

Esperienze e Sperimentazioni nell’Insegnamento Universitario dell’Informatica

Giansalvatore Mecca
Università della Basilicata
mecca@unibas.it

Versione del 04/03/2003

Sommario

Nell’Università italiana che cambia è importante sottoporre a valutazione critica anche il modo tradizionale di fare didattica. In questo lavoro vengono discusse alcune esperienze di carattere metodologico per migliorare la qualità dell’insegnamento. L’obiettivo è quello di fornire un contributo iniziale ad una più ampia discussione sui metodi e gli strumenti per la didattica universitaria.

1. Introduzione

E’ noto a tutti che l’Università italiana sta attraversando un momento di profondo mutamento. L’autonomia universitaria, la riforma dei cicli universitari e l’introduzione del cosiddetto “3+2” sono state le cause scatenanti di questo cambiamento, ed hanno costretto a ripensare impostazioni consolidate dei corsi di studio universitari per calarle nella nuova legislazione. Sarebbe però riduttivo vedere questo processo come un semplice adeguamento a norme che cambiano; viceversa, a me sembra che l’elemento più innovativo che la riforma ha portato sia la crescente sensibilità che nell’Università sta nascendo verso i problemi della qualità della didattica e dell’efficacia dei processi formativi.

Il regime di autonomia impone infatti l’esigenza di “accreditare” all’esterno i corsi di studio, una chiara novità per l’Università italiana, che sta generando interessanti riflessioni negli Atenei e sperimentazioni preliminari. Per tutte, si pensi al progetto CampusOne [CampusOne, sito Web] promosso dalla Conferenza dei Rettori delle Università Italiane (CRUI), il cui obiettivo primario è quello di “sostenere e diffondere l’innovazione tecnologica e formativa conseguente alla riforma didattica universitaria”. Questo obiettivo viene perseguito attraverso vari strumenti, tra i quali spicca la “valutazione della qualità, sulla base di metodologie di controllo che consentono di analizzare e valutare la qualità delle attività didattiche dei corsi di studio, in una prospettiva di accreditamento dei percorsi formativi.”

I sistemi di gestione e le metodologie di controllo della qualità (si veda ad esempio [Peri, Monteleone 2000]) sono uno strumento importante per programmare gli obiettivi del processo formativo e valutare la prestazione. Credo però che, nell’ottica di perseguire la qualità della formazione universitaria, sia altrettanto importante ripensare alle metodologie utilizzate per lo svolgimento delle attività didattiche. Per metodologie, in questo contesto, intendo le tecniche e le linee guida alla base dell’impostazione dell’insegnamento, della scelta dei contenuti, della preparazione dei materiali, e degli strumenti di supporto allo studio personale. Questa esigenza è ulteriormente avvertita nei contesti in cui si sperimentano forme di didattica innovativa, come la teledidattica o il *Web learning* [Colomi 2002], forme in cui la trasmissione delle informazioni da docente a studente è basata su canali non convenzionali, e di conseguenza assai sensibile all’impostazione dei contenuti.

Questo articolo descrive alcune esperienze nella sperimentazione didattica, con particolare riferimento all’insegnamento dell’informatica nei corsi di studio universitari. Le esperienze discusse sono la *Didattica Breve* [Ciampolini 1993] e la *Ricerca Metodologico-Disciplinare* [Ciampolini, Piazzi 2000], e il progetto dei *pattern pedagogici* (“Pedagogical Patterns Project”) [PPP, sito Web]. Oltre a queste, l’articolo descrive un’esperienza da me condotta negli ultimi anni, basata su una metodologia sperimentale, che io chiamo “*didattica a strati*”.

Dei numerosi aspetti che riguardano la didattica e le metodologie collegate quest’articolo ne affronta in realtà solo uno: la scelta della sequenza didattica, ovvero il problema di scegliere l’ordine in cui presentare gli argomenti di un insegnamento. Altri aspetti – come lo stile delle lezioni, il coinvolgimento degli studenti, le verifiche e la valutazione – pur essendo altrettanto importanti non sono affrontati.

Vorrei che fosse chiaro da subito che l'obiettivo di questo lavoro non è quello di suggerire teorie pedagogiche – cosa al di fuori delle mie conoscenze e delle mie capacità – né tantomeno proporre metodi definitivi per migliorare la qualità dell'insegnamento. Questo articolo ha lo scopo esclusivo di raccontare delle esperienze e, possibilmente, di rappresentare il punto di partenza per una discussione più ampia sul tema delle metodologie didattiche per la nuova Università. Del resto, tutte le esperienze raccontate in questo lavoro hanno come caratteristica comune una forte vocazione “disciplinare”, e cioè sostengono proposte metodologiche fortemente calate nei settori di riferimento – in questo articolo l'informatica – piuttosto che teorie generali di carattere pedagogico o cognitivo.

2. Metodologie Didattiche per l'Insegnamento

L'approfondimento delle metodologie per la didattica è da molto tempo un tema di studio nell'ambito della matematica (a questo proposito si veda, ad esempio, [Cuoco 2001]). E' più recente e meno consolidato lo studio della didattica nel settore dell'informatica¹.

Si potrebbe pensare che – essendo l'informatica una disciplina fondata sulla matematica discreta – molte delle proposte formulate in didattica della matematica si trasferiscano allo studio dell'informatica. Questo è vero solo parzialmente, e per varie ragioni. La ragione principale sta nel fatto che insegnare informatica vuol dire non solo insegnare concetti e metodologie, ma anche tecnologia. La necessità di mettere gli studenti in grado di risolvere problemi concreti, e cioè di analizzarli, di progettare soluzioni e di implementarle utilizzando il calcolatore, pone in molti casi problemi didattici non trascurabili. Questi sono riassumibili a mio avviso come segue:

- il primo problema è quello di stabilire il rapporto corretto tra i diversi ambiti di insegnamento – per l'appunto concetti, metodologie e tecnologie – per evitare sbilanciamenti pericolosi, e in un senso – corsi eccessivamente teorici, dai quali gli studenti apprendono molti concetti ma senza essere in grado di metterli concretamente in pratica – e nell'altro – corsi eccessivamente tecnologici, dai quali gli studenti apprendono solo competenze prettamente operative, senza la dovuta ampiezza metodologica;
- il secondo problema, una volta stabilito il giusto equilibrio tra teoria e pratica, è rappresentato dal modo in cui le nozioni dei diversi ambiti devono essere insegnate; è difficile immaginare che i principi di una disciplina e le sue metodologie possano essere insegnate allo stesso modo in cui si insegnano le regole della sintassi di un linguaggio o i comandi per l'amministrazione dei sistemi informatici, ovvero nozioni di minore rilevanza dal punto di vista concettuale e metodologico e prevalentemente di carattere mnemonico.

