



Ariel.Net: la piattaforma per il CdL online in Sicurezza dei Sistemi e delle Reti Informatiche

Mirko Bove e Daniela Scaccia

CTU - Centro di servizio per le Tecnologie e la didattica Universitaria
multimediale e a distanza Università degli Studi di Milano

bove@ctu.unimi.it; scaccia@ctu.unimi.it

Abstract

The Ctu – eLearning Center of the University of Milan, in partnership with Isvor Knowledge System S.p.a., for the academic year 2004 – 2005, has delivered its first University Degree course in dual mode: this means it been presented both in F-2-F conventional classroom and as blended learning. It's a 3 year Degree Course about digital security, entitled «Sicurezza dei sistemi e delle reti informatiche - SSRI online», Polo didattico e di ricerca di Crema, Web address: <http://www.cdlonline.unimi.it>. Ariel.Net, the software platform devoted to SSRI online represents the natural evolution of Ariel, the current platform for the Unimi eLearning, and has been planned and realized by a Ctu team of instructional designers, developers and software engineers, taking into account the specific needs of a University Course. This paper focuses on how the design and didactic requirements of the SSRI online deeply influenced and shaped the software architecture and features of Ariel.Net.

1. Il progetto didattico

La progettazione di SSRI online, condotta sulla base dell'analisi di caratteristiche, aspettative ed esigenze degli utenti potenzialmente interessati a un corso universitario, erogato in modalità online, è stata incentrata sul concetto di «flessibilità dell'offerta formativa» estendendone le applicazioni, non solo rispetto alle questioni spazio-temporali, ma anche ai contenuti, alle modalità di erogazione, agli approcci didattici e alle risorse disponibili.

Avendo identificato nello studente distante dalla sede universitaria e/o lavoratore il target di utente potenziale di SSRI online, ci si è indirizzati verso un modello di apprendimento *blended* (ovvero con l'80% del percorso didattico da frequentare online e il rimanente 20% da spendere in aula tra sessioni d'esame e approfondimenti con docenti e tutor) che riducesse al minimo i momenti da svolgersi in presenza e, allo stesso tempo, non eccedesse con le attività online sincrone, imponendo agli utenti tempi e orari di collegamento.

In considerazione della natura fortemente teorica degli insegnamenti del 1° anno si è quindi deciso di fare riferimento al modello di corso online che Mason (1998) definisce *content and support*, modello specificatamente adatto al trasferimento di conoscenze e incentrato sulla realizzazione e distribuzione di materiali didattici strutturati, di solito multimediali, sui quali il discente lavora in autonomia, sia nella fase di studio e apprendimento, sia in quella di valutazione. Si è quindi optato per un percorso formativo in «autoapprendimento», caratterizzato da una pianificazione predefinita delle attività da svolgere e incentrato sul singolo utente.

Da qui, la riorganizzazione del calendario didattico da semestrale a quadrimestrale, la pianificazione di due prove intermedie di valutazione per ogni insegnamento, la scelta di effettuare appelli d'esame in sessioni riservate agli studenti online e l'organizzazione di sessioni d'aula, destinate allo svolgimento di lezioni relative ad almeno tre insegnamenti, durante i fine settimana nei giorni di venerdì e sabato.

Analogamente, rispetto a contenuti e modalità di erogazione, è stato offerto allo studente di SSRI online un percorso formativo flessibile e articolato che permettesse, mediante l'alternanza di momenti a carattere informativo e momenti esercitativi, di acquisire una metodologia di studio incentrata sull'applicazione pratica di assunti teorici. Alla destrutturazione dei singoli Insegnamenti in Moduli, Unità Didattiche, Lezioni e Attività è quindi seguita una fase di microprogettazione che ha permesso di identificare per ogni attività la componente multimediale (audio/videolezione, dispensa, test di autovalutazione, esercizio, etc.) più adatta allo sviluppo dei contenuti.

Successivamente sono state prese in considerazione le particolari esigenze dell'utente impegnato in un percorso di apprendimento online, che hanno contribuito

alla definizione di specifiche funzionali e tecniche dei sistemi di calendarizzazione, comunicazione, tracciamento e reportistica da prevedere per SSRI online.

Per i vari insegnamenti ad esempio, oltre a poter gestire informazioni quali le date di apertura e svolgimento dei momenti in presenza, i tempi di studio previsti/consigliati per ogni singolo Modulo e i tempi indicativi di svolgimento di ogni attività, è apparso utile permettere allo studente di intervenire sulla pianificazione, ampliando, traslando o contraendo i periodi indicati.

