Zenaminer: lo standard SCORM incontra il Web of Data

Zenaminer: lo standard SCORM incontra il Web of Data

Ernesto Mudu, Luca Schiatti, Giuseppe Rizzo, Antonio Servetti, Laura Farinetti
Dipartimento di Automatica ed Informatica - Politecnico di Torino
C.so Duca degli Abruzzi, 24, 10129 Torino, Italia
{ernesto.mudu, luca.schiatti}@gmail.com, {giuseppe.rizzo, servetti, laura.farinetti}@polito.it

Sommario: In questo articolo presentiamo Zenaminer, un progetto realizzato all'interno del corso di Ambienti per la Multimedialità al Politecnico di Torino. L'idea alla base del progetto è quella di ovviare ad alcune limitazioni dello standard SCORM, il formato più utilizzato per rendere riutilizzabili i contenuti per l'E-Learning, andando incontro ai principali paradigmi di innovazione emersi con il Web 2.0. L'idea cioè del *web dei dati* e del *web come piattaforma* (web of data and web as platform). In questa ottica proponiamo una architettura web per esportare un pacchetto SCORM come un servizio REST (Web Service REST) dove ogni singolo SCO è esposto sulla rete in modo dinamico e può essere combinato al volo in una qualsiasi piattaforma di E-Learning, senza più la necessità di doverlo scaricare e scompattare. Vista l'importanza didattica del design dell'interfaccia in un contesto di E-Learning, a ciò è aggiunta la possibilità di poter ridefinire completamente l'aspetto grafico e le modalità di presentazione delle informazioni. Ridefinizione che è resa possibile dalla rigorosa separazione tra stile e contenuti nella realizzazione del materiale, pur rimanendo sempre in ambito di markup HTML. Infine, il modello SCORM è ulteriormente esteso introducendo una componente collaborativa in ottica costruttivista. L'apertura cioè dei contenuti agli studenti stessi che possono assumere un ruolo attivo, e quindi più fecondo, arricchendo e valutando il materiale didattico originale fornito dal docente.

Parole chiave: SCORM, E-Learning, web services, REST, learning object
Article ID: xxxxxx (defined by the editor)

Introduzione

L'evoluzione del settore delle telecomunicazioni e la centralità acquisita dal Web nella nostra società hanno portato a significativi cambiamenti anche nell'ambito dell'E-Learning tanto da rinominarlo con il termine "E-Learning 2.0" [3].
Delle molte facce con cui può essere descritto il passaggio dal sistema 1.0 a quello 2.0, in relazione con l'analoga trasformazione realizzatasi per il Web, vogliamo in questo lavoro sottolineare l'aspetto definito come "Web of Data".

Il Web 2.0 ha portato ad una visione della rete in cui l'informazione è scomposta in "microcontenuti distribuiti", si è passati cioè dal Web "dei documenti" al Web "dei dati" [1]. Il vantaggio nello scomporre il modello statico di un documento in quello di un'aggregazione dinamica di dati è appunto quello di offrire al lettore la possibilità di ri-aggregare e re-mixare i contenuti in modi e modalità nuove, creando nuova informazione.

In pratica, prendendo in considerazione un'altra faccia del Web 2.0, il Web si è trasformato da

"canale" passivo su cui veicolare l'informazione a "piattaforma" attiva con cui creare, condividere, remixare, riproporre nuove informazioni. Si è passati dunque da un Web "a sola lettura" ad un Web "a lettura/scrittura" (read/write Web [8]), da un mezzo ad uno strumento dove gli utenti possono diventare – e di fatto lo sono diventati – produttori di contenuti.

Questi principi non sono affatto nuovi per l'E-Learning dove il concetto di "materiale didattico" è definito in termini di "micro-contenuti" denominati "Sharable Content Objects (SCO)"[1]. Gli SCO sono alla base dei concetti di usabilità, interoperabilità e adattatività, costituiscono infatti le porzioni di dati che possono essere individualmente prodotte, archiviate, indicizzate, assemblate e valutate.

Anche in questo ambito, se i primi modelli intendevano l'insegnamento organizzato in corsi e lezioni con obiettivi pre-definiti, i moderni paradigmi invitano alla scomposizione dei contenuti in blocchi più ridotti, auto-contenuti, che possano essere utilizzati indipendentemente o assemblati dinamicamente.

Gli SCO possono essere considerati un compromesso tra i dati grezzi (es. immagini, video, note, audio) e la lezione intera: sono elementi inseriti in un contesto (ridotto), ma allo stesso modo rimangono flessibili perché i loro contenuti possono essere ri-arrangiati facilmente per formare un "nuovo" percorso didattico, o ancora, possono cambiare aspetto disponendo diversamente il testo e le immagini e cambiando il formato di titoli e loghi. Per esempio, in questo lavoro abbiamo considerato come SCO differenti i gruppi di slide di una lezione corrispondenti ad uno specifico argomento.

