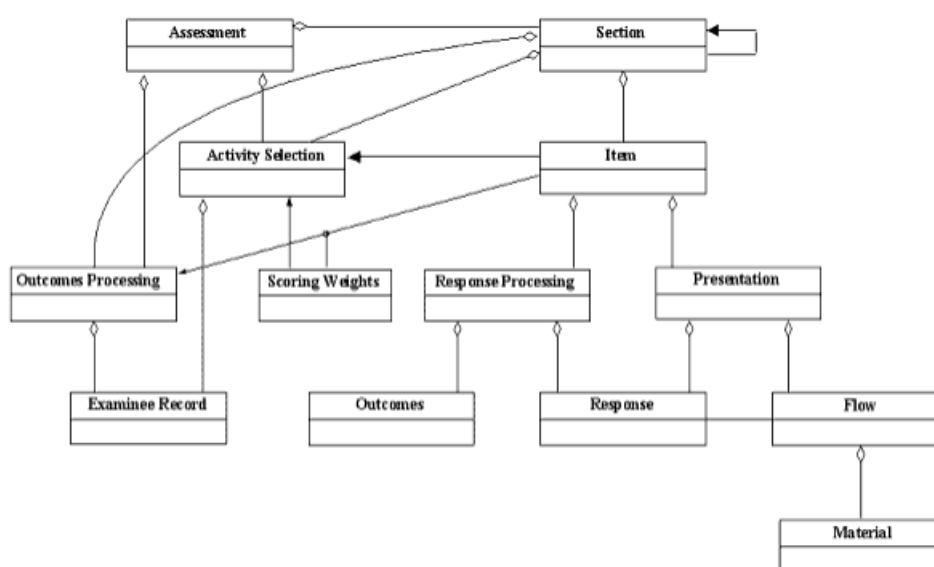


Figura 2.5: Relazione tra gli elementi principali del modello ASI [QTIASI03]

I tipi di risorse che si possono introdurre all'interno del corpo di un esercizio comprendono oltre a semplice testo anche l'utilizzo di immagini, file audio, *applet* e riferimenti a contenuti multimediali già esistenti. Tuttavia le specifiche non trattano a riguardo nessun vincolo implementativo e quindi la possibilità di poter utilizzare queste risorse viene gestita dalla piattaforma incaricata di distribuire gli esercizi.

2.2.2 Modello QTI Result Reporting

Il modello QTI *Result Reporting* si occupa della distribuzione dei risultati e della gestione dei report. Le specifiche relative a questo sistema di cui l'architettura è visibile in figura 2.6 sono state create per modellare le realtà relative alla memorizzazione dei risultati. Questa memorizzazione può avvenire:

- in modalità permanente (*permanent storage*) dove vi è la necessità di archiviare i risultati per un lungo periodo;
- tramite la raccolta di una serie di risultati (*results aggregation*) provenienti da sistemi differenti e quindi sorge il bisogno di memorizzarli in un unico sistema.

Le specifiche delineano anche un sistema per la comunicazione tra i LMS o LCMS e un eventuale sistema di distribuzione dei test esterno (*Assessment Delivery System*). Altri due aspetti affrontati nelle specifiche riguardano il *Real Time Interactive* e l'*Internal Interchange Representation*. Il primo si occupa dello scambio in *real time* tra *Assessment Delivery System* e il motore di gestione delle prove *Assessment Control Engine* che può essere esterno o interno ai LMS. Il secondo si preoccupa della portabilità tra le diverse implementazioni dei processi.

Struttura

L'elemento principale della struttura del modello QTI Result reporting è **result**. Questo elemento viene utilizzato per la memorizzazione dei risultati e per eventuali valutazioni. Un altro elemento presente nella struttura è **context** che detiene le informazioni dei partecipanti quali nome, id, nome della prova ecc.. Nel **summary_result** sono comprese informazioni riepilogative a una particolare istanza di valutazione. Ogni risultato può contenere una sola istanza di **summary_result**. Appartengono poi al modello anche gli elementi **assessment_result**, **section_result**, **item_result** utilizzati per

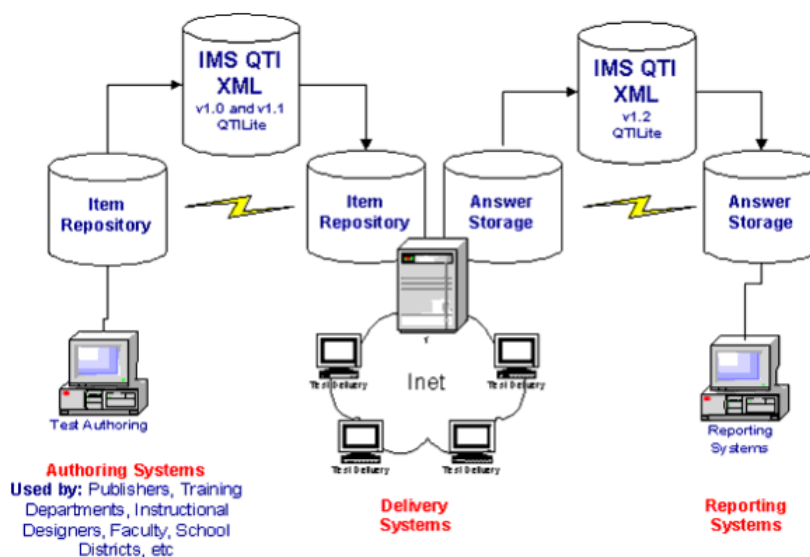


Figura 2.6: Processo di distribuzione dei report e dei risultati [QTIRRM03]

salvare le informazioni relative al compimento o meno di una determinata parte del test o del test stesso nel caso si tratti di `assessment_result`. La combinazione di questi elementi visibile in figura 2.7 permette di poter scambiare i riepiloghi dei test in maniera indipendente dal corpo della verifica. Inoltre la struttura strettamente legata al modello ASI ma indipendente permette la possibilità di estensioni degli schemi senza dover modificare le componenti del test.

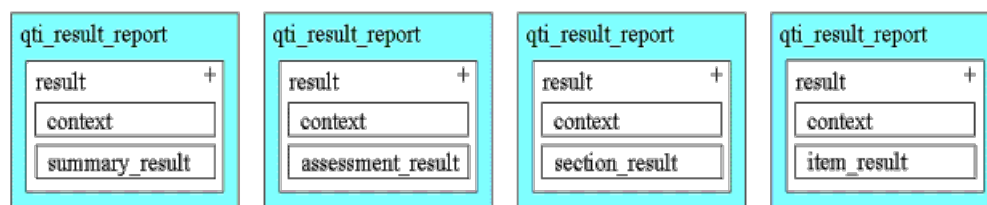


Figura 2.7: Combinazioni degli elementi nel modello *Result Reporting* [QTIRRM03]

2.3 IMS QTI 2.1

Nell'ultima versione delle specifiche IMS QTI 2.1, che ancora non è stata introdotta all'interno dell'IMS CC [QTI2112], sono state perfezionate le strutture introdotte nella riprogettazione dello standard avvenuta con la pubblicazione delle specifiche IMS QTI 2.0 [QTI205].

Come si può notare in figura 2.8 il modello concettuale è stato semplificato notevolmente, introducendo la classe `assessmentItem`, che raggruppa nella sua struttura il modello ASI della versione IMS QTI 1.2.1 [QTIXML03]. È stata inoltre introdotta la classe `itemSession` che delinea il ciclo di vita di un *item*. All'interno di un *item* di classe `assessmentItem` sono contenuti, sottoforma di attributi tutte le caratteristiche della prova come id, title, language ecc.. Tutte le classi relative a come presentare la domanda `templateProcessing`, a come valutare i risultati `responseProcessing` alla tipologia di risposta richiesta `variableDeclaration` e il corpo delle domande `itemBody` sono legate all'elemento *item* di classe `assessmentItem`. Una differenza sostanziale che si nota con la struttura precedente offerta dallo standard è l'integrazione della parte relativa al *response processing* all'interno della stessa struttura facilitando il processo di distribuzione dei risultati.

Si identifica con un elemento di classe `itemSession` una sessione di test del candidato nel quale vengono memorizzati tutti i risultati delle singole domande.

Nel diagramma di figura 2.9 è presente l'intero ciclo di vita di una sessione. La sessione viene inizializzata quando l'*item* associato viene reso disponibile per il candidato (*initial state*). Le azioni del candidato influiscono sullo stato del `itemSession` fino alla fine della sessione che si conclude con il termine della prova (*closed state*) e l'invio dei risultati.

Gli stati della sessione vengono modificati in base alla reazione dell'utente sulla tipologia di esercizio proposto che possono variare in base alle caratteristiche specifiche della prova come ad esempio prova adattiva o prova a tempo. In questa riprogettazione della struttura di test viene infatti introdotta la possibilità di creare esercizi adattivi (*Adaptive Question*) che

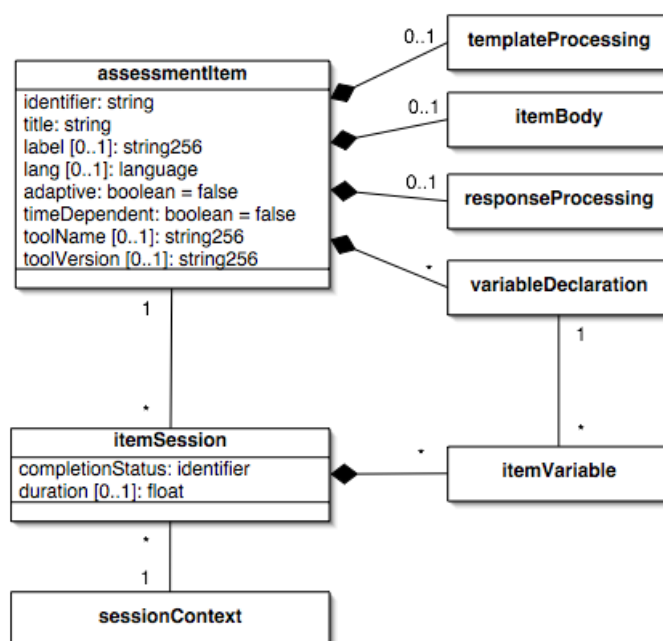


Figura 2.8: Strutture delle classi nelle specifiche IMS QTI 2.1 [QTI2112]

permettono di sottoporre domande diverse al candidato in base alle risposte ricevute. Questo tipo dinamico di domanda non può essere processata con la stessa modalità di *response processing* pertanto nella sessione sono appositamente aggiunti degli stati intermedi per prevedere il recupero di risposte già effettuate. L'accorpamento delle due strutture della versione precedente (modello ASI e modello *Result Reporting*), è anche dovuto alla gestione di questo tipo di domanda.

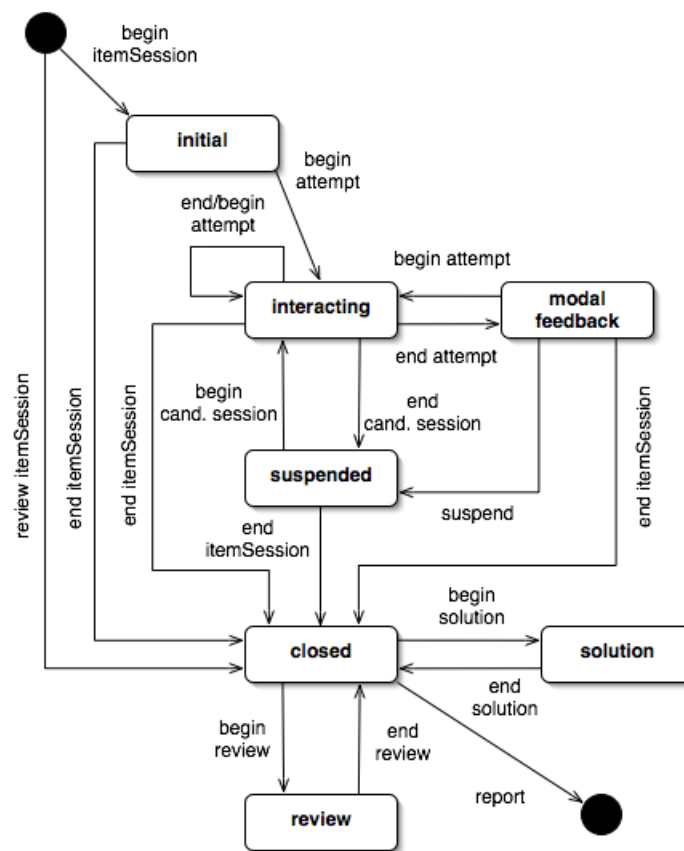


Figura 2.9: Ciclo di vita di un *item session* [QTI2112]

Capitolo 3

Piattaforme e-learning di Ateneo

L'offerta di piattaforme *e-learning* esistenti sul mercato è molto ampia, comprendendo attualmente più di un centinaio di prodotti proprietari e non. Tra queste, l'Ateneo di Bologna si affida principalmente a Moodle [MOO13], un LMS *open source* supportato da una vastissima comunità. Presso l'Università di Bologna le piattaforme *e-learning* sono utilizzate in diversi contesti, ma ricoprono un ruolo sicuramente fondamentale in ambito linguistico. Esistono infatti due piattaforme gestite dal Centro linguistico di Ateneo (CLA): Alfacert e Dalia [Alf13] [DAL13]. Dal momento che, ad oggi, non sono ammessi all'interno dell'Ateneo corsi completamente *online*, i docenti possono avvalersi di piattaforme *e-learning* come strumenti di supporto alla didattica, per il caricamento di materiale o per l'utilizzo di strumenti collaborativi. I compiti di gestione e di formazione dei docenti all'utilizzo degli strumenti a loro disposizione è affidata al Settore *e-learning*. Oltre a Moodle, una versione modificata di ATutor [ATu13], un altro LCMS *open source*, viene utilizzata nell'ambito del progetto A³, come illustrato nel seguito del capitolo.

Dopo una prima introduzione a Moodle, nel capitolo vengono descritte le piattaforme e le modalità di esercizio adottate nei corsi e modificate a seconda dei diversi ambiti di utilizzo in Ateneo. Nella parte finale viene inoltre

presentato in dettaglio AContent [ACo13], l'*editor* di contenuti integrato in ATutor che è stato utilizzato per lo svolgimento di questo lavoro di tesi.

3.1 Moodle

Moodle è l'acronimo di *Modular Object-Oriented Dynamic Learning Environment*. Si tratta di un progetto nato nel 1999 dall'idea della tesi di dottorato di uno studente australiano, Martin Dougiamas. Una delle caratteristiche principali di Moodle, rispetto ad altri sistemi, è la forte attenzione dedicata agli aspetti pedagogici dell'apprendimento mediato dalle tecnologie.