Relativamente al sistema di comunicazione tra studenti e tutor, impegnati in un percorso articolato in numerose lezioni, si è ritenuto invece di dover integrare i consueti strumenti asincroni (forum, e-mail e bacheca) con sistemi più sofisticati, volti a supportare la comunicazione contestualizzata allo svolgimento di un'attività online. Secondo la stessa logica si è rivelato inoltre utile permettere all'utente di accedere agevolmente alla reportistica dettagliata e contestualizzata delle attività effettuate, comprendente l'indicazione della data e dell'esito dello svolgimento, oltre che l'elaborato completo di ogni prova svolta.

A conclusione della fase di progettazione e sempre nel tentativo di configurare un sistema flessibile, sono stati identificati diversi profili di utenza del servizio SSRI online. Ai 120 studenti iscritti alla prima edizione sono stati affiancati otto docenti, titolari degli insegnamenti del primo anno, un tutor di processo (o mastertutor) responsabile del monitoraggio dell'intera comunità e tre tutor didattici per ogni insegnamento, responsabili del supporto sui contenuti di apprendimento. Caratteristiche ed esigenze di ogni profilo sono confluite nella progettazione di quattro ambienti online, ognuno dei quali diverso nell'interfaccia e dotato di funzionalità specifiche, volte a favorire l'attività di apprendimento o necessarie a supportare attività di monitoraggio e comunicazione.

2. La piattaforma Ariel.net

A valle di un'attenta attività di benchmarking e sulla base della mappatura delle principali esperienze di e-learning presenti sul mercato italiano, le soluzioni software messe a punto in questi anni dal CTU sono risultate idonee per supportare in modo efficace ed efficiente la realizzazione del CdL online.

Pertanto l'orientamento del gruppo di lavoro è stato quello di recuperare le competenze maturate durante lo sviluppo di Ariel, la piattaforma sviluppata dal CTU per i corsi web-enhanced d'Ateneo,¹ finalizzandole a una maggiore personalizzazione del software in ottica CdL online.

Ad oggi, Ariel.Net rappresenta l'architettura software in grado di rispondere alle esigenze progettuali e didattiche di SSRI online e, più in prospettiva, l'am-

¹ Ad oggi il numero di utenti attivi è di circa 60.000, con una media di 5.000 accessi quotidiani, e i progetti di e-learning realizzati a supporto dell'attività d'aula sono oltre 200.

biente destinato a ospitare l'offerta formativa online (master, corsi di laurea o corsi singoli) dell'Ateneo.

2.1 L'architettura software

La progettazione dell'architettura software è avvenuta in considerazione, oltre che delle specifiche e dei vincoli del progetto didattico, anche di due ulteriori ordini di fattori: da un lato la necessità di favorire la separazione in livelli dell'applicazione e, di conseguenza, la «scalabilità» della piattaforma, e dall'altro quella di nascondere e proteggere l'accesso ai dati dal livello front-end.

La logica di funzionamento dell'applicazione (fig. 1), in analogia con la modularità degli oggetti di Ariel, è realizzata mediante collezioni di classi, sviluppate utilizzando il framework Microsoft .NET 1.1. Ogni collezione descrive uno strumento o una funzionalità della piattaforma ed è a sua volta composta da classi separate per l'attivazione della struttura dell'oggetto (*structure*), per la definizione dei metodi per l'utilizzo dello strumento e l'accesso al database (*tool*) e per l'esposizione dei metodi della classe *tool* alle pagine web (*view*). Così ad esempio, secondo questo schema, lo strumento forum è composto dalla classe Forum, che provvede alla creazione dello strumento e al popolamento delle proprietà (id, titolo forum, gruppi utenti, moderatore e amministratore...), dalla classe ForumTool che fornisce i metodi per la gestione del forum e che si interfaccia direttamente al database (generazione policy di accesso, lista messaggi in un thread, invio nuovo messaggio) e dalla classe ForumView che espone alle pagine web solo le proprietà della classe *costruttore* e i metodi della classe *tool* necessari al funzionamento dell'interfaccia utente.

L'interfaccia web attiva le classi implementate nel framework ed espone all'utente le funzionalità di ogni strumento della piattaforma. L'attivazione delle classi avviene tramite il «code-behind» di pagine ASP.NET, gestite da Microsoft Internet Information Server 6.0. Il front-end è composto da quattro differenti ambienti, uno per ogni profilo utente presente nel sistema (studente, tutor, mastertutor e docente). Anche per le pagine web è stata definita una gerarchia di classi in modo da demandare a classi «padre», grazie all'ereditarietà, alcune funzionalità base come il controllo dell'autenticazione, il riconoscimento del profilo utente o il caricamento dei dati relativi alla strutturazione didattica.

