

Alma Mater Studiorum Università di Bologna

Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di laurea in Scienze e Tecnologie Informatiche

Sviluppo guidato di Learning Object con AContent

Tesi di Laurea in SISTEMI OPERATIVI

Relatore:
Chiar.ma Prof.ssa
PAOLA SALOMONI

Presentata da:
MATTEO CEPPINI

Correlatore:
Chiar.ma Dott.ssa
CATIA PRANDI

I Sessione
A.A. 2012/2013

*Alla mia famiglia **Sabrina, Giancarlo e Marco,**
che mi hanno sempre sostenuto...
A **Giorgia,**
senza la quale oggi non sarei qui...*

Introduzione

La nascita delle reti e la diffusione globale di Internet ha portato ad una notevole trasformazione nel modo di fare formazione a distanza. Essa agli albori era caratterizzata da strumenti come, CD-ROM o ancor prima, i loro progenitori floppy disk, che limitavano gli studenti ad un metodo di apprendimento esclusivamente off-line. L'avvento della rete ha portato alla nascita dell'E-learning, un'idea di formazione del tutto nuova che sfrutta le potenzialità della telematica e tutti i vantaggi offerti dalle nuove tecnologie.

E-learning significa, *electronic based learning*, una nuova metodologia di didattica che sfrutta e viene facilitata da risorse e servizi disponibili e accessibili virtualmente in rete. Dalla sua definizione si ricava come i sistemi di comunicazione e di informazione non debbano essere obbligatoriamente collegati ad Internet, lasciando così spazio ad altre tipologie di reti, come quelle aziendali, istituzionali ed anche reti del tutto off-line.

Al momento vi sono numerose piattaforme di e-learning sul mercato e questa varietà fa nascere il problema della compatibilità dei contenuti didattici creati sui diversi sistemi. Esso ci conduce al problema dell'interoperabilità, ovvero la capacità di due sistemi diversi di scambiare e utilizzare materiali didattici prodotti in diversi sistemi di e-learning.

Da questa panoramica si introducono tre concetti. Il Learning Content Management System (LCMS), un ambiente multi-utente in cui è possibile creare, salvare, gestire e inviare materiali didattici di un repository centrale. L'Authoring Tool, ovvero un editor di contenuti software oppure una piattaforma che permette all'utente di produrre pagine e contenuti web, infine LTI uno strumento per l'interpolazione dei contenuti tra diversi sistemi di e-learning.

Questo documento di tesi raccoglie la descrizione della progettazione e della fase implementativa di un'estensione per il tool autore AContent. L'obiettivo è quello di creare un nuovo strumento per la creazione delle lezioni che vada ad affiancare i due metodi già esistenti e che offra una soluzione innovativa per la modellazione dei contenuti all'interno di AContent, ma che sia anche un punto di partenza e di esempio per estensione future anche su altre piattaforme. Questo nuovo approccio che prende il nome di *wizard* si pone come obiettivo quello di sopperire alle mancanze strutturali dei metodi preesistenti. L'innovazione risiede intrinsecamente nella sua idea di base, ovvero il desiderio di fornire uno strumento che potesse essere fruibile sia ad utenti esperti, che hanno una conoscenza ampia dell'*authoring tool*, sia ad utenti che si avvicinano per la prima volta a questa piattaforma, permettendo loro di poter apprendere attraverso il suo utilizzo tutte le varie funzionalità offerte dalla piattaforma. Questo tipo di soluzione è del tutto nuova per l'*authoring tool* AContent, infatti fino ad ora, esso forniva due possibili modalità per la creazione e la modellazione delle lezioni, ovvero l'approccio *Manually* e *Structure*. Il primo consente la creazione singola di un contenuto del tipo *page* o *folder*, mentre il secondo permette di associare alla lezione che si sta modellando una struttura predefinita fornita di default da AContent, al momento esso dispone di tre strutture *creative*, *knowledge* e *meta-competency*.

Essi risultavano ad una analisi attenta poco flessibili e soprattutto di difficile comprensione per un neofita di AContent. È dall'analisi di questi metodi che nasce l'idea alla base di *wizard*. Esso raccoglie tutte le funzionalità degli approcci esistenti, ma ne elimina i difetti, dando vita ad uno strumento che permette in maniera veloce e flessibile di creare e modellare nuovi LO. La sua struttura è stata progettata diversificandola in vari passaggi, ognuno dei quali correlato di descrizioni e informazioni utili per i nuovi utenti al fine di capire e soprattutto, imparare come poter rieseguire quella funzione in futuro. La struttura composta da vari step è stata implementata in modo tale da non influire sulla fruibilità dello strumento rendendolo così efficace anche per utenti esperti.

Il documento è diviso in tre capitoli. Nel primo troviamo una panoramica sull'e-learning e sui tre concetti chiave elencati nei paragrafi precedenti. Nel secondo viene

spiegato in maniera dettagliata l'idea alla base dello strumento e di tutte le sue componenti, mentre nel terzo ed ultimo capitolo viene trattato l'aspetto implementativo di *wizard*, in particolare si pone minuziosa attenzione su tutti i passaggi che lo compongono e che consentono di creare questo nuovo strumento.

Indice

Introduzione	i
1 L'apprendimento ai tempi di internet	1
1.1 E-learning	2
1.2 Strumenti	3
1.2.1 Tool Autore	4
1.2.2 Learning Management System	7
1.3 Standard	12
1.3.1 SCORM 2004	12
1.3.2 IMS Content Packaging	15
1.3.3 IMS Common Cartridge	15
1.3.4 IMS QTI Test	19
2 Progetto	21
2.1 AContent	22
2.1.1 Caratteristiche	23
2.1.2 Tecnologie	25
2.1.2.1 Linguaggi di programmazione	25
2.1.2.2 Database	28
2.1.3 I template	30
2.1.4 Approcci di creazione contenuti	32
2.2 Wizard	35
2.2.1 Dettami di progettazione	37
2.2.2 Obiettivi	39
3 Implementazione	43
3.1 Selezione approccio wizard	43
3.2 Settaggio attributi della lezione	44
3.3 Selezione contenuto didattico	46
3.4 Funzioni dei contenuti didattici	49
3.4.1 Contenuto folder	56
3.4.2 Contenuto page	59
Conclusioni	67
Bibliografia	69

Elenco delle figure

2.1	Schermata approccio Manually e Structure	32
2.2	Schermata di scelta di una delle tre strutture predefinite di AContent	34
2.3	Icone presenti nella sezione centrale e nel tab laterale sinistro	36
2.4	Schermata di errore Login	38
2.5	Schermata di errore di Wizard	38
2.6	Architettura directory degli obiettivi	40
2.7	File XML della struttura Creative	41
3.1	Primo passaggio di wizard	43
3.2	Codice primo passaggio	44
3.3	Secondo passaggio di wizard	44
3.4	Terzo passaggio di wizard	46
3.5	Codice funzione Javascript	47
3.6	Quarto passaggio di wizard, con goals	47
3.7	Quarto passaggio di wizard, senza goals	48
3.8	Codice per caricamento automatico obiettivi di Creative	48
3.9	Schermata di errore	49
3.10	Link home page	50
3.11	Link delete content	50
3.12	Schermata delete	50
3.14	Codice iniziale controllo numero contenuti	51
3.15	Link di collegamento alla pagina di navigazione	51
3.16	Schermata di Navigazione fra i contenuti	51
3.17	Prima parte funzione <i>“printMenuWizard”</i>	52
3.18	Pulsanti Back e Next	52
3.19	Schermata Info strutture	53

3.19	Link aggiunta strutture predefinite	54
3.20	Schermata aggiunta strutture predefinite	54
3.21	Parte della funzione per la creazione della struttura predefinita	56
3.22	Schermata contenuto folder	57
3.23	Codice Html per la schermata d'inserimento di una sotto cartella	58
3.24	Schermata creazione sotto pagina	58
3.25	Schermata contenuto page	59
3.26	Schermata esclusiva dell'editor	61
3.27	Pulsanti gestione template	61
3.28	Schermata template	62
3.29	Edit template	62
3.30	Schermata layout	63
3.31	Anteprime layout Italy e Windows	64

Elenco delle tabelle

Tabella 1: Caratteristiche principali AContent 1.2	24
Tabella 2: Obiettivi delle strutture predefinite	40

Capitolo 1

L'apprendimento ai tempi di internet

Nel corso della storia possiamo diversificare il concetto di apprendimento in due famiglie distinte, la prima è la “formazione di persona”, mentre la seconda è la “formazione a distanza”, quest’ultima è andata a pari passo con la tecnologia fino a raggiungere con l’avvento di internet la sua massima espressione.

La diffusione della rete e dei sistemi informatici [FaD1] ha portato alla nascita della “formazione on-line”, conosciuta con il nome E-learning che sostituisce la vecchia modalità di apprendimento passivo, con un processo che si sviluppa sfruttando l’uso delle tecnologie informatiche multimediali e della rete per migliorare la qualità dell’apprendimento facilitando l’accesso alle risorse, ai servizi e alla creazione di comunità virtuali per lo scambio e la collaborazione fra gli utenti. In particolar modo l’E-learning permette agli utenti di pubblicare materiale didattico attraverso il web o una rete privata dando la possibilità agli utenti di usufruire di uno strumento flessibile, in quanto personalizzabile, e facilmente accessibile, cosa mai successa prima nella formazione a distanza [FaD2].

1.1 E-learning

L'E-learning è l'ultima e per ora massima espressione della Formazione a Distanza (FaD), ma è solo l'ultima delle tre grandi generazioni che fino ad oggi hanno composto la FaD. La prima nacque a metà del IX secolo con la didattica per corrispondenza postale, chiamata all'epoca *istruzione per corrispondenza* [IpC]. Essa era composta da documenti cartacei inviati per lettera, spesso corredati da informazioni su come studiarli; la didattica si basava sull'autoapprendimento e gli studenti potevano interagire con gli insegnanti solo in determinate occasioni prestabilite e la fase di valutazione avveniva tramite lo scambio di test scritti, che lo studente inviava al docente per posta dopo averli svolti. La FaD si è sviluppata in simbiosi con le tecnologie e nel XX secolo con l'avvento in successione della radio, del telefono e della televisione finisce l'era dell'istruzione per corrispondenza e si entra nella seconda generazione basata sulle tecnologie audiovisive. Queste nuove tecnologie fra cui inseriamo anche la teleconferenza portano la formazione a distanza ad uno stadio più evoluto, migliorando l'interazione fra docenti e studenti, facilitando lo scambio e l'accesso ai materiali, la didattica e la fase di valutazione. Verso gli anni ottanta con l'avvento dei personal computer si entra nell'ultima generazione, 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 passando da auto apprendimento passivo ad apprendimento attivo, cooperativo e dinamico. La FaD prende il nome di E-learning [E-lea], in cui la didattica viene agevolata 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, sistemi di informazione e comunicazione, non necessariamente in rete, servono come supporto specifico per l'attuazione del processo di apprendimento. L'effettiva connessione alla rete non è strettamente necessaria e questo permette l'utilizzo di architetture come reti private oppure sistemi totalmente offline, tuttavia questa capacità di svilupparsi in sistemi e piattaforme diverse porta il problema della compatibilità dei contenuti didattici. Nasce infatti l'esigenza di condividere materiale didattico con altri

Learning Management System(LMS) [LMS1], questa problematica porta all'introduzione del concetto di interoperabilità intesa come la capacità di due o più LMS di scambiarsi informazioni e di utilizzarle. Negli anni si sono susseguiti diversi standard per l'interoperabilità che definivano come i dati dovevano essere impacchettati per poi essere utilizzati su diversi LMS. Oltre ai contenuti è necessario fornire metadati descrittivi circa la struttura del pacchetto stesso, per questo introduciamo il concetto di Learning Object (LO) [LO] come un insieme di elementi di contenuto, elementi di pratica ed elementi di valutazione combinati in un singolo pacchetto. Naturalmente questo nuovo concetto necessitava di uno standard ben definito, il quale viene creato grazie alla collaborazione di tre grandi associazioni no profit IMS GLOBAL [IMS-G], Dublin Core Che [DC] e ARIADNE [ARI], che danno vita ad un documento, approvato il 15 Luglio 2002 dall'IEEE (Istituto degli ingegneri elettrici ed elettronici [IEEE]). All'interno di questo documento viene specificata la struttura dei metadati di un'istanza di LO ed i 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.

1.2 Strumenti

Ora che abbiamo spiegato cosa si intende con il termine E-learning dobbiamo esaminare gli strumenti che permettono la realizzazione di questa ultima frontiera del FaD. Nel capitolo precedente abbiamo già introdotto due concetti fondamentali, il learning objects (LO) e il learning management system (LMS). Le due classi che vedremo nei sotto capitoli che seguono racchiudono i concetti principali dell'e-learning, ovvero gli strumenti che permettono la creazione di LO e quelli che permettono la loro gestione.

1.2.1 Tool Autore

Questa famiglia racchiude i cosiddetti authoring tool [Aut1] o editor di contenuti. Essi sono software o servizi che permettono agli utenti di produrre pagine e contenuti web. Svareti programmi [Aut2], possono essere considerati tool autore, di seguito vediamo un elenco dei più generici:

- **WYSIWYG**: l'acronimo sta per l'inglese What You See Is What You Get, ad una tipologia di editor per la creazione di contenuti web che offre al contrario degli editor tradizionali una visione di come sarà il risultato finale [WYSIWYG]. Questa sua funzionalità lo fa entrare in conflitto con gli editor tradizionali che richiedono agli sviluppatori di inserire i markup (o codici descrittivi) e non consentono di vedere cosa queste istruzioni comportano sull'aspetto finale. Il primo vero editor WYSIWYG è stato un programma di elaborazione testi progettato da Charles Simony alla Xerox Palo Alto Research Center [XPAR] nel 1970 che prese il nome di Bravo, esso divenne la base per il lavoro di Simony presso la Microsoft e da essa nacquero altre due applicazioni famosissime Word ed Exel.
- **CMS**: l'acronimo sta per Content Management System e non è altro che un sistema di gestione dei contenuti che permette anche ad utenti che non hanno familiarità con l'HTML di gestire la creazione, la rimozione e la modifica di contenuti da un sito web senza avere necessariamente la conoscenza di un programmatore informatico [CMS1]. I CMS ci permettono di scindere la fase di realizzazione dell'architettura del sito e della architettura d'informazione dalla fase di inserimento dei contenuti, in questo modo la gestione dei contenuti può essere gestita in maniera semplice da operatori non necessariamente esperti di informatica [CMS2].
- **WIKI**: questo strumento fa parte della famiglia più ampia che raccoglie tutti i siti web che permettono agli utenti di inserire dei contenuti, come ad esempio blog, social network o altre tipologie di questo tipo. Nello specifico con il termine wiki intendiamo una collezione di documenti ipertestuali aggiornata e gestita da un team di collaboratori che ne hanno accesso [WIKI].

E' un software nato in origine nel settore del design pattern di comunità web per risolvere il problema di discutere e scrivere pattern languages. Il primo wiki venne creato dal programmatore Ward Cunningham nel 1995 e fu chiamato Portland Pattern Repository, Ward inoltre concepì il primo motore wiki. Questa soluzione implementativa verso la fine del XX secolo fu considerata la via per sviluppare le basi di una conoscenza pubblica e privata ed da essa trasse ispirazione il progetto dell'enciclopedia Nupedia [Nup] ideata da Jimmy Wales con capo redattore Larry Sanger. Oggi l'architettura wiki più sviluppata è la famosissima Wikipedia [Wikip] in lingua inglese.

- **SMIL**: il suo acronimo sta per “Synchronized Multimedia Integration Language”, conosciuto con la pronuncia inglese di sorriso, “smile” [SMIL]. Come L'HTML o l'XML anche SMIL è un linguaggio di marcatori e quindi non può essere considerato un linguaggio di programmazione vero e proprio. Il suo compito è quello di gestire la programmazione ed il caricamento di file multimediali, come immagini e video. Questo strumento è stato concepito dal World Wide Web Consortium [W3C] con specifiche aperte. E' un prodotto largamente diffuso infatti lo possiamo trovare in aziende leader come Apple, Microsoft, Adobe e Real Networks.

Tutti questi software che abbiamo appena elencato sono fra i più comuni e fra i più “semplici” disponibili sul mercato. In ambito di e-learning gli authoring tool devono permettere di creare LO che possono essere anche oggetti complicati, devono inoltre soddisfare varie caratteristiche fondamentali:

- I. *scalabilità*: si intende la possibilità di aumentare o diminuire le risorse a disposizione del software a fronte di cambiamenti dell'ambiente in cui si lavora senza influenzare negativamente le prestazioni;
- II. *modularità*: si intende la suddivisione del software in moduli separati in modo tale che essi non dipendano gli uni dagli altri;
- III. *accessibilità dei contenuti creati e del sistema stesso*: un tool autore deve attenersi agli standard internazionali sull'accessibilità;
- IV. *compatibilità con gli standard di e-learning*: la scelta del software deve considerare quale tipo di formalismo scegliere, ve ne sono di vario tipo, fra cui

citiamo IMS Common Cartridge e SCROM, che verrà descritto nella sezione 1.3.1 .