In parallelo, quindi, con la produzione di materiale per l'E-Learning si vengono a costituire archivi di "microcontenuti", i cui elementi, gli SCO appunto, sono sì "sharable" (condivisibili), ma non vanno in alcun modo a fare parte di ciò che abbiamo definito come "Web of Data". Vengono infatti condivisi come "documenti", ad esempio archivi ZIP, che un utente deve scaricare e "scomporre" autonomamente per poterli riutilizzare e integrare in un nuovo progetto.

L'idea del "web come piattaforma" [6] vorrebbe invece che gli SCO venissero esportati sul web attraverso una "Application Programming Interface (API)" che ne permettesse in modo automatico l'interrogazione, il riuso ed, eventualmente, la modifica o l'aggiunta di informazioni. In ottica 2.0 un "Learning Object Repository (LOR)" verrebbe visto come un "Web Service (WS)" dove, brevemente, ogni elemento corrisponde ad una risorsa Web (e quindi associato ad un Uniform Resource Identifier) che può essere manipolata attraverso i metodi del protocollo HTTP (GET, PUT, POST, and DELETE) dell'architettura Representation State Transfer (REST) [4]. Esempi significativi di WS sono Google Maps, per le informazioni geografiche, Flickr, per le foto e le informazioni ad esse associate, Youtube, per i video, e molti altre ancora[2].

Il compito che ci siamo prefissati in questo lavoro è, dunque, quello di definire e implementare un modello con cui esportare sulla rete, in ottica Web 2.0, del materiale didattico "impacchettato" in modo conforme allo standard SCORM. Alcuni lavori come Educateca [7] e LearnServe [10] sono già andati in questa direzione, ma con l'obiettivo, sostanzialmente differente, di trasformare in Web Services le funzionalità del LMS (ad esempio esportando lo SCORM CAM e lo SCORM RTE) e non, come si prefigge invece questo progetto, i dati "grezzi" dello SCORM stesso.

L'obiettivo non è soltanto quello di definire una nuova interfaccia verso i "microcontenuti" didattici, ma quello, ancor più significativo, di ridare loro vita facendoli entrare a far parte del Web

[1] http://www.adlnet.gov/Technologies/scorm/default.aspx

[2] si veda http://www.programmableweb.com/apis/directory per un elenco dettagliato

of Data. Un mondo dove sono gli utenti stessi a creare il valore aggiunto purché sia data loro la possibilità di accedere ai dati in forma grezza. Il valore non sarà più soltanto nei dati stessi, ma nell'uso che ne verrà fatto, in come verranno combinati in nuovi modi, in come verranno presentati con nuove interfacce, in come verranno arricchiti andando oltre il contenuto informativo originario predisposto dal docente.

Il seguito del documento è organizzato come segue. Nella Sezione 1 verrà presentata l'idea di esporre lo SCORM attraverso un Web Service REST. La Sezione 2 tratterà del concetto di separazione della forma dal contenuto, esportato dal modello della programmazione Web. Nella Sezione 3 verrà mostrata l'architettura del Web Service, mentre nella Sezione 4 è descritto il caso di studio usato. Infine, nella Sezione 5 verranno analizzati i vantaggi di questa soluzione.

1 Un Web Service per lo SCORM

Lo SCORM (Sharable Content Objects Reference Model) è un modello di riferimento per creare pacchetti per l'E-Learning. L'obiettivo del modello SCORM è di creare pacchetti che siano riusabili, interoperabili e accessibili. Ogni pacchetto si riferisce ad un corso e comprende al suo interno una serie di lezioni oggetti di apprendimento, chiamati SCO (Sharable Content Object). Per aderire allo standard SCORM, i contenuti di ciascuno SCO devono essere visualizzabili attraverso il Web. Il contenuto di uno SCO, può ad esempio essere un file in formato HTML contenente delle slide, la cui visualizzazione avviene attraverso un browser web. Il progettista di tali SCO, cioè colui che, avendo i contenuti, struttura fisicamente l'informazione (ad esempio in forma di lucidi) è detto progettista SCO.