La registrazione, l'aggiornamento e il mantenimento dei dati generati dalla piattaforma sono assicurati da un database relazionale, implementato con tecnologia Microsoft SQL Server 2000. Al fine di suddividere le operazioni tra i diversi server, migliorando così le prestazioni complessive dell'applicazione, la maggior parte delle operazioni di interrogazione e di aggiornamento del database vengono effettuate tramite Stored Procedure di SQL Server. Il database e l'applicazione sono stati progettati tenendo conto della necessità di assicurare una facile separazione

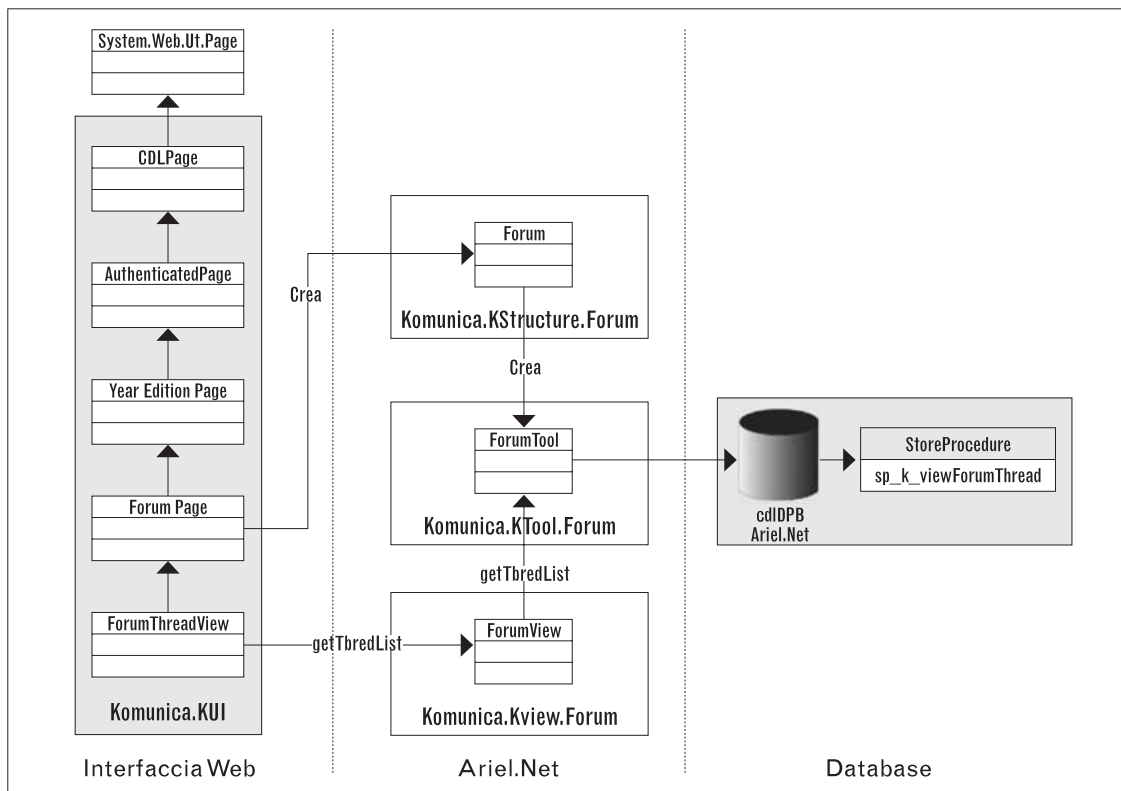


Figura 1 Architettura software.

tra fase di sviluppo e fase di produzione; tabelle di debug permettono pertanto di aggiornare o implementare nuove funzionalità nella piattaforma senza interferire con l'ambiente in uso dagli utenti.

2.2 La gestione degli utenti

L'impiego congiunto di un descrittore XML e del database ha permesso la profilazione di diverse tipologie di utenti (studente, tutor didattico, tutor di processo o mastertutor e docente) e di gestire per ogni insegnamento l'associazione tra tutor didattici e classi di studenti.

Gli utenti tramite il database sono organizzati in gruppi e gli studenti in particolare possono essere suddivisi in sottogruppi (classi). La struttura XML assicura l'associazione tra il tutor didattico di un determinato insegnamento e un sottogruppo di studenti e tra il docente e il proprio insegnamento; il mastertutor invece, dovendo monitorare l'intera popolazione degli studenti, è associato all'edizione del corso di laurea.

Il riconoscimento del profilo utente avviene durante il processo di autenticazione il quale provvede anche a inserire in una variabile di sessione i dati che

caratterizzano la tipologia di utenza: per lo studente viene generata la lista dei tutor di insegnamento associati al proprio gruppo così come per i tutor didattici e per il docente viene collegato l'insegnamento di competenza. Questo procedimento permette inoltre di destinare a ciascun profilo ambienti, funzionalità e privilegi differenti.