Vediamo ora alcuni esempi dei più diffusi esistenti:

- **Exelearning:** è uno strumento [eXe] autore basato sul web, progettato in XUL (XML User Interface Language), un linguaggio di markup per l'interfaccia utente, ideato da Mozilla Project e considerato da molti una distorsione di XML. Esso permette di creare in modo semplice learning object; è SCROM compatibile, ed è un software open source. Questo strumento è stato pensato per facilitare insegnanti e studiosi durante la fase di progettazione, pubblicazione e sviluppo di contenuti didattici web-based permettendo anche a coloro che non conoscono i linguaggi come Php, Html o XML e le varie operazioni necessarie alla web publishing di creare in maniera semplice materiali didattici basati sul web. Infatti, Exelearning fornisce componenti predefinite come oggetti multimediali, wiki, gallerie di immagini o quiz. Nonostante la vasta gamma di opzioni a disposizione la libertà di personalizzare i contenuti è abbastanza limitata.
- **Xerte:** l'acronimo significa "Xml Editor and Run-Time Engine", nato nel 2006 nell'università di Nottingham per soddisfare l'esigenza di uno strumento che offrisse un alto grado di interazione fra utenti e piattaforma. Rilasciato con licenza open source è stato implementato su un ambiente grafico ad icone, esso punta alla creazione di contenuti interattivi molto complessi, ma in maniera veloce. Grazie all'utilizzo di alcuni script, permette di integrare testo e contenuti multimediali in una interfaccia accessibile. Xerte è affiancato da una suite di strumenti web denominata Xerte Online Toolkits, utilizzata dagli sviluppatori di contenuti che permette di assemblare i materiali con semplici procedimenti wizard. Possiamo in conclusione dire che questo software offre una soluzione flessibile e facilmente estendibile, ma ha lo svantaggio di essere legato alla piattaforma Microsoft Windows.
- **AContent:** è un software rilasciato con licenza open source basato sul web, indipendente dalla specifica piattaforma. E' un sistema monolitico, ma permette

facilmente di integrare codice implementato da diversi sviluppatori. Supporta i più diffusi standard per i learning object fra cui SCROM, Common Cartridge e QTI. Nasce in origine come modulo del sistema di learning management system ATutor, ma successivi rilasci di versioni sempre più complesse ed articolate hanno portato alla separazione dei due strumenti.

Questa breve scaletta dei più utilizzati authoring tool si è conclusa con AContent il quale è lo strumento scelto per l'implementazione del progetto (che verrà trattato in dettaglio, nel prossimo capitolo).

1.2.2 Learning Management System

Un LMS è una piattaforma applicativa o un insieme di programmi [LMS2] che permettono la divulgazione di materiali online che sono stati creati utilizzando i tool autore. Queste piattaforme sono spesso associate ad una altra struttura chiamata LCMS, "Learning Content Management System" [LCMS], da come si deduce dall'acronimo essa si occupa della gestione dei contenuti, mentre LMS si occupa degli utenti e dell'analisi delle statistiche. Vediamoli più in dettaglio.

I LCMS sono moduli software presenti nelle piattaforme e-learning che racchiudono tutte le funzionalità necessarie per la gestione dei contenuti, ovvero:

- I. creazione, gestione e memorizzazione dei materiali didattici;
- II. tracciamento e archiviazione delle iterazioni fra gli utenti e i learning object;
- III. creazione e modularizzazione dei learning object;

L'ultimo aspetto fondamentale di cui si occupa un LCMS è quello inerente all'importazione e alla pubblicazione dei learning object all'interno della piattaforma di e-learning.

I LMS permettono di gestire gli utenti e l'analisi delle statistiche e sono implementati cercando di facilitare il più possibile l'accesso e la gestione al materiale didattico. Consentono agli utilizzatori finali quattro funzionalità:

- I. *registrazione*: ogni utente può approcciarsi ad un LMS e decidere se registrarsi o utilizzarlo in modalità anonima, naturalmente usando quest'ultima molte funzioni saranno disabilitate;
- II. *consegna*: a tutti gli utenti registrati, di norma, è consentita la possibilità di depositare contenuti al fine di creare nuovi corsi o per consentire verifiche sull'apprendimento;
- III. *frequenza*: si può gestire la presenza ai corsi inseriti nella piattaforma verificando l'avvenuta visione dei vari materiali da parte degli utenti;
- IV. *verifica*: vi è la possibilità di effettuare test per la verifica delle conoscenze acquisite durante un corso o dopo aver visionato una singola lezione;

Questi quattro punti racchiudono le funzionalità basilari di tutti i learning management system; strumenti aggiuntivi diversificano poi i vari LMS. Nei sistemi più complessi possiamo trovare varie funzionalità, fra cui gli strumenti per l'amministrazione di competenza, per l'analisi delle abilità, per la pianificazione di successioni, per la ripartizione delle risorse, per le certificazioni e per i codici di categoria virtuali.

Ora che abbiamo visto in cosa consistono i LMS possiamo fare una carrellata dei software più famosi che svolgono anche le funzionalità di LMS.

- **Moodle**: acronimo di "Modular Object-Oriented Dynamic Learning Environment" [Moo] , è un pacchetto software ideato per la creazione di corsi basati su internet e siti web. E' fornito come codice open source, sotto al GNU, General Public License [GPL], questo significa che Moodle è protetto da copyright, ma è permesso agli utenti di implementare funzionalità aggiuntive personalizzate, a condizione che vengano forniti i codici sorgenti, che non venga modificata o rimossa la licenza originale e i diritti d'autore, naturalmente a tutte le estensione prodotte dovrà essere assegnata la licenza originale. Moodle è stato

creato da Martin Dougiamas, un amministratore di rete presso la Curtin University in Australia, laureato in informatica e con un master di un anno in educazione. Moodle nasce durante gli studi che Martin fa per la tesi di dottorato, (tra l'altro mai realizzata), su *l'uso del software libero per aiutare un'epistemologia costruzionista sociale di insegnamento e apprendimento all'interno di comunità, con domande riflessive, basate su internet* [M.Dou] . L'ideologia costruzionista [Costr] su cui si basa Moodle, dalla quale nascerà lo statunitense "No Child Left Behind Act of 2011", è evidenziata da molti aspetti del suo sviluppo, come la possibilità di far commentare o aggiungere contenuti agli studenti o di consegnare o correggere compiti tramite la rete. Per quanto riguarda i docenti che utilizzano questo strumento si ha la possibilità di visualizzare i log degli alunni esaminando quelli che sono collegati da più tempo. Moodle inoltre non vincola l'insegnante e gli lascia la possibilità di gestire il proprio corso, offrendo un supporto di orientamento al conseguimento di determinati risultati. Per quanto concerne invece le specifiche esso può essere utilizzato su qualsiasi macchina che supporti il Php e sia in grado di gestire un database di tipo SQL, come ad esempio MySQL. Dal punto di vista dei sistemi operativi questo software è supportato da Windows, Mac e varie versioni di Linux.

- **Claroline:** [Cla] possiamo classificarla come una piattaforma collaborativa di e-Learning e di e-Working rilasciata con licenza open source. Claroline è organizzata sul concetto associato ad un determinato corso o più in generale ad un'attività didattica. In ognuno di questi spazi vengono forniti vari strumenti che permettono la creazione dei contenuti, fra cui ricordiamo solo i due più complessi, supporto agli esercizi (conforme allo standard IMS/QT2) e supporto ai percorsi didattici (conforme allo standard SCROM). Questa piattaforma è compatibile con GNU/Linux, MacOS e Microsoft Windows e si basa su tecnologie open come Php e MySQL. Il suo sviluppo iniziò nel 2000 dall'UCL (Università cattolica di Louvain, Belgio) seguendo le necessità e le esperienze indicate dagli insegnanti. In questo momento Claroline gode del sostegno finanziario della Region Wallonne per il suo sviluppo attraverso il progetto WIST, in questo programma Claroline raggruppa tre importanti partner belgi, il

CERDECAM (Research and development centre of the ECAM, Brussels), il LENTIC (Research centre on new technologies, innovation and change of the ULg, Liegi) e in fine l'IPM (University pedagogy and multimedia institute from UCL, Louvain-la-Neuve).

- **ILIAS**: l'acronimo significa "Integrated Learning, Information and Work Cooperation System" è una piattaforma basata sul web con licenza open source, supporta l'apprendimento di gestione dei contenuti (conforme allo standard SCROM 2004) e strumenti per la comunicazione, la valutazione, la collaborazione e la valutazione. Può essere eseguita su tutte le macchine che supportano Php e MySQL. L'idea di partenza di questa piattaforma è quella di offrire un ambiente flessibile per l'apprendimento ed il lavoro on-line sfruttando strumenti integrati, possiamo vederla come una sorta di libreria che fornisce apprendimento e di gestione dei materiali in qualsiasi posizione del repository, ciò comporta che la piattaforma possa essere considerata non come un magazzino chiuso, ma come un porto libero ed aperto a tutti, anche ad utenti non registrati. ILIAS è stato uno dei primi sistemi di gestione di apprendimento utilizzato dalle università, il primo prototipo è stato sviluppato alla fine del 1997, nell'ambito del progetto VIRTUS presso l'università di Colonia, il software venne rilasciato l'anno dopo ed offerto per l'apprendimento presso la stessa facoltà, ma a causa della crescente richiesta e d'interesse di altre università il team progettista decise di rilasciare ILIAS in versione open source sotto licenza GPL nel 2000. Negli anni successivi sono state sviluppate altre versioni fino ad arrivare all'ultima versione (4.3.1), rilasciata il 14 Gennaio 2013.
- **ATutor**: è il contenitore di AContent [ATu]. I campi di utilizzo di questa piattaforma sono diversi, abbiamo la gestione dei corsi on-line, lo sviluppo professionale degli insegnanti, lo sviluppo di carriera e nel campo della ricerca accademica. ATutor è all'avanguardia per le sue caratteristiche di accessibilità, infatti esso è particolarmente adatto a studenti con disabilità ed è totalmente conforme alle norme stabilite dall'ASTD (American Society for Training and

Development). Queste sue caratteristiche l'hanno portato ad essere ampiamente utilizzato a livello internazionale, basti pensare che il software è stato tradotto in oltre quindici lingue con il supporto di oltre quaranta moduli di lingua aggiuntivi attualmente in sviluppo. La prima versione di ATutor è stata rilasciata alla fine del 2002. Come risposta allo studio effettuato dagli sviluppatori negli anni precedenti che ha mostrato come sul mercato non fossero presenti sistemi di apprendimento on-line che soddisfacessero i requisiti minimi di accessibilità, basti pensare che una persona non vedente che non poteva partecipare a pieno titolo ai corsi. Per ovviare a queste mancanze del mercato ATutor introdusse varie funzioni, tra cui le due più importanti: alternative di testo per ogni elemento visivo ed accesso da tastiera ad ogni elemento del programma. Queste due nuove funzionalità permettono ad un utente non vedente di poter “ascoltare” l'intera interfaccia del sistema con l'aiuto di un lettore di schermo e di poter accedere al sistema senza utilizzare il mouse. Queste caratteristiche fanno sì che il software si adatti ad una vasta gamma di tecnologie fra cui smartphone e PDAs (Personal Data Assistants). Naturalmente sono contenute anche altre varie funzionalità molto importanti di cui riporto solo le due più importanti:

- I. *strumento di creazione*: oltre a permettere la creazione di contenuti spinge gli sviluppatori a creare materiali didattici accessibili, esso include anche un servizio web implementato, contro i vari standard internazionali, per la valutazione dell'accessibilità. Questo strumento è di per sé accessibile, ovvero permette ad utenti con disabilità ,di creare contenuti;
- II. *adattabilità*: ATutor è stato concepito per adattarsi a qualsiasi tipo di insegnamento e scenari di apprendimento. Vi sono infatti implementate quattro aree che riflettono questa architettura. La prima è “Temi”, essa è implementata per fornire la possibilità ad ogni utente di personalizzare la propria versione del software modificando i layout o per dare un look diverso alle varie categorie dei corsi. La seconda riguarda i “Privilegi”, essa permette agli amministratori o anche ai singoli docenti di poter definire

dei privilegi di gestione degli strumenti per particolari utenti. Le altre due aree sono i “Moduli” degli strumenti e dei “Gruppi”.

Al momento della stesura di questo documento l'ultima versione rilasciata del software risale al 5 Ottobre 2012 con ATutor 2.1, essa ha apportato molte migliorie soprattutto in ambito della sicurezza, ma la novità maggiore è l'estensione multisito che consente ad una singola base di codice di ATutor di eseguire in molti siti secondari indipendenti [ChangeL].

1.3 Standard

Dai paragrafi precedenti abbiamo visto che i Learning Management System non necessitano strettamente di un collegamento ad Internet e sommato questo al fatto che esistono una ampia serie di LMS introduce il concetto del problema di interoperabilità già discusso precedentemente. AContent supporta quattro diversi standard di e-learning che mirano a risolvere il problema dell'interoperabilità fra le varie piattaforme di e-learning. I primi tre standard sono stati creati dalla IMS Global Learning Consortium, conosciuta con l'acronimo IMS GLC o ITIMS, una società globale senza scopo di lucro, che ha come obiettivo quello di consentire lo sviluppo e l'impatto delle tecnologie di apprendimento per l'educazione e la formazione [IMS GLC]. L'attività principale dei membri di questa organizzazione è quella di sviluppare standard di interoperabilità e norme pratiche di adozione per l'apprendimento distribuito, alcuni dei quali, come QTI e Content Packaging, sono ampiamente utilizzati.

1.3.1 SCORM 2004

Shareable Content Object Reference Model [SCORM] è un content package che consente l'apprendimento di contenuti tramite ogni strumento LMS che sia conforme a

SCORM. Al momento attuale le ultime specifiche dello standard sono relative alla versione 1.3, conosciuta con il nome di SCORM 2004, che attualmente è giunta alla quarta edizione, anche se la versione più utilizzata rimane la 1.2. SCORM nasce dalla cooperazione tra governo, mondo accademico e l'industria, il lavoro di questi tre enti porta ad unire diverse specifiche già esistenti (IEEE, IMS, AICC) creando un unico standard che permetta di rendere i vari learning object (LO) riutilizzabili e soprattutto interoperabili. Definisce le specifiche relative al riutilizzo, tracciamento e catalogazione dei LO. In questo modo le diverse piattaforme hanno solo il compito di dialogare con l'oggetto, interpretando i messaggi che gli vengono passati, questo è reso possibile in quanto SCROM definisce fra le sue norme le caratteristiche che dovrebbero essere supportate dal LMS, inoltre la compatibilità di quest'ultimi si rende necessaria solo per "comprendere la lingua" dell'oggetto. Le specifiche di SCROM sono raccolte in tre volumi che sono chiamati book.

- I. *the SCROM Overview*: contiene la descrizione dell'iniziativa.
- II. *the SCROM Content Aggregation Model*: contiene le regole per definire come aggregare ed integrare risorse educative per costruire e distribuire LO di varia dimensione a loro volta componibili.
- III. *the SCROM Run Time Environment*: contiene le specifiche per l'avvio, la comunicazione ed il tracciamento dei contenuti all'interno dell'ambiente di iterazione Web, ovvero la piattaforma LMS.

Dal punto di vista tecnico lo standard SCROM è suddiviso in tre sottospecifiche:

- I. *content packing*: in questa sottosezione sono raccolte tutte le specifiche inerenti a come il contenuto deve essere "impacchettato" e descritto; si fonda essenzialmente sullo standard XML. L'organizzazione dei contenuti viene fatta secondo la specifica chiamata PIF(Package Interchange File) [PIF] che prevede che i dati siano inseriti in una cartella auto contenuta o compressa in formato .ZIP, viene inoltre utilizzato un file XML denominato *imsmanifest.xml*, detto semplicemente manifest, che racchiude tutte le informazioni utili al LMS per fornire i contenuti. Inoltre, divide la lezione in più parti, dette SCO

che possono essere combinate in una struttura ad albero, (*activity tree*), che rappresenta la lezione. Il manifest contiene la rappresentazione dell'*activity tree* in formato XML ed informazioni sull'esecuzione dei vari SCO e non obbligatoriamente, metadati che fungono da descrizione per l'intera lezione e tutte le sue parti.

II. *Sequencing*: è una raccolta di attributi e set di regole scritte dall'autore in formato XML e contenute nel manifest. Alcuni esempi di queste specifiche sono:

- Definire se e quali attività devono essere completate prima di altre.
- Selezionare casualmente un sottoinsieme di SCO da mostrare ad ogni nuovo tentativo, ad esempio in fase di autovalutazione.
- Specificare quali controlli di navigazione, come pulsanti after/before, devono essere mostrati dal LMS all'utente.
- Riportare l'utente ad uno specifico argomento che non è stato compreso del tutto.
- Permettere di dare una diversa importanza ad alcune parti della lezione allo stato finale o nel punteggio, ad esempio fornendo la possibilità di creare sezioni facoltative o inserire domande riflessive.

III. *Run-Time*: contiene le specifiche su come il contenuto deve essere eseguito dal browser, sia in una finestra che in un frameset, e come comunica con il LMS. Si basa principalmente su ECMAScript. Il LMS può eseguire uno SCO alla volta mentre, durante l'esecuzione viene lanciato un algoritmo che determina uno script API ECMAScript fornito dal LMS. Lo script API permette, utilizzando le sue funzioni, lo scambio di dati con il LMS.

Come ultima cosa aggiungiamo che SCORM 2004 rispetto alla sua versione precedente ha adottato le specifiche IMS Simple Sequencing Specification che

dovrebbero garantire ai progettisti di LO standard un controllo maggiore sul cammino di apprendimento previsto per diversi utilizzatori.

1.3.2 IMS Content Packaging

Conosciuto anche con l'acronimo IMS CC [IMS CP] è uno standard che racchiude le specifiche su come devono essere definite le strutture dati utilizzate per fornire interoperabilità fra i contenuti web e i tool autore, inoltre sono specificate le regole dell'interoperabilità fra i Learning Management System e gli ambienti in cui sono eseguiti. Queste regole mirano alla creazione di strutture standardizzate per lo scambio di contenuti. Queste strutture non sono altro che la base fondamentale per i progettisti di software i quali, applicando le specifiche di questi standard, possono creare materiale didattico consultabile e modificabile anche su differenti tool di authoring. Questo permette di avere ambienti di esecuzione e LMS sviluppati in modo indipendente dal sistema da eseguire.

Infine possiamo dire che l'IMS Content Packaging viene utilizzato specificatamente nel campo della definizione dell'interoperabilità nei sistemi che vogliono poter esportare, importare, aggregare e disgregare pacchetti di contenuti. Fornisce, infatti, le specifiche su come strutturare e racchiudere blocchi di materiale didattico, come ad esempio un insieme di corsi "impacchettati" che possono essere distribuiti fino a come impacchettare la singola lezione.

