

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Scienze e Tecnologie Informatiche

Progettazione e sviluppo di estensioni per AContent

Tesi di Laurea in SISTEMI MULTIMEDIALI

Relatore:
Chiar.ma Prof.ssa
PAOLA SALOMONI

Presentata da:
MAURO DONADIO

Correlatore:
Chiar.ma Dott.ssa
CATIA PRANDI

II Sessione
A.A. 2011/2012

*Ai miei genitori LUIGI e CATIA
che con il loro amore,
hanno reso tutto questo possibile...*

*A mio fratello ALESSIO,
i cui consigli e il cui appoggio
sono stati indispensabili...*

*A CHIARA,
per avermi insegnato
ad essere felice...*

Introduzione

Dopo anni in cui le tecnologie disponibili crescevano rapidamente ma con una lenta diffusione nell'ambito della formazione a distanza, ad oggi l'e-learning è diventata una modalità largamente utilizzata.

Con l'avvento del personal computer, si è passati ad un tipo di formazione a distanza svolta sia offline, tramite strumenti che non si avvalgono necessariamente della rete (floppy disk, videodischi, CD-ROM), che on-line, caratterizzata dall'avvento delle reti. Nasce così l'e-learning, una tipologia di formazione che si fonda sulla telematica e che fa uso di tutte le potenzialità che le nuove tecnologie sono in grado di offrire. Il concetto di formazione a distanza diviene, così, e-learning (inteso come *electronic based learning*) in cui la didattica viene facilitata da risorse e servizi disponibili e fruibili virtualmente per via telematica. Nella definizione di e-learning è esplicitato come i sistemi di informazione e comunicazione non debbano necessariamente essere collegati a Internet. Questo lascia spazio ad architetture come reti private oppure a sistemi completamente offline che, aggiunto al fatto che esistono diversi sistemi e piattaforme di e-learning, conduce al problema della compatibilità dei contenuti didattici. Problema che viene risollevato quando si parla di interoperabilità, ovvero l'abilità di due o più sistemi o componenti di scambiare informazioni e ad utilizzare le informazioni appena scambiate. In questa cornice di riferimento viene introdotto il concetto di Learning Content Management System (LCMS), ovvero un ambiente multi-utente in cui si possono creare, memorizzare, riusare, gestire e inviare contenuti didattici

di un repository centrale. Alla definizione data segue quella di Authoring Tool, un editor di contenuti software o un servizio che consente all'utente di produrre pagine e contenuti web.

Viene affrontato, infine, il concetto di LTI come strumento per l'interoperabilità fra diversi sistemi di e-learning. In particolare, IMS LTI sta per Learning Tools Interoperability e si propone come un Web Service che si interfaccia ai contenuti che possono essere ospitati sullo stesso computer, su un server, in una rete cloud. Il suo utilizzo di base è quello di creare una connessione sicura tramite il protocollo OAuth fra più sistemi senza necessità di sviluppare un software ad hoc per ogni integrazione. LTI vuole facilitare l'interazione di applicazioni ricche di contenuti didattici con piattaforme di LMS, portali o altri ambienti educativi.

In questo documento di tesi viene descritta la progettazione e la realizzazione di estensioni per il sistema di authoring AContent. L'idea è di creare un'estensione dell'authoring tool che implementi il concetto di template ovvero strumenti di grande efficacia e di facile utilizzo nelle fasi di redazione dei contenuti. Si prevede di aggiungerli ad AContent senza la necessità di integrare un intero motore di template ma utilizzando strutture dati esistenti e specifiche standard di e-learning. I servizi aggiuntivi da offrire agli autori sono stati organizzati secondo tre approcci diversi da cui sono emersi tre livelli di template. Il Template di Layout che determina l'aspetto grafico dei contenuti, il Template di Pagina che definisce la struttura di ogni singola pagina e il Template di Struttura che propone e imposta un modello per la struttura dell'intero contenuto didattico.

Il documento è costituito da una seconda parte di progetto che va a coinvolgere il sistema ATutor e pone grande attenzione sulle caratteristiche di interoperabilità fra l'authoring AContent e il LCMS ATutor. Lo scopo è quello di estendere le funzionalità di integrazione dei contenuti del sistema così da presentare materiale didattico esterno archiviato in AContent. Viene trattata l'integrazione di LTI all'interno dei due sistemi considerati assegnando i ruoli di Tool Provider (AContent), fornitore di contenuti didattici remoti e Tool

Consumer (ATutor), richiedente di tali contenuti. Sono considerati, infine, i due moduli di ATutor *AContent Repository* e *External Tool* che si occupano di importare materiale didattico da AContent tramite il Web Service REST. Si prevede la loro modifica affinché, attraverso il canale di comunicazione LTI stabilito, siano in grado di creare dei *Live Content Link* ovvero riferimenti a contenuti remoti (esterni alla piattaforma utilizzata) aggiornati in tempo reale. Infatti, a differenza di una normale importazione di un LO esterno è previsto che venga creato un “riferimento”. In questo modo, la modifica di una pagina sul Tool Provider AContent si ripercuoterà istantaneamente su tutti i contenuti dei Tool Consumer che hanno instaurato un Live Content Link con il provider.

Nel primo capitolo viene fatta una panoramica sull’e-learning, sui vari standard a disposizione e sulle piattaforme software per la redazione e la consultazione di materiale didattico in formato digitale. Il secondo capitolo è suddiviso in due parti che trattano la fase di progettazione delle estensioni previste. Nella prima parte è introdotto il gruppo di lavoro BEAT e l’inserimento dei template nel sistema AContent. La seconda affronta le fasi di integrazione di LTI nelle piattaforme AContent e ATutor e si occupa della progettazione di estensioni per i due moduli *AContent Repository* e *External Tool* all’interno di ATutor. Il terzo capitolo ripercorre in dettaglio tutte le fasi di implementazione dei Template di Layout, Template di Pagina e Template di Struttura nel sistema di authoring AContent. Il quarto ed ultimo capitolo, infine, tratta l’implementazione di LTI su AContent e ATutor e l’estensione dei moduli di ATutor analizzati nella fase di progettazione.

Indice

Introduzione	i
1 Stato dell'arte	1
1.1 E-learning	1
1.2 Standard per l'e-learning	3
1.2.1 SCORM	4
1.2.2 IMS Common Content Package	9
1.2.3 IMS Common Cartridge	9
1.2.4 LTI: Learning Tools Interoperability	13
1.3 Editor e piattaforme per l'e-learning	17
1.3.1 Panoramica sugli strumenti di authoring tool	21
2 Progetto	29
2.1 Prima parte: un sistema di Template per AContent	30
2.1.1 BEAT: Bologna E-learning Authoring Tool	30
2.1.2 AContent	32
2.1.3 I Template	35
2.2 Seconda parte: AContent Live Content Link	47
2.2.1 ATutor	48
2.2.2 AContent come Tool Provider	49
2.2.3 ATutor come Tool Consumer	50
3 Implementazione: i Template in AContent	59
3.1 Impatto sul sistema esistente	61

3.2	Template di Layout	63
3.2.1	Il file Themes.js	67
3.2.2	Creare un tema	68
3.3	Template di Pagina	70
3.3.1	Creare un modello	75
3.4	Template di Struttura	76
3.4.1	Creare una struttura	81
4	Implementazione: AContent Live Content Link	84
4.1	LTI	85
4.1.1	LTI su AContent	86
4.1.2	LTI su ATutor	92
4.2	AContent Repository	95
4.3	External Tool	98
	Conclusioni	104
	Bibliografia	107

Elenco delle figure

1.1	Panoramica di LTI	15
2.1	Contenuto di una pagina prima e dopo dell'applicazione del Template di Layout "Unibo"	39
2.2	Template di Layout con testo a destra e immagine a sinistra e con testo a sinistra e immagine a destra	40
2.3	Anteprima di alcuni Template di Pagina	41
2.4	Prototipo di Template di Struttura	44
2.5	Aggiunta dell'elemento in Struttura: Content part 3	45
2.6	Implementazione di un Tool Provider di IMS LTI 1.1	50
2.7	Implementazione di un Tool Consumer (lato LMS) di IMS LTI 1.1	51
2.8	Configurazione dei parametri di connessione ad AContent per il modulo "AContent Repository"	52
2.9	Pagina di modifica dei contenuti con l'opzione "AContent"	52
2.10	Motore di ricerca di ATutor per le lezioni remote di AContent	52
2.11	Lista dei risultati di una ricerca tramite il modulo "AContent Repository"	53
2.12	Menu a tendina con la struttura della lezione corrente di ATutor	54
2.13	Icona External Tool	55
2.14	Integrazione in una pagina di un contenuto remoto tramite il modulo External Tool	55

2.15	Integrazione di contenuto didattico remoto tramite il modulo External Tool	56
3.1	Pannello per la selezione e l'applicazione di un Template di Layout	66
3.2	L'amministratore può abilitare gli autori ad applicare un Template di Layout alla singola lezione	68
3.3	Pannello di abilitazione dei Template di Pagina	72
3.4	Pannello di scelta di un Template di Pagina	72
3.5	Pannello di scelta di un Template di Pagina e box di esempio .	74
3.6	Pannello di scelta di un Template di Struttura	77
3.7	Esempio di struttura chiamata "creative based"	79
3.8	Riquadro finale di un Template di Struttura	81
4.1	Nuova icona inserita (la prima a destra)	96
4.2	Integrazione in una lezione di ATutor di un contenuto remoto tramite il modulo External Tool	99
4.3	Modulo di ricerca della lezione	99
4.4	Modulo di selezione della lezione	100
4.5	Modulo di selezione delle lezioni da importare	101
4.6	Lezione remota consultata su ATutor ma importata tramite External Tool	102

Elenco delle tabelle

1.1	Principali caratteristiche del software Xerte	22
1.2	Principali caratteristiche del software eXeLearning	23
1.3	Principali caratteristiche del software MOS-SOLO	24
1.4	Principali caratteristiche del software MS LCDS	25
1.5	Principali caratteristiche del software Udutu	26
1.6	Principali caratteristiche del software AContent	27

Capitolo 1

Stato dell'arte

La formazione a distanza si può coniugare in diverse modalità le quali possono essere distinte principalmente in *formazione in presenza* e *formazione a distanza*. Quest'ultima si è evoluta in *formazione on-line* grazie alla diffusione capillare degli strumenti informatici e della rete. Con l'avvento della "formazione on-line", conosciuta come e-learning e, più in particolare, con l'avvento delle nuove tecnologie legate al web, da una modalità individuale e di auto apprendimento passivo si è passati ad un processo complesso e dinamico che prevede una tipologia di apprendimento collaborativo e cooperativo. In particolare, l'e-learning come metodologia didattica offre la possibilità di erogare contenuti formativi elettronicamente attraverso Internet o reti intranet. Per l'utente, questo rappresenta una soluzione di apprendimento flessibile in quanto fortemente personalizzabile e facilmente accessibile[1].

1.1 E-learning

La prima generazione di formazione a distanza (FaD) risale al 1850 e prende il nome di *istruzione per corrispondenza* con i relativi corsi per corrispondenza [2]. Gli strumenti didattici venivano inviati per posta e consistevano in materiale cartaceo mentre gli allievi potevano interagire con i docenti solamente nei luoghi e nei tempi d'incontro prestabiliti. La didattica

consisteva nell'auto apprendimento, fasi di autovalutazione, invio al docente tramite posta degli esercizi svolti, valutazione del docente. Con l'avvento delle nuove tecnologie, la didattica della formazione a distanza (FaD) si è adeguata ai nuovi canali di comunicazione. Dall'insegnamento per corrispondenza si è passati all'utilizzo delle nuove tecnologie del tempo come la radio che, però, limitava in modo unidirezionale l'interazione fra docente ed allievo. Viene introdotto, in questo contesto, il concetto di formazione a distanza, diverso dalla precedente formazione per corrispondenza. La FaD di seconda generazione inizia all'incirca negli anni '60, caratterizzata dalla televisione. Quest'ultima, infatti, risultava essere un canale di comunicazione facilmente fruibile, rivolto ad una grande quantità di persone e ricca di contenuti multimediali come i suoni, le immagini e il movimento (diversamente dai libri). L'introduzione delle videocassette ha permesso la distribuzione e la riproduzione delle lezioni ma l'interazione fra docente e allievi, come il concetto di classe o aula, rimaneva ancora un punto non ancora sviluppato. Differentemente dalle lezioni frontali, tuttavia, la FaD libera dal vincolo della presenza fisica e dalla costrizione temporale. Si inizia a parlare di una dimensione *virtuale* dell'insegnamento. Con l'avvento del personal computer si passa completamente ad una nuova generazione: la FaD di terza generazione svolta offline tramite strumenti che non si avvalgono della rete (floppy disk, videodischi, CD-ROM) e on-line, caratterizzata dall'avvento delle reti. Nasce così l'e-learning, una tipologia di formazione che si basa sulla telematica e che fa uso di tutte le potenzialità che le nuove tecnologie sono in grado di offrire. La modalità tradizionale di apprendimento a distanza viene stravolta e passa da auto apprendimento passivo ad apprendimento attivo, collaborativo, cooperativo, dinamico. La FaD diviene, così, e-learning in cui la didattica viene facilitata da risorse e servizi disponibili e fruibili virtualmente per via telematica. Con il termine *e-learning* si fa, quindi, riferimento a tutte le forme di apprendimento ed insegnamento tramite supporto elettronico di carattere procedurale che mirano a realizzare la costruzione della conoscenza tramite esperienza, pratica e conoscenze dello studente. In questo contesto, siste-

mi di informazione e comunicazione, non necessariamente in rete, servono come supporto specifico per l'attuazione del processo di apprendimento[3]. L'evoluzione degli strumenti per la didattica ha portato, col tempo, all'introduzione di standard per la creazione del software di e-learning e la diffusione dei contenuti di apprendimento.

1.2 Standard per l'e-learning

Nella definizione di e-learning del paragrafo precedente è stato esplicitato come i sistemi di informazione e comunicazione non debbano necessariamente essere collegati a Internet. Questo lascia spazio ad architetture come reti private oppure a sistemi completamente offline che, aggiunto al fatto che esistono diversi sistemi e piattaforme di e-learning, conduce al problema della compatibilità dei contenuti didattici. Se, per esempio, si desidera condividere una lezione o un corso con un altro *Learning Management System* (sistema di gestione dei contenuti didattici approfondito in seguito in questo capitolo) sarà necessario che l'LMS di destinazione possa interpretare in modo corretto il formato della lezione che è stato inviato. A questo proposito, viene di seguito introdotto il concetto di *interoperabilità* come l'abilità di due o più sistemi o componenti di scambiare informazioni e di utilizzare le informazioni appena scambiate[4]. Nel tempo si sono susseguiti diversi *standard di interoperabilità* alcuni dei quali definiscono come "impacchettare" il materiale didattico da utilizzare su più sistemi di e-learning. Esistono diversi standard che propongono regole sia per racchiudere i contenuti che per la loro consultazione. Oltre ai contenuti è necessario fornire metadati descrittivi circa la struttura del pacchetto stesso. Definiamo, quindi, *Learning Object* (LO) come un insieme di elementi di contenuto, elementi di pratica ed elementi di valutazione combinati in un singolo pacchetto[5]. Il 15 Luglio 2002, l'*IEEE* (Istituto degli ingegneri elettrici ed elettronici)[6] approva un documento che trae origine dal lavoro di tre importanti organizzazioni no-profit: ARIADNE[7], IMS GLOBAL[8] e Dublin Core[9]. All'interno di tale documento viene specifica-

ta la struttura dei metadati di un'istanza di LO ed emergono dettagli quali ciclo di vita, meta-metadati, caratteristiche educative, tecniche, legislative (relative ai diritti d'autore), relazionali (eventuali collegamenti o relazioni con altri contenuti), annotazioni e classificazioni in categorie. Secondo la definizione fornita nel libro *What Are Learning Objects?*(2008)[10], un LO dovrebbe essere:

- **autoconsistente (self-contained)**
ogni LO può essere usato in modo indipendente;
- **modulare**
aggregabili ad altri LO;
- **reperibile**
ogni LO ha informazioni descrittive nei metadati che consentono di essere trovato in una ricerca;
- **interoperabile**
un LO può funzionare su diverse piattaforme LMS nel rispetto degli standard definiti che definiscono regole di “impacchettamento” e ordine di fruizione dei LO;
- **riusabile**
un LO può essere usato in contesti diversi e per molteplici scopi.

Di seguito sono riportati gli standard per la definizione dei LO presi in considerazione per lo svolgimento del progetto di tesi.

1.2.1 SCORM

Sharable Content Object Reference Model (SCORM)[11] è un content package che permette l'apprendimento di contenuti tramite ogni strumento LMS che sia SCORM-conforme. Al momento della stesura del presente documento, SCORM è alla versione 1.3 (detto SCORM 2004), anche se, il più utilizzato, rimane lo SCORM 1.2. Da notare che SCORM 2004 ha subito diverse

revisioni e la più recente è la quarta (SCORM 2004 4a Edizione). SCORM è stato creato in cooperazione tra governo, mondo accademico e industria e consolida il lavoro di IMS e ARIADNE, AICC[12] e LTSC di IEEE[13] in un unico modello di riferimento unificato. Il suo obiettivo è quello di unire diverse specifiche già esistenti (IEEE, IMS, AICC) e creare uno standard che renda i learning object interoperabili e riutilizzabili. Non deve essere visto come una guida alla didattica o alla pedagogia ma come un insieme di norme tecniche per assicurarsi che tutte le piattaforme di e-learning siano in grado di lavorare l'una con l'altra così come un DVD viene letto da tutti i lettori DVD. Se un LMS è conforme a SCORM è in grado di riprodurre qualsiasi contenuto conforme. Da diverse analisi emerge una notevole riduzione dei costi e dei tempi di integrazione dei contenuti. Le specifiche SCORM sono raccolte in tre volumi chiamati *book*:

1. **“The SCORM Overview”**

dedicato alla descrizione generale dell'iniziativa;

2. **“The SCORM Content Aggregation Model”**

si occupa di definire come aggregare ed integrare risorse educative per costruire e distribuire package ovvero unità didattiche di varia dimensione a loro volta componibili;

3. **“The SCORM Run Time Environment”**

riguarda le specifiche per l'avvio, la comunicazione, il tracciamento dei contenuti all'interno dell'ambiente di interazione Web (la piattaforma LMS).

Le nuove caratteristiche nella quarta edizione semplificano notevolmente la creazione dei contenuti. ADL (Advanced Distributed Learning), iniziativa del Dipartimento della Difesa degli Stati Uniti relativa all'e-learning e delle specifiche di SCORM [14], al momento della stesura, non aveva intenzione di creare un mercato sostenibile per l'e-learning ma voleva solo far risparmiare denaro al governo. Il progetto originale prevedeva la condivisione di contenuti

fra le varie parti del dipartimento della difesa ma gli obiettivi includevano molto più che l'interoperabilità. Di seguito, sono elencati i principali:

- **remote accessibility**

la capacità di individuare e accedere agli oggetti didattici da una postazione remota e fornire loro molte altre sorgenti. In questo contesto il termine non si riferisce all'accessibilità web quanto all'accesso a contenuti remoti;

- **adaptability**

la possibilità di offrire un insegnamento su misura per esigenze individuali e organizzative;

- **affordability**

la capacità di aumentare l'efficienza e la produttività riducendo tempi e costi relativi alle consegne;

- **durability**

la capacità di adeguarsi all'evoluzione tecnologica e ai cambiamenti senza una costosa riprogettazione, riconfigurazione o ricodifica;

- **interoperability**

la capacità di poter utilizzare i contenuti didattici sviluppati in una sede con un set di strumenti o piattaforme in un altro luogo con un diverso set di strumenti o di piattaforme;

- **reusability**

la flessibilità di incorporare contenuti didattici in più applicazioni e contesti.

È utile utilizzare SCORM quando:

- si cerca l'integrazione del sistema con altri fornitori;
- si vuole creare una estesa libreria di learning object;

- il LMS è usato per fornire e gestire i contenuti didattici.

Inoltre è largamente diffuso quando si stanno progettando contenuti didattici che possono essere riutilizzati in altri contesti o che tengano traccia delle prestazioni dell'apprendimento e che controllino lo studente e si adattino in base alle sue esigenze. Al contrario, ci sono alcune situazioni in cui l'adozione di SCORM non apporta maggiori vantaggi rispetto al "costo" di utilizzo. Alcuni esempi sono:

- quando si sta progettando un contenuto semplice o statico (come una pagina web);
- quando si ha solo bisogno di pubblicare materiale di riferimento;
- quando il contenuto è temporaneo e sarà utilizzato una volta sola in un unico sistema;
- se si sta utilizzando un unico sistema interno che non cambierà mai;
- se si stanno sviluppando contesti non tradizionali di apprendimento online come simulazioni avanzate o giochi.

SCORM può essere applicato a tutti questi scenari, ma deve essere fatto in modo intelligente e per fornire benefici senza essere un ostacolo.

Dal punto di vista tecnico, SCORM è composto da tre sotto-specifiche:

1. Content Packaging

sezione che specifica come il contenuto deve essere "impacchettato" e descritto. Si basa principalmente sullo standard XML[15]. L'organizzazione dei contenuti del pacchetto deve essere fatta in una cartella "autocontenuta" o compressa in un documento .ZIP. Questa specifica è detta *Package Interchange File* (PIF) e deve sempre contenere un file XML chiamato *imsmanifest.xml* (detto manifest) nel percorso principale. Questo documento contiene tutte le informazioni di cui il LMS necessita per fornire i contenuti. Divide, inoltre, la lezione in più parti

chiamate *SCO* che possono essere combinate in una struttura ad albero che rappresenta la lezione, conosciuto anche come *activity tree*. Il manifest contiene anche una rappresentazione XML dell'activity tree, informazioni sull'esecuzione di ogni SCO e, opzionalmente, metadati che descrivono l'intera lezione e tutte le sue parti.

2. Run-Time

sezione che specifica come il contenuto dovrebbe essere eseguito nel browser (in una finestra o in un frameset) e come comunica con il LMS. Basata principalmente su *ECMAScript*[16] (JavaScript). Il LMS può eseguire uno SCO alla volta mentre, una volta lanciato, viene eseguito un algoritmo per individuare uno script API ECMAScript fornito dal LMS. Lo script API permette, tramite funzioni proprie, lo scambio di dati con il LMS.

3. Sequencing

sezione che permette all'autore del contenuto di specificare come l'utente finale può navigare le parti della lezione. Le indicazioni sono definite da un set di regole e attributi scritti in XML e contenute nel file manifest della lezione. Alcune regole sono di seguito elencate:

- determinare quali controlli di navigazione il LMS dovrebbe fornire all'utente (tasto precedente/successivo, tabella di navigazione dei contenuti etc.);
- specificare se e quali attività devono essere completate prima di altre (prerequisiti);
- fare in modo che alcune parti della lezione abbiano maggiore importanza e rilevanza rispetto ad altre allo stato finale o nel punteggio (creazione di sezioni facoltative o fornire domande riflessive);
- selezione casuale di un sottoinsieme diverso di SCO disponibili da mostrare ad ogni nuovo tentativo (per esempio in fase di autovalutazione);

- riportare l'utente ad un particolare argomento che non è stato appieno acquisito.

“SCORM 2004”, rispetto alla versione precedente, vanta l'adozione delle specifiche “IMS Simple Sequencing Specification” che dovrebbero garantire ai progettisti di Learning Object standard un maggiore controllo sul percorso di apprendimento previsto per i diversi utilizzatori.

1.2.2 IMS Common Content Package

IMS Content Packaging[17] (Agosto 2001) è uno standard che descrive le strutture dati usate per fornire interoperabilità fra i contenuti web e i sistemi di redazione dei contenuti, fra i LMS e gli ambienti in cui vengono eseguiti. Con il concetto di interoperabilità si vuole definire un insieme standardizzato di strutture che possono essere utilizzate per scambiare contenuti. Queste strutture costituiscono la base per blocchi di dati standardizzati che consentono agli sviluppatori di software di creare materiale didattico consultabile e modificabile attraverso differenti strumenti di authoring, LMS e ambienti di esecuzione creati in modo indipendenti dal sistema da eseguire. Il campo di applicazione della specifica IMS Content Packaging è focalizzato sulla definizione di interoperabilità tra i sistemi che desiderano importare, esportare, aggregare e disaggregare pacchetti di contenuti. Viene descritto come strutturare e racchiudere blocchi di materiale didattico come, ad esempio, una lezione individuale o un insieme di corsi impacchettati che possono essere poi distribuiti.

