



UNIVERSITÀ DEGLI STUDI  
DI TRENTO

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI  
CORSO DI LAUREA SPECIALISTICA IN INFORMATICA

## ELABORATO FINALE

UNA SOLUZIONE PER EROGARE VIDEOLEZIONI  
SU DISPOSITIVI MOBILI: LA PIATTAFORMA  
LODE4iPHONE E iTUNESU

RELATORE:  
PROF. MARCO RONCHETTI

LAUREANDO:  
ANTONIO MATTEI

ANNO ACCADEMICO 2009-2010



## **Indice generale**

Introduzione.....	5
1. Videolezioni.....	7
1.1 Introduzione.....	7
1.2 Un po' di storia.....	7
1.3 Caratteristiche delle videolezioni digitali.....	8
1.4 Videolezioni asincrone in dettaglio.....	11
2. Il sistema “Lode” .....	15
2.1 Introduzione.....	15
2.2 Prerequisiti e funzionamento in acquisizione.....	16
2.3 Fase di distribuzione.....	20
2.4 Considerazioni su Lode.....	22
3. iTunesU.....	23
3.1 Introduzione.....	23
3.2 iTunesU per l'Università di Trento.....	24
3.3 Funzionamento di iTunesU.....	26
3.4 Osservazioni - limiti.....	33
4. Sviluppo di un'estensione di Lode per iTunesU.....	35
4.1 Introduzione – obiettivi.....	35
4.2 Analisi dati e funzioni.....	36
4.3 Risultati.....	42
5. Apple iPhone.....	43
5.1 Caratteristiche di iPhone.....	43
5.2 Sviluppo di applicazioni per iPhone tramite l'SDK.....	44
5.3 Confronto con altre piattaforme smartphone.....	56
6. Sviluppo di Lode4iPhone.....	59
6.1 Introduzione – obiettivi.....	59

6.2 Analisi dei dati - lato server.....	60
6.3 Funzionamento di Lode4iPhone – lato client.....	63
6.4 Analisi dell'architettura di Lode4iPhone.....	72
6.4.1 Classi MVC (Model View Controller).....	73
6.4.2 Uso di strutture dati di tipo “Dizionario”.....	75
6.4.3 Classi ed oggetti per la connessione in rete .....	78
6.5 Difficoltà incontrate nello sviluppo del software.....	80
6.6 Osservazioni finali su Lode4iPhone.....	81
Conclusioni.....	83
Bibliografia.....	91

## Introduzione

L'utilizzo di risorse di tipo multimediale<sup>1</sup> in ambito didattico e formativo è una pratica in uso da diversi decenni. Tale utilizzo è stato più o meno rilevante nel tempo in relazione al progresso tecnologico e alla disponibilità di strumenti e supporti ad un costo sostenibile dalla maggioranza della comunità, perlomeno nei paesi più progrediti.

Non c'è dubbio che le tecnologie di tipo digitale hanno consentito in tal senso di compiere un passo avanti significativo, sia per quanto riguarda la produzione e l'utilizzo di queste risorse, sia per quanto riguarda la loro distribuzione. Se pensiamo per un istante alla differenza tra un'unità didattica disponibile su videocassetta vhs (che dev'essere acquistata e proiettata in un'apposita sala dotata di riproduttore e maxischermo per essere fruita da una classe) ed un moderno tutorial pubblicato su Youtube e fruibile da chiunque – individualmente o in gruppo - abbia una connessione veloce ad Internet, ci rendiamo conto di cosa abbia prodotto questo progresso.

Questo elaborato finale descrive un lavoro di sviluppo realizzato allo scopo di costruire un sistema completo per la fruizione di videolezioni asincrone (cioè fruibili in tempo differito rispetto al momento in cui si sono effettivamente svolte) su dispositivi mobili, in particolare su iPhone, il noto smartphone dell'azienda Apple inc. Si tratta, appunto, di un sistema completo, in quanto il prodotto include sia un software per visualizzare sullo smartphone una lezione (come vedremo, attraverso un'interfaccia utente avanzata che include audio, video e slides), sia un modulo che consente di pubblicare facilmente videolezioni su un server, appoggiandosi alla piattaforma iTunesU, un canale di pubblicazione di materiale didattico in forma multimediale curato da Apple ed utilizzato da molte realtà universitarie di tutto il mondo. Il risultato è un sistema integrato che tiene conto delle diverse fasi, dal

---

<sup>1</sup> Da Wikipedia: "La **multimedialità** è la compresenza e interazione di più mezzi di comunicazione in uno stesso supporto o contesto informativo. Si parla di *contenuti multimediali*, specie in ambito informatico, quando per comunicare un'informazione riguardo a qualcosa ci si avvale di molti *media*, cioè mezzi di comunicazione di massa, diversi: immagini in movimento (video), immagini statiche (fotografie), musica e testo.

momento della “produzione” della lezione a quello della sua diffusione e visione, anche su iPhone.

Il primo capitolo di questo elaborato finale è dedicato ad un'analisi generale dei sistemi di videolezione, individuandone le diverse tipologie, i punti a favore e le criticità. Il secondo capitolo invece approfondisce nel dettaglio il sistema Lode, una piattaforma il cui sviluppo è stato coordinato dal prof. Ronchetti dell'Università di Trento, e che permette di produrre videolezioni di tipo asincrono; Lode costituisce di fatto una delle basi per lo sviluppo del prodotto che sarà poi descritto negli ultimi capitoli. Il terzo capitolo, che conclude la descrizione del contesto generale e di progetto, è dedicato al sistema iTunesU: come sopra accennato, è un canale di pubblicazione di materiale didattico (audio, video, slides), e si tratta probabilmente dell'iniziativa più rilevante in assoluto in termini quantitativi in questo settore.

I capitoli seguenti sono dedicati invece a descrivere più in profondità il prodotto su cui è basato questo lavoro: nel quarto in particolare si affronta il modulo software sviluppato per espandere le funzionalità del sistema Lode in modo da realizzare un formato di output delle videolezioni prodotte da Lode adatto ai dispositivi mobili e, in particolare, ad Apple iPhone. Questo output, al contempo, è compatibile con il canale iTunesU dove quindi le lezioni possono essere direttamente pubblicate senza ulteriori conversioni. Il quinto capitolo introduce la piattaforma Apple iPhone, con particolare riferimento alla possibilità di sviluppo di applicazioni autoprodotte. Infine il sesto capitolo contiene la descrizione di Lode4iPhone, il prodotto software che costituisce l'asse portante di questo lavoro di tesi; come intuibile, si tratta di un “porting” dell'applicazione Lode (o meglio di quella parte di Lode che realizza le funzioni per visualizzare una videolezione asincrona) su iPhone. Al capitolo finale sono affidate le conclusioni.

# 1. Videolezioni

## 1.1 Introduzione

Il settore applicativo delle tecnologie multimediali, intese come trattamento ed erogazione di contenuti quali immagini, audio e video, ha vissuto in questi ultimi anni un forte sviluppo grazie alle tecnologie digitali e alla diffusione di connessioni di rete geografica a larga banda. Da un lato, la produzione e la fruizione di risorse audio/video a basso costo è diventato un elemento acquisito, grazie alla disponibilità in tal senso di molti dispositivi anche portatili; da un altro punto di vista, servizi di rete quali il noto Youtube hanno reso particolarmente agevole pubblicare e quindi dare risalto alle produzioni multimediali sia di tipo più “artigianale”, sia di tipo semiprofessionale.

In tale contesto, il campo della formazione è sicuramente uno di quelli che si può giovare delle possibilità offerte dai nuovi sistemi, pur osservando che il rapporto tra i processi formativi e le tecnologie ha presentato da sempre, oltre a diversi punti di contatto e convergenza, alcune criticità. Principalmente, queste criticità sono legate alla necessità di ripensare alle modalità complessive con cui avviene il processo di comunicazione tra docenti e discenti.

## 1.2 Un po' di storia

In realtà il rapporto tra le attività di formazione e aggiornamento (in ambito scolastico, universitario o di long-life learning) e la multimedialità risale ad alcuni decenni fa: anche prima della diffusione di massa del mezzo televisivo esistevano filmati prodotti esattamente allo scopo di diffondere le conoscenze tra la popolazione sia per gli aspetti di base (alfabetizzazione) che per contenuti più avanzati. L'avvento e la diffusione del mezzo televisivo su larga scala poi ha consolidato questa tendenza, attraverso la nascita di programmi di fascia “educational” che tutt'oggi continuano ad esistere. A titolo di esempio possiamo citare TeachersTV<sup>1</sup> (anche se in verità da settembre 2010 ha trasferito il servizio

---

<sup>1</sup> Vedi <http://www.teachers.tv>

su Internet, attraverso un sistema di streaming video) ed il consorzio Nettuno<sup>1</sup> che trasmette anche attraverso i canali della tv pubblica italiana.

Inoltre, l'avvento di supporti quali le videocassette VHS e, più tardi, i CD e DVD-VIDEO, hanno introdotto la possibilità di superare il vincolo dell'erogazione/fruizione sincrona dei contenuti (vincolo temporale) agevolando la nascita di un ruolo rinnovato per le videolezioni, in cui lo scopo non è (solo) la sostituzione delle tradizionali lezioni frontali in aula, bensì la disponibilità di uno strumento nuovo e più flessibile che può garantire differenti forme di supporto al processo di apprendimento.

Le tecnologie digitali hanno permesso poi la creazione di diversi formati di lezione con supporto multimediale (o videolezione), quali ad esempio gli screencasts (registrazione di una presentazione effettuata sul desktop del computer, con l'aggiunta di un commento audio) ed i podcast (audiolezioni con attivazione di "canali" a cui abbonarsi per semplificare la gestione e lo scaricamento dei files).

### **1.3 Caratteristiche delle videolezioni digitali**

Con l'avvento delle tecnologie digitali, molte iniziative si sono susseguite nel settore della formazione per mettere a punto piattaforme di supporto multimediale alle attività didattiche. Per un'esauriente analisi di tali iniziative si vedano anche gli articoli (1) e (3) della bibliografia.

In figura 1, un interessante schema tratto dall'articolo "The Internet-Based Lecture: Converging Teaching and Technology" di J.M.Pullen (2000) - vedi bibliografia (4)- mostra una classificazione di tali sistemi attraverso uno spazio bidimensionale i cui assi sono rappresentati dal livello di coinvolgimento (o-in altri termini: possibilità di partecipare in modo attivo) rispettivamente del docente e degli studenti. Si va quindi da un basso livello (chat testuale, slides) al livello massimo denominato "immersion" e presente allo stato attuale solo in applicazioni di realtà virtuale o simulata.

---

<sup>1</sup> Vedi <http://www.consorzionettuno.it>



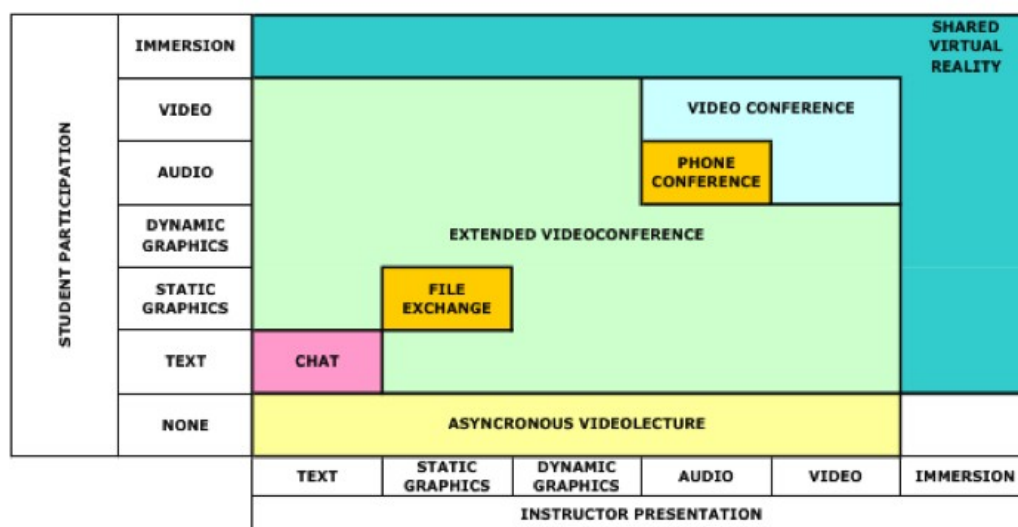


Figura 1: the tele-education space

Una prima separazione all'interno di questo "spazio" si può identificare tra attività (eventi o lezioni) sincrone e quelle asincrone. Si tratta di uno spartiacque abbastanza netto, in quanto la sincronizzazione è un requisito essenziale per avere interattività tra studente e docente nella forma più tradizionale, come avviene durante una normale lezione frontale dove la classe si trova fisicamente di fronte al docente; in questo tipo di contesti allargati, ad esempio durante una videoconferenza, gli studenti - ovunque si trovino - sono in grado di porre quesiti al docente e tutta la classe ascolterà sia le domande che le risposte in tempo reale. L'interazione peraltro si può avere oggi attraverso forme differenti, ad esempio con appositi forum pubblicati su piattaforme web, che permettono di attivare threads di discussione tra il docente e la classe, anche se in tal caso si torna nuovamente nel campo degli eventi asincroni. Più in generale, molti degli strumenti che fanno parte del cosiddetto web 2.0 (pensiamo a blogs, social networks come facebook ecc.) sono potenzialmente in grado di fornire piattaforme di comunicazioni asincrone alle comunità, anche se fino ad oggi queste forme di comunicazione non sembrano avere grande successo nel settore della formazione.

Può essere utile un breve confronto dei pro e contro delle soluzioni di videolezione sincrone e asincrone: nella tabella 1 si è cercato di sintetizzare le argomentazioni più importanti.

	<b>SINCRONE</b>	<b>ASINCRONE</b>
<b>PRO</b>	senso di maggior appartenenza alla comunità in senso tradizionale (la classe)	facilita la nascita di modi nuovi e più flessibili di usufruire dei supporti multimediali per l'apprendimento, togliendo il vincolo temporale oltre a quello spaziale
	i tempi dei moduli didattici sono scanditi in modo più ordinato, in quanto l'erogazione è simultanea per tutti i partecipanti.	hanno requisiti tecnici meno impegnativi per la realizzazione
		le unità didattiche prodotte possono essere di norma copiate su supporti di memoria rimovibili per essere utilizzate off-line (senza rete)
<b>CONTRO</b>	sono generalmente più costose ed impegnative da progettare, realizzare ed erogare efficacemente	favorisce il senso di alienazione ed isolamento degli studenti
	nelle aree geografiche dove non sono disponibili collegamenti in rete a larga banda non ne è possibile la realizzazione (laddove si trasmettano contenuti audio/video)	il docente ha meno controllo sui tempi con cui gli studenti accedono alle lezioni
	i sistemi di videoconferenza multi-points, utilizzati per realizzare questo genere di lezione, sono spesso poco intuitivi per l'utilizzatore	

Tabella 1: un confronto tra videolezioni sincrone e asincrone

Confrontando i diversi modelli di lezioni con supporto multimediale, implicitamente si fa riferimento ad una situazione in cui una lezione di tipo tradizionale viene estesa con l'ausilio di questi strumenti per raggiungere (in modo sincrono o asincrono) studenti che si trovano in un luogo fisico distinto da quello principale; in realtà è bene ricordare come ci siano esperienze in cui il docente tiene la lezione senza avere di fronte a sé alcuna platea, come avviene per esempio nell'ambito delle videolezioni trasmesse dal consorzio Nettuno, sebbene questa modalità evidenzia chiari limiti di efficacia, soprattutto per l'assenza totale di feedback verso il docente. Meno problematico è il caso in cui la forma scelta sia una videoconferenza in cui il docente sia solo ma in collegamento in tempo reale con la classe: in tale situazione l'interazione - seppur mediata dalle tecnologie - potrebbe garantire una miglior efficacia di erogazione.

Per completezza è bene osservare che un evento sincrono può - di norma - essere registrato in formato digitale, per essere nuovamente utilizzato in modalità asincrona.

#### **1.4 Videolezioni asincrone in dettaglio**

Possiamo dire che la maggioranza degli investimenti effettuata nel campo della formazione a distanza tramite tecnologie multimediali, nell'ultimo decennio, si sia concentrata nella fascia delle videolezioni asincrone (con riferimento alla figura 1, la fascia in basso sull'asse delle ascisse). Tra le iniziative più rilevanti in tal senso possiamo ricordare:

- Mitopencourseware del Massachusetts Institute of Technology (MIT) <http://ocw.mit.edu>
- iTunesU, una sezione dell'iTunes Store dedicata alla diffusione di materiale didattico da parte di molti Atenei da tutto il mondo <http://www.apple.com/itunesU> (vedi anche capitolo 3)
- webcast.Berkeley dell'Università di Berkeley <http://webcast.berkeley.edu/>
- Lynda software training online tutorials <http://www.lynda.com>

Le videolezioni asincrone sono interessanti anche perché contribuiscono alla nascita di diversi filoni di ricerca, sia per provare ad estendere la funzionalità e la completezza delle

piattaforme, sia per realizzare forme di estrazione automatica di informazioni dalle tracce audio/video o dalle immagini. Tra questi spunti per la ricerca troviamo ad esempio:

- annotazione: la possibilità da parte degli utenti di inserire note e commenti in punti precisi dei filmati
- text-to-speech: ricavare la trascrizione della traccia audio
- estrazione di parole chiave e di informazioni in forma semantica della lezione
- navigazione all'interno della lezione, consentendo all'utente di spostarsi in corrispondenza di momenti significativi (per argomento ecc.)
- costruzione di interfacce composite, che includano ad esempio video, slides e timeline interattiva (sistema LODE, vedi capitolo 2)

Alcune analisi sulle modalità di utilizzo delle videolezioni in modalità asincrona da parte di studenti di corsi universitari -vedi articoli (5) e (6) in bibliografia- hanno evidenziato che la tendenza sia quella di rivedere solo parti specifiche di una lezione, ad esempio per approfondire un passaggio o un argomento poco chiari. Anche sulla base di questa motivazione sono nati sistemi quali LODE<sup>1</sup> e Openeya<sup>2</sup> che rendono particolarmente agevole la navigazione non sequenziale all'interno di un filmato.

Alcuni altri studi sull'efficacia delle lezioni fruite attraverso strumenti multimediali in varie combinazioni (audio+slides, video+slides ecc.) hanno dimostrato che la seconda soluzione (video+slides) è la più efficace per varie ragioni:

- vedere il filmato che ritrae il docente aiuta a mantenere la concentrazione
- la visione contemporanea delle slides e del filmato riproduce la situazione a cui uno studente è abituato, sebbene si ipotizzi che possa esistere uno "split effect" che distoglie l'attenzione a causa della necessità di guardare 2 oggetti distinti (si veda in tal senso l'articolo (7) in bibliografia).

---

<sup>1</sup> <http://latemar.science.unitn.it/LODE>

<sup>2</sup> <http://www.openeya.org>

Più in generale, dovendo individuare le ragioni che spingono un'istituzione, un docente, o un'azienda ad investire nella produzione di videolezioni asincrone, possiamo dire che gli studenti - attraverso questo strumento - possono:

- vedere (o ri-vedere) una o più lezioni, che non hanno frequentato (per malattie, impegni, incompatibilità orarie, lavoro ecc.) o che hanno frequentato ma di cui hanno bisogno di capire meglio alcuni passaggi
- rivedere i contenuti, ad esempio quando si avvicinano le sessioni di esame
- richiamare i contenuti di un corso anche a distanza di tempo
- sperimentare modalità didattiche innovative: ad esempio i docenti possono somministrare preventivamente le videolezioni ed utilizzare poi le lezioni in presenza per discuterne i contenuti, seguendo così un approccio più costruttivista.

Tra i problemi che limitano per ora una diffusione ancor più pervasiva delle videolezioni in ambiente scolastico ed universitario possiamo citare:

- problematiche legate al copyright, soprattutto per quanto riguarda la diffusione di materiali di supporto alle lezioni (foto, testi, slides, filmati ecc.)
- disponibilità di liberatorie alla diffusione dei contenuti, sia da parte dei relatori che da parte degli studenti (nel caso in cui, ad esempio, la registrazione audio/video includa una domanda posta, appunto, da uno studente)
- resistenze ad essere videoregistrati da parte di alcuni docenti
- resistenze legate ad alcuni punti critici già evidenziati in questo capitolo: la tendenza all'isolamento dello studente che apprende attraverso una videolezione anziché in un ambiente sociale. Da questo punto di vista va ricordato, come già evidenziato, che l'utilizzo di queste risorse multimediali va inteso anche e soprattutto come strumento integrativo e non per forza sostitutivo delle forme tradizionali di apprendimento. E' interessante annotare, da questo punto di vista, che il MIT di Boston, ricevendo delle critiche per aver pubblicato molte videolezioni dei propri corsi da utilizzare in forma gratuita (invece di pagare 26.000\$ l'anno), ha risposto che il valore di iscriversi e frequentare il MIT sta soprattutto nell'esperienza

sociale e ambientale del lavoro all'interno delle classi e dei laboratori (riferimento (8) in bibliografia).

Nella sezione dedicata alla bibliografia è possibile trovare una raccolta di articoli – si tratta di quelli citati dal numero (10) al numero (32) compresi – che trattano in modo approfondito molti dei temi introdotti in questo capitolo: in particolare parlano di studi sull'efficacia delle videolezioni e dell'eLearning in generale, di alcuni settori di ricerca che prendono spunto dai sistemi di videolezione (video-annotazione, speech recognition ecc.) e di varie soluzioni operative per migliorare il funzionamento questi sistemi.

In conclusione, tutto quanto è stato detto in questo capitolo in merito all'importanza delle videolezioni asincrone in ambienti didattici, e a quanto siano importanti i dettagli su come l'utente viene messo in grado di fruire di queste risorse, contribuisce a descrivere la motivazione che ha portato allo sviluppo del sistema Lode4Iphone in connessione al sistema iTunesU, temi che saranno affrontati negli ultimi capitoli.

## 2. Il sistema "Lode"

### 2.1 Introduzione

Lode (Lecture On DEmand) è un software frutto del lavoro coordinato dal prof. Marco Ronchetti dell'Università di Trento che realizza un sistema di registrazione e distribuzione di videolezioni attraverso un'interfaccia multimediale avanzata (audio + video + slides).

Il suo sviluppo si è basato sull'esperienza di utilizzo del software ePresence<sup>1</sup>, un sistema sviluppato dall'Università di Toronto, inizialmente distribuito con licenze di tipo open source ed in seguito rilasciato invece attraverso canali e modalità di tipo commerciale (vedi articolo (9) in bibliografia).

Tra le caratteristiche di Lode possiamo osservare fin da subito:

- la semplicità d'uso. Tutte le operazioni si svolgono attraverso interfaccia grafica e con pochi e chiari passaggi
- l'orientamento all'utilizzo in un contesto di corsi universitari: anche se il programma può essere usato per registrare ogni tipo di evento, la struttura dati a corredo dei filmati prevede l'inserimento ed il collegamento di informazioni quali il nome del corso, il numero (in sequenza) e la data della lezione, il nome del docente ecc., rendendo appunto il sistema particolarmente adatto al contesto degli insegnamenti dei corsi di studio universitari (o di altre attività formative simili).
- la scelta di produrre videolezioni di tipo asincrono, facilmente navigabili dall'utente in modo non sequenziale. Implicitamente, il contesto più indicato per la registrazione di una lezione è quello di una lezione frontale che si svolge in modo tradizionale, cioè con un docente ed un gruppo di studenti presenti nello stesso tempo (momento) e nello stesso luogo.

---

<sup>1</sup> <http://epresence.tv/>

## **2.2 Prerequisiti e funzionamento in acquisizione**

Lode è scritto in linguaggio Java, e per funzionare ha bisogno che siano soddisfatti i seguenti prerequisiti.

Per la registrazione:

- un computer Apple con sistema operativo Mac OS X
- una videocamera digitale (preferibilmente con connessione DV)
- un microfono (preferibilmente un radiomicrofono analogico del tipo "lavalier")
- un operatore che si occupi degli aspetti tecnici della registrazione

Per la riproduzione:

- un computer dotato di uscita audio, con installato un browser in grado di eseguire oggetti di tipo Adobe Flash (compatibile con Windows/Mac OS/Linux)

Per la pubblicazione online (opzionale) delle lezioni:

- un qualsiasi server web

Le versioni di Lode precedenti a quella corrente in realtà funzionavano su computers windows e linux anche per la fase di registrazione. L'esperienza di utilizzo però, alla luce dei numerosi problemi di compatibilità di dispositivi e drivers, ha suggerito il passaggio ad un sistema Apple, dove il fatto che hardware e software siano più "monolitici" pone minori problemi di compatibilità e semplifica il funzionamento del sistema.

Per quanto riguarda la fase di visualizzazione di una lezione in Lode, si è fatto invece – rispetto alle precedenti versioni - in un certo senso il percorso inverso: a partire da filmati in formato quicktime (codec video utilizzato da Apple) si è passati poi a Flash, più moderno e più compatibile con i sistemi Linux.