2.3 La struttura del CdL online

In risposta all'esigenza di mantenere per lo studente online un'organizzazione del piano di studi analoga a quella prevista per lo studente in presenza, è stata realizzata un'architettura software, speculare a quella formativa (edizioni, anni di corso, insegnamenti, moduli, unità didattiche, lezioni, attività), in grado di gestire parallelamente più edizioni del medesimo corso di laurea e di assicurare la personalizzazione e il monitoraggio del percorso di apprendimento.

Ad ogni edizione è associato un descrittore XML che definisce l'elenco degli insegnamenti e la strutturazione degli stessi in moduli in presenza e a distanza. Il medesimo descrittore viene utilizzato per generare una struttura gerarchica composta da entità ottenute istanziando le classi della piattaforma (fig. 2). La radice della struttura è la classe *YearEdition* che corrisponde ad un anno di corso dell'edizione del CdL (ad esempio 1° anno dell'edizione 2004-2005) e che istanzia un oggetto della classe *Course* per ogni insegnamento presente nell'anno. La classe *Course* fornisce le proprietà e i metodi per la gestione dell'insegnamento: titolo del corso, forum di riferimento, definizione della calendarizzazione consigliata o definita dall'utente, ecc. Inoltre, per ogni modulo in presenza o a distanza associato all'insegnamento, la classe *Course* provvede a «istanziare» un oggetto della classe *Module* che fornisce le proprietà e i metodi di base per la calendarizzazione del modulo stesso.

I moduli attualmente presenti in Ariel.Net sono di cinque differenti tipologie: modulo didattico online interno alla piattaforma, articolato in unità didattiche, lezioni e attività e descritto tramite una struttura XML (*FadModule*), modulo didattico online esterno alla piattaforma (*ExternalResourceModule*), incontro in aula (*MeetingModule*), prova in itinere in aula (*TestModule*) ed esame in aula (*ExamModule*).

Ogni tipologia di modulo è gestita da una classe specifica che eredita o estende le funzionalità base della «classe padre» *Module*, in modo da *specializzare* l'oggetto in base alle caratteristiche del modulo stesso.

Le classi per i moduli in presenza (meeting, test, exam) gestiscono le sessioni definite per l'incontro, l'iscrizione alla sessione da parte dell'utente e l'assegnazione del voto da parte dei tutor di processo (test e exam); l'iscrizione alla sessione può essere integrata nella piattaforma (meeting e test) o effettuata tramite la procedura gestita dal sistema informativo d'Ateneo (exam).

La classe *ExternalResourceModule* gestisce i dati per l'accesso alla risorsa esterna. Infine la classe *FadModule* carica il descrittore XML definito per il modulo, prelevandone l'URL dal database. L'attivazione di un modulo da parte di un utente comporta il popolamento della struttura con i dati di tracciamento provenienti dal database, specifici per ogni tipologia di modulo: se l'indicazione di una data di inizio e fine è prevista per ogni tipologia, l'iscrizione a una sessione ha significato solo per i moduli in presenza, mentre il numero di attività in corso o superate ha senso solo per i moduli FAD. L'analisi di questi dati permette alla piattaforma di tracciare lo stato di avanzamento di fruizione del modulo da parte del singolo utente.