L'attenzione alla conformità agli standard, poco presente nelle prime versioni, è invece oggetto di miglioramento continuo nelle versioni attualmente in uscita. Nella versione più recente di Moodle la 2.4.1, sono supportati pienamente gli standard SCORM e IMS CP e CC ma non è presente alcuna esportazione o importazione di esercizi in formato IMS QTI. Nella documentazione dell'ultima versione [MOO13] viene precisato che anche importando un pacchetto IMS CC provvisto di esercizi c'è la possibilità che alcuni di essi non vengano implementati proprio per il fatto che questo standard non è pienamente supportato. Nella versione 1.9 di Moodle attraverso un'estensione sviluppata dalla comunità degli utilizzatori era possibile importare ed esportare batterie di test aderenti alle specifiche IMS QTI. Le nuove *feature* introdotte con la versione 2.0 come ad esempio il meccanismo di *hashing* all'interno del file system hanno reso inapplicabile l'estensione alla nuova versione. La possibilità di integrare una moltitudine di estensioni (*plug-in*) implementati dalla comunità e il continuo sviluppo rendono la piattaforma una delle più valide.

L'entità principale creabile all'interno di Moodle è il corso che può essere visto come un contenitore di materiale didattico e un insieme di moduli, ognuno orientato all'utilizzo di particolari strumenti legati alla didattica, come ad esempio il modulo "lezione". Il modulo lezione permette la creazione di percorsi interattivi sia semplici che complessi integrati da strumenti di

verifica. All'interno di un corso è possibile inserire strumenti di ausilio collaborativo tipo *forum*, *wiki* e *chat*. L'aggiunta di risorse invece arricchisce il corso di contenuti caricati sul *server* nella forma di pagine create con strumenti di *authoring*, collegamenti ad archivi o applicazioni Web esterne. La tracciabilità delle attività svolte dallo studente è affidata a sistemi di *tracking* integrati consultabili dal docente in qualsiasi momento all'interno delle impostazioni del corso.

Su Moodle sono implementate delle politiche di gestione degli utenti attraverso la creazione di gruppi e ruoli. Attraverso questi meccanismi è possibile realizzare delle vere e proprie classi, e permettere l'accesso a determinati contenuti solo ad un gruppo ristretto di utenti. Oltre ai ruoli di studente, docente e amministratore sono stati aggiunti nelle ultime versioni ruoli come gestore e tutor permettendo così una diversificazione delle *policy* maggiore rispetto alle versioni precedenti del sistema. L'idea di modularità ha permesso, grazie alla comunità lo sviluppo di un sistema di autenticazione *single sign-on* integrato sulle piattaforme di Ateneo attraverso l'utilizzo dell'applicazione *Shibboleth* [SHI13].

Moodle è un LMS scritto prevalentemente in PHP basato su architettura LAMP (Linux, Apache, PHP, MYSQL) anche se può funzionare anche su un server IIS(Internet Information Service) Microsoft. Inoltre le ultime versioni permettono anche la possibilità di connettersi a un database ORACLE.

3.1.1 Tipologie di esercizi

Un aspetto fondamentale dei corsi *e-learning* è la verifica di ciò che si è appreso attraverso prove intermedie o finali inseribili all'interno di un unità didattica. Avvalendosi del modulo "quiz" il docente può progettare e realizzare dei test che possono comprendere domande a scelta multipla, domande con risposta vero/falso, domande con brevi risposte aperte ecc... Le domande vengono memorizzate in un database, possono essere classificate per categoria ed essere riutilizzate all'interno dello stesso corso o di corsi differenti. La classificazione in categorie permette di creare categorie pubbliche o priva-

te e di spostare i test in maniera estremamente semplice. I test e le singole domande possono essere ripetuti più volte. È possibile omettere la visualizzazione delle risposte corrette, memorizzando automaticamente i risultati, provvedere un *feedback* immediato o fornire la risposta corretta.

Le tipologie di esercizi implementate in Moodle sono le seguenti:

- *risposta breve*: la risposta breve prevista per questo tipo di domanda viene confrontata con la stringa di risposta attesa. È possibile inserire immagini ed attivare l'opzione "case-sensitive", che differenzia una parola scritta con caratteri maiuscoli o minuscoli. In questa tipologia di domande, come per la ricerca delle stringhe su bash, è possibile un carattere "wildcard" (*) per permettere più di una risposta positiva con eventuali pesi differenti;
- *numerica*: una domanda numerica risulta simile alla domanda a risposta breve, con la differenza che le domande possono avere una tolleranza di errore nelle risposte. Non si ha una risposta possibile ma un *range* di valori per le risposte;
- *componimento*: questa tipologia di domanda permette di creare un piccolo *essay*. Non vi è nessun tipo di meccanismo di correzione automatica ed è richiesto l'intervento del docente per la correzione della prova;
- *corrispondenza*: questa modalità di quesito viene presentata al candidato come una serie di domande da collegare alle possibili risposte. Ogni corrispondenza ha un singolo peso, determinante nel risultato della prova;
- *corrispondenza con domande casuali*: tipologia simile alla corrispondenza semplice, ma con la possibilità di mostrare all'utente in modalità casuale le risposte, il cui ordine non è quindi scelto dall'autore del test;
- *domanda calcolata*: questa tipologia offre la creazione di singole domande numeriche tramite l'uso di caratteri "wildcard" (*) che vengono

sostituiti da valori effettivi quando il quiz è eseguito. In figura 3.1 sono riportate le opzioni che si possono impostare nell'*editor* della domanda con alcuni input di esempio. Nel testo della domanda, e nel campo "Formula della risposta corretta" si possono vedere "a" e "b". Queste e ogni altro identificativo sono usate come caratteri "wildcard". I possibili valori da sostituire sono impostati o generati in una pagina successiva di aiuto per l'impostazione delle domande calcolate. La formula di esempio usa l'operatore "+". Altri operatori possibili sono "-", "*", "/", e "%" dove "%" è l'operatore modulo. È anche possibile utilizzare alcune funzioni matematiche in stile PHP supportate dall'*editor*.

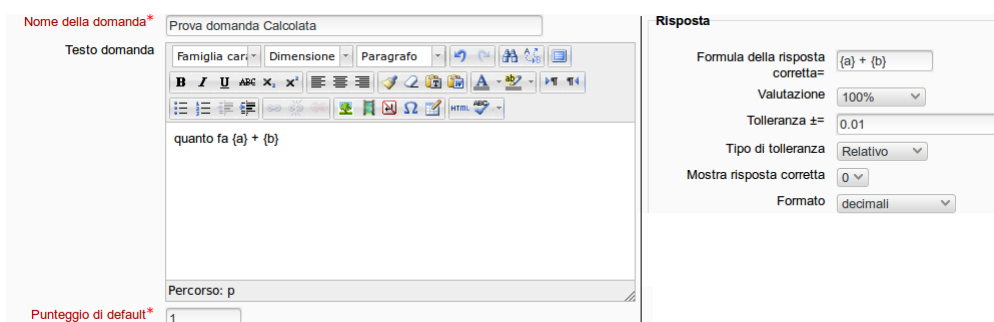


Figura 3.1

- *risposta multipla*: in questa tipologia di domande, per cui le risposte possono essere anche immagini, il partecipante può scegliere tra più soluzioni. È possibile creare con questo tipo di domanda anche domande a risposta singola. Nella prima si possono avere più risposte corrette mentre nella seconda si ha una sola risposta valida. Il peso di una domanda può essere negativo o positivo e si può scegliere il punteggio attribuibile ad ogni risposta che può essere supportata da un relativo *feedback*;
- *domanda calcolata semplice*: appartengono a questa categoria domande calcolate semplificate, dove i valori richiesti sono numerici e vengono presentati all'utente in forma di scelta multipla;

- *risposta inglobata*: questo tipo di domanda è denominata “*Cloze*” ed è simile alla tipologia “*fill in the blanks*” in IMS QTI. Le domande consistono in un blocco di testo (in formato Moodle) che ha uno o più spazi vuoti da riempire con varie possibili risposte. Il riempimento del testo può avvenire come tipo ‘scelta multipla’, ‘risposta breve’ o ‘numerica’. Attualmente non è prevista un’interfaccia grafica per preparare queste domande, pertanto l’autore deve riscrivere le domande all’interno dell’*editor* di testo utilizzando la sintassi appropriata;
- *scelta multipla calcolata*: è una combinazione delle tipologie risposta multipla e domanda calcolata;
- *vero o falso*: questa tipologia di domanda può comprendere anche un’immagine come corpo della domanda e il candidato può selezionare una risposta tra vero e falso. Questa domanda può fornire *feedback* sia positivi che negativi.
- *descrizione*: non si può considerare questa tipologia come una vera e propria domanda. Lo studente può aggiungere del testo ma non sono previsti meccanismi di valutazione;

Per ogni tipologia di domanda, può esser impostata una penalità nel caso di risposta errata in quei test che permettono allo studente di ripetere più volte la stessa domanda.

Le domande, o le intere categorie di domande presenti all’interno della piattaforma possono essere esportate in tre formati principali quali GIFT, Moodle XML e XHTML. Tuttavia, non tutte le tipologie di domande possono essere esportate con successo. Può capitare che non tutti i formati riescano ad incapsulare tutte le caratteristiche messe a disposizione da Moodle per la creazione di una domanda.

Il formato GIFT è il formato più completo, che permette di esportare le domande attraverso la creazione di un semplice file di testo.

Moodle XML è un formato proprietario di esportazione delle domande per mezzo di file XML modificati *ad hoc* dalla piattaforma. Questi file possono

essere importati successivamente in un'altra categoria all'interno della stessa piattaforma o utilizzati in qualche altro processo, come ad esempio una trasformazione XSLT. Il formato XHTML infine prevede il salvataggio completo della pagina, per il quale ogni domanda viene posizionata all'interno di un *tag div*. Come si può facilmente notare l'esportazione di test creati all'interno della piattaforma non è permessa in IMS QTI. L'implementazione di un modulo che permetta questa operazione è affidato alla comunità.

La moltitudine di opzioni presenti nella creazione di una domanda su Moodle ne limita la completa esportazione. Inoltre una volta che una domanda viene importata, il sistema la converte nel formato proprio per garantire la possibilità di sfruttare tutte le potenzialità offerte dal sistema.

A differenza delle poche possibilità offerte per l'esportazione, esistono una moltitudine di formati che possono essere importati su Moodle. Ne sono esempi il formato *Black Board* e *BlackBoard V6+* esercitazioni create con la piattaforma *BlackBoard* [Blb13]. Quest'ultima è un VLE proprietario diffuso a livello internazionale. È possibile importare domande create con la suite di *Examview*, una piattaforma per la creazione degli *assessment* anch'essa proprietaria. Attraverso il software *Hot Potatoes* [HOT13] vi è la possibilità di importare test o categorie di test con domande di varie tipologie, comprese quelle a scelta multipla e a risposta breve, cruciverba, esercizi di riempimento, ricostruzione di frasi, corrispondenze o riordinamento. È inoltre supportata l'importazione di esercitazioni create con *Learn Wise* [LEA13], un LMS proprietario diffuso negli ambienti accademici. L'importazione può avvenire in formato GIFT e Moodle XML. È possibile importare singole domande a parola mancante o risposta inglobata create in precedenza su una piattaforma Moodle.

3.2 Piattaforme Moodle di Ateneo

Attualmente le piattaforme Moodle di Ateneo gestite dal Settore *e-learning*, escludendo quelle gestite dal CLA, sono tre: Moodle per la gestione degli inse-

gnamenti, Moodle per la gestione dei progetti *e-learning* di Ateneo e Moodle per la gestione dei master *postlauream* che si svolgono in Ateneo.

Le piattaforme Moodle attualmente installate sono tutte alla versione 1.9, risalendo a circa 3 anni fa la creazione della piattaforma. È in via di sviluppo un progetto di aggiornamento la piattaforma degli insegnamenti alla versione 2.4 integrata con il database ORACLE di Ateneo per la gestione diretta dei dati relativi alla programmazione didattica. Sulle piattaforme Moodle attualmente *online* non sono state fatte particolari modifiche ai moduli relativi ai test in quanto non se ne vedeva la necessità. La richiesta della possibilità di poter usufruire di uno spazio per creare un corso in *e-learning* va inoltrata al Settore *e-learning* attraverso un modulo di richiesta apposito. Dal punto di vista gestionale/della formazione il Settore prevede alla formazione di tutor o docenti che fanno richiesta di un corso *e-learning* sulla piattaforma di Ateneo e garantisce assistenza nel caso vi siano dei problemi con l'accesso o con l'utilizzo degli strumenti della piattaforma.

3.2.1 Alfacert

La piattaforma Alfacert (Alfabetizzazione e Certificazione) [Alf13] è gestita dalla sede del Centro Linguistico di Ateneo della Romagna, è una piattaforma Moodle aggiornata alla versione più recente (2.4) che viene utilizzata per l'erogazione di corsi di lingua per i Corsi di Studio che si svolgono nelle sedi distaccate dell'Ateneo. Questa piattaforma viene utilizzata come strumento di supporto ai corsi che si tengono in aula, ma i contenuti relativi alle unità e gli esercizi di verifica sono inseriti da un reparto tecnico specializzato e non dal docente. Il docente o collaboratore linguistico prepara il materiale didattico su foglio elettronico o su documento di Word che il reparto tecnico del centro linguistico si occupa di trasformare il materiale e importare sulla piattaforma. Questo vale sia per il contenuto didattico che per gli esercizi. La gran parte del materiale contenuto all'interno della piattaforma è stato importato previo trasformazione attraverso meccanismi di conversione implementati dai tecnici le centro linguistico. Per meccanismi di trasformazione

si intendono programmi che convertono file Word o Excel in file XML, poi resi compatibili con lo standard SCORM. Gran parte degli esercizi vengono creati per mezzo del *software* Hot Potatoes e vengono dotati di soluzione e di controllo delle risposte corrette come quello di figura 3.2. I docenti o colla-

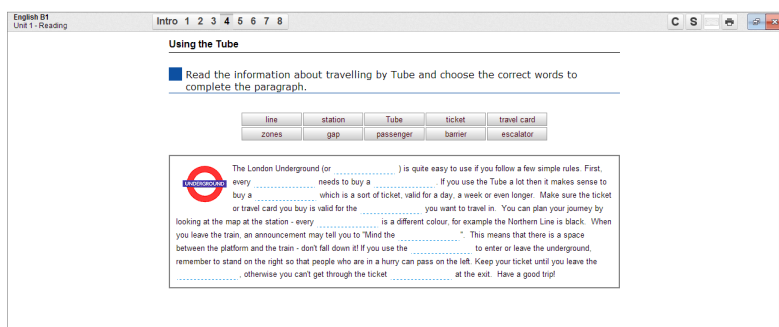


Figura 3.2: Esercizio tipico su Alfacert

boratori linguistici vengono formati per un utilizzo efficace della piattaforma dal reparto tecnico che fornisce anche assistenza.