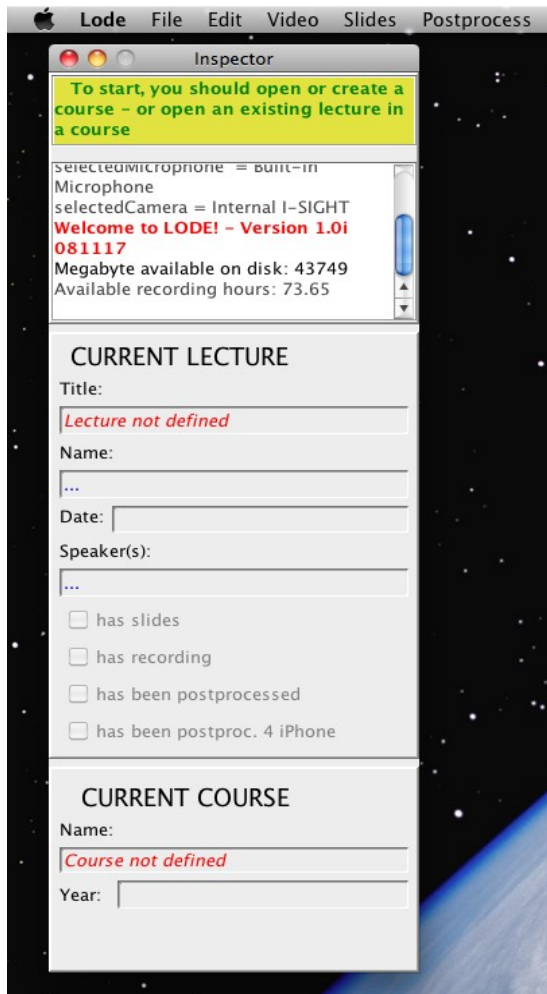


Figura 2: la schermata di avvio di LODE

In figura 2 possiamo vedere come si presenta Lode all'avvio: una schermata di benvenuto ci mostra alcune informazioni di stato, mentre in alto una serie di voci di menù (visibili in dettaglio in figura 3) consentono l'accesso alle funzioni del programma. Tra le informazioni di stato si trovano in particolare il nome del corso e del docente, la data ed il titolo della lezione - informazioni che appaiono dopo che la fase di startup e popolamento iniziale degli archivi risulterà effettuato.

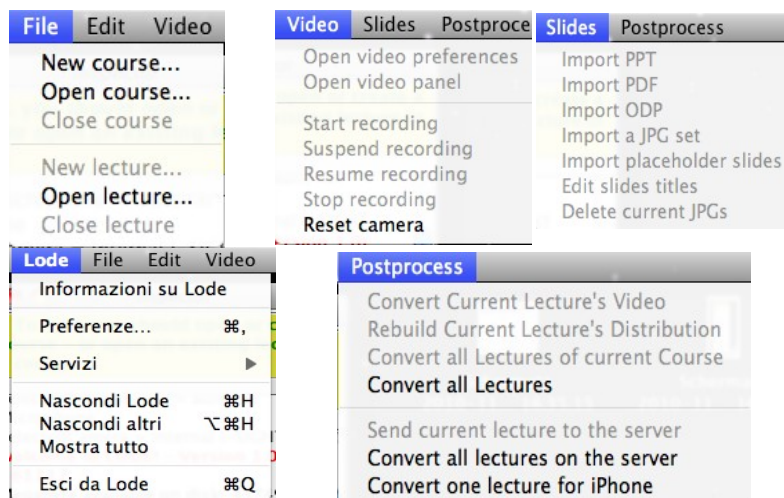


Figura 3: i menù di LODE

Tutti i comandi presenti nelle voci di menù, ad esclusione di quelli del gruppo "Postprocess", si riferiscono alla fase di registrazione della lezione, o comunque alla sua fase preparatoria.

Il comando "Preferenze" permette di scegliere i dispositivi audio e video che saranno utilizzati da Lode per acquisire i rispettivi media. La configurazione che assicura una buona qualità di registrazione è quella che prevede l'uso di un radiomicrofono (il cui trasmettitore è posizionato direttamente alla fonte – il relatore, mentre il ricevitore va connesso al computer) collegato all'ingresso audio analogico del microfono sul computer, e di una videocamera di tipo DV (con cavo firewire).

Con i comandi New – Open – Close Course e New – Open – Close Lecture, come è intuibile, si possono gestire i dati di catalogazione dei corsi e delle rispettive lezioni, in un'ottica orientata alla produzione di videolezioni per i corsi di tipo universitario, sebbene si possano utilizzare le stesse strutture dati per registrare anche eventi singoli quali seminari e conferenze.

In figura 4, a titolo di esempio, possiamo vedere come appare sullo schermo l'operazione "New lecture" attivata dopo aver aperto il corso di esempio "Web architectures 2008".

In figura 5 si può avere invece un'idea più dettagliata di quella che è sicuramente l'operazione più importante che si può svolgere all'interno del programma Lode: la registrazione di una lezione. Se la fase di configurazione dei dispositivi è stata completata correttamente (e se la videocamera e il microfono sono collegati) si può iniziare a

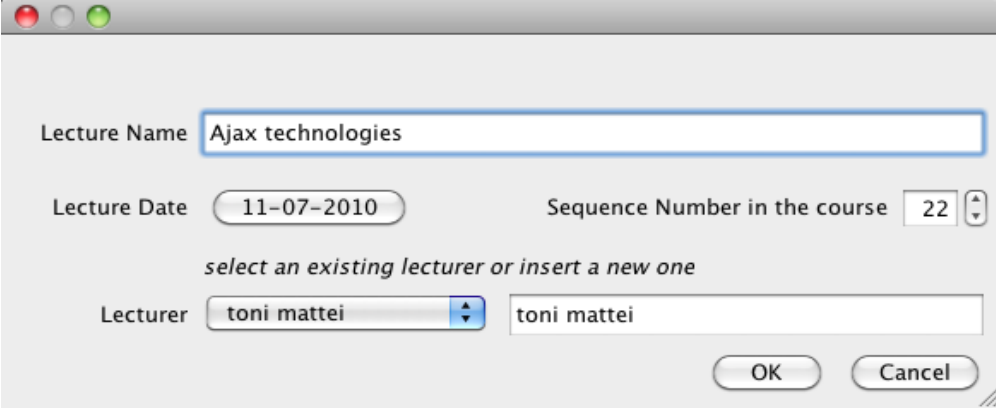
The image shows a graphical user interface window for adding a new lecture. It has a standard Mac OS X title bar with red, yellow, and green buttons. The window contains several input fields: a text box for 'Lecture Name' with the value 'Ajax technologies', a date picker for 'Lecture Date' showing '11-07-2010', and a spinner box for 'Sequence Number in the course' showing '22'. Below these is a label 'select an existing lecturer or insert a new one'. Under this label, there is a dropdown menu for 'Lecturer' showing 'toni mattei' and a text box to the right also containing 'toni mattei'. At the bottom right are 'OK' and 'Cancel' buttons.

Figura 4: inserire una nuova lezione in LODE

registrare. Nel riquadro in alto a sinistra c'è un'anteprima che permette di vedere in ogni momento che cosa sta inquadrando e registrando la videocamera. Come detto precedentemente, nella maggior parte dei casi Lode verrà utilizzato per registrare una lezione che si sta svolgendo in modalità tradizionale, frontale e con una classe fisicamente presente di fronte ed assieme al docente. In questa circostanza, molto spesso il relatore utilizzerà qualche supporto, tipicamente un file con le slides powerpoint o pdf. Lode consente all'operatore che sta registrando l'evento di aggiungere le slides alla sessione corrente (con il comando di menù Slides → import ppt (o import pdf, jpg ecc.). A questo punto, durante la lezione, quando il docente mostra una nuova slide agli studenti, nello stesso momento l'operatore di registrazione agisce in modo analogo cliccando sul pulsante "add" (con riferimento all'interfaccia di figura 5). Quest'azione consentirà in seguito all'utente, durante la riproduzione asincrona della lezione su computer, di rivivere l'esperienza completa dell'unità didattica visualizzando tutti gli elementi ed i supporti multimediali (slides, immagini, voce e volto del docente) in una modalità che si avvicina molto a quella di una lezione reale in presenza. In caso di errore nell'inserimento di un "cambio slide", l'operatore può modificare in qualsiasi momento la sequenza indicata durante la ripresa. Nella parte inferiore è visibile una serie di riquadri di anteprima che permette di vedere e sfogliare tutte le slides del file (o dei files) che è stato associato alla lezione corrente.

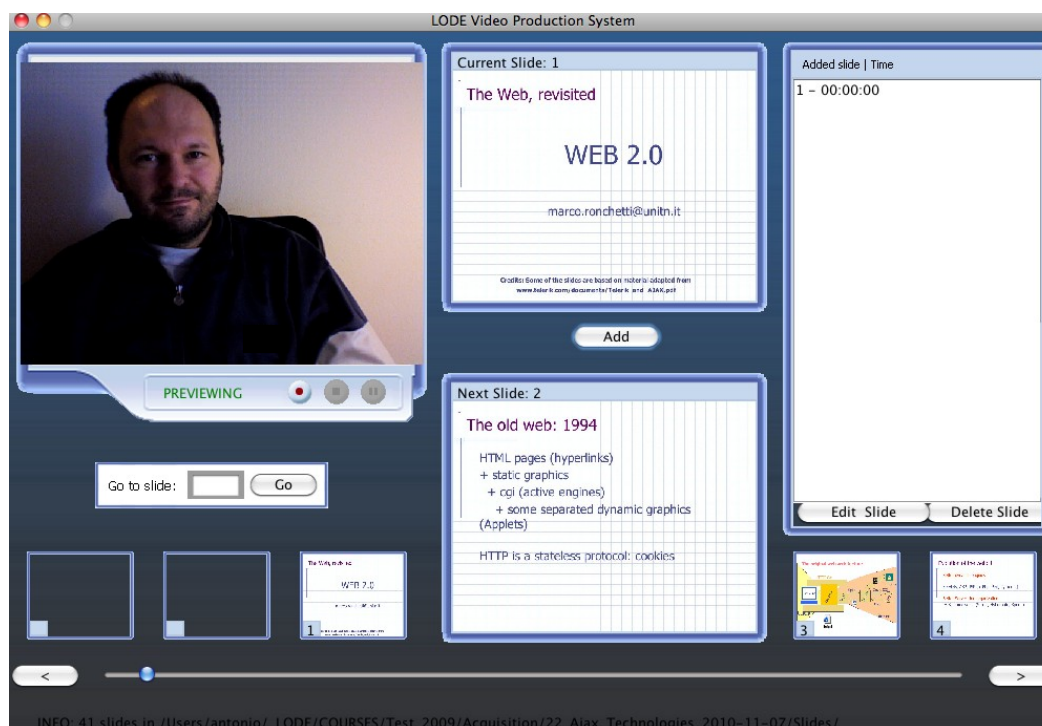


Figura 5: registrazione audio/video/slides di una lezione con LODE

Infine, sempre nell'interfaccia di registrazione, si possono notare i pulsanti che permettono di iniziare, terminare e porre in pausa la registrazione video.

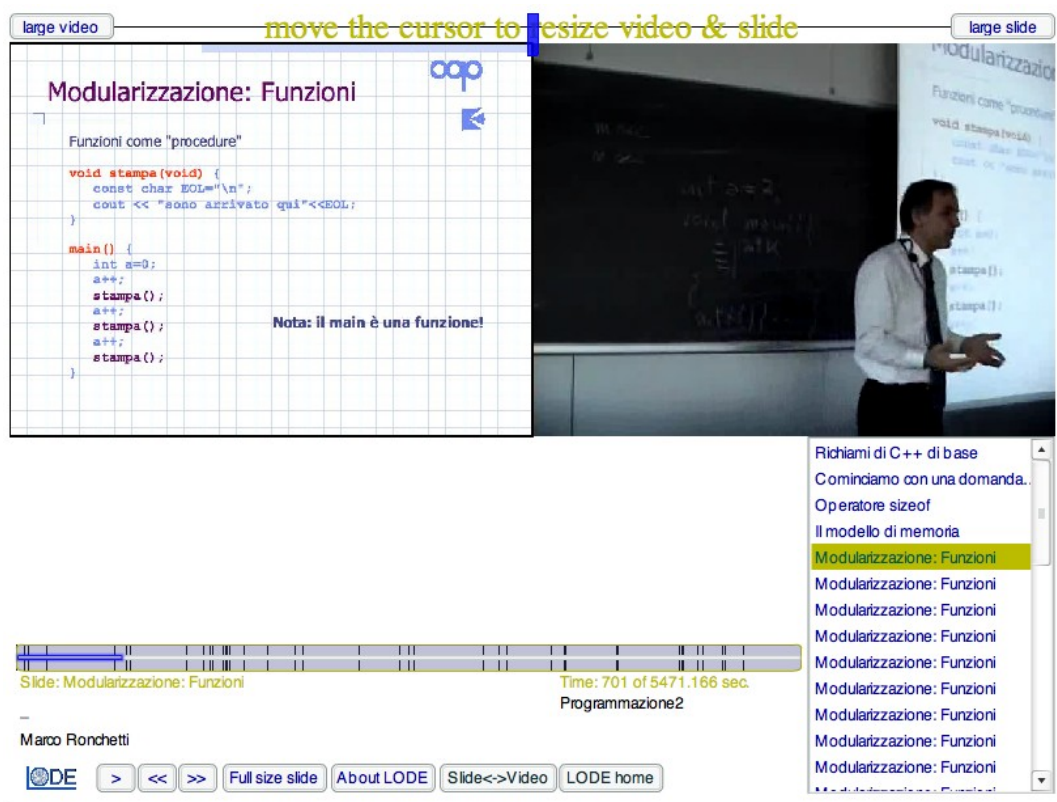
Quando la lezione termina, l'operatore deve semplicemente premere sul pulsante "stop" per concludere la registrazione e consolidare la cartella che contiene tutto il materiale acquisito (video, slides).

### 2.3 Fase di distribuzione

La fase successiva, che di norma si svolge in un momento distinto da quello della registrazione, è quella di postproduzione. L'operatore deve considerare che si tratta di un processo che richiede un certo tempo, che dipende dalla potenza di calcolo del computer su cui Lode viene eseguito, ma che in prima istanza si può stimare in un tempo pari a circa la metà della durata reale della lezione. Il comando da attivare per avviare la postproduzione si trova (come intuibile) nel menù "Postprocess", che contiene anche i comandi per ripetere il processo nel caso il risultato non sia quello desiderato, e per convertire in un solo passaggio tutte le lezioni del corso ancora non convertite.

La postproduzione effettua in background le seguenti operazioni:

- ricodifica e compressione del file contenente audio e video, da formato DV a FLV
- creazione dei files swf (flash), html, css, jpg e xml che permettono di realizzare l'interfaccia web per la visione della lezione tramite un browser web in modo asincrono, come appare in figura 6.



This video was produced with [Lode4Mac](#)

Figura 6: vedere una lezione con l'interfaccia di LODE

Il risultato della postproduzione è una cartella che contiene tutti i files necessari per la visione della lezione. Analizzando la figura 6, possiamo notare alcune caratteristiche dell'interfaccia utente:

- l'utente può scegliere di ingrandire il riquadro contenente le slides oppure quello con il video ripreso dalla videocamera (che generalmente inquadra il docente oppure altri elementi utili quali la lavagna)

- il tempo corrente della lezione (la "testina") può essere cambiato in ogni momento cliccando sulla timeline in basso, oppure scegliendo uno dei titoli delle slides dall'elenco presente nel riquadro di destra. Ovviamente, il filmato e le slides sono sincronizzati, quindi entrambi i riquadri si aggiornano ed avanzano automaticamente in modo coordinato.

#### **2.4 Considerazioni su Lode**

Il sistema Lode, che abbiamo descritto sinteticamente in questo capitolo, è stato usato con successo all'interno dell'Università di Trento, in più Facoltà ed in contesti diversi, sia di tipo seminariale che nell'ambito di corsi tradizionali all'interno della normale offerta formativa.

Il feed-back fornito dagli utenti (studenti in primis) mostra come il prodotto finale di Lode sia molto apprezzato, e viene considerato un supporto valido per la fase di apprendimento.

Una criticità del sistema Lode si può individuare nella necessità di fatto (anche se non in senso assoluto) della presenza di un operatore che controlli tutta la fase di registrazione, curando il setup iniziale della videocamera e del microfono, procurandosi una copia delle slides che il docente utilizzerà durante la lezione, ed agendo con gli appositi comandi per avviare, mettere in pausa e terminare la registrazione, nonché per indicare al sistema ogni "cambio slide" effettuato dal docente durante la sua presentazione reale di fronte agli studenti. Va ricordato infatti che il computer (Apple) che realizza la registrazione attraverso Lode non può essere lo stesso che il docente usa per la presentazione delle slides tramite videoproiettore o altro dispositivo di output per la lezione frontale. Si tratta però di un limite accettabile, in quanto questo ruolo (l'operatore) può essere esercitato anche da personale "non qualificato": ad esempio può essere svolto da uno degli studenti che frequenta la lezione in presenza, ed è ciò che effettivamente è avvenuto in molti dei casi nei quali Lode è stato usato. Tutto questo, in aggiunta ai requisiti hardware e software indicati all'inizio del capitolo, rendono Lode un sistema con costi iniziali e di esercizio particolarmente contenuti, aspetto non secondario in realtà ed in tempi come quelli odierni.

## 3. iTunesU

### 3.1 Introduzione

Il sistema iTunesU é un canale di distribuzione di materiale didattico, utilizzato principalmente dalle Università americane, che si trova all'interno del servizio "iTunes store" di proprietà della società Apple, un ambiente per la distribuzione, sia gratis che a pagamento, di risorse di tipo multimediale (musica, film, eBooks ecc.).

iTunesU é pensato per offrire una vetrina ad un numero crescente di Atenei in tutto il mondo (al momento della stesura di questo documento sono circa 370) verso un pubblico che comprende un numero elevatissimo di utenti, grazie anche alla popolarità acquisita nel tempo da iTunes store, avendo rappresentato il primo esempio significativo di vendita di musica digitale attraverso la rete Internet, con un costo accettabile per l'utente e con un sistema facile da usare grazie ai dispositivi player personali di tipo iPod (prodotti dalla stessa Apple) e grazie al software iTunes, disponibile su molte piattaforme hardware sia fisse (pc, mac) che mobili (iPod, iPhone, iPad) e che costituisce il modo di accesso (quasi obbligato per consultare tutti i contenuti di iTunes store e dunque anche di iTunesU.

L'importanza di quest'iniziativa é confermata dal "peso" degli Atenei che hanno investito in questo progetto, e vi pubblicano videolezioni con continuità ed in quantità rilevanti: MIT (Massachusetts Institute of Technologies) di Boston, Oxford University, Cambridge University, Harvard University, Stanford University e molti altri.

Tra le università italiane, al momento sono due le realtà presenti: si tratta della Federico II di Napoli e dell'Università di Trento.

Le modalità di iscrizione al canale iTunesU da parte di un ateneo non sono ne' veloci ne' scontate: si nota una certa severità nel valutare le credenziali degli interlocutori e la qualità del prodotto che si intende pubblicare. Questo garantisce un buon livello qualitativo di tutte le risorse presenti nel canale.

In sostanza, a partire dal sito web <http://www.apple.com/education/itunes-u/> si trovano alcune informazioni ed alcuni documenti introduttivi, e nella stessa pagina c'è un link "apply"



che permette di compilare alcuni form per richiedere l'iscrizione al servizio. La richiesta viene vagliata – di norma entro un paio di settimane - e dev'essere poi confermata via email da un responsabile amministrativo dell'Ateneo stesso. Il servizio è gratuito, non ci sono costi diretti per l'Ateneo, ma devono essere accettate tutte le condizioni di utilizzo. Un dettaglio di tipo tecnico assume particolare rilevanza: ad oggi tutto il materiale (files audio, video, pdf) può essere caricato sui server di Apple per le sole università statunitensi, mentre al di fuori di questi confini ogni ateneo dovrà pubblicare il materiale su un proprio server; dunque iTunesU in questo caso fornisce l'interfaccia per la distribuzione e “apre la vetrina” verso il mondo iTunes, agendo però attraverso appositi links per collegarsi ai server delle rispettive università.

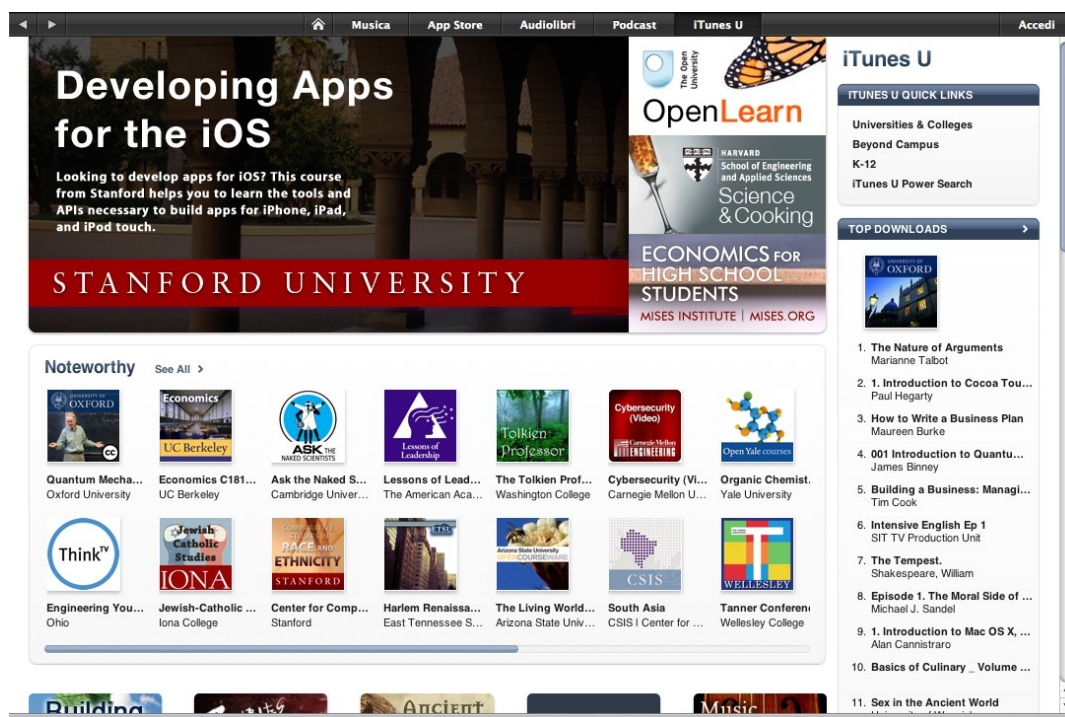


Figura 7: la home page di iTunesU, visualizzata su pc attraverso il programma iTunes

### 3.2 iTunesU per l'Università di Trento

L'Università di Trento ha attivato nel corso del 2010 il proprio canale iTunesU. L'iniziativa pone le sue basi su una concreta disponibilità materiale didattico sotto forma di risorse



audio/video, prodotto nel tempo sia attraverso il software Lode già descritto nel secondo capitolo, sia attraverso altre fonti. In tal senso può essere utile ricordare che



Figura 8: la pagina principale della sezione iTunesU per l'Università di Trento

sempre più spesso le sedi universitarie, in particolare le Facoltà, sono dotate di infrastrutture tecnologiche che facilitano di molto la registrazione di eventi: sia le sedi nuove che quelle ristrutturare negli ultimi anni, sono dotate di impianti audio video stabili (con videocamere, microfoni, videoproiettori ecc.)

Apple richiedeva, per il solo startup, la disponibilità di un minimo di contributi multimediali pubblicati pari a 150, obiettivo che è stato raggiunto in parte con alcuni corsi completi della Facoltà di Scienze, in parte con le registrazioni messe a disposizione dall'evento Festival dell'economia di Trento 2010, evento realizzato anche con il contributo dell'Università di Trento.

In seguito sono stati aggiunti altri filmati, anche da parte di altre Facoltà, e ad oggi il numero complessivo di videolezioni pubblicate per Trento è pari a 170.

Un apposito link sul portale di Ateneo <http://www.unitn.it/itunesu> permette di raggiungere immediatamente l'ambiente così predisposto (vedi figura 8).

### 3.3 Funzionamento di iTunesU

E' opportuno ora spiegare più dettagliatamente il funzionamento del sistema iTunesU in fase di pubblicazione dei contenuti.

All'atto dell'iscrizione Apple fornisce un sistema di accesso (un URL e un insieme di informazioni di sicurezza) per accedere al proprio spazio di pubblicazione in "modo amministrativo". Va segnalato fin da subito che lo strumento proposto per amministrare e gestire i contenuti è lo stesso software iTunes; questo rappresenta al contempo un limite (l'interfaccia è spesso lenta ed alcuni passaggi sono poco intuitivi) ed un vantaggio in termini di portabilità ed organizzazione del lavoro. Recenti contatti con Apple hanno messo in evidenza che a breve questo tipo di interfaccia sarà abbandonata a favore di una soluzione più flessibile.