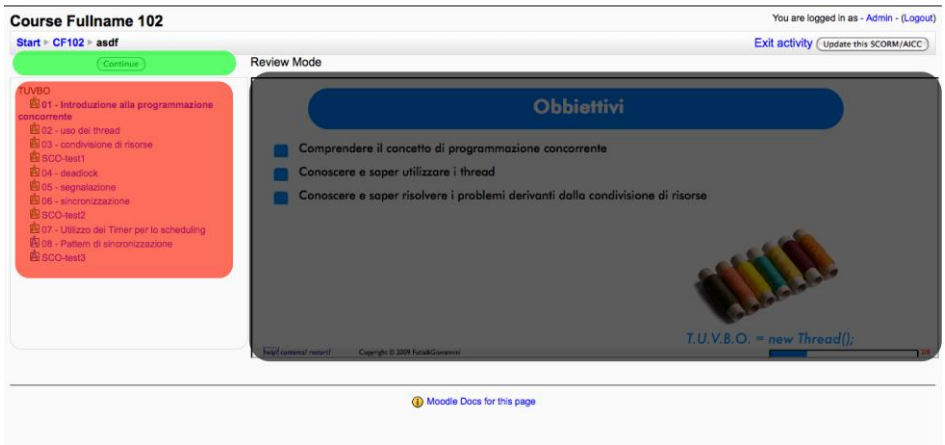

Fig. 1: Screenshot dell'interfaccia del Learning Management System Moodle. A sinistra in basso la lista delle lezioni disponibili nel pacchetto SCORM, a sinistra in alto i bottoni di navigazione tra le varie lezioni e l'area nella quale sono mostrati i lucidi delle lezioni.

I pacchetti SCORM vengono acquisiti e mostrati ai fruitori di un corso (consumatori, generalmente studenti) tramite un Learning Management System (LMS). I LMS sono in grado di comprendere la struttura del pacchetto SCORM, individuare le lezioni (SCO) di cui è composto e renderle disponibili agli studenti. I LMS a cui facciamo riferimento in questo articolo, sono quelli accessibili attraverso un browser Web. Dato che frequentemente le lezioni sono costituite da slide, l'approccio di un LMS (che in questo contesto definiamo “tradizionale”) è quello di creare una pagina standard che include: la lista delle lezioni disponibili nel pacchetto, i bottoni di navigazione

tra le varie lezioni e un'area nella quale sono mostrati i lucidi delle lezioni. In Figura 1 l'area a destra è dunque lo spazio che un "LMS tradizionale" dedica allo SCO. Questa è l'area a disposizione del progettista SCO. Questo tipo di interfaccia presenta una serie di problematiche:

- il progettista SCO ha completa libertà nel definire il singolo SCO ma non ha nessun controllo sull'interfaccia che gestisce il collegamento tra gli SCO (demandata al LMS).
- lo spazio di azione è limitato alla porzione di schermo che il LMS concede: risulta quindi più complessa ed è meno efficace l'integrazione di contenuti diversi appartenenti allo stesso SCO, ad esempio se la lezione è composta dalle slide più un video del fornitore (generalmente il docente) che spiega usando tali lucidi.

Gli SCO vengono importati nel LMS con lo stile grafico che il progettista SCO ha deciso di dare, nel momento in cui SCO prodotti da progettisti differenti sono integrati la differenza di stili risalta, minimizzando la riusabilità. Se forma e contenuto fossero separati, sarebbe facile per un progettista SCO prendere il contenuto di un altro SCO, riadattandolo secondo i propri bisogni di visualizzazione. Infine, un'interfaccia statica e l'impossibilità di astrarre lo stile dal contenuto rendono difficile adattare l'interfaccia del LMS a differenti dispositivi di visualizzazione (es.: smartphone, tablet, netbook, etc.).

Alla luce dei limiti di questo approccio, in questo articolo presentiamo Zenaminer, un LMS che supera i limiti dei LMS tradizionali proponendo un'architettura di tipo Web Service REST, un'architettura cioè dove il pacchetto SCORM viene condiviso non più come un mattoncino unico (il pacchetto ZIP per intenderci), ma come un insieme di componenti, gli SCO appunto, ciascuno usufruibile indipendentemente dagli altri. Si passa quindi dall'esposizione sul web di un "documento" alla pubblicazione dei suoi dati (grezzi) e alla possibilità di leggerli e arricchirli (r/w web) tramite l'interfaccia REST, come di seguito illustrato.

La chiave di ciò è la capacità di Zenaminer non soltanto di importare un pacchetto SCORM, ma anche di riconoscere i formalismi con cui sono stati definiti i contenuti dei singoli SCO, separando quindi in modo selettivo la forma dal contenuto. Si è quindi in grado di riconoscere le singole slide all'interno di una presentazione e salvare i contenuti separati dallo stile imposto dal progettista SCO. Ciò è possibile grazie all'introduzione di un "formalismo leggero" nella creazione dei contenuti HTML prodotti utilizzando uno strumento del W3C denominato Slidy[3]. Slidy definisce semplici parole chiave da assegnare all'attributo *class* dei tag HTML per strutturare il documento e poter identificare slide, titoli, sidecar, elenchi su cui applicare particolari stili di visualizzazione con i CSS o azioni tramite Javascript.