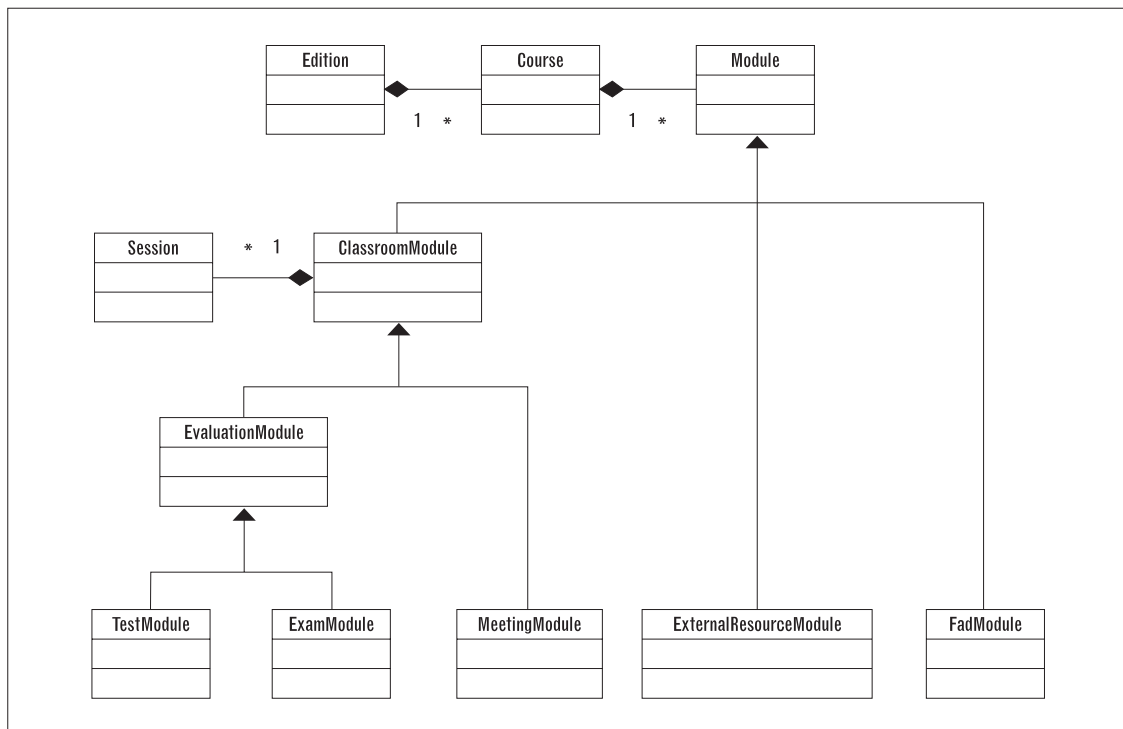
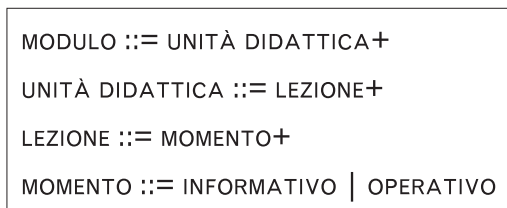


Figura 2 Struttura didattica.

Il modulo didattico online (*FadModule*) rappresenta l'entità su cui è basata l'erogazione online dei contenuti ed è descritto da una struttura XML che ne definisce l'articolazione in unità didattiche, lezioni e attività. Ogni attività viene definita come un singolo momento di lavoro, distinto a sua volta, in informativo o operativo.



La struttura gerarchica e il tracciamento dei moduli è gestita dal Komunica Assignment Engine (KAE), motore della piattaforma. Nell'ambiente KAE sono definite tre entità fondamentali: *module*, *assignment* e *activity*. Il *module* funge da collegamento con la struttura definita per l'insegnamento ed è costituito da uno o più *assignment*. L'*assignment* rappresenta un contenitore di attività o di altri *assignment*. Infine l'*activity* è l'unità singola tracciabile e rappresenta la foglia della struttura ad albero del modulo.

```

MODULE ::= ASSIGNMENT+
ASSIGNMENT ::= (ASSIGNMENT | ACTIVITY)+
ACTIVITY ::= <TERMINAL>
    
```

Mappando la struttura del CdL online con la struttura definita nell'ambiente KAE, otteniamo il seguente schema di corrispondenza:

KAE	CdL online
Module	modulo
Assignment	unità didattica, lezione
Activity	momento

La mappatura non permette la distinzione tra unità didattica e lezione che può tuttavia essere recuperata mediante il livello gerarchico delle entità.

Ad ogni singola attività è associato un descrittore XML che ne descrive le caratteristiche. Sono considerate attività i moduli multimediali, le dispense, gli