Nel seguito di questo articolo vengono discusse, come detto, alcune esperienze legate alle metodologie di insegnamento. Le tre esperienze hanno origini e storia diversissime. Le prime due – la Didattica Breve e il progetto dei Pattern Pedagogici – rappresentano progetti pluriennali e di ampia portata.

La Didattica Breve ha una storia trentennale e vaste sperimentazioni nella scuola secondaria superiore italiana. Non si tratta in effetti di una metodologia definita, bensì di un contesto di riferimento, fatto di sollecitazioni, linee guida e spunti metodologici che dovrebbero mettere il singolo docente in grado di ricercare i metodi migliori per insegnare la propria disciplina.

Il progetto dei Pattern Pedagogici è più recente (risale alla metà degli anni '90) e fortemente radicato in un settore specifico, quello della programmazione orientata agli oggetti. L'idea alla base del progetto è quella di documentare – come già fatto con successo per la programmazione con i “design patterns” – soluzioni consolidate a problemi ricorrenti di insegnamento. Questa base di dati di “soluzioni didattiche” dovrebbe consentire a nuovi insegnanti di capitalizzare l'esperienza dei loro predecessori per affrontare meglio l'insegnamento di discipline che la pratica ha dimostrato essere particolarmente ostiche. Un esempio è rappresentato dalla programmazione orientata agli oggetti, che oramai rappresenta uno standard tecnologico di mercato, ma per la quale gli insegnanti hanno a lungo faticato a trovare tecniche di insegnamento efficaci.

Questo breve panorama basta per intuire come azzardare un paragone tra queste proposte sia assai arduo; tuttavia credo che sia possibile delinearne alcuni tratti comuni, in particolare per quanto riguarda l'argomento di questo articolo, e cioè le linee guida suggerite per la scelta della sequenza didattica.

¹ Esistono in Italia gruppi di ricerca che affrontano problemi di didattica dell'informatica. Si veda, ad esempio [NRDI, sito Web].

Le Idee di Fondo

Gli elementi di contatto tra i pattern pedagogici e la didattica breve possono essere sintetizzati con i due principi: “riappropriarsi della materia” e “sperimentare sequenze di insegnamento alternative”. Alla base di entrambi i principi c'è la considerazione che la strategia tradizionale di insegnamento – in cui si parte dai concetti semplici e dai prerequisiti per poi sviluppare i concetti più avanzati – non è sempre quella più efficace.

Per ovviare a questo problema, per cominciare il docente dovrebbe avviare una approfondita riflessione sui contenuti del proprio insegnamento, per riscoprirne i principi ed i collegamenti essenziali tra questi, ed in ultima analisi per “riappropriarsi” delle logiche della disciplina. Inevitabilmente, questa operazione è lunga e faticosa; in alcuni casi richiede, oltre che buona volontà, anche coraggio, perché corrisponde a mettere in discussione l'organizzazione tradizionale della disciplina, consolidatasi nel tempo e documentata nella gran parte della letteratura.

Dopo questa analisi approfondita, il docente dovrebbe essere in grado di concepire strategie alternative per lo svolgimento dell'insegnamento. Sia i pattern pedagogici – attraverso i pattern “early bird” e “spiral”, discussi successivamente – sia per certi versi la didattica breve – attraverso il concetto di “sequenza inversa” - suggeriscono l'opportunità, in alcuni casi, di “rivoltare” la sequenza di presentazione degli argomenti, introducendo immediatamente (o quasi subito) i concetti fondanti della materia, quelli più importanti e anche tradizionalmente più difficili da apprendere; a questa fase iniziale e cruciale del corso dovrebbe poi seguire una seconda fase in cui gli stessi concetti vengono ripresi e ridiscussi per migliorare il livello di apprendimento da parte degli studenti.

In questo modo si avrebbe il vantaggio di chiarire meglio agli studenti cosa è più importante apprendere, e cioè quello su cui è necessario concentrare gli sforzi e lo studio. Un ulteriore vantaggio, evidentemente, è quello di consentire agli studenti di metabolizzare meglio l'apprendimento dei concetti fondanti, presentandone progressivamente aspetti diversi.

Applicazioni all'Informatica

La mia esperienza didattica mi ha insegnato che i principi appena discussi sono in realtà straordinariamente importanti per insegnare con qualità. Eppure, la stessa esperienza insegna che, tentando di applicarli all'insegnamento universitario dell'informatica, il docente si scontra con una serie di problemi non trascurabili.

Uno dei problemi riguarda la necessità di mettere gli studenti in grado di sperimentare le nozioni impartite a lezione. La possibilità per gli studenti di esercitarsi a casa o in laboratorio utilizzando il calcolatore – allo stesso tempo peculiarità dell'informatica rispetto ad altre discipline e straordinaria opportunità di apprendimento – pone vincoli notevoli sull'organizzazione didattica. Come già discusso, per essere in grado di utilizzare il calcolatore è infatti indispensabile affrontare lo studio di numerosi dettagli tecnologici – dalle procedure di installazione e di utilizzo, ai dettagli della sintassi dei linguaggi. Paradossalmente, volendo perseguire una forma “integrale” di apprendimento, in cui lo studente apprende a lezione e immediatamente sperimenta in laboratorio o a casa, tutti questi dettagli finiscono, pur nella limitata rilevanza concettuale, per diventare di fatto centrali in termini di efficacia dell'insegnamento. Ciò rende delicata la scelta della sequenza didattica, e pone ostacoli al tentativo di sperimentare sequenze inverse come quelle descritte sopra.

Un altro problema significativo riguarda lo studio. E' inevitabile che gli studenti adottino l'organizzazione del corso anche come guida per lo studio personale. In alcuni casi una sequenza didattica che funziona bene in aula può rendere difficile lo studio successivo a casa. Infatti, per molti degli argomenti insegnati in informatica – specialmente i linguaggi di programmazione – è indispensabile che il materiale, oltre che una chiara illustrazione dei concetti, fornisca anche una guida di riferimento, da consultare all'occorrenza per ritrovare un aspetto specifico di quello o quell'altro argomento. Il problema dei riferimenti per gli studenti è un problema che la didattica breve si pone e supera brillantemente proponendo continue forme di “ripasso regressivo” da parte degli studenti; in altri termini, si assume che durante le frequenti interrogazioni, il docente stimoli gli studenti a ripercorrere i blocchi in cui è organizzata la materia, rivisitandone le relazioni logiche e i percorsi tematici. Purtroppo questa forma di apprendimento – che è chiaramente pensata per la scuola superiore – difficilmente si applica ad una lezione universitaria tipica – e cioè ad un'aula che può contenere fino a 100 studenti e oltre, in cui l'interazione tra docente e studenti è limitata. Per ovviare a questo problema, è dunque utile, in alcuni casi, cercare un compromesso opportuno tra efficacia della sequenza didattica e supporto allo studio individuale.