1.2.3 IMS Common Cartridge

IMS Common Cartridge (IMS CC)[18] fornisce un formato standard per la creazione e condivisione di contenuti digitali. Al momento della stesura del presente documento, Common Cartridge è alla versione 1.1, pubblicata in data 10 maggio 2011). *Common Cartridge Accredited Profile Management*

Group (CCAPMG)[19] è responsabile del mantenimento, conformità ed approvazione delle specifiche Common Cartridge. Lo scopo principale dello standard è di consentire una stretta interoperabilità di contenuti e sistemi. Le Common Cartridge offrono, infatti, grande flessibilità nella tipologia stessa dei contenuti supportati (in questo caso, i contenuti possono essere anche applicazioni vere e proprie) e rispetto a dove questi contenuti possono essere situati (nelle applicazioni Common Cartridge, possono essere distribuiti su più piattaforme geograficamente distribuite). Common Cartridge specifica cinque punti fondamentali:

1. un formato per lo scambio di contenuti tra sistemi cosicché ci sia un modo comune di interpretazione ed organizzazione. Il contenuto è descritto in un manifesto e le componenti che costituiscono il manifesto possono essere cambiate all'interno del pacchetto scambiato fra le parti o esterno al pacchetto (per riferimento URL). Questo consente al contenuto o alle applicazioni "protette" (quelle che richiedono una licenza) di essere raggruppate in modo flessibile a contenuti non protetti;
2. uno standard per i metadati che descrivono i contenuti - basato su Dublin Core[20]. Common Cartridge è estensibile e ammette altri schemi di metadati;
3. uno standard per gli elementi di prova, test, e le valutazioni. Questo standard consente ai sistemi di apprendimento di comprendere valutazioni importate e native in modo che possano essere manipolate (ad esempio decidere quali elementi devono essere utilizzati e in quale parte della lezione) in base alle necessità del sistema di apprendimento;
4. uno standard per l'esecuzione e lo scambio di dati con applicazioni esterne in modo che possano essere parte di una singola esperienza didattica gestita attraverso il LMS. Si può avere qualsiasi tipo di applicazione situata in qualsiasi luogo, come ad esempio il social networking, wiki, sistemi di valutazione esterna, tutor personalizzati, diverse varietà di librerie web-based di contenuti o altri sistemi di questo tipo;

5. uno standard per la popolazione di forum on-line destinati alla collaborazione tra studenti. In questo modo, tali forum possono essere pre-popolati con esercizi, argomenti di discussione e così via.

Dato il forte legame fra SCORM e IMS Common Cartridge, si possono individuare diverse caratteristiche condivise sia nell'organizzazione dei contenuti, sia nella struttura propria dei relativi content package. Inizialmente, SCORM è stato sviluppato per supportare la portabilità di contenuti di apprendimento autocontenuti mentre Common Cartridge è stato sviluppato principalmente per supportare l'uso di materiali e libri digitali in un contesto didattico. Non è stato progettato per sostituire SCORM. Tuttavia i vari scenari educativi esistenti richiedono analisi sui progressi di apprendimento, contenuti interattivi, la definizione di una sequenza dei contenuti, collaborazione, aiuti e autorizzazione a cui SCORM non è in grado di rispondere ma che Common Cartridge riesce a gestire. In termini di interoperabilità, Common Cartridge è stato progettato per superare SCORM rimuovendo le componenti Run-Time e raggiungendo un accordo su specifici sottoinsiemi (spesso indicati come profili di applicazione) di specifiche ampiamente utilizzate. Vista la nativa "comprensione delle domande e delle prove", Common Cartridge non funziona come una "scatola nera" come in SCORM. Inoltre, non forza la sequenza dei corsi bensì crea una situazione di "competizione" fra le varie piattaforme di e-learning basata sulla complessità delle esercitazioni. È consentita la creazione di sequenze complesse di porzioni di contenuti (così come nei tutor personalizzati, giochi o simulazioni); questo è possibile tramite gli strumenti *LTI* introdotti nei paragrafi successivi. Infine, i fornitori che si attengono alle specifiche di Common Cartridge hanno avuto la possibilità di accordarsi su approcci molto specifici ai pacchetti di contenuti, elementi di valutazione e autorizzazioni. Questa direzione semplifica notevolmente i test di interoperabilità così da poter effettuare valutazioni di conformità attraverso una suite di strumenti informatici a disposizione. Infine, secondo lo stesso IMS Global Learning Consortium, come si apprende dal sito web, i principali benefici della Common Cartridge sono i seguenti:

1. scelta più vasta dei contenuti: consente la raccolta di risorse per l'apprendimento di varie tipologie e da varie fonti;
2. argina la chiusura del software da parte dei produttori: stabilisce i formati nativi di Common Cartridge approvati dal mondo accademico e supporta un'ampia varietà di formati prestabiliti per i contenuti, eliminando, così, complicazioni legate alla chiusura del software;
3. più opzioni di valutazione: sostiene apertamente il più utilizzato standard per lo scambio di elementi di valutazione QTI[37];
4. aumenta la flessibilità, la condivisione e il riuso: si inserisce nel contesto educativo consentendo agli istruttori di assemblare lezioni di varie tematiche e pubblicarle come pacchetti riutilizzabili e modificabili, facili da creare, condividere e migliorare.

Common Cartridge definisce un nuovo formato di scambio per i contenuti di apprendimento in grado di funzionare su qualsiasi piattaforma compatibile con LMS. La versione 1.0 supporta le seguenti funzionalità:

1. rich content:
 - (a) HTML 4.01, XML;
 - (b) web links;
 - (c) media files (per es. mpg, jpg, mp3);
 - (d) application files (per es. Microsoft Office);
2. valutazioni integrate:
 - (a) scelta multipla (a risposta unica o multipla);
 - (b) vero/falso;
 - (c) tema/saggio breve;
 - (d) compila gli spazi vuoti;

- (e) abbinamenti corretti;
- 3. forum di discussione;
- 4. metadati: sia metadati descrittivi per CC, sia basati sui ruoli per le risorse;
- 5. autorizzazione per i contenuti protetti.

Nella *recommendation* dello standard, ovvero nel documento finale approvato, è riportato che il gruppo di lavoro CCAPMG ha in programma di aggiungere ai loro prodotti il supporto per Common Cartridge nell'immediato futuro.

1.2.4 LTI: Learning Tools Interoperability

IMS LTI[21] è l'acronimo di *Learning Tools Interoperability* ed è stato creato da IMS Global Consortium all'inizio del 2010. Strutturato secondo regole formalmente definite, mira a favorire la comunicazione tra due o più entità. L'esigenza di questo strumento nasce dal fatto che diversi LMS hanno uno o più moduli o blocchi propri per gestire l'interazione con altri LMS[22] vincolando il fruitore o il fornitore di contenuti a supportare lo stesso modulo (o blocco). La soluzione adottata è la creazione di un unico plugin per tutti i LMS: LTI si propone come un Web Service[23] che si interfaccia ai contenuti che possono essere ospitati sullo stesso computer, su un server, in una rete cloud. L'utilizzo di base di LTI è quello di creare questa connessione sicura (mantenendo, comunque, tutte le problematiche legate alla sicurezza informatica) fra più sistemi senza necessità di sviluppare un software ad hoc per ogni integrazione. Più in generale, LTI vuole facilitare l'interazione di applicazioni ricche di contenuti didattici con piattaforme di LMS, portali o altri ambienti educativi. Secondo il formalismo definito, le applicazioni educative sono chiamate *Tool* (fornite dal *Tool Provider* (TP)), mentre gli LMS o piattaforme sono chiamate *Tool Consumer* (TC). Lo strumento appena presentato rende superflui i seguenti servizi comunemente utilizzati:

Web Service REST[24], utilizzato per la richiesta del TC al TP, Web Service SOAP[25] per la risposta dal TP al TC. Possiamo riassumere brevemente le azioni e le operazioni eseguite in un collegamento LTI nei seguenti punti:

1. dal browser, l'utente clicca su un collegamento ad una risorsa che richiede una connessione LTI;
2. il LMS che fa da TC elabora la richiesta come segue:
 - cerca le informazioni sull'utente e sulla lezione;
 - le informazioni sono "firmate" usando la chiave privata e OAuth (approfondito in seguito);
 - restituisce i dati al browser sotto forma di "form";
3. il browser viene immediatamente inviato tramite lo standard HTTP (e tramite il metodo POST) al TP tramite JavaScript;
4. il TP conosce la chiave privata, conosce la provenienza dei dati ed esegue le seguenti operazioni:
 - controlla, con la chiave privata, la validità del messaggio ovvero se il contenuto del messaggio è stato manipolato;
 - in caso positivo, legge le informazioni inviate;
 - imposta una sessione per l'utente oggetto del messaggio;
 - restituisce un cookie al browser che indica l'avvenuta autenticazione;
5. TC e TP condividono la stessa sessione e il browser, autenticato, può richiedere liberamente i contenuti al TP.

Nella Figura 1.1 è riportata una panoramica delle fasi appena elencate in cui, il *Tool Proxy* rappresenta uno strumento di definizione:

- dei servizi offerti;

- delle specifiche di sicurezza;
- delle capacità di ciascuna parte della comunicazione.

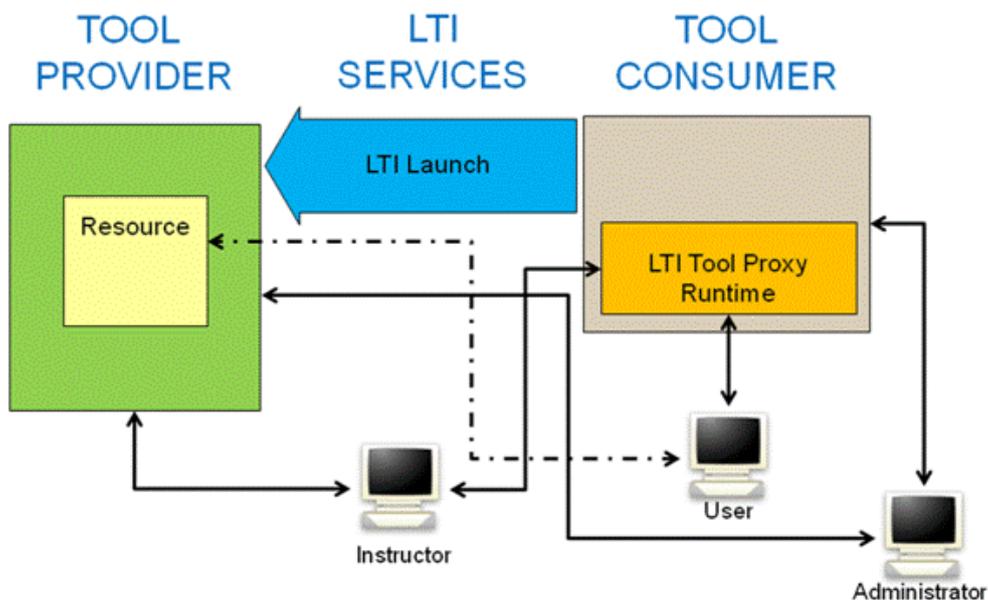


Figura 1.1: Panoramica di LTI

IMS propone LTI come uno standard innovativo perché prevede, nei prossimi cinque anni, un notevole e critico incremento dei LMS aziendali; definisce, inoltre, LTI come uno strumento affidabile, scalabile, coerente che sostiene un modello di LMS solido e centrale. Di seguito sono riportate le varie versioni che portano al consolidamento della versione attuale di LTI[21]:

- **LTI 1.0 (Basic LTI)**

inizialmente chiamato *BasicLTI*, LTI 1.0 (rilasciato a Maggio 2010) fornisce un metodo semplice ma standard per stabilire un collegamento sicuro ad un *Tool* da un *Tool Consumer*. L'adozione di questo sistema permette di offrire la stessa esperienza didattica a studenti che accedono ad applicazioni provviste di queste specifiche. La versione 1.0 è il primo passo per lo sviluppo di LTI;

- **LTI v1.1**

rilasciato nel Marzo del 2012, aggiunge la possibilità di restituire un output al Tool Consumer. Questa novità permette agli studenti di ricevere un *grado* o *punteggio* dall'applicazione con la quale è stabilita l'interazione;

- **LTI v1.1.1**

rilasciato nel Luglio del 2012, include il supporto per il ruolo di *Mentor* che può essere utilizzato per rappresentare un genitore, un tutore o un revisore.

Al momento della stesura del presente documento esiste solo un Draft di LTI 2.0 che sembra voler aggiungere nuovi servizi e arricchire gli strumenti di interazione presenti sul mercato attuale ed in continua evoluzione. Fra le nuove specifiche troviamo:

1. l'introduzione di un *Tool Proxy* che rappresenta uno strumento di definizione dei servizi offerti, descrizione delle specifiche di sicurezza e definizione delle capacità di ciascuna *parte* della comunicazione;
2. l'adozione di un framework basato su REST con una definizione dei dati scambiati descritta utilizzando JSON-LD[26].

Nonostante sia molto recente, LTI è già largamente diffuso in tutto il mondo in ambito educativo.

OAuth Authentication Protocol

Tool Consumer e un Tool Provider instaurano una connessione sicura tramite il protocollo OAuth.

OAuth (novembre 2006 [27]) è l'acronimo di *Open standard for authorization* e consiste in un protocollo aperto per una autenticazione ed autorizzazione sicura, semplice, standard fra applicazioni web, per dispositivi mobili e desktop. Si tratta di un servizio complementare a *OpenID*[28] ma, nello stesso

tempo, differente. In generale, consente ad una applicazione terza di ottenere un accesso limitato ad un servizio HTTP. Può consentire agli utenti, per esempio, di condividere le loro risorse private (foto, video, contatti) memorizzati in un unico sito con un altro sito senza dover maneggiare nuovamente le loro credenziali, in genere, senza dover inserire nuovamente nome utente e password.

1.3 Editor e piattaforme per l'e-learning

Sono introdotti, di seguito, alcuni software per l'e-learning.

Un *Learning Management System* (LMS) è un sistema software che automatizza la gestione, il monitoraggio e la reportistica di eventi di formazione. Secondo una pubblicazione di Ellis R. K. (2009)[29], un valido LMS dovrebbe avere le seguenti caratteristiche:

- centralizzare e automatizzare l'amministrazione;
- utilizzare servizi autonomi;
- assemblare e fornire rapidamente contenuti di apprendimento;
- corsi e lezioni su una piattaforma web-based scalabile;
- portabilità e supporto standard;
- personalizzare i contenuti e consentirne il riutilizzo.

Diversamente, un *Learning Content Management System* (LCMS) è un ambiente multi-utente in cui si possono creare, memorizzare, riusare, gestire e inviare contenuti didattici di un repository centrale.

Un *repository* è un archivio di contenuti digitali generalmente provvisto di un sistema per la gestione e la ricerca degli stessi. L'accesso è consentito ad applicazioni terze, indipendenti dal sistema di gestione del repository.

Una piattaforma di e-learning è un ambiente software strutturato, spesso,

come un percorso didattico con corsi, lezioni e docenti. Questi ultimi hanno, solitamente, a disposizione ambienti integrati di creazione e gestione dei contenuti e strumenti di interazione fra gli utenti come forum e chat. La piattaforma è studiata per l'apprendimento collaborativo, ricca di strumenti e di risorse informative.

Un editor di contenuti, detto anche authoring tool, è un software o un servizio che consente all'utente di produrre pagine e contenuti web[30]. Diversi programmi possono essere considerati editor dei contenuti come[31]:

- editor per la produzione di contenuti Web come *what-you-see-is-what-you-get* (WYSIWYG);
- strumenti che offrono la possibilità di salvare i contenuti in un formato web-compatibile come, per esempio, i Word Processor;
- strumenti che trasformano i documenti in contenuti web-compatibili;
- strumenti che producono materiale multimediale orientato al Web come, per esempio, la produzione di video o di editor video, *SMIL*[32];
- strumenti per la gestione o pubblicazione di siti come *Content Management Systems* (CMS);
- strumenti per la gestione del layout come gli editor CSS;
- siti Web che permettono agli utenti l'inserimento di contenuti come i blog, wiki, siti di condivisione immagini o social network.

Nell'ambito dell'e-learning, gli authoring tool sono utilizzati, spesso, per creare pacchetti di contenuto didattico. In alcuni casi, come nel caso di AContent, questi pacchetti seguono uno standard internazionale come SCORM o Common Cartridge. Viene ora riportato un elenco dei requisiti fondamentali presi in considerazione per la scelta dell'editor e della piattaforma di e-learning utilizzati nel progetto.

Scalabile

La scalabilità rappresenta la capacità di aumentare le risorse per ottenere un incremento (idealmente) lineare nella capacità del servizio. La caratteristica principale di un'applicazione scalabile è costituita dal fatto che un carico aggiuntivo richiede solamente risorse aggiuntive anziché un'estesa modifica dell'applicazione stessa. Per quanto il livello di prestazioni influisca sulla definizione del numero di utenti che l'applicazione è in grado di supportare, la scalabilità e le prestazioni costituiscono due entità distinte. In effetti, le operazioni effettuate per migliorare le prestazioni possono, talvolta, influire negativamente sulla scalabilità. La scalabilità deve essere integrata nel processo di progettazione poiché non è una funzionalità distinta che può essere aggiunta in un secondo momento[33].

Usabile

Un editor dei contenuti deve essere usabile. Con questo termine ci si riferisce alla efficacia, efficienza e soddisfazione con cui specifici utenti possono raggiungere determinati obiettivi in particolari ambienti[34].

Modulare

Con modularità ci si riferisce all'architettura dell'editor utilizzato, in particolar modo alla suddivisione del sistema in moduli. Questo garantisce che porzioni di programma non dipendano da altre e possano essere riutilizzate rendendo più semplice la realizzazione dei nuovi progetti.

Accessibile

Editor, piattaforme di e-learning e Learning Object devono attenersi a requisiti tecnici di accessibilità definiti come standard dal W3C nelle diverse specifiche stabilite e sulla base degli strumenti utilizzati. Esistono diversi standard di accessibilità ognuno dei quali ricopre una tematica diversa[35].

Integrabile

In questo caso ci si riferisce alla capacità di un LO di integrarsi con le piattaforme di e-learning così da permettere la modifica della risorsa anche una volta esportata e poi caricata.

Tipologia di editor

Gli editor possono essere di diverse tipologie. Quelle prese in considerazione sono le *applicazioni web* e le *applicazioni standalone*. Sulla base della tipologia si hanno differenti specifiche di sviluppo e differenti vincoli di utilizzo.

Configurazione

La configurazione dell'editor indica i modelli di pagina con oggetti strutturati predefiniti o permette di impostare gli strumenti sulla base delle esigenze espresse.

Tecnologia di realizzazione

Molto importante è anche la scelta della tecnologia utilizzata per la realizzazione dell'editor. Esistono strumenti realizzati in HTML, Adobe Flash[36], C# e diversi altri linguaggi e tecnologie. Ognuno di questi rappresenta una serie di vincoli al sistema operativo, all'hardware, ai componenti aggiuntivi. La soluzione scelta è quella più versatile ed elastica.

Formati dei LO prodotti

Come già accennato, la scelta dell'autoring tool deve avvenire contemporaneamente a quella del tipo di formalismo da adottare. SCORM, IMS Common Cartridge, IMS Question Test Interoperability (QTI)[37] sono tutti formalismi validi, ma non tutti gli editor sono in grado di supportarli.

1.3.1 Panoramica sugli strumenti di authoring tool

Di seguito verranno valutati strumenti *integrati, accessibili e usabili* che accompagnino l'autore dalla creazione del singolo Learning Object alla generazione di intere unità didattiche tramite la composizione di più risorse. Si giungerà, infine, ad una selezione dei migliori strumenti di e-learning che rispondano alle principali caratteristiche richieste.

Riportiamo, ora, i sistemi di authoring analizzati. Per ognuno sono state fatte valutazioni sulla base dei requisiti stilati in precedenza.

Xerte

Xml Editor and Run-Time Engine (Xerte) è un software open source prodotto dall'università di Nottingham nel 2006. È stato creato per offrire un alto grado di interazione fra utente e piattaforma e permette agli sviluppatori la “composizione” di contenuti interattivi sofisticati tramite alcuni script. Strutturato su ambiente grafico ad icone, punta alla velocità di realizzazione dei corsi tramite script predefiniti; senza scrivere codice, permette di integrare testo e media in una interfaccia accessibile. Un utente può creare strutture complesse o criteri di interazione modificando parti di codice, personalizzando l'interfaccia, cambiando il sistema di navigazione o i LO. A Xerte si affianca Xerte Online Toolkits, una suite di strumenti web che si rivolge agli autori di contenuti e permette di “assemblare” i contenuti con semplici procedimenti wizard. Può essere facilmente estesa tramite la piattaforma stessa e facilita la collaborazione nelle fasi di creazione del materiale didattico. Nonostante la sua flessibilità e la sua estensibilità, Xerte rimane legato alla piattaforma Microsoft Windows e sviluppato tramite il software proprietario Adobe Flash.

CARATTERISTICA	VALORE
Open source	Si
Ultima versione	2.17
Data ultimo release	23 Agosto 2011
Progetto in attività	Si
Tipologia applicazione	Standalone
Sistema Operativo	Microsoft Windows
Monolitico o modulare	Modulare
Linguaggio di programmazione	ActionScript XML
Formati in input	Learning Object Learning Template
	MXML Object
Formati in output	SCORM 1.2 SCORM 2004

Tabella 1.1: Principali caratteristiche del software Xerte

ExeLearning

eXeLearning è interamente sviluppato in XUL (XML User interface Language), linguaggio di markup per l'interfaccia utente, sviluppato da Mozilla Project e considerato una distorsione di XML. Non è uno standard W3C ma fa uso di Javascript, CSS, DTD, RDF e permette la realizzazione di interfacce grafiche cross-platform (portabili su diversi sistemi operativi) come Mozilla Firefox[38], Mozilla Thunderbird[39] e Mozilla Seamonkey[40]. eXeLearning fornisce componenti predefinite come la galleria di immagini, oggetti di multi selezione, wiki, quiz e altri da concatenare per creare un LO. Per ognuno è possibile definire caratteristiche descrittive, posizione rispetto agli altri e parametri di funzionamento (per esempio per i quiz). Il menu offre anche sette profili grafici predefiniti da poter applicare all'intero LO. I formati di output

sono diversi, fra cui, gli standard SCORM e Common Cartridge. Nonostante la quantità di opzioni a disposizione, la libertà di personalizzazione dei contenuti alle diverse esigenze è, tuttavia, limitata. Infine, per motivi non divulgati, il progetto di eXeLearning è stato abbandonato.

CARATTERISTICA	VALORE
Open source	Si
Ultima versione	1.04
Data ultimo release	09 Settembre 2008
Progetto in attività	Progetto terminato
Tipologia applicazione	Standalone
Sistema Operativo	Microsoft Windows (XP, Vista)
	MAC OS X
	Linux
Monolitico o modulare	Monolitico
Linguaggio di programmazione	Python
	XUL
Formati in input	.elp
Formati in output	.elp
	IMS Common Cartridge
	SCORM 2004
	Pagine autoreferenziate in file .zip
	File di testo
	SCORM Quiz

Tabella 1.2: Principali caratteristiche del software eXeLearning

MOS-SOLO

Come altri tool analizzati, anche MOS-SOLO, software sviluppato dall'azienda francese Mindonsite[41], offre un set di template grafici predefiniti da poter applicare all'intero LO. Questo strumento si distingue dagli altri per

l'alto numero di metadata a disposizione per descrivere la lezione. L'autore può specificare persino un obiettivo primario e uno o più secondari; questa scelta è stata presa fortemente in considerazione nel progetto di tesi in quanto ritenuta molto interessante anche dal punto di vista pedagogico. L'editor dei contenuti permette il controllo automatico dell'ortografia e offre le stesse caratteristiche di formattazione del testo degli editor *what-you-see-is-what-you-get* (WYSIWYG), ormai largamente diffusi. MOS-SOLO è legato alla piattaforma Microsoft Windows e non è open source, tuttavia offre diversi spunti e valide considerazioni per il progetto di questa tesi.