Alla fine della procedura di importazione del pacchetto SCORM, i contenuti sono disponibili mediante delle URI, diventano cioè delle *risorse* del Web Service. Il progettista SCO può liberamente disegnare l'interfaccia che meglio risponde alle esigenze di presentazione dei contenuti del produttore e del consumatore richiedendo in maniera puntuale e dinamica (tramite AJAX) i contenuti da Zenaminer solo quando necessario e solo quelli necessari. Se dunque gli SCO sono identificati nel modello SCORM come l'unità minima in termini di riusabilità, interoperabilità e accessibilità che un progettista SCO può (con tutti i limiti sopra descritti) riusare nella creazione di un corso, tali SCO non sono più limitati ad essere solo delle lezioni, come è prassi negli LMS tradizionali, ma possono anche essere le singole slide, cioè il più piccolo atomo d'informazione presente all'interno di una lezione.

[3]http://www.w3.org/Talks/Tools/Slidy2/

In aggiunta, Zenaminer permette di affiancare ad ogni SCO contenuti e informazioni aggiuntive come ad esempio il video della lezione contenuta nello SCO, dati di sincronizzazione tra il video e le slide, ecc..

Per quanto riguarda invece il classico approccio all'E-Learning 2.0 in termini di collaborazione tra gli studenti e tra questi ed il docente, Zenaminer permette al "consumatore" di arricchire i contenuti deli SCO inserendo commenti nei tre principali livelli gerarchici: il corso, la lezione ed il lucido. I commenti sono potenzialmente visibili e modificabili da tutti coloro che usufruiscono del corso, ma l'abilitazione a poter fare ciò è demandata al progettista SCO che ne gestisce la presentazione. Benché già questa funzionalità sia sufficiente ad abilitare, di fatto, la cooperazione tra "consumatori", in Zenaminer proponiamo anche il superamento del limite di *carenza di informazione* intrinseco del testo semplice tramite l'utilizzo dell'approccio Linked Data [1]. Tutti i commenti inseriti sono automaticamente annotati con link alle risorse presenti in DBpedia [2]. Questa annotazione contestualizzata (elaborando il testo, Zenaminer è in grado di capire il contesto) permette all'utente di navigare tra le risorse della Wikipedia correlate al testo ed esplodere quindi la rete di inferenze dei propri commenti o di quelli di altri consumatori, di fatto aumentando il proprio bagaglio culturale.

2 Presentazione dei contenuti

Al fine di agevolare la presentazione dei contenuti si è scelto di utilizzare per la loro descrizione l'HTML ed in particolare il formalismo definito da Slidy. Esso permette di rappresentare in modo strutturato i contenuti, e inoltre prevede la divisione tra presentazione (anche detta vista) e contenuto stesso. Quindi, parallelamente alla definizione dei dati, si può definire come la vista deve essere creata: questo lo si realizza attraverso i fogli di stile CSS. In essi, si possono definire modelli di visualizzazione associati a specifici elementi della pagina o ad insiemi di essi, chiamati classi. L'interazione tra il foglio di stile e una pagina HTML viene resa possibile attraverso l'uso di selettori, che sono in grado di selezionare in modo puntuale elementi specifici o classi di una pagina HTML. In questo contesto, un terzo attore è il JavaScript, un linguaggio di "scripting" che fornisce la possibilità di accedere selettivamente ad elementi della pagina, potendone cambiare la visualizzazione. Esso, inoltre, agisce come strumento di arricchimento dinamico della pagina, ad esempio creando tabelle dei contenuti (TOC), espandendo acronimi, modificando il font o la dimensione delle finestre create. Inoltre, svolge un ruolo chiave per l'accesso dinamico di informazioni strutturate, ma prive di contenuto visuale, che sono residenti all'interno dello spazio Web. La tecnologia che permette di interrogare risorse esterne è l'Asincronous Javascript and XML (AJAX).