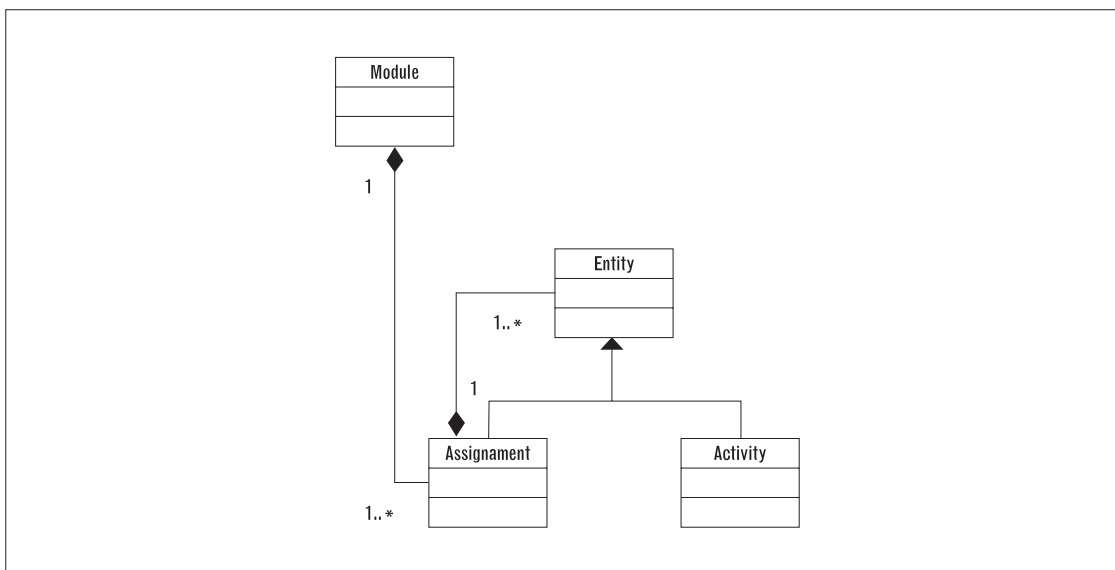


Figura 3 Struttura modulo FAD.

esercizi con correttore automatico, i test, gli esercizi con risposta aperta e così via. Un XSLT associato ad ogni macrocategoria permette la rappresentazione logica e di interfaccia dei contenuti dell'attività.

Il motore KAE impone, per ogni tipologia di attività, la presenza di tre operazioni basilari che ne descrivono il funzionamento: *print* per la rappresentazione dei contenuti, *send* per l'invio dell'attività da parte dell'utente e *view* per la visualizzazione dei risultati ottenuti nel singolo tentativo. Questa generalizzazione permette di definire ad hoc ulteriori tipologie di attività per la gestione di esercizi complessi. Ad esempio per l'insegnamento di Architetture e Reti Logiche è stato costruito un parser di espressioni logiche in modo da poter valutare automaticamente la complessità della rete e la tabella delle uscite corrispondenti all'espressione logica inserita dall'utente.

Oltre a queste operazioni, la piattaforma fornisce un metodo per l'annotazione nella struttura dei dati di tracciamento provenienti dal database. Per ogni entità sono memorizzate, ad esempio, la data di attivazione, la valutazione massima conseguita e il numero di tentativi effettuati. Per l'entità *activity* è previsto un tracciamento di livello superiore: la storia e i risultati di ogni singolo tentativo.

2.4 Le funzionalità di Ariel.Net

Oltre ai tradizionali strumenti di comunicazione, tracciamento e monitoraggio previsti dalle piattaforme comunemente in uso per l'erogazione di corsi online, Ariel.Net, rispondendo alle esigenze progettuali di SSRI online e grazie all'architettura software sopra descritta, permette di agganciare gli strumenti reciprocamente e di contestualizzarne le funzionalità rispetto alla strutturazione didattica. Nello specifico gli strumenti e le funzionalità di Ariel.Net permettono di:

- Supportare la comunicazione asincrona e sincrona uno a uno o uno a molti: all'e-mail e ai tradizionali forum, organizzati tematicamente rispetto agli Insegnamenti, si aggiungono un sistema di messaggistica privata studenti-tutor, associato alle attività didattiche (instant messaging), una «bacheca» virtuale riservata alla comunicazione, da parte del tutor di processo, di avvisi di interesse collettivo e l'eventuale utilizzo di un'aula virtuale per la videocomunicazione sincrona e la condivisione delle applicazioni (Centra).
- Gestire la calendarizzazione degli insegnamenti permettendo l'attivazione degli insegnamenti secondo la tempistica imposta dal calendario didattico: lo studente può seguire gli insegnamenti del quadrimestre in corso e di quello precedente, ma non di quello successivo.
- Intervenire sulla pianificazione delle attività online mediante lo strumento «planning» che, oltre al tradizionale calendario, utile alla visualizzazione del percorso di apprendimento «consigliato», permette la dilatazione, compressione o lo spostamento dell'attività didattica di ogni modulo.

- Gestire la fruizione asincrona di contenuti mono e multimediali (dispense e audio-videolezioni) in modalità streaming e download, con la possibilità di stampare i contenuti testuali delle lezioni.
- Supportare lo svolgimento dei momenti esercitativi, con funzionalità per test di autovalutazione, esercizi o esercitazioni avanzate a risposta chiusa con feedback automatico ed esercizi a risposta aperta con correzione del tutor o che prevedono il confronto con la soluzione del docente riproposta dal sistema.
- Supportare il processo di studio individuale mediante strumenti di annotazione virtuale da agganciare direttamente alle singole attività: i post-it per i promemoria e gli appunti personali dello studente e i suggerimenti per permettere al tutor didattico di annotare commenti e osservazioni utili a tutta la classe assegnata.
- Contestualizzare gli strumenti rispetto al livello dell'ambiente attivato (homepage, insegnamento, modulo o attività), in modo da presentare dati e funzionalità differenti.
- Supportare, per i profili tutor, il monitoraggio sul singolo studente, la condivisione e lo scambio di documenti (tipicamente elaborati di esercizi da correggere) e la moderazione della comunicazione.

