
Progettazione e realizzazione di materiale didattico

Donatella Persico
Istituto per le Tecnologie
Didattiche - CNR,
Genova

Un approccio sistematico allo sviluppo di courseware, dall'ideazione alla validazione.

In questo lavoro affrontiamo le problematiche connesse allo sviluppo di materiale didattico, con particolare attenzione alle fasi di progettazione (ossia alle cosiddette fasi “alte” del ciclo di sviluppo) e a materiale che preveda l’uso di tecnologia. In generale, sarà privilegiato il punto di vista di chi sviluppa courseware a livello professionale, per la formazione o l’addestramento, piuttosto che l’approccio da “bricolage” di chi realizza una dispensa, degli appunti o anche un software di supporto alle proprie lezioni. Ciò non significa che questo secondo “tipo” di materiale non abbia una sua ragione di esistere e una sua dignità; ma soltanto che l’uso di un approccio sistematico allo sviluppo di courseware si fa tanto più importante quanto più vasta è la portata (in termini di contenuti e di popolazione obiettivo) dell’intervento didattico che esso deve supportare. Una ulteriore doverosa precisazione riguarda il significato con cui, in questo contesto, sarà utilizzato il termine *courseware*. In generale, infatti, questo termine identifica un insieme di materiali didattici comprendenti software, progettati allo scopo di promuovere conoscenze o capacità nell’ambito di un dato dominio. Sulla base di questa definizione assai generale, quindi, rientra nel courseware una ampia categoria di prodotti: da quelli basati su software didattico “direttivo” di tipo tutoriale o “drill and practice” ai cosiddetti am-

bienti di apprendimento, ossia quegli ambienti interattivi capaci di simulare situazioni reali o artificiali in cui lo studente può autonomamente “costruire” la propria conoscenza. Anche i sistemi che utilizzano tecniche di intelligenza artificiale per conferire flessibilità e adattività al software didattico (indipendentemente dalla direttività della strategia) rientrano ai fini di questo lavoro nella categoria del courseware. In tutti questi casi, comunque, a meno che gli obiettivi didattici che si intendono perseguire non siano estremamente limitati e specifici, il courseware non può essere progettato come un oggetto a sé stante, del tutto avulso dal suo contesto di utilizzo e indipendente dalle possibili (ed eventualmente molteplici) modalità di inserimento in un progetto didattico globale. Di conseguenza, è probabile e addirittura auspicabile che un courseware destinato ad affrontare una porzione significativa di un itinerario formativo venga progettato dai suoi autori come un sistema didattico complesso, e sia corredato da materiali (documenti di progetto, manuali, dispense, guide) che ne chiariscano le modalità d’uso agli occhi degli utenti e ne completino, se necessario, le funzionalità pedagogiche. Sulla base di questa interpretazione del termine courseware non vi è da stupirsi se le metodologie di sviluppo che prenderemo in esame si riferiscono in generale allo sviluppo di sistemi

didattici, indipendentemente dal fatto che essi comprendano o meno del software.

Nel seguito di questo lavoro presenteremo quindi alcuni modelli di sviluppo di sistemi formativi e alcuni strumenti e metodi specifici che possono essere utilmente adottati nel corso di tale attività. Si noti però che se i modelli illustrati sono sufficientemente generali da poter essere considerati indipendenti dalle caratteristiche del prodotto finale, i metodi e gli strumenti che verranno utilizzati per illustrare le problematiche realizzative sono spesso tali da prestarsi maggiormente alla produzione di un certo tipo di courseware piuttosto che un altro, e sono quindi stati scelti in modo tale da coprire un po' tutta la gamma di prodotti che, nella nostra accezione, ricadono sotto la definizione di courseware.

LE FASI DI SVILUPPO DI UN SISTEMA DIDATTICO

Gli strumenti e i metodi per lo sviluppo di courseware hanno subito nell'ultimo decennio profondi mutamenti, sotto la spinta delle esigenze del mercato e dell'evoluzione/evoluzione dell'hardware e del software. Per comprendere meglio la situazione attuale può essere utile ripercorrere alcune tappe fondamentali dello studio dei modelli di sviluppo di courseware e analizzare i problemi che più frequentemente si incontrano nella pratica e le soluzioni attualmente prospettate in ambito di ricerca. In tal modo sarà possibile raffinare gli aspetti metodologici alla luce - tra l'altro - di diversi contesti di produzione.

Nel 1984 [Ferraris et al, 1984a] proponevano un modello per lo sviluppo di sistemi didattici ispirato a quello cosiddetto "a cascata" per lo sviluppo di software, che prevede la definizione dei requisiti del sistema, delle specifiche, il progetto, l'implementazione, il controllo qualità e la manutenzione del sistema stesso. Ciascuna di queste fasi del ciclo di vita del software (e del courseware) pone le sue radici sulla precedente e prepara le basi per la successiva anche se non sussistono rigidi vincoli di sequenza temporale tra di esse: nel corso dello svolgimento di una data fase può infatti evidenziarsi la necessità di aggiustamenti o modifiche a livello delle fasi precedenti. Il controllo qualità e la manutenzione poi richiedono per definizione uno o più passi indietro lungo la "cascata" delle fasi precedenti, per correggere eventuali problemi di impostazione o di realizzazione del sistema.