Un esempio interessante dei vantaggi e svantaggi che possono derivare dall'adottare una sequenza didattica non tradizionale è il recente libro sulla programmazione Java [Barnes, Kolling 2002] basato sullo strumento di sviluppo BlueJ. Il libro è pensato per essere un testo introduttivo sulla programmazione, e affronta da subito la programmazione a oggetti. Avendo una solida impostazione basata su pattern pedagogici, il libro adotta una sequenza didattica inversa, in cui il concetto fondamentale della disciplina – il concetto di oggetto software – viene introdotto fin dalla prima pagina, attraverso alcune definizioni essenziali, e prima quindi degli elementi tradizionalmente considerati introduttivi alla programmazione – come i tipi di dato, le variabili e le strutture di controllo. Il libro riprende a più riprese, nei capitoli successivi, aspetti diversi legati agli oggetti software, e contestualmente introduce i concetti che in precedenza erano stati trascurati.

E' mia opinione che il libro sia estremamente efficace nel perseguire il suo obiettivo dichiarato – quello di spiegare con chiarezza l'uso degli oggetti in un'applicazione Java. D'altro canto, però, è possibile individuare almeno due debolezze legate all'organizzazione proposta. La prima debolezza è che non è chiaro quanto, parallelamente alle nozioni sugli oggetti, lo studente apprenda di altri aspetti della programmazione – per esempio le strutture di controllo – che sono introdotte come aspetti collaterali del comportamento di un oggetto; tra l'altro, a causa di questo stile di presentazione, per un periodo di tempo abbastanza lungo gli studenti non sono in grado di svolgere che semplicissimi esercizi di programmazione. La seconda debolezza consiste nel fatto che la presentazione di vari degli argomenti trattati risulta “distribuita” tra i capitoli del libro, che quindi non può essere considerato come una adeguata guida di riferimento per lo studio successivo alla lezione frontale.

Dal tentativo di trovare un compromesso tra pregi e debolezze legati alle sequenze didattiche non tradizionali nasce l'idea della didattica a strati. Si tratta di una sperimentazione condotta negli ultimi anni nell'ambito dei corsi da me insegnati. L'idea alla base della didattica a strati è quella di organizzare la didattica di un corso in una sequenza di argomenti (o unità didattiche), in modo tradizionale; la presentazione di ciascun argomento viene però organizzata in modo non lineare, distinguendo chiaramente le diverse categorie di concetti; queste categorie, denominate “concetti fondamentali”, “dettagli e approfondimenti” e “concetti avanzati” rappresentano in qualche modo strati concentrici – simili a quelli di una cipolla, analogo caro agli informatici – attraverso l'esplorazione dei quali viene costruito l'insegnamento.

Nel seguito le varie esperienze discusse vengono presentate in maggior dettaglio, per metterne ulteriormente in evidenza le analogie e le differenze.

3. La Ricerca Metodologico-Disciplinare

La Didattica Breve [Ciampolini 1993] è una proposta metodologica per l'insegnamento sviluppata dal Prof. Filippo Ciampolini alla fine degli anni '70. E' possibile definirla come *“l'insieme di tutte le metodologie didattiche di qualsiasi tipo, vecchie e nuove, che, nel rispetto del rigore scientifico da un lato, e dei contenuti di programma dall'altro, aggiungono ai tradizionali obiettivi della didattica anche quello di ottenere significativi guadagni temporali nell'insegnamento e nell'apprendimento”* [Ciampolini, Piazzi 2000].

A proposito di questa definizione, l'opuscolo [IRRSAE] riporta le seguenti affermazioni del Prof. Ciampolini: *“Desidero innanzitutto sottolineare che la didattica breve ragiona a parità di condizioni rispetto alla didattica tradizionale. A parità di rigore scientifico innanzitutto, se non addirittura cercando di migliorare il rigore stesso. Non è, dunque, la didattica breve una didattica frettolosa, approssimativa, riassuntiva [...] Al contrario, si tratta di una didattica molto 'premeditata' e, in particolare, molto curata negli approfondimenti concettuali”*.

L'obiettivo legato alla velocità dell'insegnamento ha in effetti una radice storica. La didattica breve è infatti nata come uno strumento per favorire l'aggiornamento rapido dei docenti delle scuole secondarie, contesto in cui i tempi limitati a disposizione dell'attività didattica sono evidentemente un vincolo notevole. Ben presto, però, la didattica breve si è trasformata da strumento per l'aggiornamento professionale dei docenti a strumento per l'insegnamento nella scuola, fino a diventare popolare negli istituti secondari italiani, e a trasformarsi nella più recente incarnazione, la Ricerca Metodologico-Disciplinare [Ciampolini, Piazzi 2000].

Per quanto la vocazione originale sia fortemente legata alla scuola – questo emerge chiaramente dalla lettura degli scritti sulla didattica breve, dalle priorità e dai problemi affrontati, a cominciare dal recupero scolastico – recentemente si è avviato un tentativo di introdurre la didattica breve come metodologia di riferimento anche all'interno dell'Università, prima nell'ambito dei diplomi universitari (le cosiddette “lauree brevi”) e attualmente nei corsi di laurea triennale (a questo proposito si veda, ad esempio, il corso

metodologico organizzato dall'Università di Siena nell'ambito delle attività legate a CampusOne [Siena, sito Web]).

Riassumere i dettami della didattica breve in questo breve paragrafo è impresa non facile. Questo è dovuto a due ragioni. La prima ragione è l'ampiezza dell'analisi, la varietà delle tecniche, e la numerosità delle linee guida descritte nei due testi di riferimento [Ciampolini 1993] [Ciampolini, Piazzi 2000]. Questo, che è sicuramente un sintomo di grande maturità del metodo, da una parte ne rappresenta la ricchezza e il valore; dall'altro può rappresentare un limite in tutti i casi in cui il novizio voglia avvicinarsi al soggetto senza precedenti conoscenze (come lo stesso Ciampolini sostiene, in effetti l'apprendimento delle tecniche della didattica breve richiede uno studio tutt'altro che breve).