La Separation of Concerns (SoC) è la chiave innovativa di Zenaminer. La netta distinzione che vi è tra vista e dati permette di navigare tra le risorse sviluppando delle viste che meglio rispondono alle esigenze del fornitore o del consumatore. I dati relativi ai contenuti veicolati ai consumatori sono esposti in maniera grezza, privi di alcun tipo di formattazione visuale, ma ricchi di contenuto semantico. Questo approccio implementa il modello MVC sopra descritto, nel quale la Vista è l'insieme dei dati utili alla presentazione, il Modello sono i dati grezzi, privi di formattazione, mentre il Controllore è l'insieme dei metodi che servono a creare il canale di comunicazione tra il modello e la vista. Il Modello dei dati sarà residente sul sistema che ospiterà i pacchetti SCORM, mentre la Vista sarà istanziata sulle macchine dei consumatori ogniqualvolta decidano di navigare i contenuti. Il Controllore, invece, svolgerà il ruolo di mediatore ed avrà una

componente residente sulla macchina del consumatore ed un componente (il gestore delle richieste) sul LMS che ospiterà i pacchetti SCORM.

Con l'obbiettivo di fornire la massima interoperabilità e riusabilità, i contenuti accessibili dal LMS sono lucidi, lezioni e corsi. Tutti questi contenuti sono delle risorse Web, accessibili mediante delle URI tempo invarianti, fornendo così durabilità alla risorsa. I contenuti dei lucidi sono descritti attraverso il formalismo di HTML Slidy; ogni lucido è composto da un titolo, immagini, barre laterali o di scorrimento, elementi interattivi. Le lezioni, invece, sono l'insieme di lucidi e frammenti video, ossia porzioni dello stesso video relativo alla lezione corrente. Per mantenere l'associazione tra lucido visualizzato e frammento di video riprodotto, si è scelto di associare all'elemento lezione e all'elemento video un file di sincronizzazione. Il video, così come per la risorsa lucido, è accessibile mediante una URI all'interno dello spazio del Web. Corso, lezione e lucido implementano la SoC, separando la vista dai dati stessi. L'unica informazione che conserva lo stile è il video, il quale viene trasmesso senza modifica.

3 Architettura

I blocchi costitutivi del server sono mostrati in Figura 2:

1. il REST controller è l'interfaccia di Zenaminer verso Internet. Raggruppiamo le funzionalità in tre insiemi: gestione autenticazione utente (login), invio e reperimento dei commenti (comment) e gestione dei pacchetti SCORM importati (scorm, item, outline, sync e slide). Ognuna di queste funzioni è implementata secondo il modello REST e rende disponibile le risorse tramite URI nello spazio Web.
2. Un validatore locale è usato per verificare l'integrità dei file in formato (X)HTML prima che vengano elaborati. Quando un nuovo pacchetto SCORM viene importato, i file HTML contenenti i lucidi vengono singolarmente validati inviando delle richieste al validatore ufficiale del W3C[4].
3. Ogni volta che Zenaminer riceve un commento (arricchimento del testo) questo viene passato al client Spotlight che inviando una richiesta al servizio Spotlight di DBpedia[5] ne effettua l'annotazione automatica.
4. Il database viene usato per il salvataggio delle informazioni relative a ciascun pacchetto SCORM importato, compresi i commenti e le annotazioni.

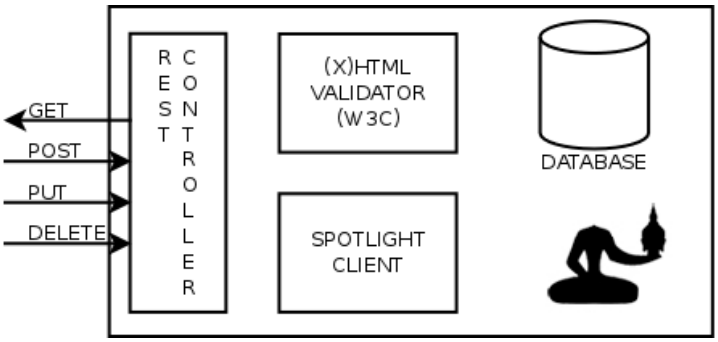

Fig. 2: L'architettura di Zenaminer.

In Figura 3 si vede come i singoli componenti dell'architettura di Zenaminer interagiscono nell'importazione di un pacchetto SCORM e nella ricezione di un commento. I passi logici sono:

[4]http://validator.w3.org/
[5]http://dbpedia.org/spotlight

1. un client invia una richiesta POST alla pagina /scorm di Zenaminer, inserendo nel corpo della richiesta il pacchetto SCORM da inviare;
2. il REST controller riceve il pacchetto, lo analizza e identifica i file contenenti i lucidi, invia i file al validatore W3C;
3. il validatore W3C invia una richiesta per ogni file da validare e comunica al REST controller i risultati;
4. se tutti i file sono validati correttamente, l'elaborazione del pacchetto prosegue e il suo contenuto è mappato nel Database. Se i file non sono correttamente validati viene inviato un messaggio di errore al client, il cui corpo contiene gli errori generati dal validatore W3C.