Altri ricercatori [Romiszowsky, 1984, Rowntree, 1986, Hannafin et al, 1988, Dick et al, 1990] hanno affrontato il problema di definire un approccio sistematico allo sviluppo di sistemi didattici, di volta in volta approfondendone alcuni aspetti, oppure proponendo modelli generali diversi da quello citato, ma quasi sempre basati su una suddivisione del processo in fasi progettuali, fasi realizzative e momenti finalizzati al controllo qualità del prodotto. È quindi possibile stabilire un denominatore comune tra i modelli proposti che può essere rappresentato come in fig. 1 [Ferraris et al, 1985]. In questo modello, le attività di progetto comprendono la definizione dei requisiti e delle spe-

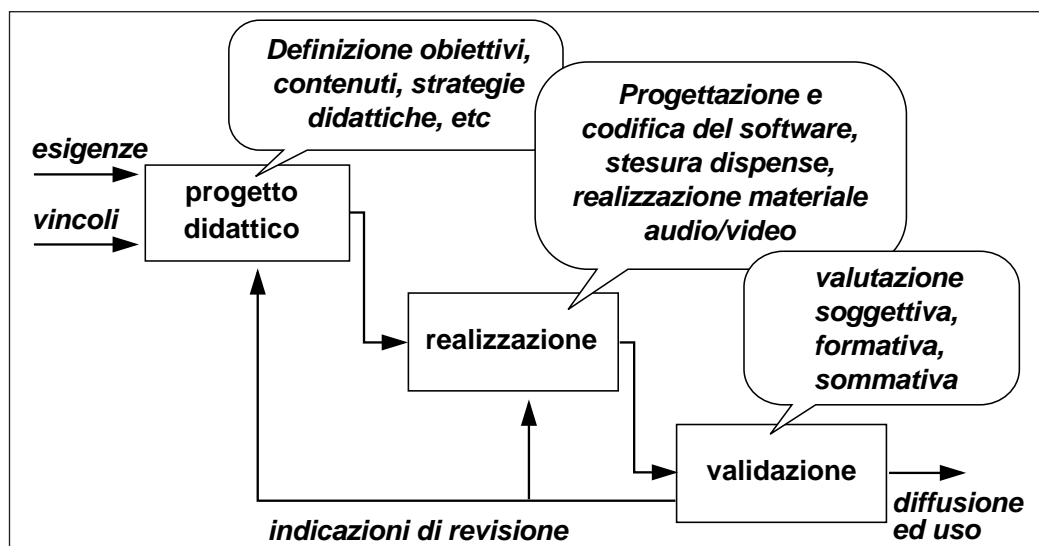


Fig. 1
Un modello per lo sviluppo di courseware

cifiche del sistema didattico, la realizzazione comprende sia l'implementazione di software sia la produzione di materiale a stampa, audiovisivo, o qualunque altro materiale si ritenga opportuno, la validazione comprende tutte le misure volte a garantire la qualità del prodotto, siano esse messe in atto nel corso dello sviluppo oppure a prodotto finito. Ma vediamo in dettaglio, fase per fase, di cosa si tratta e quali attività e competenze esse comportano.

Il progetto didattico

La definizione dei requisiti del sistema didattico richiede attività quali l'analisi del problema che si intende affrontare, del contesto educativo in cui si intende intervenire e l'analisi delle esigenze e dei vincoli che potrebbero condizionare il successo dell'intervento. Con "analisi del problema" si intende l'identificazione delle carenze dell'approccio corrente a cui si vuole far fronte. Ad esempio, il "problema" può risiedere nella impreparazione dei docenti ad affrontare specifici contenuti, oppure nella scarsa efficacia dell'approccio corrente, nella mancanza di motivazione degli studenti, nella disuniformità della loro preparazione di base, nella difficoltà intrinseca dei contenuti, nella distanza fisica tra docenti e studenti, nel cattivo rapporto numerico tra gli stessi, e molto spesso, in una combinazione di fattori di questo tipo. L'analisi del contesto educativo comprende lo studio dell'ambito per il quale il sistema didattico viene proposto (livello e tipologia della popolazione obiettivo, esigenze e caratteristiche del contesto, vincoli a cui l'intervento innovativo dovrà sottostare). Per alcuni autori [Farquhar et al, 1994] questa fase deve comprendere anche una analisi delle condizioni che potrebbero favorire od ostacolare l'"adozione" delle innovazioni progettate, per evitare che il sistema didattico proposto venga rigettato a priori (a torto o a ragione) in quanto troppo "rivoluzionario" rispetto al contesto. L'analisi delle esigenze e dei vincoli, infine, si intreccia con le attività elencate in precedenza, contribuendo a chiarire le condizioni al contorno del problema da affrontare, anche in termini di risorse disponibili (risorse umane, economiche, tempi di realizzazione, etc) e di strumenti didattici già esistenti (non è raro, purtroppo, il caso in cui si investano energie e fondi ignorando soluzioni già proposte da altri).