1.3.3 IMS Common Cartridge

L'avvento di questo standard [IMS CC] è riconducibile al 1997 quando NLII(National Learning Infrastructure Initiative of Educause) avviò un progetto per la creazione di uno standard per lo scambio di contenuti, le università fondatrici furono la California State, Michigan e North California. Oggi l'IMS CC fornisce un formato standard per la creazione e condivisione di contenuti digitali. L'ultima versione è la 1.1 rilasciata il 10 Maggio 2011, la gestione delle specifiche e degli aggiornamenti sono

gestite dal CCAPMG(Common Cartridge Accredited Profile Management Group). Lo standard IMS Common Cartridge permette una stretta interoperabilità di contenuti e sistemi e garantiscono una grande flessibilità nella tipologia stessa dei contenuti supportati. IMS Common Cartridge specifica cinque punti fondamentali:

- I. uno standard per i metadati che descrivono i contenuti-basato su Dublin Core [DC], Common Cartridge, è comunque estendibile e permette altri schemi di metadati;
- II. uno standard per i forum on-line destinati alla collaborazione tra studenti, permettendo il pre-inserimento di esercizi, discussioni, argomenti e così via; un formato per lo scambio dei dati tra sistemi, in modo da far sì che ci sia un modo comune di interpretare ed organizzare i contenuti scambiati. Questo è possibile utilizzando un manifesto che descrive il contenuto, le sue componenti possono essere modificate all'interno del pacchetto scambiato o esternamente al pacchetto, utilizzando il riferimento URL. Questo permette al contenuto o alle applicazioni che richiedono una licenza di essere raggruppate in modo flessibile a contenuti non protetti;
- III. uno standard per elementi di valutazione e di test che permetta al sistema di apprendimento di riuscire a comprendere valutazioni importate e native in modo che possano poi essere modificate a seconda delle necessità del sistema stesso;
- IV. uno standard per lo scambio e l'esecuzione di dati con applicazioni esterne in modo che possano essere parte di una singola esperienza didattica gestita per mezzo di un LMS. Tramite questo supporto è possibile avere diversi tipi di applicazioni situate in qualsiasi luogo, come ad esempio tutor personalizzati, social networking, wiki, sistemi di valutazione esterna, diverse varietà di librerie web-based di contenuti o altri sistemi di questo tipo;

SCORM e IMS Common Cartridge hanno uno stretto legame e possiamo individuare alcune caratteristiche comuni sia nella struttura dei rispettivi content package, sia nell'organizzazione dei contenuti. In origine Common Cartridge è stato sviluppato

principalmente per supportare l'utilizzo di libri e materiali digitali in un contesto didattico, mentre SCORM è stato implementato per supportare la portabilità di contenuti di apprendimento autocontenuti. L'idea che guidò la progettazione di Common Cartridge non era quella di creare un software che sostituisse SCORM. Nonostante questo gli scenari educativi esistenti necessitano di analisi sui progressi di apprendimento, la definizione di una sequenza di contenuti, aiuti, collaborazione, l'analisi sui contenuti interattivi e autorizzazioni a cui SCORM non è in grado di soddisfare, al contrario di Common Cartridge.

Considerando la funzionalità dell'interoperabilità Common Cartridge è stato sviluppato per superare SCORM rimuovendo le componenti run-time e raggiungendo un accordo su specifici profili di applicazione, ovvero sottoinsiemi di specifiche ampiamente utilizzate.

Common Cartridge consente la creazione di sequenze complesse di contenuti, grazie all'utilizzo degli strumenti LTI di cui faremo una panoramica nei paragrafi successivi. In conclusione i progettisti che si attengono alle specifiche di Common Cartridge hanno la possibilità di accordarsi su approcci specifici dei pacchetti di contenuti, elementi di valutazione e autorizzazioni. Questa strada permette una notevole semplificazione per quanto riguarda i test di interoperabilità così da poter effettuare valutazioni di conformità attraverso suite di strumenti informatici. Secondo lo stesso IMS Global Learning Consortium i principali vantaggi e benefici introdotti con Common Cartridge sono:

- I. maggiori opzioni di valutazione;
 - II. scelta più vasta di contenuti;
 - III. maggiore flessibilità, condivisione e riuso;
 - IV. stabilisce i formati nativi di Common Cartridge approvati dal mondo accademico e consente l'utilizzo di vari formati prestabiliti per i contenuti.
- Questo permette di arginare la chiusura del software;

L'ultima precisazione su questo standard è che fornisce un nuovo formato di scambio dei materiali di apprendimento in grado di funzionare su qualsiasi piattaforma Learning Management System compatibile. Esaminando la versione 1.0 possiamo vedere le

funzionalità supportate:

- **Rich content:** HTML 4.01, XML, web links, media files e application files;
- **Valutazioni integrate:** scelta multipla, vero o falso, tema o saggio breve;
- **Forum di discussione;**
- **Metadati:** sia metadati descrittivi per Common Cartridge, sia basati sui ruoli per le risorse;
- **Autorizzazione per i contenuti protetti;**

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

1.3.4 Learning Tools Interoperability

IMS Learning Tools Interoperability [IMS LTI] è stato creato da IMS Global Consortium all'inizio del 2010 con l'obiettivo di favorire la comunicazione tra due o più entità è strutturato secondo regole formalmente definite. L'idea che ha portato alla creazione di questo strumento deriva dal fatto che diversi learning management system hanno uno o più moduli per gestire l'interazione con altri LMS vincolando il fornitore o il fruitore dei contenuti a supportare lo stesso modulo.

Per ovviare a questo problema si è deciso di adottare come soluzione un unico plugin per tutti i learning management system; LTI si pone come un Web Service [WS] che si interfaccia ai contenuti che possono essere ospitati sullo stesso computer, su un server, in una rete cloud. L'impiego specifico del learning tools interoperability è quello di creare questa connessione "sicura" (poiché non risolve le problematiche legate alla sicurezza informatica) fra vari sistemi senza dover creare uno specifico software per ogni iterazione. Dal punto di vista generale LTI si pone l'obiettivo di facilitare l'interazione di applicazioni ricche di contenuti didattici con piattaforme di LMS, portali o qualsiasi altro ambiente educativo. In base al formalismo definito, le applicazioni educative sono chiamate Tool e sono fornite dal Tool Provider (TP),

mentre gli LMS o piattaforme sono denominate Tool Consumer (TC). Lo strumento che abbiamo appena visto agevola servizi comunemente utilizzati fra cui:

- I. Web Service SOAP: [WSS] usato per la risposta dal TP al TC;
- II. Web Service REST: [WSR] usato per la richiesta del TC al TP;

IMS definisce LTI come uno strumento affidabile, scalabile e coerente che sostiene un modello di LMS solido e centrale. Allo stesso tempo lo pone come uno strumento innovativo perché prevede che nel futuro prossimo, all'incirca nel prossimo quinquennio, si avrà un notevole e critico incremento dei learning management system aziendali.

1.3.5 IMS QTI Test

Question and Test Interoperability [IMS QTI] è una raccolta di specifiche che definiscono un linguaggio XML per lo scambio di test e domande tra diversi LMS. Nasce dal consorzio industriale e accademico IMS Global Learning Consortium e consistono in uno standard di dati che definiscono la struttura delle domande e dei test e ne specificano il collegamento in XML. La specifica consiste in un modello di dati che definiscono la struttura di domande, valutazioni e risultati insieme ad una associazione di dati XML che definisce essenzialmente un linguaggio per scambio di informazioni e materiale di valutazione diversa. L'associazione XML è ampiamente usata per lo scambio di informazioni tra i diversi strumenti di authoring e dagli editori. Le parti di valutazione e sui risultati del disciplinare sono meno diffuse. E' largamente utilizzato per lo scambio fra diversi creatori di test e diversi autori. La prima versione 1.0 era basata sul linguaggio QML creato da Questionmark [QM], ma negli anni il linguaggio si è evoluto notevolmente permettendo di descrivere qualsiasi domanda un utente voglia creare. Gli ultimi release risalgono al 2002 per la versione 1.2 e al 2005 per la versione 2.0.

Capitolo 2

Progetto

Il progetto di tesi presentato si pone come obiettivo quello di creare una nuova estensione denominata *Wizard*. Essa avrà il compito di creare un percorso guidato che supporta l'utente durante la creazione di contenuti con AContent.

L'idea di base è quella di implementare una nuova funzionalità orientata a tutte le tipologie di utenti che interagiscono con la piattaforma. Essa consentirà ad utenti che si avvicinano per la prima volta a questa piattaforma di poter imparare a creare i propri materiali didattici attraverso una successione di passi guidati, e allo stesso tempo funga da tutor di insegnamento, per far sì che l'utente al di fuori dell'estensione *wizard* possa riconoscere le diverse funzioni di creazione ed eseguirle in perfetta autonomia. Offrirà allo stesso tempo la possibilità ad utenti esperti di creare lezioni articolate sfruttando le nuove potenzialità di questa estensione, che migliora le preesistenti funzionalità di creazione di lezioni di AContent.

Il *wizard* è stato strutturato in modo da permettere l'aggiunta di opzioni e/o vincoli. Quelli per ora pensati riguardano: la possibilità di associare obiettivi ai contenuti per aderire a percorsi di apprendimento definiti; il vincolo di inserire indifferentemente contenuti accessibili dai bisogni della persona ed, infine, la possibilità di associare un copyright (diritto di copia) ed immagini, video o testi.

2.1 AContent

Nel corso del primo capitolo abbiamo visto una ampia gamma di Authoring tool disponibili sul mercato, ma non ci siamo soffermati ad elencare i punti secondo i quali è opportuno scegliere un determinato strumento autore rispetto ad un altro. Adottare un tool rispetto ad un altro influisce sulla qualità dell'output finale, in particolar modi in ambienti di sviluppo nei quali i contenuti creati devono essere esportati e condivisi su larga scala. Nel nostro caso specifico sono stati tre i punti fondamentali che hanno pilotato la scelta di un determinato authoring tool.

- I. *Open Source*: il software deve avere questo tipo di licenza per permettere agli sviluppatori di analizzare il funzionamento interno e di poter modificare il codice sorgente.
- II. *Output*: l'authoring tool deve produrre in output contenuti conformi ai principali standard per l'e-learning per permettere la portabilità attraverso diverse piattaforme.
- III. *Moodle compatibile*: lo strumento deve essere accessibile e supportare la creazione di contenuti accessibili in modo da essere esportato ed importato in Moodle.

La soddisfazione di questi punti hanno pilotato la scelta su AContent [AC1], un software open source di creazione di contenuti didattici utilizzato per generare materiali di apprendimento interoperabili, accessibili, adattivi, basati sul web. È doveroso infine ricordare l'ultima motivazione per cui si è adottato AContent, ovvero durante la fase di scelta è stata presa in considerazione la Legge Stanca [LStanca1] del 9 Gennaio 2004 omonima del suo promotore, Lucio Stanca, allora Ministro per l'Innovazione e le Tecnologie, essa raccoglie in se numerose proposte di legge sull'accessibilità proveniente da numerosi schieramenti politici. Essa prende il nome completo di "*Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici*" [LStanca2]. I suoi obiettivi e finalità sono riportate direttamente nel testo dalla

Gazzetta Ufficiale n.13 del 17 gennaio 2004.

Art. 1.

(Obiettivi e finalità)

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.

Ma il punto fondamentale che ha veicolato la scelta è stato indubbiamente il punto 1 dell'articolo 5 di suddetta legge.

Art. 5.

(Accessibilità degli strumenti didattici e formativi)

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

2.1.1 Caratteristiche

AContent è un software open source dedicato all'Authoring, esso è basato sul web e quindi è applicabile a qualsiasi piattaforma. È un sistema monolitico, ovvero definisce un'interfaccia virtuale di alto livello sull'hardware, nonostante ciò non soffre dei problemi tipici di questi sistemi, permettendo una semplice integrazione di blocchi di codice sviluppato da terzi. Può essere usato insieme a sistemi di gestione per sviluppare, condividere e archiviare i materiali di apprendimento. L'interoperabilità è evidente durante la fase di esportazione dei contenuti generando un tool autosufficiente da usare con tutti i sistemi che permettono l'importazione di contenuti compatibili con gli standard IMS. In particolare gli standard supportati per gestire l'importazione e l'esportazione dei contenuti sono i seguenti:

- IMS Content Package;
- IMS Common Cartridge;

- IMS QTI;
- W3C XHTML 1.1;
- WCAG 2.0.

Mentre per quanto riguarda la creazione dei contenuti AContent fornisce le seguenti funzionalità implementative:

- Testo e link;
- file multimediali come immagini, video e audio;
- tutti i linguaggi di markup e CSS per la modifica della formattazione dei contenuti;
- LaTeX [Latex] per equazioni e notazione scientifica.

AContent nasce come modulo del LCMS ATutor, ma nel corso del tempo e con il susseguirsi di rilasci di nuove versioni molto più complesse ed articolate dell'allora modulo, l'organizzazione creatrice di ATutor e AContent ha deciso di separare i due strumenti. Al momento attuale esiste ancora la versione implementata come modulo di ATutor, ma solo per problemi legati alla retro compatibilità. AContent è diventato uno strumento di authoring completamente indipendente.

Alla data di stesura di questo documento l'ultimo release risale al 7 Dicembre 2012, con la versione BETA 1.3, mentre l'ultima versione definitiva, la 1.2, è stata rilasciata il 14 Ottobre 2011. AContent resta tuttora un software in pieno sviluppo, infatti il progetto su cui si basa questa tesi rientrerà in una release futura.

Nella tabella seguente abbiamo raccolto tutte le principali caratteristiche del tool autore:

CARATTERISTICA	VALORE
Licenza	Open Source
Tipologia Applicazione	Web Application
Architettura	Monolitico
Linguaggio di Programmazione	Php , Javascript
Formati in Input e Output	IMS Common Cartridge , SCROM 2004 , IMS QTI Test
Accessibilità	WCAG 2.0

Tabella 1: Caratteristiche principali AContent 1.2

2.1.2 Tecnologie

AContent è una piattaforma di authoring tool che utilizza una vasta gamma di tecnologie, ma per non inserire in questo documento materiale non inerente al progetto (che verrà trattato nel proseguo) si è deciso di trattare in questo capitolo solo due aree tecnologiche fondamentali. Nella prima faremo una panoramica sui linguaggi di programmazione utilizzati in AContent e nell'implementazione del progetto di questa tesi, mentre nella seconda area analizzeremo il database, strumento che questa piattaforma utilizza ampiamente per gestire vari aspetti.

2.1.2.1 Linguaggi di programmazione

AContent è un software complesso ed articolato progettato facendo interagire diversi linguaggi di programmazione che permettono di gestire tutti gli aspetti fondamentali di una piattaforma di e-learning. Di seguito vediamo la carrellata di tutti i linguaggi utilizzati:

- **PHP** : Hypertext Preprocessor [PHP1], è il codice su cui si basa AContent ed è un linguaggio di programmazione interpretato con licenza open source e libera. Nato nel 1994 grazie al danese Rasmus Lerdorf era in origine una raccolta di script CGI che permettevano una facile gestione delle pagine personali. Il pacchetto originario venne in seguito esteso e riscritto dallo stesso Lerdorf in C, aggiungendo funzionalità quali il supporto al database mSQL e prese a chiamarsi PHP/FI, dove FI è l'acronimo di Form Interpreter, prevedendo la possibilità di inserire codice PHP nel codice HTML in modo da semplificare la realizzazione di pagine dinamiche. Nel 1998 grazie alla collaborazione dei due programmatori Zeev Suraski e Andi Gutsman con l'ideatore Lerdorf nasce la versione PHP 3.0, dove viene riscritto il motore che prende il nome di Zend [Zend] (dalla contrazione dei loro nomi). In questi anni questa versione è stata installata nel 10% dei server web presenti su internet. A questo punto PHP è in grado di competere con la sua versione concorrente implementata da Microsoft e conosciuta con il nome ASP. Negli anni successivi sono seguite due nuove

versioni, la 4.0 nel 2000, che prevedeva notevoli migliorie, ampliate notevolmente nell'ultima versione la 5.0 rilasciata nel 2004 e sviluppata da un team di programmatori che comprendono ancora i Lerdorf, Suraski e Gutsman [PHP2]. Dal febbraio 2008 la versione PHP 5.0 è l'unica versione stabile in fase di sviluppo e sempre da questa data molti dei progetti open-source di alto livello hanno cessato di supportare la versione 4.0 nelle nuove versioni del codice promuovendo così la versione PHP 5.0. Nel 2010 è avvenuto il rilascio della versione 5.3. Quando ci riferiamo a PHP intendiamo un linguaggio di scripting, impiegato principalmente sul lato server, utilizzabile per generare dinamicamente informazioni in HTML. PHP è connesso ad un server web, tipicamente Apache o IIS(Internet Information Server) e dopo aver correttamente creato il codice html lo invia al server che si occuperà automaticamente del rinvio al client. Nonostante il suo uso principale sia sui server web esso è utilizzato anche in altri contesti come PC desktop o riga di comando.

- **HTML e CSS** : il primo, Hypertext Text Markup Language [HTML1], non è un linguaggio di programmazione vero e proprio in quanto non prevede la definizione di variabili, strutture dati, funzioni o strutture di controllo. Esso è infatti definito come linguaggio di markup che definisce le modalità di impaginazione, formattazione o visualizzazione grafica del contenuto, testuale e non, di una pagina web attraverso tag di formattazione e l'integrazione del linguaggio informatico Cascading Style Sheets (CSS) che fornisce tutte le regole per la formattazione dei layout di una pagina web. L'HTML è un linguaggio di pubblico dominio la cui sintassi è stabilita dal W3C (World Wide Web Consortium), e che è basato su un altro linguaggio con obiettivi più generici, l'SGML. Nasce verso la fine degli anni ottanta dall'autore Tim Berners-Lee al CERN di Ginevra e ottiene una grande diffusione nel 1994 in seguito ai suoi primi utilizzi in ambito commerciale nel web. Nel corso degli anni ha subito notevoli aggiornamenti indicati secondo la classica notazione numerica dei software, fino ad arrivare all'ultima versione rilasciata, la 4.01, il 24 Dicembre 1999. L'ente W3C dopo anni di progettazione dedicata alle definizioni di XHTML(applicazioni a HTML di regole e sintassi in stile XML) e dei fogli di

stile CSS versione 3, nel 2007 è iniziata la progettazione di HTML 5, (ancora allo stadio di bozza). Attualmente i documenti HTML sono in grado di incorporare nuove tecnologie per associare al documento ipertestuale controlli più sofisticati sulla resa grafica [HTML2]. Nel nostro caso il software che stiamo esaminando non utilizza file HTML puri, ma incorpora i suoi tag all'interno di file PHP o Javascript.