L'altra ragione è che – come già discusso – la didattica breve ha una natura fortemente “disciplinare”. In altri termini, non pretende di essere una collezione di metodi e di linee guida di carattere multidisciplinare e di applicabilità generale. Bensì, richiama fortemente alla ricerca metodologica radicata negli specifici settori disciplinari, in ciascuno dei quali il docente dovrebbe riscoprire le tecniche più appropriate al miglioramento didattico. In questo spirito, il prof. Ciampolini riassume la didattica breve con le tre parole “ricerca-metodologia-disciplina” (da cui ricerca metodologico-disciplinare), e in questo spirito è possibile tratteggiare il percorso tipico del docente che voglia avvicinarsi alla didattica breve.

Il primo passo consiste tipicamente nel progettare o tenere un corso utilizzando metodologie tradizionali. Dall'impostazione tradizionale nasce il percorso di ricerca, il cui obiettivo è quello di consentire all'insegnante di “riappropriarsi” della propria materia. Il primo passo di questo processo – un passo addirittura preliminare alla vera ricerca metodologica – consiste nel “distillare” la materia.

Il termine “distillare”² che, come efficacemente riportato su [Crosio, sito Web], richiama l'idea di “far decantare, filtrare, estrarre isolando, estrapolare quanto è essenziale, separare le parti di un tutto indifferenziato”, descrive il primo passo dell'aspirante didatta breve, che è quello di analizzare l'oggetto del suo insegnamento, scomponendone i contenuti fino ai minimi termini, per poi ricostruire i legami e le relazioni tra i frammenti, e progressivamente ricostruire il tutto secondo logiche nuove.

L'attività di distillazione viene condotta in due fasi. La prima fase è la cosiddetta “distillazione verticale” della materia, il cui obiettivo è quello di “elencare tutti gli argomenti che la compongono, ponendoli nella sequenza che il docente segue nella propria esposizione agli studenti”. In altri termini, il distillato verticale di una materia è una lunga lista di argomenti, di taglia abbastanza minuta e di genere diverso – definizioni, dimostrazioni, metodologie, sperimentazioni ecc. – che, uno dopo l'altro compongono il programma del corso; gli argomenti sono tipicamente organizzati in blocchi che costituiscono le voci del programma del corso. Un esempio di distillato verticale per lo studio della meccanica in fisica prevederà voci come “Generalità dei vettori (prerequisito)”, “Somma e differenza di vettori (prerequisito)” e successivamente “Moto (definizione)”, “Traiettoria (definizione)”. Il distillato verticale è in sintesi l'occasione in cui il docente è costretto a misurarsi con i collegamenti essenziali all'interno della sua materia, individuando da una parte i contenuti fondanti, dall'altra le logiche che li attraversano.

Il passo successivo alla distillazione verticale è la cosiddetta “distillazione orizzontale”, che, in estrema sintesi, corrisponde a stabilire una strategia di presentazione – ovvero una sequenza di “passi elementari” – secondo cui introdurre ciascuno degli argomenti presenti nel distillato verticale.

Una volta completata l'attività di distillazione, comincia la ricerca metodologica vera e propria. Questa corrisponde in larga parte ad un lavoro di smontaggio e ricostruzione del distillato verticale, in cui il docente ha due obiettivi:

- cercare le logiche fondamentali – ovvero i collegamenti, le relazioni, le dipendenze tra i concetti e i blocchi della materia, rappresentate anche attraverso mappe e grafici di vario genere – per metterle in evidenza e rendere manifesta la struttura essenziale della disciplina, che così diventa rapidamente “ricostruibile”; questa ricerca consente tra l'altro di riconoscere i concetti fondanti della materia – il cosiddetto “zoccolo duro” - e cioè quelli attorno ai quali si addensano le logiche fondamentali;
- avviare l'attività di “contrazione”, ovvero tutte quelle iniziative che scarnificano la presentazione della disciplina attorno alle logiche fondamentali, anche utilizzando le relazioni con altri insegnamenti.

Sulla base delle due attività elencate, il didatta breve ricostruisce, dopo aver smembrato, i contenuti del suo corso, e così l'intero processo è riassumibile con la sequenza:

2 Il termine “distillare”, tra l'altro, non è una invenzione originale dei ricercatori di didattica breve, ma è stato mutuato dalle teorie del ricercatore italo-americano Robert Fano, del MIT.

in cui gli originali contenuti, di impostazione tradizionale, vengono preventivamente distillati e poi, attraverso le riflessioni sul metodo, vengono riorganizzati in nuovi contenuti di didattica breve.

In questo processo il didatta breve di avvale di numerosi strumenti metodologici. Volendo limitare l'analisi dei molti aspetti della didattica breve all'oggetto di questo scritto, tra questi discuteremo le proposte relative alle "sequenze di insegnamento", ovvero i modelli secondo cui organizzare i contenuti del corso dopo averne distillato, contratto e ripensato i contenuti.

Oltre al modello tradizionale, chiamato "sequenza storica", secondo il quale gli argomenti vengono introdotti seguendo un ordine logico che muove dal semplice verso il complesso – prima i prerequisiti e le nozioni semplici, poi le elaborazioni e i concetti complessi – vengono suggeriti altri modelli alternativi. I più interessanti sono la "sequenza inversa" e le sue varianti, come la "sequenza a rampa ripida" e la "sequenza iterativa", basate tutte sulla seguente idea: insegnare subito le leggi più complesse e fondamentali della disciplina – ovvero lo "zoccolo duro" – per poi ripercorrere, ad uno ad uno, i blocchi del programma. Come già discusso, la filosofia alla base di queste proposte è quella di dare priorità ai concetti ritenuti più importanti, compensando il minor tempo destinato ai prerequisiti con ulteriori iterazioni nel seguito del corso per completare le conoscenze degli studenti.

In [Ciampolini, 1993], la cosiddetta "sequenza iterativa" viene presentata come fatta di due passaggi attraverso il distillato verticale del corso. Il primo passaggio è orientato ad insegnare lo "zoccolo duro" della disciplina; il secondo passaggio – in cui il distillato viene ripreso dall'inizio – è una forma di iterazione finalizzata a completare l'apprendimento della materia. Questa versione iniziale della sequenza viene poi ulteriormente articolata in [Ciampolini, Piazzini 2000] introducendo una terza iterazione; in questa nuova versione, "le prime due iterazioni nel loro insieme non esauriscono completamente l'intero distillato della materia in questione. Restano infatti alcuni argomenti, di più elevato livello, che possono ritenersi non indispensabili per una più che decorosa preparazione della materia in oggetto." Questi argomenti vengono esauriti in un terzo passaggio attraverso il distillato, detto di "eccellenza".