3.2.2 DALIA

Il progetto DALIA (Didattica per l'Apprendimento delle Lingue in Autonomia) nasce nel 2009 dalla necessità di armonizzare i percorsi formativi offerti dai due centri linguistici dell'Università di Bologna, CILTA e CLIRO, ora accorpatisi in un unico centro linguistico, il CLA [Alf13]. Il progetto consisteva nel migrare i contenuti dal vecchio sistema di erogazione dei corsi su VLE ILIAS [ILIA13] ad una piattaforma Moodle. In questo progetto sono stati coinvolti tecnici del centro linguistico che hanno realizzato un sistema di creazione dei contenuti da affiancare agli strumenti della piattaforma. Questo *tool* creato permette di creare pacchetti di contenuto o esercizi SCORM compatibili che possono essere importati su Moodle. La quantità di esercizi e le interfacce disponibili su Moodle sembravano limitate cosicché si è passati all'implementazione di nuovi esercizi e interfacce al fine di creare un pacchetto di esercizi valido e specifico per la preparazione dell'utente all'apprendimento

delle lingue. In un documento fruibile sul sito del centro linguistico vengono elencate le principali tipologie di esercizi con relativa modifica della scelta dell'esercizio [CIL13]. Di seguito vengono elencate le tipologie di esercizi disponibili che aiutano l'utente a migliorare le capacità lessico-sintattiche della lingua oggetto di studio:

- *accoppiamento linea-categoria*: questa tipologia di esercizio di figura 3.3 permette di associare più elementi a una stessa categoria. Possono essere incluse file multimediali, immagini e suoni per creare le associazioni;

The screenshot shows a web interface for a language exercise. At the top, there are logos for 'ALMA MATER STUDIUM UNIVERSITÀ DI BOLOGNA' and 'CILTA CENTRO INTERFACOLTÀ DI LINGUISTICA TEORICA E APPLICATA T. HEILMANN'. The page title is 'Home > FR A2 > Française A2 v.1'. The exercise is titled 'Quelques mots français dans la langue italienne : 1'. The instruction is 'Choisissez un mot et transportez-le sur l'image de sa catégorie.' There are three categories: 'Amusement' (with a boat image), 'Cuisine' (with a food image), and 'Habits' (with a woman in a dress image). A list of French words is provided: abat-jour, baguette, blouson, brioches, béchamel, camembert, champignon, crêpe, dessert, décollante, foulard, lingerie, mouquette, vin brûlé, à la croque.

Figura 3.3: Esercizio accoppiamento linea categoria DALIA

- *riordino frasi e paragrafi*: questo esercizio è una sorta di corrispondenza che utilizza funzioni di *drag & drop* dove la seconda scelta è l'ordinamento della frase;
- *ordinamento delle parole*: in questa tipologia viene utilizzata la stessa modalità del riordinamento di frasi ma si limita il box di contenimento ad un'unica parola e l'interfaccia viene presentata con un *layout* differente;

- *riempimento spazi vuoti*: è l'esercizio base di Moodle, con la particolarità che la creazione avviene in una modalità più intuitiva attraverso il *tool* di editing implementato dal centro linguistico;
- *scelta multipla classica*: è l'esercizio classico di scelta multipla o singola con la possibilità di inserire una moltitudine di opzioni e di singole domande;
- *componimento*: è la domanda aperta di Moodle implementata nel contesto delle lingue. Questa domanda non viene corretta in automatico dal sistema ma viene fornita la possibilità di stampare la risposta ed eventualmente avvalersi di un docente per la correzione;
- *accoppiamento*: consiste nell'esercizio base di Moodle di corrispondenza tramite un'interfaccia più intuitiva;
- *cruciverba*: tipologia di esercizio di figura 3.4 non presente su Moodle che prevede di poter creare cruciverba;

The screenshot shows a Moodle page for a crossword puzzle. The page title is "Crossword". Below the title, there is a grid of letters and numbers. To the right of the grid, there are two columns of questions:

Orizzontali

5. Rivista is _____ in English. (7)
7. _____'s Jenny from? She's English. (7)
8. The plural of key. (7)
9. Enrico's _____, He's from Rome. (7)
10. What's 'busta per lettera' in English? An _____. (7)

Verticali

1. 'Are you _____ Naples?' Yes, I am. (7)
2. He's from Paris. He's _____. (7)
3. She isn't American. She's _____. (7)
4. 'This' is singular. '_____' is plural. (7)
6. We're _____, We're from Berlin. (7)
7. _____ are these in English? They're apples. (7)

Figura 3.4: Esercizio cruciverba su DALIA

- *ascolta e ripeti*: per questa tipologia di esercizio ci si avvale di un'*applet* java denominata “nanogong” [Nan13] per effettuare le registrazioni all'interno della piattaforma. L'esercizio prevede di registrare la propria voce e di confrontarla con una traccia audio che contiene l'esatta pronuncia delle parole richieste.

In questi esercizi sono integrati meccanismi di correzione e soluzione che permettono di confrontare subito i risultati ottenuti.

3.3 ATutor

ATutor è un LCMS *open source* progettato come strumento di insegnamento e apprendimento *online*. Nasce come progetto sviluppato da un gruppo di ricerca dell'Università di Toronto con la collaborazione dell'ATRC (*Adaptive Technology Resource Centre*) [ATu13]. Il suo continuo sviluppo è reso possibile grazie al contributo di individui, istituzioni e Università situate in diversi Paesi, tra cui Canada, USA, Italia e Norvegia.

I punti di forza della piattaforma sono dati dal fatto che è un software *open source*, pertanto è distribuito liberamente attraverso il sito Web ufficiale, modificabile ed integrabile con altri sistemi senza restrizioni legali. La struttura modulare e non monolitica permette la progettazione e lo sviluppo di servizi aggiuntivi senza dover modificare il *core* del sistema. ATutor presenta un'interfaccia ben organizzata ed accessibile, conforme alle linee guida di accessibilità previste dal W3C [W3C13]. Poiché l'interfaccia utente è molto semplice, gli utenti stessi possono imparare velocemente ad utilizzare l'applicazione in modo efficace. La navigazione, inoltre, è molto chiara per consentire anche agli utenti con difficoltà di apprendimento un corretto utilizzo, senza necessità di una formazione preventiva sul funzionamento. La parte di *authoring* per creare contenuti riutilizzabili e facilmente condivisibili è affidata a un modulo denominato AContent che viene descritto in dettaglio alla fine di questo capitolo. Le connessioni con i principali *social network* vengono gestite attraverso il modulo *ATutor Social*. Gli standard supportati

per quanto riguarda la gestione degli LO sono IMS CC, IMS CP e SCORM. In aggiunta lo standard IMS QTI è supportato sia in importazione che in esportazione. Nel prossimo capitolo viene descritto in dettaglio l'aspetto di implementazione degli esercizi in quanto parte integrante del progetto di tesi.

3.3.1 Piattaforma A³

Il progetto Ambiente Accessibile d'Apprendimento (A³) sviluppato dal Dipartimento di Scienze dell'Informazione (divenuto DISI) dell'Università di Bologna è nato nel 2003 dalla necessità di insegnare competenze di informatica di base in quei Corsi di Studi non prettamente a indirizzo informatico. Come piattaforma di partenza la scelta è ricaduta su ATutor in quanto al tempo era il sistema più in linea con gli standard e più facilmente modificabile. Il progetto si poneva come obiettivo quello di creare dei corsi divisi per argomento che comprendessero le nozioni base di informatica con il supporto di prove di verifica integrate all'interno del sistema. Oltre alla modifica dei CSS e del layout delle singole pagine sono state aggiunte delle funzionalità per la gestione dei gruppi, per prevedere la creazione di classi, e strumenti per la gestione delle *policy* degli utenti. La parte relativa agli esercizi è stata pesantemente ridisegnata, non tanto per variare (da un punto di vista delle tipologie) gli esercizi ma per la loro suddivisione e presentazione all'utente. In figura 3.5 è mostrato un esempio di presentazione di esercizio su A³. Per l'importazione delle domande all'interno della piattaforma è stato creato un sistema personalizzato: attraverso un file XML sono specificate le direttive relative alla creazione di una istanza di test (inteso come contenitore di singoli esercizi) per ogni singola domanda con il vantaggio di aggirare le problematiche al raggruppamento delle domande.

Analogamente agli altri casi, anche in questo progetto è stato implementato un processo di post-produzione che trasforma il contenuto didattico da file in formato Word a LO. Il sistema di post produzione *Immediate Site Activator* (ISA) è stato sviluppato dal dipartimento per velocizzare il sistema di creazione dei contenuti.

The screenshot shows the A3 (Ambiente Accessibile d'Apprendimento) interface. At the top, there is a header with the A3 logo and the text 'AMBIENTE ACCESSIBILE D'APPRENDIMENTO' and 'ABILITÀ INFORMATICHE'. Below the header, there is a navigation bar with links for 'Accessibilità', 'Supporto', 'Home', and 'Logout'. The user's name 'Sara Cenni' is displayed. The main content area shows a breadcrumb trail: 'Fondamenti di elaborazione multimediale > Concetti di base > Domande in itinere'. The page is titled 'Test n°1' and asks the user to 'Indicare la risposta vera'. The question is 'Quali dei seguenti media non è di tipo continuo?'. The options are: 'Immagine', 'Video', 'Animazione', 'Audio', and 'Non rispondo'. The 'Non rispondo' option is selected. There is a 'Invia Test Alt-s' button below the options. On the left side, there is a navigation menu with categories like 'Obiettivi', 'Introduzione (Video)', 'Concetti di base', and 'Domande in itinere'. The 'Domande in itinere' category is currently selected.

Figura 3.5: Classica pagina di esercizi su A³

3.3.2 AContent

AContent si può classificare come un *open source learning content authoring system* [ACo13] e nasce dalla scorporazione della componente di creazione dei contenuti da ATutor. AContent permette la redazione di contenuti didattici per l'apprendimento interoperabili e adattivi. Essendo AContent una porzione di ATutor e non contenendo tutte le funzionalità di un LCMS, il codice presenta una struttura monolitica che può essere modificato per le proprie esigenze dopo uno studio del *kernel* e del motore di *rendering*. Uno dei punti di forza di AContent è la conformità con gli standard di *e-learning*, garantisce la portabilità del materiale su sistemi differenti e la facilità di utilizzo nella creazione e nella modifica dei contenuti. La scelta di separare i due sistemi (AContent e ATutor) permette di sfruttare il *tool* di *editing* come *repository* comune di più piattaforme, fruibili da remoto attraverso *web service* (i due sistemi si trovano su macchine diverse). In AContent è integrato l'*editor* per la produzione di contenuti *what-you-see-is-what-you-get* (WYSIWYG)

TinyMCE [Tin13] che permette di introdurre funzioni matematiche, o equazioni in notazione scientifica in \LaTeX , immagini, video, animazioni, tracce audio e *link* in aggiunta alla funzionalità di *editor* di CSS. La versione attuale di AContent (1.2 stable) supporta la versione 1.0 di IMS CC e IMS CP. Le specifiche IMS QTI 1.2.1 per i test sono infine pienamente supportate con la possibilità di esportare in test in IMS QTI 2.1.

Capitolo 4

Progetto

La diffusione negli ultimi anni delle metodologie di *e-learning* è stata in parte frenata dal difficile utilizzo delle piattaforme da parte di docenti e *tutor* che spesso richiedono l'intervento di personale tecnico specializzato nelle fasi di post-produzione e di messa *online* dei contenuti [Bon06]. Nel capitolo precedente è stato illustrato come tutti gli strumenti *e-learning* di Ateneo soffrano attualmente di questo problema. Il progetto alla base di questo lavoro di tesi è parte integrante di un'idea volta a consentire ai docenti di creare direttamente contenuti strutturati da inserire e gestire autonomamente all'interno delle piattaforme Moodle di Ateneo. L'*authoring tool* AContent, opportunamente modificato con l'introduzione di *template* per la creazione di materiale didattico può essere usato come editor dei contenuti esterno alle piattaforme già in uso nell'Ateneo. Il punto di partenza è l'individuazione delle tipologie di esercizi mancanti ma richieste dai docenti a cui segue l'implementazione di tali esercizi con particolare attenzione alla modalità di esportazione dei test conformi allo standard IMS QTI 1.2.1.

In questo capitolo viene introdotto il progetto OERTest, punto di partenza del progetto discusso che ha visto la modifica di AContent per la creazione di contenuti strutturati e la nascita di AContentBEAT. Vengono poi analizzate le tipologie di esercizi presenti sulla piattaforma, mostrandone i vantaggi e svantaggi dal punto di vista delle interfacce, dell'usabilità, della procedura

di creazione e di modifica e del meccanismo di importazione ed esportazione. In conclusione vengono descritti gli esercizi aggiunti in integrazione a quelli già presenti, i cui dettagli implementativi sono discussi nel prossimo capitolo.

4.1 AContent BEAT

La piattaforma AContent BEAT (Bologna E-learning Authoring Tool) è frutto di parte del lavoro svolto per il progetto OERTest [OER12], un progetto europeo iniziato nell'ottobre del 2010 e conclusosi a settembre del 2012. OERTest sostiene l'integrazione in ambito didattico delle risorse educative di libera condivisione (*Open Educational Resource*) per la valutazione dell'apprendimento e prevede la ricerca di standard volti a regolamentare l'offerta di tali risorse e i servizi di valutazione. Ad esempio, nell'ambito di OERTest sono state stilate delle linee guida, criteri e scale per la valutazione e strumenti di controllo della qualità e del carico del lavoro. Il gruppo di lavoro composto da pedagogisti e informatici, supervisionato dal Settore *e-learning* di Ateneo ha modificato la piattaforma AContent per offrire un semplice e valido strumento che fornisce metodologie e suggerimenti per la progettazione e produzione di materiale didattico compatibile con gli standard più conosciuti. L'idea di partenza è stata quella di implementare un'estensione che permette di introdurre dei *template* per personalizzare graficamente e strutturalmente il contenuto di un corso che si vuole creare. Basandosi su percorsi didattici e modelli pedagogici esistenti, il gruppo di lavoro di Scienze della Formazione ha individuato e proposto una serie di strutture di base da proporre all'autore in fase di creazione del corso. L'introduzione dei *template* vuole dare coerenza ai documenti creati, ovvero rendere uniforme la stesura dei contenuti tramite un insieme di strumenti ed indicazioni generali. Vuole fornire, inoltre, un sistema di controllo dell'organizzazione della lezione (obiettivi, requisiti ecc..) facile e flessibile ma in grado di modellare i LO adeguandoli alle diverse esigenze di insegnamento. Successivamente si è pensato di introdurre anche una componente di personalizzazione del *layout* del contenuto

e sono stati introdotti dei temi che comprendono la possibilità di scegliere diverse combinazioni per collocare testo e immagini con un eventuale *banner* di presentazione.