- **Javascript** : è un linguaggio di scripting orientato agli oggetti [JS], la sua caratteristica principale è di essere un linguaggio interpretato, ovvero il codice non viene compilato, ma ben si interpretato sul lato client, da un interprete incluso nei browser, la sua sintassi è simile a quella del comune C, C++ e del Java. Definisce le funzionalità tipiche dei linguaggi ad alto livello e consente l'utilizzo del paradigma object oriented. Al contrario del php è un linguaggio lato client. Progettato da Brendan Erich della Netscape Communications con il nome originario di Mocha e successivamente di Live Script. Il cambio del nome nell'attuale javascript avvenne quando la Netscape incluse il supporto per la tecnologia Java nel suo browser Navigation, il cambio di nominativo creò molte polemiche in quanto non vi è una relazione vera e propria tra Java e Javascript, le loro semantiche sono differenti e i loro object model non hanno relazioni e sono totalmente incompatibili, l'unica somiglianza che si ravvisa in questi due linguaggi è nella sintassi, infatti entrambi derivano dal linguaggio C. In AContent viene utilizzato per gestire runtime l'iterazione fra l'utente ed il software creando una sinergia dinamica fra essi.

All'interno del software si utilizza la libreria di javascript chiamata JQuery, pubblicata per la prima volta il 22 agosto 2005 da John Resing, ha raggiunto la sua prima versione stabile il 26 agosto del 2006. Essa permette di semplificare la programmazione lato client, e permette di gestire gli eventi in maniera automatica con l'utilizzo di AJAX che implementa richieste ricorsive, permette il caricamento di contenuti dinamici e l'interazione con javascript.

- **SQL** : Structure Query Language [SQL] è un linguaggio standardizzato per database basati sul modello RDBMS progettato per supportare tutti e quattro i

più importanti linguaggi di programmazione dei database: Data Definition; Manipulation Query e Control Language. Non si tratta, quindi, di un semplice linguaggio query, ma include anche gli altri linguaggi tipici per la programmazione e gestione/amministrazione del database. L'SQL nasce nel 1974 dal suo ideatore Donald Chamberlin, nei laboratori dell'IBM, il suo primo nome fu SEQUEL ed era un linguaggio progettato per lavorare con database relazionali, nel 1975 viene sviluppato il prototipo SEQUEL-XRM, grazie al quale nasce nel '77 la nuova versione del linguaggio che avrebbe dovuto chiamarsi SEQUEL/2, ma che per problemi legali prese il nome di SQL. A partire dagli anni ottanta divenne il linguaggio standard, precisamente nel 1986 fu adottato dall'ANSI (American National Standards Institute) e nell'anno successivo l'ISO (International Organization for Standardization) fece lo stesso, per i modelli relazionali. Questo processo di standardizzazione non riuscì tuttavia a creare un linguaggio utilizzabile su tutti i DBMS relazionali perché i vari produttori implementarono varie versioni del linguaggio apportando ognuno varie modifiche limitandosi ad adottare gli standard ad un livello non superiore al minimo, definito dall'ANSI come Entry Level.

2.1.1.2 Database

Come abbiamo già esposto nei capitoli precedenti AContent è indipendente da qualsiasi sistema su cui lo si sta eseguendo. Al momento dell'installazione il software procede a creare automaticamente un database nel quale vengono memorizzate tutte le informazioni inerenti alle strutture utilizzate nel software e che in seguito ospiterà anche tutti i dati inseriti dai vari utenti che interagiranno con AContent.

Sul sito ufficiale atutor.ca/acontent/ nella sezione dedicata al download è possibile vedere i tre sistemi suggeriti per la gestione del database, suddivisi in base al sistema operativo che si ha in dotazione. Vediamo quali sono questi software:

- **EasyPHP** : dedicato ai sistemi Windows, è un ambiente di sviluppo web-

database del tipo WAMP [WAMP] che permette di far eseguire un server web basato sull'interprete Php. Messo sul mercato nel 1999 dagli sviluppatori Laurent Abbal, Emmanuel Faivre e Thierry Murail è stato il primo pacchetto WAMP.

EasyPHP [EPHP] non è un semplice software, ma un ambiente che comprende un server web Apache, un server MySQL, un interprete di script Php e un amministratore di database MySQL dotato di una interfaccia grafica chiamata phpMyAdmin, che permette di gestire gli utenti, l'avvio e lo spegnimento dei server. Tutte queste componenti sono installate contemporaneamente e forniscono la possibilità di sviluppare in locale siti web in Php, infatti il server Apache crea automaticamente di default un dominio virtuale raggiungibile dall'indirizzo *localhost*. L'ultimo aspetto interessante di questo ambiente è che può essere eseguito come una applicazione portatile, attraverso l'uso di una chiavetta USB.

- **XAMPP:** l'acronimo sta per “ X:Cross-platform A:ApacheHTTP Server M:MySQL P:Php P:Perl”, è una piattaforma software gratuita costituita da Apache HTTP Server, il database MySQL e tutti gli strumenti necessari per supportare i linguaggi di programmazione Php e Perl [Perl]. È rilasciato sotto la GNU General Public License ed oltre ad essere gratuita è caratterizzata da un approccio user friendly. È stata sviluppata da Kay ‘Oswald’ Seidler e Kay Vogelgesang . Può essere eseguita su tutti i sistemi operativi, da Windows a Sun Solaris passando da Mac OS X e senza dimenticare GNU/Linux.
- **MAMP:** [MAMP] al contrario dei due precedenti è utilizzata principalmente come piattaforma di sviluppo e raramente come ambiente di produzione. Il suo acronimo significa ”M:Mac OS X A: Apache M:MySQL P:Php o Perl o anche Python” [Python], le componenti sono le stesse dei due software precedenti e svolgono esattamente le stesse funzioni. MAMP è rilasciato in due versioni, la prima con il nome di MAMP con licenza GNU General Public License, mentre la seconda è una versione commerciale con il nome di MAMP PRO [MPRO].
Tutti e tre i software che vengono suggeriti per supportare AContent sfruttano le

potenzialità di MySQL [MYSQL]. MySQL, chiamato Oracle MySQL, è un Relational Database Management System (RDBMS), composto da un client a riga di comando e un server. Il codice di questo software venne implementato già a partire dal 1979 dalla ditta TcX ataconsult, poi rinominata MySQL AB, ma solo nel 1996 fu rilasciata una versione che supportava l'SQL, questo rilascio supportava la maggior parte della sintassi SQL. Il software fu acquistato nel 2008 dalla Sun Microsystems e due anni dopo nel 2010 questa azienda fu incorporata dalla Oracle Corporation. Per quanto riguarda la mia esperienza personale ho utilizzato XAMPP durante la progettazione dell'estensione, la mia scelta è ricaduta su questo strumento perché era quello utilizzato dagli altri sviluppatori ed anche perché compatibile con tutti i sistemi operativi.

2.1.3 I template

Una funzionalità fondamentale di AContent, che verrà sfruttato ampiamente nel corso di questo progetto, riguarda i template. Nell'ambito informatico si intende un template come un modello predefinito che può essere utilizzato come punto di partenza per la creazione di un documento. La definizione generica appena esposta può assumere naturalmente diverse sfaccettature a seconda degli ambienti di utilizzo dei template: possiamo avere template grafici che propongono varie vesti al documento, template che offrono modelli di risoluzioni di problemi o template che offrono modelli organizzativi per determinare la struttura dei documenti etc. I vantaggi offerti dall'utilizzo dei template sono principalmente dati dal fatto che è possibile riapplicarli in diverse situazioni simili dovendoli definire una sola volta e che consentono un risparmio di tempo nella creazione e nella definizione di una struttura che rispetti determinati formalismi.

In AContent i template sono stati aggiunti senza la necessità di integrare anche un motore di template, ma utilizzando strutture dati esistenti e specifici standard di e-learning. Il gruppo chiamato Bologna Elearning Authoring Tool [BEAT], ha deciso di organizzare i servizi aggiuntivi da offrire agli autori utilizzando tre approcci diversi.

La soluzione scelta è stata strutturata secondo i seguenti livelli di template adottati nella versione 1.2:

- **Template di Layout:** questo primo livello consente all'utente di applicare una veste grafica al proprio contenuto. Al momento AContent ne fornisce cinque: *Canada, Italy, Unibo, Seti e Microsoft Windows* . Ognuno di essi è implementato attraverso uno specifico foglio di stile (CSS). I vantaggi di adottare un set di template già pronti sono principalmente due. Il primo è dato dal fatto che è possibile sapere a priori gli effetti che essi avranno all'interno del software, mentre il secondo è che ci forniscono una soluzione semplice e facilmente utilizzabile anche da tutti coloro che non sono pratici di informatica;
- **Template di Pagina:** questo livello agisce sull'organizzazione del contenuto della pagina. AContent fornisce vari template di pagina i quali differiscono in base alla disposizione del testo, che può essere influenzata dalla presenza di una immagine, di un elenco puntato o di un titolo. I template grafici non inseriscono contenuti grafici né testuali, ma offrono un supporto organizzativo per la struttura del contenuto. Il vantaggio principale dell'utilizzo di questi template risiede nel fatto che essi offrono una soluzione veloce per organizzare il proprio contenuto, offrendo comunque la possibilità di modificare la struttura selezionata aggiungendo parti o eliminandole.
- **Template di Struttura:** quest'ultimo è stato creato prendendo spunto da altri authoring tool fra cui Microsoft LCDS [LCDS]. La funzione di questo strumento è quella di fornire, in fase di creazione di una lezione, la possibilità all'utente di associare al suo contenuto una struttura predefinita, con la possibilità di partire poi da quella per gestire la propria lezione. I tre template di struttura offerti da AContent sono *Knowledge, Creative e Meta-Competency*. A queste tre strutture sono stati assegnati degli obiettivi didattici. Questi possono essere scelti durante il wizard e permettono all'utente vedere quali obiettivi didattici. Questi possono essere espressi con l'utilizzo delle strutture predefinite di AContent. Il vantaggio di adottare una soluzione di questo tipo è che si permette all'utente di strutturare la propria lezione con un semplice passaggio.

I vantaggi che nascono dall'utilizzo di un sistema di template sono dati dalla possibilità per gli autori di usufruire di un supporto grafico per la presentazione e per l'organizzazione dei contenuti, incrementando l'usabilità dei tool autore e offrendo nuove possibilità per la produzione di materiale didattico. La creazione di un set di regole da seguire per la definizione dei template è necessaria ogni qual volta si voglia che i contenuti siano prodotti su larga scala. Questi template introdotti permettono di curare oltre che alla grafica, una strutturazione di alcune caratteristiche didattiche dei contenuti (valutazione, metodologia e descrizione degli obiettivi). Utilizzando questo approccio gli autori hanno un supporto durante la fase di redazione, prestando attenzione sia all'aspetto didattico che pedagogico. I tre template sono indipendenti l'uno dall'altro, questo sta a significare che possono essere utilizzati singolarmente e senza alcuna dipendenza.

2.1.4 Approcci di creazione dei contenuti

L'autoring tool AContent offre la possibilità ai propri utenti di inserire materiale didattico nelle lezioni che essi stanno creando, attraverso due approcci molto diversi. Le due possibilità offerte dal software sono quella denominata "Manually" e quella chiamata "Structure". Possiamo vedere nelle immagini sottostanti come si presenta la schermata di AContent per la creazione dei contenuti all'interno di una lezione.

How to create lesson content

Manually

Structure

Create Lesson: Manually [create a content page](#) or [create a content folder](#).

These tools are also available at side menu "Content Navigation" as "Add top page" or "Add top folder" links.

(a) Schermata approccio Manually

How to create lesson content

Manually

Structure

Create Lesson: [select one structure](#) to create a structured lesson

(b) Schermata approccio Structure

Figura 2.1: Schermata approccio Manually e Structure

Vedremo di seguito una descrizione di questi due approcci per meglio chiarire la decisione della creazione della nuova funzionalità Wizard.

Manually

L'approccio manuale all'interno del processo di creazione di una lezione permette all'utente di inserire nella propria lezione un singolo contenuto, scelto tra una cartella ed una singola pagina, come si può vedere nella Figura 2.1 (a). Da questa osservazione possiamo notare come questo approccio offra visibili vantaggi, fra cui la semplicità di creare un contenuto da parte di un utente esperto, ma anche palesi limitazioni fra cui l'impossibilità di creare una struttura articolata o di crearne anche una semplice che contenga più di un contenuto. E' altresì vero, che questo approccio può anche essere considerato solo un punto di partenza per creare il primo elemento didattico della lezione, ma è altresì vero che questa soluzione si esaurisce subito dopo la creazione del contenuto e l'utente viene lasciato "solo" nel proseguo della creazione dei contenuti della lezione.

Structure

L'approccio strutturale è fondamentalmente diverso da quello manuale appena visto e sopperisce alle mancanze del primo. Come possiamo vedere dalla Figura 2.1 (b) l'utente viene indirizzato attraverso il link ad una nuova schermata, visibile nella figura sottostante, Da questa schermata selezionando una delle tre strutture predefinite (creative, knowledge based, meta-competency based), delle quali è possibile vedere

l'effettiva composizione e ricreare all'interno della propria lezione la struttura desiderata. Questo tipo di approccio permette in maniera veloce all'utente di creare all'interno della propria lezione una struttura predefinita, contenente cartelle pagine e anche test. L'approccio structure pone rimedio alla mancanza dell'approccio manually per quanto riguarda la creazione di contenuti articolati all'interno della lezione, ma ne introduce altri, legati all'impossibilità di poter creare in piena libertà la struttura della propria lezione, perché essendo la riproduzione di un percorso didattico, contiene dei tipi di contenuti non eliminabili.

Edit content structure

*** Choose the structure**

creative based
Description: creative based
[\[show outline \]](#)

knowledge based
Description: knowledge based
[\[hide outline \]](#)

- knowledge
 - overview !
 - goals !
 - references !
 - contents 1 !
 - contents 2 ✘
 - content 2.1 !
 - content 2.2 !
 - content 2.3 ✘
 - self_assessment !
 - self_assessment ✓
 - forum ✘
 - retrieval_for_learning ✘

meta-competency based
Description: meta-competency based
[\[show outline \]](#)

Save

Figura 2.2: Schermata di scelta di una delle tre strutture predefinite di AContent

Inoltre, un altro svantaggio da non sottovalutare è che un utente inesperto può si creare una struttura articolata in maniera veloce e semplice, ma non assimilerà i passaggi necessari per creare gli specifici contenuti, rimanendo incapace di creare lezioni strutturate in autonomia, senza utilizzare l'approccio “structure”.

2.2 Wizard

Dopo aver visto i due metodi forniti da AContent per la creare le lezioni, con i loro pregi e difetti, possiamo introdurre il concetto che ha guidato lo sviluppo di questa tesi. L'idea di utilizzare *Wizard* nasce dall'esigenza di sviluppare una nuova funzionalità che premetta agli utenti di creare dei contenuti nelle proprie lezioni attraverso una nuova tipologia, completamente diversa da quella *manually* e quella *structure*. L'idea alla base di questo nuovo approccio e che ci ha guidato durante la progettazione di questa estensione di AContent è quella di creare un nuovo strumento che racchiuda i vantaggi dei suoi predecessori e che elimini le loro carenze. Dall'analisi dei due approcci preesistenti sono emersi tre problematiche fondamentali da affrontare:

- I. non vi è un meccanismo di apprendimento dei metodi di creazione dei contenuti;
- II. poca flessibilità nella modifica delle strutture;
- III. troppa libertà nella creazione manuale delle strutture.

Per quanto riguarda il primo punto direi che non vi sono obiezioni da sollevare, in quanto in entrambi i due approcci che abbiamo elencato nel paragrafo precedente non vi sono ne informazioni su come poter creare un determinato contenuto, ne indicazione su quali azioni compiere per crearli. Al contrario il secondo punto offre vari punti interrogativi in quanto alcuni potrebbero obiettare che utilizzando l'approccio *structure* è possibile modificare la struttura della lezione dopo averla creata. Questo è vero solo in parte, in quanto da come possiamo vedere dalla Figura 2.2, alcuni contenuti delle strutture predefinite, ovvero quelli che possiedono il simbolo del punto esclamativo accanto al nome, non possono essere eliminati, in quanto in queste strutture predefinite sussistono dei vincoli per i quali alcuni dei contenuti devono per forza esistere. Oltre a questa problematica, che senza dubbio influisce negativamente sulla flessibilità dell'approccio ve ne è un'altra, che si collega poi al primo punto, ovvero che l'utente inesperto dovrà andare a modificare la struttura della sua lezione interagendo con un software che non conosce e che non offre nessun supporto per imparare come eliminare o aggiungere dei contenuti ad una struttura. Dall'analisi di queste carenze nasce l'idea di implementare un nuovo approccio che sopperisca a questi problemi e che fornisca un

nuovo metodo per la creazione dei contenuti all'interno delle lezioni. Sono stati seguiti due concetti fondamentali durante la progettazione di questo nuovo strumento, il primo è quello di tutor mentre il secondo è quello della praticità. Vedremo nel proseguo cosa racchiudono questi due concetti.

Guida

La definizione di Guida, sta a significare “*lo strumento il quale indica la via da percorrere per arrivare all'obiettivo prefissato*”. Wizard nasce proprio seguendo questa definizione. Esso si pone come obiettivo quello di creare una struttura guidata formata da una successione di passaggi, ognuno adeguatamente descritto per informare l'utente su le operazioni che può svolgere e le conseguenze che esse comportano. Prima dell'implementazione di wizard all'interno del software un utente dopo che aveva creato la propria lezione, sia utilizzando *manually* e *structure*, poteva apportare modifiche ai propri contenuti sfruttando delle icone grafiche, che venivano visualizzate nel momento in cui si accedeva alla lezione (vediamole nelle figure successive).