In aggiunta, la Figura 4 mostra la ricezione di un commento:

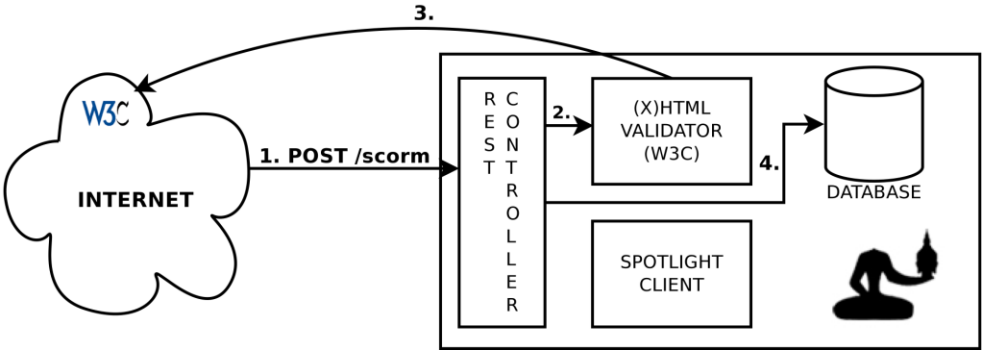

Fig. 3: Workflow dell'upload di un pacchetto SCORM

1. un client invia una richiesta POST alla pagina /comment di Zenaminer, il corpo di tale messaggio contiene il testo del commento;
2. il messaggio è inviato al client Spotlight.
3. ricevuto il messaggio, il client Spotlight invia una richiesta al servizio Spotlight di DBpedia e riceve come risposta una versione annotata del messaggio.

Il commento e le relative annotazioni sono salvati nel database.

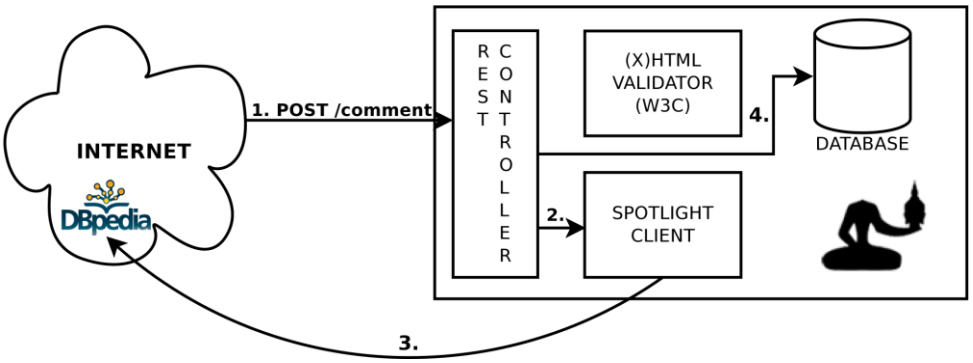

Fig. 4 Workflow della gestione dei commenti ricevuti

Un pacchetto SCORM è un archivio zip che include un file (in formato XML) chiamato "manifest", che elenca il titolo del corso, i titoli delle lezioni appartenenti al corso, le risorse (i file) associati ad ogni lezione e le eventuali regole per passare da uno SCO al successivo (regole di sequencing). La struttura ad albero in Figura 5 mostra come il file manifest presente nel pacchetto SCORM è mappato nel database relazionale di Zenaminer. In un package sono definiti vari SCO e per ciascuno di essi è presente una risorsa HTML con i lucidi, un file CSS ed uno Javascript (che

ne definiscono stile e comportamento sempre nell'ottica della SoC, e che possono essere sostituiti nella vista finale proposta all'utente) e altre risorse generali dello SCO come, ad esempio, un

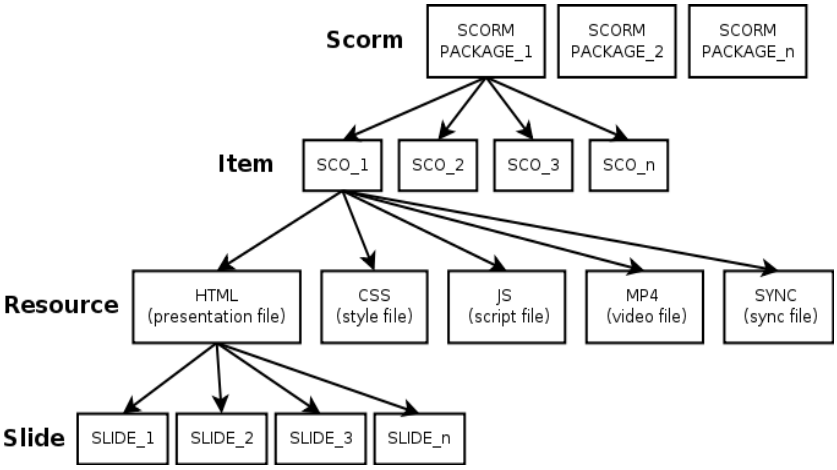

Fig. 5: La rappresentazione ad albero di un pacchetto SCORM all'interno del database di Zenaminer