4.2 Analisi delle tipologie di esercizi esistenti

La piattaforma AContent, introdotta nella sezione 3.3.2, consente di gestire il profilo dell'utente (sezione *Profile*), accedere ai contenuti caricati sul file system (sezione *File Manager*), accedere ai corsi (sezione *Home*) e gestire gli esercizi e le prove (sezione *Manage Test and Surveys*). Quest'ultima è separata dalla sezione relativa alla gestione delle attività inseribili all'interno di un corso ed è il punto di partenza da cui ci si muove per la creazione di nuovi esercizi e l'analisi di quelli già esistenti.

La sezione *Manage Test and Surveys* è suddivisa in tre *tab* che permettono di creare test, singoli esercizi, o categorie di esercizi, come mostrato in figura 4.1. All'utente vengono inizialmente mostrati i test già creati, nel caso in cui si voglia cancellare o modificare un test già esistente. In alternativa, è possibile caricare un nuovo pacchetto di test, che ha come requisito primario la conformità alle specifiche IMS QTI 1.2.1.



Figura 4.1: Opzioni sezione *Manage and Surveys*

Nel *tab* di creazione del test, *Create Test/Survey*, viene richiesto di assegnare al test un nome univoco, che può essere accompagnato da una descrizione opzionale. Il procedimento per l'inserimento di un esercizio all'interno

di un corso prevede obbligatoriamente la creazione di un test al quale associare le singole domande. Viceversa, anche se creata all'interno di un corso e quindi recuperabile da *database* tramite l'identificativo (ID) del corso, una singola domanda non può essere inserita all'interno del contenuto se non viene prima inserita all'interno di un test. Le singole domande create per un corso non sono fruibili all'interno di altri corsi se non attraverso un'importazione. Il test creato risulta "figlio" della pagina nel quale viene inserito ed è visualizzabile alla fine del contenuto della pagina. Nel momento in cui un esercizio è associato ad un test, l'utente viene reindirizzato ad una maschera che assegna i punteggi per ogni singola domanda aggiunta.

Per la creazione di un singolo esercizio occorre spostarsi sul *tab Question Bank*, dove viene mostrato un *form* (si veda figura 4.2) con elencate le tipologie di domande disponibili. Due sezioni dedicate consentono l'importazione delle singole domande e la ricerca delle domande per categoria, rispettivamente.

Nella parte inferiore di questo *tab* vengono mostrati tutti gli esercizi già creati e le opzioni di esportazione in IMS QTI 1.2.1 e 2.1, le opzioni di *preview* per visualizzare la presentazione della domanda e l'opzione per la cancellazione dell'esercizio. Bisogna tenere in considerazione il fatto che in AContent gli esercizi vengono creati, modificati e importati ma, anche se contenuti all'interno del corso, non vengono svolti o salvati i risultati degli utenti. Questo perchè AContent, essendo uno strumento per l'*editing*, non presenta le funzionalità di esecuzione degli *assessment* e la componente di *result reporting* presente in ATutor.

Le opzioni per la creazione delle categoria alla quale associare gli esercizi ed il raggruppamento delle domande per obiettivi formativi sono contenute in un *tab* dedicato, *Question Catagories*.

La principale limitazione strutturale è legata alla memorizzazione dei dati relativi agli esercizi su database. Gli esercizi creati vengono memorizzati nella tabella *test_question* e ad ogni domanda viene associato:

- l'ID del corso a quale appartiene;

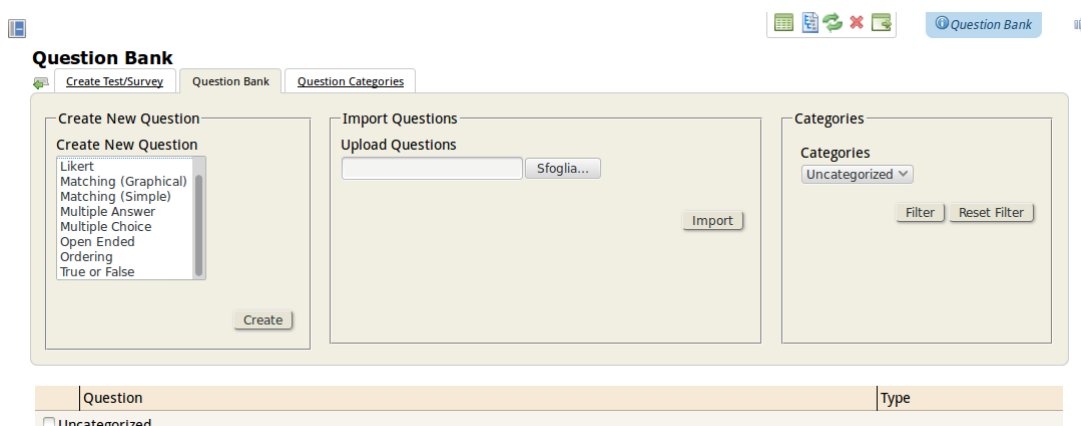


Figura 4.2: Creazione di un singolo esercizio

- l'eventuale *feedback*;
- il corpo dell'esercizio chiamato *question*;
- dieci campi denominati *choice*;
- dieci campi denominati *answer*;
- dieci campi denominati *option*.

Questi campi vengono utilizzati per inserire le opzioni degli esercizi, incluse le risposte, qualora la tipologia di esercizio le preveda. Il sistema offre al massimo dieci opzioni di scelta a domanda, come viene descritto nell'analisi successiva delle singole tipologie di esercizio. Questa limitazione è dovuta alla costruzione della tabella nel database.

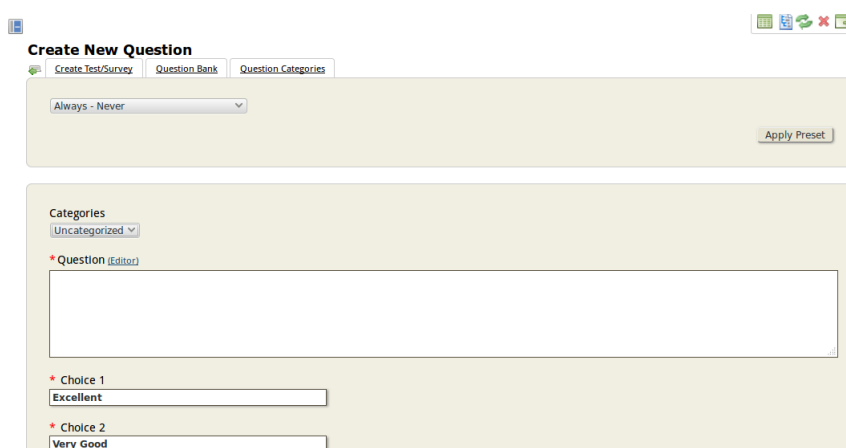
Per non creare disallineamento con gli esercizi già implementati e con le funzioni principali di *import* ed *export*, questo sistema non è stato modificato durante l'implementazione dei nuovi esercizi, ma sono state adottate delle tecniche diverse per ogni esercizio per la memorizzazione dei dati su database.

Di seguito, vengono analizzate nel dettaglio le tipologie di esercizi standard, discutendone i principali difetti e delineandone i possibili margini di miglioramento emersi in fase di creazione e di presentazione, prima di descrivere la progettazione delle nuove tipologie.

4.2.1 Likert

La maschera di creazione di un esercizio *likert*, mostrata in figura 4.3, si presenta con l'opzione di scelta dei *preset*, ovvero di una scala di valori preimpostati che vanno da un valore minimo ad un massimo che dipende dal contesto. In particolare l'autore può scegliere tra i seguenti *preset*:

- “*Always - Never*” al quale sono associate le seguenti sei *choice*: *Always*, *Very Frequently*, *Occasionally*, *Rarely*, *Very Rarely*, *Never*. Questa è l'opzione di default;
- “*Excellent - Very Poor*” a cui sono associati i seguenti sei *choice*: *Excellent*, *Very Good*, *Good*, *Fair*, *Poor*, *Very Poor*;
- “*Strongly Agree - Strongly Disagree*” al quale sono associate le seguenti cinque *choice*: *Strongly Agree*, *Agree*, *Undecided*, *Disagree*, *Strongly Disagree*;
- “*Very Important - Unimportant*” a cui sono associati i seguenti cinque *choice*: *Very Important*, *Important*, *Moderately Important*, *Of Little Importance*.



The screenshot shows a web interface titled "Create New Question". At the top, there are navigation tabs: "Create Test/Survey", "Question Bank", and "Question Categories". Below the tabs, there is a dropdown menu currently set to "Always - Never" and an "Apply Preset" button. Underneath, there is a "Categories" section with a dropdown menu set to "Uncategorized". The main area is labeled "* Question (Editor)" and contains a large text input field. Below the text field, there are two "Choice" input fields: "Choice 1" with the value "Excellent" and "Choice 2" with the value "Very Good".

Figura 4.3: Maschera di creazione di un esercizio Likert

Si possono creare *preset* personalizzati che possono contenere scale di valori fino a dieci elementi. Alla domanda è possibile associare una categoria esistente o non categorizzarla. È previsto un *feedback* sulla domanda.

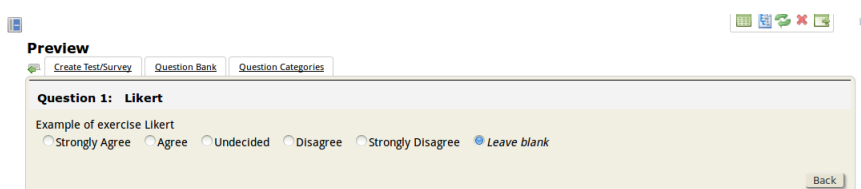


Figura 4.4: Anteprima di un esercizio Likert

Nella maschera che viene presentata all'utente per svolgere l'esercizio, riportata in figura 4.4, la domanda viene mostrata con lo stesso layout della pagina di creazione degli esercizi. Anche con l'inserimento di immagini come corpo della domanda la presentazione rimane ottimale. Le difficoltà in questa tipologia di esercizio sono quelle relative al non poter assegnare un singolo peso a ogni risposta, difatti è previsto solo un punteggio totale della domanda. Anche la mancanza di un feedback per ogni risposta può essere segnalata come una mancanza dell'esercizio. Questa proprietà non è prevista in nessun esercizio della piattaforma.

4.2.2 Open Ended

La maschera di creazione delle domande *Open Ended* (si veda figura 4.5) presenta un'area di testo per inserire il corpo della domanda ed eventuali *feedback*. Viene data all'utente la possibilità di scegliere il numero di righe di testo previste in risposta all'esercizio creato.

La maschera di presentazione all'utente è minimale: l'unica azione dell'utente è quella di inserire testo all'interno dell'area predisposta, senza alcuna forma di interazione aggiuntiva. Il layout di questa pagina, mostrato in figura 4.6, può non coincidere con il layout del corso in quanto il tema di presentazione è lo stesso di quello che viene visualizzato in fase di creazione.



Create New Question

Create Test/Survey | Question Bank | Question Categories

Categories: Uncategorized

Optional Feedback [\(Editor\)](#)

* Question [\(Editor\)](#)

Answer Size

- One Word
- One Sentence
- Short Paragraph
- One Page

Save Cancel

Figura 4.5: Creazione di una prova open ended



Preview

Create Test/Survey | Question Bank | Question Categories

Question 1: Open Ended

Exampel of Open Ended

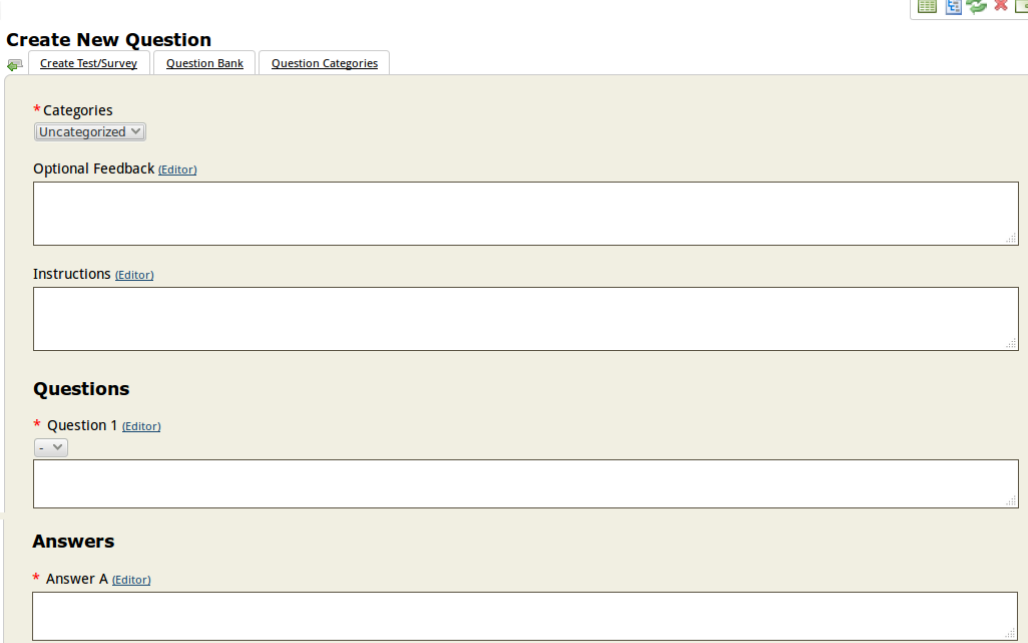
Back

Figura 4.6: Anteprima tipologia Open Ended

Questa modalità di esercizio non presenta un risultato di validità che può essere verificato con un sistema automatico pertanto il problema dei pesi non si pone. Analoga situazione per i *feedback*. In fase di memorizzazione, la differenza principale rispetto agli altri esercizi è data dall'utilizzo dei campi del database per memorizzare le singole righe di testo inserite, che in fase di importazione può rendere macchinosa la ricostruzione della risposta per intero.

4.2.3 Matching graphical e simple

Le maschere di creazione degli esercizi *matching graphical* o *simple* (è la medesima per entrambi gli esercizi), presentata in figura 4.11, fornisce fino a dieci elementi *question* da collegare a dieci elementi *answer*. Gli elementi *question* vengono numerati con numeri da uno a dieci mentre gli elementi *answer* sono etichettati con lettere dalla A alla J. La modalità di associazione è implementata per mezzo di un menu a tendina che può associare a ogni *question* un'*answer*. In questa tipologia di esercizio viene inserita un'area di testo per fornire eventuali istruzioni, oltre alla classica area di testo per il *feedback* opzionale.