CARATTERISTICA	VALORE
Open source	No
Ultima versione	Mos Solo R1 036
Data ultimo release	-
Progetto in attività	Si
Tipologia applicazione	Standalone
Sistema Operativo	Microsoft Windows (XP, Vista, 7)
Monolitico o modulare	-
Linguaggio di programmazione	basato su ActiveX (IE 6+)
Formati in input	SCORM 1.2
	SCORM 2004
Formati in output	SCORM 1.2
	SCORM 2004

Tabella 1.3: Principali caratteristiche del software MOS-SOLO

MS LCDS

MicroSoft Learning Content Development System (MS LCDS) offre diverse possibilità di creazione e fruizione dei contenuti. Il software è estremamente legato alla piattaforma e non è open source. Consente la creazione di *eLearning Course* e *Learning Snack*. Il primo è basato sulle ActiveX di Inter-

net Explorer e permette l'esportazione dei contenuti in diversi formati (per esempio Word Document, Index etc) di cui SCORM 1.2 è l'unico standard. Il secondo, vuole creare un sistema di consultazione dei contenuti didattici basato sul framework Silverlight[42] di Microsoft. A parte l'esportazione nell'ormai sorpassato SCORM 1.2, MS LCDS non mostra alcuna apertura agli standard e nessuna propensione all'interoperabilità.

CARATTERISTICA	VALORE
Open source	No
Ultima versione	2.8
Data ultimo release	-
Progetto in attività	Si
Tipologia applicazione	Standalone
Sistema Operativo	Microsoft Windows (XP, Vista, 7)
Monolitico o modulare	-
Linguaggio di programmazione	-
Formati in input	Word Document, Index, SCORM 1.2 ...
Formati in output	Word Document, Index, SCORM 1.2 ...

Tabella 1.4: Principali caratteristiche del software MS LCDS

Udutu

Udutu è un'applicazione web che permette di creare contenuti didattici ed esportarli nel formato SCORM. La piattaforma offre anche la possibilità di consultazione del materiale didattico. Un autore vedrà organizzata una lezione secondo una struttura ad albero che si dirama fino alle singole pagine. Nelle fasi di creazione dei contenuti, Udutu permette di selezionare un layout per ogni pagina della lezione con una modalità simile a quella di Microsoft Power Point[43]. Il layout scelto determina il tipo di contenuti (titolo, testo e immagine oppure titolo, testo e video) e la loro disposizione. Giacché Udutu non è open source, oltre alle caratteristiche offerte, non c'è alcuna

possibilità di apportare modifiche al sistema. Tuttavia, la piattaforma appena analizzata ha offerto ampio spunto per quanto riguarda i template di pagina e i layout grafici. Per ogni LO creato si può scegliere un tema grafico da un elenco predefinito e si possono personalizzare diverse componenti per adattarlo alle proprie esigenze. È possibile cambiare le immagini, gli stili dell'editor (l'editor adottato è *ASP.net WYSIWYG*) come il colore del testo, lo sfondo della pagina, la dimensione e il tipo di carattere, il set di icone per la navigazione e, infine, intervenire direttamente sul codice CSS incluso nel LO finale.

CARATTERISTICA	VALORE
Open source	No
Ultima versione	-
Data ultimo release	-
Progetto in attività	Si
Tipologia applicazione	Web Application
Sistema Operativo	-
Software	-
Monolitico o modulare	-
Linguaggio di programmazione	ASP.net
Formati in input	SCORM 1.2 SCORM 2004
Formati in output	SCORM 1.2 SCORM 2004

Tabella 1.5: Principali caratteristiche del software Uduu

AContent

AContent è un software di authoring open source e basato su web, quindi, non dipendente da una specifica piattaforma. Si tratta di un sistema monolitico, tuttavia permette una facile integrazione di blocchi di codice sviluppato

da terzi. È supportata l'importazione e l'esportazione dei più diffusi standard per i LO come SCORM, Common Cartridge e lo stesso per QTI, standard che racchiude le specifiche XML per lo scambio di domande e compiti/esercizi. AContent nasce come modulo del sistema del LCMS ATutor. Nel corso del tempo e con il rilascio di versioni sempre più consistenti dell'allora modulo, il consorzio artefice di ATutor e di AContent, ha deciso di separare i due strumenti. Alla versione attuale, nonostante sia ancora un modulo di ATutor per ragioni di retro compatibilità, ATutor è diventato un sistema di authoring completamente indipendente.

CARATTERISTICA	VALORE
Open source	Si
Ultima versione	1.2
Data ultimo release	14 Ottobre 2011
Progetto in attività	Si
Tipologia applicazione	Web Application
Monolitico o modulare	Monolitico
Linguaggio di programmazione	PHP
Formati in input	IMS Common Cartridge
	SCORM 2004
	IMS QTI Tests
Formati in output	IMS Common Cartridge
	SCORM 2004
	IMS QTI Tests

Tabella 1.6: Principali caratteristiche del software AContent

Capitolo 2

Progetto

Il progetto di tesi presentato potrebbe essere suddiviso in due parti. La prima riguarda la progettazione e l'implementazione di un sistema di template per AContent. Per template si intendono strumenti di grande efficacia e di facile utilizzo nelle fasi di redazione dei contenuti al fine di strutturarli, organizzarli e definirli graficamente. Infatti, si prevede di aggiungerli ad AContent senza la necessità di integrare un intero motore di template ma utilizzando strutture dati esistenti e specifiche standard di e-learning. I servizi aggiuntivi da offrire agli autori sono stati organizzati secondo tre approcci diversi da cui sono emersi tre livelli di template. Nel paragrafo che segue verranno approfonditi i concetti accennati e illustrati i tre tipi di template emersi in fase di progettazione.

La seconda parte tratta l'interazione tra AContent e ATutor. Si vuole creare, tra i due sistemi, un canale di comunicazione LTI autenticato per mezzo del protocollo OAuth. Quindi sono descritti gli interventi necessari a trasformare le due parti in Tool Provider e Tool Consumer. Infine vengono trattati due moduli di ATutor e progettata una loro estensione così da stabilire uno scambio di informazioni strutturato su LTI.

2.1 Prima parte: un sistema di Template per AContent

Lo scopo di questo paragrafo è presentare un'estensione per il sistema AContent progettata e sviluppata per supportare l'utilizzo di template durante la creazione e la modifica dei contenuti di e-learning. L'idea nasce dall'esigenza dell'Università di Bologna di trovare un Authoring Tool da affiancare a Moodle ma da esso indipendente, che possa fornire all'autore la possibilità di una creazione "guidata" dei LO, essere di facile utilizzo per qualsiasi docente, prevedere un insieme di "meta template" (descrizione generale e di dettaglio sul singolo LO) che lo guidino nella creazione di un percorso formativo. L'introduzione dei template vuole dare coerenza ai contenuti creati e rappresentare una sorta di standard di qualità tecnica e didattica. Vuole fornire, inoltre, un sistema di navigazione facile e flessibile con il pieno controllo delle funzionalità tecniche mantenendo un formato di visualizzazione disponibile per i diversi browser web e riuscendo a modellare i LO adeguandoli ai diversi modelli educativi. L'idea di introdurre i template è ispirata ai diversi strumenti analizzati nella fase iniziale del progetto e di cui riportiamo i dettagli e le considerazioni formulate.

2.1.1 BEAT: Bologna E-learning Authoring Tool

Il progetto di tesi si colloca nella cornice di riferimento di *OERtest*[44], un progetto europeo della durata di due anni (Ott. 2010 - Sett. 2012) per sostenere l'integrazione delle *Open Educational Resources* (risorse educative di libera condivisione) nell'ambito dell'istruzione superiore e per la valutazione dell'apprendimento conseguito. In particolare, l'obiettivo si riassume in una serie di standard per regolamentare l'offerta di risorse educative e di servizi mirati alla valutazione tra cui:

- linee guida per le valutazioni;
- strumenti di verifica della qualità e del carico di lavoro;

- criteri di aggiudicazione e associazioni con la scala di valutazione ECTS[45];
- una struttura di lavoro per programmi pilota.

Proprio all'interno di questo contesto emerge la stretta collaborazione tra le diverse figure professionali di informatici e pedagogisti come frutto del lavoro coordinato tra l'università di Scienze della Formazione di Bologna, partner del progetto OERtest, e Scienze e Tecnologie Informatiche di Cesena. Sulla base di questa cooperazione è stato creato *BEAT*, acronimo di Bologna E-learning authoring tool, gruppo di lavoro il cui obiettivo è offrire agli autori un valido e semplice strumento che fornisca suggerimenti e metodologie per la progettazione e produzione di materiale didattico. Il funzionamento deve essere indipendente dalla piattaforma Moodle (adottato da tempo dall'Università di Bologna come valido strumento LCMS) ma, allo stesso tempo, produrre LO compatibili tramite l'adozione dei più diffusi standard di e-learning.

L'idea è di creare un'estensione di un authoring tool che implementi il concetto di template ovvero di strumenti di grande efficacia e di facile utilizzo che aiutino l'autore nelle fasi di redazione dei contenuti. Pur essendo Moodle un valido strumento didattico, non è tuttavia provvisto di un sistema sofisticato di creazione dei contenuti che permetta di strutturarli, organizzarli e definirli graficamente. Da qui, l'esigenza dell'università di trovare un authoring tool da affiancare a Moodle ma da esso indipendente, che possa fornire all'autore la possibilità di una creazione "guidata" dei LO, essere di facile utilizzo per qualsiasi docente, prevedere un insieme di "meta template" (descrizione generale e di dettaglio sul singolo LO) che lo guidino nella creazione di un percorso formativo. 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 etc) facile e flessibile mantenendo un formato di visualizzazione indipendente dal sistema operativo e dal browser e riuscendo a modellare i LO adeguandoli alle diverse esigenze di insegnamento.

2.1.2 AContent

La scelta di utilizzare un authoring tool rispetto ad un altro ha un forte impatto sulla qualità finale dell'output prodotto e questo risulta evidente soprattutto nelle situazioni in cui i contenuti sono prodotti e condivisi su larga scala. I punti principali su cui focalizzarsi nella scelta sono i seguenti:

- conformità ai formati standard di e-learning per garantire la portabilità attraverso diverse piattaforme/sistemi;
- creazione di nuovi contenuti, nonché la modifica di quelli esistenti, senza la necessità di specifiche competenze tecnologiche.

In particolare, la scelta è stata guidata dai seguenti fattori:

- l'authoring tool deve essere open source per permetterne l'analisi del funzionamento interno e la modifica del codice sorgente;
- deve produrre un output conforme ai principali standard dell'e-learning;
- deve essere accessibile e supportare la creazione di contenuto accessibile in modo da essere esportato ed importato in Moodle.

Per questi motivi si è scelto di lavorare con AContent[46], un sistema open source di redazione di contenuti didattici utilizzato per creare contenuti di apprendimento interoperabili, accessibili, adattativi, basati sul Web. AContent è un progetto open source, un sistema di redazione e raccolta dei contenuti. Viene usato per creare contenuti per l'apprendimento interoperabili, accessibili ed adattabili. Basato su web, dà la possibilità di costruire, condividere ed archiviare materiale didattico e test d'apprendimento. L'interoperabilità è evidente durante la fase di esportazione dei contenuti producendo uno strumento autosufficiente da usare con qualsiasi sistema che supporti contenuti IMS standard. AContent fornisce le seguenti funzionalità per la creazione di contenuti:

- testo;

- equazioni/notazione scientifica (in LaTeX);
- immagini;
- video/Animazioni;
- audio;
- link;
- tutti i linguaggi di markup e CSS per la modifica dinamica della formattazione dei contenuti (utilizzato per creare slide delle lezioni e dispense stampabili).

È disponibile l'importazione ed esportazione dei contenuti secondo i seguenti standard:

- IMS Common Cartridge;
- IMS Content Package;
- importazione and esportazione di test o archivi di domande in pacchetti IMS QTI.

Anche Moodle, dalla versione 2.2, fornisce pieno supporto per l'importazione e l'esportazione dei pacchetti Common Cartridge e Content Package estendendo la sua capacità di interoperabilità. Inoltre, in AContent, i seguenti standard sono pienamente supportati:

- W3C XHTML 1.1;
- WCAG 2.0.

Nella scelta dello strumento di progetto è stata presa in considerazione anche la legge "9 gennaio 2004, n. 4", conosciuta come Legge Stanca in onore del ministro che l'ha voluta, approvata all'unanimità dal Parlamento

italiano.[47]. Particolare attenzione è ricaduta sulle specifiche relative all'utilizzo di Javascript: come specificato nell'allegato A intitolato "Verifica tecnica e requisiti tecnici di accessibilità delle applicazioni basate su tecnologie internet"[48], i contenuti e l'utilizzo delle pagine devono essere fruibili anche in caso di disattivazione degli script. Inoltre, bisogna evitare di creare collegamenti ipertestuali che usino "javascript" come URL. Queste specifiche pongono un notevole problema in quanto gli strumenti scelti per il progetto basano parte del loro funzionamento proprio su Javascript. Disabilitandone l'utilizzo, si andrebbero a creare malfunzionamenti o un uso parziale degli strumenti stessi. Va specificato, comunque, che una revisione dell'allegato A del DM 8 luglio 2005 (versione 26 aprile 2010) intitolata "Criteri e metodi per la verifica tecnica e requisiti tecnici di accessibilità per i contenuti e i servizi forniti per mezzo di applicazioni basate su tecnologie Web"[49] specifica chiaramente che l'utilizzo di ecma-script (Javascript) è compatibile con i requisiti di accessibilità. Nello stesso documento, viene annunciato che i requisiti tecnici di accessibilità e i relativi punti di controllo per la verifica di conformità sono stati definiti sulla base delle WCAG 2.0[50] del W3C, specifiche già rispettate dagli strumenti adottati. La legge sancisce, inoltre, le "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici". L'articolo 1 si occupa di definire in modo chiaro Obiettivi e Finalità della legge:

1. *La Repubblica riconosce e tutela il diritto di ogni persona ad accedere a tutte le fonti di informazione e ai relativi servizi, ivi compresi quelli che si articolano attraverso gli strumenti informatici e telematici.*
2. *È tutelato e garantito, in particolare, il diritto di accesso ai servizi informatici e telematici della pubblica amministrazione e ai servizi di pubblica utilità da parte delle persone disabili, in ottemperanza al principio di uguaglianza ai sensi dell'articolo 3 della Costituzione.*

Il punto 1 dell'articolo 5 intitolato "Accessibilità degli strumenti didattici e

formativi” di seguito riportato specifica l’obbligo di rendere accessibile anche il materiale formativo e didattico a disposizione.

1. Le disposizioni della presente legge si applicano, altresì, al materiale formativo e didattico utilizzato nelle scuole di ogni ordine e grado.

2.1.3 I Template

Nel contesto informatico, un *template* è un modello predefinito ricorrente usato come base di partenza per la creazione di un documento che, solitamente, può essere modificato. Il concetto appena espresso può essere avere diverse sfumature: esistono modelli grafici che propongono una veste (immagini, colori etc) al documento o al programma, modelli organizzativi che determinano la struttura dei contenuti, modelli di risoluzione dei problemi etc. I principali vantaggi son dati dal riutilizzo di uno stesso template che può essere definito una volta sola ed è applicabile in situazioni simili, dal risparmio di tempo nella creazione e dalla definizione di una struttura che rispetti determinati formalismi. I template sono stati aggiunti ad AContent senza la necessità di integrare un intero motore di template ma utilizzando strutture dati esistenti e specifiche standard di e-learning. I servizi aggiuntivi da offrire agli autori sono stati organizzati secondo tre approcci diversi; dal gruppo BEAT e dal contributo fornito da questa tesi segue una soluzione strutturata secondo i seguenti livelli di template:

1. **Template di Layout**
per il controllo dell’aspetto grafico;
2. **Template di Pagina**
per la struttura di ogni singola pagina;
3. **Template di Struttura**
per la struttura dell’intero contenuto didattico.

I principali vantaggi nell'utilizzo di un sistema di template sono dati dalla possibilità per gli autori di usufruire di un supporto grafico per l'organizzazione e la presentazione dei contenuti aumentando, inoltre, l'usabilità degli authoring tool e offrendo nuove efficaci forme per la produzione di materiale didattico. La disposizione di un set di regole da seguire per la definizione dei template si mostra necessaria ogni qual volta si desidera che i contenuti siano prodotti in larga scala. Oltre alla grafica, i nuovi strumenti introdotti consentono di strutturare alcune caratteristiche didattiche dei contenuti (metodologia, descrizione degli obiettivi, valutazione). Questo approccio offre supporto agli autori nelle fasi di redazione permettendogli di prestare attenzione, allo stesso tempo, all'aspetto pedagogico e didattico. I tre template sono indipendenti l'uno dall'altro, questo significa che possono essere utilizzati singolarmente e senza alcuna dipendenza. Tuttavia, viene di seguito suggerita la seguente sequenza di applicazione su tre livelli:

1. primo livello

ogni lezione può essere creata sulla base di un Template di Struttura che offre una organizzazione predefinita della lezione come gli obiettivi, i contenuti, le valutazioni, gli strumenti, i riferimenti etc;

2. secondo livello

ogni pagina della struttura è associata ad un Template di Pagina che schematizza il contenuto della pagina suddividendolo in sottoparti;

3. terzo livello

ad ogni pagine è associata ad un Template di Layout che ne determina l'aspetto grafico.

L'intero progetto svolto è stato presentato alla conferenza "E-LEARN 2012 - World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education" [51] a Montréal, Québec nell'ottobre del 2012 e riassunto nella pubblicazione "BEAT: an AContent Extension to Support Authors in Developing Learning Pathways" [52]. L'obiettivo principale della

prima parte della tesi è stato progettare e implementare una o più funzionalità per supportare l'utilizzo di template durante la creazione e la modifica dei contenuti di e-learning. Solitamente, i template web sono applicati run-time mediante un motore di template incorporato nella piattaforma che fornisce e gestisce contenuti. I template per l'e-learning, invece, devono essere compatibili con i ben noti meccanismi standard di "impacchettamento" dei contenuti: autocontenuti e indipendenti dalla piattaforma. Di conseguenza, non possono fare affidamento sui cosiddetti motori previsti dai LCMS o dai *repository* e vengono utilizzati solo durante la fase di progettazione e di editing per creare pacchetti standardizzati di contenuti di apprendimento.

I principali vantaggi ottenuti con l'introduzione dei template nei sistemi di authoring sono:

- il supporto degli autori nella progettazione (tramite percorsi predefiniti) e nella creazione (tramite esempi di contenuti) dei contenuti. I template aumentano l'usabilità degli strumenti di authoring fornendo nuovi e più efficaci modi per la produzione di contenuti didattici e per la loro personalizzazione;
- la disponibilità di un insieme di regole da seguire. Questo è generalmente molto utile ma risulta necessario ogni qual volta i contenuti vengano prodotti su larga scala e quando alcuni standard qualitativi/quantitativi debbano essere soddisfatti da tutti gli autori.

Template di Layout

Sono, di seguito, presentate le nuove funzionalità aggiunte in AContent 1.2. Le caratteristiche introdotte permettono all'autore di applicare al contenuto selezionato un template, detto *Template di Layout*, inteso come veste grafica. L'interfaccia grafica facilita notevolmente le operazioni di modifica dell'aspetto dei contenuti. Un'applicazione d'esempio dalla quale è stato preso spunto è eXeLearning. Nel menu a tendina "Styles" del programma è mostrato un elenco di temi grafici applicabili al contenuto della lezione che

si sta creando. Nella tesi, dal punto di vista tecnico, ma senza scendere nel dettaglio, la creazione di un template avviene tramite la manipolazione di un *foglio di stile* (CSS)[53]. Il CSS (dall'inglese *Cascading Style Sheets*) è un linguaggio standard definito dal W3C che serve a descrivere alcune caratteristiche di visualizzazione delle pagine web come i colori, l'impaginazione e il tipo di carattere. È indipendente da HTML e questo consente una separazione fra contenuti e presentazione rendendo più semplice la manutenzione delle applicazioni web[54]. Dall'editor integrato è possibile inserire e modificare codice CSS per ogni pagina. Tuttavia, l'idea di appoggiarsi a tale strumento è stata esclusa principalmente per i seguenti motivi:

- l'aggiunta di codice CSS può essere fatta solo pagina per pagina e non per una lezione intera;
- la possibilità di inserire liberamente codice CSS permette all'autore di interferire con il foglio di stile principale dell'intero sistema;
- la scrittura di codice CSS richiede competenze informatiche che non tutti possono vantare.

Creando un sistema di template già pronti si ha maggiore controllo sul loro utilizzo, sugli effetti all'interno del software e si fornisce uno strumento di facile utilizzo fruibile anche da tutti coloro che non si occupano di informatica. Un Template di Layout è, quindi, un foglio di stile preconfezionato che l'autore può applicare al contenuto. L'amministratore del sistema dovrebbe poter controllare, tramite un campo opzionale, se permettere di intervenire anche sulla singola pagina, oltre che sull'intera lezione.

L'idea risultante è mostrata nella Figura 2.3 la quale mostra un blocco di testo semplice prima dell'applicazione di un template. Come si evince dall'immagine, le uniche modifiche apportate sono date dal carattere grassetto e dal tipo "intestazione 1", assegnato al titolo della pagina. L'immagine successiva rappresenta lo stesso testo ma con applicata una veste grafica dell'Università di Bologna. I contenuti testuali sono rimasti invariati mentre sono

state aggiunte, in modo completamente trasparente all'autore, immagini ed interlinee proprie del tema selezionato.



Figura 2.1: Contenuto di una pagina prima e dopo dell'applicazione del Template di Layout "Unibo"

L'applicazione di nuovi template grafici al contenuto consente non solo di personalizzarne la visualizzazione ma anche di definirne le modalità di rappresentazione sotto alcuni aspetti di accessibilità web. Un esempio può essere il colore del testo, una particolare sottolineatura, testo con bordo o senza oppure un differente allineamento dei contenuti. Una volta adottato un Template di Layout è possibile apportare ulteriori modifiche alla veste grafica tramite lo strumento di editing TinyMCE[55] a disposizione su AContent. Nella Figura 2.5 (a), per esempio, è stato applicato il Template di Layout "Unibo" che prevede l'allineamento del testo a destra e il logo a sinistra. L'autore ha modificato le impostazioni di base spostando a sinistra del testo l'immagine, come mostrato nella Figura 2.5 (b).



(a) Template di Layout con testo a destra e immagine a sinistra



(b) Template di Layout con testo a sinistra e immagine a destra

Figura 2.2: Template di Layout con testo a destra e immagine a sinistra e con testo a sinistra e immagine a destra

L'utilizzo di template grafici ai contenuti non risulta limitativo rispetto alla creatività dell'autore. Dall'editor di AContent è possibile modificare direttamente il codice HTML del contenuto così da poter assegnare nuove classi e nuovi identificatori HTML; il tutto personalizzabile da codice CSS. Come da progettazione, l'esportazione nei diversi formati come Common Cartridge o SCORM manterrà la nuova configurazione grafica. Questa strategia assicura che i contenuti non siano strettamente dipendenti dalla piattaforma per la quale il contenuto stesso è stato creato e sottolinea, inoltre, la portabilità del sistema progettato. E' previsto, infatti, che, durante l'esportazione dei contenuti didattici, le informazioni relative ai Template di Layout siano mantenute nel file esportato e ripristinate successivamente all'importazione (per

esempio in Moodle). L'attenzione a questi particolari ha fatto sì che l'Università di Bologna utilizzasse AContent come authoring tool e continuasse ad usare Moodle come piattaforma per l'e-learning. La corretta importazione dei nuovi contenuti prodotti con AContent ha dimostrato, quindi, l'ottimo grado di interoperabilità con Moodle.