(a) Icone presenti nella sezione centrale



(b) Icone presenti nel tab laterale sinistro

Figura 2.3: Icone presenti nella sezione centrale e nel tab laterale sinistro

Questa soluzione, offre la possibilità all'utente di usare tutta l'ampia gamma di

funzioni di creazione, modifica o eliminazione di AContent, semplicemente cliccando sull'icona desiderata. Tuttavia è una soluzione efficace per tutti gli utenti che conoscono già il software, mentre rimane di difficile comprensione per coloro che si avvicinano per la prima volta ad AContent. Naturalmente questo approccio non verrà cambiato da wizard, ma in esso, all'interno dei vari passi vengono richiamate queste icone, associandole alle funzioni che esse svolgono allegandovi anche una piccola descrizione. In questo modo si favorisce l'apprendimento dell'utente facilitando l'associazione visiva dell'icona con la sua funzione, permettendo una volta fuori da wizard e trovandosi di fronte alle icone presenti nella Figura 2.3 di interagire facilmente con loro sfruttandone appieno le loro funzionalità, precedentemente imparate nei vari passaggi.

Praticità

Il secondo concetto che è stato seguito per l'implementazione di wizard è la praticità, ovvero si è cercato di costruire uno strumento semplice da usare che permettesse di modellare facilmente la struttura delle lezioni. Come abbiamo già detto in precedenza questo nuovo approccio è strutturato in una successione di passaggi, in ognuno dei quali sono stati inseriti comandi per la navigazione avanti e indietro per permettere la completa navigabilità dello strumento. Un ulteriore accorgimento per rendere fruibile wizard è stato quello di inserire, oltre alle descrizioni delle operazioni, come abbiamo appena detto nel paragrafo precedente, una nuova funzionalità di navigazione, essa permette di spostarsi all'interno dei contenuti della lezione permettendo all'utente di accedere a piacimento al contenuto desiderato.

2.2.1 Dettami di progettazione

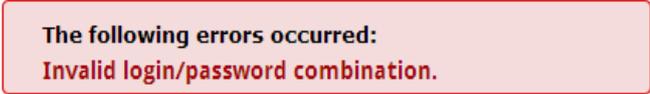
Nel paragrafo precedente abbiamo visto le idee su cui si è sviluppata l'idea di *wizard*. Su di esse si è implementato questo nuovo strumento ma in questo paragrafo vogliamo esaminare le regole che si sono seguite per creare uno strumento conforme ad AContent. Wizard è una estensione che non ha precedenti fra gli authoring tool, che si

sono esaminati nel primo capitolo, per questo la scelta di come strutturarli è stata del tutto sviluppata da zero. Per creare uno strumento efficiente abbiamo deciso di seguire due concetti.

Conformità grafica

Gli autori di AContent prestano molta attenzione all'aspetto grafico della piattaforma e richiedono che tutte le nuove estensioni seguano dettagliatamente i layout da loro impostati. Questa regola è stata tenuta molto in considerazione per ottenere alla fine un nuovo strumento che si amalgamasse perfettamente con la piattaforma, fino al punto di ottenere la sensazione che *wizard* fosse da sempre parte di AContent. Per ottenere questo risultato abbiamo cercato di utilizzare laddove era possibile lo stesso codice per rendere la grafica identica e per sviluppare ancora di più l'idea di *guida*. Naturalmente con la parola codice stiamo considerando gli elementi puramente grafici, poiché ovviamente la parte di gestione di essi è stata riadattata per la nuova funzionalità. Quando ci siamo trovati di fronte alla necessità di implementare schermate grafiche del tutto nuove per AContent si è deciso di crearle rendendole il più simile possibile alla grafica preesistente. Di seguito possiamo vedere un esempio, nella prima figura notiamo una notifica di errore che compare all'utente quando si inseriscono dati sbagliati nella sezione di login.

Login



The following errors occurred:
Invalid login/password combination.

Figura 2.4: Schermata di errore Login

Mentre ora possiamo vedere una schermata di errore implementata ex nova per *wizard*.



The following errors occurred:
Parameter set wrong, start again and remember to fill in all fields.

Figura 2.5: Schermata di errore Wizard

Come possiamo vedere dalle due figure precedenti la grafica dei nuovi costrutti creati per wizard è del tutto simile a quella precedentemente implementata. Questa tecnica di impostazione grafica è stata seguita per tutta la progettazione di wizard e ha permesso di ottenere una estensione perfettamente amalgamata con la piattaforma.

Fruibilità

Con questa regola abbiamo considerato un aspetto molto importante, ovvero quello di determinare la lunghezza giusta (numero di passaggi) di *wizard*. Può sembrare una banalità, ma scegliere correttamente questa misura è stato fondamentale per creare uno strumento che fosse allo stesso tempo educativo, facilmente fruibile e che non richiedesse un tempo troppo lungo per la sua esecuzione. Dopo aver esaminato le funzionalità che *wizard* deve trattare, abbiamo deciso di suddividerlo in quattro passaggi, che esamineremo dettagliatamente nel capitolo successivo, grazie a questi step lo strumento finale soddisfa tutti i punti che abbiamo elencato.

2.2.2 Obiettivi

Il concetto di obiettivi introdotto nei paragrafi precedenti è stato presentato per la prima volta alla conferenza E-Learn di Ottobre 2012 a Montreal, nella quale si è deciso di creare una costruzione assistita delle strutture a partire dagli obiettivi, per fornire un sistema dinamico di strutturazione della lezione sulla base di trigger (in partenza obiettivi). Le tre strutture predefinite di AContent sono Knowledge, Creative e Meta-competency, per ognuna di esse sono stati assegnati vari obiettivi, li vediamo nella tabella successiva. L'introduzione degli obiettivi all'interno di *wizard* consente di usufruire di due vantaggi il primo riguarda quello strutturale del software, mentre il secondo influisce sulla semplicità dello strumento fornito all'utente finale.

STRUTTURA PREDEFINITA	OBIETTIVI ASSEGNATI
Knowledge	remembering, classifying, listing, executing task/exercise, retrieving.
Meta-competency	outlining, recognising, interpreting, carryng out, applying/implementing, verifying/testing, explaining, summarizing.
Creative	comparizing, describing, finding solutions, hypotesising, recognize key problems, designing, planning, venting/devising, producing/costructing.

Tabella 2: Obiettivi delle strutture predefinite

Vantaggio strutturale

Dal punto puramente software e implementativo l'utilizzo di obiettivi migliora e semplifica la gestione delle strutture. Infatti all'interno del codice sono presenti, in una determinata directory (che possiamo vedere nella figura sottostante), tre file del tipo XML che contengono per le tre strutture predefinite sopra citate i loro obiettivi.

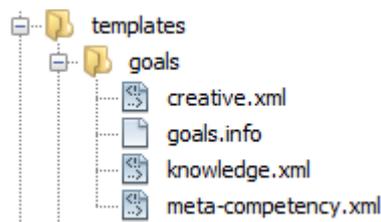


Figura 2.6: Architettura directory degli obiettivi

Vediamo nella Figura 2.7 il file XML di *creative*. Questa specifica architettura permette una banale gestione dei dati in quanto il codice implementato, carica automaticamente tutte le strutture, con annessi obiettivi, ricercandoli semplicemente all'interno della loro directory. Questo permette in caso di sviluppi futuri una semplice gestione degli obiettivi, basterà infatti inserire o eliminare, a seconda delle esigenze i file XML delle strutture desiderate senza dover operare modifiche all'interno del codice.

```
<?xml version="1.0" encoding="UTF-8"?>
<type name="creative">
  <goal>comparizing</goal>
  <goal>describing</goal>
  <goal>finding solutions</goal>
  <goal>hypotesising</goal>
  <goal>recognize key problems</goal>
  <goal>designing</goal>
  <goal>planning</goal>
  <goal>producing/constructing</goal>
  <goal>venting/devising</goal>
</type>
```

Figura 2.7: File XML della struttura Creative

Un vantaggio molto grande soprattutto considerando che AContent è sviluppato da team di programmatori e tale soluzione fa sì che un programmatore non debba lavorare su codice che lui non ha implementato e in ultimo basta una semplice conoscenza dell'XML per poter inserire nuove strutture predefinite con i loro obiettivi.

Semplicità degli obiettivi

Il secondo vantaggio si riscontra nel rapporto fra l'utente finale e la piattaforma. Principalmente i vantaggi interessano gli utenti che hanno una scarsa conoscenza di AContent e che stanno creando le loro prime lezioni. Un utente potrà semplicemente modellare la creazione della sua lezione associando ad essa uno o più degli obiettivi proposti. Nel proseguo della fase di creazione potrà poi scegliere in piena autonomia se associare o meno la struttura predefinita associata all'obiettivo.

Capitolo 3

Implementazione

In questo capitolo tratteremo nel dettaglio l'aspetto implementativo di *wizard*. Come abbiamo accennato nel capitolo precedente *wizard* è strutturato in una serie di passaggi interconnessi che permettono all'utente di navigare autonomamente tra di essi a seconda delle sue preferenze. Nel proseguo esamineremo singolarmente tutti i vari step osservando l'aspetto grafico e quello implementativo.

3.1 Selezione approccio wizard

Il primo passaggio di wizard viene visualizzato nella schermata che abbiamo visto nella Figura 2.1 . La nuova finestra che comparirà all'utente sarà la seguente.



Figura 3.1: Primo passaggio di wizard

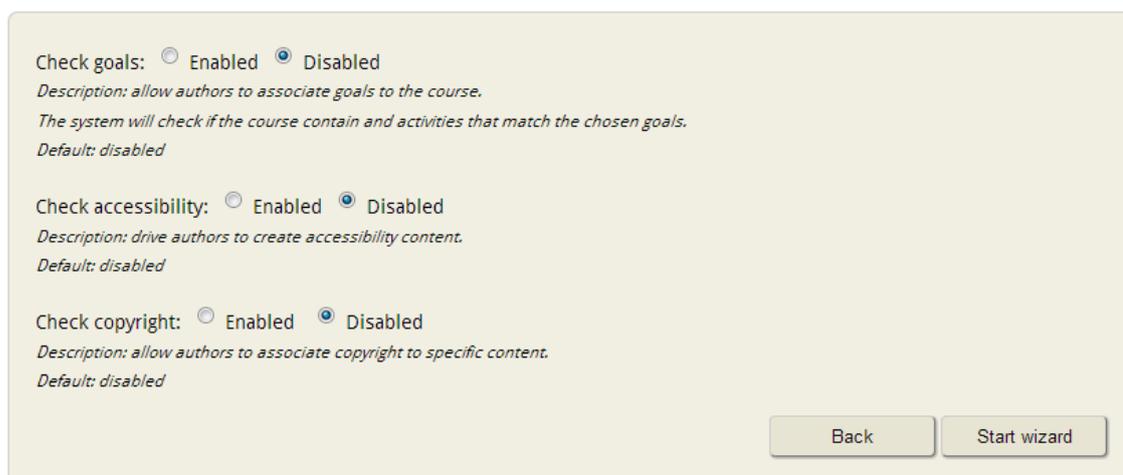
È immediato vedere come l'aspetto grafico sia esattamente identico a quello utilizzato negli approcci *manually* e *structure*, per mantenere una continuità grafica rispetto alle funzionalità preesistenti. Per l'implementazione di questo primo passaggio si è creata una nuova tab denominata "Wizard", che si può vedere nella Figura 3.1. Essa è costituita da una stringa contenente un link, memorizzata su database. Cliccando sul link l'utente accede al secondo passaggio che permette di creare una lezione tramite procedura guidata.

```
<div style="weight: 10%; margin: 10px;">
<?php
    echo _AT('create_content_4', TR_BASE_HREF.'home/editor/edit_content_wizard_step1.php?_course_id='.$_course_id, "");
?>
</div>
```

Figura 3.2: Codice primo passaggio

3.2 Settaggio attributi della lezione

Cliccando poi sul link "use wizard" si accede al secondo passaggio, la schermata che lo rappresenta è visibile nella figura sottostante.



Check goals: Enabled Disabled
Description: allow authors to associate goals to the course.
The system will check if the course contain and activities that match the chosen goals.
Default: disabled

Check accessibility: Enabled Disabled
Description: drive authors to create accessibility content.
Default: disabled

Check copyright: Enabled Disabled
Description: allow authors to associate copyright to specific content.
Default: disabled

Back Start wizard

Figura 3.3: Secondo passaggio di wizard

In questa seconda schermata possiamo vedere la prima scelta che l'utente dovrà fare per

modellare il contenuto della sua lezione. Esaminiamo i tre punti.

- I. *Goals*: abilitando questa opzione sarà possibile associare degli obiettivi al contenuto che si sta creando. Gli obiettivi forniti in wizard, come già detto in precedenza, appartengono alle tre strutture predefinite (creative, knowledge e meta-competency) fornite da AContent e permettono all'utente di poter assegnare ad ogni materiale didattico il tipo di competenza che si richiede acquisire. Una cosa molto importante è che l'eventuale scelta di assegnare degli obiettivi non sarà strettamente vincolante nel proseguo del processo di creazione del contenuto didattico.
- II. *Accessibility*: come abbiamo visto durante la discussione di AContent nei paragrafi precedenti, esso è uno strumento che permette la creazione di materiale didattico accessibile. Questa opzione permette all'utente di poter creare una lezione i cui contenuti sono completamente accessibili. Con questa ultima affermazione, si intendono ad esempio, contenuti video dotati di sottotitoli per gli utenti non udenti. Questa particolare funzione comunque non fa parte di questo progetto ed è stata inserita solamente come base per estensioni future.
- III. *Copyright*: l'ultima opzione riguarda il copyright e permette all'utente di inserire un diritto d'autore ad uno specifico contenuto, come ad un'immagine, ad un video o a qualsiasi altro contenuto inserito nella lezione.

Come possiamo vedere tutte e tre le opzioni sono fornite con annessa descrizione e specificando qual è il settaggio predefinito. Notiamo infine come la pagina sia dotata di due pulsanti che permettono di tornare indietro o di proseguire con la creazione. Dal punto di vista del codice la pagina è gestita con l'iterazione di codice HTML, inerente alle stringhe e ai tre radio button, e il PHP che gestisce lo scambio di informazioni tra i vari passaggi. La grafica è gestita attraverso file CSS ed è stata progettata per essere il più simile possibile ai layout utilizzati da AContent, per rendere *wizard* il più possibile conforme con la piattaforma.

3.3 Selezione contenuto didattico

Dopo aver settato a piacimento le tre opzioni, l'utente prosegue nel suo percorso cliccando sul pulsante "Next" ed avanza al terzo passo di wizard.

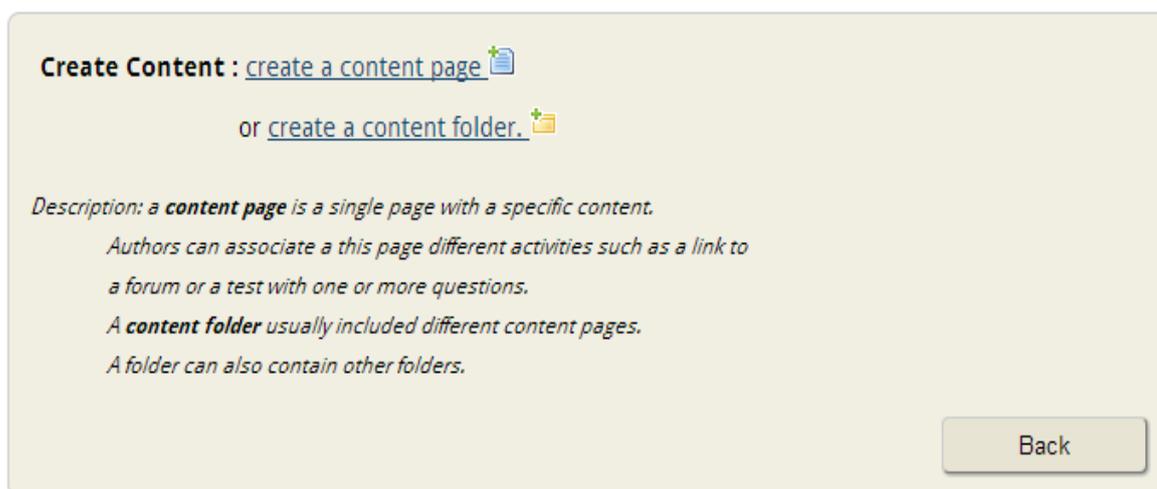


Figura 3.4: Terzo passaggio di wizard

Esso è suddiviso in due schermate differenti che si alternano grazie all'utilizzo di codice Javascript che gestisce in runtime l'iterazione fra l'utente e il software. La prima pagina, visibile nella figura sovrastante, è costituita da due link, gestiti con lo stesso codice visto nella Figura 3.1, e dalla descrizione di cosa comporta la scelta di uno rispetto all'altro. Per la prima volta compare anche il tasto "Back" che permette di tornare al passo precedente e di modificare i settaggi visti nel capitolo precedente.

In questa pagina si chiede all'utente quale delle due tipologie di contenuto didattico voglia creare: una pagina (primo link) o una cartella (secondo link). Possiamo notare come siano state inserite accanto ai due link le icone che abbiamo già esaminato nel capitolo precedente e visto nella Figura 2.3. L'inserimento di quest'ultime rientra nell'idea di rendere wizard una guida, infatti l'utente ora potrà rendersi conto che quelle due icone servono per la creazione di una pagina e di una cartella ed una volta fuori da wizard potrà utilizzarle autonomamente.

Selezionando uno dei due link, attraverso una serie di funzioni Javascript, di cui ne vediamo un esempio nella pagina successiva, viene mostrata la seconda schermata che compone il terzo passaggio di wizard.

```

$('#label_create_folder').live("click",function(){
    $('#ftitle_folder').css('display','inline');
    $('#step2_intro').css('display','none');
    $('#step2_body').css('display','inline');
    $('#buttons_body').css('display','inline');
    $('#buttons_intro').css('display','none');
    $('#hidd_page').remove();
});

```

Figura 3.5: Codice funzione Javascript

Di seguito vediamo la schermata che viene generata nel caso in cui l'utente abbia scelto di creare come contenuto una pagina e abbia settato enable la voce *goals* al primo passaggio.