video e le informazioni di sincronizzazione tra questo e le slide (tali risorse aggiuntive potrebbero anche essere sottotitoli o altro). La versione attuale di Zenaminer però non riconosce le regole di sequencing associate ad un pacchetto SCORM, dunque la gestione di tali regole è lasciata agli Progettista SCO, che in autonomia decidono che libertà di navigazione tra i contenuti concedere agli studenti.

4 Caso di studio

La fase di validazione del Web Service è stata condotta dagli studenti del corso di Ambienti per la Multimedialità nell'anno accademico 2010/2011. Il corso fa parte del II anno della laurea Specialistica in Ingegneria del Cinema e dei Mezzi di Comunicazione del Politecnico di Torino. Lo scopo didattico dello strumento è una delle chiavi del lavoro. Esso infatti è stato concepito come uno strumento di supporto allo sviluppo dei progetti di E-Learning (pacchetti SCORM) del laboratorio didattico del corso sopra citato. La consegna dei progetti prevedeva di ricreare le lezioni di alcuni corsi didattici di Ingegneria del Cinema e dei Mezzi di Comunicazione, unendo l'aspetto testuale (testo descrittivo ed esercizi) a quello visuale (immagini, video) e di includerli all'interno di oggetti SCORM. Gli studenti sono stati suddivisi in gruppi, per un totale di 20, i quali per ogni consegna hanno potuto personalizzare la vista dei contenuti (sfruttando il concetto di SoC) e creando viste differenti.

Gli studenti sono diventati progettisti SCO, fornendo delle viste ai contenuti dei corsi. Il materiale sviluppato è diventato il caso d'uso di Zenaminer, potendo così verificare la potenzialità della separazione tra forma e contenuto SoC. E' possibile consultare la demo in cui è possibile navigare tra le varie viste (http://eridano.polito.it:8080/).

Il Web Service è stato sviluppato usando il framework Pylons 1.0[6]. Il codice è liberamente scaricabile[7] con licenza GPLv3. Per ragioni di bilanciamento e distribuzione del carico si è deciso di utilizzare Apache 2.2 HTTP Server[8], come interfaccia tra le richieste provenienti dalla nuvola

[6]http://pylonshq.com
[7]https://sourceforge.net/projects/zenaminer/
[8]http://httpd.apache.org

Internet verso il Web Service, strumento di presentazione e di archiviazione dei contenuti. Al fine di poter salvare le informazioni in modo permanente, si è utilizzato il gestore della base dati PostgreSQL 9[9]. Esso ha svolto tutte le funzioni di archiviazione dei dati grezzi, elaborati dal Web Service a partire dall'oggetto SCORM, in tabelle relazionali. Il validatore W3C è stato utilizzato per validare i contenuti descritti in HTML delle lezioni. Qualora i contenuti non rispettassero le specifiche del W3C dei messaggi di errore vengono presentati all'autore della lezione. Infine, si è utilizzato il servizio Spotlight per generare delle annotazioni contestualizzate sui commenti introdotti dal consumatore.

5 Conclusioni

In questo articolo abbiamo presentato e discusso un Web Service per l'E-Learning che, superando alcuni limiti del modello SCORM, permette la completa separazione tra contenuto e visualizzazione. L'approccio adottato consente ai creatori di contenuti di definire l'ambiente di apprendimento in modo libero e non vincolato dalle esigenze del LMS con cui operano: Zenaminer consente, quindi, una completa libertà nella definizione delle interfacce, nella scelta delle grafiche e delle modalità di interazione tra l'utente ed i singoli SCO.

Consideriamo inoltre che, nel contesto dell'E-Learning, la visualizzazione (l'interfaccia) non coinvolge solamente aspetti estetici e di usabilità ma incide direttamente nel processo di apprendimento dell'utente coinvolto: uno dei fattori di successo di un corso E-Learning è proprio il design dell'interfaccia [9] e l'adattabilità di essa relativamente ai bisogni dell'utente [5].
La 'Separation of Concerns (SoC)' permette, ad esempio, di creare ambienti diversi per lo stesso corso in accordo con il grado di alfabetizzazione informatica degli utenti, oppure consente di incrementare il grado di interoperabilità degli SCO permettendo la definizione di ambienti diversi a seconda del dispositivo con cui essi verranno usufruiti (es. smart phone).