Template di Pagina

Le caratteristiche introdotte, in questo caso, permettono all'autore di impostare un'organizzazione della pagina, detta *Template di Pagina*, intesa come un modello di pagina. Una volta applicato, determina la struttura dei contenuti come, ad esempio, la disposizione di oggetti testuali e grafici all'interno della pagina desiderata. Il tutto avviene, però, senza modificare i contenuti stessi. Un modello può rappresentare la disposizione del testo sulla base di un'immagine o la disposizione di titolo, sottotitolo e testo, in funzione di come è stato creato il modello stesso. Non definisce né contenuti testuali, né contenuti grafici. La Figura 2.3 mostra l'anteprima di alcuni esempi di Template di Pagina contenuti nel riquadro.

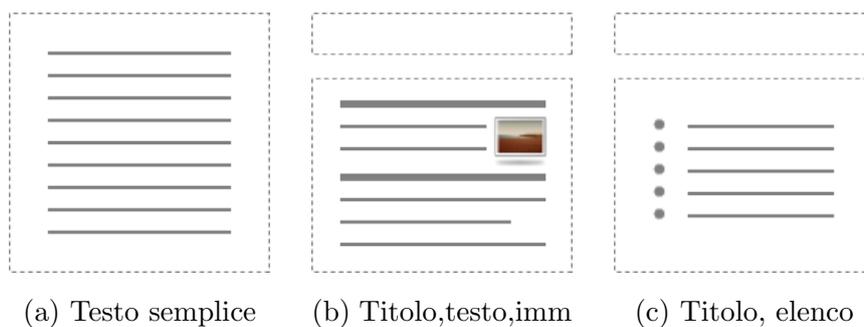


Figura 2.3: Anteprima di alcuni Template di Pagina

L'idea di un Template di Pagina è ispirata alle “diapositive” del programma Microsoft Power Point[43] per creare presentazioni e ai “layout” di Uduu. Come per il progetto di tesi, i template di pagina rappresentano un'anteprì-

ma della collocazione, all'interno della pagina, di testo, titolo, immagini e di tutte le componenti che un autore può voler inserire. Si è voluto, in un certo senso, richiamare la semplicità con la quale si può determinare il modello di pagina da applicare. Una volta selezionato e applicato un modello, la struttura può essere ulteriormente modificata aggiungendo e rimuovendo sottoparti. Anche in questo caso, l'editor di AContent permette all'autore di modificare i contenuti redatti senza che i layout impongano alcuna limitazione. L'autore può, per esempio, creare una struttura di pagina con immagine e testo, rimuovendo il testo e sostituendolo con un file audio. Il meccanismo è progettato per supportare la facile creazione dei contenuti senza limitare, in alcun modo, la creatività dell'autore. Ogni pagina di template, come si vedrà meglio nel capitolo relativo all'implementazione, è un frammento HTML strutturato secondo gli standard XHTML.

Creato il 26 Gennaio del 2000 come standard del W3C, *eXtensible HyperText Markup Language* (XHTML)[56] è un linguaggio di markup che associa alcune proprietà di XML con quelle di HTML. È, sostanzialmente, una riformulazione di HTML 4 basata su XML 1.0 e tre *DTD* corrispondenti a HTML 4. Le semantiche degli elementi e dei loro attributi sono definite nella *recommendation* del W3C per HTML 4. Queste semantiche definiscono le fondamenta per una futura estensibilità di XHTML. L'adozione di XHTML 1.0 porta i seguenti vantaggi:

- i documenti XHTML sono XML conformi, quindi, possono essere facilmente visualizzati, modificati e validati con strumenti XML standard;
- i documenti XHTML sono scritti per funzionare meglio di quanto non facesse prima l'HTML 4 con i browser;
- possono utilizzare applicazioni (ad esempio, script e applet) che si basano sia sul modello HTML Document Object Model (HTML DOM) che su XML Object Model;

- per ogni evoluzione della famiglia di linguaggi XHTML, i documenti conformi a XHTML 1.0 avranno sempre più probabilità di interagire correttamente all'interno e tra diversi ambienti XHTML.

XHTML è nato dalla necessità di un linguaggio dotato di una sintassi migliore dell'HTML e dalla diffusione dei nuovi dispositivi diversi dai tradizionali computer. Ormai, dato il largo utilizzo dello standard, quasi tutti i browser compatibili con XHTML rendono correttamente anche i documenti HTML.

Template di Struttura

L'ultimo livello di template prodotto è il *Template di Struttura*. Il meccanismo è ispirato ad alcuni authoring tool presi in analisi come MyUDUTU[57] e MSLCDS[58]. Il primo, durante la creazione di una lezione, propone all'autore la scelta di alcuni tipi di struttura come, per esempio, *struttura base*, *struttura di valutazione*, *struttura avanzata* etc. Per ognuno, permette (opzionalmente) o richiede (obbligatoriamente) di inserire i contenuti, selezionare il layout, aggiungere una descrizione etc. Il secondo suggerisce, in modo più diretto, la struttura di base di ogni lezione e la presenta all'autore composta da introduzione, due argomenti e autovalutazione finale. In entrambi i casi è possibile personalizzare ulteriormente l'ordine o le componenti della struttura stessa. Tramite meccanismi differenti, queste applicazioni offrono agli autori un modo per creare una lezione base standardizzata partendo da un set di modelli predefiniti. Nel caso del progetto di tesi, l'intenzione è di permettere agli autori di creare strutture vuote di lezioni partendo da modelli predefiniti. I Template di Struttura sono organizzati come una gerarchia di cartelle e di pagine, ognuna delle quali associata ad un Template di Pagina. Nella creazione di un template di questo tipo è possibile determinare quali pagine saranno obbligatorie o opzionali per l'autore. Le pagine obbligatorie di una data struttura, al contrario di quelle opzionali, non possono essere rimosse dalla lezione. Di seguito viene proposto un prototipo di un Template di Struttura.

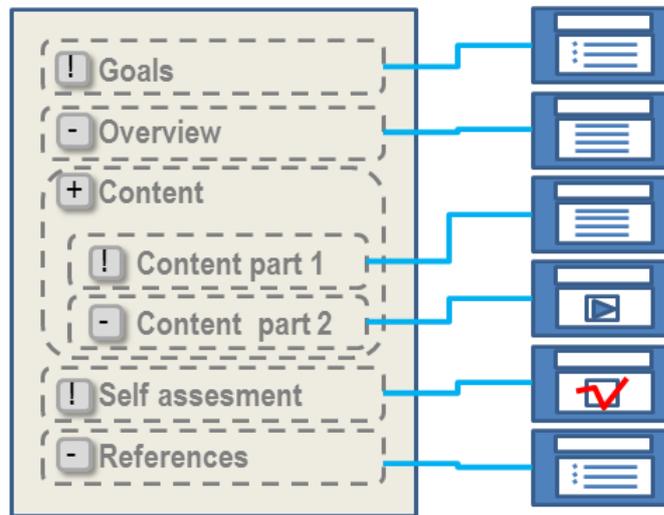


Figura 2.4: Prototipo di Template di Struttura

La figura precedente mostra:

1. una pagina obbligatoria chiamata *Goals*;
2. una pagina opzionale chiamata *Overview*, associata ad un template testuale;
3. una sotto struttura dal titolo *Contenuto* composta da:
 - (a) una pagina obbligatoria dal titolo “Contenuto parte 1”, associata al template “Testo”;
 - (b) una pagina facoltativa chiamata “Contenuto dell’articolo 2”, associata al template “MM”;
4. una pagina obbligatoria dal titolo “Valutazione”, associata al Template di Pagina “Scelta multipla”;
5. una pagina opzionale intitolata “Riferimenti” relativa al Template di Pagina “Lista”.

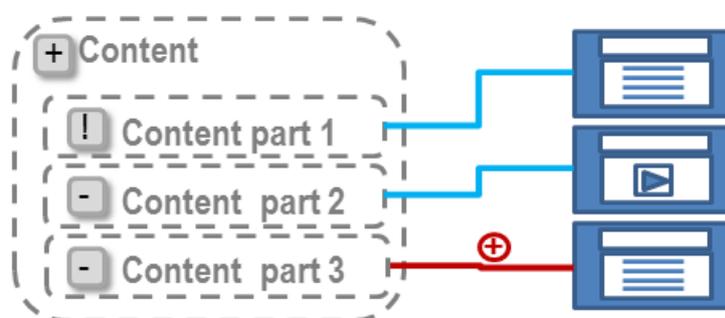


Figura 2.5: Aggiunta dell'elemento in Struttura: Content part 3

L'immagine 2.5 mostra il contenuto della sotto struttura dopo l'aggiunta della pagina "Content part 3" associata al Template di Pagina di tipo "testo". Gli elementi obbligatori non possono essere rimossi dall'autore e devono essere parte della lezione. Quelli facoltativi possono essere eliminati mentre possono essere aggiunti nuovi elementi ai template dagli autori. I Template di Struttura sono basati su file manifest, usati per definire l'organizzazione della lezione. Inizialmente, è stato preso in considerazione un insieme di elementi comuni a tutti i modelli di LO di seguito riproposto:

- **introduzione**

descrive in modo generale contenuti, finalità, obiettivi, attività e modalità di verifica dell'apprendimento;

- **obiettivi**

risultati di apprendimento che ci si aspetta di ottenere al termine di un percorso formativo;

- **contenuti e risorse**

tipologie di contenuti didattici (audio, testo, video etc) utilizzabili in funzione degli obiettivi e delle attività previste.

- **attività**

attività che dovrebbero essere condotte dagli studenti (in modo indivi-

duale, in coppia, in gruppo) prima, durante e al termine del percorso formativo;

- **strumenti di CMC e del Web 2.0**

strumenti utilizzati e loro ruolo all'interno sia del processo formativo;

- **strumenti per monitoraggio e valutazione**

strumenti e attività da utilizzare per il monitoraggio e la valutazione degli studenti;

- **riferimenti**

bibliografia/sitografia di riferimento/approfondimento: tutte le fonti da cui è stato tratto il materiale e i riferimenti ad eventuali materiali integrativi.

Alla luce delle nuove considerazioni, sono stati definite tre tipologie di struttura di esempio:

1. **tipologia “knowledge based”**

ha come finalità quella di seguire un percorso di auto apprendimento per acquisire le conoscenze “di base” dell'argomento/tema trattato;

2. **tipologia “meta-competency based”**

privilegia le logiche di un approccio costruttivistico alla costruzione della conoscenza. Si muove nella prospettiva della meta-cognizione e di uno sviluppo delle conoscenze e competenze che promuove in modo sistematico l'uso da parte del singolo e del gruppo di strumenti di indagine diretta (atteggiamenti, metodi, tecniche) che aprano alla possibilità della concettualizzazione, della generalizzazione, della trasferibilità dei saperi prodotti e delle modalità utilizzate per produrli;

3. **tipologia “creative based”**

centrata sulle motivazioni ed “emozioni” del soggetto che apprende. Questo tipo di LO persegue l'attivazione di competenze non facilmente misurabili con procedure docimologiche oggettivanti proprio perché

scarsamente predefinibili e fortemente connesse con la sfera dell'individualità. Tra queste, la capacità di decentramento culturale, la disponibilità ad assumere punti di vista differenti, di attivare le forme del cosiddetto pensiero creativo ecc. Le modalità didattiche privilegiate riprendono il patrimonio formativo dell'animazione culturale: consistono in strategie, anche provocatorie, di stimolazione nello studente di riflessioni che vanno oltre (precedono, accompagnano, seguono) il piano della competenza oggettiva e dell'abilità professionale per toccare la sfera del significato personale assunto da "quel" sapere per il soggetto apprendente. Tali riflessioni costituiscono comunque un quadro di competenze determinante in quanto pre-condizionano in modo anche inconsapevole l'atteggiamento dello studente nei confronti dell'apprendimento e contribuiscono a definirne la qualità effettiva.

2.2 Seconda parte: AContent Live Content Link

AContent non è sempre stato un Authoring Tool autonomo: fino al 2010 era integrato nel LCMS ATutor. Successivamente, l'intento di separare la parte di redazione dei contenuti dalla parte di gestione e fruizione ha fatto in modo che AContent diventasse un sistema a sé stante e che ATutor continuasse a "contenerlo" come modulo per ragioni di retro compatibilità. In questo contesto si colloca la seconda parte del progetto, denominata *AContent Live Content Link* che andrà ad intervenire sul LCMS ATutor. Lo scopo è quello di estendere le funzionalità di integrazione dei contenuti del sistema così da presentare contenuti esterni archiviati in AContent. Sarà trattata l'integrazione di LTI all'interno di ATutor come Tool Consumer e all'interno di AContent come Tool Provider. Successivamente verranno modificati i due moduli *AContent Repository* e *External Tool* di ATutor che, attraverso il canale di comunicazione LTI stabilito, saranno in grado di creare dei *Live Content Link* ovvero riferimenti a contenuti remoti (esterni alla piattaforma

utilizzata) che descriveremo con maggiore dettaglio nel corso del capitolo. Come detto, la piattaforma di e-learning utilizzata è ATutor[59], sistema open source e indipendente dalla piattaforma, creato nel 2002 e spicca tra le altre piattaforme analizzate come una fra le più accessibili.

2.2.1 ATutor

ATutor, con particolare riferimento alla versione usata nel progetto, la 2.1, al momento non ancora ufficialmente rilasciata, è un LMS Open Source basato su web e usato per gestire e fornire lezioni on-line. Sviluppato e curato da Adaptive Technology Resource Centre (ATRC) presso l'Università di Toronto, si tratta di un sistema modulare estendibile nelle sue funzionalità tramite moduli o nuovi temi personalizzati per cambiarne l'aspetto estetico. I titolari delle lezioni possono assemblare, "impacchettare", ridistribuire i loro contenuti. Successivamente, possono importare pacchetti interi di lezioni al fine di tenere le proprie lezioni in modo completo, esteso e distribuito. Gli studenti hanno, invece, la possibilità di apprendere in un ambiente sociale *accessibile* e *adattabile* alle diverse esigenze. Proprio su queste due caratteristiche è stato creato ATutor: l'accessibilità dei contenuti è stata, sin dall'inizio, una priorità per fare in modo che tutte le tipologie di utenti possano avere accesso al sistema senza vincoli di tecnologia. Tutti gli utenti con disabilità, che utilizzano tecnologie assistive, con versioni datate di browser, con una connessione internet lenta, tramite *Personal Data Assistant* (PDA) o tramite telefono cellulare possono accedere ad ATutor e prendere parte pienamente alle diverse attività di insegnamento.

ATutor utilizza l'editor HTML TinyMCE, accessibile ad una vasta gamma browser e tecnologie assistive e che spinge gli autori a produrre contenuti accessibili. ATutor è usato in tutto il mondo e tradotto in più di trenta lingue diverse. È stato adattato e svolge un ruolo fondamentale per università, college, scuole primarie e secondarie, occupazione e formazione, ambito governativo etc.

2.2.2 AContent come Tool Provider

Il ruolo ricoperto da AContent nella seconda parte del progetto è quello di Tool Provider. Il suo compito è di provvedere, inizialmente, ad una fase di autenticazione del TC tramite il protocollo OAuth 1.0a. All'interno del sistema considerato sono già implementate delle *OAuth API* che permettono al Web Service del TC di accedere a risorse protette senza richiedere agli utenti di inserire le credenziali relative al TP. Le specifiche relative alla chiamata ai metodi e ai parametri da passare sono viste in dettaglio nel capitolo relativo all'implementazione. Per il momento si assuma che, a seguito dell'autenticazione, il TC sia provvisto di una sessione per accedere ai contenuti del TP tramite gli strumenti LTI. A questo punto, ATutor potrà inoltrare richieste ad AContent sotto forma di messaggi inviati con il metodo POST e strutturati secondo le specifiche LTI; da ricordare che esistono campi obbligatori, consigliati o opzionali. Viene suggerito, tuttavia, di compilare e trasmettere più dati possibili al TP in modo da favorire una maggiore completezza della risposta. Nella sezione del sito web relativa allo sviluppo[60], IMS GLOBAL ha messo a disposizione un TC e un TP per testarne la funzionalità. L'immagine 2.6 rappresenta, infatti, la risposta di AContent ad ATutor. Si tratta di una struttura dati completa di variabili e rispettivi valori. Sono, inoltre, restituiti valori non espressamente richiesti dal TC ma che il TP ha accodato al messaggio di ritorno per descrivere con maggiore dettaglio le informazioni da fornire. Al verificarsi di un errore (mancata autenticazione, dati non trovati etc) viene generato l'errore "Could not establish context" che dovrà poi essere gestito dal TC.

```

IMS LTI 1.1 PHP Provider
This is a very simple reference implementation of the Tool side (i.e. provider) for IMS LTI 1.1.
Return to tool consumer (with a message or with an error)
Note: This launch can submit a grade back to the LMS using LTI 1.1 Outcome Service. Press here to send a grade back via LTI Outcome Service.
Context Information:
isInstructor() = true
getConsumerKey() = 12345
getUserKey() = 292832126
getUserKey() = 12345:292832126
getUserID() =
getUserEmail() = user@school.edu
getUserShortName() = user@school.edu
getUserName() = Jane Q. Public
getUserImage() = https://www.gravatar.com/avatar.php?gravatar_id=447512ec60136862fa6f52382f0ffc23&size=40
getResourceKey() = 12345:120988f929-274612
getResourceID() =
getResourceTitle() = Weekly Blog
getCourseName() = S1182
getCourseKey() = 12345:456434513
getCourseID() =
getOutcomeSourceID() = feb-123-456-2929:128883
getOutcomeService() = http://www.imsglobal.org/developers/LTI/test/v1pl/common/tool_consumer_outcome.php?b64=MTIzNDU0jzFNyEXQ=
Base String:
POST&http%3A%2F%2Fwww.imsglobal.org%2Fdevelopers%2FLTI%2Ftest%2Fv1p1%2Ftool.php&context_id%3D456434513%26context_label%3DS1182%26context_title%
US%26iaunch_presentation_return_url%3Dhttp%253A%252F%252Fwww.imsglobal.org%252Fdevelopers%252FLTI%252Ftest%252Fv1p1%252Fims_return.php%26iis_ou
123-456-2929%253A%253A28883%26iti_message_type%3Dbasic-lti-launch-request%26iti_version%3DLTI-
1p0%26oauth_callback%3Dabout%253Ablank%26oauth_consumer_key%3D12345%26oauth_nonce%3Db813941b9d1ef0b03d8c3601fda1687e%26oauth_signature_meth
SHA1%26oauth_timestamp%3D1348415962%26oauth_version%3D1.0%26resource_link_description%3DA%2520weekly%2520blog%26resource_link_id%3D120988f929
274612%26resource_link_title%3DWeekly%2520Blog%26roles%3DInstructor%26tool_consumer_info_product_family_code%3Dims%26tool_consumer_info_version%3D1.
Raw POST Parameters:
context_id=456434513 (ASCII)
context_label=S1182 (ASCII)

```

Figura 2.6: Implementazione di un Tool Provider di IMS LTI 1.1

2.2.3 ATutor come Tool Consumer

ATutor, nelle vesti di Tool Consumer, assumendo sia già autenticato tramite OAuth, invia al TP un form HTML propriamente compilato. Si ricordi nuovamente che LTI stabilisce campi obbligatori, consigliati o opzionali. L'immagine 2.7 rappresenta il modulo da spedire. In una comunicazione standard, l'inoltro avviene automaticamente tramite Javascript. Le variabili sono disposte in modo che, vicino ad ogni etichetta, sia possibile personalizzare il contenuto di quella di interesse. A questo punto, ATutor si pone in uno stato di ricezione, in attesa dei dati di ritorno. La Figura 2.6 mostra una schermata con i dati restituiti. In una comunicazione LTI, le fasi di autenticazione, le eventuali conferme e l'invio e ricezione dati sono trasparenti all'autore e non chiedono alcuna interazione. Come descritto in seguito, il comando "Importa contenuto" genera una chiamata ad una catena di eventi (autenticazione, invio, ricezione) che termina con la restituzione di un errore o con un messaggio di corretta importazione del contenuto selezionato.

IMS LTI 1.1 Consumer Launch

This is a very simple reference implementation of the LMS side (i.e. consumer) for IMS LTI 1.1.

[Toggle Resource and Launch Data](#)

LTI Resource

Launch URL:

Key:

Secret:

Launch Data

resource_link_id:

resource_link_title:

resource_link_description:

user_id:

roles:

lis_person_name_full:

lis_person_name_family:

lis_person_name_given:

lis_person_contact_email_primary:

lis_person_sourcedid:

context_id:

context_title:

context_label:

tool_consumer_info_product_family_code:

tool_consumer_info_version:

tool_consumer_instance_guid:

tool_consumer_instance_description:

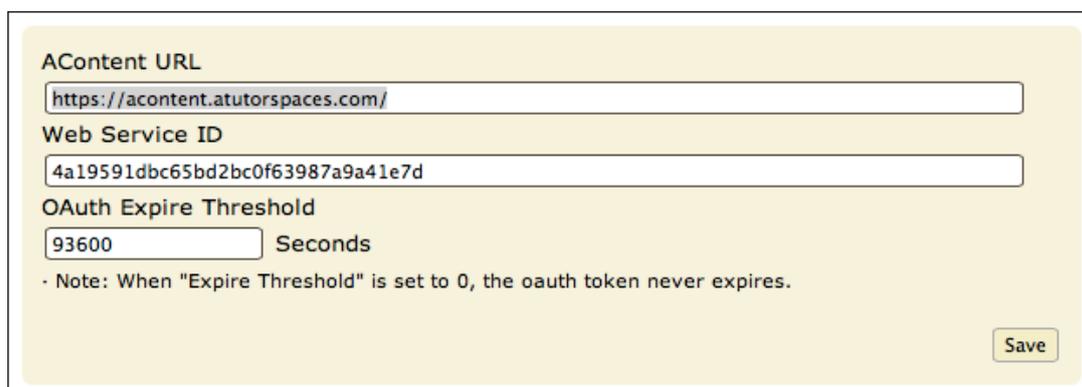
launch_presentation_locale:

Figura 2.7: Implementazione di un Tool Consumer (lato LMS) di IMS LTI 1.1

Moduli di ATutor: AContent Repository

AContent Repository è un modulo di ATutor che permette di importare contenuti remoti da AContent. Il funzionamento è basato su REST Web services e richiede una configurazione espressa nei seguenti punti:

- dalle preferenze di sistema di ATutor, l'amministratore imposta l' "URL" del repository sorgente;
- il "Web Service ID", secondo parametro richiesto, è l'identificatore univoco che contraddistingue ogni installazione di AContent;
- "OAuth Expire Threshold", terzo ed ultimo parametro, indica la durata in secondi della validità di una sessione OAuth. Impostando quest'ultimo valore a 0 si specifica una durata illimitata.



AContent URL

Web Service ID

OAuth Expire Threshold
 Seconds

· Note: When "Expire Threshold" is set to 0, the oauth token never expires.

Figura 2.8: Configurazione dei parametri di connessione ad AContent per il modulo “AContent Repository”

Una volta configurate le impostazioni di connessione ad AContent per il modulo “AContent Repository”, ogni autore dispone di un’opzione nella pagina di modifica dei contenuti.



Edit Content

[Content](#) [Create](#) [Arrange](#) [Import/Export](#) [AContent](#) [Usage](#) [Search OpenLearn](#)

Content Properties Glossary Terms Adapted Content Tests and Surveys

Close after saving

Figura 2.9: Pagina di modifica dei contenuti con l’opzione “AContent”

L’opzione del menu nell’immagine precedente conduce ad un’altra schermata con un motore di ricerca interno ad ATutor ma per contenuti remoti. Infatti, l’avvio di una ricerca tramite il modulo prevede l’invio di una richiesta REST e la restituzione della lista delle lezioni di AContent corrispondenti ai criteri inseriti nella casella di ricerca.