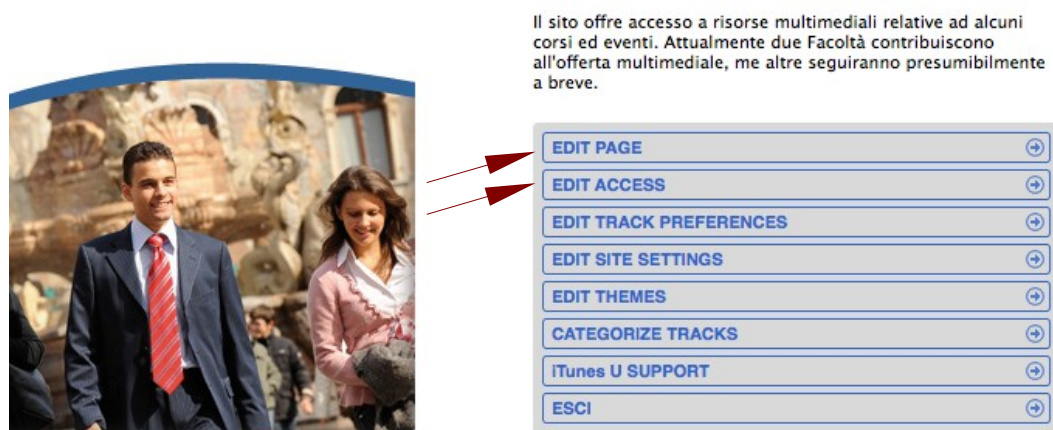


Figura 9: il menù di amministrazione di iTunesU

In figura 9 possiamo vedere la pagina principale dell'attuale ambiente amministrativo. Una serie di links consente di accedere ai comandi; "edit access" è importante perchè permette di impostare i diritti di accesso ad ogni singola risorsa multimediale pubblicata. Su questo argomento è utile soffermarsi in quanto va segnalata da un lato la possibilità di decidere che un contenuto sia pubblico e liberamente scaricabile da chiunque (oppure, per contro, che sia protetto in modo da essere accessibile solo a certe categorie di utenti) e dall'altro

lato la disponibilità di una serie di strumenti per gestire in modo efficiente ed efficace l'identità degli utenti e, appunto, i diritti di accesso.

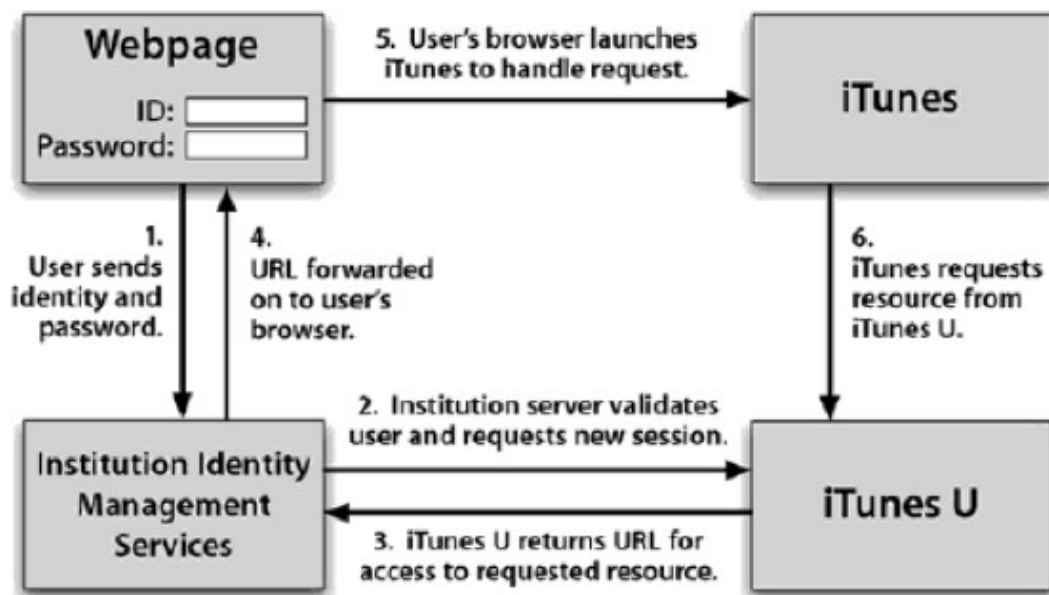


Figura 10: integrazione con sistema di identificazione esistente (tratto dal documento "iTunesU administration guide" di Apple)

In sostanza iTunesU permette di attivare una modalità di autenticazione che si appoggia ad un'infrastruttura già esistente presso le rispettive università; quest'eventualità è praticamente una certezza in quanto la necessità di erogare servizi online (servizi allo studio come Esse3, LMS (Learning Management Systems) quali Moodle, o accessi wifi solo per citare gli esempi più evidenti) fa capire che ogni struttura universitaria possiede già sistemi di identificazione e autenticazione dei propri utenti (docenti, personale di staff, studenti ecc.).

Non è tra gli scopi di questo elaborato finale approfondire nel dettaglio tutte le possibilità e le modalità operative per attivare questo canale di connessione iTunesU → autenticazione locale di ogni ateneo.

E' utile comunque focalizzare alcuni punti:

- tra i sistemi di autenticazione compatibili con iTunesU ci sono sia i sistemi web-based di qualsiasi tipo implementati presso ogni ateneo (es. Ldap, Active Directory,

con application servers ecc.) ma anche Shibboleth (che è lo standard emergente del settore), in quanto Apple fa parte della federazione di providers autorizzati. Citando la documentazione presente sul sito di Apple: "iTunes U runs a Shibboleth Service Provider (SP) within the InCommon Federation that works with any Identity Provider (IdP) in the Federation. If you run a Shibboleth IdP in the InCommon Federation, you can integrate with iTunes U and access iTunes U pages, features, and content."

- Apple fornisce una serie di esempi di codice, in vari linguaggi di programmazione, che permettono attraverso poche personalizzazioni di realizzare la connessione tra il proprio sistema di autenticazione web-based e iTunesU. La figura 10 contiene uno schema rappresentativo del processo di autenticazione realizzato in questa modalità congiunta
- la gestione di questi aspetti di autenticazione è importante perchè permette tra l'altro di far fronte ad alcuni dei problemi evidenziati quando si è parlato di videolezioni: la tutela dei diritti d'autore e l'assenza di regolari liberatorie di pubblicazione dei contenuti.

Un altro comando fondamentale dell'interfaccia di amministrazione di iTunesU è "Edit page". Come intuibile, si tratta dell'editing vero e proprio dei contenuti. I files multimediali sono organizzati in modo gerarchico all'interno di contenitori (o cartelle) ai quali può essere assegnato un nome coerente con il tipo di suddivisione che si decide di presentare all'utente. Nel caso di Trento, si è deciso ad esempio di dividere le risorse per Facoltà, e di riservare uno spazio a parte, in evidenza nella prima pagina, per gli eventi e le conferenze più importanti. Quest'organizzazione può essere modificata in qualsiasi momento, ed è opportuno che ciò avvenga nel momento in cui il numero dei files dovesse diventare molto elevato, e questo dovesse rendere poco agevole la consultazione di tutti i contenuti all'utente. Ad esempio, come vediamo in figura 11, la Facoltà di Scienze ha creato due sottosezioni: "didattica" e "seminari".

Come possiamo notare nella stessa figura 11 ed anche in figura 12, quando è stato selezionato il comando “Edit page” appaiono i pulsanti di controllo (ad esempio la classica “matita”) che segnalano la possibilità di editare il contenuto di aree specifiche.

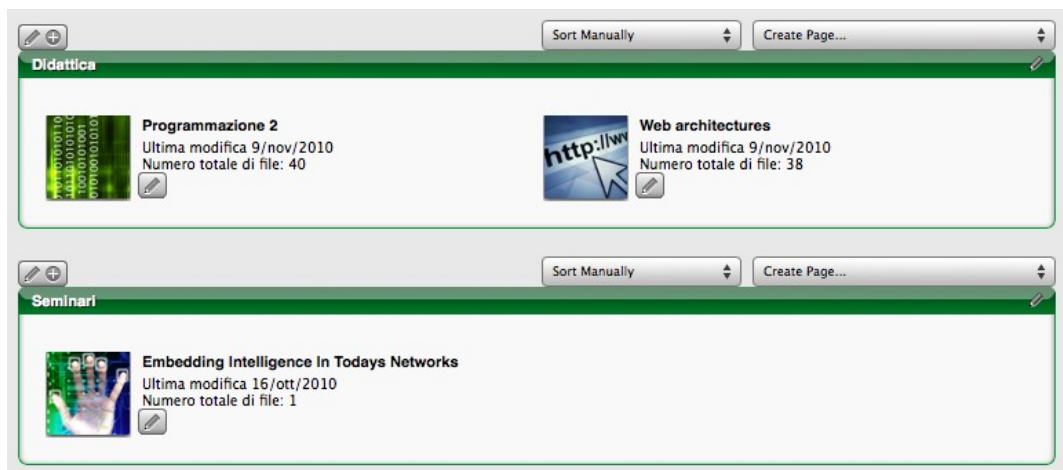


Figura 11: contenitori di files multimediali per la Facoltà di Scienze su iTunesU

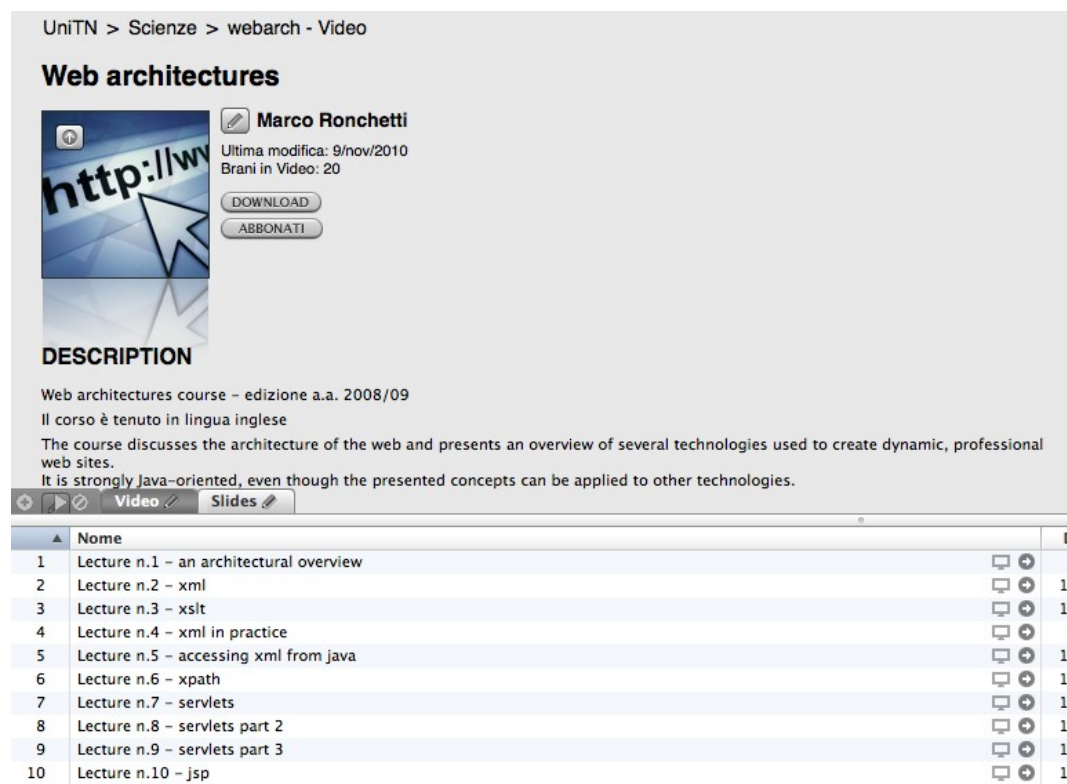


Figura 12: Editing pagina corso "Web architectures" su iTunesU

Come possiamo vedere, ogni corso ed ogni sezione sono caratterizzati tra l'altro da un'immagine rappresentativa che dev'essere anch'essa caricata in fase di pubblicazione o modifica dei contenuti.

L'ultimo passo da affrontare per editare i contenuti è l'inserimento dei riferimenti alle risorse audio/video; come detto in precedenza, si tratta di riferimenti in quanto fisicamente i files rimangono residenti sui server delle rispettive università. Per indicare ad iTunesU tutte le informazioni di ciascun corso e soprattutto il percorso dei files multimediali, è necessario inserire un url valido che punti ad un feed rss: si tratta in pratica di un file xml che deve osservare le regole indicate dall'apposita grammatica. Nella sezione seguente troviamo un estratto di alcune porzioni significative di questo file (di nuovo, non è interessante qui un'analisi nel dettaglio): si tratta di tags sufficientemente autocomentanti. In figura 13 possiamo invece visualizzare la finestra di inserimento dell'url contenente questo feed.

```
<?xml version="1.0" encoding="UTF-8"?><rss version="2.0"
xmlns:content="http://purl.org/rss/1.0/modules/content/" xmlns:wfw="http://wellformedweb.org/CommentAPI/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"
xmlns:itunesu="http://www.itunesu.com/feed">

<channel>
  <title>Web architectures</title>
  <atom:link
href="http://latemar.science.unitn.it/itunes/feeds/webarch/indexwebarch.html" rel="self"
type="application/rss+xml" />
  .....

<itunes:author>Marco Ronchetti</itunes:author>
  <itunes:summary>A course on Web architectures, held in a.y. 2008-09 for the Master in
Computer Science.</itunes:summary>
<description>A course on Web architectures, held in a.y. 2008-09 for the Master in Computer
Science.</description>
.....

<item>
<title>Web Architectures - an architectural overview - video</title>
<itunesu:category itunesu:code="101102" />
<itunes:order>1</itunes:order>
<dc:creator>Marco Ronchetti</dc:creator>
<itunes:author>Marco Ronchetti</itunes:author>
<enclosure
url="http://latemar.science.unitn.it/itunes/archive/video/webarch/lecture1_15-09-
2008.mp4" length="325790767" type="video/mp4" />
<guid>http://latemar.science.unitn.it/itunes/archive/video/webarch/lecture1_15-09-
2008.mp4</guid>
<itunes:subtitle>Web Architectures - an architectural overview</itunes:subtitle>
<itunes:summary>Lecture n.1 - An overview of the main concepts of the Web</itunes:summary>
<itunes:keywords>informatica, programmazione, università, trento, web</itunes:keywords>
<itunes:explicit>no</itunes:explicit>
<itunes:duration>58:33</itunes:duration>
<pubDate>Mon, 15 Sep 2008 14:04:55 +0000</pubDate>
</item>
```

Notiamo che esiste, dopo i primi tag iniziali con le informazioni generali, un tag <item> che identifica la lezione; quindi avremo per ogni corso N tags <item> - uno per ogni lezione – contenenti al loro interno, tra le altre informazioni, anche un tag <enclosure> che all'attributo “url” indica la posizione del file con il filmato della lezione.

Per quanto riguarda i formati dei files multimediali ammessi alla pubblicazione sul sistema iTunesU, le indicazioni sono le seguenti:

AUDIO: Apple raccomanda l'utilizzo di files codificati in formato AAC (Advanced Audio Coding), anche se fornisce il supporto per il formato MP3 (meno efficiente a parità di bitrate)

VIDEO: il formato preferito di codifica è l'H.264, mentre è ammesso anche l'mpeg-4

SLIDES: al momento si possono pubblicare solo files PDF.

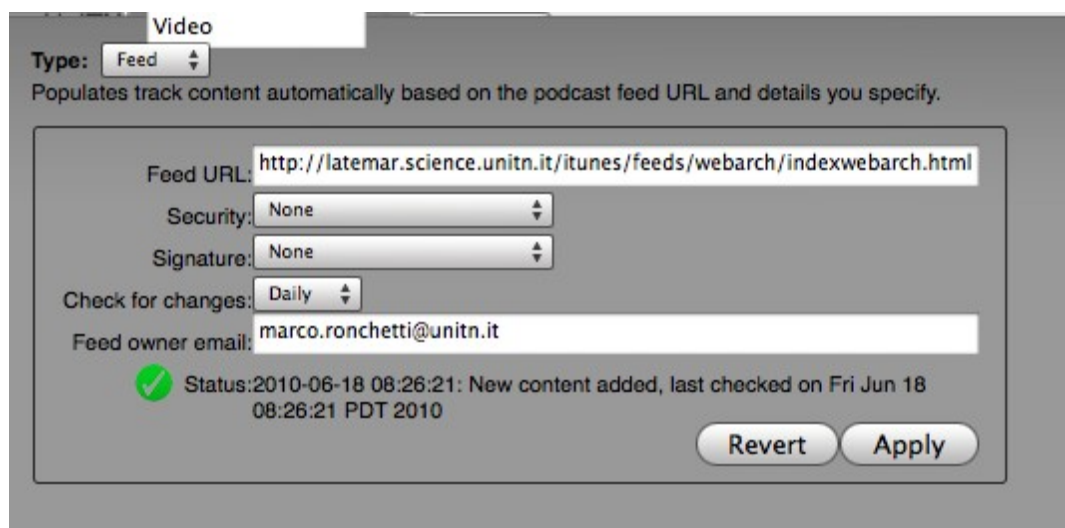


Figura 13: Editing del feed RSS associato al corso di Web architectures

Un ultimo aspetto molto importante è quello della disponibilità di un'interfaccia di comunicazione con iTunesU tramite web services. In pratica, è possibile compiere tutte le operazioni di gestione dei contenuti, che di norma si svolgono come detto attraverso il software iTunes, utilizzando invece software scritto autonomamente, in un linguaggio a propria scelta (deve solo supportare l'utilizzo di connessioni di rete, in quanto l'interazione con i web services avviene tramite xml, che è indipendente dalla piattaforma utilizzata).

Seguendo questo approccio, dovremo produrre del codice che effettua chiamate a questi web services, rispettando tutte le regole di sicurezza per l'autorizzazione all'accesso al server iTunesU (fornendo cioè le credenziali di accesso che garantiscano l'identità del mittente), ed aggiungendo tutti i parametri richiesti dall'operazione desiderata (es. "Aggiungi un corso", "elimina sezione" ecc.). In figura 14 possiamo vedere la struttura base dell'xml utilizzato per comunicare con questi web services. Il grosso vantaggio di usare questo strumento è quello di poter integrare sistemi software esistenti con la piattaforma iTunesU, per realizzare forme di automazione nella pubblicazione e gestione dei contenuti multimediali.

```
<?xml version="1.0" encoding="UTF-8"?>
<ITunesUDocument>
  <Version>1.1.4</Version>
  <Operation>
    <entity>...</entity>
  </Operation>
  <Operation>
    <entity>...</entity>
  </Operation>
  <Operation>
    <entity>...</entity>
  </Operation>
</ITunesUDocument>
```

*Figura 14: schema base del documento di descrizione di un web service iTunesU*

Dal punto di vista dell'utilizzatore, il funzionamento di iTunesU è piuttosto semplice: aprendo il programma iTunes si può scegliere di navigare tra tutte le videolezioni (o podcasts) presenti nel canale, oppure di selezionare dall'elenco uno specifico Ateneo, oppure ancora utilizzare il potente motore di ricerca che ci permette, impostando alcune parole chiave, di individuare le risorse multimediali più adatte alle nostre esigenze. In relazione a quest'ultimo punto, va sottolineato che in fase di pubblicazione dei contenuti gli autori devono porre particolare attenzione alla corretta compilazione del contenuto dei tags di descrizione delle lezioni (all'interno del feed rss introdotto a pagina 30); tale attenzione



consentirà al motore di ricerca di individuare e catalogare correttamente le risorse pubblicate.

In alternativa, allo scopo di semplificare l'accesso alle videolezioni, è possibile utilizzare un link diretto alle risorse di uno specifico Ateneo; nel caso di Trento, visitando con un web browser l'url <http://www.unitn.it/itunesu> si aprirà automaticamente la finestra di iTunes dove verrà caricata la pagina principale delle risorse multimediali iTunesU dell'Università di Trento (come appare in figura 8). Gli utenti Linux (o comunque tutti gli utenti che non desiderano usare iTunes) possono aprire l'url:

<http://latemar.science.unitn.it:8080/iTunesU/index.groovy> che permette di accedere alle stesse risorse ma direttamente attraverso un'interfaccia web.

### **3.4 Osservazioni - limiti**

Il sistema iTunesU, così come è stato descritto, è interessante perchè non presenta particolari difficoltà di utilizzo, è sufficientemente aperto e flessibile per un uso in vari contesti, e soprattutto garantisce grande visibilità ai contenuti pubblicati grazie alla vasta platea di iTunes store.

Esistono però anche alcuni aspetti di criticità. C'è ad esempio un aspetto puramente tecnico: siccome i files audio/video pubblicati su iTunesU risiedono in realtà sui server dei rispettivi atenei di provenienza, si pone un problema di dimensionamento del carico (soprattutto di rete).

Inoltre, se torniamo a quanto scritto nel corso del secondo capitolo, dove abbiamo evidenziato i vantaggi del sistema Lode, dobbiamo osservare che le videolezioni asincrone erogate (e fruite) con iTunes rappresentano in tal senso un passo indietro. Infatti:

- non è disponibile il riquadro contenente le slides utilizzate dal relatore, che in certi tipo di lezione sono fondamentali per una buona comprensione dei contenuti
- la navigazione all'interno della lezione, cioè lo spostamento del punto di ascolto e visione in modo non sequenziale, è sì possibile, però in modo non ordinato: ricordiamo che in Lode tale spostamento può essere effettuato ricercando sulla timeline o nell'elenco dei titoli delle slides un punto ben identificato (un argomento)

- il software iTunes, necessario per l'accesso e lo scaricamento da parte degli utenti ai contenuti pubblicati, è disponibile per i sistemi windows e mac os, ma non ad esempio in ambiente Linux, mentre le lezioni distribuite con Lode funzionano su qualsiasi sistema dotato di un browser web e di un plugin per oggetti Flash. Quest'osservazione deve tenere conto del fatto che in realtà esistono sistemi per “aggirare l'ostacolo”, come abbiamo visto nel precedente paragrafo, ma si tratta di sistemi con limitazioni e non disponibili senza apposite installazioni e configurazioni ad hoc.

I limiti sopra evidenziati di iTunesU hanno suggerito, oltre alla motivazione principale di ottenere un porting di Lode su dispositivi mobili, lo sviluppo di un modulo software da integrare in Lode, di cui si parlerà nel prossimo capitolo.

## 4. Sviluppo di un'estensione di Lode per iTunesU

### 4.1 Introduzione – obiettivi

Tra i temi che sono stati introdotti nei capitoli precedenti, è utile e funzionale all'argomento di questo capitolo richiamare:

- il sistema Lode, e le videolezioni asincrone prodotte attraverso questo strumento
- iTunesU, il canale di Apple per pubblicare online lezioni in formato multimediale (audio o video)

Il lavoro di sviluppo del software descritto in questo capitolo è un'estensione di Lode ed un elemento di raccordo tra Lode e iTunesU, e permette di valorizzare il patrimonio di videolezioni prodotte dal primo grazie alla visibilità offerta dal secondo.

I requisiti di questo “plugin” software erano da un lato il tentativo di riprodurre seppur in minima parte l'interfaccia utente di una lezione registrata con Lode, e dall'altro il rispetto dei requisiti tecnici necessari per pubblicare files su iTunesU. Per quanto riguarda il primo di questi due requisiti, si è proceduto operando l'inserimento nel filmato, per alcuni secondi, di un'immagine jpg al posto del filmato, immagine contenente la slide proiettata dal docente durante la lezione originale, e così ripetutamente ad ogni “cambio slide”. In questo modo l'utente, scaricando le videolezioni da iTunesU, potrà visualizzare almeno le slides oltre al filmato, pur non potendo navigare in modo flessibile all'interno dei contenuti di una lezione, con strumenti quali la timeline interattiva e l'elenco dei titoli delle slides presenti in Lode.

La “fusione” tra il file video ricodificato e le immagini delle slides è possibile in modo ordinato e preciso grazie ad alcuni files già presenti nella cartella prodotta da Lode in fase di registrazione della lezione: in particolare si tratta:

- del file `timed_slides.xml` che contiene  $n$  elementi, uno per ogni “cambio slide”, con l'indicazione del tempo (numero di secondi trascorsi dall'inizio della registrazione) e del numero della slide proiettata
- delle immagini jpeg contenenti le slides.

Una doverosa precisazione riguarda l'espressione "conversione per iPhone" usata più avanti in questo capitolo: anche se il device iPhone è in effetti il principale destinatario di tutto questo elaborato finale, in questo capitolo sarebbe meglio parlare più in generale di "conversione per iTunesU", perchè le videolezioni convertite con il modulo software qui descritto possono essere in effetti fruite da qualsiasi computer che possa eseguire iTunes, dunque non solo dall'iPhone. Le due espressioni quindi in questo contesto sono da considerarsi (imprecisamente) equivalenti.

Il modulo software è stato chiamato "estensione" o "plugin" di Lode in quanto è stato effettivamente incluso in una nuova versione di Lode.