Customize settings for the content page:

Enter the title:

Description : enter the title of this content page.

Choose the goals:

Knowledge

<input type="checkbox"/> remembering	<input type="checkbox"/> classifying	<input type="checkbox"/> listing
<input type="checkbox"/> executing task/exercise	<input type="checkbox"/> retrieving	

Meta-competency

<input type="checkbox"/> outlining	<input type="checkbox"/> recognising	<input type="checkbox"/> interpreting
<input type="checkbox"/> carryng out	<input type="checkbox"/> applying/implementing	<input type="checkbox"/> verifying/testing
<input type="checkbox"/> explaining	<input type="checkbox"/> exemplifying	<input type="checkbox"/> summarizing

Creative

<input type="checkbox"/> comparizing	<input type="checkbox"/> describing	<input type="checkbox"/> finding solutions
<input type="checkbox"/> hypotesising	<input type="checkbox"/> recognize key problems	<input type="checkbox"/> designing
<input type="checkbox"/> planning	<input type="checkbox"/> producing/constructing	<input type="checkbox"/> venting/devising

Figura 3.6: Quarto passaggio di wizard, con goals

Possiamo vedere nella figura sovrastante, come gli obiettivi siano suddivisi in base alla struttura predefinita di AContent, come detto precedentemente. Naturalmente nel caso l'utente setti a disabled l'opzione goals la schermata che si presenterà sarà quella che

vediamo nella Figura 3.7 . A partire da questa schermata l'utente è chiamato ad inserire il titolo del contenuto e nel caso abbia abilitato i goals, a selezionare uno degli obbiettivi.

Figura 3.7: Quarto passaggio di wizard, senza goals

La distinzione degli obbiettivi all'interno delle strutture predefinite a cui appartengono, come in Figura 3.6 viene gestito in maniera automatica attraverso la simbiosi di codice PHP e HTML, di seguito vediamo uno stralcio inerente alla struttura Creative.

Figura 3.8: Codice per caricamento automatico obiettivi di Creative

```

<tr>
<td>
<?php $xml= simplexml_load_file(TR_BASE_HREF.'templates/goals/creative.xml'); ?>
<div id="goals_creative">
<fieldset><legend><?php echo _AT('creative'); ?></legend>
  <table>
    <tr>
<?php
$support=1;
foreach($xml->children() as $child) {
  if($support>3){ ?>
  </tr>
  <?php $support=1; ?>
  <tr>
  <?php } ?>
  <td>
    <input id="<?php echo $child;?>" type="checkbox" name="goal_creative_<?php echo str_replace(" ", "", $child);?>" value="<?php echo $child;?>"/>
  </td><td><label style="margin-right:10px;"><?php echo $child; ?></label>
  </td>
  <?php
  $support++;
}
}
?>
</tr>
</table>

```

Le due iterazioni d'inserimento richieste all'utente sono gestite da *wizard* con un controllo che viene eseguito al momento del click sul pulsante "Next". I parametri

richiesti sono passati attraverso un form che utilizza il metodo POST. Nel caso in cui le variabili contenenti il titolo o gli obiettivi selezionati (qualora essi siano abilitati) non esistono, wizard interrompe la successione dei suoi vari passaggi facendo comparire la schermata di errore contenente anche la spiegazione di quale errore è stato commesso, dando così modo all'utente di capire dove ha sbagliato.



Figura 3.9: Schermata di errore

Questa schermata (Figura 3.9) offre la possibilità di ritornare indietro e riprovare la creazione del contenuto desiderato, attraverso l'uso del tasto "Back", oppure di uscire da wizard, interrompendo la creazione della lezione ritornando nella home di AContent cliccando sul pulsante "Cancel".

3.4 Funzioni dei contenuti didattici

Il quarto passaggio di wizard è l'ultimo step di questo nuovo approccio. Questa ultima parte si suddivide in due ramificazioni distinte che si differenziano a seconda del contenuto che l'utente sta creando e contengono tutte le funzioni che possono essere eseguite su un determinato contenuto. Nonostante la diversificazione vi sono alcune funzioni simili, che spieghiamo ora per non ripeterci durante la descrizione delle singole schermate. Le componenti che appartengono a tutte le ramificazioni sono sei, le vediamo in successione.

- I. *funzione home*: è costituita da un semplice link che permette all'utente di ridirigersi alla home page di AContent, ponendo fine all'utilizzo di wizard. Lo possiamo vedere nella Figura 3.11 nella pagina seguente.



Figura 3.11: Link home page

Da notare l'utilizzo dell'icona e della descrizione della funzione che permettono all'utente di apprendere e riconoscere in futuro tale funzionalità;

- II. *funzione delete*: tramite questa operazione l'utente potrà eliminare il contenuto su cui sta lavorando. Vi si accede tramite il link:



Figura 3.12: Link delete content

Naturalmente nella Figura 3.12 è visualizzato il caso in cui si stia lavorando su un *folder*, in caso contrario la dicitura comparirebbe, mostrando *page*. Questo collegamento ci porta alla schermata:

Edit content wizard delete



Figura 3.13: Schermata delete

Da quest'ultima è possibile eliminare il contenuto tramite il pulsante "*Delete*" oppure ritornare indietro senza effettuare la cancellazione tramite il bottone "*Back*". Durante l'implementazione di questa funzione si è presentato il problema di come comportarsi nel caso in cui la lezione a cui apparteneva il contenuto che si stava eliminando conteneva o meno più contenuti. La soluzione che si è adottata si basa su un controllo su database che determina il numero dei materiali che compongono la lezione e in base al loro numero si effettua un diverso re indirizzamento. Di seguito possiamo vedere la prima parte del controllo che si occupa di determinare l'effettivo numero dei contenuti della lezione in esame.

```
$sql="SELECT COUNT(*) AS tot FROM ".TABLE_PREFIX."content WHERE course_id=".$course_id." GROUP BY course_id";
$result=$contentDAO->execute($sql);
if(is_array($result))
{
    foreach ($result as $support) {
        $tot=$support['tot'];
        break;
    }
}
```

Figura 3.14: Codice iniziale controllo numero contenuti.

Il primo caso è quello in cui il contenuto che si sta cancellando è l'unico della lezione, allora l'utente sarà riportato alla schermata vista in figura 3.1. Altrimenti se la lezione è composta da più di un contenuto si rimanderà l'utente alla schermata delle funzioni di un altro materiale;

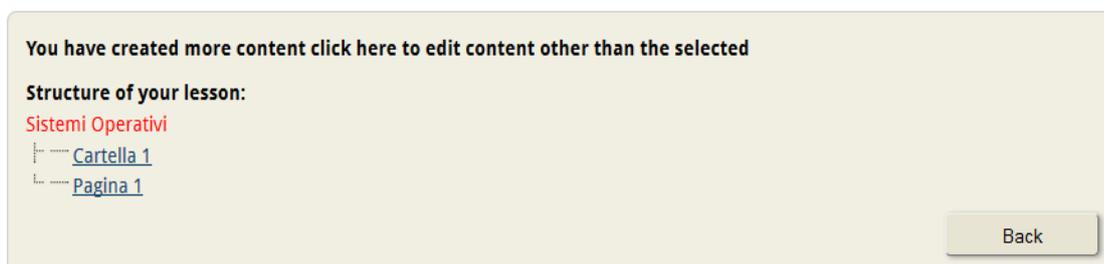
- III. *funzione di navigazione fra i contenuti*: questa funzionalità consente all'utente di navigare velocemente tra i contenuti della lezione che sta modellando. Il link che ci consente di accedere alla schermata che offre questa possibilità è il seguente,

You have created more content click here to edit content other than the selected

Figura 3.15: Link di collegamento alla pagina di navigazione

Cliccando sulla stringa in Figura 3.15 si apre la pagina seguente, dalla quale l'utente può vedere la struttura della lezione che ha creato fino a quel determinato momento.

Edit content wizard



You have created more content click here to edit content other than the selected

Structure of your lesson:

Sistemi Operativi

- Cartella 1
- Pagina 1

Back

Figura 3.16: Schermata di Navigazione fra i contenuti

Possiamo notare come in rosso sia rappresentato il titolo della lezione che si sta modellando, mentre in blu vi sono i vari contenuti. Cliccando su di essi si accede alla schermata delle funzioni inerenti al materiale selezionato. Dal punto di vista

implementativo abbiamo deciso di adottare questa grafica per mantenere una continuità nei layout della piattaforma e perché dal punto di vista della fruibilità è una soluzione semplice e intuitiva. Per la sua creazione e modellazione abbiamo implementato nuove funzioni, denominate “*printMainMenuWizard*” e “*printMenuWizard*”, entrambe si basano su funzioni preesistenti, che sono state riadattate all’esigenze di wizard. Senza entrare nel dettaglio, entrambe sfruttano l’accesso al database per recuperare tutti i contenuti che appartengono alla lezione su cui si sta lavorando poi tramite tag html creano la grafica che vediamo nella figura nella pagina precedente. Di seguito possiamo vedere una parte del codice di una delle due funzioni (*printMenuWizard*).

```

/* @See include/html/menu_menu.inc.php */
/* Access: PRIVATE */
function printMenuWizard($parent_id, $depth, $path, $children, $truncate, $ignore_state, $course_id, $from = '') {
    global $cid, $my_uri, $base_path, $rtl, $substr, $strlen, $current_user;
    static $temp_path;

    if (!isset($temp_path)) {
        if ($cid) {
            $temp_path = $this->getContentPath($cid);
        } else {
            $temp_path = $this->getContentPath($_SESSION['s_cid']);
        }
    }

    $highlighted = array();
    if (is_array($temp_path)) {
        foreach ($temp_path as $temp_path_item) {
            $_SESSION['menu'][$temp_path_item['content_id']] = 1;
            $highlighted[$temp_path_item['content_id']] = true;
        }
    }

    if ($this->start) {
        reset($temp_path);
        $this->start = false;
    }

    if (isset($this->_menu[$parent_id]) && is_array($this->_menu[$parent_id])) {
        $stop_level = $this->_menu[$parent_id];
        $counter = 1;
        $num_items = count($stop_level);
    }

```

Figura 3.17: Prima parte funzione “*printMenuWizard*”

IV. *funzioni Next e Back*: queste due permettono all’utente di navigare all’interno di *wizard* accedendo a tutte le sue funzionalità. Come detto nel capitolo precedente questa estensione si prepone l’obiettivo di fungere da *guida* agli utenti e per questo si è deciso di renderla del tutto “navigabile”. Bisogna tuttavia fare una precisazione, la funzione *Back*, che viene svolta dal pulsante che vediamo nella figura sottostante, ha il solo compito di riportare l’utente alla schermata



Figura 3.18: Pulsanti Back e Next

precedente, mentre il tasto “Next”, svolge un duplice compito, a seconda di dove essi si trovano. Infatti quelli che troviamo nei primi tre passaggi svolgono il semplice compito di far proseguire l’utente fra i vari step. Mentre, quando si giunge al quarto ed ultimo passaggio, essi permettono di salvare le modifiche e di creare nuovi contenuti. Per quanto concerne ad esempio per il tasto “Next” che troviamo nella schermata delle funzioni dei contenuti, in Figura 3.22 e 3.25, ci consente di collegarci al terzo passaggio, in Figura 3.4, dal quale possiamo creare un nuovo contenuto;

- V. *funzione Info strutture*: questa parte si occupa di fornire agli utenti una descrizione delle strutture predefinite associate agli obiettivi che si sono associati al contenuto. Per l’implementazione abbiamo utilizzato un classico link, ma con l’aggiunta di una funzione javascript che si attiva al momento del click e che non conduce ad una nuova schermata, ma apre una nuova pagina del browser contenente la seguente schermata.

Info AContent default structure

Here you can view the predefined AContent that offers to create lessons.

CREATIVE BASED
Description: centered on the motivations and "emotions" of the learner.
[\[show outline \]](#)

KNOWLEDGE BASED
Description: is to follow a path of self-learning to acquire the "base" of the subject matter.
[\[show outline \]](#)

META-COMPETENCY BASED
Description: emphasizes the logic of a constructivist approach to the construction of knowledge.
[\[show outline \]](#)

Figura 3.18: Schermata Info strutture

Si è deciso di aprire una nuova pagina del browser per permettere all'utente di mantenere aperta la finestra mentre si lavora sulla creazione della propria lezione, per consentirgli così di poter avere sempre sotto mano le varie strutture predefinite, così da poterne trarre suggerimenti o in altri casi facilitandone la copiatura. Dal punto di vista grafica, come anche per le schermate che abbiamo visto sopra, si è implementato lo stesso layout utilizzato nella sezione di creazione di strutture predefinite secondo l'approccio *structure*, visibile in Figura 2.2 , sempre per garantire una continuità grafica con le vecchie configurazioni.

- VI. *funzione aggiunta struttura predefinita*: questa è l'ultimo costruito in comune tra le due ramificazioni. Questa funzionalità risulta visibile solo nel momento in cui un utente associ alla propria lezione uno o più obiettivi. In tal caso vedrà nelle schermate delle funzioni dei suoi contenuti il seguente link:

[You have selected the objectives, go to the screen to add a predefined structure](#)

Description : it's possible to insert a predefined structure in the lesson according to the objectives selected.

Figura 3.19: Link aggiunta strutture predefinite

Questo link permette di accedere alla schermata successiva,

Edit content wizard control structure

You have selected the following objectives
remembering carryngout describing

Below you can see the default structures associated with the goals that you have chosen:

CREATIVE BASED

Description: centered on the motivations and "emotions" of the learner.
[\[show outline \]](#)

KNOWLEDGE BASED

Description: is to follow a path of self-learning to acquire the "base" of the subject matter.
[\[show outline \]](#)

META-COMPETENCY BASED

Description: emphasizes the logic of a constructivist approach to the construction of knowledge.
[\[show outline \]](#)

Now if you want you can associate a predefined structure to the lesson select it and click save

Figura 3.20: Schermata aggiunta strutture predefinite

In questa pagina è possibile vedere gli obiettivi scelti dall'utente durante la fase di creazione della lezione. Nella parte centrale notiamo le tre strutture predefinite, da specificare, che nel caso riportato in figura i tre obiettivi sono rispettivamente appartenenti uno per uno ad una singola struttura. Nel caso in cui si scelgano più obiettivi della stessa struttura predefinita comparirà solo il radio button di quella determinata struttura. In fine nel taglio basso possiamo vedere tre diversi bottoni che permettono di:

1. *Save*: viene salvata la struttura predefinita all'interno della lezione su cui si sta lavorando e si rimane nella stessa schermata visibile in figura 3.20, permettendo così agli utenti di inserire altre strutture qual ora lo si voglia;
2. *Back*: consente di tornare alla schermata precedente se apportare modifiche alla lezione, utile nel momento in cui l'utente possa decidere di non voler più assegnare una delle strutture predefinite;
3. *Next*: ha il compito di salvare la struttura predefinita selezionata dall'utente, con un click sul radio button, ma a differenza di *Save* reindirizza alla schermata delle funzioni.

Dal punto di vista implementativo ho creato nuove funzioni per il salvataggio su database della struttura predefinita selezionata, basandomi sempre su vecchie funzioni, per rendere semplice e veloce la comprensione del codice a nuovi programmatori. Vediamo nella pagina seguente, in Figura 3.21 una parte di codice della funzione che esegue la creazione della struttura predefinita.

Abbiamo inserito anche un controllo sui pulsanti *Save* e *Next* che permette di verificare che sia stato effettivamente selezionata una struttura dall'utente. In caso negativo comparirà una schermata di errore, come quella vista nella Figura 3.9, dove il pulsante "*Back*" riporta alla schermata della scelta della struttura predefinita, mentre "*Cancel*" conduce alla homepage.

```

if ($_POST['title'] == '') {
    $msg->addError(array('EMPTY_FIELDS', _AT('title')));
}

if (!$msg->containsErrors())
{
    $_POST['title'] = $content_row['title'] = $_POST['title'];
    // find out ordering and content_parent_id
    if ($pid)
    { // insert sub content folder
        $ordering = count($contentManager->getContent($pid))+1;
    }
    else
    { // insert a top content folder
        $ordering = count($contentManager->getContent(0)) + 1;
        $pid = 0;
    }
    $scid = $contentManager->addContent($_SESSION['course_id'],
                                        $pid,
                                        $ordering,
                                        $_POST['title'],
                                        '','','0','0','1',
                                        CONTENT_TYPE_FOLDER);

    $struc_manag = new StructureManager($_POST['title']);
    $page_temp = $struc_manag->get_page_temp();

    $struc_manag->createStruct($page_temp, $scid, $_course_id);
}

```

Figura 3.21: Parte della funzione per la creazione della struttura predefinita

Ora che abbiamo esaminato nel dettaglio tutte le funzioni in comune tra le due ramificazioni possiamo esaminare come si differenziano le due schermate inerenti al contenuto *page* e *folder*.

3.4.1 Contenuto Folder

In questo sotto capitolo vedremo come è composta la schermata delle funzioni legate al contenuto *Folder*. In AContent questo tipo di contenuto didattico può svolgere due tipi di compiti: il primo è quello comune di una qualsiasi cartella, contenitore di LO di tipo *page* o di altri di tipo *folder*, mentre il secondo è quello di poter essere correlato ad uno o più obiettivi per poter contenere strutture predefinite associate a questi obiettivi. Nella pagina seguente in Figura 3.22 possiamo vedere come si presenta all'utente la schermata che raccoglie le funzioni dedicate all'elemento *folder*.