4. Il Progetto dei "Pattern Pedagogici"

Il "*Pedagogical Pattern Project*" [PPP, sito Web] nasce nella metà degli anni '90. È ispirato, nel nome e nelle tecniche, ai noti "*design patterns*" [GOF 1994]. In generale, per "pattern" si intende una soluzione sperimentata ad una classe di problemi ricorrenti; il principio alla base della diffusione dei pattern è quello secondo il quale documentare soluzioni consolidate consente ad altri di imparare dall'esperienza, in modo da riapplicare le soluzioni apprese in tutte le circostanze in cui si ripresenta un'istanza del problema relativo. Mentre i "design patterns" (o pattern di progetto) sono soluzioni di progetto per lo sviluppo di codice orientato agli oggetti, i "pedagogical patterns" (o pattern pedagogici) sono soluzioni metodologiche per l'insegnamento dell'informatica.

Uno degli aspetti interessanti del progetto è che nasce dalla comunità di ricerca della programmazione orientata agli oggetti e dell'ingegneria del software. I primi articoli appaiono infatti negli atti dei convegni del settore [Goldblatt 1995] ed essenzialmente si pongono il seguente problema: una volta note le tecniche che consentono concretamente di migliorare la qualità della programmazione (i design pattern, per l'appunto), come insegnarle efficacemente agli studenti ?

Il problema nasce dalla constatazione – comune a tutti i lavori sui pattern pedagogici – che la programmazione a oggetti e le tecniche collegate sono un argomento "difficile" da insegnare, di gran lunga più complesso e rischioso della programmazione procedurale tradizionale. A questo problema, i ricercatori hanno pensato di dare una risposta documentando tecniche di insegnamento che nel corso degli anni si sono rivelate efficaci nel risolvere questo o quel problema.

I pattern pedagogici sono di conseguenza una collezione di tecniche per l'insegnamento dell'informatica (per un elenco delle principali si veda ad esempio [Bergin 2000]). Alcune di queste tecniche sono ispirate al buon senso e di fatto fanno parte del bagaglio di qualsiasi buon insegnante; si veda ad esempio il pattern "*fixer upper*", che suggerisce di sottoporre agli studenti esempi complessivamente ben strutturati, ma contenenti errori, sfidandoli a trovarli e a correggerli, oppure il pattern "*gold star*", che suggerisce di gratificare gli studenti per i loro buoni risultati.

Altre tecniche, viceversa, hanno un impatto profondo sul modo di concepire l'insegnamento delle discipline. Particolarmente interessanti, in quest'ottica, sono i pattern "*early bird*" e "*spiral*".

Il pattern “*early bird*” (che potrebbe essere tradotto approssimativamente in “uccello mattiniero” o “uccello che canta per primo”) predica di organizzare la sequenza degli argomenti insegnati in un corso in modo che gli argomenti più importanti (“the big ideas”) siano insegnati per primi. Esempi concreti suggeriti dall’articolo di Bergin sono: in un corso di programmazione orientata agli oggetti, insegnare gli oggetti per primi; in un corso di fondamenti di programmazione, insegnare i sottoprogrammi prima delle strutture di controllo e la ricorsione prima dell’iterazione. La filosofia alla base del pattern è quella secondo la quale i concetti importanti devono essere introdotti presto e rivisitati spesso dal docente, in modo che lo studente abbia più tempo per apprenderli e metabolizzarli.

Strettamente collegato a “*early bird*” è il pattern “*spiral*”, secondo il quale alcuni argomenti, invece che affrontati tutti in una volta, dovrebbero viceversa essere introdotti poco alla volta, in più riprese, ogni volta aggiungendo ulteriori informazioni a quelle introdotte precedentemente. Questo, tra l’altro, consente di risolvere il problema delle dipendenze e dei prerequisiti, che in alcuni casi apparentemente impedisce di affrontare subito i concetti importanti: secondo il pattern, i concetti importanti e tutti i relativi prerequisiti dovrebbero essere introdotti rapidamente e nelle linee fondamentali all’inizio del corso, per poi ritornarci successivamente aggiungendo ulteriore materiale e discutendone aspetti nuovi inizialmente trascurati. Un esempio di applicazione fornito nell’articolo di Bergin riguarda per esempio le strutture di controllo nei linguaggi di programmazione: non essendo considerate dall’autore come centrali rispetto all’apprendimento delle tecniche di programmazione, il pattern suggerisce di introdurli subito nelle linee fondamentali (esempio: *if* senza *else*), per poi riaffrontarli in un momento successivo del corso aggiungendo gli aspetti trascurati (*else*). In questo senso, la proposta è simile nello spirito a quella suggerita dalle sequenze didattiche inverse e a rampa ripida della didattica breve.

5. Esperienze di Didattica a Strati

In questo paragrafo discuterò alcune esperienze didattiche da me condotte negli ultimi anni nell’insegnamento di corsi universitari presso l’Università della Basilicata. I corsi a cui farò riferimento sono:

- il corso di Basi di Dati della laurea in Informatica offerto nell’anno accademico scorso, un modulo di 6 crediti³ didattici previsto al secondo anno, i cui contenuti prevedono lo studio dei modelli per le basi di dati relazionali, dei linguaggi di interrogazione (SQL), e delle metodologie di progettazione
- il corso di Tecnologie di Sviluppo per il Web, un modulo di 9 crediti didattici previsto al secondo anno, e offerto contemporaneamente al precedente, i cui contenuti prevedono lo studio delle tecnologie e delle metodologie per lo sviluppo di applicazioni Web, sia sul lato del “client” (HTML e tecnologie per la presentazione, accessibilità), che sul lato “server” (persistenza, programmazione Web, componenti).

Rispetto ai due moduli elencati, vale la pena di notare che si tratta di due insegnamenti dalla storia completamente diversa. Il modulo di Basi di Dati è un modulo tradizionalmente offerto nell’ambito dei corsi di laurea di informatica e di ingegneria informatica. Esistono testi di ottimo livello (per esempio [Atzeni 1999]), con a corredo materiale didattico già pronto per l’uso. Il modulo di Tecnologie di Sviluppo per il Web è viceversa un insegnamento relativamente nuovo, per il quale non esiste un’impostazione consolidata, né tantomeno veri e propri testi di riferimento. In effetti, il fatto di dover costruire ex-novo il programma di un nuovo insegnamento – quello di Tecnologie Web – senza poter contare su lavoro già fatto ma anche senza vincoli di sorta, ha rappresentato la motivazione principale per questa sperimentazione metodologica. Il corso di Basi di Dati – da me già insegnato in forma tradizionale in passato – è servito in qualche modo da “prova di verifica” dell’efficacia del metodo, essendo io in maggior misura capace di valutare il livello raggiunto dagli studenti ed i risultati dell’insegnamento.