The screenshot shows the 'Create New Question' interface. At the top, there are three tabs: 'Create Test/Survey', 'Question Bank', and 'Question Categories'. Below the tabs, there is a 'Categories' section with a dropdown menu set to 'Uncategorized'. The 'Optional Feedback' section has a text area with an 'Editor' link. The 'Instructions' section has a text area with an 'Editor' link. The 'Questions' section has a dropdown menu set to '1' and a text area with an 'Editor' link. The 'Answers' section has a text area with an 'Editor' link. There are also some utility icons in the top right corner.

Figura 4.7: Creazione di un singolo esercizio

Le maschere di *preview* delle due prove visibili in figura 4.12 si presentano in maniera pressoché identica, la differenza data solo dalla rappresentazio-

ne grafica della corrispondenza tra le risposte giuste. L'utente seleziona la risposta secondo lui corretta attraverso la selezione del menu a tendina.

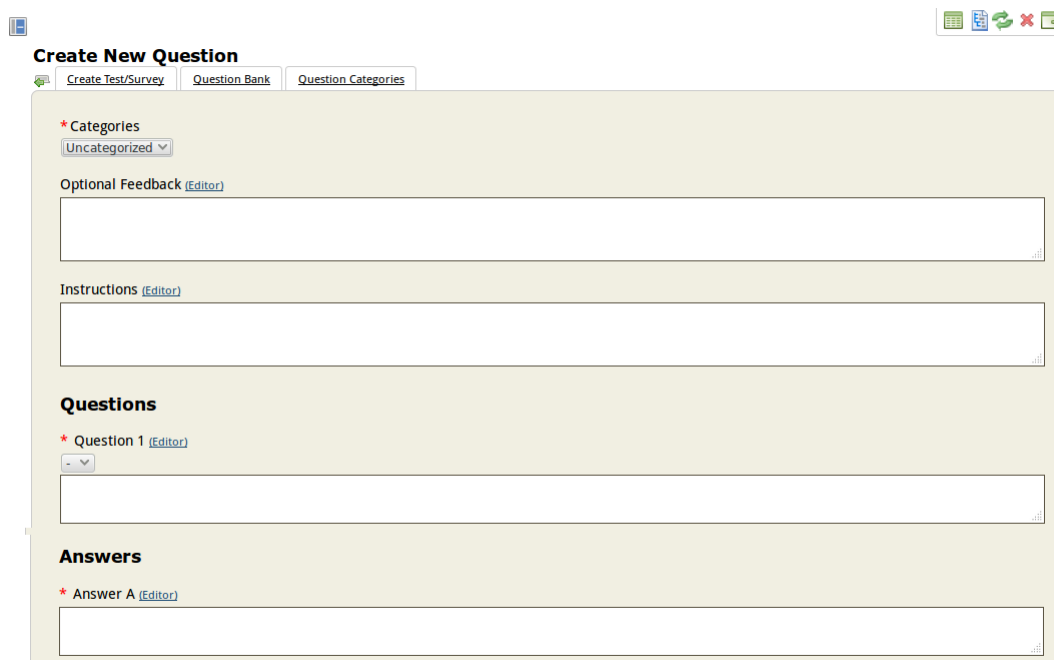


Figura 4.8: Creazione di un esercizio matching

Questa tipologia di esercizio, molto utilizzata per lo studio delle lingue per la possibilità di fare associazioni anche tra immagini, si presenta poco intuitiva dal punto di vista della creazione e soprattutto non offre una resa grafica ottimale che coinvolga l'utente. In particolare si è notato che, in fase di svolgimento, il collegamento tra gli elementi *question* e *answer* risulta difficoltoso.

4.2.4 Multiple choice e multiple answer

Le maschere di creazione della tipologia di *multiple choice* e *multiple answer* coincidono e sono presentate in figura 4.9. Le maschere presentano una modalità di inserimento simile a quella creata per la tipologia *matching*, in

quanto vengono visualizzate due aree di testo, una per le domande e una per i *feedback*. Attraverso una *checkbox* si possono selezionare le risposte esatte, che variano a seconda del tipo della domanda. La modalità di inserimento risulta intuitiva, il che fa sì che questa tipologia di esercizi sia una delle più utilizzate.

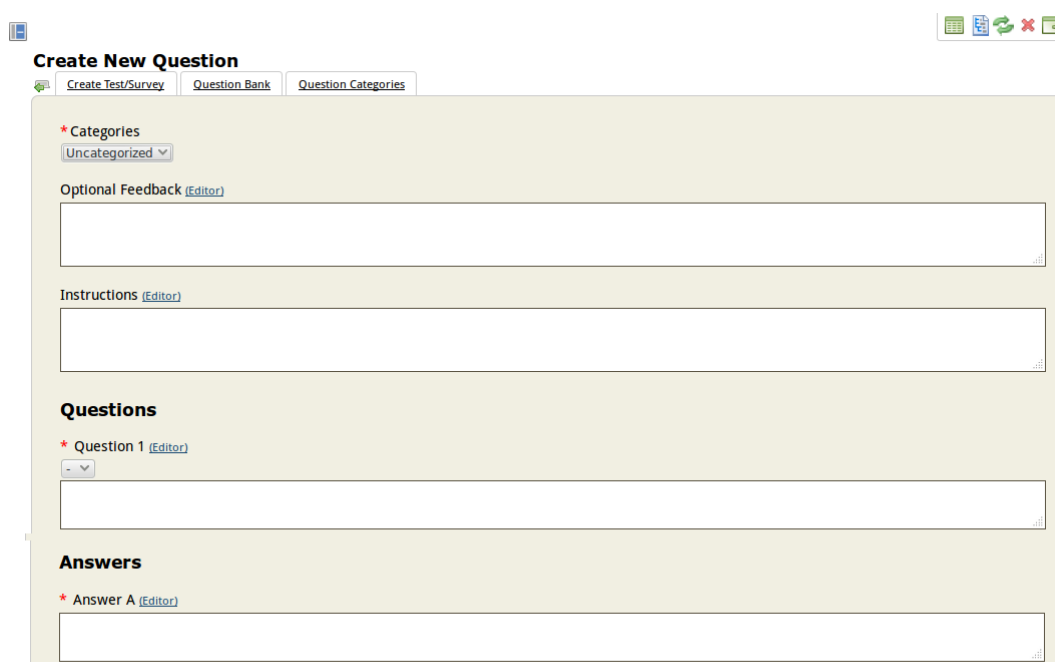


Figura 4.9: Creazione di un singolo esercizio

La maschera di preview riportata in figura 4.10 mostra come viene presentato all'utente l'esercizio: vista la facilità di presentazione non ci sono stati spunti per applicare modifiche.

4.2.5 Ordering

La maschera di creazione dell'esercizio *ordering* si presenta con una serie di aree di testo, denominate *item*, nelle quali inserire le varie parti della domanda.

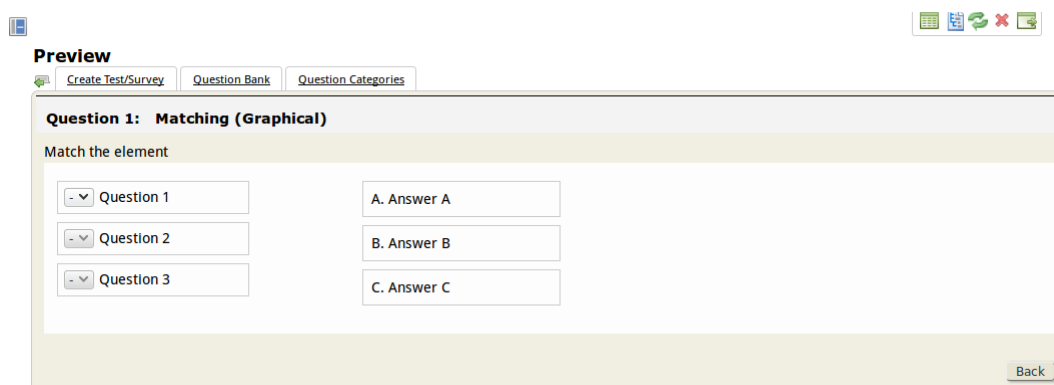


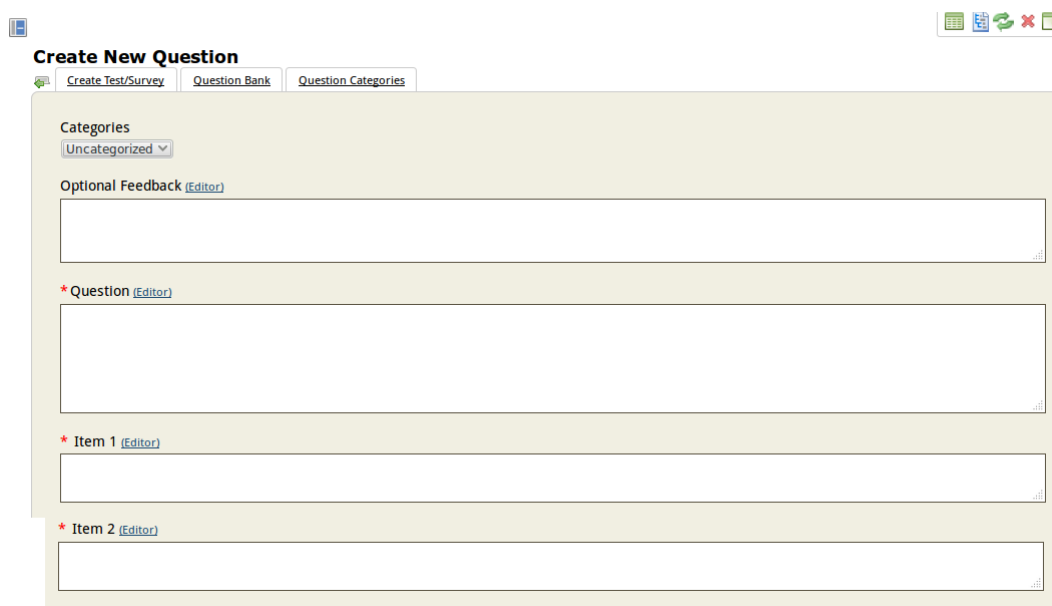
Figura 4.10: Anteprima di un esercizio matching simple

In fase di *preview* viene mostrato il testo della domanda che in questo caso potrebbe solo contenere la consegna dell'esercizio. Vengono poi mostrate in ordine casuale le possibili parole per la composizione selezionabili con un menu a tendina. Per quanto riguarda l'*ordering* soprattutto in fase di creazione della domanda si può pensare a un modo più intuitivo di creazione della tipologia di esercizi. In fase di visualizzazione si può curare di più l'aspetto presentazionale inoltre nel caso di *ordering* di più parole la visualizzazione risulta poco intuitiva. Per quanto riguarda l'*ordering*, si è notato che si potrebbe pensare ad un modo più intuitivo di creazione della tipologia di esercizi. In fase di visualizzazione l'aspetto presentazionale andrebbe curato maggiormente, specialmente nel caso di *ordering* di più parole, dove la visualizzazione risulta poco intuitiva. Non è stato preso in analisi l'esercizio *Vero o Falso* in quanto non è stato reputato una tipologia di esercizio significativa da utilizzare in ambito universitario.

4.2.6 Riepilogo delle osservazioni emerse

Nel corso dell'analisi delle tipologie di esercizi standard sono emerse alcune osservazioni, che offrono lo spunto per apportarvi migliorie.

In tutte le tipologie di esercizio, come già accennato nell'analisi della tipologia *Likert*, non è possibile applicare pesi e *feedback* su singola risposta,



The screenshot shows a web interface titled "Create New Question". At the top, there are three tabs: "Create Test/Survey", "Question Bank", and "Question Categories". Below the tabs, there is a "Categories" dropdown menu set to "Uncategorized". Underneath, there is a section for "Optional Feedback" with an "Editor" link and a large text area. This is followed by a section for "* Question" with an "Editor" link and another large text area. Below that are two sections for "* Item 1" and "* Item 2", each with an "Editor" link and a large text area. The interface is clean and uses a light beige background.

Figura 4.11: Creazione esercizio ordering

il che sarebbe di grande utilità per le fasi di valutazione.

Il procedimento di inserimento di file multimediali nel corpo della risposta e della domanda risulta inoltre complesso e formato da parecchi passaggi che potrebbero essere evitati. Per inserire un contenuto multimediale, infatti, è necessario attivare l'opzione *editor*, visualizzabile su ogni area di testo. Questa apre una finestra *pop-up* che permette di caricare il file su *file system*. Quindi occorre selezionare la risorsa per aggiungere il file nell'editor di testo mostrato nella parte superiore della pagina, premere il tasto per inserire il riferimento al contenuto multimediale all'interno dell'area di testo, salvare e chiudere la finestra.

Alcune tipologie di esercizi, inoltre, potrebbero beneficiare di una nuova veste grafica, per consentire di avere delle interfacce più intuitive e di facile fruizione da parte dell'utente. Infine, come già accennato all'inizio della sezione, le modalità di memorizzazione dei dati relativi agli esercizi su database limitano le possibilità e la versatilità di alcune tipologie di esercizi, come l'*Ordering*.



Figura 4.12: Anteprima esercizio Ordering

4.3 Progettazione di nuovi esercizi

L'analisi condotta in collaborazione con il Settore E-learning ha consentito di individuare sia alcune tipologie di esercizi non implementati nell'attuale versione di AContent, sia alcune soluzioni per migliorare, ridefinendoli, i *form* degli editor degli esercizi e della presentazione a disposizione dei docenti/autori.

Al fine di verificare la reale efficacia del lavoro svolto in termini di usabilità si è ritenuto opportuno implementare anche quegli esercizi che necessitavano solo di una completa rivisitazione dell'interfaccia grafica per semplificare le attività di *editing*.

La decisione di aggiungere alcune tipologie di prove, quali cruciverba o riempimento spazi vuoti, ecc..., e di migliorare l'usabilità dell'interfaccia delle prove già presenti risponde a due diversi ordini di necessità:

- fornire ai docenti la possibilità di scegliere tra un numero maggiore di tipologie di prove in relazione alle funzioni che possono essere svolte dalla valutazione;
- accompagnare il docente nel processo di creazione della prova facilitandone la costruzione tramite maschere intuitive ed esemplificazioni.