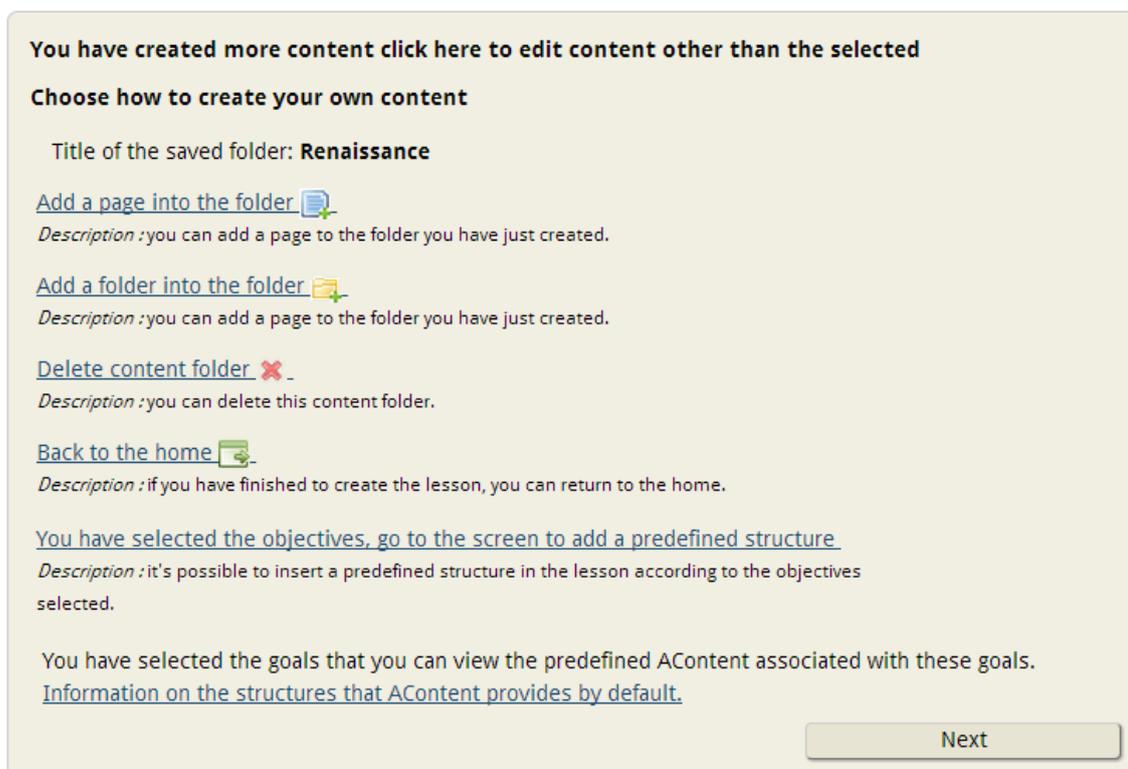


Figura 3.22: Schermata contenuto folder

Vediamo immediatamente la presenza di tutti i link in “comune” visti nei paragrafi precedenti, ma ora vedremo le due funzioni aggiuntive che si possono eseguire su un contenuto del tipo *folder*.

- I. *Add a folder into the folder*: questa funzione permette l’inserimento di una cartella all’interno della cartella su cui stiamo lavorando. Per la realizzazione della schermata che esegue questa operazioni si è utilizzata la stessa grafica, per mantenere una continuità nell’aspetto. Dal punto di vista dell’implementazione si sono utilizzate funzioni simili a quelle già presenti nel vecchio software, per non allontanarsi troppo dalle precedenti implementazioni che gestivano questa funzionalità. Di seguito nella Figura 3.23 possiamo vedere uno spezzone di codice, che implementa la schermata grafica.
- II. *Add a page into the folder*: questa funzione permette l’inserimento di una pagina all’interno della cartella, la schermata che viene visualizzata cliccando su questo link è quella visibile nella Figura 3.24.

```

<link type="text/css" href="<?php echo TR_BASE_HREF;?>themes/default/forms.css" rel="stylesheet">
<script type="text/javascript" src="<?php echo TR_BASE_HREF;?>home/editor/js/wizard.js"></script>
<form action="<?php echo TR_BASE_HREF;?>home/editor/edit_content_wizard_subfolder.php?course_id=<?php echo $course_id; ?>&pid=<?php echo $content_id; ?>">
<div class="input-form" style="width:95%;margin-left:1.5em;">
<input type="hidden" value="<?php echo $content_id; ?>" name="cid">
<div style="width: 10%; margin: 10px 0px 0px 10px;">Enter the name of the subfolder.</div>

  <div class="row">
    <div style="font-weight:bold;"><span class="required" title="<?php echo AT('required_field'); ?>">*</span>
    <label for="ftitle"><?php echo AT('content_folder_title'); ?></label></div>
    <input type="text" name="title" id="ftitle" size="70" class="formfield" value="" />
  </div>

  <div class="row buttons">
    <label id="lab_back" name="<?php echo $course_id; ?>"><label class="button_back2">Back</label></label>
    <input type="submit" name="submit" value="<?php echo AT('save'); ?>" title="<?php echo AT('save_changes'); ?>" alt-s="accesskey="s" />
  </div>
</div>
</form>

```

Figura 3.23: Codice Html per la schermata d’inserimento di una sotto cartella

Da questa, l’utente, ha la possibilità di utilizzare tutte le funzionalità applicabili previste da AContent su un contenuto di tipo pagina. Nel caso in cui, dopo aver visionato le varie possibili caratteristiche di creazione, l’utente volesse tornare indietro senza effettuare una creazione gli basterebbe cliccare sul pulsante “Back” per ritornare indietro fino alla schermata visibile nella Figura 3.22.

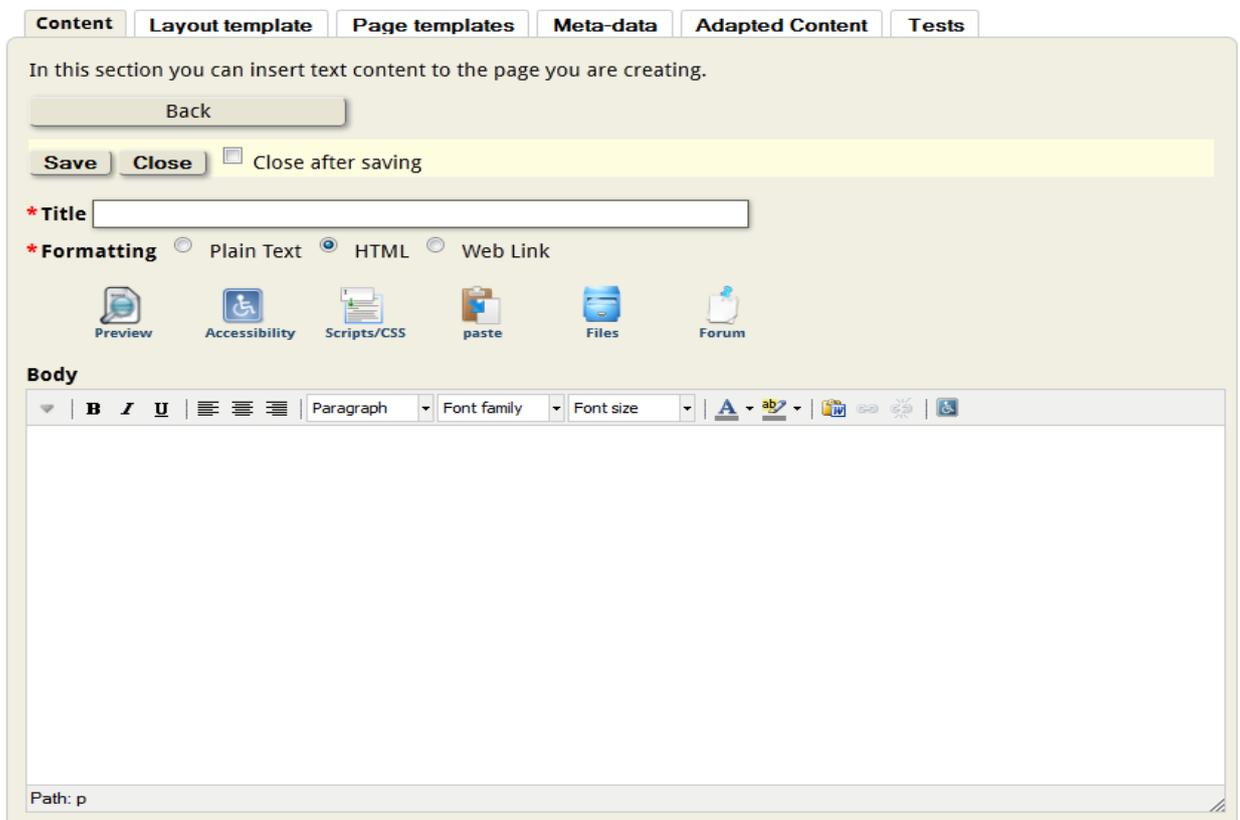


Figura 3.24: Schermata creazione sotto pagina

3.4.1 Contenuto Page

Questa sezione raccoglie tutte le funzioni che l'utente può eseguire nel momento in cui crea un contenuto di tipo page. Nella Figura 3.25, visibile nella pagina seguente, vediamo come si presenta la schermata. In essa ritroviamo tutte le varie operazioni in comune che abbiamo analizzato all'inizio di questo capitolo. Vi sono presenti quattro funzioni specifiche per il contenuto page, su queste va fatta una precisazione. Com'è possibile vedere dalla figura seguente solo la prima delle quattro è corredata dall'icona. Essa, come detto ampiamente nel corso del secondo capitolo, svolge la funzione d'insegnamento permettendo agli utenti di associare in maniera visiva la funzione svolta con l'icona. È presente una sola icona per il semplice motivo che le altre tre opzioni proposte per la modellazione del contenuto page non sono implementate singolarmente

You have created more content click here to edit content other than the selected

Choose how to create your own content

Title of the saved page: **Introduction**

[Edit this page via the text editor](#)
Description : use the text editor to add content to the page.

[Assign a template to content](#)
Description : allow to add different templates to content.

[Assign a layout to content](#)
Description : allow to add a layout to content

[Delete content page](#) ✖
Description : you can delete this content page.

[Edit this page via classic editor](#) ✎
Description : create text content and associate to it a template and/or a layout via the classic editor.

[Back to the home](#) 🏠
Description : if you have finished to create the lesson, you can return to the home.

[You have selected the objectives, go to the screen to add a predefined structure](#)
Description : it's possible to insert a predefined structure in the lesson according to the objectives selected.

You have selected the goals that you can view the predefined AContent associated with these goals.
[Information on the structures that AContent provides by default.](#)

Next

Figura 3.25: Schermata contenuto page

in AContent, ma sono tutte raggruppate in una sola schermata, come è possibile vedere nella Figura 3.24. In *wizard* è stato deciso di creare tre percorsi singoli che indirizzassero l'utente direttamente all'editor, al settaggio dei layout o all'aggiunta di

template. Questa decisione è stata fatta per rendere più veloce la modellazione della page, in quanto queste tre funzionalità sono quelle più comunemente utilizzate.

Ora vediamo nel dettaglio in cosa consistono queste quattro operazioni:

- I. *Insert content through classic editor*: questo primo link permette di accedere alla schermata che consente di svolgere tutte le funzionalità dei contenuti di tipo page. Visivamente è identica alla Figura 3.24, con l'unica eccezione della text box contenente il titolo, che in figura è vuota, mentre in questo caso conterrà il nome della pagina che abbiamo creato. Dal punto di vista puramente implementativo si è cercato di rimanere il quanto più possibile fedeli al codice già scritto in precedenza e che gestiva queste funzioni, sono state tuttavia necessarie l'aggiunta di nuove funzioni per permettere il corretto funzionamento di *wizard*.
- II. *Enter through the content editor*: questa funzione permette di accedere al schermata specifica dell'editor di testo, dal quale è possibile sia inserire normale testo, ma anche immagini, video o altri file multimediali. La prima versione implementata di questa funzionalità è stata realizzata creando ex novo un editor, ma dopo una attenta analisi ci si è resi conto che questa disparità, in maggioranza solo grafica, fra il nuovo editor e il vecchio poteva causare confusione negli utenti, per cui si è deciso di implementarlo come il precedente dandogli l'aspetto che possiamo vedere nella Figura 3.26. Si nota che la grafica è del tutto identica a quella del vecchio editor, con la sola differenza della stringa in e del bottone "*Back*" in alto a destra, quest'ultimo consente come detto di tornare indietro alla schermata delle funzioni del contenuto *page*. Dal punto di vista del codice si è seguito, per quanto possibile, l'implementazione base dell'editor esistente, modificando e aggiungendo nuove funzioni che permettessero il corretto funzionamento dello strumento editor. Le nuove funzioni inserite hanno preso lo stesso nome delle precedenti, con l'aggiunta di *wizard* nel titolo, per consentire a programmatori futuri di localizzare con facilità l'ambiente di utilizzo di quest'ultime.

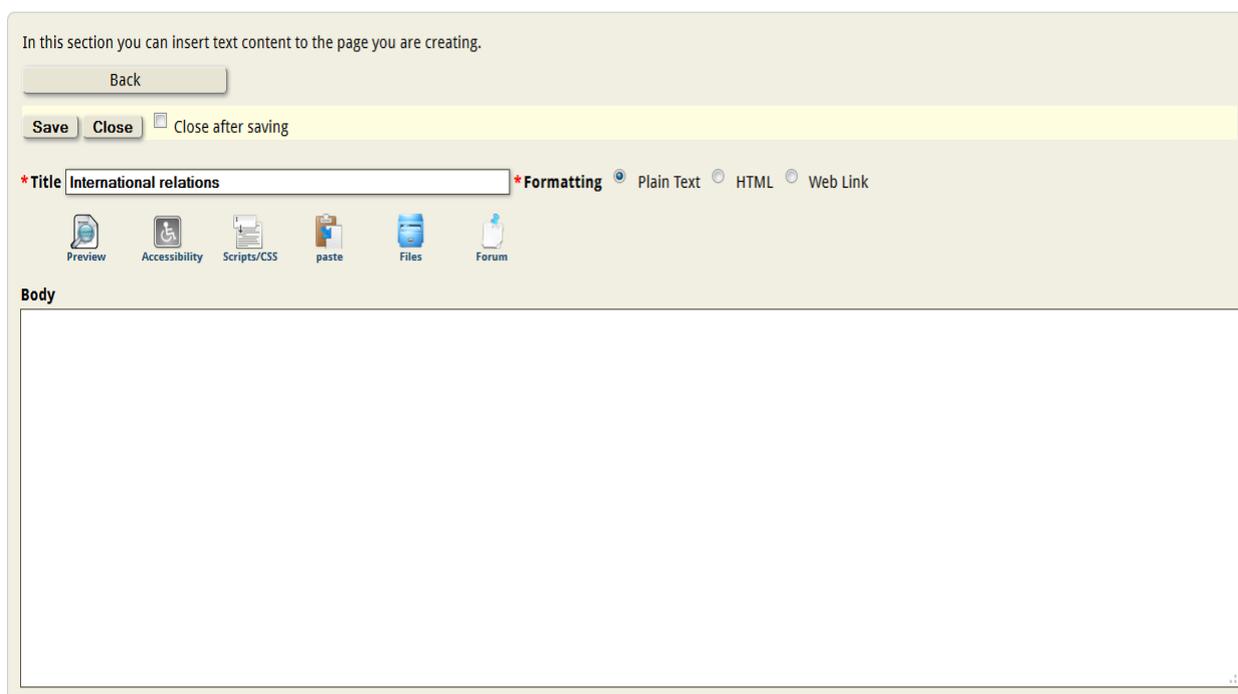
Edit content wizard editor

Figura 3.26: Schermata esclusiva dell'editor

- III. *Assign a template to content*: questa terza funzione consente all'utente l'inserimento di uno o più template alla sua *page*. Con la parola *template* intendiamo una struttura predefinita per il contenuto di una pagina, al momento in AContent esistono dodici template predefiniti che si differenziano fra loro per la disposizione del testo, inserimento di immagini nel testo, stile del titolo e altre piccole varietà nell'organizzazione del testo. La gestione dei template avviene attraverso la Figura 3.28, da essa l'utente può sia aggiungere template, eliminare i template precedentemente inseriti e modificarne la disposizione all'interno della pagina, scegliendo l'ordine che preferisce. La visualizzazione dei template ed il loro salvataggio avviene grazie alla combinazione di questi quattro pulsanti.



Figura 3.27: Pulsanti gestione template

“*Hide Templates*” e “*Add Templates*” gestiscono la scomparsa e la ricomparsa della tabella (a sfondo bianco) che contiene i dodici template di default, visibile nella figura sottostante al centro.