Il programma di entrambi i corsi è stato organizzato per argomenti – o anche “unità didattiche” – in modo da ricoprire l’insieme dei contenuti previsti. Esempi di argomenti per il corso di Basi di Dati sono: “Il Modello Relazionale”, “Le Strutture Fisiche”, “Il Linguaggio SQL”, “Concorrenza e Affidabilità dei DBMS”, “La Progettazione Logica”. Esempi di argomenti per il corso di Tecnologie Web sono “Il Linguaggio XML”, “Utilizzo di Fogli di Stile”, “Architetture e Metodologie delle Applicazioni Web”, “Java Servlet”.

Gli argomenti però non sono stati sviluppati in modo lineare – e cioè dalle nozioni iniziali fino a quelle più avanzate – bensì secondo tre diversi strati di presentazione.

3 I due corsi a cui faccio riferimento hanno nel corso degli anni subito cambiamenti nel numero di crediti. Per semplicità farò riferimento alla distribuzione di crediti più recente, ovvero 6 crediti per il modulo di Basi di Dati e 9 crediti per il modulo di Tecnologie Web.

Il primo strato è quello dei “concetti fondamentali”. Si tratta dei concetti di ciascun argomento senza la comprensione dei quali non è possibile dire di averne veramente appreso gli aspetti importanti; si tratta in altri termini di tutte quelle nozioni per le quali è indispensabile che lo studente faccia uno sforzo di comprensione, e che quindi devono essere insegnati nel modo più chiaro possibile, evitando di appesantire la presentazione con aspetti di importanza minore; per fare un esempio: nella presentazione del modello relazionale, credo che sia importante che lo studente comprenda e apprenda il concetto di base di dati, di tabella, di enunzia e la natura dei vincoli di integrità; è opportuno viceversa che non sia distolto da aspetti collaterali a questi argomenti, come le varianti sintattiche che l'SQL offre per specificare i vincoli, oppure la gestione degli aggiornamenti in cascata per i vincoli di integrità referenziale.

Il secondo strato è quello dei “dettagli e gli approfondimenti”; questo è lo strato in cui la presentazione dell'argomento viene completata introducendo tutti i dettagli – per esempio la sintassi dei linguaggi, che lo studente deve apprendere per essere capace di sperimentare la tecnologia – e gli approfondimenti – quegli aspetti complementari, come gli aggiornamenti in cascata, importanti da conoscere ma non fondamentali ai fini della comprensione dell'argomento; ci sono quindi due criteri secondo i quali separare i concetti di questo strato rispetto ai concetti fondamentali: (a) il tipo di apprendimento richiesto – i dettagli non dovrebbero richiedere particolare sforzo concettuale da parte dello studente, richiedendo viceversa un impegno di carattere quasi esclusivamente mnemonico; (b) la rilevanza nella comprensione dell'argomento – gli approfondimenti dovrebbero essere conoscenze meno rilevanti nel contesto dell'argomento rispetto ai concetti fondamentali;

L'ultimo strato è quello dei “concetti avanzati”. Per alcuni argomenti l'esperienza dice che alcuni aspetti tipicamente risultano ostici agli studenti, e per una intrinseca complessità concettuale, e per via di prerequisiti di cui gli studenti non sempre sono completamente in possesso; nella presentazione degli argomenti questi concetti sono stati separati rispetto agli altri, e confinati in ulteriori lezioni di approfondimento. Esempi di concetti avanzati sono per esempio gli indici multilivello nello studio delle strutture fisiche, oppure le interrogazioni avanzate in SQL (raggruppamenti e nidificazioni), oppure la programmazione MVC (Modello-Vista-Controllo) basata su framework per lo sviluppo Web; enucleare questi concetti ha l'utilità di mettere “sull'avviso” lo studente, e cioè chiedergli da subito un impegno e un'attenzione maggiore, e durante la lezione e durante lo studio individuale.

Un esempio di materiale didattico costruito a strati è disponibile in linea all'indirizzo [Mecca, sito Web]. Sulla base di quanto discusso sopra, la presentazione di un argomento – che normalmente richiede più di una lezione – è organizzata come segue:

- per prima cosa viene svolta la lezione (o le lezioni) in cui vengono presentati i concetti fondamentali; gli studenti sono avvertiti dell'organizzazione in strati del materiale, e quindi fanno di stare assistendo ad una lezione di concetti fondamentali; le lezioni in questione dovrebbero essere svolte utilizzando molti esempi, cercando di introdurre le nozioni lentamente e curando che gli studenti mostrino di avere compreso quanto presentato;
- successivamente vengono svolte le lezioni in cui si introducono i dettagli e gli approfondimenti; è opportuno che questo avvenga in lezioni distinte dalle precedenti; il materiale per queste lezioni è organizzato in modo da ripercorrere sinteticamente i concetti fondamentali – il che consente agli studenti una limitata ma utile forma di apprendimento per regressione – arricchendoli di volta in volta di aspetti nuovi prima trascurati. A differenza delle lezioni sui concetti fondamentali, che tendono ad essere più interessanti e a suscitare maggiormente l'interesse degli studenti, le lezioni sui dettagli in alcuni casi rischiano di risultare abbastanza noiose (per quanto un insegnante si sforzi di vivacizzarla, la sintassi del linguaggio SQL è noiosa); per ovviare a questo inconveniente è utile intermezzare la presentazione dei dettagli con esercizi e approfondimenti vari;
- infine, se l'argomento lo richiede, vengono svolte le lezioni sui concetti avanzati; la maggioranza degli studenti, essendo anche in questo caso avvertiti della natura delle nozioni introdotte, tende a prestare particolare attenzione a queste lezioni; costruire il materiale per queste lezioni è un compito abbastanza delicato; è infatti necessario, pur nella complessità dell'argomento, prestare particolare attenzione alla chiarezza; in effetti, utilizzando un gioco di parole è possibile dire che ciò che viene presentato sono “gli aspetti fondamentali dei concetti avanzati”, ovvero un “sublimato” delle nozioni che sia il più possibile comprensibile agli studenti, anche in questo caso, per quanto possibile, scevro da dettagli superflui.