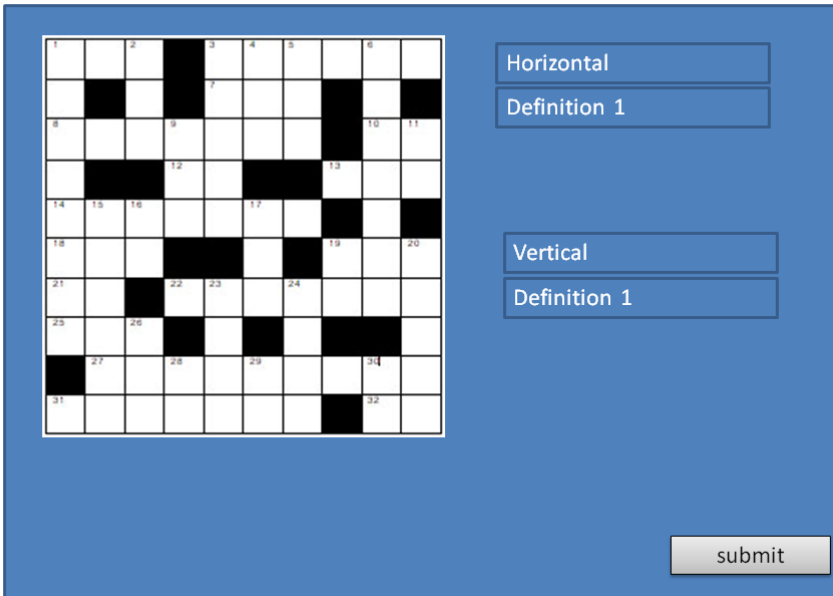
Di seguito viene spiegato come sono state progettate le varie tipologie di esercizi scelte cercando di tenere conto delle limitazioni di database di cui è

parlato in precedenza.

4.3.1 Crosswords

L'esercizio *crosswords*, presente nella piattaforma DALIA, è stato scelto in quanto considerato una valida aggiunta all'offerta di esercizi già presenti. La maschera di creazione progettata in figura 4.13 deve permettere di inserire all'interno di una griglia 10x10 un numero di parole limitato a un massimo di nove verticali e nove orizzontali.

Crosswords form design



1		2		3	4	5		6	
				7					
8			9					10	11
			12			13			
14	15	16			17				
18						19		20	
21			22	23		24			
25		26							
	27	28		29				30	
31									32

Horizontal

Definition 1

Vertical

Definition 1

submit

Figura 4.13: Progettazione antprima esercizio Crosswords

L'unico vincolo è il fatto di dover lasciare uno spazio vuoto alla fine di ogni parola. Una volta che sono state inserite tutte le parole, attraverso il tasto *submit* vengono create tante coppie di aree di testo (parola-definizione) quante sono le parole verticali e orizzontali costruite e vengono colorati di nero tutti i quadrati rimasti sprovvisti di lettere. Non appena le definizioni sono

state inserite, i dati vengono memorizzati nel database. La memorizzazione avviene inserendo le parole orizzontali e verticali concatenate all'interno della tabella *test_questions* nei campi del database *choice* mentre le definizioni orizzontali e verticali sono affidate ai campi *answer* e *option*. Per l'importazione e l'esportazione di questo tipo di esercizio non previsto nelle specifiche IMS QTI ci si avvale di una modifica alle funzioni della tipologia *matching simple*.

4.3.2 Text entry

Secondo la terminologia adottata dalle specifiche IMS QTI 2.1, con *text entry* si intende quella tipologia di esercizio in cui viene mostrato ai candidati del testo con aree di testo da riempire con un'unica parola esatta. Questa tipologia di esercizio è risultata la tipologia più comune mancante all'interno dell'editor, dove è possibile ottenere un esercizio simile utilizzando la tipologia *open ended*, con la limitazione però di avere un solo spazio vuoto. È stata adottata la nomenclatura dell'ultima versione dello standard in previsione della possibile implementazione di una nuova versione di *import* da parte della comunità di sviluppatori.

La maschera di creazione progettata in figura 4.14 si presenta con un'area di testo per la frase completa che si vuole inserire.

Successivamente, attraverso il tasto “trasforma” vengono creati tanti *checkbox* quante sono le parole da far “indovinare” all'utente. Questa modalità di inserimento risulta più facile di quella utilizzata per lo stesso tipo di esercizio su Moodle, che invece chiede di inserire la sintassi per la creazione degli spazi vuoti all'interno dell'editor.

L'inserimento all'interno del database avviene memorizzando, oltre alla frase completa nella domanda, anche l'indice della posizione delle parole mancanti nei campi *answer*, cosicché la limitazione di questo esercizio è la necessità di avere di un massimo di dieci spazi vuoti.



Figura 4.14: Progettazione preview Text entry

4.3.3 In-line choice

L'esercizio *in-line choice* è una versione del *text entry* che permette di inserire delle opzioni di scelta nelle aree vuote. Questa tipologia di esercizi viene molto utilizzata nella preparazione al superamento delle prove di idoneità delle lingue per lo studio della grammatica, perché, avendo a disposizione più opzioni possibili di risposta, il candidato si esercita a ragionare sulla risposta corretta che è sicuramente presente tra le varie opzioni. In previsione di un utilizzo ufficiale del *tool* su tutte le piattaforme di Ateneo si è deciso di implementare anche questa tipologia di esercizio.

La maschera di creazione in figura 4.15 si presenta simile a quella del *text entry*, con la differenza che in fase di *transform* vengono presentate 2 aree di testo a fianco della parola da trasformare per permettere di inserire le alternative alla risposta esatta. Il salvataggio della tupla su database nella tabella *tests_questions* avviene inserendo in *question* tutto il corpo della domanda, nei campi *answer* gli indici delle parole mancanti e in *option* il concatenamento delle due opzioni possibili. Anche in questa tipologia di

esercizio è impostato un vincolo massimo di dieci parole mancanti con relative proposte si soluzione. In fase di presentazione è mostrato all'utente un menù a tendina nello spazio vuoto con le tre soluzioni possibili in ordine casuale.

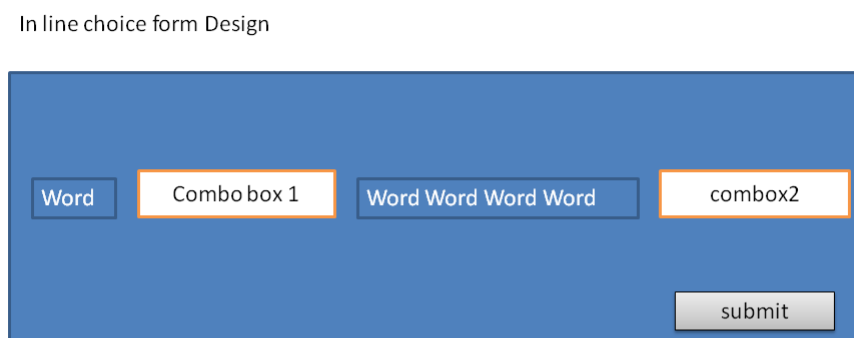


Figura 4.15: Progettazione preview In line choice

4.3.4 Ordering Words

Si è deciso di riprogettare questo esercizio già presente all'interno dell'editor per cercare di crearne un'alternativa che risulti più intuitiva sia in fase di creazione che in fase di presentazione. La maschera creazione progettata in figura 4.16 si presenta semplicemente con un'area di testo nel quale inserire una frase. La frase poi, come nel caso dell'esercizio *text entry*, viene scomposta in singole parole e memorizzata su database, ancora in modo analogo all'esercizio *text entry*. In fase di presentazione viene mostrata all'utente

un'area contenente tutte le parole della frase, con accanto tante *text area* quante sono le parole della frase.

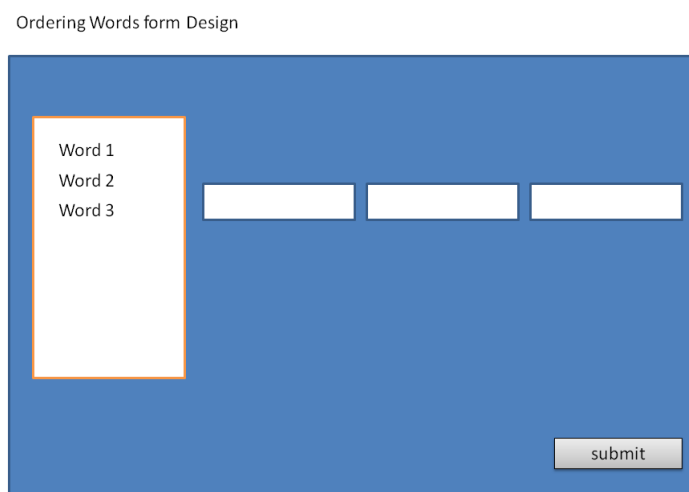
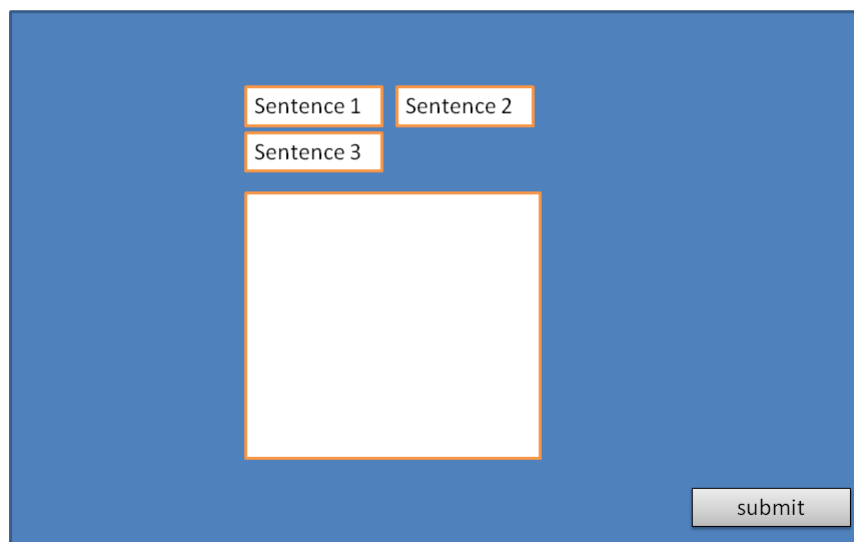


Figura 4.16: Progettazione anteprima Odering

4.3.5 Ordering Sentences

L'esercizio *ordering sentences* è un'estensione dell'esercizio precedente che prevede in più la creazione di una serie di aree di testo che possono contenere più parole limitate a dieci blocchi di testo. Questo esercizio è stato implementato per quelle situazioni in cui si vogliono utilizzare costruzioni di frasi complesse in ambito linguistico. Nella maschera di creazione pensata in figura 4.17 vengono presentati all'utente dieci aree di testo vuote nelle quali può inserire in ordine sparso le parole che formano la frase. Nel database vengono salvate le aree testo create dall'utente nei campi *choice*, pertanto la limitazione è anche in questo caso di dieci aree. In fase di presentazione vengono presentati all'utente le frasi in ordine casuale tramite un *select* e viene fornita un'area di testo per costruire correttamente la frase.

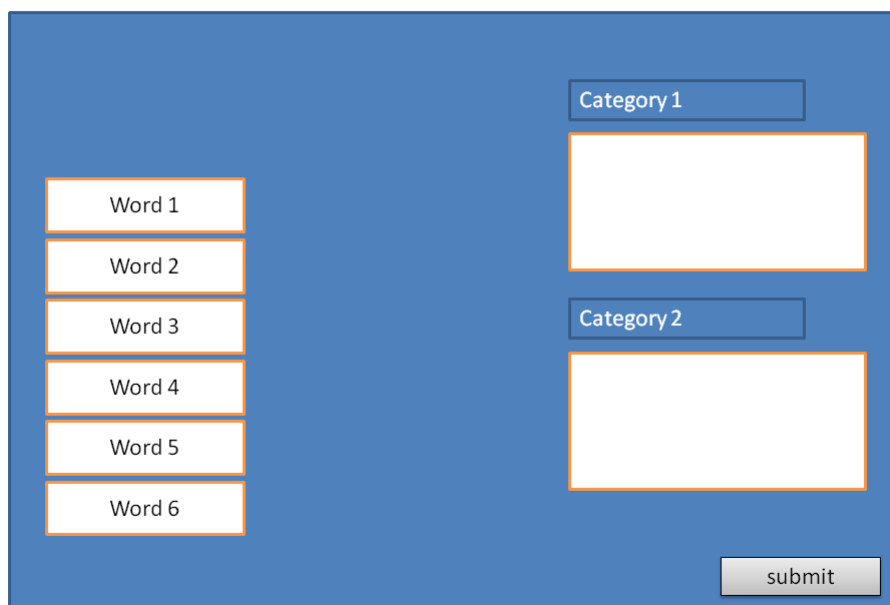
Ordering Sentences & drop form Design

**Figura 4.17:** Progettazione anteprima Ordering Sentences

4.3.6 Category Matching

Il *category matching* si presenta come un'estensione della tipologia di corrispondenza presente tra l'offerta di esercizi di AContent. In questo esercizio vengono create delle categorie al quale è possibile associare delle parole. La maschera di creazione di questo esercizio è rappresentata in figura 4.18, dove si può osservare una coppia di aree di testo. Nella prima è possibile inserire la categoria, mentre nell'area di testo sottostante è possibile inserire le parole corrispondenti alla categoria, separate da spazio. Anche in questo caso il salvataggio dei dati su database è vincolato a dieci categorie che verranno salvate nei campi *choice*. Il numero di parole per ogni categoria viene associato a un campo *option*, mentre l'insieme di parole concatenate con un carattere speciale vengono memorizzate nel campo *answer*. In fase di presentazione sono visualizzati i *select* nel quale inserire le categorie e le parole.

Ordering Sentences & drop form Design



The image shows a web interface for a category match exercise. On the left side, there is a vertical list of six white boxes with orange borders, labeled 'Word 1' through 'Word 6'. On the right side, there are two categories: 'Category 1' and 'Category 2', each with a corresponding empty white box with an orange border. A 'submit' button is located at the bottom right of the interface.

Figura 4.18: Progettazione interfaccia category match

4.3.7 Drag & Drop Matching

Il *drag & drop matching* è la versione della corrispondenza già presente in AContent creata attraverso l'utilizzo del *drag and drop*. Per rendere accessibile questa modalità occorre seguire le specifiche WAI-ARIA (*Web Accessibility Initiative - Accessible Rich Internet Applications*)[WAI13]. Le WAI ARIA comprendono documenti pubblicati dal W3C che specificano come rendere accessibili i contenuti dinamici che solitamente possono essere aggiunti alle interfacce utente attraverso AJAX, HTML, Javascript, ecc.. .

La maschera di creazione progettata in figura 4.19 è pensata in ottica semplificata rispetto a quella presentata da AContent. Vengono mostrate due aree di testo affiancate dove l'utente deve inserire la corrispondenza giusta tra elementi. Quando viene salvato l'esercizio vengono memorizzate nel database le associazioni attraverso i campi *choice* e *answer*. In fase di presentazione

viene data al candidato la possibilità di trascinare le parole per collocarle nel giusto insieme o categoria.

Matching drag & drop form Design

The image shows a matching drag-and-drop form design on a blue background. On the left side, there are three question boxes labeled "Question 2", "Question 3", and "Question 4". On the right side, there are four answer boxes labeled "Answer 1", "Answer 2", "Answer 3", and "Answer 4". A "submit" button is located at the bottom right of the form.