Non secondaria è la possibilità di integrare e gestire contenuti multimediali come entità separate rispetto ad un certo contenuto didattico. Con un LMS 'tradizionale' un contenuto multimediale deve essere integrato nel contenuto di uno SCO per essere visualizzato, mentre, nella proposta presentata in questo articolo i video, e più in generale un media diverso dal testo, possono essere gestiti e integrati a proprio piacere rispetto al contenuto di una slide o di una lezione. Inoltre queste possibilità per il creatore di contenuti non compromettono la retro-compatibilità di uno SCORM che potrà essere comunque utilizzato e riconosciuto da un LMS "tradizionale".

Infine, Zenaminer permette al creatore del corso di implementare un modello di didattica collaborativa all'interno delle sue lezioni. Lo studente può assumere un ruolo attivo nella definizione dei contenuti grazie alla possibilità di arricchire il contenuto delle lezioni con dei propri contributi. Anche in questo caso è il creatore di contenuti che può decidere i meccanismi di collaborazione scegliendo, ad esempio, se gli studenti possono interagire in un modello stile Wiki, oppure in un modello con un grado minore di collaborazione.
Alla didattica collaborativa si aggiunge inoltre la possibilità di espandere i contenuti grazie agli strumenti offerti dal Web Semantico. L'annotazione semantica del testo consente di incrementare il valore informativo dei commenti aggiungendo in modo automatico dei collegamenti ipertestuali verso i contenuti della Wikipedia che permettono agli studenti di approfondire ed integrare le proprie conoscenze avendo a disposizione altri materiali più approfonditi relativi ai contenuti delle

[9] http://www.postgresql.org/

lezioni. Inoltre, in un ottica di didattica collaborativa, gli studenti potrebbero anche agire su queste annotazioni automatiche disambiguandole o integrandole.

L'architettura proposta in questo articolo consente al creatore di contenuti un alto grado di libertà nell'attuazione delle teorie dell'insegnamento (es. Cognitivismo, Costruttivismo) e nella scelta delle strategie di insegnamento che ritiene più adatte. Ad esempio il creatore di contenuti potrebbe scegliere di minimizzare il contenuto del corso ed enfatizzare l'arricchimento delle nozioni grazie agli strumenti della didattica collaborativa: questo approccio interpreta il costruttivismo che ritiene l'aiuto reciproco e la collaborazione tra studenti uno dei punti chiave per l'apprendimento [11].

Ringraziamenti

Un ringraziamento particolare va agli studenti del corso di Ambienti per la Multimedialità a.a. 2010-2011, i quali sono stati sviluppatori e validatori del Web Service, in un particolare spirito di cooperazione che ha coinvolto tutti ottenendo questo importante obbiettivo.

Riferimenti:

[1] Berners-Lee, Tim, Linked Data. International Journal on Semantic Web and Information Systems, W3C, 2006.

[2] Bizer C., Lehmann J., Kobilarov G. and Auer S., Becker C., Cyganiak R., Hellmann Sebastian, DBpedia - A crystallization point for the Web of Data, Web Semantics: Science, Services and Agents on the World Wide Web, pp. 154—165, 2009.

[3] Downes, S., E-learning 2.0, ACM, 2005.

[4] Fielding, Roy T.; Taylor, Richard N., Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery) 2 (2): 115–150, 2002.

[5] Mödritscher F., Manuel V., Barrios G., Gütl C., Enhancement of SCORM to support adaptive E-Learning within the Scope of the Research Project AdeLE. World Conference on E-Learning (E-Learn), AACE, 2004, 2499-2505.

[6] Murugesan S., Understanding Web 2.0. IT Professional, IEEE, 2007,9(4):34-41.

[7] Redondo, R.P.D. and Vilas, A.F. and Arias, J.J.P., Educateca: A Web 2.0 Approach to e-Learning with SCORM, in IFIP Advances in Information and Communication Technology, 2010, vol. 341, pp. 118-126.

[8] Richardson L., Ruby S., RESTful Web Services, O'Reilly Media, 2007.

[9] Selim H.M., Critical success factors for e-learning acceptance: Confirmatory factor models. Computers & Education, Elsevier, 2007, 49(2), 396 – 413.

[10] Vossen, G. and Westerkamp, P., Why service-orientation could make e-learning standards obsolete, in International Journal of Technology Enhanced Learning, 2008, vol. 1, no. 1, pp. 85-97.

[11] Wilson B. G., Constructivist Learning Environments: Case Studies in Instructional Design. Educational Technology Publications, 1996.