Edit content wizard template

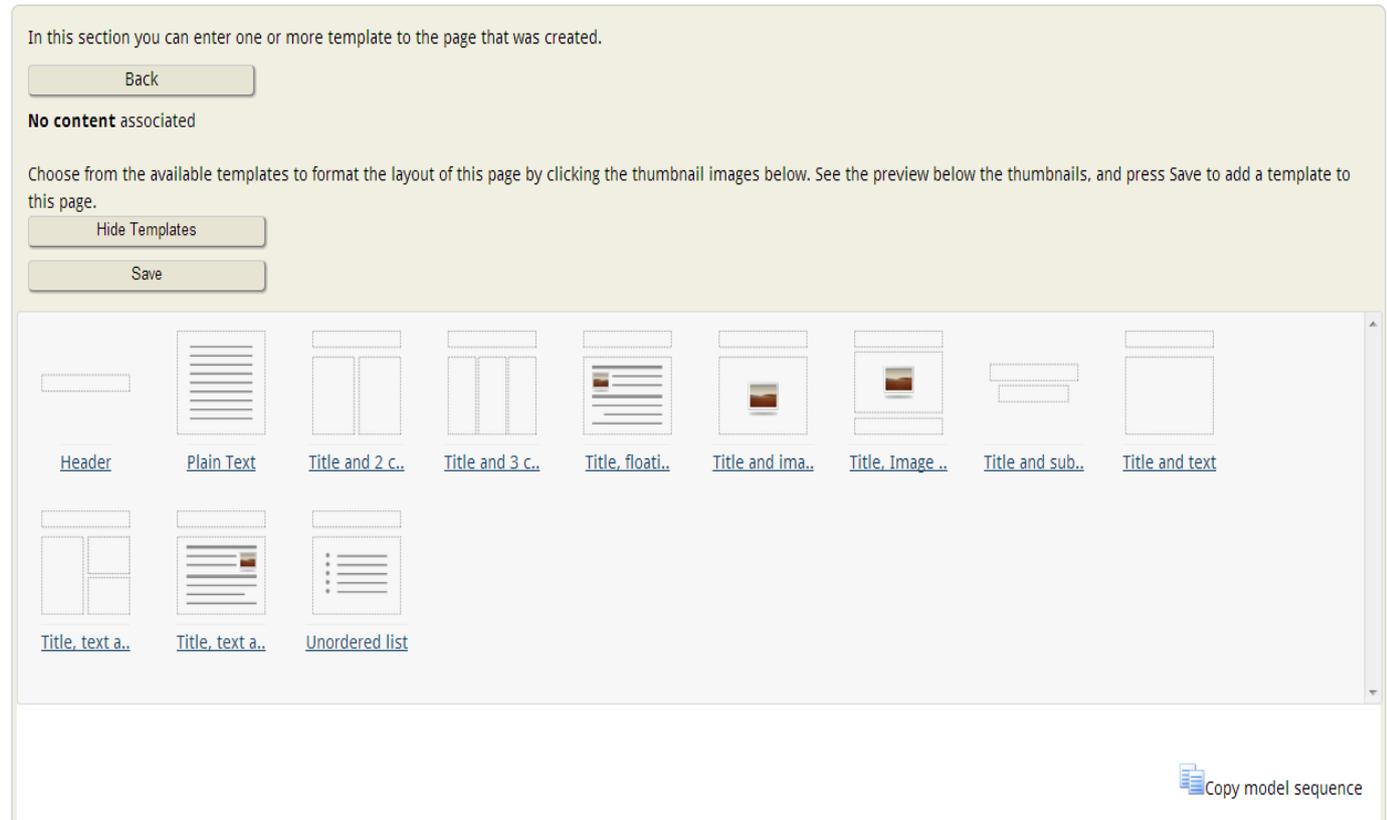


Figura 3.28: Schermata template

“*Edit Templates*” consente di svolgere una doppia funzionalità, quella di eliminare un template aggiunto, tramite l’icona “X” in alto a sinistra nella, o di intervenire sull’ordine dei template selezionati utilizzando i quattro pulsanti in basso a destra, visibili in Figura 3.29.

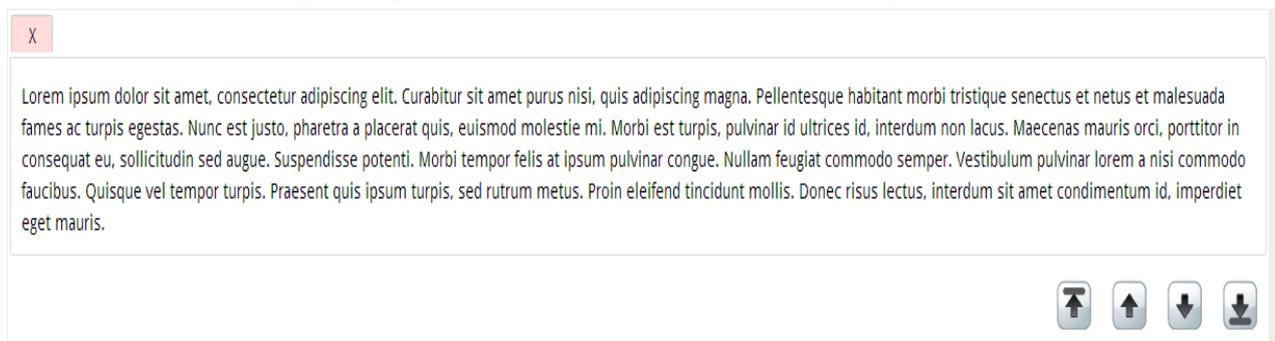


Figura 3.29: Edit template

Mentre “Save” permette il salvataggio dei template scelti e di tutte le modifiche a loro applicate.

- IV. *Assign a layout to content*: questa ultima funzione permette all’utente di assegnare alla propria pagina un layout predefinito messo a disposizione da AContent per personalizzare i propri contenuti. Al momento vi sono cinque layout predefiniti ognuno dei quali si differenzia dal punto di vista grafico. La schermata che li raccoglie e gestisce è la seguente.

Edit content wizard layout

In this moment there isn't an associated layout.

Select your favorite layout:

	Name	Description	Icone
<input type="radio"/>	Nothing	Without layout	
<input type="radio"/>	Canada	Canada	
<input type="radio"/>	Italy	Italy	
<input type="radio"/>	Seti	Scienze e Tecnologie Informatiche	
<input type="radio"/>	Unibo	Tema ufficiale dell'Unibo	
<input type="radio"/>	Microsoft Windows	Microsoft Windows	

Figura 3.30: Schermata layout

In questa pagina la prima cosa da notare è la stringa in alto a sinistra che avverte che al momento non vi sono layout associati alla lezione, in caso contrario comparirebbe il nome del layout associato, tale controllo avviene con una interrogazione al database gestita con codice PHP. Sotto di essa vediamo due pulsanti “Back”, come in tutte le altre schermate svolge il compito di riportare alla schermate delle funzioni di page, senza effettuare modifiche, mentre “Apply layout to this content” memorizza su database il layout scelto, attraverso la selezione di un radio button, da

parte dell'utente. Tale operazione viene svolta attraverso funzioni ad-hoc, che sfruttano l'iterazione di comandi PHP ed SQL per andare a scrivere su database i dati sensibili per la memorizzazione del layout prescelto.

Per consentire all'utente di poter visionare come la propria lezione si modifichi applicando uno dei diversi layout si è inserita una funzione Javascript che permette in runtime di mostrare un'anteprima di come si modifica l'aspetto del contenuto della pagina su cui si sta lavorando nel momento in cui si seleziona un radio button.

Preview italy:

Obiettivi

Il pensiero divergente (apprendimenti superiori divergenti). Analisi (analizzare, confrontare, indurre), sintesi (sintetizzare, schematizzare, dedurre), intuizione (tentare soluzioni, formulare ipotesi), invenzione.

- » tentare soluzioni ...
- » formulare ipotesi ...
- » riconoscere problemi chiave ...
- » generare ...
- » pianificare ...
- » produrre soluzioni nuove ...
- » formulare soluzioni nuove ...

Preview windows:

Obiettivi

Il pensiero divergente (apprendimenti superiori divergenti). Analisi (analizzare, confrontare, indurre), sintesi (sintetizzare, schematizzare, dedurre), intuizione (tentare soluzioni, formulare ipotesi), invenzione.

- ✔ tentare soluzioni ...
- ✔ formulare ipotesi ...
- ✔ riconoscere problemi chiave ...
- ✔ generare ...
- ✔ pianificare ...
- ✔ produrre soluzioni nuove ...
- ✔ formulare soluzioni nuove ...

Figura 3.31: Anteprime layout Italy e Windows

Come possiamo vedere dalla figura sovrastante, che rappresenta due diverse anteprime, si ha un esempio di come il layout “*Italy*” e quello

“*Windows*” modificano in maniera diversa lo stesso template. Attraverso queste anteprime l’utente ha la possibilità di scegliere quale layout applicare alla propria lezione.

Conclusioni

L'estensione illustrata in questa tesi si colloca all'interno del progetto BEAT Bologna E-learning Authoring Tool ed ha visto la collaborazione del nostro dipartimento con quello di Scienze dell'Educazione per l'individuazione dell'idea di base su cui implementare *wizard*, ovvero quella di realizzare uno strumento che permetta di creare un LO in modo intuitivo, accessibile ad utenti con formazioni diverse e che permetta di soddisfare alcuni vincoli. Questa collaborazione ha portato all'individuazione di diversi obiettivi, i quali hanno formato le linee guida per lo sviluppo di *wizard*. Il primo obiettivo preso in considerazione è stato quello di semplificare la procedura di creazione dei LO. Con questa estensione è infatti possibile, seguendo semplici passi, costruire corsi in AContent anche con strutture complesse fornendo assistenza durante la fase di costruzione degli LO. *Wizard* presenta all'utente tutte le funzionalità per la composizione di strutture articolate, permettendo di ampliare anche quelle predefinite aggiungendovi cartelle o nuove pagine, descrivendo ogni azione in modo semplice e chiaro. In ogni suo step, fornisce collegamenti diretti e intuitivi a tutte le funzionalità messe a disposizione dalla piattaforma guidando l'utente senza obbligarlo all'interno di un percorso di creazione predefinito. L'altro obiettivo che ha indirizzato lo sviluppo di *wizard* è stato appunto quello di creare uno strumento funzionale per tutte le tipologie di utenti. Questo ha portato all'implementazione di una estensione che permette ad utenti esperti di modellare in maniera personalizzata le proprie lezioni senza dover eseguire passi aggiuntivi o ripetitivi, dando così la possibilità di ottenere, al termine della procedura di *wizard*, la struttura desiderata e già completa, cosa che non era possibile utilizzando i vecchi approcci. Per i nuovi utenti *wizard* è invece un importante strumento che fornisce un percorso di apprendimento che permette, in una successione di pochi step, di far conoscere e apprendere appunto tutte le funzionalità di creazione dei contenuti di cui dispone la piattaforma. Ogni attività che

L'utente può eseguire è infatti descritta in modo chiaro e semplice ed associata alle icone e alle label che si ritrovano nella creazione libera di LO. Questo tipo di soluzione permette di semplificare l'approccio ad AContent da parte di utenti neofiti, facilitando così l'ampliamento del bacino di utenza della piattaforma.

L'ultimo importante obiettivo che si è voluto conseguire è stato quello di permettere all'utente di creare LO seguendo dei vincoli/opzioni che l'autore potrebbe liberamente decidere di attivare oppure essere obbligato a seguire in modo da creare corsi con determinate caratteristiche. Si è così introdotto una costruzione assistita delle strutture a partire dagli obiettivi. Questo perché se si sceglie un certo obiettivo formativo si dovrà poi creare un certo tipo di contenuto ed attività da svolgere che permetta di soddisfarlo. Per quanto riguarda sviluppi futuri vi sono due vincoli/opzioni che sono state predisposte all'interno di *wizard* oltre a quelli riguardanti gli obiettivi formativi. Esso le gestisce consentendo agli utenti di abilitarle o disabilitarle all'interno delle proprie lezioni. La prima è l'*accessibility*, che garantisce che i LO creati siano del tutto accessibili ad utenti diversamente abili. La seconda funzionalità è quella del *copyright* un modo che permette di assegnare ad ogni file, dalle immagini ai video, che l'utente inserisce nelle proprie lezioni una didascalia che riporti l'autore.

Bibliografia

- [FaD1] Guglielmo Trentin, *Dalla formazione a distanza all'apprendimento in rete* FrancoAngeli, 2001 .
- [FaD2] ASFOR. Il glossario di E-learning di ASFORD. Edizione 2003
www.asfor.it/LETTERA%20ASFOR/Lettera%20Asfor%202003%20nr%203.pdf .
- [IpC] D. Parmigiani, *Tecnologie per la didattica. Dai fondamenti dell'antropologia Multimediale all'azione educativa* , FrancoAngeli, 2004 .
- [E-lea] ANEE (Italia), *E-learning. Stato dell'arte e prospettive di sviluppo*, Apogeo Editore, 2004.
- [LMS1] Giuseppe Esposito, Giuseppe Mantese, *E-learning: una guida operativa. Come realizzare e valutare un progetto*. FrancoAngeli, 2003 .
- [LO] Articolo di Corrado Petrucco, Learning Objects: un innovativo supporto all'e-learning?, www.edscuola.it/archivio/software/learning_objects.pdf .
- [IMS-G] IMS Global, www.imsglobal.org .
- [DC] Dublin Core che, www.dublincore.org .
- [ARI] ARIADNE, www.ariadne.it .
- [IEEE] Istituto degli ingegneri elettrici ed elettronici, www.ieee.org .
- [Aut1] W3C. Authoring Tool. www.w3.org/WAI/intro/atg.php .

-
- [Aut2] Centro Nazionale per l'Informatica nella Pubblica Amministrazione.
Authoring Tool Editor ,
www.pubbliaccesso.gov.it/biblioteca/manualistica/accessibilita_siti/tool/authoring_tool.htm.
- [GeoG1] GeoGebra , www.geogebra.org .
- [RD2] Reader 2D, www.limparo.it/reader_2d.html .
- [CLIL] Content and Language Integrated Learning, www.universoclil.ue .
- [WYSIWYG] Articolo di Margaret Rouse, Marzo 2011,
www.whatis.techtarget.com/definition/WYSIWYG-what-you-see-is-what-you-get .
- [XPAR] Xerox Palo Alto Research Center, www.parc.com .
- [CMS1] Content Management System www.atout.it/cms.html .
- [CMS2] Content Management System, articolo del 3 Settembre 2007,
blog.joocode.com/cms/cose-un-cms-content-management-system/ .
- [WIKI] Bo Leuf, Ward Cunningham, The Wiki way: quick collaboration on the Web,
Addison-Wesley, 2001 .
- [Nup] Nupedia Wiki, www.nupedia.wikia.com .
- [Wikip] Wikipedia, l'enciclopedia libera www.it.wikipedia.org/wiki/Wikipedia:A_proposito_di_Wikipedia
- [eXel] Exelearning , www.exelearning.it .
- [SMIL] Synchronized Multimedia Integration Language , www.a-smil.org .
- [W3C] World Wide Web Consortium , www.w3c.it .
- [LMS2] Lorenzo Cantoni, *Elearning. Capire, progettare, comunicare* , Franco Angeli,
2007.
- [LCMS] Learning Content Management System , www.outstart.com .
- [Moo] Modular Object-Oriented Dynamic Learning Environment , www.moodle.org

-
- [GPL] General Public License , www.gnu.org , www.it.gnu.org .
- [M.Dou] Martin Dougiamas , Tesi per il Phd mai realizzata secondo questa fonte
<http://dougiamas.com/theis/> .
- [Costr] Costruzionista , <http://it.wikipedia.org/wiki/Costruzionismo> .
- [Cla] Claroline , www.claroline.net .
- [ATu] ATutor, www.atutor.ca .
- [ChangeL] Multisite, informazioni sull'ultima funzionalità di ATutor,
https://atutor.ca/atutor/change_log.php .
- [IMS GLC] IMS Global Learning Consortium, www.imsglobal.org .
- [IMS CP] IMS Content Package, www.imsglobal.org/content/packaging/ .
- [IMS CC] IMS Content Cartridge, www.imsglobal.org.commoncartridge.html .
- [DC] Dublin Core, www.dublincore.org .
- [IMS LTI] IMS Learning Tools Interoperability, www.imsglobal.org/lti/index.html .
- [WS] Web Service, www.w3.org/TR/ws-arch .
- [WSR] Web Service REST, www.ics.uci.edu .
- [WSS] Web Service SOAP, www.w3.org/TR/soap .
- [IMS QTI] Cristina Delogu, *Tecnologia per il web learning. Realtà e scenari*, Firenze University Press, 2008 .
- [QM] Questionmark, www.questionmark.com .
- [SCROM] Specifiche SCROM,
www.imsglobal.org/ssp/sspv1p0/imssp_prflv1p0.html
- [PIF] Package International File,
<http://ijedict.dec.uwi.edu/viewarticle.php?id=853&layout=html>

-
- [AC1] AContent, www.atutor.ca/accontent/ .
- [LStanca1] Legge Stanca, www.governo.it/presidenza/web/regolamento09072004.html .
- [LStanca2] Testo Legge Stanca, www.camera.it/parlam/leggi/040041.htm .
- [Latex] LaTeX, linguaggio di markup, www.latex-project.org .
- [PHP1] Hypertext Preprocessor, www.php.net .
- [PHP2] Peter MacIntyre, PHP. Le tecniche per scrivere il codice migliore, Tecnologie Nuove , 2010.
- [Zend] Zend, www.zend.com .
- [Apa] Server Apache, www.apache.org .
- [HTML] HyperText Markup Language, www.html.it .
- [HTML2] Anders Møller, Michael Schwartzbach, *Introduzione alle tecnologie WEB*, Pearson Italia S.p.a. , 2007 .
- [JS] Christian Wenz, *In tasca Javascript*, Pearson, 2007 .
- [SQL] E. Kevin Kline, Daniel Kline, *Sql Guida Di Riferimento*, Apogeo Editore,2001 .
- [WAMP] Piattaforma di sviluppo web/database, www.wampserver.com .
- [EPHP] EasyPHP, www.easyphp.org .
- [Perl] Linguaggio di programmazione Perl, www.perl.it , www.perl.org .
- [Python] Linguaggio di programmazione Python, www.python.it , www.python.org .
- [MAMP] MAMP, www.mamp.info/en/mamp/index.html .

- [MPRO] MAMP PRO, www.mamp.info/en/mamp-pro/index.html .
- [MYSQL] MySQL, www.mysql.com .
- [BEAT] BEAT: an AContent Extension to Support Authors in Developing Learning Pathways, 2012. Salomoni P., Mirri S., Prandi C.
- [LCDS] Microsoft LCDS, www.microsoft.com/learning/en/us/training/lcds.aspx .

Ringraziamenti

Voglio aprire la carrellata dei miei ringraziamenti citando per prima la Prof.ssa Paola Salomoni, per la sua disponibilità nell'affidarmi questo progetto di tesi. Con lei, devo ringraziare la Dott.ssa Catia Prandi, la quale mi ha dedicato innumerevole tempo, seguendomi passo passo durante tutta la fase di progettazione di Wizard e offrendomi ottimi consigli e suggerimenti lungo il cammino che ho seguito per la stesura di questa tesi. Un grazie doveroso va anche a tutti i docenti con i quali ho avuto il piacere di interagire in questi anni, per la loro disponibilità e per il loro supporto.

Ringrazio la mia famiglia, che mi ha sempre sostenuto senza interferire con le mie decisioni, lasciandomi libero di gestire questo mio tortuoso cammino universitario.

Infine, l'ultimo ringraziamento è per la persona senza la quale probabilmente non avrei avuto la forza e lo stimolo per finire questo percorso accademico. Grazie Giorgia per avermi sostenuto e spronato a raggiungere questo traguardo, ma soprattutto per essermi accanto da ormai cinque bellissimi anni.