Non tutti gli argomenti prevedono uno sviluppo completo nei tre strati. Per la maggior parte degli argomenti concetti fondamentali e dettagli esauriscono la trattazione. In altri casi – quelli in cui non è possibile individuare dettagli secondari – l'argomento viene presentato esclusivamente nei suoi concetti fondamentali; è questo per esempio il caso della progettazione logica delle basi di dati, tecnica per la quale esiste un

Sistemi per Basi di Dati - Lezione introduttiva
Modello Logico dei DBMS Relazionali - Concetti Fondamentali
Modello Logico dei DBMS Relazionali - Dettagli e Approfondimenti
Modello Fisico - Concetti Fondamentali
Modello Fisico - Dettagli e Approfondimenti
Modello Fisico - Concetti Avanzati
Algebra Relazionale - Concetti Fondamentali
Algebra Relazionale - Dettagli e Approfondimenti
SQL-92 - Concetti Fondamentali
SQL-92 - Dettagli e Approfondimenti
SQL-92 - Concetti Avanzati
Tecnologia dei DBMS - Concetti Avanzati
Progettazione di Basi di Dati - Lezione Introduttiva
Modello Concettuale - Concetti Fondamentali
Modello Concettuale - Dettagli e Approfondimenti
Progettazione Logica - Concetti Fondamentali
Forme Normali - Concetti Fondamentali
Progettazione Fisica e Tuning - Concetti Avanzati
<i>Tabella 1 Organizzazione delle lezioni per un modulo di Basi di Dati</i>

algoritmo chiaro e ben definito, che è possibile introdurre in modo completo in una o due lezioni. In altri casi ancora l'argomento è per sua natura complesso e quindi prevede solo concetti avanzati; è stato questo il caso, per esempio, della gestione della concorrenza e dell'affidabilità nei DBMS.

La strategia di presentazione illustrata è quindi basata su una forma di "iterazione" all'interno dei singoli argomenti, e cioè delle singole "unità didattiche", secondo la quale gli aspetti dell'argomento sono progressivamente e ripetutamente approfonditi. Nella tabella 1 viene riportato la sequenza delle lezioni relative ad un modulo di 7-8 crediti di basi di dati organizzato a strati⁴.

Sono possibili delle eccezioni locali a questa regola: in alcuni casi gruppi di argomenti strettamente collegati suggeriscono una strategia alternativa, in cui vengono presentati per primi tutti i concetti fondamentali dei vari argomenti, poi tutti i dettagli e infine i concetti avanzati. Un esempio di questa organizzazione è stato da me sperimentato nella parte del corso di Tecnologie Web che riguarda la programmazione lato server con la piattaforma J2EE (Java 2 Enterprise Edition). Questa parte del corso prevede di affrontare tre argomenti strettamente collegati: (a) i concetti legati alle applicazioni Web basate su Java nel loro complesso (definizione di applicazione Web, gestione dell'applicazione, aspetti metodologici, modello 1 e modello 2); (b) i concetti legati ai Java servlet (ciclo di vita dei servlet, gestione degli oggetti fondamentali, richiesta, risposta, sessione ecc.); (c) i concetti legati alle pagine JSP (ciclo di vita, elementi di sintassi e semantica).

Ciascuno dei tre argomenti ha concetti fondamentali, dettagli e concetti avanzati. Per essere concretamente in grado di esercitarsi nello sviluppo di applicazioni Web⁵ uno studente deve aver visto almeno gli aspetti essenziali di ciascun argomento. Di conseguenza, piuttosto che cominciare e concludere

4 Per illustrare meglio l'organizzazione degli argomenti ho preso in considerazione un modulo di taglia leggermente maggiore rispetto a quello offerto nel nostro corso di laurea in Informatica.

5 Contemporaneamente alle lezioni e alle esercitazioni in aula, nei due corsi menzionati sono state previste esercitazioni pomeridiane in laboratorio. L'esercitazione di laboratorio prevede normalmente che lo studente svolga utilizzando un calcolatore un esercizio sui concetti appresi a lezione. Nel caso dei corsi di cui stiamo trattando, il numero relativamente basso di studenti – 70 studenti circa iscritti al corso, dei quali circa 55 frequentanti le lezioni e le esercitazioni regolarmente – ha consentito di mettere in piedi un meccanismo di reale valutazione continua del livello di apprendimento.

Applicazioni Web J2EE – Concetti Fondamentali
Java Servlet – Concetti Fondamentali
Pagine JSP – Concetti Fondamentali

Applicazioni Web J2EE – Dettagli e Approfondimenti
Java Servlet – Dettagli e Approfondimenti
Pagine JSP – Dettagli e Approfondimenti

Applicazioni Web J2EE – Concetti Avanzati
Java Servlet – Concetti Avanzati
Pagine JSP – Concetti Avanzati

Tabella 2 Organizzazione delle lezioni relative alle applicazioni Web con J2EE

ciascun argomento in sequenza, risulta molto più efficace presentare i concetti secondo la strategia illustrata in Tabella 2 – e cioè prima tutti i concetti fondamentali, poi tutti i dettagli, e infine tutti i concetti avanzati. Questa organizzazione ha il grande vantaggio di consentire agli studenti di esercitarsi nello sviluppo di applicazioni già dopo la terza lezione di questo ciclo.

L'esempio citato mostra anche come questa strategia di organizzazione renda l'utilizzo del materiale didattico abbastanza flessibile, consentendo di “smontare” e “rimontare” le unità didattiche in vari modi, e quindi anche di costruire moduli di taglia e obiettivi diversi. Sul sito già citato [Mecca, sito Web] è disponibile per il materiale didattico di Basi di Dati anche una mappa delle dipendenze tra le varie lezioni, e viene illustrato come, rispettando le dipendenze, è possibile costruire moduli di insegnamento basati sul materiale di dimensione variabile dai 3 agli 8 crediti didattici.

Un ulteriore vantaggio riguarda lo studio personale. L'impostazione a strati delle lezioni ha infatti come conseguenza immediata quella di riflettersi sullo studio degli studenti. I vari strati dovrebbero fungere da “linea guida” per lo studente, e dovrebbero aiutarlo ad affrontare nel modo adeguato lo studio di ciascuna parte del corso, privilegiando in alcuni casi la chiarezza e la comprensione dei concetti, in altri concentrandosi sulla memorizzazione dei dettagli, in altri casi ancora affrontando con la dovuta concentrazione gli argomenti avanzati. Inoltre, l'organizzazione del materiale consente allo studente di ritrovare facilmente i riferimenti agli argomenti del corso.