Enter keywords to search AContent repository.

Keywords

Figura 2.10: Motore di ricerca di ATutor per le lezioni remote di AContent

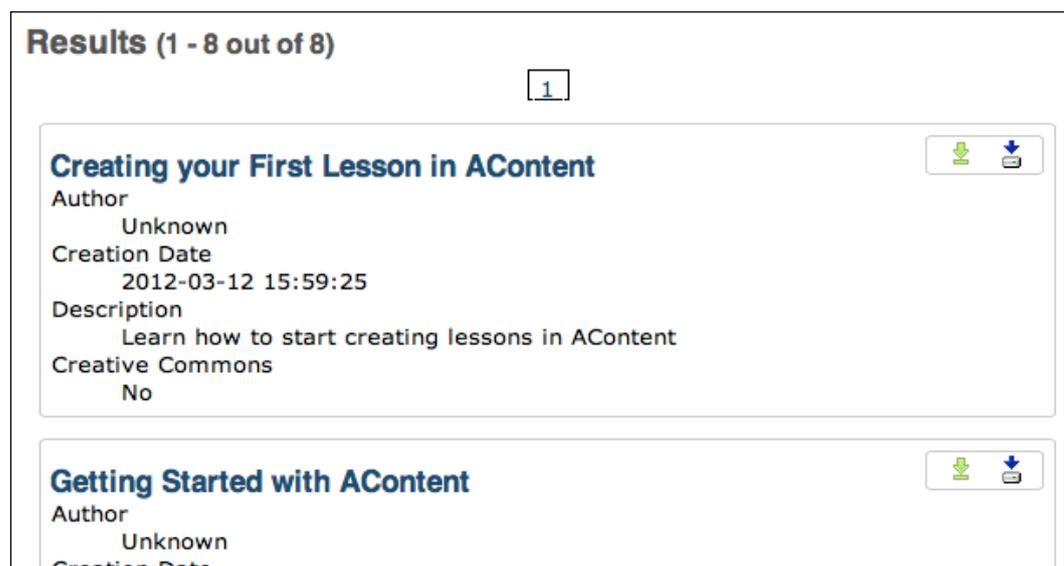


Figura 2.11: Lista dei risultati di una ricerca tramite il modulo “AContent Repository”

Oltre alla lista delle lezioni, l’utente può visualizzare un’anteprima per ogni risultato cliccando sul titolo della lezione. Nella Figura 2.11, alla destra di ogni risultato, compaiono due icone che permettono di scaricare il content package e di importare il contenuto selezionato, rispettivamente. Quello appena descritto è il funzionamento del modulo prima delle modifiche previste dal progetto di tesi. Secondo le nuove specifiche, oltre a poter scaricare o importare fisicamente i contenuti, l’insegnante può creare un *Live Content Link* (LCL) ovvero una sorta di collegamento fra ATutor e AContent mostrando i contenuti stessi come riferimenti. La soluzione adottata fa in modo che gli aggiornamenti ai contenuti di AContent si propaghino attraverso tutte le lezioni in ATutor con questi riferimenti. Per ogni risultato l’autore può scegliere la lezione da collegare. L’idea è quella di aggiungere una terza icona che guida l’autore nella creazione di un LCL. Quest’ultima opzione è visibile solo se entrambi gli strumenti di e-learning (in questo caso AContent e ATutor) supportano LTI. Cliccando su questa nuova icona dal titolo “Importa contenuti da AContent”, sarà chiesto all’autore di selezionare a

quale elemento accodare il contenuto di AContent in relazione alla lezione di ATutor che si sta navigando. Viene di seguito raffigurato il menu a tendina con la struttura della lezione di ATutor. Lo stesso menu viene mostrato per l'importazione semplice dei contenuti (cioè senza LCL) ma con un diverso indirizzo di destinazione del form.

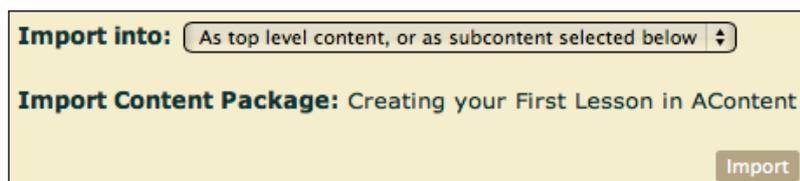


Figura 2.12: Menu a tendina con la struttura della lezione corrente di ATutor

Cliccando sul bottone “Importa” viene avviata la procedura che collega, sotto forma di LCL, l’intera struttura della lezione selezionata all’interno di ATutor. La pagina visualizzata sarà ricaricata e nel menu laterale saranno mostrati i nuovi contenuti importati. Infine, nel momento in cui un autore di AContent apporta una modifica al contenuto “collegato”, gli aggiornamenti si propagano automaticamente in ATutor.

Moduli di ATutor: External Tool

External Tool è un modulo di ATutor che permette la gestione di contenuti importati da AContent. Si tratta di un primo tentativo concreto di integrazione di Basic LTI. L’integrazione coinvolge i ruoli di amministratore e insegnante. Entrambi possono, tramite il pannello di amministrazione, inserire un nuovo elemento compilando una scheda con i seguenti campi:

- title: nome identificativo del tool che si sta creando;
- toolID: codice ID univoco;
- description: descrizione del tool e di come dovrebbe essere usato;
- tool Launch URL: indirizzo URL della risorsa remota;

- tool key (oauth_consumer_key): chiave pubblica per l'autenticazione OAuth;
- tool secret: chiave segreta per l'autenticazione OAuth;
- varie: impostazioni varie opzionali per mostrare o nascondere alcuni dati.

Dopo aver inserito almeno un tool nella lista, l'insegnante può, in fase di modifica dei contenuti, associare un tool ad una pagina. Cliccando sull'icona "External Tool" mostrata in Figura 2.13, si aprirà una finestra popup con la lista di tool presenti e alcune impostazioni circa la visualizzazione dei contenuti remoti (Figura 2.14).

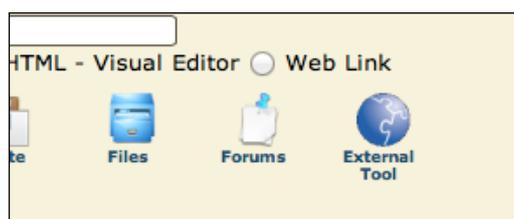


Figura 2.13: Icona External Tool

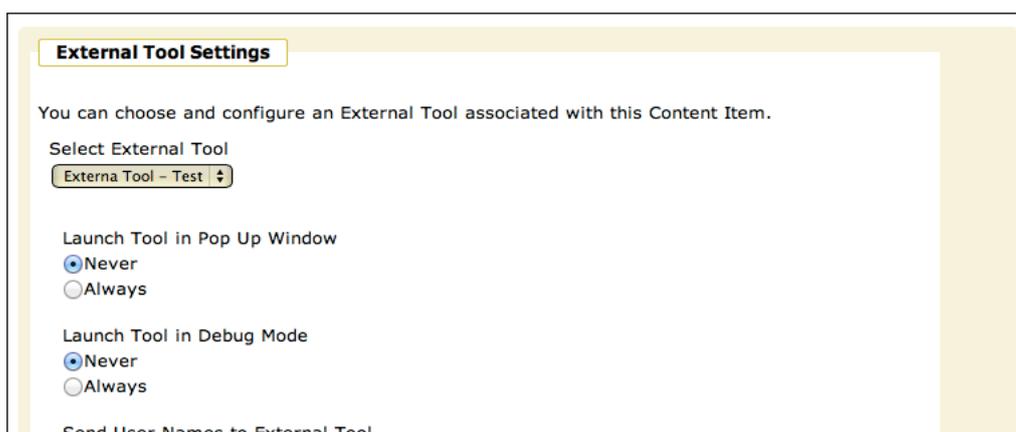


Figura 2.14: Integrazione in una pagina di un contenuto remoto tramite il modulo External Tool

L'integrazione di materiale didattico tramite il modulo External Tool è mostrato tramite un iframe (Figura 2.15) il cui sorgente fa riferimento alla pagina di AContent con la lezione specificata nell'External Tool.

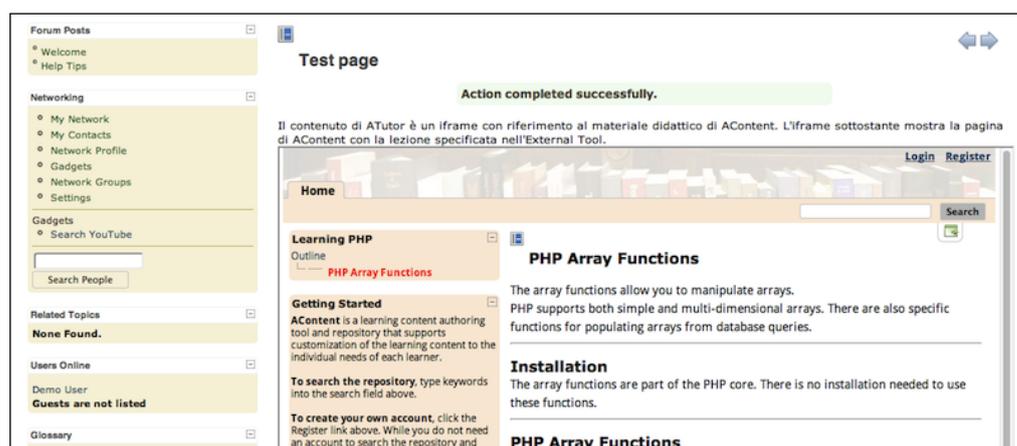


Figura 2.15: Integrazione di contenuto didattico remoto tramite il modulo External Tool

Il primo intervento previsto consiste nell'aggiornamento di BasicLTI all'ultima versione al momento disponibile: LTI 1.1. Data la retrocompatibilità dello standard, non dovrebbero verificarsi errori di funzionamento dopo tale modifica. Si vuole creare, come per il modulo "tile_search" descritto in precedenza, un motore di ricerca interno ad ATutor per contenuti remoti. Anche in questo caso viene inoltrata una richiesta tramite REST al repository di AContent il cui indirizzo è definito nella configurazione dell'External Tool. Viene restituita la lista delle lezioni corrispondenti ai criteri inseriti nella casella di ricerca; da qui, l'insegnante può selezionare una lezione e visualizzarne la struttura ad albero di tutte le lezioni che ne fanno parte. Può scegliere, infine, una o più lezioni (anche tutte o nessuna) da integrare in ATutor. L'intento è quello di offrire un dettaglio maggiore nella scelta dei contenuti da "importare" tramite la ricerca di una specifica pagina e la selezione del solo materiale didattico di interesse. Mentre la ricerca delle lezioni avviene, come già detto, attraverso il protocollo REST, la creazione dei *live content link* avviene interamente attraverso LTI 1.1. Al momento della realizzazione del

progetto, la versione 2.0 di LTI è ancora un *draft*, ovvero, in fase di revisione; nonostante questo, IMS Global ha fornito una copia riservata del documento affinché il modulo External Tool potesse essere preventivamente predisposto nel progetto di tesi anche per connessioni LTI 2.0.

Capitolo 3

Implementazione: i Template in AContent

In questo capitolo verrà condotta un'analisi dell'implementazione del progetto di Template di Layout, Template di Pagina e Template di Struttura. Come precedentemente detto, si vuole mantenere la modularità e la flessibilità delle componenti. Questo obiettivo è raggiungibile aggiungendo parti di codice indipendenti dal funzionamento complessivo del sistema. Le modifiche prevedono l'interazione con il *core* di AContent tramite la lettura di variabili globali, costanti e parametri di configurazione ma non richiedono alcun cambiamento del suo codice sorgente restante. L'authoring tool è scritto interamente in PHP[61], linguaggio di programmazione interpretato con licenza open source per applicazioni web lato server, script da terminale e stand alone con GUI. Sono integrati, inoltre, script lato client come Javascript e AJAX per richieste asincrone tramite JQuery[62], libreria molto diffusa di funzioni Javascript cross-browser. Lo stile delle pagine è definito in CSS mentre il database sul quale il sistema si appoggia è MySQL[63], RDBMS multi piattaforma con licenza GNU GPL[64] e licenza commerciale. Gli LO generati dopo l'applicazione di Template di Layout, Template di Pagina e Template di Struttura sono strutturati secondo le specifiche degli standard Common Cartridge e Content Package e mantengono, anche dopo

l'importazione in Moodle, i template applicati. Da notare che nel codice del progetto i Template di Layout sono chiamati *temi* (o themes) mentre i Template di Pagina sono chiamati *modelli* (o models). Infine per “lezione” (nel testo è considerata sinonimo di “corso”) si intende l’insieme di “pagine” (nel testo è considerata sinonimo di “contenuto”) che l’utente può visualizzare singolarmente alla destra del menu.

Utenza

Si illustrano, di seguito, i due tipi di utenze previsti per l’utilizzo delle nuove funzionalità nel contesto dei Template di Layout e dei Template di Pagina. Si assumano, per chiarezza espositiva, disgiunti, i ruoli di ogni utente (Amministratore non può essere anche Autore e vice versa).

- **Utente non autenticato**

un qualsiasi utente non autenticato è autorizzato a fruire dei contenuti a disposizione;

- **Autore**

l’autore, in modalità di anteprima dei contenuti, avrà a disposizione tre menu laterali, Template di Layout, Template di Pagina e Template di Struttura;

- **Amministratore**

l’amministratore avrà nascoste le opzioni di utilizzo; saranno visibili, solamente, i titoli dei blocchi aggiunti così da indicare l’abilitazione delle nuove funzionalità. Per quanto riguarda i Template di Layout potrà, dalla pagina di impostazioni, abilitare o disabilitare l’opzione per permettere agli autori di applicare il layout scelto alle singole pagine di una lezione.

3.1 Impatto sul sistema esistente

Per avere la possibilità di memorizzare, per ogni pagina, un Template di Layout, è stato necessario intervenire sul database MySQL aggiungendo il campo *theme* alla tabella *AC_content* al cui interno sono memorizzate informazioni come titolo, testo, intestazione, id della pagina etc. Nella fase di applicazione di un tema ad un particolare contenuto, sarà compilato il campo inserendo il nome univoco del tema applicato. Al termine di questa modifica, il sistema di Template di Layout provvederà ad includere automaticamente il codice css del tema applicato.

Anche per i Template di Struttura è stato aggiunto un campo nel database di AContent; più precisamente è stato creato il campo *structure* nella tabella *AC_content* per memorizzare il nome della struttura usata nella pagina.

Per tutti e tre i tipi di template, i file di configurazione sono salvati in locale all'interno di AContent. La scelta è dovuta principalmente alle specifiche imposte che richiedono di incidere il meno possibile sulla base di dati del sistema. Inoltre, essendo AContent un software monolitico e non modulare, nello svolgimento del progetto si è cercato di mantenere separato il codice inserito dal core di AContent. A questo proposito è stata creata un'unica cartella *dnd_themod* per tutte le librerie utilizzate.

Le prime modifiche apportate alla struttura fisica dei dati e comuni a tutti e tre i tipi di template creati mirano all'integrazione delle nuove caratteristiche al sistema AContent; si tratta di visualizzare due nuovi blocchi nel menu laterale sinistro. Per fare questo, sono state aggiunte alcune righe di codice al file */include/side_menu.inc.php* di cui riportiamo il frammento modificato:

```
1 // se l'utente non è autenticato
2 if (!isset($_SESSION['user_id']) || $_SESSION['user_id'] == 0) {
3 // inclusioni di sistema
4 $side_menu [] = TR_INCLUDE_PATH.'sidemenus/getting_start.inc.
5     php';
6 $side_menu [] = TR_INCLUDE_PATH.'sidemenus/category.inc.php';
7 } else {
```

```

7 // nel caso di un utente autenticato

9 // inclusioni di sistema
$side_menu [] = TR.INCLUDE_PATH.'sidemenus/my_courses.inc.php';
11 $side_menu [] = TR.INCLUDE_PATH.'sidemenus/category.inc.php';

13 // mostro il blocco laterale Template di Layout
$side_menu [] = TR.INCLUDE_PATH.'sidemenus/themes.inc.php';
15 // mostro il blocco laterale Template di Pagina
$side_menu [] = TR.INCLUDE_PATH.'sidemenus/models.inc.php';
17 // mostro il blocco laterale Template di Struttura
$side_menu [] = TR.INCLUDE_PATH.'sidemenus/structures.inc.php';
19 }

```

Listato 3.1: Inclusione Template nel menu laterale

La condizione *if...else* della riga 1 identifica se l'utente è anonimo (non autenticato) o autenticato (autore, amministratore etc.). Il vettore `$side_menu[]` contiene la lista di tutti i blocchi da mostrare mentre le righe 14, 16 e 18 inseriscono nell'array i nuovi file da caricare per la visualizzazione di modelli, temi e strutture nel menu laterale sinistro. In questo caso, la visualizzazione avverrà solamente nel caso in cui l'utente sia autenticato. In caso contrario, il pannello nel menu laterale non sarà neppure caricato da AContent, quindi, non visualizzato.

Dopo aver impostato questi riferimenti, sono stati creati i documenti *themes.inc.php*, *models.inc.php* e *structures.inc.php* nella cartella `/include/sidemenus/`. All'interno di ognuno, vengono dichiarate e definite variabili, istanziate classi e chiamati metodi. Le query al database sono inviate tramite classi e metodi *Data Access Object*[65] esistenti in AContent senza alcuna modifica aggiuntiva.

Infine, per dare uniformità e integrità al lavoro svolto, è stata aggiunta la cartella *dnd.themod* alla radice del LCMS al cui interno sono presenti le seguenti cartelle (analizzate in dettaglio nei paragrafi che seguono):

- **models**

contiene tutti i modelli creati e caricati;

- **structures**

contiene tutte le strutture create e caricate;

- **system**

contiene codice e immagini necessari al funzionamento di modelli e temi;

- **themes**

contiene tutti i temi creati e caricati.

3.2 Template di Layout

I Template di Layout, chiamati anche temi, in questo contesto, basano il loro funzionamento sulla classe *Themes.class.php* contenuta nella cartella */dnd_themod/system/* di cui sono elencati e descritti di seguito le principali funzioni.

Creazione dell'oggetto *Themes.class.php*

Costruttore della classe che ricava le variabili *content_id* e *course_id*, ovvero, l'identificatore del contenuto e quello della lezione, rispettivamente. Il metodo gestisce eventuali richieste POST relative alle nuove funzioni aggiunte inoltrandole ai metodi *applyThemeCourse()* e *applyThemeLesson()*. Infine, legge dal file di configurazione *config.ini* una variabile che determina se abilitare o no la possibilità per l'autore di applicare il tema alla singola pagina in quanto l'opzione di default prevede l'applicazione del tema scelto all'intera lezione. È stato scelto di memorizzare tale variabile in un file di configurazione per evitare di intervenire sul database MySQL modificando la base di dati. Si ricordi che nelle specifiche è richiesto di modificare il meno possibile la base di dati e il core del sistema.

Acquisizione dei temi esistenti

Il metodo *getThemesList()* legge tutti i temi fisicamente copiati nella cartella */dnd_themod/themes/*, li inserisce in un array e li passa come parametro al metodo *validateTheme()*. L'elemento restituito sarà il valore di ritorno del metodo *getThemesList()*.

Validazione di un tema

Il metodo *validateTheme(\$dir = array())* determina la validità di ogni tema contenuto nella cartella */dnd_themod/themes/* e verifica che il tema sia una cartella all'interno della quale deve esistere *theme.info*, un documento strutturato come un file di configurazione con i seguenti campi:

- **core**: versione di AContent. Il valore è obbligatorio e consente al sistema di determinare la compatibilità fra le differenti versioni di AContent e di temi realizzati. Nel caso il controllo della versione non venga superato, il template viene automaticamente scartato dal sistema e il controllo degli altri campi non viene eseguito;
- **name**: indica il nome pubblico del tema ovvero il nome visibile all'utente nella scelta del tema da inserire. Se non viene specificato, il sistema userà automaticamente lo stesso nome della cartella del tema;
- **description**: breve descrizione del tema per fornire all'autore un'idea degli adattamenti grafici che verranno apportati;
- **author**: nome dell'autore del tema.

I temi ritenuti validi e conformi ai vincoli imposti sono inseriti in un array, restituito dal metodo. La variabile in input *\$dir = array()* contiene la lista dei temi caricati nella cartella e che devono essere valutati.

Creazione dell'interfaccia per l'utilizzo dei temi

Il metodo `createUI($themesList)` provvede alla generazione di un form per mostrare graficamente all'utente la lista di temi (passati come parametro) caricati e validati che l'autore può applicare al contenuto. Il valore di ritorno consiste in una stringa XHTML che rappresenta il form, integrato successivamente all'interfaccia del modulo.

L'elenco dei temi viene offerto all'autore tramite un menu laterale con le seguenti caratteristiche:

- un menu a tendina con la lista dei nomi dei temi disponibili;
- la screenshot che rappresenta il tema scelto. Quando l'autore cambia opzione dal menu a tendina, l'immagine viene ricaricata in modo dinamico sulla base della nuova selezione;
- bottone “Apply the layout to the lesson” che permette di applicare il tema scelto alla lezione visualizzata;
- bottone “Apply the layout to the page” che permette di applicare il tema scelto alla lezione che l'autore sta visualizzando.

La Figura 3.1 mostra il menu laterale dal quale è possibile scegliere un Template di Layout dal menu a tendina (nell'esempio è stato creato appositamente un template dal nome “Unibo”) ed applicarlo alla singola lezione o all'intera lezione.

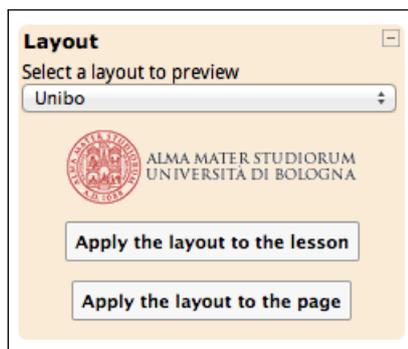


Figura 3.1: Pannello per la selezione e l'applicazione di un Template di Layout

Applicazione del tema alla lezione

Il metodo *applyThemeCourse()* svolge due funzioni fondamentali. Per ogni lezione appartenente a quella specifica lezione:

1. avvolge il contenuto fra i tag `<div id="content-text">` e `</div>` così da isolarlo dal resto della pagina web. Lo aggiorna, poi, nel campo *text* del database, precisamente, nella tabella *AC_content* nella quale sono memorizzate tutte le informazioni relative ai contenuti come id, titolo, testo etc. Modificando lo stile CSS della lezione senza delimitare il contenuto tramite tag, si viene a modificare l'intera pagina web di AContent e, in generale, di ogni sistema che importa il pacchetto dati. Questa prima operazione vuole definire, quindi, quella netta separazione necessaria all'applicazione del tema al contenuto desiderato anziché all'intero sistema di authoring;
2. aggiorna il campo *theme* del database MySQL, situato nella tabella *AC_content*, con il nome del tema selezionato. In questo modo il sistema di Template di Layout sarà in grado di includere automaticamente il codice css del tema applicato.

Applicazione del tema alla lezione

Il metodo *applyThemeLesson()* esegue le stesse due operazioni del metodo *applyThemeCourse()* ma solo per la lezione corrente. Si è voluto creare un metodo distinto, rispetto al precedente, per dare la possibilità di integrare, in futuro, nuove caratteristiche in base alla pagina o alla lezione. Inoltre, mentre l'applicazione di un tema ad una singola lezione è una unica operazione di aggiornamento di un campo del database, quella per un'intera lezione è strutturata in modo ricorsivo così da percorrere lezioni, cartelle e sottocartelle.