## **4.2 Analisi dati e funzioni**

Per realizzare il modulo software aggiuntivo sopra introdotto, sono state estese alcune classi già esistenti nell'architettura generale di Lode. Nel dettaglio questi oggetti sono:

classe Lecture: alla classe che rappresenta l'entità "Lezione" è stata aggiunta la proprietà "has\_iphone\_post\_processing" che memorizza l'informazione dell'avvenuta (o meno) conversione per iPhone (cioè per iTunesU) della lezione corrente;

classe LodeConstants: sono state aggiunte alcune costanti utilizzate nella classe MacPostProducer: una di queste indica il percorso della cartella dove Lode memorizzerà i files del postprocessing per iPhone/iTunesU, mentre le altre si riferiscono a parametri utilizzati per la conversione del video (bitrate, framerate e durata in secondi delle slides inserite nel filmato)

classe MenuManager: qui è stato integrato l'elenco delle voci di menù esistenti aggiungendo "Convert one lecture for iPhone" per invocare il comando di conversione (attraverso un'istanza della classe ActionController) vedi anche figura 15

classe ActionController: in conseguenza al punto precedente, in questa classe è stato aggiunto il codice necessario per rispondere all'azione dell'utente quando seleziona la nuova voce di menù.

classe LecturePane: l'intervento su questa classe, che si occupa di visualizzare un pannello informativo nella schermata principale, è semplicemente l'aggiunta del flag che indica se la lezione corrente è stata o meno convertita per iPhone

classe MacPostProducer: in questa classe si è concentrata la quasi totalità del lavoro di sviluppo svolto per questo modulo software. È stato aggiunto un nuovo metodo "convertVideo4Iphone" che accetta come parametro in input un puntatore ad un'istanza di "Lecture", e realizza la conversione secondo le modalità che di seguito verranno spiegate nel dettaglio.

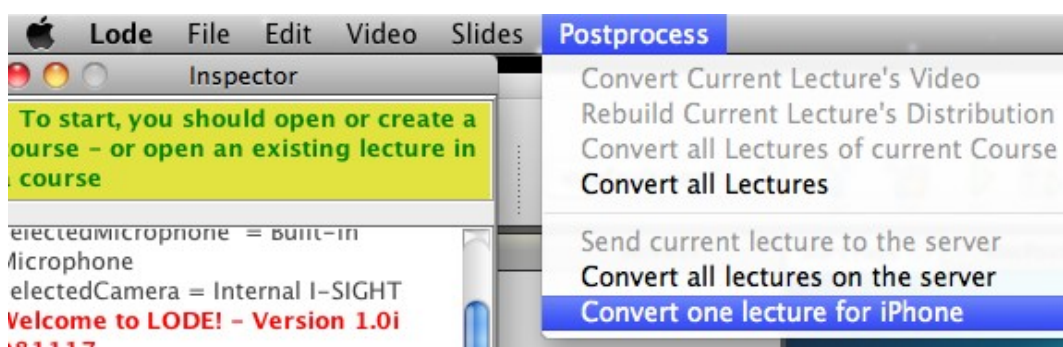


Figura 15: Il comando di conversione per iPhone/iTunesU aggiunto nel menù

La figura 16 riporta un diagramma uml, e precisamente uno use-case diagram, che rappresenta graficamente il caso d'uso implementato in questo modulo software. Nel seguito del paragrafo alcuni activity-diagrams (figure 17, 18 e 19) approfondiranno la definizione degli aspetti funzionali.

A breve commento di questi diagrammi, è opportuno precisare che:

- come primo passaggio, all'utente viene chiesto di scegliere, attraverso un'apposita finestra di dialogo, quale videolezione convertire tra quelle presenti nell'archivio di Lode;
- l'utente può scegliere di riconvertire la lezione nel caso in cui l'applicazione rilevi, tramite l'apposito flag, che tale processo sia già stato fatto in precedenza;
- la conversione viene prima realizzata "a tavolino", nel senso che tutti i comandi necessari vengono inseriti in un array di stringhe e poi lanciati sequenzialmente solo alla fine del processo. La ragione di questa scelta è che la sintassi di questi

comandi dipende anche da parametri (come ad esempio il tempo dei “cambi slide”) che vengono ricavati scorrendo altri array con appositi cicli, quindi separare il lavoro in due fasi è parsa la scelta migliore: prima l'individuazione dei parametri e della sintassi completa dei comandi e poi l'esecuzione dei comandi stessi;

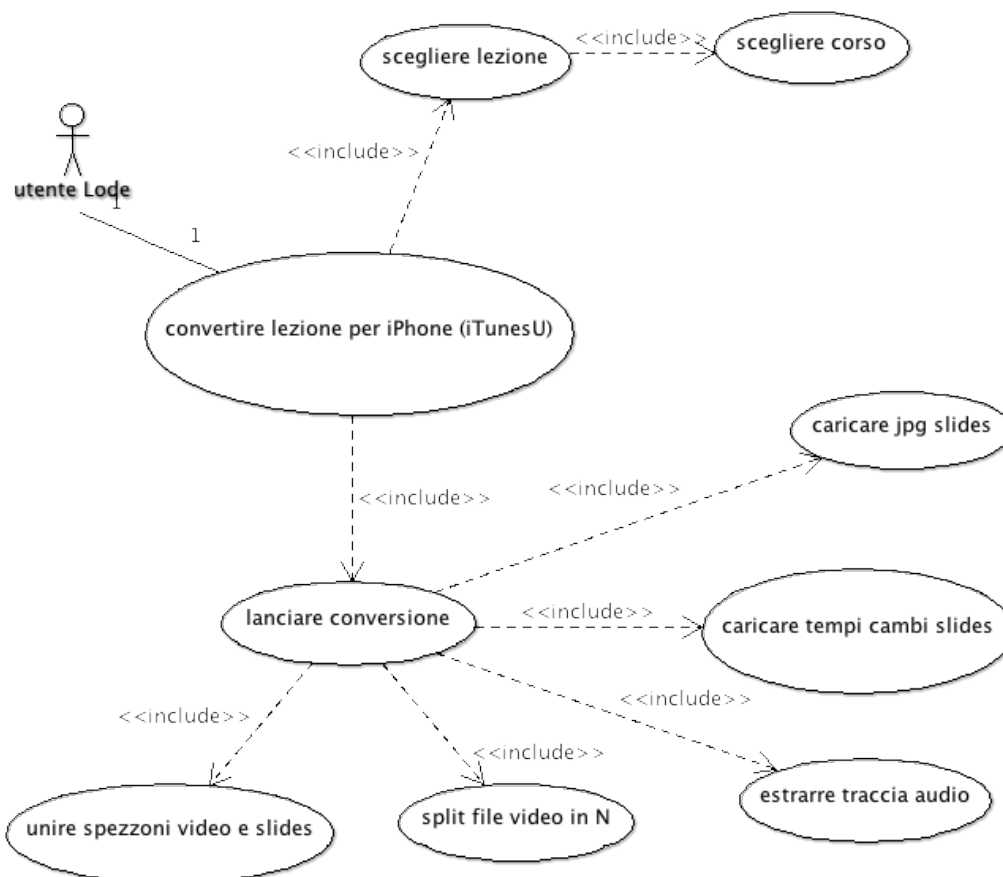


Figura 16: use case diagram del modulo sviluppato

- la separazione della traccia audio del video originale, effettuata come primo passaggio della conversione, è dovuta al fatto che, mentre la traccia video deve essere profondamente modificata (ricodificata, divisa e unita alle immagini delle slides) la componente audio con la voce del docente deve rimanere completamente invariata, dunque escludendola dai processi di rielaborazione del video si alleggerisce l'intera procedura e si preserva la qualità della fonte sonora;
- la “fusione” di video e slides della lezione originale avviene dividendo in  $n+1$  parti il filmato, dove  $n$  è (circa) il numero di “cambi slide” effettuato dal docente ; il “circa” è

dovuto al fatto che ci può essere una variazione di +/-1 a seconda che all'inizio del filmato ci fosse o meno una slide proiettata e che l'ultimo cambio slide sia avvenuto o meno a ridosso del termine del filmato stesso. Tra le sezioni di filmato così divise vengono inserite le immagini jpg delle slides, per una durata di X secondi (con X inserito come costante nella classe LodeConstants, per default 5 secondi), e in corrispondenza X secondi di filmato originale vengono eliminati

- per compiere la maggioranza delle operazioni sulle tracce audio/video si è utilizzato il software ffmpeg, un'utility open source disponibile anche per computers Apple, che gestisce vari formati audio/video/immagini e consente anche operazioni quali lo "splitting" dei filmati; in particolare, i "templates" dei comandi usati rispettivamente per generare lo spezzone di filmato con la slide e per estrarre una parte di filmato originale da unire poi con quello della slide, sono i seguenti:  
 1) `ffmpeg -f mjpeg -itsoffset time-slide -i jpegfile_slide -s 640x480 -an -r framerate -vframes 100 -y OUTPUT_FILE`  
 2) `ffmpeg -i videofile -an -sameq -r framerate -y -t duration -ss offset OUTPUT_FILE`

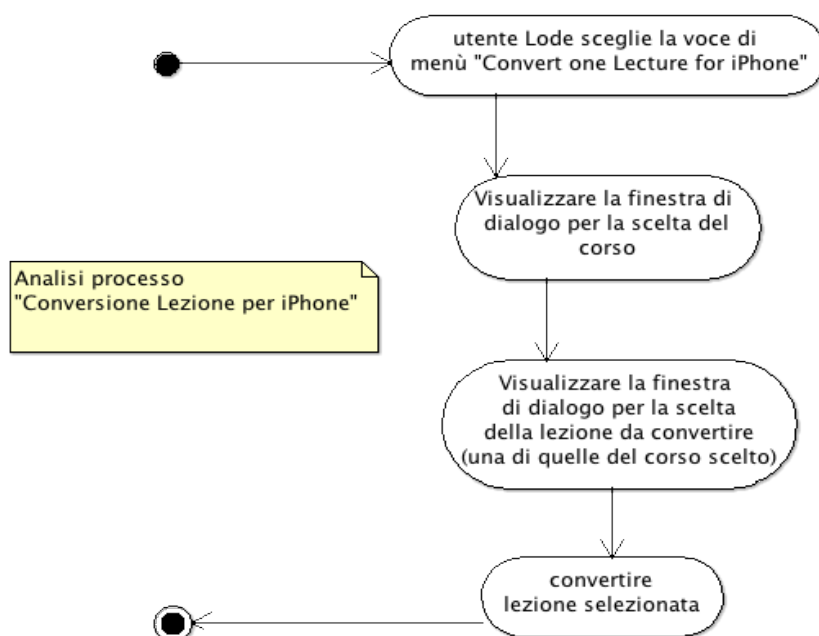


Figura 17: activity diagram del processo di conversione (1/3)

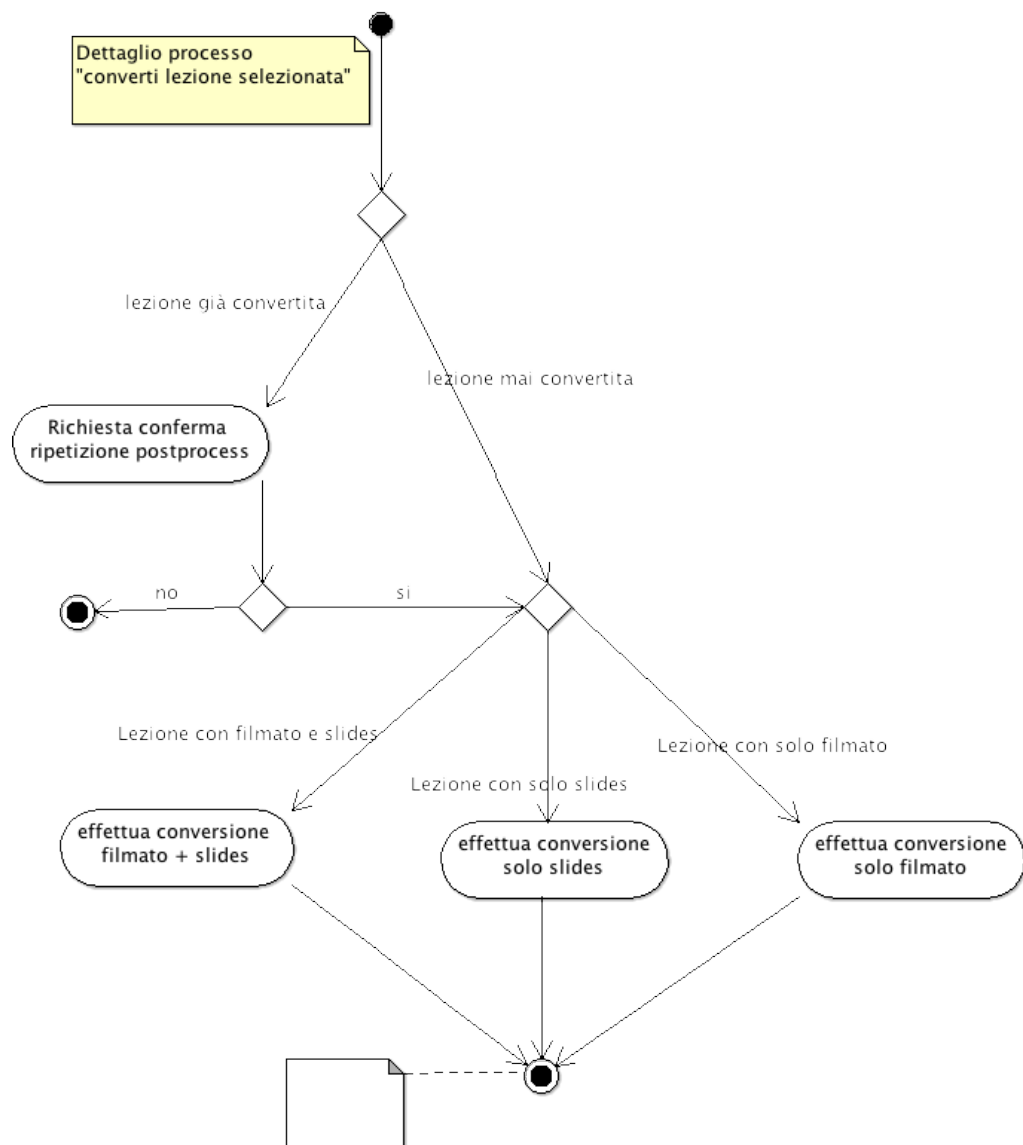


Figura 18: activity diagram del processo di conversione (2/3)



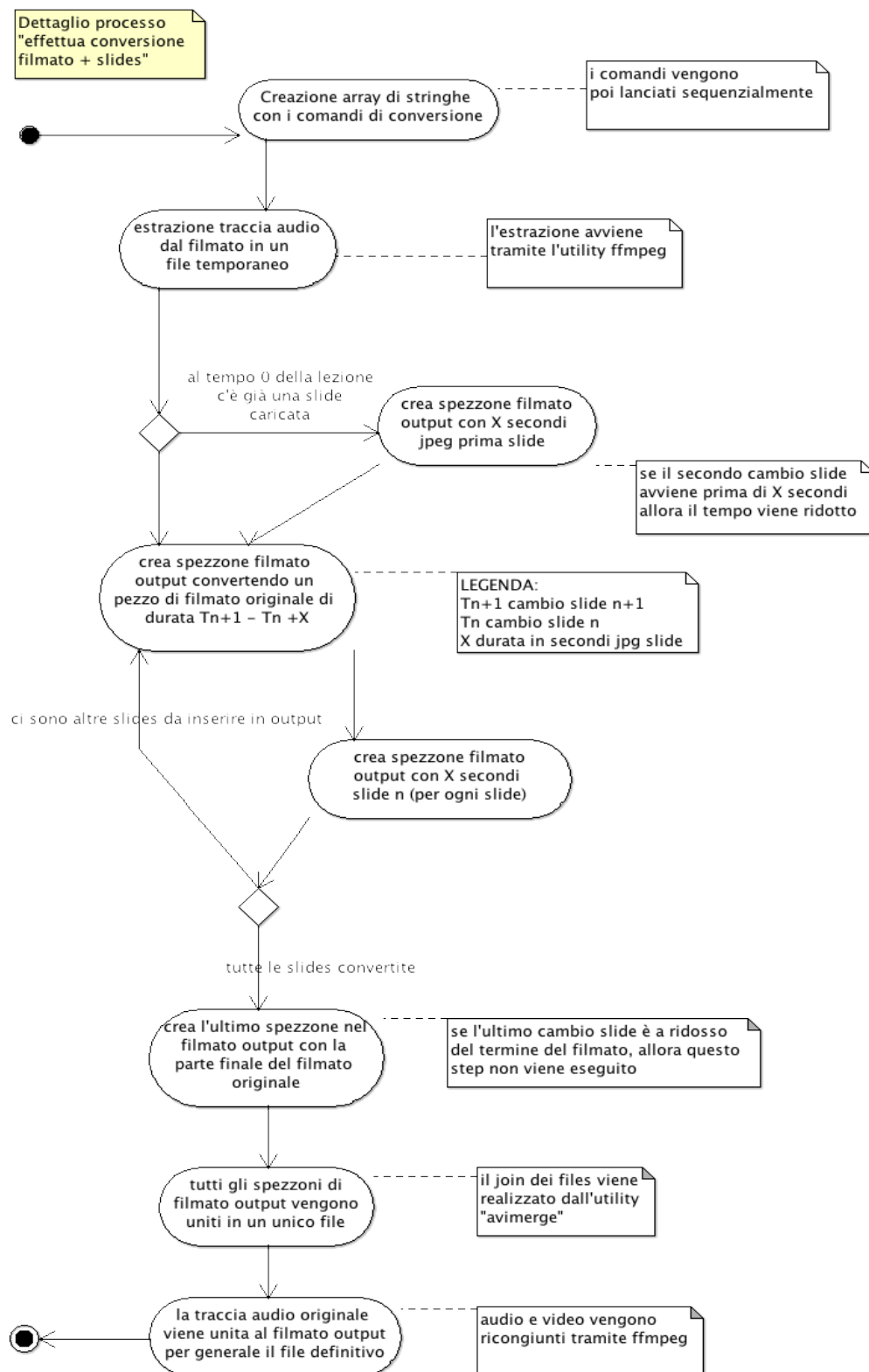


Figura 19: activity diagram del processo di conversione (3/3)

La messa a punto dei parametri da utilizzare con ffmpeg è stata la vera difficoltà di questo lavoro, ed ha richiesto molto tempo e ripetute prove comparative. Si sono verificati diversi problemi, soprattutto errori di sincronizzazione tra le tracce audio e video. Questo ha determinato la necessità di introdurre, nella procedura di conversione, un parametro di correzione per inserire un ritardo di circa mezzo secondo ogni 5 minuti nella traccia video del filmato; in questo modo il prodotto finale, in base alle prove eseguite, è privo di errori.

Un'ultima annotazione tecnica riguarda il “merge” finale di tutti gli spezzoni video così prodotti: per implementare questa funzione si è ricorsi all'utility “avimerge”, che in ambiente Mac OSX si trova nello stesso pacchetto (denominato ffmpegX) in cui si trova anche ffmpeg.

### **4.3 Risultati**

Il prodotto della conversione sopra descritta è, in conclusione, un file in formato mp4 contenente la videolezione convertita dal formato di Lode a quello compatibile con iTunesU. Questo file può essere (per esempio) direttamente copiato sul server dell'Università di Trento che è connesso al canale iTunesU, e così facendo potrà essere usufruito direttamente da tutti gli utenti della platea iTunes store, dopo le opportune operazioni di gestione del canale attraverso iTunes. Infatti, come descritto nel terzo capitolo, per le università non statunitensi Apple fornisce (tramite feeds rss) solo la pubblicazione dei riferimenti alle rispettive risorse audio/video all'interno di iTunes store, e dunque spetta ai singoli atenei allestire un server che contenga e distribuisca tali risorse. Questo semplifica da un certo punto di vista la possibilità di effettuare la pubblicazione delle videolezioni così convertite verso iTunesU utilizzando varie forme di automatismo.

Il file mp4 così prodotto si presta comunque ad altri utilizzi, quali ad esempio una distribuzione interna e diretta agli eventuali utenti interessati.

Un'ultima annotazione è utile per dire che, anticipando i contenuti dei prossimi due capitoli, se sul server di Ateneo connesso al canale iTunesU, assieme al file video, si pubblicano anche alcuni appositi files xml, è possibile ottenere due servizi con un'unica installazione, in quanto le stesse risorse possono essere usate anche da Lode4iPhone.

## 5. Apple iPhone

### 5.1 Caratteristiche di iPhone

Il prodotto iPhone è stato introdotto sul mercato dall'azienda Apple inc. nel 2007, anche sulla scorta del successo di un altro dispositivo mobile prodotto dalla stessa Apple: l'iPod, che ha rappresentato un significativo passo in avanti nella modalità di distribuzione e ascolto della musica digitale.

iPhone è un telefono ma anche un dispositivo per connettersi ad Internet, gestire l'email, vedere pagine web, vedere filmati youtube, ascoltare musica, scattare e vedere foto, visualizzare mappe, usare il gps, giocare... ed altro ancora. In sintesi è uno smartphone, anche se il termine è un po' riduttivo alla luce di tutte queste funzionalità.

Tutte le informazioni fornite in questo capitolo si riferiscono -salvo diversa indicazione- al dispositivo iPhone4 ed al sistema operativo iOS4.1, che sono stati rilasciati nel corso del 2010.

Per gli scopi di questo elaborato finale, va sottolineato in particolare che iPhone:

- è dotato del software iTunes, che gli permette ad esempio di connettersi ad iTunesU
- è in grado di riprodurre files audio e video anche di grande dimensione
- ha uno schermo sufficientemente ampio (3,5" diagonale) e con buona risoluzione (max. 960x640 pixel a 326 ppi), adatto alla visualizzazione di videolezioni
- è predisposto all'installazione di applicazioni di terze parti, se sviluppate con l'apposito sdk – come vedremo nel seguito del capitolo.

Le caratteristiche hardware principali sono:

- Processore Apple A4
- memoria 16 o 32 GBytes
- ram 512 MBytes

- connettività UMTS/HSDPA/HSUPA e GSM/EDGE
- Wi-Fi 802.11b/g/n
- GPS assistito
- Bussola digitale
- Sensori: giroscopio a 3 assi, accelerometro, sensore di prossimità e sensore di luce ambientale

Un tratto distintivo di iPhone e di iOS (il sistema operativo di iPhone) è il multitouch: il touchscreen capacitivo consente di interagire con il dispositivo in modo naturale utilizzando la pressione di più dita contemporaneamente, per esempio quando si vuole fare uno zoom su una foto o su un filmato.

Inoltre iOS4 (solo se installato su iPhone4 o iPhone3GS) supporta il multitasking, ovvero l'esecuzione di diverse applicazioni contemporaneamente: questa caratteristica non era presente nelle precedenti versioni e ciò era stato oggetto di critiche da parte degli utenti.

Uno dei fattori del successo commerciale del prodotto iPhone è sicuramente la disponibilità del servizio App store. Si tratta di un ambiente accessibile in rete da cui si possono scaricare (gratuitamente o a pagamento) applicazioni software (in gergo “apps”) per i dispositivi iPod touch, iPhone e iPad. Il numero e la tipologia delle applicazioni è in costante aumento: al momento della stesura di questo documento se ne contano circa 250.000. Il fatto che siano di facile accesso, scaricamento ed acquisto e che siano generalmente a basso costo o gratuite, fa sì che costituiscano un elemento portante dell'intera piattaforma. Nel prossimo paragrafo sarà approfondito il tema dello sviluppo di queste applicazioni.

## **5.2 Sviluppo di applicazioni per iPhone tramite l'SDK**

Lo sviluppo di applicazioni per iPhone è possibile grazie al fatto che Apple, nel 2008, ha rilasciato un SDK (Software Development Kit) che contiene tutto il necessario per scrivere e produrre software compatibile con questo dispositivo. In fase di test è possibile provarne il funzionamento su iPhone simulator (che si trova all'interno del kit), mentre poi se si

desidera pubblicare il codice su App store è necessaria un'approvazione preventiva formale da parte di Apple. Quest'approvazione avviene innanzi tutto attraverso un'analisi del codice compiuta da un sistema automatico che effettua alcune verifiche, in particolare la ricerca di eventuali chiamate ad API (Application program interface) non ufficiali, cioè non facenti parte delle librerie rilasciate e documentate da Apple stessa. Questo controllo viene effettuato per ragioni di sicurezza e stabilità del sistema operativo iOS. Inoltre l'applicazione deve essere compatibile con una serie di regole abbastanza corposa, contenuta in un documento denominato "App store review guideline". Alcuni dei requisiti principali indicati da questo documento si possono così sintetizzare:

- sono vietate le applicazioni che fanno un uso eccessivo della banda telefonica
- sono vietate le applicazioni che rappresentano prove, test, beta versions o comunque "di limitata utilità", o che replicano funzioni già fornite dal Sistema Operativo
- sono vietate le applicazioni di contenuto dubbio o di cattivo gusto (es. pornografia, violenza ecc.)
- applicazioni di dimensioni maggiori di 20 MBytes, o che presentino bugs, o che scaricano codice aggiuntivo non sono ammesse

Per poter pubblicare le applicazioni sull'App store, ma anche solo per testarle su un dispositivo reale anziché sul simulatore, è necessario sottoscrivere un contratto di tipo "iPhone Developer Program". Esistono tre tipologie di contratto:

- iscrizione individuale o aziendale (con un costo annuale di 99\$)
- iscrizione enterprise – per aziende – con servizi aggiuntivi quali la possibilità di definire applicazioni "in-house" cioè solo per i dipendenti, e l'assistenza personalizzata su problemi con il codice (costo annuale pari a 299\$)
- iscrizione "Developer University Program" (gratuita, ma con molte limitazioni: non è possibile pubblicare il software su App store ma solo installarlo su un set di dispositivi specifici ed individuati a priori, per un numero massimo di 200)

Passando ora ai requisiti hardware necessari per sviluppare codice per iPhone, essi consistono in un computer Apple con processore Intel, dotato di una versione recente ed aggiornata del sistema operativo Mac OS X (al momento in cui scriviamo questo documento, versione 10.6.4) e con una quantità di memoria ram non inferiore ad 1 GBytes (raccomandati 2)

Il linguaggio di programmazione da usare per le applicazioni iPhone è l'objective-c. Si tratta di una versione di C modificata in modo tale da diventare orientata agli oggetti, pur mantenendo la compatibilità sintattica con il C base. L'utilizzo degli oggetti avviene attraverso una sintassi particolare, che deriva da Smalltalk, e si basa sullo scambio di "messaggi". Objective-c era utilizzato sui calcolatori Next, ed è anche il linguaggio su cui si basa "Cocoa", una delle librerie grafiche su cui si fonda anche il sistema operativo Mac OS X.