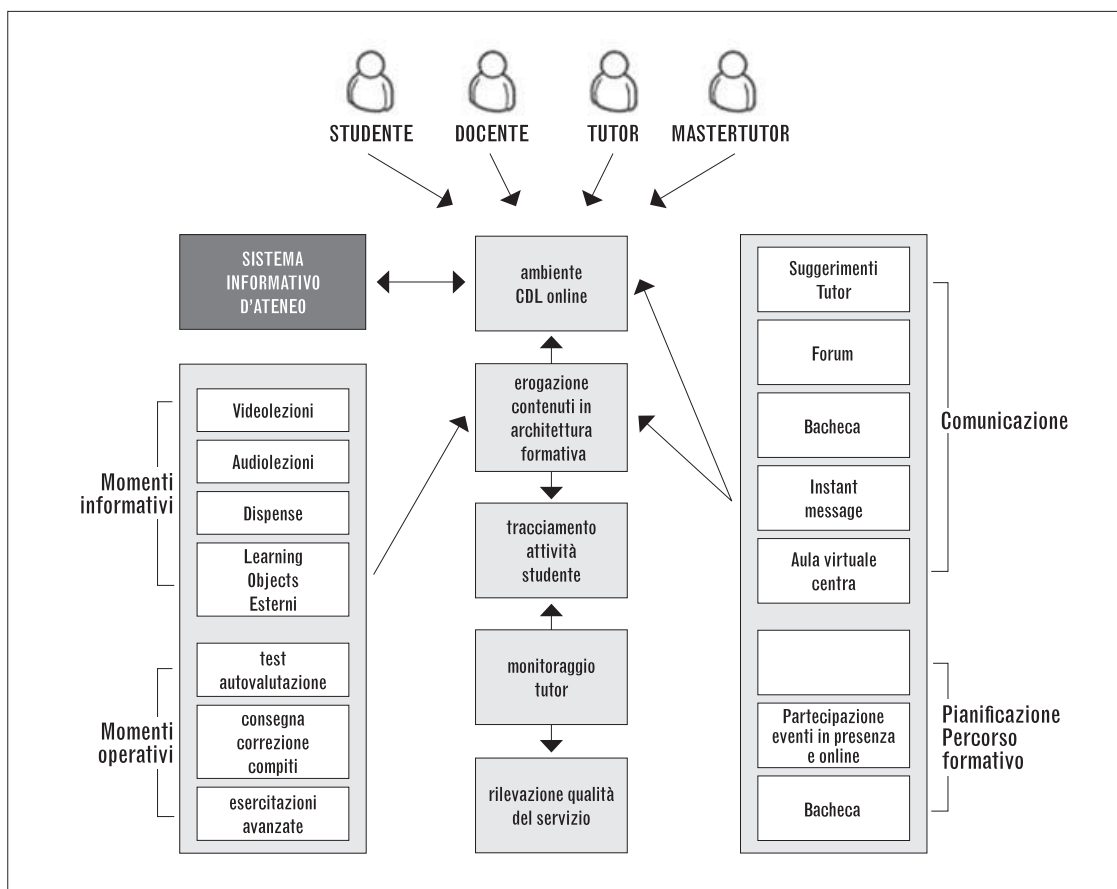


Figura 4 Funzionalità di Ariel.Net.

- Gestire e organizzare le attività in presenza mediante funzionalità quali: l'iscrizione ai momenti di verifica (prove intermedie o esami), l'attestazione della presenza o assenza e la registrazione/memorizzazione dell'esito delle prove.

2.5 L'architettura hardware

L'architettura tecnologica di Ariel.net si basa sul *modello n-tier* che prevede la suddivisione in livelli differenti per quanto riguarda i servizi di persistenza dei dati, la logica dell'applicazione e l'interfacciamento con l'utente.

Dal punto di vista hardware i tre livelli vengono distribuiti fisicamente su macchine server differenti, in modo da garantire la scalabilità del sistema. Basandosi sull'esperienza maturata con la piattaforma Ariel, si è optato per l'utilizzo di due macchine server, una per il database e una per i due livelli rimanenti.

La sicurezza delle comunicazioni tra i vari livelli e verso il mondo esterno è assicurata da un firewall che consente di separare fisicamente i segmenti di rete coinvolti definendone le regole di accesso.

La rete dati del Ctu è collegata direttamente con l'anello ottico della rete d'Ateneo.

I servizi aggiuntivi offerti nel corso di laurea online, come la fruizione in modalità streaming di contributi multimediali (videolezioni, presentazioni multimediali, slideshow, ...) e la comunicazione sincrona audio-video, vengono erogati tramite server dedicati (Helix Server 9.0 e CentraOne Symposium).