3.2.1 Il file Themes.js

Alla classe *Themes.class.php* appena analizzata è stato affiancato il file *Themes.js* come supporto alle operazioni sui temi. Il Javascript contenuto fornisce all'autore la possibilità di visualizzare un'anteprima del tema selezionato prima di applicarlo alla lezione o alla lezione. Le librerie usate sono JQuery e, in questo caso, si occupano di aggiungere al tag *<head>* della pagina di AContent un nuovo elemento *link*. Appena l'autore seleziona il tema desiderato dal menu a tendina, Javascript intercetta la selezione ed esegue il seguente codice:

```
1 // se il contenuto attuale ha un tema, lo includo
$( 'head' ).append( '<link rel="stylesheet" href="' + current_theme
+ '" type="text/css" />' );
```

Listato 3.2: Inserimento dinamico del css per i Template di Layout

in cui *current_theme* è il nome del tema adottato dal contenuto che si sta visualizzando. Viene, quindi, incluso il CSS relativo al tema selezionato mostrando un'anteprima istantanea all'autore.

Un'altra caratteristica completamente gestita da JQuery è l'opzione aggiunta nel pannello di amministrazione. Tramite Javascript, infatti, è stato creato il

riquadro mostrato in Figura 3.2 nella pagina di amministrazione di AContent.



Figura 3.2: L'amministratore può abilitare gli autori ad applicare un Template di Layout alla singola lezione

Come da didascalia dell'immagine precedente, l'amministratore può abilitare o disabilitare la possibilità per gli autori di applicare un tema alla singola lezione. Il form è creato e mostrato tramite puro Javascript. La memorizzazione della preferenza dell'amministratore, invece, viene scritta nel file di configurazione *config.ini* tramite una chiamata asincrona in AJAX al file *AJAX_actions.php* in cui una funzione PHP sovrascrive un flag che indica l'abilitazione o meno dell'opzione. Il file *config.ini* evita di dover apportare modifiche al database.

3.2.2 Creare un tema

La creazione di un tema avviene come segue:

1. si crea una cartella in */dnd.themod/themes/* alla quale viene assegnato il nome del tema da installare. Quest'ultimo deve essere univoco all'interno della cartella. Sarà usata, come esempio, la cartella *tema*;
2. all'interno della nuova cartella, si crea un documento dal nome *theme.info*. Nell'esempio, il file si trova nella cartella *themes/tema/* con il seguente contenuto:

```
# Nome del tema: etichetta nel menu a tendina
name = "Tema"
# Breve descrizione del tema
description = "Il mio primo tema"
```

```
# Nome dell'autore del tema
author = "Mauro Donadio indirizzo@email.com"
# Versione di AContent con il quale il tema è compatibile
core = "1.2"
```

3. il passo successivo consiste nella creazione del foglio di stile che determinerà la veste grafica del tema creato. A questo, sarà assegnato lo stesso nome del tema ma con estensione `.css`.

Ogni classe o identificatore del documento dovrà necessariamente essere contenuta all'interno dell'identificatore `#content-text`. Per il contenuto del documento si rimanda al sito ufficiale del W3C concernente le specifiche relative ai fogli di stile.

Nell'esempio, il documento è `tema.css` nella cartella `themes/tema/` con il seguente contenuto di esempio:

```
#content-text body{
2  background-color: #FFF;
   color: #000;
4  background-image: url(tema/logo.png);
}
```

Listato 3.3: Contenuto del css di un Template di Layout

4. (opzionale) se si desidera inserire elementi grafici, ogni riferimento deve essere relativo ad una cartella con lo stesso nome del tema. Nell'esempio, la cartella sarà `tema` e conterrà l'immagine `logo.png`;
5. (opzionale) infine, nella cartella del tema appena creato, sarà inserita l'immagine `screenshot.png` che rappresenterà, nel menu dei temi, quello selezionato. Nell'esempio sarà creata l'immagine `screenshot.png` nella cartella `themes/tema/` con il logo del tema.

3.3 Template di Pagina

I Template di Pagina, chiamati anche modelli, in questo contesto, basano il loro funzionamento sulla classe *Models.class.php* contenuta nella cartella */dnd_themod/system/* di cui sono elencati e descritti di seguito i principali metodi.

Creazione dell'oggetto *Models.class.php*

Costruttore della classe che ricava le variabili *content_id* e *course_id*, ovvero, l'identificatore del contenuto e quella della lezione, rispettivamente. Il metodo gestisce eventuali richieste POST relative alle nuove funzioni aggiunte inoltrandole al metodo *applyModelContent()*.

Acquisizione dei modelli esistenti

Legge tutti i modelli fisicamente salvati nella cartella */dnd_themod/models/*, li inserisce in un array e li passa come parametro al metodo *validateModel()*. L'elemento restituito sarà il valore di ritorno del metodo *getModelsList()*.

Validazione di un modello

Il metodo determina la validità di ogni modello contenuto nella cartella */dnd_themod/models/* e verifica che il modello sia una cartella all'interno della quale deve esistere *models.info*, un documento strutturato come un file di configurazione con i seguenti campi:

- **core**: versione di AContent. Il valore è obbligatorio e consente al sistema di determinare la compatibilità fra le differenti versioni di AContent e di modelli realizzati. Nel caso il controllo della versione non venga superato, il template viene automaticamente scartato dal sistema e il controllo degli altri campi non viene eseguito;
- **name**: indica il nome pubblico del modello ovvero il nome visibile all'utente nella scelta del modello da inserire. Se non viene specifica-

to, il sistema userà automaticamente lo stesso nome della cartella del modello;

- **description:** breve descrizione del modello per fornire all'autore un'idea degli adattamenti che verranno apportati alla struttura del contenuto;
- **author:** nome dell'autore del modello.

I modelli ritenuti validi e conformi ai vincoli imposti sono inseriti in un array, restituito dal metodo. La variabile in input $\$dir = array()$ contiene la lista dei modelli caricati nella cartella e che devono essere valutati.

Creazione dell'interfaccia per l'utilizzo dei modelli

Il metodo `createUI($modelsList)` provvede alla generazione di un form con due opzioni di tipo checkbox. La prima, "Activate page templates", mostra un pannello di selezione dei modelli, visibile sopra il contenuto della pagina. Una volta deselezionata, lo script provvede alla memorizzazione delle modifiche apportate dall'autore.

La seconda, "Arrange page templates" permette di ordinare o eliminare i modelli inseriti.

I modelli inseriti all'interno della pagina vengono visualizzati dall'utente come "blocchi" strutturati di contenuto così da impartire un ordine nella pagina stessa e aiutare l'autore nella comprensione delle varie sezioni della lezione. La Figura 3.3 mostra il menu laterale alla sinistra dei contenuti con le due checkbox mentre la Figura 3.4 raffigura il box a scomparsa con la lista dei modelli disponibili. Il riquadro compare una volta spuntata la casella di abilitazione dei Template di Pagina e scompare alla disattivazione della stessa. Il salvataggio effettivo delle modifiche apportate tramite questa interfaccia grafica avviene al momento della scomparsa del box

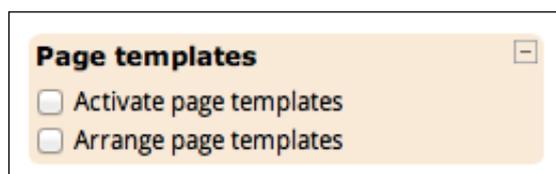


Figura 3.3: Pannello di abilitazione dei Template di Pagina

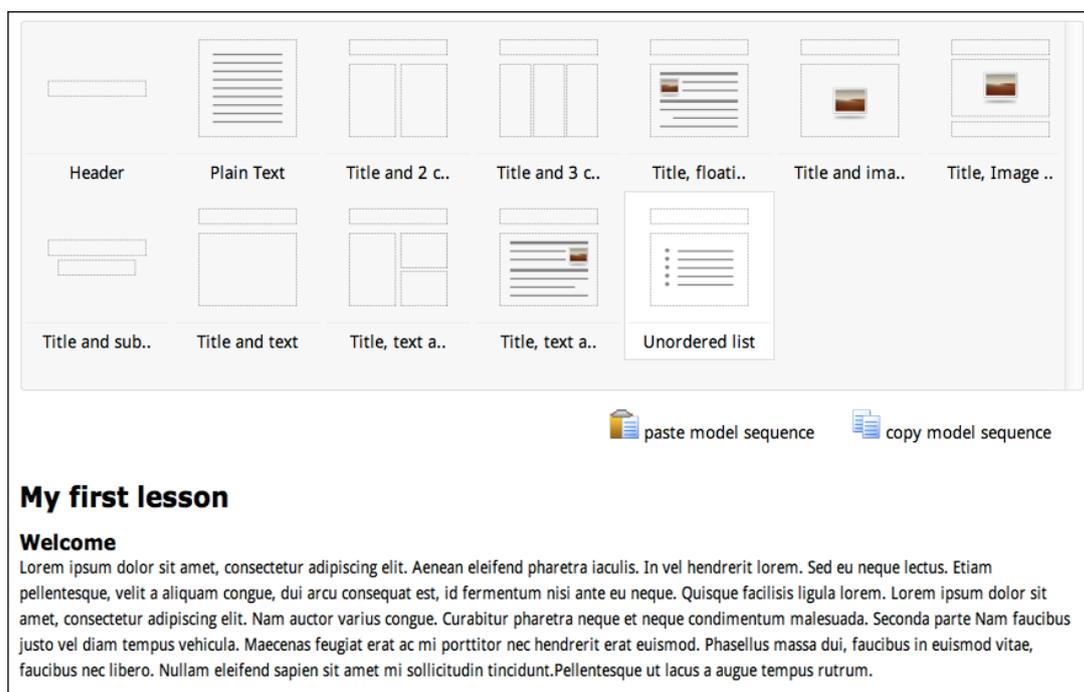


Figura 3.4: Pannello di scelta di un Template di Pagina

Applicazione del modello al contenuto

L'applicazione di un modello ad un contenuto avviene tramite l'inserimento del blocco all'interno del contenuto testuale della lezione. Viene interrogato il database circa il contenuto della lezione. Viene, poi, inserito il codice XHTML del modello in cima al testo risultante dalla query. L'innesco del codice viene fatto in Javascript e mostrato istantaneamente a video. Quanto l'autore andrà a salvare il contenuto della lezione, il nuovo conte-

nuto “arricchito” verrà salvato sul database aggiornando il campo *text* della tabella *AC_content*.

Il file *Models.js*

Alla classe *Models.class.php* appena analizzata è stato affiancato il file *Models.js* come supporto alle operazioni sui modelli. In questo caso, però, diversamente da *Themes.js*, il file Javascript risulta parte integrante del codice e fondamentale per il funzionamento dei Template di Pagina. Gestisce, cliccando sull’opzione “Activate page templates”, la visualizzazione del pannello mostrato in Figura 3.4 contenente tutti i modelli che l’autore può scegliere. Cliccando sull’icona di un modello, viene inserita la struttura HTML del modello stesso in cima al contenuto della lezione. Ogni nuovo blocco inserito può essere eliminato cliccando sulla X di chiusura posta in alto a sinistra ad ogni Template di Pagina selezionato.

Nonostante l’opzione possa sembrare ridondante perché già offerta dalla seconda checkbox, anche in questo caso è possibile ordinare i modelli inseriti tramite quattro bottoni in basso a destra di ogni blocco. È possibile spostare il blocco di una posizione alla volta verso l’alto e verso il basso oppure spostarlo in cima o in coda a tutto il contenuto. Un’ultima caratteristica molto utile fornita dai Template di Pagina e realizzata in Javascript è la “Copia della sequenza dei modelli”: dopo aver inserito e posizionato i modelli all’interno della pagina, l’autore può copiarne l’intera disposizione e tipologia per riutilizzarla in un’altra lezione. Tramite il bottone “copy model sequence” la sequenza viene copiata; tramite il bottone “paste model sequence” la sequenza viene incollata mantenendo disposizione e tipo di modello. Cliccando nuovamente su “Activate page templates” viene nascosta la schermata di selezione dei modelli e la configurazione viene salvata sul database. Anche dopo queste modifiche, l’autore può cambiare il contenuto tramite gli strumenti messi a disposizione da AContent. L’immagine 3.5 raffigura il box di scelta e un esempio di modello inserito anche in fase di modifica.

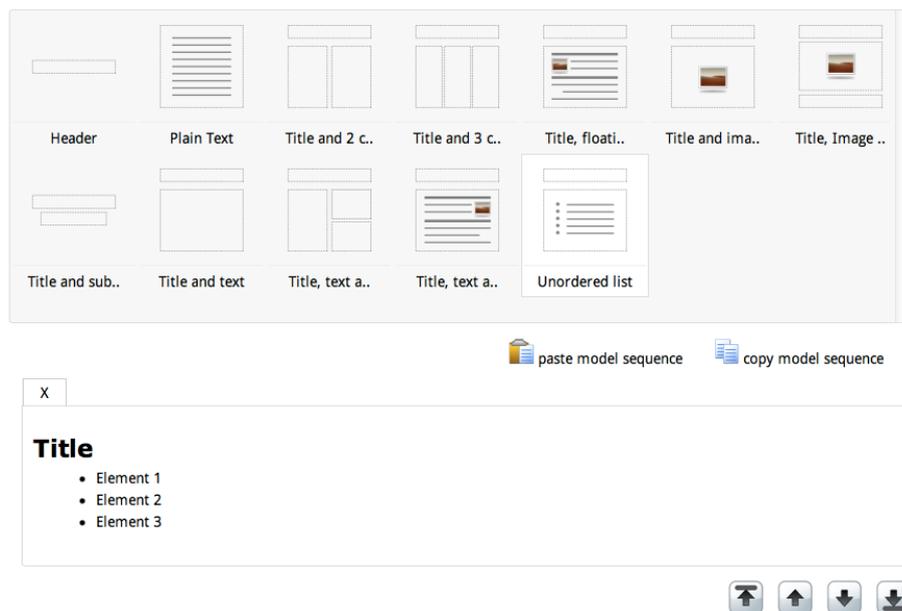


Figura 3.5: Pannello di scelta di un Template di Pagina e box di esempio

Anche in questo caso, le librerie usate sono quelle del pacchetto JQuery e, come per i temi, la versione è la stessa di quelle utilizzate per il resto del sistema AContent. Un'ultima aggiunta al codice, seppur non rilevante ai fini del progetto, viene espressa di seguito. Si immagini il seguente scenario in cui:

- l'utente posiziona il cursore del mouse su una casella di testo a più righe (per esempio una casella di tipo `<textarea>` oppure il riquadro dei modelli nei Template di Pagina);
- scorre la rotella del mouse verso il basso;
- quando l'ultimo elemento della casella sarà visualizzato, lo scorrimento proseguirà portando l'utente in fondo all'intera pagina web.

La conseguenza indotta dall'ultimo punto è stata prevista e gestita. Una volta raggiunto l'ultimo modello del riquadro, lo scorrimento del mouse verso il basso viene temporaneamente disabilitato finché il cursore non viene

spostato all'esterno del riquadro. Questo consente all'utente di mantenere l'attenzione (il focus) sul riquadro di interesse senza spostare la visualizzazione dell'intera pagina.

3.3.1 Creare un modello

Le fasi iniziali per la creazione di un modello sono molto simili a quelle della creazione di un tema:

1. si crea una cartella in `/dnd_themod/models/` alla quale viene assegnato il nome del modello da installare. Quest'ultimo deve essere univoco all'interno della cartella. Sarà usata, come esempio, la cartella *modello*;
2. all'interno della nuova cartella, si crei un documento dal nome *model.info*. Nell'esempio, il documento si trova nella cartella *models/modello/* con il seguente contenuto:

```
# Nome del modello: etichetta che l'autore visualizzerà
# insieme alla screenshot di anteprima
name = "Modello"
# Breve descrizione della struttura del modello
description = "Il mio primo modello"
# Nome dell'autore del modello
author = "Mauro Donadio indirizzo@email.com"
# Versione di AContent con il quale il modello è compatibile
core = "1.2"
```

3. il passo successivo consiste nella creazione del documento HTML che determinerà la struttura del modello creato. Anche il documento HTML avrà lo stesso nome del modello ma estensione `.htm`. Nell'esempio, il

documento è *modello.htm* nella cartella *models/modello/* con il seguente contenuto:

```
1 <div>
   <h1>Titolo del nuovo modello</h1>
3   <div>
     Contenuto testuale del nuovo modello
5   </div>
</div>
```

Listato 3.4: Contenuto del file .htm di un Template di Pagina

4. (opzionale) infine, come per i temi, nella cartella del modello appena creato, sarà inserita l'immagine *screenshot.png* che rappresenterà, nel riquadro dei modelli, quello selezionato.

Nell'esempio sarà creata l'immagine *screenshot.png* nella cartella *models/modello/* che rappresenterà la miniatura stilizzata di come sarà il modello una volta inserito nel contenuto.

3.4 Template di Struttura

L'aggiunta dei Template di Struttura ad AContent va a proporre all'autore strutture dei contenuti "semplici" o "complesse". Anche in questo caso, come per i Template di Layout, è stato necessario aggiungere il campo *structure* nella tabella *AC_content* del database di AContent per memorizzare il nome della struttura usata nella pagina.

Il file *structures.inc.php*, caricato all'apertura di AContent, si occupa di istanziare la classe *Structures.class.php* contenuta nella cartella */dnd.themod/system/* la quale presenta funzionalità di seguito elencate e descritte.

Acquisizione delle strutture esistenti

getStructsList() è il metodo che legge tutte le strutture caricate e disponibili nella cartella */dnd.themod/structures/* e, dopo averle "validate" tramite

il metodo *checkStructs()* della stessa classe, ne restituisce la lista sotto forma di array.

Validazione di una struttura

Chiamato dal metodo *getStructsList()*, *checkStructs(\$dir = array())* prende in input la lista di tutte le strutture presenti all'interno della cartella */dnd_themod/structures/* e le valuta secondo le stesse specifiche usate per Template di Layout e Template di Pagina. Restituisce, infine, la lista delle strutture considerate valide.

Il file *structures.inc.php*, dopo aver chiamato il metodo *getStructsList()* della classe *Structures.class.php*, stampa in output la lista delle strutture disponibili e valide nel menu laterale di AContent. Come da Figura 3.6, il menu sinistro dei Template di Struttura è organizzato secondo due categorie: “Simple structures” e “Complex structures” in cui le seconde sono una composizione delle prime rendendo più complessa e più completa la struttura che l'autore più adottare. Verranno trattate entrambe nello stesso approfondimento facendo riferimento con “sottostrutture” le strutture semplici e le singole strutture che vanno a comporre una struttura complessa.

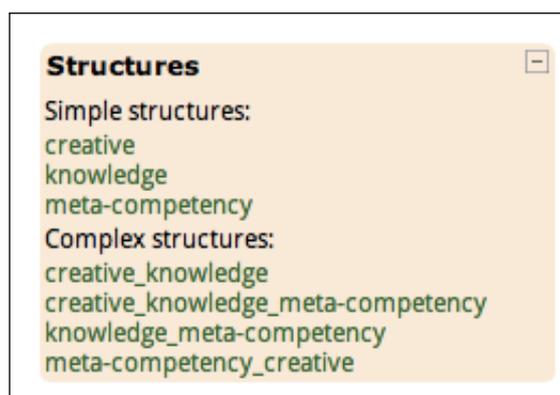


Figura 3.6: Pannello di scelta di un Template di Struttura

Cliccando su una delle opzioni a disposizione, il sistema mostrerà l'anteprima della struttura scelta indicando, con immagini, quali parti sono richie-

ste e quali opzionali (come in Figura 3.7). Per raggiungere questo risultato è stato creato il documento *outline.php* nella cartella */home/structs/* che esegue in sequenza le seguenti operazioni:

1. include la classe *StructureManager.class.php* contenuta nella cartella */home/classes/*;
2. include il file *Struct.js* contenuto nella cartella */dnd_themod/system/*;
3. per ogni sottostruttura:
 - istanzia la classe *StructureManager(\$name)*;
 - chiama il metodo *printPreview(\$flag_button, \$structs)*.

Mentre il file *Struct.js* vuole solo formattare lo stile della visualizzazione della struttura, di seguito, sarà preso in analisi il più corposo *StructureManager.class.php* e i suoi metodi principali.

Gestione delle strutture

StructureManager(\$name) è il costruttore che prende in input il nome della struttura scelta dall'autore e lo imposta come variabile globale all'interno della classe. Chiama, infine, il metodo *setInfo()* che legge e memorizza su variabili globali i valori di configurazione del file *structure.info*. Quest'ultimo è contenuto, con lo stesso nome, in ogni struttura caricata fisicamente su AContent.

Visualizza l'albero della struttura

Sulla base delle indicazioni sulla “composizione” della struttura espresse nel file di configurazione *structure.info*, il metodo *printStruct()* costruisce e stampa in output l'albero dei contenuti come mostrato nella Figura 3.7.

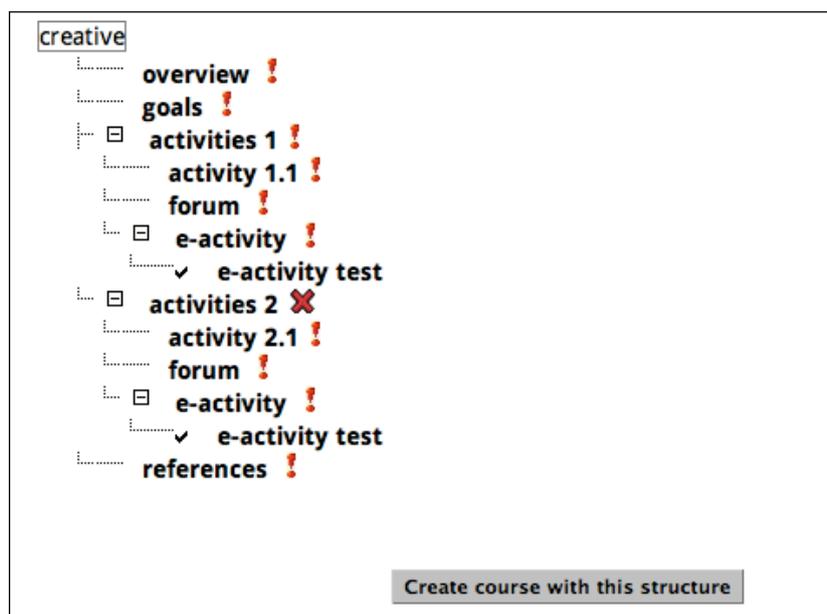


Figura 3.7: Esempio di struttura chiamata “creative based”

Nel Listato 3.5 è riportato il contenuto del file *structure.info*, preso come esempio per la creazione della struttura chiamata “creative based”. Sono raffigurate, nell’immagine, le informazioni base di un Template di Struttura tramite le quali il sistema può sottoporre il template a validazione.

```

; Nome del template visibile dall'utente
2 name = creative based

; Breve descrizione del template di struttura
4 description = Breve descrizione del template di struttura

; * Autore del template di struttura
6
8 author = Mauro Donadio indirizzo@email.com

; * versione di AContent per la quale il template è stato creato
10 core = 1.2

```

Listato 3.5: Contenuto del file *structure.info*. Listato 1

Il Listato 3.5 è composto da un secondo frammento riportato nel Listato 3.6. Il codice di esempio definisce la composizione di una pagina chiamata *overview* in cui *min* e *max* sono il minimo e massimo numero di sottoelementi (per esempio, per essere sicuri che ci siano da 2 a 5 valutazioni finali). Il valore “x” determina un numero non definito maggiore di 0.

```
1 ; page template
3 pageTemplName [] = overview
  min_overview = 1
5 max_overview = 1
7 eTivities [] = eTivity
  min_eTivity = 1
9 max_eTivity = x
```

Listato 3.6: Contenuto del file *structure.info*. Listato 2

Creazione ricorsiva di una struttura

Il metodo ricorsivo *createStruct(\$page_temp, \$id_folder, \$course_id)* che crea e memorizza sul database la struttura scelta dall'autore. Per la scrittura su MySQL, si appoggia alle classi *ContentDAO* e *CoursesDAO* messe a disposizione da AContent. La ricorsione permette di percorrere l'albero prendendo in analisi ogni singolo sotto albero fino alle foglie. Giacché l'obiettivo è la memorizzazione su database della struttura scelta, il metodo non restituisce alcun valore. Al termine dell'esecuzione, lo script indirizza l'utente alla prima pagina del contenuto appena creato mostrando, sul menu laterale di AContent, un blocco contenente la struttura ad albero della lezione. Come raffigurato in Figura 3.8, sono mantenute le diverse icone che indicano l'obbligatorietà o l'opzionalità dei singoli contenuti.