In qualche modo è possibile sostenere che la proposta didattica qui descritta è basata su un “contratto con gli studenti”, in cui il docente mette in chiaro la rilevanza e la difficoltà di tutto quello che viene trattato a lezione. Lo studente viene invitato a esplorare la materia di studio, progredendo dai concetti fondamentali fino a quelli avanzati. In questo processo, lo studente aderisce al contratto proposto scegliendo il livello di impegno e di studio più conforme alle proprie aspettative e necessità.

Per concludere, credo che sia utile a questo punto discutere ulteriormente le relazioni tra questa proposta e quelle illustrate in precedenza. E' opportuno innanzitutto precisare che si tratta di relazioni individuate a posteriori, dal momento che le sperimentazioni di didattica a strati sono state da me condotte prima di venire a conoscenza di queste altre proposte. E' però indiscutibile che – come già discusso – esistano degli aspetti collegati, in particolare per quanto riguarda le sequenze di insegnamento.

Esistono anche significative differenze. Rispetto alla didattica a strati, oltre alla differenza già citata tra iterazione a livello globale (di insegnamento) e iterazione a livello locale (di unità didattica), una ulteriore differenza è legata al fatto che le sperimentazioni di didattica a strati non hanno mai avuto come finalità quella di guadagnare tempo nell'insegnamento. Un'altra differenza attiene all'impostazione complessiva delle due proposte: la didattica breve non è in effetti un metodo definito di insegnamento, ma una collezione molto più ampia di prescrizioni che dovrebbero guidare e addestrare il docente alla ricerca metodologica all'interno della propria disciplina. La didattica a strati, viceversa, ha portata molto più limitata: si tratta di una proposta specifica per l'organizzazione dell'insegnamento dell'informatica. In quest'ottica, credo che sia possibile inquadrare correttamente il rapporto tra le due proposte vedendo la didattica a strati come un esempio di ricerca metodologico-disciplinare, e cioè come una esperienza di approfondimento metodologico applicata specificatamente alle discipline informatiche.

Volendo cercare una differenza rispetto ai pattern pedagogici, è possibile rifarsi di nuovo all'ingegneria del software: infatti la relazione tra le due proposte è molto simile a quella nota agli ingegneri del software tra design pattern e “framework”. Infatti, volendo assimilare i pattern pedagogici ai “design pattern” – soluzioni

riutilizzabili per categorie specifiche di problemi – è possibile assimilare viceversa la didattica a strati ad un framework di sviluppo – ovvero un contesto strutturato che è possibile utilizzare per organizzare in modo sistematico la didattica di un corso. Estendendo ulteriormente questo paragone è possibile dire che – così come nello sviluppo del software i framework sono spesso basati sull'utilizzo di design pattern – così è possibile integrare molti dei pattern pedagogici in un corso costruito a strati.

Riferimenti

- [Atzeni 1999] P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone – Basi di Dati: Concetti, Linguaggi, Architetture - 2a Edizione - McGraw-Hill, 1999 – <http://www.dia.uniroma3.it/~atzeni/libroBD/>
- [Barnes, Kolling 2002] D. J. Barnes, M. Kolling – Object First with Java: A Practical Introduction using BlueJ – Prentice Hall, 2002
- [Bergin 2000] J. Bergin – Fourteen Pedagogical Patterns for Teaching Computer Science – Proceedings of the Fifth European Conference on Pattern Languages of Programs (EuroPLoP), 2000 - <http://csis.pace.edu/~bergin/PedPat1.3.html>
- [BlueJ, sito Web] BlueJ: The Interactive Java Environment – <http://www.bluej.org>
- [CampusOne, sito Web] Progetto CampusOne - <http://www.campusone.it>
- [Ciampolini 1993] F. Ciampolini – La Didattica Breve – Il Mulino, 1993
- [Ciampolini, Piazzì 2000] F. Ciampolini, F. Piazzì – La Ricerca Metodologico-Disciplinare: Una Strategia per il Rilancio della Scuola Italiana – Il Mulino, 2000
- [Colorni 2002] A. Colorni – Web Learning: Esperienze, Modelli, Tecnologie – Mondo Digitale n.1, 2002
- [Crosio, sito Web] R. Crosio – La Didattica Breve e La Ricerca Metodologico-Disciplinare - <http://www.valsesiascuole.it/crosior/db/>
- [Cuoco 2001] A. Cuoco – Mathematics for Teaching – Notices of the American Mathematical Society, vol. 48 (2), 2001
- [GOF 1994] E. Gamma, R. Helm, R. Johnson, and J. Vlissides (The Gang of Four) – Design Patterns: Elements of Reusable Object-Oriented Software – Addison Wesley, 1994
- [Goldblatt 1995] D. Lynne Goldblatt Anthony – Patterns for Classroom Education – Proceedings of the Second Annual Conference on Pattern Languages of Programs (PLoP), 1995 - <http://st-www.cs.uiuc.edu/%7Echai/writing/classroom-ed.html>
- [Gruppi RMD, sito Web] Gruppi di Ricerca Metodologico-Disciplinare - <http://rmd.scuole.bo.it/>
- [IRRSAE] Opuscolo Divulgativo sul Progetto InterIRRSAE – Ministero della Pubblica Istruzione – “Un Percorso di Didattica Breve: Verso la Qualità del Recupero Scolastico e nella Ricerca Metodologico-Disciplinare” - a cura di Stefano Contadini, IRRSAE Emilia Romagna. Estratti disponibili in linea su [Gruppi RMD, sito Web] e [Crosio, sito Web].
- [Mecca, sito Web] G. Mecca – Materiale Didattico per il Corso di Basi di Dati – <http://www.difa.unibas.it/users/gmecca/diogene/basiDiDati>
- [NRDI, sito Web] Sito del Nucleo di Ricerca in Didattica dell'Informatica, Università di Udine - <http://www.dimi.uniud.it/nid/>
- [Peri, Monteleone 2000] C. Peri, E. Monteleone – La Riforma degli Ordinamenti Didattici nelle Facoltà di Agraria. I. La Valutazione della Prestazione Didattica – Studio Editoriale Fiorentino, 2000
- [PPP, sito web] Pedagogical Patterns Project - <http://www.pedagogicalpatterns.org>, <http://www-lifia.info.unlp.edu.ar/ppp/>
- [Siena, sito Web] Università di Siena – Corso Metodologico di Studio e Insegnamento delle Materie Scientifiche - http://www.unisi.it/campusone/relazione_corso_metodologico.htm