Lo sviluppo vero e proprio (cioè la scrittura, la compilazione ed il test del codice) avviene grazie ad alcuni ambienti forniti dall'sdk:

- XCode: un IDE (cioè Integrated Development Environment) del tutto simile a prodotti quali Eclipse o NetBeans per lo sviluppo in linguaggio java. Xcode permette di scrivere codice con l'assistenza sulla sintassi e sulle librerie collegate, e lanciare la compilazione e il debugging. In figura 20 si può vedere come appare XCode con un progetto aperto e già popolato con alcuni files
- Interface builder (un prodotto che permette di costruire interfacce grafiche in modo interattivo e visuale)
- iPhone simulator (un simulatore per testare il codice prodotto)
- Instruments (è un tool di analisi del comportamento dell'applicazione in runtime – particolarmente utile per individuare problemi di gestione della memoria)

Proprio quest'ultimo punto rappresenta probabilmente lo scoglio più importante da superare per imparare a programmare per iPhone: infatti, l'assenza di meccanismi automatici di “garbage collection” fa sì che si debba essere assolutamente precisi ed ordinati nei processi di allocazione e rilascio di oggetti in memoria. Si debbono seguire rigidamente le seguenti regole, tratte dal documento “Memory management programming guide” pubblicato sul sito web di Apple (vedi riferimento (36) in bibliografia):

- quando una porzione di codice ha il possesso di un oggetto (cioè quando gli ha inviato un messaggio “alloc” oppure “retain”) allora è anche responsabile del suo rilascio (mediante release o autorelease)
- in tutti gli altri casi (oggetto allocato altrove e ricevuto in utilizzo) non si deve procedere al suo rilascio.

Questa regola, apparentemente semplice, in realtà non è sempre di facile applicazione, soprattutto in casi in cui gli oggetti entrano in gioco come parametri di chiamata ad altri metodi, oppure quando è necessario che la “vita” di un oggetto sia lunga abbastanza per garantire lo svolgimento di tutta la logica del programma. Tale regola rimane comunque un principio da rispettare in tutte le fasi di sviluppo del software.

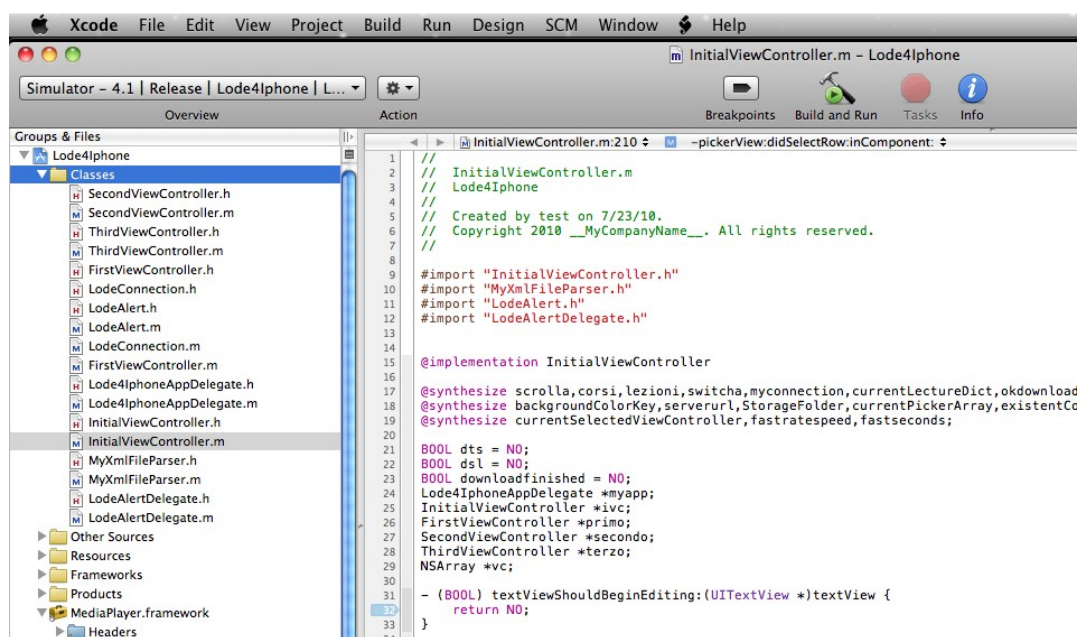


Figura 20: la schermata di progetto di Xcode

Lavorando con Xcode è importante capire il concetto di framework: si tratta di insiemi di librerie di software, raggruppate in modo omogeneo per ambito di applicazione, agganciabili al proprio progetto per essere utilizzate per costruire funzioni più complesse. Ad esempio, nel progetto Lode4iPhone di cui si parlerà nel prossimo capitolo, uno dei fondamenti del progetto è costituito dal framework MediaPlayer, che si occupa di gestire audio e video. I frameworks UIKit (user interface) e Foundation (classi base di objective-c) sono associati di default ad ogni nuovo progetto. Altri esempi di framework sono AddressBook, MapKit, Accelerate ecc.

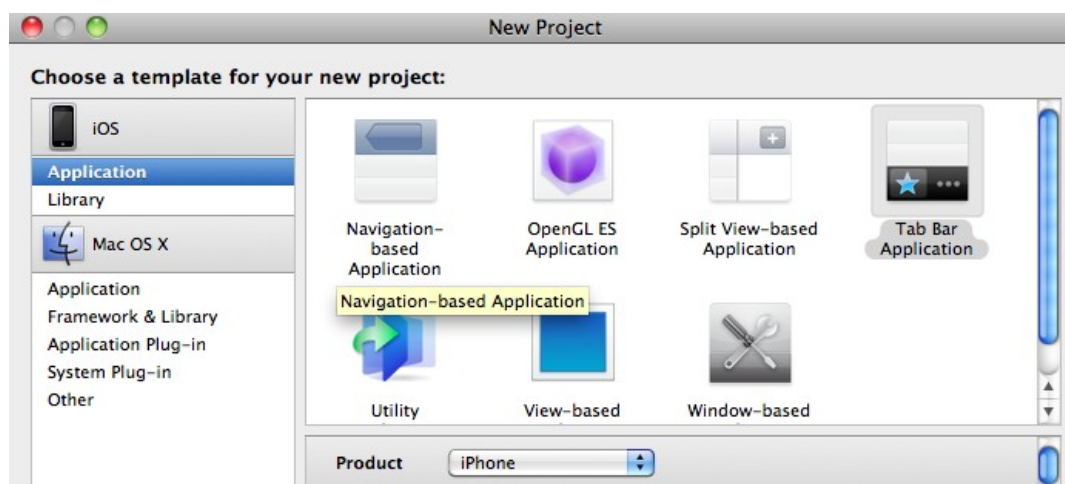


Figura 21: scelta del template per un nuovo progetto in Xcode

I primi passi da compiere per creare una nuova applicazione per iPhone con Xcode comprendono la scelta del template su cui basare il nuovo progetto. Xcode presenta le possibilità di scelta indicate in figura 21. Nel caso di Lode4iPhone è stato scelto “Tab Bar application” per avere un layout dell'interfaccia grafica preimpostato in modo da riservare una porzione nella parte inferiore dello schermo per una barra comandi contenente alcuni pulsanti che permettono – nel nostro caso – di cambiare “view”, quindi in qualche modo “vista” o “contesto grafico” dell'applicazione (come vediamo in figura 22).

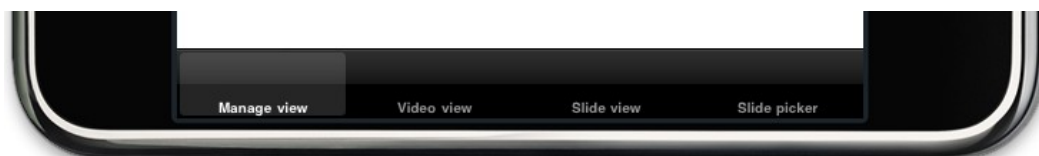


Figura 22: tab bar per l'applicazione Lode4iPhone



La scelta iniziale (che di norma, per i progetti più semplici, è “view-based application”) non preclude comunque di riorganizzare anche in seguito la gerarchia di oggetti grafici in modo diverso; però gli automatismi del setup iniziale legati a ciascun template facilitano il compito del programmatore creando gli oggetti nel modo più corretto per quel tipo di applicazione, e preimpostando alcune inizializzazioni.

E' importante dire che le applicazioni per iPhone devono essere organizzate, dal punto di vista del funzionamento dell'interfaccia grafica (interfaccia utente) secondo il pattern MVC (Model View Controller) rappresentato graficamente per blocchi in figura 23.

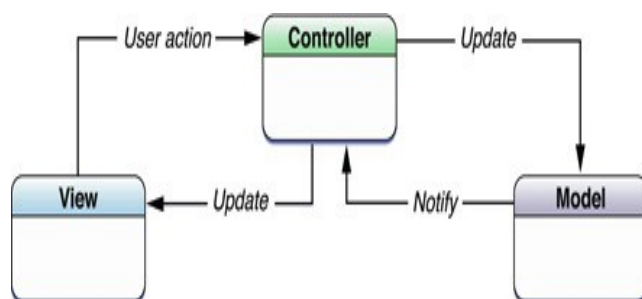


Figura 23: organizzazione e ruoli degli oggetti nel pattern MVC (da [www.devapp.it](http://www.devapp.it))

In ogni progetto Xcode quindi vanno create delle classi che svolgono il ruolo di controller (e sono sottoclassi della classe base “UIViewController”) ed hanno il compito di:

- raccogliere le azioni utente che si svolgono nell'ambito degli oggetti “view” (deputati alla visualizzazione vera e propria dell'interfaccia)
- invocare i metodi negli oggetti “model” che si occupano di gestire i dati e le informazioni
- aggiornare l'aspetto e il contenuto delle views

Le classi “view” create nelle nostre applicazioni sono sottoclassi di UIView, mentre le classi model hanno una struttura libera e personalizzata per ogni applicazione (al limite possono essere unite con i viewcontrollers).

Per quanto riguarda più in dettaglio il contenuto degli oggetti view, che si occupano come detto di gestire l'interfaccia utente (in particolare la GUI Graphical User Interface), per la

loro costruzione si utilizza il tool “Interface builder” che fa parte della suite di programmi fornita con l'sdk.

Questo tool è molto facile da usare e svolge dietro le quinte un enorme lavoro, scrivendo il codice xml necessario per realizzare le interfacce. Il disegno delle views avviene in modalità interattiva trascinando sulla finestra principale gli oggetti di cui abbiamo bisogno prendendoli dall'apposta Library: si tratta ad esempio di DatePicker se abbiamo bisogno che l'utente inserisca una data, di TextField se l'utente deve inserire un testo, di un Button se si deve chiedere la conferma per invocare un'azione, e così via. Gli oggetti della Library sono molti (38), nelle figure 25 e 26 sono riportati i più significativi. Da notare che ciascuno di questi controlli è una sottoclasse della classe principale view, e dunque ne eredita tutte le proprietà ed i metodi.

Interface builder ci permette di effettuare un'altra operazione fondamentale per ogni progetto Xcode: il collegamento tra gli oggetti grafici disegnati sull'interfaccia e le proprietà ed i metodi del rispettivo oggetto viewcontroller collegato. Infatti, in ogni interfaccia disegnata con Interface builder va innanzi tutto impostato il cosiddetto “File's owner”, che può assumere come valore il nome di una delle classi della nostra applicazione che sia sottoclasse di UIViewController. Ad esempio, nell'applicazione Lode4iPhone di cui parleremo nel prossimo capitolo, il “File's owner” dell'interfaccia SecondView è la classe “SecondViewController” (a runtime quest'associazione avverrà con istanze di questa classe). Se tale proprietà è impostata correttamente, allora con Interface builder è possibile collegare i controlli (textfields, labels, datepicker, pulsanti ecc.) con le corrispondenti proprietà e metodi della classe collegata. Infatti, in fase di creazione di ogni ViewController, si possono “etichettare” proprietà e metodi in modo da “avvisare” Xcode che quegli elementi andranno collegati con oggetti di Interface builder. Per le proprietà, si deve inserire in fase di dichiarazione la stringa “IBOutlet”, nel caso dei metodi “IBAction”.

In figura 24 possiamo visualizzare un esempio di queste associazioni. Nel caso specifico, quando l'utente toccherà il pulsante “next”, il controllo sarà passato al metodo nextSlide dell'istanza corrente di SecondViewController.



Figura 24: collegamento tra il pulsante "next" ed il metodo "nextSlide" in Interface builder

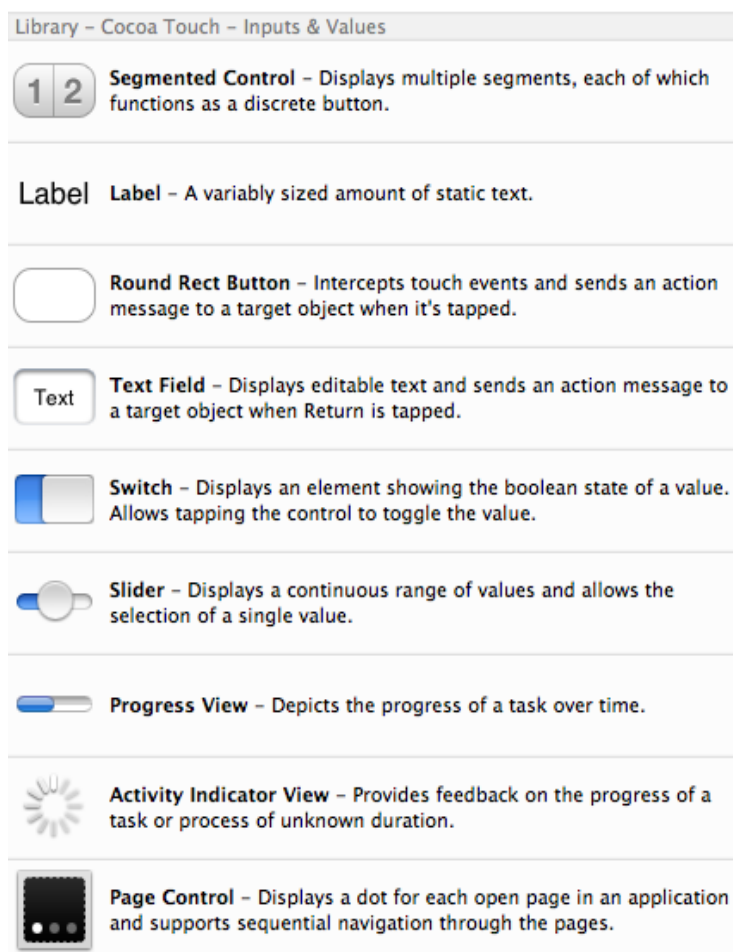


Figura 25: una parte della libreria di controlli di interface builder (1/2)

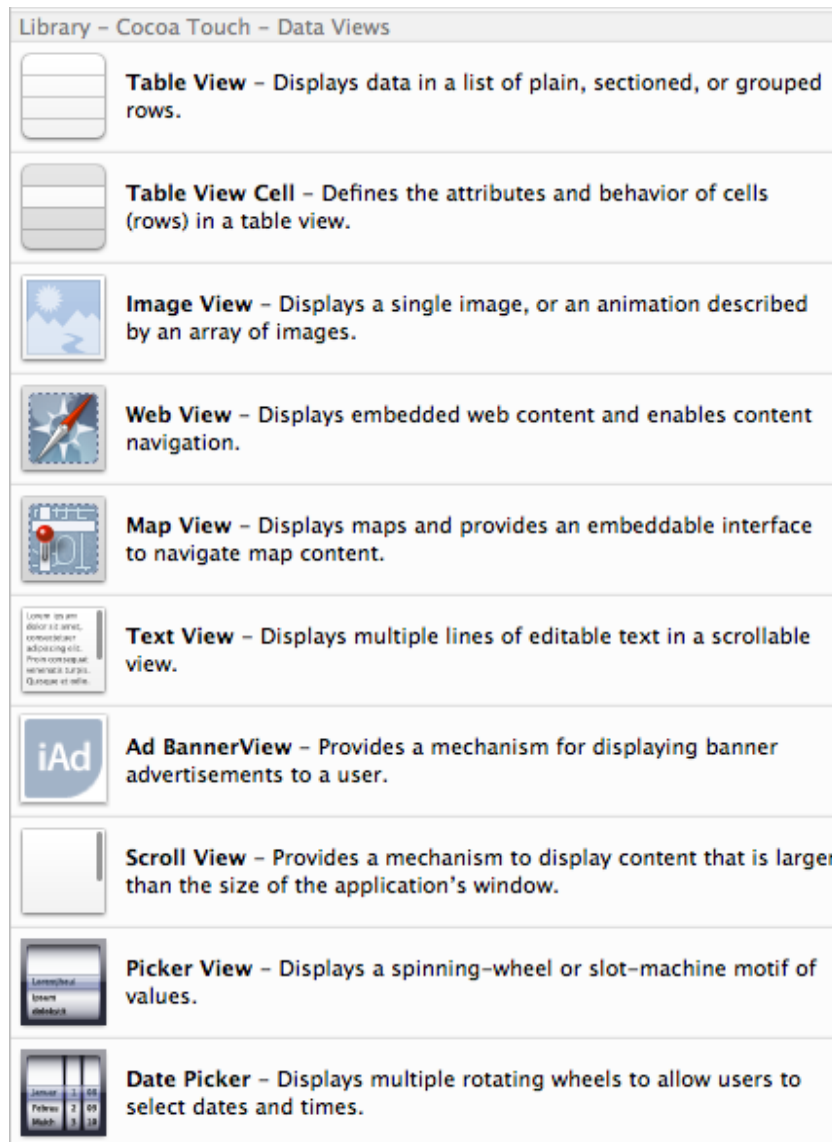


Figura 26: una parte della libreria di controlli di interface builder (2/2)

Altri aspetti fondamentali della programmazione in objective-c per iPhone sono:

1. delegation and protocols
2. internazionalizzazione
3. gestione dei parametri di configurazione (settings) dell'applicazione

Nella sezione seguente si approfondirà seppur brevemente ciascuno di questi tre argomenti.

L'azione di "delegation" rappresenta una soluzione per definire tra più oggetti una forma di coordinamento e collaborazione (di tipo gerarchico) secondo precise regole. Con un esempio pratico si potrà essere più chiari:

supponiamo di aver definito una ClasseA in cui, in qualche metodo, abbiamo bisogno di effettuare il parsing di un file XML, utilizzando l'apposita classe fornita dall'sdk (NSXMLParser):

### **ClasseA**

```
.....
MyParserDelegate *parserDelegate = [[MyParserDelegate alloc] inizializza];
.....
NSXMLParser *parser = [[NSXMLParser alloc] initWithContentsOfURL:[NSURL
fileURLWithPath:base_path]];
[parser setDelegate:[self parserDelegate]];
BOOL okparse = [parser parse];
if (!okparse) {
    LodeAlertWithMessage(NSLocalizedString(@"Error during xml parsing
operation...",@"error parse"));
}
else
.....
```

come possiamo vedere, ho definito un "delegate" per l'oggetto parser: si tratta di un oggetto della classe MyParserDelegate, che sarà definita come segue:

### **MyParserDelegate**

```
@interface MyXmlFileParser : NSObject <NSXMLParserDelegate> {
.....

- (void)parser:(NSXMLParser *)parser didStartElement:(NSString *)elementName
namespaceURI:(NSString *)namespaceURI qualifiedName:(NSString *)qualifiedName
attributes:(NSDictionary *)attributeDict {
```

Qui notiamo due cose:

- nella dichiarazione dell'interfaccia di quest'ultima classe, dichiariamo di implementare il “protocollo” `NSXMLParserDelegate`, in pratica dobbiamo obbligatoriamente implementare (dichiarare e scrivere il codice) alcuni metodi
- il metodo `“didStartElement”` è uno dei metodi obbligatori del protocollo in questione, e nello specifico contiene le azioni da svolgere ogni volta che il parser individua l'apertura di un nuovo tag nei dati xml.

Riassumendo, nella classe `A` il parser inizia ad esaminare il file xml assegnato (azione “parse”); a quel punto, ogni volta che si verificherà uno degli eventi previsti dal protocollo (per esempio l'inizio di un nuovo tag), il controllo di flusso sarà passato all'oggetto indicato come delegate (dunque un oggetto della classe `MyParserDelegate`) che richiamerà il metodo previsto per quel tipo di azione (per esempio `didStartElement`).

Tutto ciò permette di dare più ordine al codice e suddividere i ruoli tra le diverse classi.

Per quanto riguarda l'attivazione dell'internazionalizzazione delle applicazioni per iPhone, si tratta di un'attività che non pone particolari difficoltà, anche se la procedura è poco intuitiva. Per impostare questa caratteristica, si agisce come segue:

- supponendo di voler supportare inglese e italiano, si debbono creare manualmente dall'ambiente “Finder” di Mac OSX, due cartelle `“en.lproj”` e `“it.lproj”` all'interno della cartella del nostro progetto Xcode.
- poi si creano in Xcode due files di tipo `“String”`, denominandoli `“Localizable.strings”` e salvandoli rispettivamente nella cartella `en.lproj` e `it.lproj`.
- in ognuno dei due files, dovrò inserire una serie di righe con la sintassi `“key” = “stringatradotta”`. Per esempio, rispettivamente `“msgalert” = “Are you sure to delete?”` e `“msgalert” = “Sei sicuro della cancellazione?”`
- infine, all'interno del codice delle classi che compongono il progetto Xcode, ogni volta che ho bisogno di una stringa che assuma valori diversi a seconda della

lingua che è stata selezionata nel sistema, dovrò usare la funzione `NSStringLocalizedString(@"key",@"")` per ricavare il valore della stringa tradotta

- è da notare il fatto che, in tale configurazione, l'aggiunta di un'ulteriore localizzazione con altro idioma è particolarmente semplice: si tratta solo di aggiungere un'altra cartella `XX.lproj` ed un file contenente tutte le stringhe con le nuove traduzioni.

Ultimo argomento per quanto riguarda la programmazione in objective-c per iPhone è l'impostazione di parametri di configurazione dell'applicazione. E' possibile (ed auspicabile) far sì che questi parametri, che tipicamente determinano alcuni comportamenti del software, appaiano nell'ambiente "Impostazioni" al pari di tutte le altre applicazioni di iOS. In figura 27 vediamo ad esempio il caso dell'applicazione Lode4iPhone, le cui impostazioni si trovano a fianco di quelle di Safari e di altre applicazioni del sistema.



Figura 27: menù "impostazioni" per Lode4iPhone in iOS

Per definire poi effettivamente quali sono i parametri e descriverne la tipologia ed i valori possibili, occorre aggiungere al progetto Xcode:

1. un oggetto di tipo "Settings.bundle"
2. un file di tipo .plist (property list, altro non è in realtà che un file xml con una propria grammatica) denominato Root.plist e collocato nella cartella Settings.bundle

A questo punto, rispettando una sintassi che non riporteremo qui per ragioni di spazio (è sufficiente comunque utilizzare i nomi e le proprietà dei tags xml come previsto dalla grammatica) si possono effettivamente definire i parametri di funzionamento dell'applicazione. Per esempio, nel caso di Lode4iPhone (vedi figura 27) abbiamo inserito un parametro "URL del server" di tipo stringa, ed il cui valore di default è:

["http://latemar.science.unitn.it/itunes/feeds"](http://latemar.science.unitn.it/itunes/feeds). L'utente dell'applicazione, accedendo all'ambiente "impostazioni", avrà la possibilità di modificare il valore di questo parametro se vuole, ad esempio, puntare ad un server differente.

### **5.3 Confronto con altre piattaforme smartphone**

La scelta di utilizzare la piattaforma iPhone per sviluppare una versione di Lode da utilizzare su dispositivi mobili è stata naturale in conseguenza del fatto che Lode stesso (nella sua componente di produzione delle videolezioni) è stata sviluppata e funziona solo su computers Apple, ed anche perchè uno dei pilastri di questo elaborato finale, ovvero iTunesU con la sua possibilità di fornire una grande vetrina alle videolezioni, a sua volta è un servizio fornito da Apple, e l'accesso ad i suoi contenuti avviene in modo semplice e immediato tramite iTunes, il software che funziona – tra i dispositivi mobili – solo su iPhone. Nonostante questo, se ci si ponesse l'obiettivo di estendere il più possibile l'utilizzo delle videolezioni asincrone su dispositivi mobili (ed in particolare su smartphones), sarebbe necessario prendere in considerazione un "porting" di Lode4iPhone almeno su dispositivi dotati di sistema Android, ed in prospettiva anche su quelli dotati di Windows Phone, il sistema operativo rilasciato recentemente, proprio nel mese di ottobre 2010, da Microsoft.