Figura 3.8: Riquadro finale di un Template di Struttura

Tramite la prima icona in alto a destra, indicata dalla freccia rossa, è possibile accodare una nuova struttura a quella appena creata andando a comporre, così, una “struttura complessa”. L’opzione è disponibile anche per contenuti non “generati” direttamente dal Template di Struttura.

3.4.1 Creare una struttura

La creazione di una struttura è volutamente simile a quella di temi e modelli così da mantenere una coerenza nei passi da seguire e per non disorientare l’autore o l’amministratore che desidera cimentarsi in questa operazione. Di seguito sono elencati i punti da seguire in sequenza per creare una nuova struttura in base alle proprie esigenze:

- si crea una cartella in `/dnd.themod/structures/` alla quale viene assegnato il nome della struttura da installare. Come per Template di Layout e Template di Pagina, quest’ultimo deve essere univoco all’interno della cartella.
Sarà usata, come esempio, la cartella *struttura*;
- all’interno della nuova cartella, si crea un documento dal nome *structure.info* contenente le informazioni di configurazione della nuova struttura e la composizione dei suoi elementi. Il contenuto è mostrato nei Listati di esempio 3.5 e 3.6.

- il passo successivo consiste nella creazione del documento *struct_pages_ass.info* che definisce i Template di Pagina adottati dagli elementi in struttura; il nome di questo file sarà lo stesso per tutte le strutture create. Per ogni elemento della struttura viene specificato un Template di Pagina sulla base del quale modellare il contenuto. Un esempio è mostrato di seguito:

```

; nome elemento struttura = template di pagina
2 overview [] = plain_text
4 activity [] = plain_text
  activity [] = title_image
6 activity [] = title_image_text
8 references [] = unordered_list
10 eTivity [] = true_or_false
  eTivity [] = multiple_choise
12 eTivity [] = matching_graphical
  eTivity [] = matching_simple
```

Listato 3.7: Contenuto del file *struct_pages_ass.info*

Il Listato 3.7 assegna all'elemento della struttura chiamato *overview* il Template di Pagina chiamato *plain_text* che corrisponde a un blocco di testo semplice. la corrispondenza è la stessa per ogni elemento della struttura e il riferimento è ad un Template di Pagina esistente e validato dal sistema AContent.

Il codice scritto gestisce l'interpretazione della configurazione e guida l'autore nella composizione di uno o più strutture "semplici" o "complesse".

Capitolo 4

Implementazione: AContent Live Content Link

La seconda parte del progetto di tesi tratta l'implementazione di LTI su ATutor e AContent con lo scopo di stabilire un canale di comunicazione sicuro fra i due sistemi. LTI si propone come un Web Service che si interfaccia ai contenuti che possono essere ospitati sullo stesso computer, su un server, in una rete cloud. Il progetto, denominato AContent Live Content Link, attribuisce ad AContent il ruolo di Tool Provider e ad ATutor quello di Tool Consumer. In questo capitolo si andrà ad analizzare l'implementazione di LTI in entrambi i sistemi e gli aggiornamenti apportati ai moduli *AContent repository* e *External Tool*, appartenenti ad ATutor. Si prevede la loro modifica affinché, attraverso il canale di comunicazione LTI stabilito, siano in grado di creare dei *Live Content Link* ovvero riferimenti a contenuti remoti (esterni alla piattaforma utilizzata) aggiornati in tempo reale. Infatti, a differenza di una normale importazione di un LO esterno è previsto che venga creato un "riferimento". In questo modo, la modifica di una pagina sul Tool Provider AContent si ripercuoterà istantaneamente su tutti i contenuti dei Tool Consumer che hanno instaurato un Live Content Link con il provider. Il funzionamento dei moduli modificati avviene solo se entrambi gli estremi della comunicazione supportano LTI. In caso contrario, saranno disponibili

solo le operazioni di base. IMS Global Consortium, creatore di LTI, ha provveduto a fornire sul sito del consorzio uno strumento web per verificare la corretta implementazione ed il corretto utilizzo degli strumenti LTI forniti. Previa autenticazione, è possibile effettuare test di conformità di Tool Provider e Tool Consumer separatamente. Il controllo è automatizzato e basato sulle specifiche fornite nella documentazione di LTI. Se superato, il sistema memorizza e pubblica il nome dell'organizzazione di cui l'utente è referente. La lista diffusa, liberamente consultabile, rappresenta l'elenco completo di strumenti che supportano pienamente LTI. Infine, ad ogni partecipante la cui valutazione ha avuto esito positivo, viene conferita una certificazione ufficiale. In questo modo è possibile mantenere un controllo sulle conformità allo standard e aiutare le organizzazioni ad effettuare test sul loro software. Esistono diverse versioni di LTI; il progetto presentato è certificato IMS Global Consortium LTI per le versioni Basic-LTI e 1.1. Al momento dello sviluppo non era ancora disponibile lo strumento di autocertificazione per la versione 2.0 in quanto il documento era ancora in fase di stesura; tuttavia, grazie alla disponibilità del consorzio è stato possibile consultare il draft e prevederne il supporto.

4.1 LTI

Essendo la comunicazione fra Tool Provider e Tool Consumer basata su LTI, entrambi devono supportare lo strumento di interoperabilità. Affinché questo sia possibile è stato necessario intervenire sui sistemi di e-learning aggiungendo librerie e file di configurazione. Così facendo, si riesce ad abilitare entrambi all'uso di LTI e, precisamente, AContent come Tool Provider e ATutor come Tool Consumer. Come da specifiche, si è cercato di modificare il meno possibile il core dei due strumenti e la struttura dei loro database. Il progetto, infatti, prevede in gran parte l'innesto di file e cartelle con una minima modifica al codice per adattarlo ai nuovi frammenti inseriti. Altri dati e informazioni e preferenze dell'utente vengono immagazzinate all'inter-

no di documenti mantenuti sulla memoria locale del sistema. I parametri per instaurare un canale di comunicazione LTI sono, invece, tratti da variabili di sistema, variabili di sessione e variabili temporanee impostate durante l'operazione dall'autore stesso. Nel testo si parlerà di *LTI Tool Provider* e di *LTI Tool Consumer* in quanto il codice differisce in base al ruolo che si desidera conferire. La modalità di comunicazione è simile a quella che avviene fra server e client nel quale l'ultimo inoltra una richiesta al server che la elabora e la restituisce al primo. Il pacchetto di librerie LTI a disposizione racchiude in sé entrambe le parti ma necessita di alcune modifiche per essere parametrizzato e configurato correttamente. È offerta, inoltre, piena retro compatibilità, garanzia di piena interoperabilità fra Tool Provider e Tool Consumer di versioni differenti. Tuttavia, non sono implementate librerie per LTI v2.0 in quanto, al momento dello svolgimento del progetto e della stesura della presente tesi, la documentazione è ancora in fase di stesura. IMS Global Consortium ha fornito ugualmente, in modo ufficioso, una copia del draft tramite la quale, basandosi sui punti cardine fino a quel momento stabiliti, è stato possibile predisporre i due sistemi di e-learning AContent e ATutor al supporto della nuova versione. Di seguito sono analizzate le modifiche apportate ai due sistemi di e-learning, modifiche necessarie ad abilitarli al supporto di LTI (BasicLTI o v1.0, v1.1 e 2.0).

4.1.1 LTI su AContent

In questo contesto, AContent è il Tool Provider che fornirà le informazioni richieste al Tool Consumer. La prima operazione per integrare LTI Tool Provider in AContent è l'aggiunta della cartella *lti/* nella directory */oauth/*. All'interno sono contenute tutte le librerie base per il tool. Nel corso dello svolgimento del progetto, il contenuto della cartella è stato arricchito da frammenti di librerie e frammenti di codice così da permettere, come richiesto dalle specifiche, il supporto di un canale LTI v.2.0, non ancora ufficialmente pubblicato. Non seguirà un'analisi della struttura di LTI ma verranno trattati due file, in particolare, che sono stati modificati per adattare lo strumento

di comunicazione alle esigenze dei due sistemi di e-learning. Le funzioni base, infatti, non soddisfano completamente i requisiti necessari alla realizzazione del progetto *AContent Live Content Link* in quanto sono richieste al Tool Provider particolari informazioni non previste in una comunicazione standard. Sono, quindi, stati modificati i file *tool.php* e *lti_util.php* contenuti nelle cartelle */oauth/lti/* e */oauth/lti/util/*, rispettivamente. Nel listato sottostante viene mostrato il primo frammento di codice in cui è definito il valore del vettore *\$parm*, passato poi al costruttore della classe *BLTI()*. Le variabili dell'array corrispondono ai parametri necessari allo scambio di chiavi tramite il protocollo OAuth e sono di seguito elencate:

- **AC_oauth_server_consumers**
nome della tabella di AContent che memorizza i dati relativi a OAuth.
Da questa tabella vanno letti:
 - **key_column**
nome della colonna per la chiave pubblica per l'autenticazione OAuth;
 - **secret_column**
nome della colonna per la chiave privata per l'autenticazione OAuth;
 - **context_column**
nome della colonna per l'indirizzo URL della repository remota.

La chiamata al metodo *dump()* restituisce l'output delle informazioni estrapolate nella richiesta LTI.

```
1 <?php
  // params to set OAuth on "database mode"
3 $parm = array( 'table' => 'AC_oauth_server_consumers',
                'key_column' => 'consumer_key',
5                'secret_column' => 'consumer_secret',
                'context_column' => 'consumer'
```

```

7     );

9 // Instantiate the BasicLTI class
  // (OAuth_db_config_array, use_session, do_redirect)
11 $context = new BLTI($parm, false, false);

13 // return LTI requested data
  echo $context->dump();
15 ?>

```

Listato 4.1: Contenuto del file “tool.php”

Il secondo frammento di codice presentato richiede un maggiore approfondimento in quanto tratta la definizione della classe *BLTI* ovvero Basic LTI che restituisce al Tool Consumer le informazioni richieste. La prima modifica apportata, mostrata nel Listato 4.2 e contenuta nel costruttore della classe, abilita il Tool Provider ad effettuare un'autenticazione OAuth basata su database anziché sulle sessioni. Lo scambio di chiavi avviene, quindi, appoggiandosi a MySQL in quanto, in questo contesto, offre maggiore sicurezza nella gestione delle chiavi.

```

1 [...]

3 } else {
  define('TR_INCLUDE_PATH', '../..//include/');
5  include_once(TR_INCLUDE_PATH.'config.inc.php');

7  mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
  mysql_select_db(DB_NAME);

9

11  $sql = 'SELECT * FROM '.$parm['table'].' WHERE ' .
        '($parm['key_column'] ? $parm['key_column'] : '
          oauth_consumer_key') .
        '=' .
13  "''".mysql_real_escape_string($oauth_consumer_key)."''";

15  $result = mysql_query($sql);

```

```

17  $num_rows = mysql_num_rows($result);
    [...]

```

Listato 4.2: Impostazione scambio chiavi OAuth su MySQL

Nel metodo *dump()* viene aggiunta la riga 2 del listato che segue affinché la “risposta” del Tool Provider al Tool Consumer contenga le informazioni aggiuntive richieste dal progetto. I dati sono restituiti dal metodo *aContent_LiveContentLink_TP()*, esposto ed analizzato di seguito, memorizzati nella variabile *\$ret* e restituiti a chi ha istanziato la classe (in questo caso, il file *tool.php*).

```

1  [...]
   $ret .= "aContent_LiveContentLink() = ".$this->
       _aContent_LiveContentLink_TP()."\n";
3  return $ret;
   [...]

```

Listato 4.3: Restituzione di informazioni aggiuntive necessarie

Il metodo del Listato 4.4 istanzia la classe *AContent_LiveContentLinkDAO()* ed effettua una chiamata al metodo *getContent(v_id,course)*. Al momento dell’implementazione è stato scelto di appoggiarsi al metodo privato sottostante invece di stanziare direttamente la classe *AContent_LiveContentLinkDAO()* così da mantenere “pulito” il codice iniziale di LTI.

```

private function _aContent_LiveContentLink_TP(){
2
   // call to a specific DAO method returning the requested data
   structure
4  define('AT_INCLUDE_PATH', '../.. /include/');
   include AT_INCLUDE_PATH . 'classes/DAO/
       AContent_LiveContentLinkDAO.class.php';
6
   $course_id = $this->info['tile_course_id'];

```

```
8 // info['course'] could be 1 for course
//                                0 for lesson
10 $course = $this->info['course'];

12 // get content from AContent database
return AContent_LiveContentLinkDAO::getContent($course_id ,
    $course);
14 }
```

Listato 4.4: Implementazione del metodo `_aContent_LiveContentLink_TP()`

La classe *AContent_LiveContentLinkDAO()*, contenuta nella cartella */include/classes/DAO/*, raccoglie le informazioni richieste su pagine e lezioni dal database di AContent e le struttura in un documento XML. L'output, quindi, è un documento XML restituito al Tool Consumer. La decisione di creare una classe separata dal resto del codice è data dalla necessità di personalizzare i dati raccolti tramite alcuni metodi appositamente creati e senza modificare il core del sistema di e-learning. La connessione al database MySQL è instaurata tramite le classi DAO messe a disposizione da AContent e gestita, nella codice in esame, tramite un design pattern di tipo Singleton. Di seguito sono riportati i principali metodi utilizzati.

Estrapolazione delle informazioni richieste tramite LTI

Il metodo *getContent(\$v_id, \$course)* della classe prende in input due variabili:

1. **\$v_id**

rappresenta l'id della lezione rispetto al quale formulare la query SQL al database;

2. **\$course**

un flag che specifica se le operazioni sono relative ad una singola pagina ($\$course = 0$) o ad una lezione intero ($\$course = 1$). Nel secondo caso verrà chiamato un metodo ricorsivo che eseguirà la query SQL per ogni elemento e sotto elemento della lezione.

Successivamente, il metodo crea una istanza di DOM per XML all'interno del quale inserirà, sfruttando l'intrinseca struttura ad albero del linguaggio di markup, l'insieme di pagine e lezioni richiesto dal Tool Consumer. Il metodo restituisce il contenuto del documento XML in formato testuale *plain text* (non come un oggetto).

Analisi ricorsiva di pagine e lezioni

Il metodo *_recursiveFolderScan(\$parentID, \$root)* è ricorsivo e prende in input l'id del sotto albero considerato e la radice della intera struttura. Percorre ricorsivamente pagine e lezioni (cartelle comprese) interrogando il database di AContent e restituisce i dati raccolti in un frammento di XML che verrà poi integrato al documento finale.

Elaborazione dei dati estrapolati

Il metodo *_xmlFillFields(\$content, \$root)* manipola i dati estrapolati dal database: giacché la query eseguita si deve attenere alle specifiche imposte dalle classi DAO e non può essere personalizzata, i dati ottenuti non sono sempre utili. Il metodo in questione filtra le informazioni necessarie da quelle ritenute inutili. Sostituisce, inoltre, al campo *text* l'indirizzo URI della pagina di AContent relativa al contenuto.

Adattamento del contenuto di AContent all'iframe di ATutor

Un'ultima modifica ad AContent è stata apportata al file *header.tmpl.php* contenuto nella cartella */themes/default/include/* al quale è stato aggiunto uno script JQuery. Il codice permette di nascondere intestazione, menu laterali e piè di pagina nella visualizzazione del contenuto di AContent se inserito all'interno dell'iframe di AContent Live Content Link.

4.1.2 LTI su ATutor

Per integrare LTI all'interno di ATutor è stata creata la cartella *AContent.lcl* nel percorso */include/classes/*. Questo, per raccogliere in un unico percorso tutte le classi necessarie a stabilire un canale di comunicazione LTI tra Tool Consumer e Tool Provider e i file per la gestione delle estensioni dei moduli *AContent Repository* e *External Tool*. Di seguito sono brevemente elencati e descritti i file e le directory contenuti nella cartella *AContent.lcl*.

ims-blti/

Questa cartella contiene il core di LTI Tool Consumer. Le librerie contenute all'interno si occupano di tutte le fasi di autenticazione OAuth e delle istruzioni LTI per richiedere informazioni al Tool Provider.

images/

Cartella contenente le immagini della struttura ad albero creata dalla classe *TreeGenerator()*.

Instaurazione della comunicazione LTI

AContentLiveContentLink.class.php è la classe principale del sistema progettato e supporta tre versioni di LTI: v1.0, v1.1 e v2.0. Le funzionalità della classe si basano sulle librerie di LTI Tool Consumer e consistono nello stabilire un canale di comunicazione autenticato tramite OAuth. Tramite questo, avviene l'invio automatico del form LTI per la richiesta dei dati al Tool Provider. Infine, la classe, memorizza all'interno database di ATutor le informazioni ottenute dal Tool Provider. Vista la mole della classe, sono riportati, di seguito, i principali metodi ed una breve descrizione.

- **Creazione dell'oggetto *AContentLiveContentLink.class.php***

Al momento della creazione dell'oggetto viene identificata la versione di LTI disponibile. Questo, perché diverse versioni richiedono diverse specifiche nella comunicazione. In entrambi i casi, sono estrapolate le

variabili e i parametri necessari e, una volta strutturati in un array, viene eseguita l'autenticazione tramite il metodo `_OAuth()`. Nel caso quest'ultimo metodo restituisca errore, la computazione termina. In caso contrario, il costruttore prosegue svolgendo operazioni indipendenti dalla versione di LTI e nella creazione dell'array `_Launch_Data` contenente i parametri minimi richiesti dal Tool Provider. Viene, poi, chiamato il metodo `_LTIrequest()` che, interfacciandosi alle librerie di LTI Tool Consumer, inoltra la richiesta di informazioni tramite il canale di comunicazione sicuro creato. L'esecuzione termina chiamando il metodo `_import()` e restituendo l'esito dell'operazione.

- **Interfacciamento a LTI tramite il metodo `_LTIrequest()`**

Le operazioni eseguite consistono nella chiamata al metodo `signParameters()` di LTI al quale sono passate una lunga serie di parametri di configurazione e le chiavi generate per l'autenticazione OAuth. L'oggetto risultante è una stringa contenente un form precompilato secondo le specifiche LTI. Successivamente viene chiamato il metodo `_curlFormAutoSubmit()` che, tramite una funzione CURL[66], invia un form tramite il metodo POST al Tool Provider. La risposta sarà un oggetto di tipo stringa contenente un vasto insieme di informazioni, fra cui, anche quelle richieste. Altri metodi della classe come `_import()`, `_recursiveFolder` e `_storeData()` si occuperanno di filtrare i dati di interesse, organizzarli e inserirli nel database di ATutor utilizzando le classi già a disposizione, proprio come fossero nuovi contenuti del LCMS.

- **Autenticazione tramite `_OAuth()`**

Il metodo `_OAuth()` si occupa “indirettamente” dell'autenticazione OAuth necessaria all'esecuzione di LTI; questo perché non implementa interamente i passaggi necessari all'autenticazione ma si basa su librerie esistenti che ne svolgono egregiamente tutte le funzioni. Il motivo della scrittura di questo metodo è dato dall'elevato numero di librerie incluse. L'autenticazione, infatti, si appoggia anche a tre file di LTI che ne

agevolano l'integrazione con lo strumento: *request_token.php*, *authorization.php*, *access_token.php*. Il metodo *_OAuth* si occupa, inoltre, di definire e dichiarare tutte le variabili necessarie. Chiamare il metodo *_register_consumer()* il quale, tramite le *OAuth API* messe a disposizione da AContent, ottiene la chiave pubblica e la chiave privata per la comunicazione sicura e le restituisce al metodo chiamante. Restituire un flag che indica l'esito dell'operazione di autenticazione.

Interpretazione e memorizzazione dei dati ottenuti tramite LTI

La classe *AContent_lcl_importxml.class.php* analizza i dati XML ricevuti dal Tool Provider. Dalla mole di informazioni ottenute, filtra quelle ritenute utili per il passo successivo, ovvero, la memorizzazione nel database di ATutor. Infatti, il flusso di dati ottenuto tramite LTI deve essere ridotto e organizzato per poter fare uso della classe *ContentDAO.class.php*, opportunamente scritta come estensione della classe *DAO.class.php*.

Elaborazione e visualizzazione della struttura di una lezione per il modulo *External Tool*

La classe *TreeGenerator.class.php* elabora la struttura dati passata in input come array. Percorrendo l'albero XML e scorrendo lezioni, cartelle e sottocartelle, stampa in output la vista di pagine e lezioni in diversi formati: ad albero, ad albero con checkbox, con radio button etc.. Le diverse modalità di visualizzazione vogliono offrire all'autore la possibilità di selezionare uno o più elementi dall'albero per una scelta più dettagliata dei contenuti da importare.

Estensione della classe *DAO* per l'accesso al database di AContent

La classe *ContentDAO.class.php* estende la classe *DAO.class.php* la quale fornisce un'interfaccia astratta al database di ATutor senza mostrare i dettagli implementativi di connessione al database. Si tratta della stessa implementazione che si trova in AContent ma adattata al database di ATutor.

La classe *DAO* per l'accesso al database di AContent

Nel file *Utils.php* è contenuta una funzione di fondamentale importanza: *AContent_has_lcl_support()* determina se il Tool Provider individuato come repository remoto sia in grado di supportare LTI. Come detto, le estensioni dei moduli *AContent Repository* e *External Tool* basano il loro funzionamento su questa compatibilità e, a questo proposito, fanno riferimento alla funzione *AContent_has_lcl_support()*.

4.2 AContent Repository

Come descritto in precedenza, il modulo *AContent Repository* permette di importare in ATutor contenuti provenienti da un repository esterno di AContent. Tramite un motore di ricerca interno, gli autori di ATutor possono effettuare ricerche fra i contenuti remoti di AContent. La richiesta della lista delle lezioni ad AContent avviene tramite il Web Service REST e il tipo di dato restituito è un documento XML che la pagina *index.php* del modulo è pronta ad interpretare e restituire all'autore sotto forma di lista propriamente formattata. Per ogni risultato è possibile visualizzare l'anteprima del contenuto oppure importarlo come copia completa della lezione. Il modulo, indicato con il nome di *tile_search*, è situato nella cartella */mods/_standard/tile_search/*. Senza compromettere il suo normale funzionamento, è stato espanso aggiungendo un'opzione per ogni risultato della ricerca disponibile solo se sia Tool Provider che Tool Consumer supportano LTI. Il listato che segue propone il frammento di codice aggiunto per inserire l'immagine. Il file modificato è *index.tmpl.php* e si trova nella cartella */themes/default/tile_search/*.

```
<?php
2  if ( AContent_has_lcl_support () ) {
    ?>
```

```

4 <a href="mods/_standard/tile_search/import.php?tile_course_id
   =<?php echo $result['courseID']; ?>&title=<?php echo
   urlencode($result['title']); ?>&mode=LCL&desc=<?php echo
   urlencode($result['description']); ?>">
   " style="border-left
     :1px solid #CCC; padding-left: 15px;" title="<?php echo
     _AT("link_alc").' '. $result['title']; ?>" border="0">
6 </a>
   <?php
8 }
   ?>

```

Listato 4.5: Inserimento immagine per la creazione di Live Content Link

E' stata adottata una separazione delle parentesi dell'*if* per mantenere una coerenza con il resto del codice. La Figura 4.1 mostra la nuova icona inserita e posizionata all'estremità destra.