Figura 4.19: progettazione anteprima matching drag and drop

Capitolo 5

Implementazione

In questo capitolo vengono mostrate le scelte implementative adottate per le modalità di esercizi discussi. Vengono inizialmente presentate la struttura del codice di AContent e le applicazioni che vengono utilizzate per creare l'ambiente. Nell'ultima parte vengono quindi mostrate tutte le modifiche originali apportate durante lo svolgimento del progetto.

5.1 Struttura del codice AContent

Per creare una prova all'interno di AContent è necessario comprendere la struttura del codice e come vengono create le pagine all'interno dell'editor di contenuti. Il *core* dell'applicazione è scritto in PHP e ci si avvale di un RDBMS, MySQL in questo caso, per la memorizzazione delle informazioni su database. Essendo esso parte di ATutor, anche in AContent la logica applicativa è separata da quella presentazionale. Questo compito è affidato al *template engine* Savant2 [SAV13], che ha la particolarità di utilizzare come linguaggio di templating lo stesso PHP con il quale è scritto il *core* del *tool* di *editing*.

La versione di partenza del codice è stata prelevata dal *repository* del Progetto AContentBEAT presente su Github. La versione scaricata è la

versione 1.2 di AContent con incorporate le modifiche per la creazione di contenuto strutturato.

Come prima fase del progetto si è studiato il sistema di creazione degli esercizi all'interno dell' editor dei contenuti. La parte relativa alla memorizzazione dei dati su database è gestita dal modulo delle classi DAO che regolano lettura e scrittura da e su database. Se si vogliono inserire nuove tuple all'interno del database bisogna utilizzare le politiche di creazione delle query delle classi DAO. Se si vogliono creare servizi aggiuntivi utilizzando Savant2 occorre tenere in considerazione che l'implementazione, ovvero le funzioni che prevedono la computazione della pagina, devono essere separate dalla pagina di *template*, che si occupa solo della visualizzazione del contenuto. La struttura delle directory, infatti, oltre a prevedere un *kernel* contenuto all'interno della cartella *lib*, prevede la classe *themes* con la sottoclasse in cui vengono raccolti tutti i *template* di visualizzazione della pagina. Secondo questo principio, se si vuole aggiungere un tema, è sufficiente replicare le cartelle relative ai *template* associandovi CSS differenti e lasciando la computazione delle pagine immutata. Questa logica di separazione tra contenuto e presentazione tipica delle piattaforme LCMS ed LMS, in questo caso viene gestita tramite variabili globali che vengono passate dalla pagina al *template* di visualizzazione in *post* tramite URL. Infine, l'aggiunta di codice e la sua sottomissione alla Comunità per poterlo includere in versioni di AContent future prevede che si seguano le linee guida di implementazione fornite dalla Comunità.

5.2 Creazione di un esercizio su AContent

La procedura di creazione di un esercizio in AContent richiede la modifica di diverse funzioni di inizializzazione e la creazione di numerosi file, sia *template* di presentazione sia file che contengono la computazione degli esercizi. Nel seguito viene descritto l'iter di creazione della tipologia di esercizio *textentry*, progettata nel capitolo precedente, seguendo passo passo

tutte le operazioni per l'implementazione delle pagine di creazione, modifica, visualizzazione, importazione/esportazione, statistiche e risultati. Anche se queste ultime due non vengono utilizzate in AContent, sono state comunque implementare per un utilizzo sulla piattaforma ATutor per il processo di *result reporting*.

Come prima operazione per la creazione di un esercizio bisogna aggiungere la *query* di inserimento della tupla relativa all'esercizio nella tabella `tests_questions`. La *query* in questione, da inserire all'interno del file `test_question_queries.inc.php` contenuto nella cartella `include/lib`, è la seguente:

```
define('TR_SQL_QUESTION_TEXTENTRY',
"INSERT INTO ".TABLE_PREFIX."tests_questions VALUES
    ( NULL, %d, %d, 10, '%s', '%s',
    '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',
    %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
    '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', 0, 0)");
```

L'identificativo `TR_SQL_QUESTION_TEXTENTRY` è il nome per richiamare la *query* che esegue l'operazione di `INSERT` all'interno del database della tupla formata dai tipi di dato elencati dopo il prefisso della tabella. I campi della tabella sono i seguenti:

```
question_id, category_id, course_id, type, feedback, question, choice_0
choice_1, choice_2, choice_3, choice_4, choice_5, choice_6, choice_7,
choice_8, choice_9, answer_0, answer_1, answer_2, answer_3, answer_4,
answer_5, answer_6, answer_7, answer_8, answer_9, option_0, option_1,
option_2, option_3, option_4, option_5, option_6, option_7, option_8,
option_9, proprieties, content_id
```

La *query* è creata seguendo le regole stabilite dalle classi DAO. I campi `course_id`, `category_id`, `type` e `question_id` sono parametri che vengono utilizzati rispettivamente per le informazioni riguardanti il corso al quale l'esercizio è legato, la categoria di appartenenza dell'esercizio, la tipologia di prova in termini della posizione nell'array delle prove (in questo caso è il numero 10) e l'identificativo univoco della domanda all'interno della tabella nel database. I campi rimanenti invece possono essere utilizzati per contenere i

dati della domanda.

Una volta impostata la query per l'inserimento bisogna aggiungere nel file `page_constant_inc.php`, contenuto all'interno della directory `include` le seguenti righe di comando:

```
$_pages['tests/edit_question_textentry.php']['title_var'] = 'edit_question';
$_pages['tests/edit_question_textentry.php']['parent']    = 'tests/question_db.php';

$_pages['tests/create_question_textentry.php']['title_var'] = 'create_question';
$_pages['tests/create_question_textentry.php']['parent']    = 'tests/question_db.php';
```

che permettono al sistema di riconoscere le pagine create come pagine all'interno dell'*engine* e quindi renderle richiamabili da Savant2. Senza questi comandi il sistema non riesce a procedere e segnala errore nell'oggetto Savant incaricato di creare la pagina.

La parte più sostanziale di modifica del codice viene fatta nella file della classe `testQuestions.class.php` contenuto nella directory `include/classes`. Al suo interno bisogna aggiungere all'array `question_classes` la classe da creare per l'esercizio, che in questo caso è rappresentata dalla classe `TextEntryQuestion`. Quest'ultimo valore viene impiegato ogni qualvolta un esercizio di quella tipologia viene individuato come oggetto passato tramite variabili di `_POST`, per redirigere all'*overload* dei metodi della classe della prova individuata tramite la funzione `getQuestionType()`. All'interno della classe `TextEntryQuestion` le funzioni da implementare per un corretto funzionamento delle procedure di import/export, modifica, creazione ed esecuzione dell'esercizio sono:

- `assignQTIVariables()`: funzione che inizializza le variabili utilizzate in eventuali operazioni da effettuare al momento della visualizzazione, come ad esempio l'esecuzione di un ciclo per la visualizzazione di dati estratti da un array;
- `assignDisplayResultVariables()`: funzione che viene richiamata contestualmente al template che contiene i risultati dell'esercizio. All'interno di `AContent` il modulo che richiama questa pagine non è disponibile ma

è necessario creare la funzione affinché i controlli implementati creino la tipologia di prova;

- `assignDisplayVariables()`: funzione che viene richiamata dalla pagina di visualizzazione dei risultati, nella quale possono essere implementate operazioni sulla visualizzazione dei dati, come ad esempio la presentazione casuale;
- `assignDisplayStatisticsVariables()`: funzione utilizzata per visualizzare le statistiche della domanda richiamata nel template delle statistiche. Questa funzione però non può essere richiamata all'interno di AContent;
- `importQti()`: funzione che si occupa di memorizzare nel database le informazioni relative alla prova importata, una volta estrapolati i dati tramite *parsing* da file XML;
- `mark()`: ulteriore funzione utilizzata per evidenziare i risultati giusti in caso la prova lo preveda.

Con la creazione di queste funzioni si concludono le modifiche ai file già esistenti all'interno di AContent, pertanto di seguito vengono descritte l'implementazione della logica presentazionale applicata e le procedure di *import/export* della tipologia di esercizio *Text entry* creata.

5.2.1 Maschera di creazione/modifica

Per la maschera di creazione e modifica sono stati implementati i file `create_question_textentry.php` e `edit_question_textentry.php`, contenuti all'interno della cartella `tests`, per la logica applicativa, e il file `create_edit_question_textentry.tpl.php`, per la logica presentazionale.

Nel file `create_question_textentry.php` vengono inserite tutte le funzioni associate a bottoni presenti nel form di inserimento di un esercizio, mostrato in figura 5.1. Una volta che l'utente preme un pulsante di quelli

presenti nella maschera, viene memorizzata una variabile in `$_POST` che segnala che quell'evento si è verificato. Con un controllo sull'inizializzazione di quell'evento è possibile catturarlo.

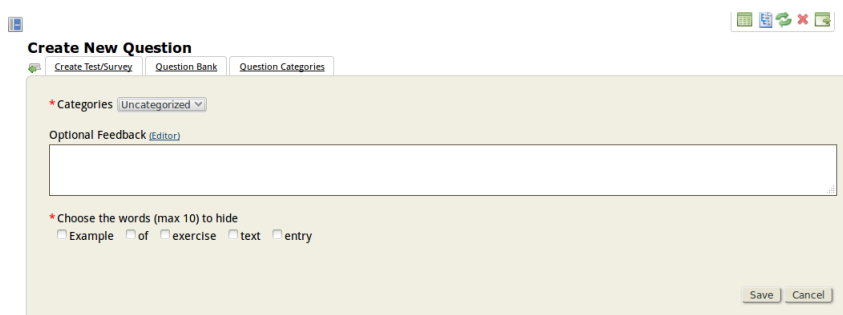


Figura 5.1: Creazione di un esercizio text entry

In questo esercizio, le operazioni possibili sono `transform`, `cancel` e `save`, il quale però è visibile solo in un secondo momento, all'attivazione del tasto `transform`. Nel caso venga premuto il tasto `cancel` si viene reindirizzati alla pagina di creazione degli esercizi. Tramite il tasto `transform` viene controllato che l'area di testo non sia vuota e compare un *checkbox* che permette di selezionare le parole della frase da nascondere. Nel caso venga selezionata l'opzione `save`, vengono recuperati tutti i dati dalla pagina creata per mezzo del file "tmpl" e vengono effettuate una serie di operazioni per salvare i dati all'interno di un array costruito in modo analogo alla tupla che la tabella `tests_questions` si aspetta. Una volta creato l'array secondo le specifiche progettate viene richiamata la *query* `TR_SQL_QUESTION_TEXTENTRY` che, attraverso le classi DAO, si occupa di effettuare la memorizzazione dei dati su database. Se l'operazione è andata a buon fine compare un messaggio di conferma e si viene reindirizzati alla pagina principale di scelta delle tipologie di esercizio. La pagina di modifica delle domande presenta modalità del tutto analoghe, con la particolarità che all'interno del testo è già compresa la domanda scritta in precedenza.

5.2.2 Maschera di preview

La maschera di *preview* di figura 5.2 mostra i risultati estrapolati dal database ed elaborati nel file `tmpl` che si occupa della presentazione dell'esercizio. Per rendere possibile la visualizzazione viene creato un ciclo e viene ricostruita la frase, controllando se l'indice dell'array coincide con uno di quelli compresi nel campo `answer`. Nel caso il gli indici confrontati siano uguali viene sostituita la parola da visualizzare con un'area di testo.

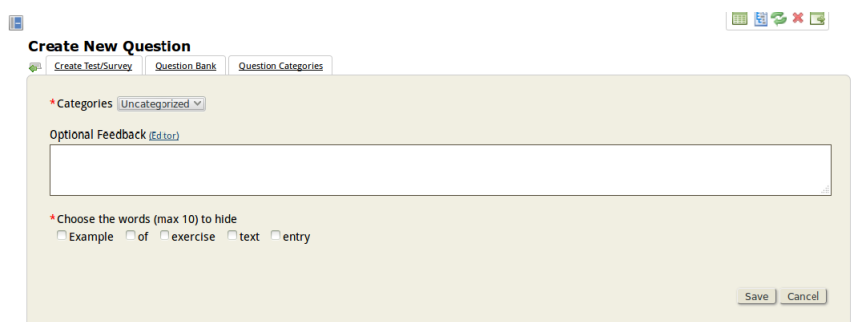


Figura 5.2: Anteprima dell'esercizio text entry

5.2.3 Import/export

Le funzioni di esportazione della piattaforma sono incluse all'interno della classe `testQuestions.class.php` anche se, in realtà, una volta individuata la modalità di esercizio da esportare, il *template engine* reindirige alla pagina che contiene la struttura XML prevista per immagazzinare le informazioni in maniera conforme allo standard. All'interno di AContent per ogni esercizio sono già presenti sia il file `tmpl` contenente XML in linea con le specifiche IMS QTI 1.2.1, che il file XML in linea con le specifiche IMS QTI 2.1. Dopo aver scelto la versione di standard di esportazione i dati contenuti nella tabella del database vengono passati dalla funzione `exoprTQTI()` che tramite oggetto `Savant2` richiama la pagina `tmpl` contenente la struttura XML. Analogamente ai file di presentazione, si opera attraverso le variabili globali passate tramite `_POST` per il popolamento delle strutture. Le differenze tra

le strutture delle versioni 1.2.1 e 2.1 di IMS QTI sono sostanziose proprio per la radicale riprogettazione. Per la modalità di esercizio *text entry* sono stati implementati entrambi i file denominati `textentry_qti_1p2.tpl.php` e `textentry_qti_1p2.tpl.php`. I file prodotti sono stati validati con il validatore ufficiale del IMS Global per la versione 1.2.1, in quanto validatore di pacchetti IMS CC. Per la versione IMS QTI sono state utilizzate le specifiche di XML *binding* [?] presenti sul sito e gli esempi proposti.