Da una prima analisi, peraltro poco approfondita, degli strumenti forniti con l'sdk del sistema Android, pare che il porting sia possibile su questa piattaforma, in quanto le funzionalità offerte da Lode4iPhone trovano – sulla carta – sufficiente copertura nelle corrispondenti librerie software fornite da Android. Peraltro, come si avrà modo di sottolineare nel prossimo capitolo di questo elaborato finale, è difficile prevedere il funzionamento di molte librerie e la loro quantità di bugs prima di un test approfondito.

Gli altri sistemi operativi per dispositivi mobili sono meno interessanti nell'ottica di questo progetto, in quanto non forniscono all'utilizzatore un SDK per sviluppare applicazioni



personalizzate. E' interessante in tal senso il grafico rappresentato in figura 28 (fonte Gartner) che si riferisce al secondo quadrimestre 2010; come si può vedere, oltre ai già citati Android e Apple iOS, la quota prevalente di smartphones venduti utilizza il sistema Symbian, mentre una quota comunque significativa (RIM) è detenuta anche dal sistema Blackberry. Tali quote significative si spiegano, a parere di chi scrive, da un lato con ragioni di continuità rispetto al passato: si tratta dei primi sistemi che hanno occupato il settore (in particolare con servizi quali la push email molto apprezzati tra gli utilizzatori professionali); dall'altro lato, con ragioni economiche: gli smartphones con funzionalità più avanzate (tra questi i dispositivi con iOS e Android) hanno un costo maggiore.

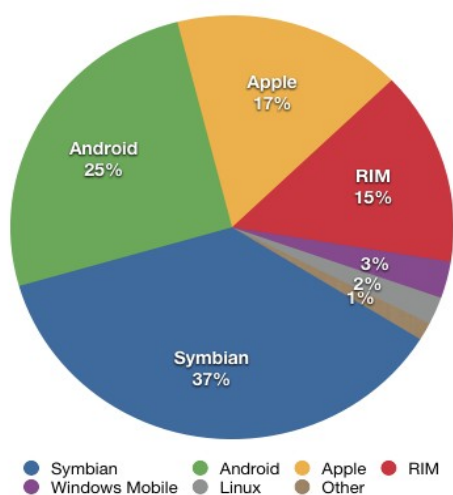


Figura 28: diffusione dei sistemi operativi per smartphone - secondo quadrimestre 2010 (fonte: wikipedia / Gartner)

Più in generale, è interessante notare come lo stesso articolo di Gartner<sup>1</sup> riporti l'indicazione che, tra il secondo quadrimestre 2009 ed il secondo quadrimestre 2010, le vendite mondiali di smartphones siano aumentate del 50.5%, mentre nello stesso periodo le vendite di dispositivi mobili in generale (diciamo per semplicità telefoni cellulari) si sono incrementate solo del 13.8%.

<sup>1</sup> <http://www.gartner.com/it/page.jsp?id=1421013>



## 6. Sviluppo di Lode4iPhone

### 6.1 Introduzione – obiettivi

Nei capitoli precedenti si è parlato dell'importanza di un'interfaccia funzionale e flessibile nell'utilizzo delle videolezioni. Su questo punto, abbiamo visto nel capitolo due come il sistema Lode rappresenti un buon esempio di questo tipo di interfacce, in quanto permette



*Figura 29: l'icona di avvio di Lode4iPhone*

all'utente di vedere in due riquadri -affiancati e sincronizzati- sia il filmato che ritrae il relatore, sia le slides che questo ha utilizzato, ed inoltre sono presenti strumenti efficaci per navigare all'interno dei contenuti (timeline interattiva, accesso per argomento).

Si è parlato poi del canale iTunesU, e di come sia importante per un Ateneo disporre di uno strumento che permette di raggiungere un pubblico molto vasto, come è quello di iTunes store, per divulgare i contenuti multimediali autoprodotti che altrimenti resterebbero per forza confinati in un ambito più ristretto. Questa considerazione ha portato alla scrittura del modulo software descritto nel capitolo quattro, un plugin che realizza la conversione delle videolezioni registrate con Lode verso un formato compatibile con iTunesU, realizzando al contempo un mix di filmato e slides per cercare di replicare in minima parte l'interfaccia di Lode sopra ricordata.

A questo punto, anche alla luce di quanto detto nel capitolo cinque in merito all'importanza crescente che i dispositivi mobili, e soprattutto i cosiddetti smartphones, hanno assunto nel mercato del settore ICT, la decisione di affrontare lo sviluppo di un "porting" di Lode su piattaforma mobile, ed in particolare su iPhone, è una scelta naturale; infatti, questa scelta consente da un lato di riprodurre su uno smartphone (una categoria di dispositivi sempre più diffusa) un'interfaccia per vedere le videolezioni simile a quella di Lode, dall'altro di ottenere – come vedremo - un riutilizzo delle risorse multimediali già pubblicate per iTunesU, evitando inutili duplicazioni.

Questo capitolo dunque si pone l'obiettivo di descrivere gli aspetti di sviluppo e di funzionamento di Lode4iPhone, che è poi l'elemento principale di questo elaborato finale.

L'applicazione Lode4iPhone per funzionare ha bisogno che siano soddisfatti i seguenti requisiti:

- dispositivo iPhone 3, 3GS o 4 (con ogni probabilità può funzionare anche su iPad e iPod touch ma non è stato fatto a riguardo alcun test)
- sistema iOS 4.1 o superiore
- (per scaricare le lezioni) un server web accessibile in modo pubblico, senza autenticazione, che rispetti la gerarchia di cartelle e files - che sarà in seguito descritta - per erogare i contenuti multimediali che costituiscono la videolezione.
- (per scaricare le lezioni) una connessione di rete preferibilmente wifi; quando una lezione è stata scaricata in locale su iPhone, per visualizzarla non occorre alcuna connessione di rete attiva.

## 6.2 Analisi dei dati - lato server

Per quanto riguarda il lato server, affinché Lode4iPhone possa attivare correttamente le proprie funzioni di accesso e selezione dei corsi e delle relative lezioni, la struttura della cartella web che eroga i contenuti (ed il cui url può essere specificato dall'utente tra le impostazioni di configurazione di Lode4iPhone) deve essere come rappresentato in figura 30.

Per una miglior comprensione è opportuno commentare tale struttura:

- il file COURSES.XML contiene l'elenco dei corsi per i quali sono pubblicate lezioni compatibili con Lode4iPhone. Lo schema è molto semplice, e contiene tanti tag "COURSE" quanti sono i corsi da erogare, ad esempio:

```
<COURSES>
  <COURSE>
    <IDC>WebArchitectures</IDC>
    <TITOLOC>Web Architectures</TITOLOC>
    <YEAR>2008</YEAR>
    <DOCENTEC>Marco Ronchetti</DOCENTEC>
    <FOLDERC>WebArchitecture</FOLDERC>
  </COURSE>
  <COURSE>
    .....
  </COURSE>
</COURSES>
```

WebLode4iPhone	
Nome	Data di modifica
COURSES.XML	oggi, 17.01
WebArchitecture	oggi, 17.11
lecture1	oggi, 17.08
lecture2	oggi, 17.08
lecture3	oggi, 17.09
img	oggi, 17.09
1.jpg	25 settembre 2008 20.02
2.jpg	25 settembre 2008 20.02
3.jpg	25 settembre 2008 20.02
4.jpg	25 settembre 2008 20.02
5.jpg	25 settembre 2008 20.02
6.jpg	25 settembre 2008 20.02
7.jpg	25 settembre 2008 20.02
8.jpg	25 settembre 2008 20.02
9.jpg	25 settembre 2008 20.02
TIMED_SLIDES.XML	24 settembre 2008 15.02
lecture4	oggi, 17.11
lecture5	oggi, 17.11
lecture6	oggi, 17.11
lecture7	oggi, 17.11
LECTURES.XML	10 ottobre 2010 22.54
Programmazione2	oggi, 17.10
lecture1	oggi, 17.10
lecture2	oggi, 17.10
lecture3	oggi, 17.10
lecture4	oggi, 17.10
LECTURES.XML	10 ottobre 2010 22.02

Figura 30: cartella web necessaria per distribuire le videolezioni per Lode4iPhone

- ogni corso possiede inoltre una propria cartella, il cui nome deve coincidere con il contenuto del suddetto tag "FOLDERC". All'interno di questa cartella troviamo innanzi tutto un file LECTURES.XML, che, come è intuibile, conterrà l'elenco delle lezioni di quel corso. Anche lo schema di LECTURES.XML è semplice:

```
<LECTURES IDC="WebArchitecture">
  <LECTURE>
    <IDL>1</IDL>
    <URLLEZ>http://server/VideoLecture1.mp4</URLLEZ>
    <FOLDERL>WebArchitecture/lecture1</FOLDERL>
    <DATEL>15-09-08</DATEL>
    <DOCENTEL>Marco Ronchetti</DOCENTEL>
    <TITOLOL>01_Architectural Overview</TITOLOL>
  </LECTURE>
  <LECTURE>
    .....
    .....
  </LECTURE>
</LECTURES>
```

ogni tag "LECTURE" indica una lezione, che deve essere popolata con le informazioni previste. Tra queste, "URLLEZ" è l'url http che punta al filmato della lezione in formato mp4. Nel caso dell'Università di Trento, questo url coinciderà con quello indicato, per quella lezione, anche all'interno del canale iTunesU;

- proseguendo l'analisi della struttura rappresentata in figura 30, vediamo che ogni lezione ha una propria sottocartella: "lecture1", "lecture2" ecc. All'interno di ciascuna di queste sottocartelle ci sono due ulteriori elementi: il primo è una cartella "img" che contiene n immagini jpg, tante quante sono le slides utilizzate dal docente per quella lezione; il secondo è un altro file xml denominato TIMED\_SLIDES.XML. Anche in questo caso, osserviamone brevemente la struttura attraverso un esempio:

```
<TIMED_SLIDES>
  <slide>
    <tempo>0</tempo>
    <titolo>XML Serialization</titolo>
    <immagine>img/3.jpg</immagine>
  </slide>
  <slide>
    <tempo>182</tempo>
    <titolo>Using XML to serialize Java Classes</titolo>
    <immagine>img/4.jpg</immagine>
  </slide>
  <slide>
    .....
    .....
  </slide>
</TIMED_SLIDES>
```

ogni tag "slide" rappresenta un "cambio slide" fatto dal docente durante la lezione e replicato dall'operatore di Lode. Ciò significa che, siccome ogni slide può essere proiettata più volte nell'ambito di una lezione (e non necessariamente in modo

sequenziale e consecutivo rispetto all'ordine nel file di origine), si può avere un numero di tag “slide” diverso dal numero effettivo di slides che compongono la presentazione powerpoint/pdf. Il tag “tempo” indica (in secondi) l'istante in cui è avvenuto un certo cambio di slide, mentre “titolo” e “immagine” si riferiscono rispettivamente al titolo della slide (molto importante, come vedremo in seguito) e al percorso dove si trova l'immagine jpg della slide stessa (percorso relativo rispetto alla cartella della lezione).

Dunque, per allestire su un comune server web una cartella compatibile con l'applicazione Lode4iPhone non è necessario un lavoro particolarmente oneroso: i files TIMED\_SLIDES.XML e le immagini jpg delle slides sono già disponibili nella cartella prodotta da Lode nel post-process di ciascuna lezione, e pertanto devono solo essere copiati. Il lavoro aggiuntivo quindi consiste nella creazione degli altri due files xml (COURSES.XML e LECTURES.XML) nonché della gerarchia di cartelle sopra definita. Da sottolineare nuovamente che il file con il filmato .mp4 può essere (fisicamente) lo stesso usato per il canale iTunesU.

### **6.3 Funzionamento di Lode4iPhone – lato client**

L'applicazione Lode4iPhone, nella sua componente client (quella visibile all'utente), è il risultato della trasposizione su iPhone di un'interfaccia funzionalmente analoga a quella prodotta da Lode per la visualizzazione di videolezioni asincrone, che tenga conto inoltre delle dimensioni ridotte del display.

Per raggiungere questo scopo, durante la progettazione si è tenuto in considerazione che, date appunto le dimensioni dello schermo di un iPhone, non è possibile presentare nella stessa schermata tutti gli elementi di cui è composta l'interfaccia originale di Lode (riquadri video e slides, timeline, elenco titoli slides selezionabili). La scelta dunque è stata quella di dividere l'output in quattro diverse “views”, tra le quali l'utente può muoversi cliccando uno dei pulsanti della tab bar posizionata nel lato inferiore dello schermo. Tale suddivisione non determina però un comportamento indipendente e disordinato delle viste, ma come

vedremo l'applicazione è sviluppata in modo tale che vi sia coerenza tra gli oggetti mostrati nei quattro diversi punti di vista.

All'avvio, che avviene quando l'utente tocca il pulsante (icona) raffigurato in figura 29, il primo oggetto eseguito è la classe statica main (come in ogni programma C). Subito dopo avviene il caricamento della prima finestra grafica, che nel nostro caso si chiama MainWindow; tale comportamento è correlato al contenuto del file di configurazione Lode4iPhone-Info.plist che contiene diverse impostazioni per l'applicazione.

La finestra MainWindow carica tutti gli altri elementi principali dell'interfaccia grafica: la tab bar e poi la vista principale, ovvero "Gestione" (o "Manage view" se la lingua selezionata è l'inglese) . Il risultato è quello visualizzabile in figura 31.

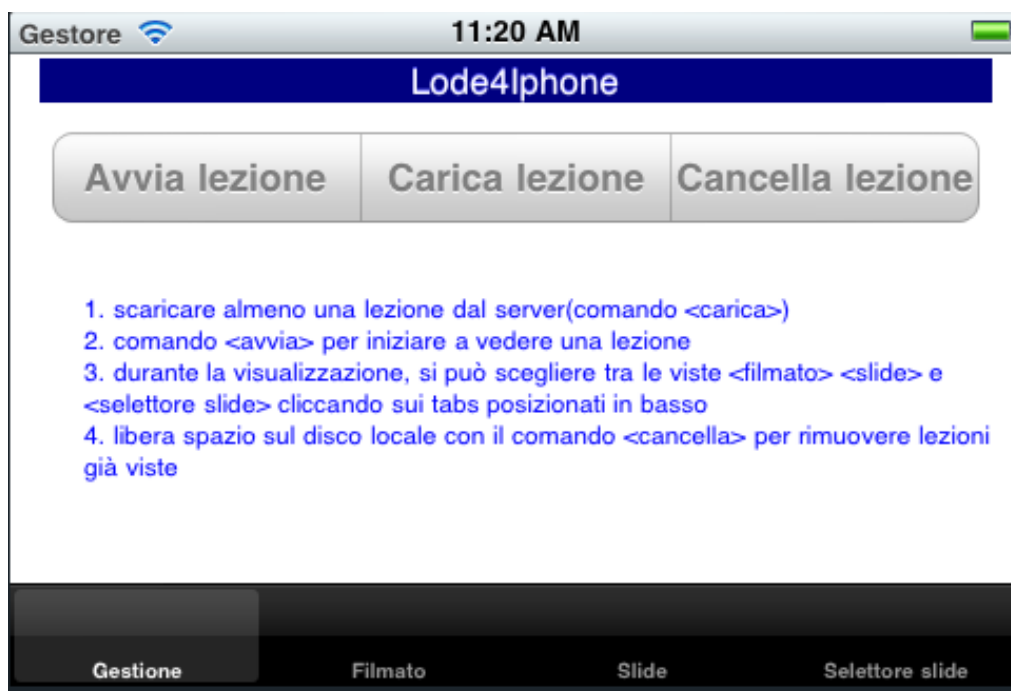


Figura 31: la schermata di avvio di Lode4iPhone

Nella vista "Gestione" di possono eseguire tre operazioni:

1. avviare una lezione, cioè caricarla dal disco locale dell'iPhone (se è stata scaricata in precedenza dal server) ed iniziarne la riproduzione. Dopo aver scelto questo comando, all'utente viene richiesto di indicare quale tra le lezioni presenti sul disco locale vuole visualizzare



2. scaricare una lezione dal server, via rete. Ricordiamo che l'url del server contenente le lezioni è personalizzabile attraverso le impostazioni dell'applicazione (vedi anche figura 27 nel capitolo 5). Scegliendo questo comando si caricano gli oggetti che ci consentono di scegliere quale lezione (e di quale corso) scaricare dal server verso il dispositivo iPhone. Questa procedura sarà spiegata più in dettaglio nel seguito del capitolo;
3. rimuovere una lezione dal disco locale, per liberare spazio.

Vediamo il caso 1: riprodurre una lezione già presente sul disco locale.



Figura 32: avvia lezione su Lode4iPhone

Nella figura 32, la situazione rappresentata è quella in cui solo una lezione è presente sul disco locale dell'iPhone, e quindi viene solo chiesto di confermare la scelta. Se ci fossero state in archivio lezioni di corsi diversi, l'applicazione avrebbe richiesto la scelta prima del corso e poi della lezione all'interno del corso scelto.

Il caso 2: scaricare una lezione dal server attraverso la rete (comando "Carica lezione")



Figura 33: scelta del corso (comando "Carica lezione")

In figura 33 possiamo vedere come, se la connessione di rete è attiva (diversamente apparirà una finestra di errore che ci avvisa dell'assenza di connettività) e se il server è configurato correttamente, otteniamo una lista sotto forma di "picker view" che ci invita a scegliere uno tra i corsi di insegnamento contenenti videolezioni presenti sul server.

Quando l'utente ha scelto il corso, la schermata successiva – vedi figura 34 – richiede un'ulteriore scelta, questa volta tra le lezioni disponibili all'interno del corso selezionato.

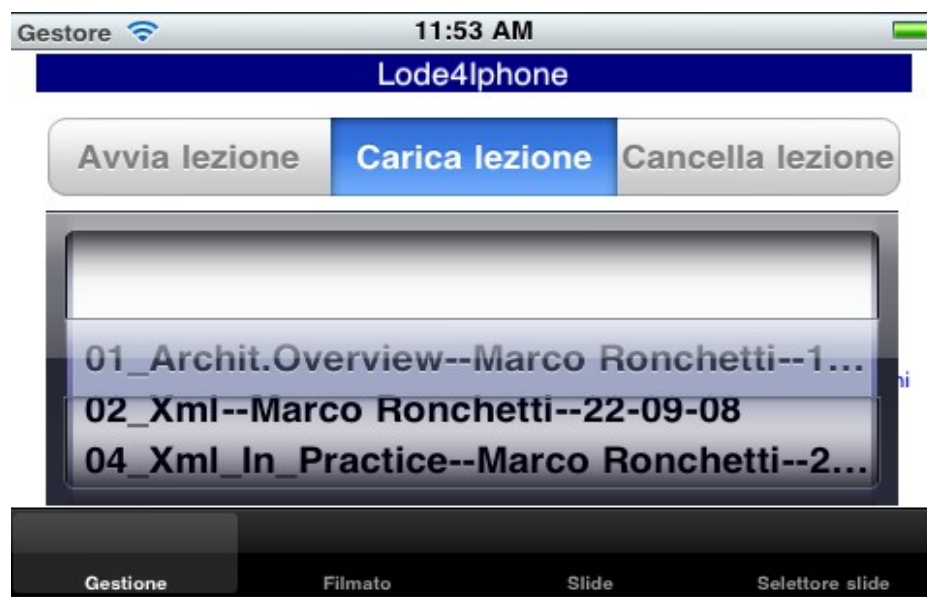


Figura 34: scelta della lezione (comando "Carica lezione")

Come ultimo passaggio di quest'operazione, viene richiesta all'utente una conferma per procedere allo scaricamento della videolezione dal server, poiché si tratta di un'operazione che richiede un certo tempo e terrà occupato il dispositivo (nel caso l'applicazione venga terminata prima della conclusione del download non ci sono comunque conseguenze sulla stabilità dell'applicazione stessa). In figura 35 vediamo rappresentato questo avviso, mentre in figura 36 vediamo come la progressione numerica mostrata durante il download comunichi all'utente lo stato di avanzamento.



Figura 35: finestra di conferma per iniziare il download di una lezione

Ovviamente il tempo di scaricamento dipende da più fattori in parte non quantificabili a priori (velocità della connessione di rete, carico di lavoro del server ecc.). Indicativamente, le prove eseguite hanno mostrato che una lezione di un'ora e mezzo richiede circa quindici minuti di download (ricordiamo che, oltre al file mp4 del filmato, debbono essere scaricati i files xml di servizio e le immagini jpg delle slides).

Una soluzione per evitare tempi di attesa che talvolta possono essere fastidiosi potrebbe essere quella di individuare un metodo per caricare la lezione direttamente da un computer

dove sia stata scaricata in precedenza, attraverso il cavetto usb in dotazione dell'iPhone, ma questa soluzione è risultata di non facile realizzazione.



Figura 36: stato di avanzamento del download della lezione

Quando il download è terminato, l'applicazione cambia automaticamente vista e si posiziona su "Video view", caricando il filmato nell'apposito riquadro (come si può vedere in figura 37).

Tornando ai comandi selezionabili dalla finestra principale, l'opzione 3 è la cancellazione di una lezione dal disco locale. Il funzionamento di quest'operazione è analogo al caso 2 dal punto di vista dell'interfaccia, nel senso che ci viene richiesto di scegliere attraverso un oggetto "pickerview" il corso/la lezione da cancellare, e quindi viene richiesta una conferma per procedere, data l'irreversibilità della cancellazione.

Una volta che una lezione da riprodurre è disponibile, che sia stata appena scaricata o che fosse già presente sul disco locale, la vista "Video" è quella da cui partire per analizzare tutte le opzioni disponibili all'utente per fruire dei contenuti.

Osservando la figura 37, possiamo annotare alcune caratteristiche dell'interfaccia:

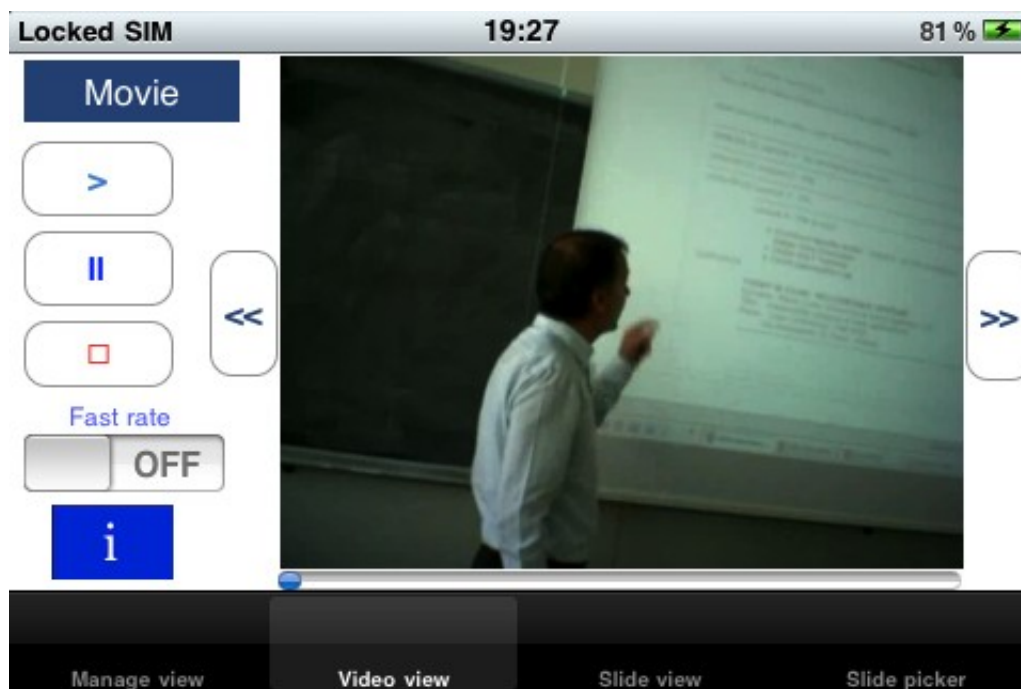


Figura 37: la vista principale (Video view) durante la riproduzione di una lezione

- a sinistra sono posizionati i pulsanti play, pause e stop che comandano la riproduzione del filmato e – indirettamente – dell'intera lezione (slides comprese)
- lo switch "Fast rate" (o "Accelerazione") ci permette di riprodurre la lezione ad una velocità maggiore di quella normale, con fattore di 1.3X (nelle impostazioni si può selezionare anche 1.5X). Quest'opzione può essere utile per vedere più rapidamente l'unità didattica mantenendo la comprensione dell'audio
- il pulsante contrassegnato da una "i" apre una finestra che fornisce informazioni (nome corso, titolo lezione, nome docente) sulla lezione correntemente visualizzata
- infine, i due pulsanti posizionati ai lati del riquadro del filmato consentono all'utente di avanzare (o retrocedere) velocemente spostando il punto di visione (e ascolto) di 10 secondi (nelle impostazioni questo valore può essere aumentato a 30 secondi). Da notare che, subito sotto il riquadro contenente il filmato, si trova una piccola barra di avanzamento da cui si può capire visivamente il tempo trascorso dall'inizio della lezione rispetto alla sua durata complessiva.