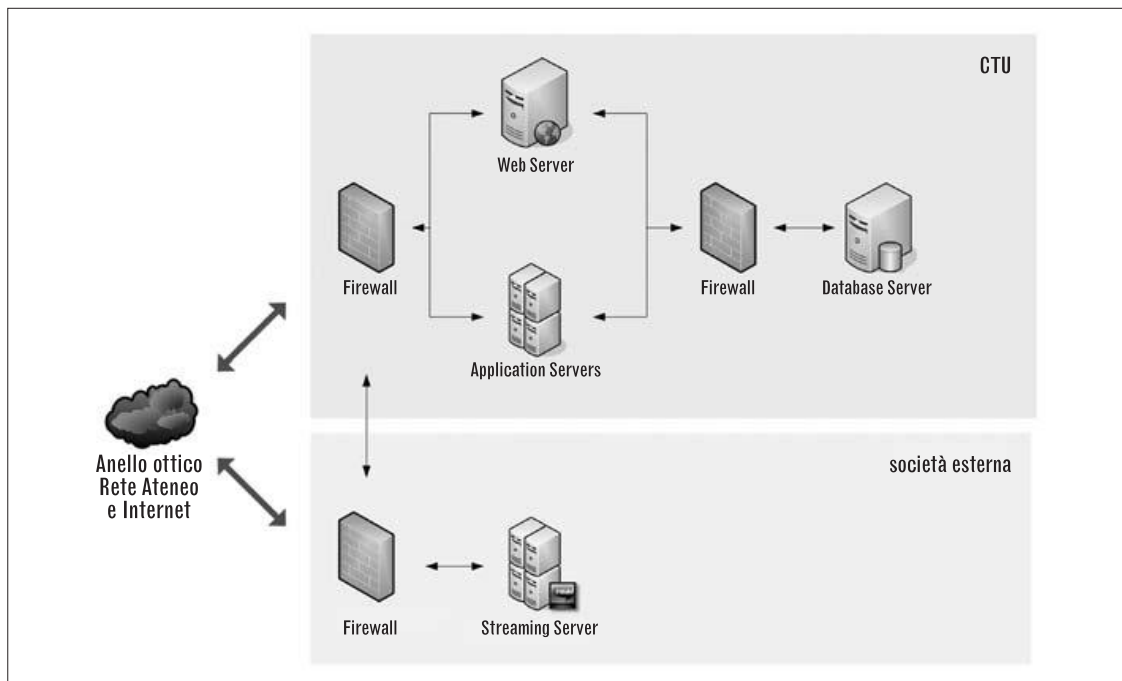


Figura 5 Architettura hardware di SSRI online.

L'erogazione di moduli multimediali in streaming necessita di una buona disponibilità di banda e di una garanzia di funzionamento del servizio vicina al 100%. Al fine di non incrementare ulteriormente il carico della connessione tra rete universitaria e rete commerciale si è optato per l'hosting dello streaming server presso una società esterna agganciata direttamente alla rete di I.net.

3. Conclusioni

L'attività di progettazione e sviluppo di Ariel.Net è stata condotta nell'arco di circa sei mesi a opera di due sviluppatori senior affiancati da tre tecnici software, impegnati prevalentemente nell'attivazione delle strutture degli insegnamenti e nell'implementazione dei materiali didattici. Da questo punto di vista, tempo e risorse disponibili hanno rappresentato alcune delle maggiori criticità riscontrate in fase di realizzazione del progetto.

Relativamente alla fase di erogazione, attualmente in corso, sono invece da segnalare alcune anomalie di funzionamento causate da modalità, non previste, di interazione dello studente con l'ambiente e difficoltà di gestione/implementazione delle modifiche sui contenuti che impattano sul tracciamento pregresso delle attività svolte.

I prossimi mesi vedranno il team di progettisti e sviluppatori di SSRI online impegnati nell'attivazione degli insegnamenti del 2° e 3° quadrimestre, nella progettazione del 2° anno di questa prima edizione del CdL online e nell'attivazione di una seconda edizione. In vista di queste attività si prevedono alcuni sviluppi per Ariel.Net quali ad esempio: la progettazione di funzionalità a supporto del lavoro collaborativo, l'ottimizzazione delle funzionalità di monitoraggio e reportistica a disposizione del tutor per il controllo sulla classe studenti, l'implementazione di strumenti per la gestione del servizio dal lato amministratore e l'adeguamento dell'interfaccia rispetto a browser differenti da MS Internet Explorer.

BIBLIOGRAFIA

Mason, R. (1998). *Models of online courses, proceedings of conference networked lifelong learning: Innovative approaches to education and training through the internet*, University of Sheffield.