Una volta implementate le funzioni e per verificare la corretta esportazione si è provato a importare i file in formato IMS QTI 1.2.1 nella piattaforma. Il sistema non riconosce la tipologia di esercizio: analizzando le funzioni di *parsing* presenti nella classe `QTIParser.class` si è capito che non vengono presi in considerazione tutti gli esercizi che possono essere implementati attraverso lo standard. Per queste ragioni, una dettagliata comprensione delle funzioni `startElement()` e `endElement()` presenti all'interno della classe si è resa necessaria per trovare un approccio per la modifica della classe che non sconvolgesse il funzionamento classico per le tipologie esistenti. La difficoltà maggiore è stata quella di modificare il codice relativo agli attributi dell'elemento *response*, in quanto, come si può notare dalla porzione di codice sottostante,

```

case 'material':

//check who is mattext's ancestor, started from the most known inner layer
if (in_array('response_label', $this->element_path)){
    if(!in_array($this->mat_content, $this->choices)){
        //This is one of the choices.
        if (!empty($this->response_label[$this->item_num])){
            $this->choices[$this->item_num][array_pop($this->response_label[$this->item_num])] =
                $this->mat_content[$this->item_num];
        }
    }
} elseif (in_array('response_grp', $this->element_path) ||
    in_array('response_lid', $this->element_path)){
//for matching, where there are groups
//keep in mind that Respondus handles this by using response_lid
    $this->groups[$this->item_num][] =
        $this->reconstructRelativePath($this->mat_content[$this->item_num]);
} elseif (in_array('presentation', $this->element_path)&& $this->first ==1){
    $this->question[$this->item_num] =
        $this->reconstructRelativePath($this->mat_content[$this->item_num]);
    $this->first=0;
} elseif (in_array('itemfeedback', $this->element_path)){
    $this->feedback[$this->item_num] = $this->mat_content[$this->item_num];
}
//once material is closed, reset the mat_content variable.
$this->mat_content[$this->item_num] = '';
break;

```

vengono salvate diverse parti della struttura in base al *response* individuato. Anche se il sistema riconosce la tipologia di esercizio, il *parser* non riesce a prelevare le parti della struttura determinanti, pertanto è stato necessario modificare e popolare l'oggetto `QTIParserObject` con il codice per svolgere questa funzione.

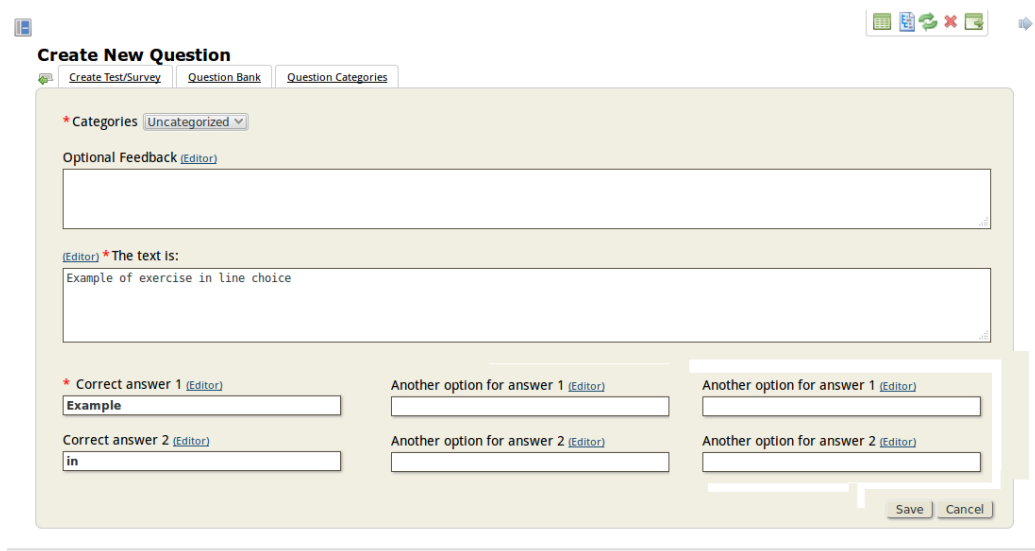
5.3 Altri esercizi implementati

5.3.1 In line choice

L'esercizio in *line choice* si mostra del tutto analogo all'esercizio *text entry*, con la particolarità che è stata creata un'altra maschera per aggiungere le parole da inserire tra le possibili scelte nel *combo box* invece di lasciare

un'area di testo. Il salvataggio delle opzioni nel database è stato gestito attraverso il salvataggio delle parole da visualizzare nel menu a tendina nei campi *option*. Poiché sono previste due opzioni per ogni *combo box*, viene usato uno speciale operatore di concatenazione. Nella figura 5.3 è mostrata la sezione che permette di aggiungere due opzioni per ogni parola selezionata.

Anche in fase di presentazione l'esercizio risulta analogo alla tipologia *textentry*. Questo esercizio non è presente nelle specifiche IMS QTI 1.2.1 pertanto non sarebbe necessario creare funzioni di *import/export* per questa tipologia, che tuttavia sono state comunque implementate allo scopo di effettuare delle prove e dei test sulla piattaforma.



The screenshot shows a web-based interface for creating a new question. At the top, there are navigation tabs: 'Create Test/Survey', 'Question Bank', and 'Question Categories'. The main form is titled 'Create New Question' and includes a dropdown menu for 'Categories' set to 'Uncategorized'. Below this is a text area for 'Optional Feedback'. A section labeled 'The text is:' contains a text editor with the example text 'Example of exercise in line choice'. The bottom section is for defining answers, with two columns. The first column has 'Correct answer 1' (with 'Example' entered) and 'Correct answer 2' (with 'in' entered). The second column has 'Another option for answer 1' and 'Another option for answer 2'. The third column has 'Another option for answer 1' and 'Another option for answer 2'. At the bottom right, there are 'Save' and 'Cancel' buttons.

Figura 5.3: Anteprima dell'esercizio inline choice.

5.3.2 Crosswords

Questa tipologia di prova ha richiesto l'utilizzo di javascript per la realizzazione del template di creazione. In particolare è stato necessario per salvarsi l'indice della griglia per il salvataggio della domanda e per riproporlo in fase di definizione. Per le operazioni di import ed export di questa tipologia di esercizio ci si è ricondotti alla modalità corrispondenza in quanto

questo esercizio non è previsto in nessuna specifica dello standard sia versione 2.1 che 1.2.2 di IMS QTI.

In conclusione, le maschere di visualizzazione degli esercizi *ordering words*, *ordering sentences*, *matching*, *drag & drop*, e la maschera di creazione degli esercizi di *category matching* sono state implementate.

Conclusioni

L'Ateneo di Bologna mette a disposizione dei docenti piattaforme Moodle per il supporto all'insegnamento e la condivisione di contenuti didattici, la cui gestione è affidata ad un Settore specializzato. Il Settore *e-learning* si occupa infatti di fornire aiuto e supporto sull'inserimento dei contenuti in piattaforma e si adopera per la formazione degli utenti. Negli ultimi anni, con il diffondersi dell'utilizzo della piattaforma, è emerso che il sistema di creazione dei contenuti su Moodle risultava macchinoso e non subito intuitibile per la gran parte dei docenti. Nel contesto del progetto OERtest sono state apportate le prime modifiche della piattaforma AContent per facilitare l'inserimento dei contenuti attraverso l'impiego di *template* strutturali e di *layout*. Questo ha dato vita alla piattaforma AContent BEAT. Al fine di fornire un prodotto completo anche dal punto di vista della creazione delle prove di verifica, si è quindi vista la necessità di ampliare l'offerta di esercizi tramite l'aggiunta di nuove tipologie. Lo sviluppo e l'implementazione di tali esercizi sono stato l'oggetto di questo lavoro di tesi. Durante una serie di incontri tenutisi con il Settore *e-learning* sono state individuate le tipologie di esercizi mancanti e diverse interfacce da semplificare, come descritto nel quarto capitolo. Nel quinto capitolo sono forniti i dettagli di implementazione degli esercizi sulla piattaforma.

Il prossimo passo del progetto consiste in una fase di test della piattaforma da parte dei docenti, durante la quale gli sarà richiesto di tentare l'utilizzo del pacchetto di esercizi implementato. Contemporaneamente si sta progettando un meccanismo di comunicazione tra l'editor di contenuti e

la piattaforma Moodle di Ateneo. Vale la pena sottolineare, concludendo, che le prospettive di ampliamento di questo progetto sono molteplici. Per il futuro si pensa di ridefinire le interfacce di inserimento dei test all'interno dei corsi e di implementare un sistema di *import* conforme allo standard IMS QTI 2.1, in previsione della pubblicazione della nuova versione dello standard IMS CC. La prima fase del progetto, già avviata, dovrebbe terminare nel dicembre del 2013, quando questi sistemi saranno messi ufficialmente a disposizione dei docenti che si avvalgono di materiale *e-learning* come supporto alla didattica.

Bibliografia

- [SoA51] World's Frist School of Air Opened,
1951 , <http://trove.nla.gov.au/ndp/del/article/45718205> - March 2013
- [Asf03] ASFOR: Il glossario di e-learning di ASFOR,
2003, <http://www.asfor.it/>, - January 2013
- [Kar07] Tony Karrer, “eLearning Defined”, 2007,
<http://elearningtech.blogspot.it/2007/10/elearning-defined.html> January 2013
- [Bon06] Bonaiuti Giovanni: *E-Learning 2.0 - Il futuro dell'apprendimento in rete, tra formale e informale*, Erikson, 2006
- [Dig09] DigitPA - Ente nazionale per la digitalizzazione della pubblica amministrazione, 2009,
<http://www.digitpa.gov.it> - December 2013
- [RSS13] *RSS Specification*,
<http://www.rssboard.org/rss-specification>, 2013
- [Lam09] Lamandini A. “ L’evoluzione dell’e-learning ed e-learning in evoluzione”, *Ricerche di Pedagogia e Didattica*, 4, 2009, Didattica e Nuove Tecnologie
- [HodWay02] Hodgins, Wayne, *The future of Learning Object*,2002

- [Wil02] David A. Wiley, II, *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*, 2002
- [Jac03] Jacobsen, P: "LMS vs. LCMS", 2003,
<http://www.elearnspace.org/Articles/contentmanagement.htm>,
January 2013
- [IEEE06] IEEE: Learning Object Metadata, 2002
- [Dow02] S.Dowes : *The Learning Object Economy*, 2002
- [LTSC05] LTSC, IEEE Learning Technology Standards Committee Learning Objects Definition, 2005,
<http://www.ieeeltsc.org> , January 2013
- [CNIP04] Vedemecum CNIPA, 2004
- [ARI05] Alliance of Remote Instructional Authoring and Distribution Networks for Europe, 2005,
<http://www.ariadne-eu.org> January 2013
- [Dub05] Dublin Core metadata Initiative, 2005,
<http://dublincore.org>, January 2013
- [IEEE02] IEEE: Learning Object Metadata, 2002
- [IMS05] IMS Global Learning Consortium, 2005,
<http://www.imsglobal.com> , February 2013
- [ADL05] *Advance Distributed Learning Initiative*, 2005,
<http://www.adlnet.org> ,February 2013
- [Dod97] *Department of Defense*, 1997
<http://www.defense.gov/> , January 20013

- [DalMar06] F. Dal Fiore, G. Martinotti: *e-learning*, McGraw-Hill Companies, Milano, IT, 2006.
- [IMSCC08] IMS GLC Common Cartridge Specification, 2009
<http://www.imsglobal.org/commoncartridge.htm>, February 2013
- [IMSCP09] IMS GLC Content Packaging, 2008,
<http://www.imsglobal.org/>, February 2013 [content/packaging/](http://www.imsglobal.org/content/packaging/) , 2013
- [IMSLT10] IMS Learning Tools Interoperability, 2010,
<http://www.imsglobal.org/toolsinteroperability2.cfm> . , February 2013
- [IMSLI08] IMS Learning Information Service, 2008,
<http://www.imsglobal.org/lis/> , February 2013
- [IMSQT02] IMS Question Tools Interoperability, 2006,
<http://www.imsglobal.org/question/> , February 2013
- [QTIO02] IMS Question & Test Interoperability
An Overview Final Specification Version 1.2, 2006,
http://www.imsglobal.org/question/ktiv1p2/imsqti_oviewv1p2.html, February 2013
- [QTIA03] IMS Question and Test Interoperability
Addendum Version 1.2.1 Final Specification , 2003,
http://www.imsglobal.org/question/ktiv1p2p1/imsqti_addv1p2p1.html ,February 2013
- [QTI205] IMS Question & Test Interoperability
An Overview Final Specification Version 2.0, 2005,
http://www.imsglobal.org/question/qti_v2p0/imsqti_oviewv2p0.html, February 2013
- [QTI2112] IMS Question & Test Interoperability
An Overview Final Specification Version 2.1, 2012,

http://www.imsglobal.org/question/qtiv2p1/imsqti_oviewv2p1.html, February 2013

[QTIASI03] IMS Question & Test Interoperability:
ASI Information Model Specification
Final Specification Version 1.2, 2003,
http://www.imsglobal.org/question/qtiv1p2/imsqti_asi_infov1p2.html ,February 2013

[QTIRRM03] IMS Question & Test Interoperability:
Results Reporting Information Model Final
Specification Version 1.2, 2003,
http://www.imsglobal.org/question/qtiv1p2/imsqti_res_infov1p2.html ,February 2013

[QTIXML03] IMS Question & Test Interoperability: ASI
XML Binding Specification Final Specification Version 1.2, 2003,
http://www.imsglobal.org/question/qtiv1p2/imsqti_asi_bindv1p2.html, February 2013

[MOO13] Moodle, 1999, <https://moodle.org/> - March 2013

[ATu13] ATutor, <http://atutor.ca/> , March 2013

[ACo13] AContent, <http://atutor.ca/acontent> , March 2013

[Alf13] Alfacert Università di Bologna , 1998,
www.alfacert.unibo.it/ March 2013

[DAL13] DALIA Università di Bologna, 2009
<http://dalia.cilta.unibo.it/> March 2013

[SHI13] Shibboleth, <http://shibboleth.net/> March 2013

- [Blb13] Black Board, <http://www.blackboard.com/International/EMEA/Overview.aspx?lang=enus> March 2013
- [HOT13] Hot Potatoes, <http://hotpot.uvic.ca/> March 2013
- [LEA13] <http://www.learn-wise.com/> March 2013
- [ILIA13] ILIAS, 1997, www.ilias.de/ - March 2013
- [Bri12] Orsola BRIZIO *Da ALTAIR a DALIA: le fasi di una progettazione*, n. 1 2012
Le français dans le contexte plurilingue des Centres Linguistiques Universitaires italiens, 2, July 2012, http://www.dorif.it/ezone/ezone_articles.php?id=19 March 2013
- [CIL13] CILTA Guida allo studio in autonomia, 2009, http://dalia.cilta.unibo.it/file.php/1/Documenti_per_gli_studenti/Guida_allo_Studio_in_Autonomia.pdf March 2013
- [Nan13] Nanogong, <http://gong.ust.hk/nanogong/> March 2013
- [W3C13] W3C, <http://www.w3.org/> March 2013
- [Tin13] TinyMCE, March 2013
- [OER12] Progetto Europeo OERtest, 2012, http://www.eng.scedu.unibo.it/ScienzeEducazioneEn/Research/ProjectsAndActivities/Project_OER_Test. March 2013
- [WAI13] WAIARIA, <http://www.w3.org/WAI/intro/aria>, March 2013

[SAV13] Savant PHP, <http://phpsavant.com/> March 2013

[XMLQTI12] IMS Question & Test Interoperability
(QTI) XSD Binding, [http://www.imsglobal.org/
Squestion/ktiv2p1/imsqti_bindv2p1.html](http://www.imsglobal.org/Question/ktiv2p1/imsqti_bindv2p1.html) March 2013