Durante la visione di una lezione, sia che il filmato (e dunque anche l'audio) sia in riproduzione sia che si trovi in pausa, possiamo passare alla vista "Slide" premendo l'apposito pulsante sulla tab bar in basso. La vista si modifica come mostrato in figura 38: viene mostrata la slide corrente, cioè la slide che in quel preciso momento della lezione il docente stava utilizzando. Se il filmato è in riproduzione, si continua a sentirne l'audio.

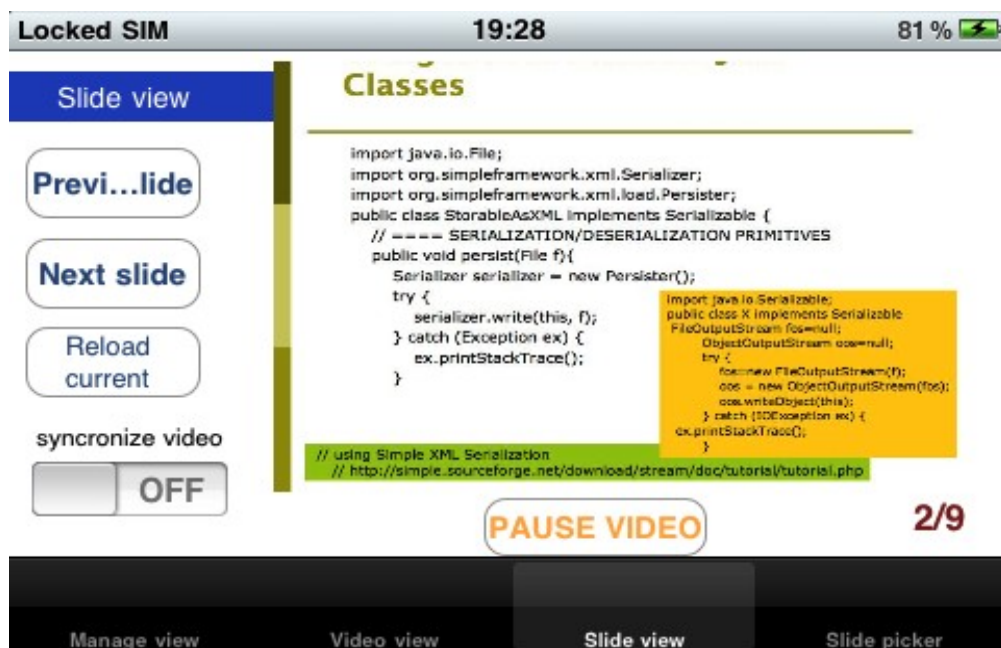


Figura 38: vista "Slide" durante la riproduzione di una lezione

Il filmato può essere messo in pausa anche senza tornare alla vista "Video view", grazie all'apposito pulsante "pause video" collocato in basso. Sul lato sinistro sono attivi alcuni pulsanti di comando: "Previous slide" e "Next slide" caricano rispettivamente la slide precedente o successiva a quella corrente. Quest'azione ha conseguenze diverse a seconda che sia stato o meno attivato lo switch "sincronize video": se è attivo, allora le suddette azioni previous e next slide, oltre a cambiare la slide visualizzata, spostano anche la riproduzione del filmato in corrispondenza dell'istante in cui la nuova slide ha iniziato ad essere proiettata nella lezione originale. Se lo switch non è attivo, allora quest'operazione non provoca alcun effetto sul filmato. Il pulsante "reload current" ricarica la slide corrente (cioè quella che, in base al tempo di riproduzione attuale nel filmato, risulta essere la slide proiettata dal docente). In pratica, quando lo switch "sincronize video" è spento è possibile fare un browsing tra tutte le slides senza interrompere o modificare il flusso audio/video.



Due annotazioni finali sulla “slide view”:

1. se l'utente non ha cambiato (mediante i pulsanti previous e next) la slide corrente, allora eventuali cambi di slide “natural” che si rendano necessari mano a mano che la riproduzione del filmato prosegue, in base alle azioni originarie del docente, vengono eseguiti automaticamente dall'applicazione attraverso un apposito timer interno
2. se la lettura della slide risulta difficoltosa, è possibile usare le funzioni multitouch di iPhone per ingrandire i dettagli con lo zoom (vedi esempio in figura 39)



Figura 39: ingrandimento (zoom) sui dettagli di una slide

L'ultima view di Lode4iPhone è denominata “Slide picker”, o “Selettore slide” in lingua italiana. In questa schermata è possibile semplicemente scegliere, in base al titolo, una slide tra tutte quelle associate alla lezione corrente. Dopo la scelta, automaticamente l'applicazione si sposta sulla view “Video” modificando sia il punto di riproduzione del

filmato (in corrispondenza della slide scelta, che come è ormai noto ha un proprio “tempo di ingresso”), sia la slide correntemente caricata nella view “Slide”. Lo slide picker appare come in Figura 40.

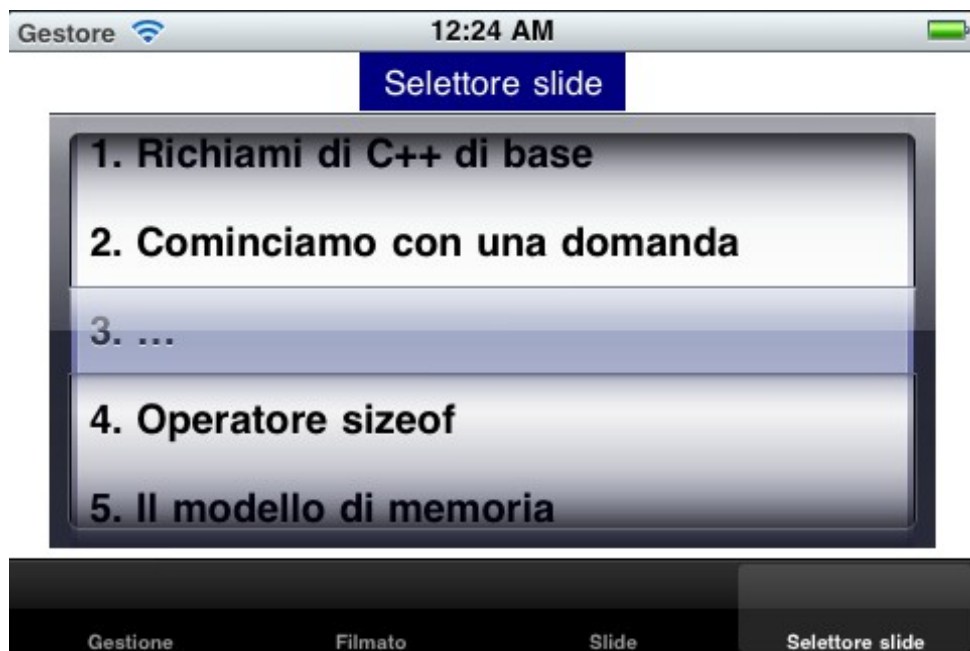


Figura 40: vista "selettore slide"

Riassumendo, abbiamo visto come le varie opzioni attivabili per caricare e poi riprodurre una lezione (che, ricordiamo, deve essere stata registrata con il sistema Lode) permettono di vivere un'esperienza didattica che, seppur mediata dalle tecnologie, si fa apprezzare per completezza e flessibilità, raggiungendo obiettivi simili a quelli delle lezioni prodotte e visualizzate direttamente con il sistema Lode, ad eccezione della “non contemporaneità” tra la traccia audio/video e le slides.

#### 6.4 Analisi dell'architettura di Lode4iPhone

Una descrizione realmente esaustiva dell'architettura del progetto software Lode4iPhone esula dagli scopi di questo documento. Pertanto in questo paragrafo si procederà a definirne ad alto livello la struttura complessiva, e si metteranno in evidenza gli aspetti più significativi.



#### 6.4.1 Classi MVC (Model View Controller)

Dal punto di vista del disegno delle classi, uno dei fattori più importanti è legato alla struttura per la realizzazione dell'interfaccia grafica, che nelle applicazioni per iPhone, come già evidenziato nel capitolo cinque, dev'essere conforme al pattern MVC (Model View Controller). Per implementare questo pattern, in Lode4iPhone sono presenti cinque classi controllers:

CONTROLLER	VIEW CONTROLLATA
Lode4iPhoneAppDelegate	view principale (MainWindow e TabBar)
InitialViewController	Gestione (Manage) view: è la vista iniziale
FirstViewController	Video view: contiene il player del filmato
SecondViewController	Slide view: gestisce la visualizzazione delle slides
ThirdViewController	Selettore slide (Slide picker): tramite l'apposito controllo si scelgono e selezionano i titoli delle slides

Un estratto del contenuto della classe FirstViewController ci permette di capire meglio la relazione esistente tra i diversi elementi in gioco:

```
FirstViewController.h
.....
@interface FirstViewController : UIViewController {
    IBOutlet UIButton *playButton;
    IBOutlet UIButton *pauseButton;
    IBOutlet UIButton *stopButton;
.....
}
.....
@property (nonatomic,retain) IBOutlet UIButton *playButton;
@property (nonatomic,retain) IBOutlet UIButton *pauseButton;
@property (nonatomic,retain) IBOutlet UIButton *stopButton;
.....
- (IBAction) play;
- (IBAction) pause;
- (IBAction) stop;
.....
```

nella prima parte della dichiarazione si trovano le proprietà, mentre le ultime tre righe si riferiscono ai metodi. La parola chiave “IBOutlet” anteposta alla dichiarazione di una proprietà serve per “etichettarla” come variabile che potrà essere connessa ai controlli posizionati sulle interfacce grafiche (le “views”) create con il tool “interface builder”. Analogamente, l'etichetta “IBAction” classifica i metodi della classe come “collegabili” alle azioni che si svolgono (di norma) in risposta ad eventi delle interfacce grafiche, come touch,

zoom, scroll ecc. Quindi in pratica la proprietà UIButton playButton verrà collegata al corrispondente controllo Button, mentre il metodo play verrà collegato all'evento touchUpInside dello stesso pulsante, come visibile in figura 41, che rappresenta uno screenshot di Interface builder.

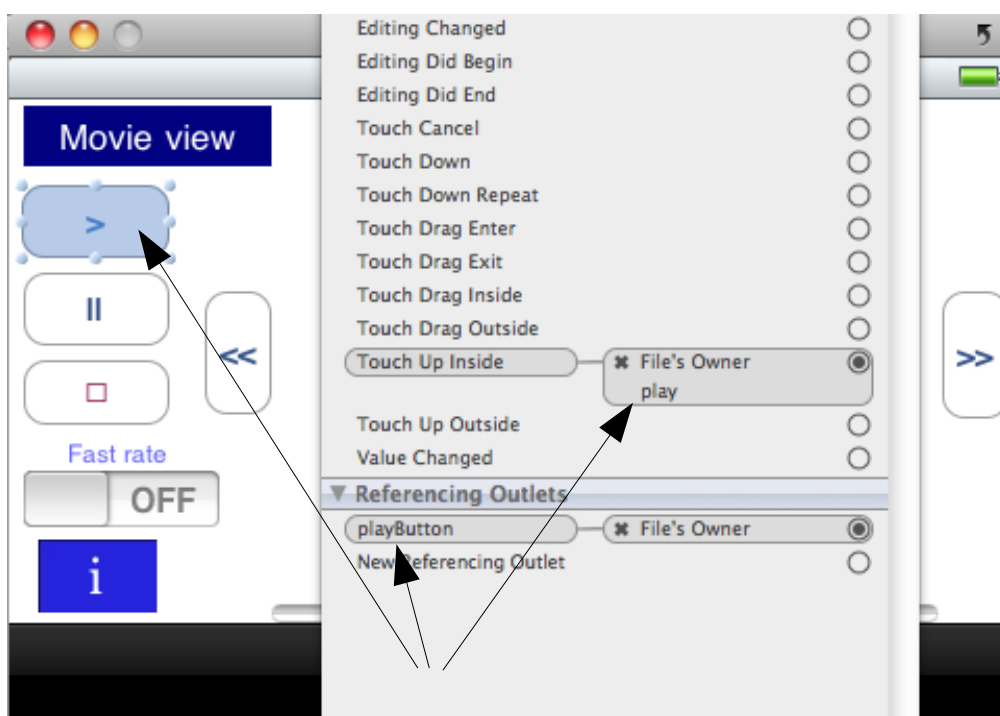


Figura 41: collegamento di FirstViewController con l'interfaccia FirstView

Tra le cinque classi che svolgono il compito di ViewControllers, una si differenzia dalle altre: si tratta di Lode4iPhoneAppDelegate. Innanzi tutto non è una sottoclasse di UIViewController, dunque formalmente è inesatto assegnarle questo ruolo; però si tratta della classe che implementa il protocollo (o, se vogliamo usare terminologie java "implementa l'interfaccia") UITabBarControllerDelegate, che contiene alcuni metodi tra i quali il più importante è certamente:

```
.....
- (void)tabBarController:(UITabBarController *)tabBarController
didSelectViewController: (UIViewController *)viewController
{
}
.....
```

che viene invocato automaticamente ogni volta che l'utente seleziona una delle quattro views disponibili tramite i pulsanti della tab bar. In questo metodo dunque si svolgono molte

azioni cruciali che debbono compiersi quando si cambia la vista corrente. Per esempio, se è stata selezionata la vista “Slide picker”, verrà eseguito il codice qui di seguito riportato per spostare la selezione iniziale del picker stesso in corrispondenza dell'elemento (slide) corrente calcolato in base al tempo trascorso dall'inizio del filmato, calcolo basato anche sulle informazioni contenute in un apposito array creato in fase di partenza dell'applicazione.

```
if (viewController == [controlli objectAtIndex:3]) { //se è stato selezionato n.3
    ....
    if (!(ivc.currentLectureDict == nil)) { //se una videolezione è caricata
        [[tvc pickerview] reloadDataAllComponents];
        // ricarica la sorgente dati del picker

        currenttime = [NSNumber numberWithDouble:[fvc curtime]];
        //legge il tempo corrente del filmato e lo memorizza in currenttime

        int i=0;

        // scorre l'array dei tempi dei “cambi slide” fino a quando non trova
        // quello il cui tempo in secondi è superiore a currenttime
        for (NSNumber *num in [svc arraytempi])
        {
            if ([currenttime doubleValue] > [num intValue]) {
                i++;
                if ([svc arraytempi count] == i) {
                    i--;
                }
            }
            else {
                if (i>0) {
                    i--;
                }
                break;
            }
        }

        //seleziona la riga del picker con il titolo corrispondente
        //alla riga individuata nel ciclo for qui sopra
        [[tvc pickerview] selectRow:i inComponent:0 animated:YES];
    }
}
```

#### 6.4.2 Uso di strutture dati di tipo “Dizionario”

Un altro aspetto fondamentale dell'architettura di Lode4iPhone riguarda la componente di memorizzazione delle informazioni non volatili, cioè che devono essere mantenute anche dopo che una sessione di lavoro del programma è terminata. Tra queste ci sono in particolare i dati delle lezioni scaricate dal server sul disco locale dell'iPhone: oltre al file principale (il filmato in formato mp4) ed agli altri files a corredo della lezione (immagini, files xml) che sono già memorizzati fin da subito sulla memoria di massa, è indispensabile

archiviare anche altre informazioni quali il titolo del corso e della lezione, il nome del docente, il percorso ed il nome del file del filmato ecc.

Per ottenere questo risultato si è scelto di creare una struttura dati d'appoggio di tipo dizionario, che fa uso di elementi organizzati internamente secondo coppie “chiave → valore”, caricata in memoria ram, che garantisce un certo livello di astrazione, un efficiente metodo di accesso ai dati ed una buona leggibilità del codice anche quando (come nel nostro caso) si memorizzano strutture dati di tipo gerarchico a più livelli (corsi → lezioni).

All'atto del caricamento del programma, questa struttura dati viene inizializzata e popolata con eventuali dati già presenti sul disco, caricandoli dalla cartella (cache, tmp o documents a seconda di come è stata impostata nelle preferenze) che fa parte dello spazio assegnato alla nostra applicazione. Ricordiamo infatti che un programma in esecuzione in iOS non può leggere e scrivere indiscriminatamente in qualsiasi posizione del disco, ma solo in un apposito spazio riservato.

Ogni volta che l'utente porta a termine il download di una lezione dal server, il dizionario viene aggiornato con il nuovo record e scritto sul disco, così da essere disponibile per ogni successiva esecuzione del programma; allo stesso modo, quando l'utente rimuove una lezione dall'archivio locale, il dizionario viene anche in questo caso modificato e riscritto su disco.

In figura 42 vediamo rappresentato nel dettaglio questa struttura dati: come si può evincere dallo schema, ci sono quattro livelli di dizionari connessi tra loro gerarchicamente, che tutti assieme compongono l'intero dizionario “corsi/lezioni”. A prima vista tutto ciò può apparire eccessivo per gestire una quantità di informazioni tutto sommato limitata, ma in realtà quest'organizzazione da un lato è adatta sia a memorizzare il dettaglio di ogni elemento (attraverso gli appositi campi), ed inoltre tiene conto della numerosità dei records sia dei corsi sia delle lezioni. Infine, con il tipo dati dizionario i processi di serializzazione/deserializzazione verso la memoria di massa si effettuano praticamente con una sola riga di codice.

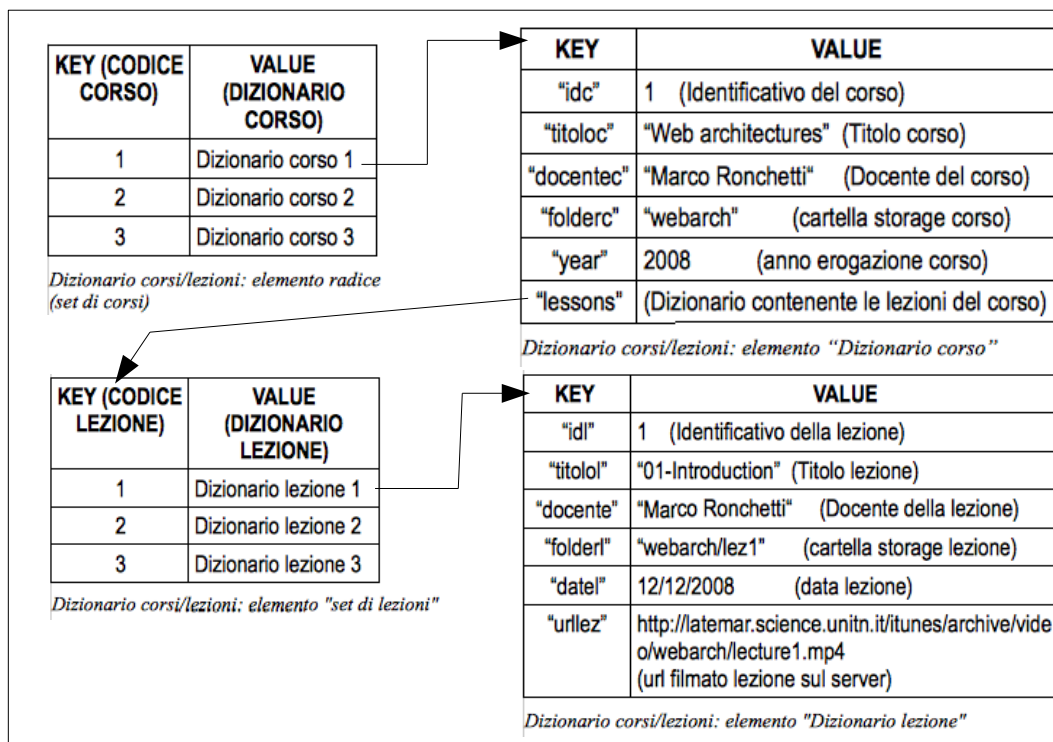


Figura 42: schema della struttura dati corsi → lezioni tramite dizionari

Nella sezione seguente vi sono alcuni esempi di codice -completi di commento- estratti dal progetto Lode4iPhone, che fanno uso di queste strutture dati di tipo dizionario.

```
// funzione che restituisce un puntatore ad un dizionario contenente i
// corsi e lezioni scaricate dal server, leggendolo dal file "mydict"
// della cartella "cache", se esistente (la "de-serializzazione")

- (NSMutableDictionary *) loadDictFromFile
{
    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSCachesDirectory,
    NSUserDomainMask, YES);
    NSString *dataPath = [[paths objectAtIndex:0]
    stringByAppendingPathComponent:@"mydict"];
    NSMutableDictionary *td;
    if ([NSFileManager defaultManager] fileExistsAtPath:dataPath) {
        td = [NSMutableDictionary alloc] initWithContentsOfFile:dataPath;
    }
    return td;
}

// funzione con due parametri di input: un codice corso ed un puntatore ad
// un dizionario, e restituisce il titolo del corso
// corrispondente al codice fornito, o null se non esiste nel dizionario

- (NSString *) findTitleforCourseID:(NSString *) courseID onDictionary:
(NSMutableDictionary *) dictionary
{

```

```
NSString *retval;
NSMutableDictionary *item = [dictionary valueForKey:courseID];
if (item == nil) {
    retval = NULL;
}
else {
    retval = [item valueForKey:@"titoloc"];
}
return retval;
}

// funzione richiamata quando l'utente chiede di cancellare una lezione
// (precedentemente scaricata dal server) dal disco locale. I parametri di
// input sono rispettivamente il codice della lezione e quello del corso
// il dizionario utilizzato è existentCourseLectDict (una variabile
// globale)

-(void) removeLecture: (NSString *) idl onCourseID: (NSString *) idc
{
    // ricavo il puntatore all'elemento dizionario della lezione
    // da cancellare
    NSMutableDictionary *lecToDel = [self findDictforLessonID:idl
    onCourseID:idc onDictionary:existentCourseLectDict];

    // ricavo dal dizionario la cartella del disco locale dove si trovano
    // i files della lezione (filmato, file xml e immagini slides)
    NSString *folderl = [lecToDel valueForKey:@"folderl"];
    NSString *baselectpath = [
    [NSSearchPathForDirectoriesInDomains(NSCachesDirectory,
    NSUserDomainMask, YES) objectAtIndex:0]
    stringByAppendingPathComponent:folderl];

    // rimuovo i files .mp4 .xml e .jpg della lezione
    [[NSFileManager defaultManager] removeItemAtPath:baselectpath
    error:NULL];

    // rimuovo dal Dizionario i dati della lezione
    [[existentCourseLectDict objectForKey:idc] objectForKey:@"lessons"]
    removeObjectForKey:idl];

    // se il corso non contiene altre lezioni oltre a quella eliminata,
    // allora rimuove anche il corso dal dizionario stesso
    if ([[existentCourseLectDict objectForKey:idc]
    objectForKey:@"lessons"] count] == 0) {
        [existentCourseLectDict removeObjectForKey:idc];
    }
    lecToDel = nil;
    folderl = nil;
    baselectpath = nil;
}
```

### 6.4.3 Classi ed oggetti per la connessione in rete

La connessione in rete è fondamentale per il funzionamento di Lode4iPhone: infatti, sia la fase di ricerca di una videolezione scaricabile dal server (quello indicato nelle impostazioni

dell'applicazione), sia poi lo stesso download del filmato e degli altri files che compongono la lezione, avvengono attraverso il collegamento a risorse di rete con protocollo http (HyperText Transfer Protocol).

Come per il parsing di documenti xml (vedi capitolo 5.2), anche l'apertura di connessioni di rete e il download di files si appoggiano al protocollo "delegation". In questo modo, quando si avvia lo scaricamento di una risorsa remota, l'oggetto delegato viene invocato quando uno degli eventi previsti dal protocollo si verifica (ad esempio, download terminato, errore di rete ecc.) Il disegno dell'applicazione secondo questi canoni permette sia di sviluppare programmi che aderiscono a metodologie standard nella produzione del codice, sia di distinguere in modo chiaro il ruolo degli attori (gli oggetti, le classi) in gioco.