Figura 4.1: Nuova icona inserita (la prima a destra)

Cliccando sull'immagine, l'autore viene rimandato alla pagina *import.php* situata nella cartella */mods/_standard/tile_search/* portando con sé, tramite il metodo GET, alcune variabili fra cui *mode=LCL* che indicherà la scelta di utilizzare LTI per l'importazione. Da qui potrà scegliere, tramite un menu a tendina, quale sarà la radice alla quale "appendere" il contenuto importato. Se non si desidera accodare il repository remoto ad alcuna lezione, basta lasciare immutata l'opzione e cliccare sul bottone *Import*. In modo trasparente all'utente, il seguente frammento di codice aggiunge una lunga

lista di variabili nascoste al form. Questi parametri saranno poi utilizzati per l'autenticazione OAuth e per la connessione al repository tramite LTI.

```
1  $vars = array('tile_course_id' => htmlentities($_GET['
    tile_course_id']),
    'cid'      => '',
3   'title'    => htmlentities($_GET['title']),
    'desc'     => htmlentities($_GET['desc']),
5   'submit'   => 'Save',
    'formatting' => 2,
7   'head'     => htmlentities($_GET['tile_course_id']),
    'body_text' => '',
9   'weblink_text' => htmlentities($_GET['url'])
    );
11 ?>
13 <form name="form1" method="post" action="<?php echo $action; ?>"
    onsubmit="openWindow('<?php echo AT_BASE_HREF; ?>tools/prog.
    php?tile=1');">
15 <?php
    foreach($vars as $name => $value){
17     echo '<input type="hidden" value="' . $value . '" name="' . $name .
        '" />';
    }
19 ?>
```

Listato 4.6: Inserimento dei parametri necessari al Live Content Link

Cliccando sul bottone *Import* la pagina viene ricaricata, intercetta la sottomissione del form e istanzia la classe *AContent_LiveContentLink()*. L'esecuzione viene, quindi, passata al costruttore. Dopo le fasi di autenticazione e passaggio dei dati viste in precedenza, restituirà un flag che indica l'avvenuta importazione o lo stato di errore.

4.3 External Tool

Il modulo *External Tool* implementa già la versione 1.0 di LTI ma richiede un aggiornamento (AContent Live Content Link offre supporto fino alla v2.0 nonostante, al momento, sia ancora in fase di revisione). Lo scopo del progetto è anche di estendere la capacità di importazione di una lezione aggiungendo la possibilità di cercare fra pagine e lezioni e selezionarne una o più sottoparti. La fase di ricerca dei contenuti è sviluppata sulla base del Web Service REST, come per quella effettuata dal modulo *AContent Repository*. La fase di importazione vera e propria viene fatta completamente tramite LTI. Il modulo, indicato con il nome di *basictli*, è situato nella cartella */mods/_standard/basictli/*. Dal nome della cartella si evince l'intenzione di voler offrire supporto alla prima versione di LTI, ovvero, BasiLTI. L'unico file modificato all'interno della directory è *content_edit.php*, situato nella cartella */mods/_standard/basictli/tool/*. Vengono analizzati, di seguito, gli innesti di codice effettuati. Nell'intestazione, fra le inclusioni dei file e delle librerie è stata aggiunta l'importazione del file *Utils.php* per il verificare il supporto di LTI da parte del Tool Provider. I blocchi di codice aggiunti in seguito sono eseguiti solo nel caso in cui la connessione possa essere instaurata. In caso contrario, le parti di codice saranno ignorate e il modulo *External Tool* manterrebbe la sua funzione originaria.

```
require_once(AT_INCLUDE_PATH . 'classes/AContent_lcl/Utils.php');
```

Listato 4.7: Importazione della funzione per il controllo del supporto LTI

La Figura 4.2 mostra il normale funzionamento del modulo dopo la selezione di un External Tool dal menu a tendina. Uno script Javascript intercetta la selezione effettuata e mostra la lista delle impostazioni circa la visualizzazione dei contenuti remoti.

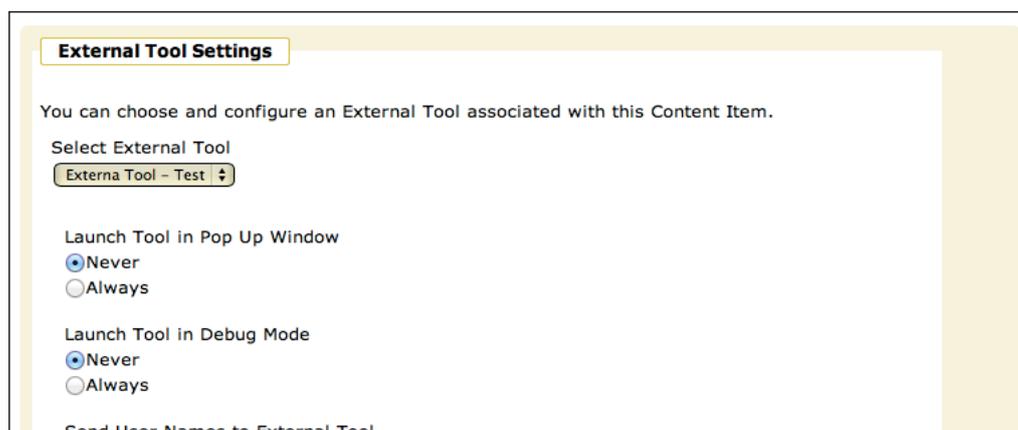


Figura 4.2: Integrazione in una lezione di ATutor di un contenuto remoto tramite il modulo External Tool

Le modifiche rimanenti possono essere suddivise in due blocchi:

1. il primo concerne la costruzione delle componenti dell'interfaccia tramite le quali l'autore può “sfogliare” e selezionare pagine e lezioni da importare. Tali componenti sono visualizzabili solo dopo la selezione di un External Tool;
2. il secondo blocco esegue l'importazione dei contenuti dal repository tramite istanze di classi e chiamate a metodi.

Il primo punto prevede un modulo di ricerca della lezione nel repository di AContent. Il funzionamento è lo stesso del modulo *tile_search*. L'immagine 4.3 ne mostra l'anteprima.

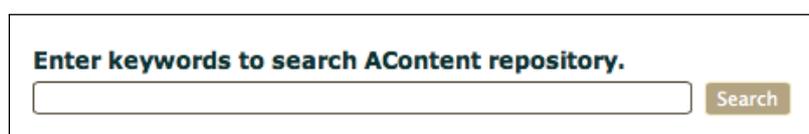


Figura 4.3: Modulo di ricerca della lezione

Una volta inserito il termine di ricerca e inviato il modulo, il pannello *Browse Courses* proporrà all'autore una lista delle lezioni che rispondono

alla parola chiave introdotta (Figura 4.4). Mentre la ricerca viene inoltrata tramite il Web Service REST, la selezione della lezione per mezzo del modulo in Figura 4.4 viene gestita dalle classi dell'estensione AContent Live Content Link. Queste classi, tramite una comunicazione basata su strumenti LTI, restituiscono, per la lezione selezionata, l'intera struttura di pagine, cartelle e sottocartelle. Il box permette di selezionare una sola voce alla volta così da mantenere ridotto il contenuto della pagina del modulo External Tool.

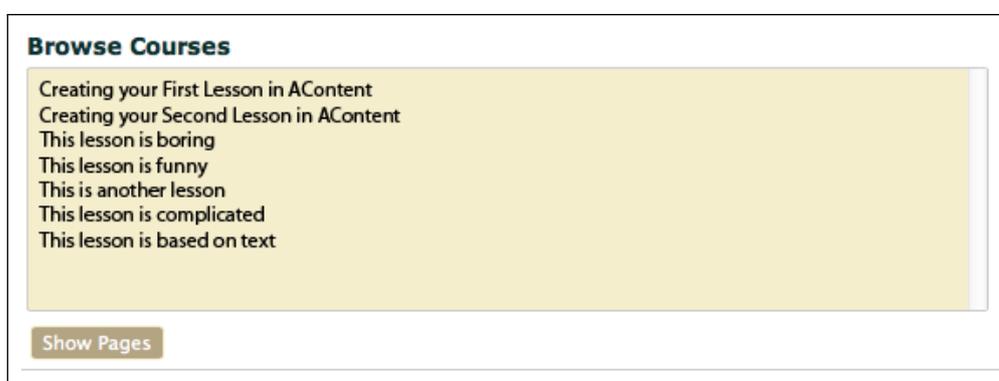


Figura 4.4: Modulo di selezione della lezione

Selezionata la lezione di interesse e cliccato sul bottone *show pages*, comparirà l'ultimo blocco di interfaccia grafica. Il box propone all'autore una struttura ad albero della lezione scelta con pagine, cartelle e sottocartelle. La visualizzazione può essere personalizzata da codice scegliendo una struttura ad albero con checkbox, radio button etc. È chiesto di selezionare, nel caso delle checkbox, per esempio, uno o più "rami" da importare in ATutor. Questo approccio offre all'autore la possibilità di una scelta capillare delle lezioni di interesse così da riuscire a comporre una lezione con diversi frammenti di pagine prese dal repository di AContent. La Figura 4.5 mostra una schermata dell'ultima componente grafica aggiunta al modulo External Tool.



Select the lesson(s) to import

This lesson is funny!

- Page 1
- Folder
 - Subfolder
 - Subpage
- Page 3
- Page 4
- Another folder
 - Page 2

([check all](#) / [uncheck all](#))

Save

Figura 4.5: Modulo di selezione delle lezioni da importare

Come si può notare dall'immagine 4.5, sono stati aggiunti, per comodità di utilizzo, due link che permettono di selezionare e deselezionare tutti gli elementi della lezione con un solo clic. Si provi ad immaginare una situazione in cui le pagine di una lezione siano venti o trenta e l'autore desideri importare quasi tutto il contenuto, eccetto alcune pagine. La Figura 4.6 mostra la fruizione di una lezione remota di AContent importata in ATutor tramite AContent Live Content Link. La visualizzazione del contenuto non differisce da una qualsiasi altra pagina memorizzata in locale; il meccanismo, tuttavia, è differente. Nella fase di caricamento, ATutor inserisce il contenuto remoto in un iframe senza bordo richiamandolo da AContent tramite il riferimento creato con AContent Live Content Link. Nel frattempo, l'authoring tool, come specificato nel corso del capitolo, è stato fornito di uno script Javascript eseguito nel caso in cui il contenuto venisse visualizzato all'esterno della finestra dell'authoring tool stesso. Il codice provvede a nascondere intestazione, menu laterale e piè di pagina dalla visualizzazione della lezione così da isolare il testo della pagina. L'obiettivo è di rendere trasparente all'utente finale il meccanismo di interazione tra i diversi sistemi di e-learning che supportano LTI. L'unica differenza percepibile fra contenuto remoto e contenuto locale

può essere data dal tempo di caricamento (il trasferimento su rete del materiale didattico) che dipende dalla latenza nello stabilire un link remoto in tempo reale fra i contenuti di Tool Provider e Tool Consumer.

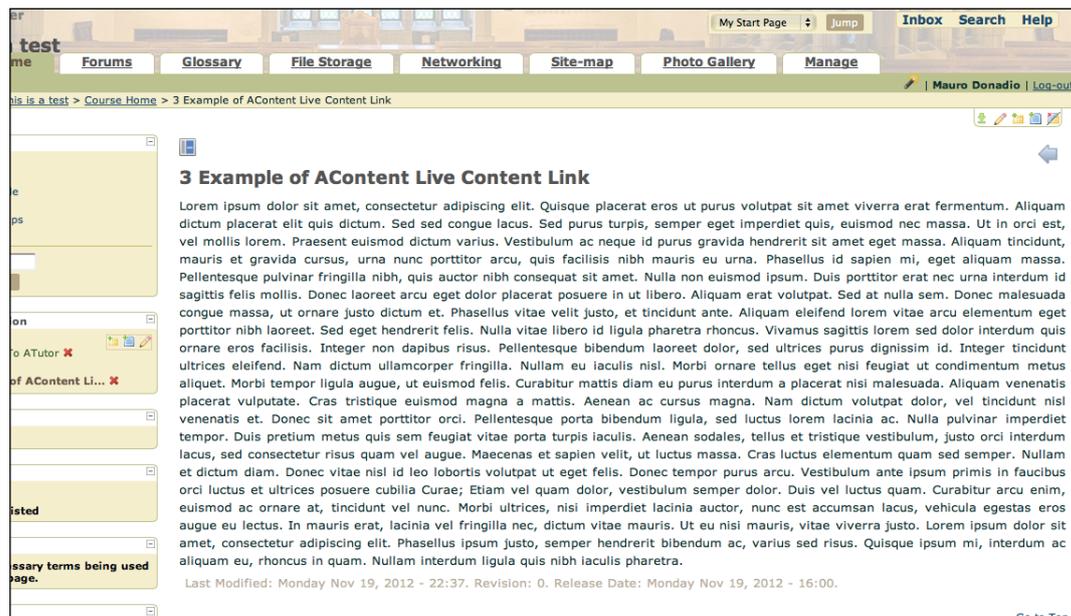


Figura 4.6: Lezione remota consultata su ATutor ma importata tramite External Tool

Conclusioni

La prima parte del progetto di tesi si colloca nella cornice di riferimento di OERtest, un progetto europeo della durata di due anni (Ott. 2010 - Sett. 2012) per sostenere l'integrazione delle Open Educational Resources (risorse educative di libera condivisione) nell'ambito dell'istruzione superiore e per la valutazione dell'apprendimento conseguito. L'intero progetto svolto è stato presentato alla conferenza "E-LEARN 2012 - World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education" [51] a Montréal, Québec nell'ottobre del 2012 e riassunto nella pubblicazione "BEAT: an AContent Extension to Support Authors in Developing Learning Pathways" [52]. L'intento è ora quello di proporre i nuovi strumenti, risultato della prima parte di progetto, affinché siano inseriti nel core della prossima release di AContent. Sviluppi futuri vedono l'integrazione in AContent di un sistema di gestione dei Template di Layout, Template di Pagina e Template di Struttura da parte dell'amministratore del sistema per configurare o, meglio, creare nuovi modelli di template direttamente web. I dati di configurazione e altri dettagli memorizzati in locale potrebbero essere riscritti in formato XML piuttosto che come testo semplice su file con estensione .info; questo cambiamento porterebbe ai template tutti i vantaggi che offre lo standard XML del W3C.

In merito alla seconda parte del progetto di tesi, le prospettive future vedono una crescente diffusione di LTI nel mondo accademico dell'e-learning e una costante attenzione alle caratteristiche di interoperabilità offerte. Il documento delle specifiche di LTI 2.0 propone una revisione della precedente

versione e una serie di nuove specifiche e parametri. L'intenzione è quella di semplificare la comunicazione fra i sistemi che offrono tale supporto e l'integrazione all'interno dei sistemi. Viene mantenuto il protocollo OAuth come principale strumento per la messa in sicurezza della comunicazione ma sono imposti nuovi vincoli ai valori da trasmettere così da rendere richiesta e risposta più complete e dettagliate possibili. Vista la preventiva implementazione della versione 2.0 di LTI su AContent e ATutor, è prevista, non appena sarà supportata dal sistema web, la validazione sul sito di IMS Global Consortium tramite lo strumento automatico offerto. Oltre alla certificazione, garanzia del corretto funzionamento dei Learning Tools Interoperability, il riconoscimento pone i due sistemi di e-learning in cima agli elenchi per quanto riguarda l'accessibilità e l'interoperabilità in questo ambito. Il 5 Ottobre 2012, pochi mesi dopo la conclusione del progetto, è stata ufficialmente rilasciata la versione 2.1 di ATutor comprensiva delle nuove modifiche apportate. LTI 1.1 è supportato e il codice scritto è diventato parte delle prossime release del LCMS. Per quanto riguarda AContent, si rimane in attesa del prossimo rilascio che incorporerà LTI e fornirà pieno supporto allo strumento di interoperabilità. Si prevede, anche in questo caso, l'inserimento del codice scritto per il progetto di tesi all'interno delle prossime distribuzioni di AContent. Nel periodo di tesi trascorso in Canada presso l'istituto di ricerca IDRC[67] della University of Toronto è emersa una discussione ancora aperta circa l'adozione del Web Service REST. Il protocollo, già da tempo utilizzato e perfettamente integrato all'interno dei sistemi, ricopre un ruolo rilevante nelle fasi di comunicazione LTI trattate. Il dibattito vede, da una parte chi sostiene di voler mantenere REST come strumento tradizionalmente di base per alcuni canali di trasferimento delle informazioni, dall'altra chi propone la sostituzione di REST con LTI in tutte le situazioni in cui venga richiesto di instaurare una "connessione remota" con un repository esterno così da aggiornare completamente gli strumenti verso una nuova prospettiva di standardizzazione. Infine, c'è chi intende bilanciare l'utilizzo di entrambi così da mantenere l'offerta dei contenuti a livello dell'ormai consolidato protocollo

REST e, allo stesso tempo, in vista del nuovo standard LTI.

Bibliografia

- [1] ASFOR. Il glossario di e-learning di ASFOR. <http://www.asfor.it/LETTERA%20ASFOR/Lettera%20Asfor%202003%20nr%203.pdf>, 2003.
- [2] MANTESE G. ESPOSITO G. *E-learning: una guida operativa : come realizzare e valutare un progetto*. FrancoAngeli, 2003.
- [3] NÖLTING K. TAVANGARIAN D., LEYPOLD M. *Is e-Learning the Solution for Individual Learning?* Journal of e-Learning, 2004.
- [4] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=182763&contentType=Standards&sortType%3Dasc_p_Sequence%26filter%3DAND%28p_Publication_Number%3A2267%29, 1991.
- [5] Cisco Systems. Learning Object. http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf, 06 1999.
- [6] IEEE. <http://www.ieee.org/index.html>.
- [7] ARIADNE. <http://www.ariadne-eu.org>.
- [8] IMS GLOBAL. <http://www.imsglobal.org/>.
- [9] Dublin Core. <http://dublincore.org>.

-
- [10] Robert J. Beck. *What Are Learning Objects?* Learning Objects, Center for International Education, University of Wisconsin-Milwaukee, 04 2008.
- [11] IMS GLOBAL. Specifiche SCORM. http://www.imsglobal.org/ssp/sspvlp0/imsssp_prflvlp0.html.
- [12] AICC. <http://www.aicc.org/>.
- [13] IEEE LTSC. <http://ieeeltsc.org/>.
- [14] Advanced Distributed Learning. <http://www.adlnet.org>.
- [15] W3C. XML. <http://www.w3.org/XML/>.
- [16] Ecma International. ECMAScript. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=33835.
- [17] IMS GLOBAL. IMS Content Packaging. <http://www.imsglobal.org/content/packaging/>.
- [18] IMS GLOBAL. Common Cartridge. <http://www.imsglobal.org/commoncartridge.html>.
- [19] CCAPMG. <http://www.imsglobal.org/cc/commoncartridge.html>.
- [20] Dublin Core DCMI. Dublin Core standard DCMI. <http://dublincore.org/specifications/>.
- [21] IMS GLOBAL. IMS LTI. <http://www.imsglobal.org/toolsinteroperability2.cfm>.
- [22] IMS GLOBAL. IMS LTI risorsa video. <http://vimeo.com/14100773>.
- [23] W3C. Web Services. <http://www.w3.org/TR/ws-arch/>.
- [24] Roy Thomas Fielding. REST. http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, 2000.

-
- [25] W3C. SOAP. <http://www.w3.org/TR/soap/>.
- [26] JSON. <http://www.json.org>.
- [27] OAuth. <http://oauth.net>.
- [28] Open ID. <http://openid.net>.
- [29] Ellis R. K. *A field guide to Learning Management Systems*. Learning Circuits, ASTD, 2009.
- [30] W3C. Authoring Tool. <http://www.w3.org/WAI/intro/atag.php>.
- [31] Centro Nazionale per l'Informatica nella Pubblica Amministrazione. Authoring Tools Editors. http://www.pubbliaccesso.gov.it/biblioteca/manualistica/accessibilita_siti/tool/authoring_tool.htm.
- [32] W3C. SMIL. <http://www.w3.org/AudioVideo/>.
- [33] Microsoft. Scalabile. [http://msdn.microsoft.com/it-it/library/aa292172\(v=vs.71\).aspx](http://msdn.microsoft.com/it-it/library/aa292172(v=vs.71).aspx).
- [34] W3C. Usabile - ISO 9241. <http://www.w3.org/2002/Talks/0104-usabilityprocess/slide3-0.html>.
- [35] W3C. WAI. <http://www.w3.org/WAI/>.
- [36] Adobe. Adobe Flash. <http://www.adobe.com/it/products/flash.html>.
- [37] IMS GLOBAL. IMS QTI. <http://www.imsglobal.org/question/>.
- [38] Mozilla Firefox. <http://www.mozilla.org/en-US/firefox/new/>.
- [39] Mozilla Thunderbird. <http://www.mozilla.org/en-US/thunderbird/>.
- [40] Mozilla Seamonkey. <http://www.seamonkey-project.org>.

- [41] Mindonsite MOS-SOLO. <http://www.mindonsite.com>.
- [42] Microsoft Silverlight. <http://www.silverlight.net>.
- [43] Microsoft. Power Point. <http://office.microsoft.com/it-it/powerpoint/>.
- [44] OERtest Europe. <http://www.oer-europe.net>.
- [45] ECTS. http://ec.europa.eu/education/lifelong-learning-policy/ects_en.htm.
- [46] AContent. <http://atutor.ca/acontent/>.
- [47] L. 9 gennaio 2004, n. 4.
- [48] Centro Nazionale per l'Informatica nella Pubblica Amministrazione. Verifica tecnica e requisiti tecnici di accessibilità delle applicazioni basate su tecnologie internet. <http://www.pubbliaccesso.gov.it/normative/DM080705-A.htm>.
- [49] Ministro per la Pubblica Amministrazione e la Semplificazione. Criteri e metodi per la verifica tecnica e requisiti tecnici di accessibilità per i contenuti e i servizi forniti per mezzo di applicazioni basate su tecnologie Web. http://www.funzionepubblica.gov.it/media/556981/linee_guida_acc.pdf.
- [50] W3C. WCAG. <http://www.w3.org/TR/WCAG/>.
- [51] AACE. <http://aace.org/conf/elearn/>.
- [52] BEAT: an AContent Extension to Support Authors in Developing Learning Pathways, 2012. Salomoni P., Mirri S., Prandi C.
- [53] W3C. CSS. <http://www.w3.org/Style/CSS/>.
- [54] W3C. HTML & CSS. <http://www.w3.org/standards/webdesign/htmlcss>.

-
- [55] TinyMCE. <http://www.tinymce.com>.
 - [56] W3C. W3C XHTML. <http://www.w3.org/TR/xhtml1/>.
 - [57] MyUDUTU. <http://www.udutu.com/>.
 - [58] MSLCDS. <http://www.microsoft.com/learning/en/us/training/lclds.aspx>.
 - [59] ATutor. <http://atutor.ca/atutor/>.
 - [60] IMS LTI test. <http://www.imsglobal.org/developers/LTI/test/>.
 - [61] PHP. <http://php.net>.
 - [62] JQuery. <http://jquery.com>.
 - [63] MySQL. <http://www.mysql.com>.
 - [64] Licenza GNU GPL. <http://www.gnu.org/licenses/gpl.html>.
 - [65] ref.DAO. <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
 - [66] ref.CURL. <http://curl.haxx.se>.
 - [67] ref.IDRC. InclusiveDesignResearchCentre-<http://idrc.ocad.ca>.

Ringraziamenti

Inizio ringraziando chi mi ha seguito e guidato nella stesura di questa tesi. Ringrazio la Prof.ssa Paola Salomoni per la sua disponibilità e la sua fiducia nell'affidarmi questo lavoro di tesi. Con lei, ringrazio la Dott.ssa Catia Prandi e la Dott.ssa Silvia Mirri per l'altrettanta disponibilità mostratami e per gli ottimi consigli e suggerimenti.

Ci tengo anche a ringraziare il Project Manager Greg Gay e la progettista e sviluppatrice Cindy Li dell'Università di Toronto per avermi seguito nelle fasi di implementazione della seconda parte di progetto.

Ringrazio i miei genitori e mio fratello per essermi stati accanto e aver condiviso con me i momenti più difficili.

Ringrazio i miei amici e tutti coloro che mi hanno sostenuto e sopportato nei miei alti e bassi.

Ringrazio Chiara per avermi insegnato ad essere felice.