Per esempio, in Lode4iPhone il metodo `loadRemoteCourseList`, il cui codice è riportato qui di seguito, si occupa di scaricare il file `COURSES.XML` dal server per ottenere (dopo averne effettuato il parsing) l'elenco dei corsi disponibili per lo scaricamento di videolezioni. Come vediamo, viene creata un'istanza della classe `LodeConnection` (che si occupa di aprire la connessione tramite la classe `NSURLConnection` fornita dall'sdk) il cui "delegate" è la stessa classe chiamante (in questo caso è l'istanza corrente di `InitialViewController`) che dovrà implementare quindi i due metodi obbligatori del protocollo, e cioè `connectionDidFail` e `connectionDidFinish`.

```
- (void) loadRemoteCourseList {
    [currentPickerArray removeAllObjects];
    NSString *fullserverurl = [self.serverurl
stringByAppendingString:@" /COURSES.XML"];
    NSURL *rurl = [NSURL URLWithString:fullserverurl];
    [self setCurrentIDLez:NULL];
    [self setCurrentIDCourse:NULL];
    LodeConnection *lc = [[LodeConnection alloc] initWithURL:rurl
delegate: self];
    [self setMyconnection:lc];
    [[[self myconnection] connection] start];
    [lc release];
    fullserverurl = nil;
    rurl = nil;
}
.....

.....
- (void) connectionDidFail:(LodeConnection *)theConnection
{
    LodeAlertWithMessage(@"è fallita la connessione");
    [dic dismissWithClickedButtonIndex:1 animated:YES];
}
```

```
- (void) connectionDidFinish: (LodeConnection *)theConnection
.....// avvia le azioni conseguenti al download del file
.....
```

### **6.5 Difficoltà incontrate nello sviluppo del software**

Durante le fasi di implementazione e test di Lode4iPhone si sono verificati alcuni inconvenienti che hanno causato rallentamenti nel portare a termine il lavoro. Tali inconvenienti si possono riassumere in:

- presenza di alcuni bugs nel framework MediaPlayer: lo sviluppo dell'applicazione è iniziato utilizzando la versione 4.0 (prima in beta, poi ufficiale) dell'sdk, salvo poi concludersi con la versione 4.1 che nel frattempo è stata rilasciata da Apple. Il cambio di release fortunatamente ha risolto buona parte dei bugs che altrimenti avrebbero rappresentato un deficit importante per l'affidabilità complessiva del software. L'errore più importante che al momento non è ancora stato corretto si verifica durante il passaggio da una qualsiasi view (es. Slide view) verso la vista "Video" che ospita il riquadro dove scorre il filmato, passaggio dopo il quale viene introdotto uno sfasamento di circa un secondo del filmato stesso. Per correggere quest'errore, l'applicazione procede a modificare di circa un secondo la proprietà "currentTime" ogni volta che la vista converge, appunto, su "Video view". Quest'intervento, pur risolvendo il problema, causa un leggero "scatto" nell'ascolto della traccia audio
- utilizzo di iPhone simulator: il simulatore è stato utilissimo per il test dell'applicazione sia durante tutte le fasi intermedie dello sviluppo, sia per le verifiche sulla versione finale; l'eccezione è rappresentata dagli aspetti legati alla gestione della memoria ram: infatti, la quantità di memoria occupata durante il runtime varia abbastanza sensibilmente tra la piattaforma simulata e quella reale (un dispositivo iPhone)
- memory leaks: lo strumento Instruments, fornito nel kit dell'sdk, consente di individuare i "memory leaks", cioè allocazioni di oggetti in memoria che non sono



più raggiungibili da alcun puntatore e dunque costituiscono un “garbage” che occupa inutilmente una parte di memoria ram, che nel caso dei dispositivi di tipo mobile non è abbondante. Instruments mette in evidenza anche tutte le operazioni di allocazione di memoria, e con un click del mouse si può raggiungere il punto esatto del codice dove si trova l'istruzione che esegue tale allocazione. L'esperienza di utilizzo di questo tool suggerisce che sicuramente si tratta di un valido supporto, anche se in alcune situazioni ha fornito risultati difficili da interpretare; in particolare, un paio di segnalazioni di memory leaks sono pervenute su istruzioni che (realisticamente) non possono essere causa di questo errore. Probabilmente si tratta in questi casi di errori generati da classi dello stesso framework di sistema (le classi core dell'sdk).

## **6.6 Osservazioni finali su Lode4iPhone**

A conclusione di questo capitolo possiamo dire che lo sviluppo del software Lode4iPhone si è concluso positivamente, ed il sistema ha raggiunto gli obiettivi che inizialmente erano stati posti. Sul lato server, come è stato spiegato, è stata realizzata l'integrazione tra iTunesU ed i files necessari al funzionamento di Lode4iPhone in fase di elencazione e scaricamento delle risorse (corsi e lezioni), e quindi i filmati delle videolezioni non debbono essere pubblicati separatamente ma sono fisicamente lo stesso file condiviso tra i due sistemi. Vale la pena di ricordare inoltre che il canale iTunesU può essere utilizzato direttamente anche dall'iPhone, attraverso il software iTunes che fa parte della dotazione software di base dello smartphone. Quindi gli utenti di iPhone interessati a queste videolezioni potrebbero scegliere tra l'accesso a iTunesU e l'installazione di una “app” (che poi è Lode4iPhone) per avere una migliore interfaccia di supporto alla lezione. Questa scelta dipende probabilmente anche dal settore scientifico e dai contenuti di ogni specifico oggetto, e di conseguenza dell'importanza (o meno) che assumono le slides di supporto o la possibilità di muoversi tra i contenuti della lezione con un accesso “per argomento”. Si è usato il condizionale in quanto, al momento, Lode4iPhone non è stata ancora stata proposta ad Apple per l'approvazione e per la pubblicazione su App store.

Il software naturalmente può essere ulteriormente integrato e migliorato. Ad esempio si potrebbero introdurre:

- un miglioramento nella funzionalità dell'oggetto "picker", usato in più punti del programma. Attualmente, soprattutto quando ci sono più di cinque/sei elementi tra cui operare la scelta, il funzionamento dell'oggetto non è ottimale. Si tratta in realtà di un limite della classe offerta dal framework di base dell'sdk, ma forse lavorando sulla gestione degli eventi e delle "gestures" (azioni quali il tocco, il multitouch ecc.) si potrebbe ottenere un risultato migliore
- la possibilità di importare le lezioni da un dispositivo locale (es. pendrive usb, computer) anziché da un server remoto. I 15-20 minuti necessari in media per il download di una lezione (periodo nel quale iPhone non può essere utilizzato per altre funzioni) può essere fastidioso
- gestione del multitasking: questa caratteristica, introdotta con l'ultima release di iOS, potrebbe essere utile ad esempio per limitare il disagio del download di cui si è parlato nel punto precedente, ma anche per gestire situazioni quali l'arrivo di una telefonata
- la possibilità di attivare un sistema di abbonamento a feeds rss, per scaricare automaticamente le nuove lezioni dei corsi per i quali è stata attivata la sottoscrizione. Questa nuova caratteristica diventerebbe molto utile soprattutto nel caso in cui, in un sistema "in produzione", il numero complessivo dei corsi e delle lezioni diventasse molto elevato.

## **Conclusioni**

Questo capitolo conclusivo sarà dedicato alla sintesi del lavoro svolto in questo elaborato finale, dalle premesse ai diversi temi correlati a questo lavoro e ad una valutazione dei risultati ottenuti. Infine sarà dedicato un breve spazio per delineare le prospettive in termini di sviluppi possibili.

Come si è avuto modo di approfondire nel primo capitolo, le videolezioni sono uno strumento di supporto alla didattica importante ed in linea con la tendenza di utilizzo delle tecnologie soprattutto da parte delle nuove generazioni. Sebbene nella comunità scientifica il dibattito sia ancora ampio circa la reale efficacia di un'unità didattica somministrata (o ricevuta) attraverso il mezzo informatico, la specificità delle videolezioni asincrone ha aperto in realtà, come si è avuto modo di descrivere in dettaglio, nuovi scenari di utilizzo, delineando un contesto nel quale il processo di apprendimento può giovare contemporaneamente di molti canali: le lezioni frontali tradizionali, le piattaforme software di tipo LMS (Learning Management Systems) per la pubblicazione di materiali di supporto e per l'attivazione di strumenti di servizio quali forum e bacheche elettroniche, e, appunto le videolezioni non solo e non sempre in sostituzione delle lezioni frontali, ma ad esempio per approfondimenti e ripasso di parti specifiche, o per rimediare ad eventuali assenze. Proprio quest'ultima idea ha suggerito la creazione di uno strumento software innovativo quale il sistema Lode, descritto nel secondo capitolo, che da un lato fornisce supporto alle fasi di registrazione di una lezione frontale, e dall'altro genera unità didattiche dove l'interfaccia è di tipo più avanzato rispetto ad un normale player video, e permette di visualizzare filmato e slides, di ascoltare la voce del docente, e di navigare efficacemente ("per argomento") tra i contenuti della lezione.

A completamento dell'idea che le videolezioni asincrone non sono solo un "surrogato" per chi non può partecipare alle lezioni in presenza, è utile ricordare la già citata spiegazione fornita dal MIT di Boston all'obiezione posta da chi osservava che la pubblicazione dell'intero archivio di audio e videolezioni su iTunesU in forma gratuita pone interrogativi sul

perchè pagare 26.000\$ all'anno di tasse d'iscrizione a questa università: Il MIT rispose che il valore di iscriversi e frequentare il MIT sta soprattutto nell'esperienza sociale e ambientale del lavoro all'interno delle classi e dei laboratori. Alla luce del ragionamento sopra introdotto, quest'affermazione non è in contrasto con lo sviluppo delle videolezioni asincrone, che peraltro lo stesso MIT continua a sostenere.

Il fatto che Lode sia un sistema che, come visto, consente di realizzare videolezioni con un costo davvero contenuto, è un elemento certamente molto importante. Infatti, la scarsa diffusione di soluzioni complete di videolezioni di tipo sincrono è dovuta probabilmente ai costi eccessivi delle piattaforme che offrono le necessarie garanzie di efficacia, oltre che alle difficoltà tecniche di utilizzo di questi sistemi di videoconferenza, che proiettati sulle dimensioni dell'offerta formativa di una Facoltà anche di medie dimensioni, fa capire quali ostacoli incontrerebbe un uso intensivo di tale soluzione.

A parere di chi scrive, Lode offre anche un altro vantaggio: il fatto di richiedere ad un docente poco lavoro aggiuntivo rispetto a quello svolto per la preparazione delle normali lezioni frontali; diversamente, alcuni docenti potrebbero essere portati a pensare che il risultato (le videolezioni) non giustifichino uno sforzo aggiuntivo troppo importante rispetto alla già complessa attività di insegnamento.

La popolarità raggiunta (e in continuo aumento) dal canale iTunesU, altro argomento di cui si è parlato nel corso di questo elaborato finale, è un altro indice di come diversi soggetti del mondo universitario stiano investendo energie per produrre questo tipo di supporti alla didattica. Lo sviluppo del "plugin" software, descritto nel quarto capitolo, è nato dall'idea di fornire una vetrina importante alle risorse prodotte con Lode, e anche per costruire una base per il funzionamento dell'altro software sviluppato: Lode4iPhone.

Per quanto riguarda il dispositivo iPhone e le possibilità offerte dagli strumenti che fanno parte del Software Development Kit distribuito da Apple, il giudizio è senz'altro positivo: è sorprendente il livello qualitativo delle applicazioni che si possono ottenere, tanto che dopo le difficoltà iniziali di apprendimento (comuni ogni volta che si affronta un nuovo linguaggio

di programmazione ed un nuovo ambiente di sviluppo) non si avverte alcun vincolo significativo nel progettare e scrivere codice per uno smartphone, fatta eccezione per le dimensioni limitate del display che obbligano ad un'attenta progettazione dell'interfaccia utente. Altro punto a favore è la notevole disponibilità di documentazione, sia sul sito di Apple sia da molte altre fonti: si trovano facilmente le descrizioni delle varie API, esempi di codice, forum a tema ecc. I punti critici che sono stati individuati si possono invece così riassumere:

- una procedura troppo complessa (ed in parte discrezionale) per ottenere la pubblicazione del prodotto su App store: se da un lato questo può essere una garanzia di qualità e sicurezza del software che gli utenti installano sui propri dispositivi, dall'altro appesantisce l'intero processo di produzione dei programmi, e questo aspetto è oggetto di critica da parte della comunità di sviluppatori;
- autonomia della batteria: i consumi energetici sono sempre stati un tallone d'Achille per l'iPhone. Durante il playback di un filmato la batteria esaurisce abbastanza velocemente la sua carica, e questo può essere un limite in quanto applicazioni come Lode4iPhone sono pensate soprattutto per un utilizzo "fuori ufficio", dove la possibilità di ricaricare il dispositivo può non essere a portata di mano;
- alcuni frameworks che fanno parte della distribuzione ufficiale, ad esempio "MediaPlayer", non sono esenti da errori interni, anche se l'avanzamento delle release sembra progressivamente porre rimedio a questo problema: in questo senso la concorrenza della piattaforma Android e di altre che si stanno affacciando in questo segmento di mercato dovrebbero stimolare un ulteriore miglioramento;
- le scarse possibilità di accedere a funzioni di basso livello per controllare meglio il funzionamento del dispositivo: le API ufficiali offrono in generale un set di istruzioni soddisfacente per le esigenze più comuni, ma in certi casi, come ad esempio la gestione di risorse multimediali, si sente l'esigenza di avere un controllo ancora più approfondito e a basso livello degli oggetti utilizzati. Questo purtroppo non è possibile, pena il rifiuto di Apple di includere il prodotto nell'App store.

Per quanto riguarda i due software che sono stati sviluppati nel corso di questo elaborato finale, è utile tracciare un bilancio dei risultati ottenuti. Ricordiamo che si tratta di:

- un'estensione di Lode per convertire le videolezioni in un formato compatibile con il canale iTunesU
- Lode4iPhone, un porting su piattaforma mobile – ed in particolare iPhone – delle lezioni registrate con il sistema Lode

Innanzitutto va ricordato che questi due moduli software sono tra loro fortemente interconnessi: infatti, non solo hanno lo stesso punto di partenza (il sistema Lode), ma anche i rispettivi prodotti convergono verso un'idea di diffusione integrata delle videolezioni verso più destinatari, che nel caso specifico sono gli utenti di iTunesU e quelli (ancora potenziali) di Lode4iPhone. È importante sottolineare che, il fatto che questi due canali di distribuzione condividano fisicamente una parte della struttura dati lato server (su tutto, i files video in formato mp4) rappresenta un elemento di economicità e rapidità di esecuzione molto importante per dare una prospettiva di sviluppo concreta di questi strumenti all'interno delle realtà universitarie.

Entrando nello specifico dei due programmi, si può dire che il primo – il meno complesso – ha richiesto uno sforzo di messa a punto dei parametri di configurazione della fase di conversione dei files video, dovuto ad errori di sincronizzazione tra le tracce audio e video in realtà piuttosto comuni in questo genere di applicazioni. Gli obiettivi posti sono stati comunque raggiunti, e in effetti buona parte del contenuto del canale iTunesU dell'Università di Trento è stato prodotto convertendo le videolezioni registrate presso la Facoltà di Scienze Matematiche, Fisiche e Naturali proprio attraverso Lode.

Il secondo software, certamente più complesso ed impegnativo, ha raggiunto anche in questo caso gli obiettivi che erano stati definiti in fase di pianificazione: in primo luogo ottenere su un dispositivo mobile come lo smartphone iPhone un'esperienza di utilizzo di una videolezione asincrona simile a quella offerta dal sistema Lode, cioè con elementi di duttilità dell'interfaccia utente che permettono di vedere il video, ascoltare l'audio, vedere le slides ed ingrandirle nei dettagli, fermare o riavviare in qualsiasi momento la riproduzione e cercare punti specifici della lezione con un accesso “per argomenti” che avviene attraverso

la selezione dei titoli delle slides. Oltre a quest'obiettivo, si può considerare raggiunto anche quello di fornire un sistema efficiente per cercare e scaricare le lezioni, attraverso la rete, da un server opportunamente configurato: come mostrato nel capitolo sei, quest'operazione avviene in modo semplice ed immediato per l'utente, anche se andrebbero introdotti dei miglioramenti nel caso in cui il numero di lezioni disponibili sul server dovesse diventare quantitativamente rilevante. Ricordiamo che, quando l'utente scarica una lezione sul dispositivo, questa può essere poi riprodotta in locale senza la necessità di alcun collegamento in rete.

Una videolezione riprodotta tramite Lode4iPhone è un'esperienza che, seppur non priva ovviamente dei limiti imposti da questo tipo di risorsa (su tutti il senso di "isolamento" dello studente) offre a parere di chi scrive un supporto efficace (grazie soprattutto alla possibilità di vedere le slides) ed anche sufficientemente dinamico: la possibilità di cambiare il "punto di vista" di una lezione riduce il rischio del calo di attenzione che si verifica invece più facilmente quando la lezione è erogata attraverso oggetti quali filmati, screencast e – ancor di più – podcasts audio.

Per quanto riguarda le prospettive di sviluppo del lavoro svolto, queste certamente non mancano. Partendo da alcune premesse:

- da un lato la disponibilità presso gli utenti -e dunque anche presso gli studenti ed i docenti universitari- di dispositivi mobili intelligenti come gli smartphones è in forte aumento (+50% nell'ultimo anno)
- gli investimenti nella produzione di videolezioni asincrone sono in costante aumento, visto anche il successo mondiale del canale iTunesU
- se facciamo riferimento nello specifico alla realtà dell'Università di Trento, ma la considerazione probabilmente potrebbe essere estesa ad altre realtà, le aule didattiche sono dotate sempre più frequentemente di dispositivi efficienti di tipo audiovisivo (microfoni, videocamere, videoproiettori ecc.) che semplificano molto il processo di produzione di una videolezione, e ne diminuiscono ulteriormente il costo,

che, come si è avuto modo di valutare, è già di per sé molto basso anche grazie a sistemi come Lode.

Da queste considerazioni si può dedurre come ci sia spazio per progettare un uso più pervasivo delle videolezioni come strumento di supporto alla didattica, a patto che questo obiettivo sia compatibile con alcuni ostacoli già evidenziati nei capitoli precedenti: problemi con i diritti d'autore, resistenze di una parte dei docenti a registrare le proprie lezioni, dubbi sulla reale efficacia di questo strumento.

Posto che l'obiettivo di incrementare in modo significativo l'uso delle videolezioni deve passare eventualmente attraverso un progetto complessivo e condiviso da parte degli attori in gioco, che deve affrontare sia aspetti tecnici che organizzativi (in primis il fattore umano in tutte le sue declinazioni), si possono fin d'ora individuare alcune linee di sviluppo:

- richiedere ad Apple la pubblicazione di Lode4iPhone su App store, correggendo l'applicazione nel caso in cui ci fossero parti specifiche non compatibili con i requisiti indicati
- sviluppare moduli software per automatizzare la pubblicazione di audio/videolezioni su iTunesU attraverso le interfacce di accesso con web services offerte dal canale di Apple. Questa automazione, che è vincolata alla presenza di una struttura consolidata delle diverse "sezioni" all'interno del canale iTunesU, consentirebbe un forte risparmio di tempo nelle fasi di pubblicazione, limitando l'intervento degli operatori umani per la loro gestione;
- introdurre un sistema di protezione dei contenuti iTunesU, agganciato ai sistemi di autenticazione già in uso presso l'Ateneo, per definire, se necessario, aree ad accesso riservato dove gli utenti (studenti, personale) possano accedere utilizzando le stesse credenziali di accesso utilizzate per i sistemi informativi locali;
- favorire la produzione di videolezioni attraverso specifici progetti all'interno delle Facoltà: si tratta di curare aspetti tecnici ed organizzativi in modo da allestire ambienti e competenze che rendano la registrazione di unità didattiche un fatto normale. In questo contesto può essere interessante cercare ulteriori forme di



automazione per limitare la necessità di intervento da parte degli operatori: per esempio, siccome molte attrezzature di videoregistrazione sono programmabili, si potrebbero costruire interfacce di programmazione per schedulare opportunamente le fasi di registrazione, pur considerando che in questo modo vengono a mancare le funzioni di regia e gestione delle videocamere che rendono migliore la qualità del filmato. Un'altra possibilità interessante è quella di progettare una nuova versione di Lode che sia in grado di funzionare, anche in questo caso, senza l'intervento di un operatore: questo obiettivo è teoricamente possibile, a patto di trovare un modo per intercettare e registrare l'avanzamento delle slides direttamente sul computer utilizzato dal docente;

- progettare e realizzare forme di integrazione tra le piattaforme software di LMS (Learning Management System) quali ad esempio Moodle, e le risorse multimediali, come le videolezioni, con particolare riferimento al canale iTunesU;
- realizzare una versione di Lode4iPhone che funzioni su dispositivi dotati del sistema Android, che, come abbiamo visto, è attualmente più diffuso di iOS, il sistema di iPhone. In prospettiva vale la pena di valutare anche il porting verso altre piattaforme, come Windows Phone, che si stanno affacciando su questo mercato;
- se dovesse verificarsi un'espansione del mercato dei dispositivi "iPad-like", come lo stesso Apple iPad oppure il Samsung Galaxy tab per citare gli esempi più noti, cioè dispositivi che aggiungono alla portabilità uno schermo di dimensioni maggiori (7-10") rispetto agli smartphones, si dovrebbe valutare anche in questo caso un porting di Lode4iPhone verso questi nuovi dispositivi.

Tutte le informazioni raccolte per la realizzazione di questo elaborato finale indicano che i contesti di riferimento in cui si è operato (multimedialità e dispositivi mobili in particolare) sono in forte espansione. È auspicabile quindi che il contributo fornito da questo lavoro aiuti allo sviluppo di nuove idee e nuovi progetti per attivare forme di diffusione della conoscenza più innovative, sia negli ambiti tradizionali, sia nei settori oggi spesso inesplorati che ricadono nella definizione generica di "long-life learning".



## Bibliografia

- (1) Ronchetti, M. (2008) - Requirements for videolectures: which system is the best for you?
- (2) Ronchetti, M., Stevovic J. (2008) - Extending the podcasting approach: *lectures on the phone*
- (3) Ronchetti, M. (2010) Perspectives of the application of video streaming to education
- (4) Pullen, J.M. (2000) "The Internet-Based Lecture: Converging Teaching and Technology"
- (5) Zhang, D, L. Zhou, B. Briggs, & J. F. Nunamaker (2006). Instructional Video In E-learning: Assessing the Impact of Interactive Video on Learning Effectiveness
- (6) Ronchetti M., "Has the time come for using video-based lectures over the Internet? A Test-case report"
- (7) Friedland G., Rojas R. (2008) - Anthropocentric video segmentation for lecture webcasts
- (8) Abelson, H. (2008) The creation of OpenCourseWare at MIT
- (9) Baecker, R. Moore, G. Zijdemans, A. (2003) Reinventing the Lecture: Webcasting Made Interactive
- (10) Barger D., Gupta A., Grudin J, & Sanocki E. (1999) Annotations for streaming video on the web
- (11) Bennett E. & Maniar N.. (2008) Are videoed lectures an effective teaching tool?
- (12) Berner E.S. & Adams B. (2004) Added value of video compared to audio lectures for distance learning
- (13) Bianchi, M. (1998) AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations
- (14) Bousdira, N., Myers, E., Neal, H.A., Severance C., Storr M. & Vitaglione G. (2001) WLAP: The Web Lecture Archive Project

- (15) Brown B.W. & Liedholm C.E.L. (2004) Student Preferences in Using Online Learning Resources
- (16) Canessa E., Fonda C. & Zennaro M, (2008) Academic Webcasting using the Automated EyA Recording System
- (17) Chen H.-Y., Chen G.-Y. & Hong J.S. (1999) Design of a Web-based synchronized multimedia lecture system for distance education
- (18) Chen H.-Y., Chen G.-Y. & Hong J.S. (1999) Design of a Web-based synchronized multimedia lecture system for distance education
- (19) Chen Y. & Heng W.J. (2003) Automatic synchronization of speech transcript and slides in presentation
- (20) Chiu, C., Lee, G., & Yang, J. (2006). A comparative study of post-class lecture video viewing
- (21) Correia N. & Cabral D. (2005) VideoStore: A system to store, annotate and share video based content. Recent Research Developments in Learning Technologies
- (22) Demetriadis, S., & Pombortsis, A. (2007). e-Lectures for Flexible Learning: a Study on their Learning Efficiency
- (23) Foertsch J., Moses G., Strickwerda J. & Litzkov M (2002) Reversing the lecture/homework paradigm using eTEACH web-based streaming video software
- (24) Fujii, A, Itou, K, & Ishikawa, T. (2007) LODEM: A system for on-demand video lectures
- (25) Gigonzac, G., Pitie, F. & Kokaram.,A. (2007) Electronic slide matching and enhancement of a lecture video
- (26) Hürst W., Weite M. & Jung S. (2007) An evaluation of the mobile usage of e-lecture podcasts
- (27) Lauer T., Müller R. & Trahasch S. (2004) Learning with lecture recordings: key issues for end- users
- (28) Mertens R., Ketterl M., Vornberger O. (2006) Interactive Content Overviews for Lecture Recordings

- (29) Repp S. & Meinel C. (2008) Segmentation of Lecture Videos Based on Spontaneous Speech Recognition
- (30) Soong S.K.A, Chan L.K. & Cheers C. (2006) Impact of video recorded lectures among students
- (31) Wald. M. (2005) 'SpeechText': Enhancing Learning and Teaching by Using Automatic Speech Recognition to Create Accessible Synchronised Multimedia
- (32) Zupancic, B. & Horz, H. 2002. Lecture recording and its use in a traditional university course
- (33) Iacubino, A. (2009) – Creare applicazioni di successo per iPhone e iPad
- (34) Apple web site - iTunesU Administration Guide
- (35) App store review guidelines -  
<https://developer.apple.com/appstore/resources/approval/guidelines.html>
- (36) Memory management programming guide -  
<http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/MemoryMgmt/MemoryMgmt.html>
- (37) The Objective-C Programming Language -  
<http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
- (38) articoli vari del sito web <http://www.devapp.it>
- (39) Steele, J.; To, N. (2010) - *The Android Developer's Cookbook: Building Applications with the Android SDK*