La definizione delle specifiche didattiche comprende invece la precisazione degli scopi dell'intervento, dei contenuti che esso affronterà, la scelta delle strategie didattiche adottabili e dei mezzi più adeguati per condurre l'intervento. Tutte queste attività sono strettamente correlate tra loro e vanno condotte tenendo presente i vincoli e le condizioni al contorno identificati in precedenza. Un altro aspetto che va considerato in questa fase è il cosiddetto rapporto costi benefici, dove costi e benefici non sono intesi in termini puramente economici: i benefici sono spesso valutabili in termini di rilevanza e diffusione dei risultati ottenibili, mentre i costi sono fortemente influenzati anche dai tempi di realizzazione (per il software, ad esempio, si valuta che il costo di un'ora di fruizione si aggiri intorno alle 200/400 ore di sviluppo). Per buona parte delle attività di specifica, sono stati proposti strumenti concettuali ad hoc o "presi a prestito" da altri ambiti lavorativi, il cui scopo è duplice: facilitare sia lo svolgimento dell'attività sia lo scambio di informazioni tra i diversi attori del processo di sviluppo.

Prendiamo ad esempio l'attività di strutturazione dei contenuti, ossia la rappresentazione degli stessi in una forma tale da esplicitarne le varie componenti e i legami che sussistono tra di esse, con il livello di dettaglio richiesto dalle esigenze realizzative del courseware. Ebbene, numerose esperienze dimostrano che l'adozione di un formalismo specifico (grafi [Mispelkamp H. et al, 1994], gerarchie di apprendimento [Persico et al, 1985], reti di Petri [Ferraris et al, 1984b], o altro) consente all'autore di dominare la complessità dei contenuti in fase di analisi, ma ha anche altri significativi vantaggi: il prodotto finale è facilmente leggibile, non contiene le ambiguità tipiche del linguaggio naturale ed è gestibile con strumenti automatici, nelle fasi successive dello sviluppo. Al di là di queste esperienze, va considerato che se un indice strutturato può essere una rappresentazione sufficiente e adeguata in fase di progettazione di un libro, altrettanto non si verifica qualora si voglia progettare un software didattico. Infatti, la "sequenzializzazione" dei contenuti intrinseca dell'indice potrebbe inficiare la costruzione di un ambiente software che consenta percorsi altamente adattivi e ramificati, e in ultima analisi la flessibilità di fruizione da parte degli studenti.

Nel caso in cui si intenda produrre un software che utilizzi metodologie e tecniche di intelligenza artificiale, i contenuti non costituiscono la sola base di conoscenza necessaria a condurre il dialogo didattico. Per tener conto sia delle caratteristiche cognitive e attitudinali dello studente sia delle possibili modalità di insegnamento della disciplina, questi sistemi inglobano infatti tre categorie di conoscenze distinte: sul dominio di contenuti, sulle strategie didattiche e sullo studente. Sin dalla fase di definizione delle specifiche occorre quindi progettare questi tre moduli e i relativi protocolli di comunicazione utilizzando formalismi adeguati. Tali attività possono essere svolte in parallelo da gruppi di autori diversi, tanto più che le competenze necessarie sono, in generale, ben differenziate (non sempre l'esperto dei contenuti è anche esperto di didattica e di ingegneria della conoscenza¹). Nel seguito vedremo quali strumenti possono agevolare gli autori di tali basi di conoscenza sia nella loro costruzione, sia nella integrazione dei rispettivi prodotti.

In fase di progetto didattico è già possibile pianificare una serie di interventi volti alla verifica della qualità del courseware. Infatti, una volta definiti gli obiettivi globali del courseware e quelli specifici delle sue componenti principali, è opportuno stabilire criteri e metodi per valutare il grado di conseguimento di tali obiettivi. Effettuare questa pianificazione prima di passare alla realizzazione del materiale didattico vero e proprio, incluso quello per la validazione, consente di legare direttamente le attività valutative ai "sapere" e "saper fare" che lo studente dovrebbe conseguire con la fruizione del courseware. Questa precauzione, che può sembrare inutile o banale, è tutt'altro che superflua. A conferma di ciò basta guardarsi attorno e constatare quante volte in ambito scolastico e non le attività di verifica dell'apprendimento e validazione del processo didattico si sposino solo in parte con il processo stesso e i suoi obiettivi.

La realizzazione

In questa fase, il prodotto delle attività descritte in precedenza costituisce la base di lavoro per chi dovrà tradurre il progetto didattico in un insieme di materiali concreti. Ad esempio, è qui che viene implementato il software, che si produce il materiale grafico

e audiovisivo, che si scrivono i testi destinati a studenti e docenti, le istruzioni per l'uso, etc. Tutte queste attività possono essere svolte in parallelo e in collaborazione tra più persone, tanto più che richiedono competenze diverse, che vanno dalla capacità di programmare, alla capacità di scrivere un testo con scopo educativo, a competenze grafiche e di comunicazione didattica in generale.

La suddivisione dei compiti tra i vari attori coinvolti nella fase di realizzazione del courseware porta in modo naturale ad una scomposizione in moduli del prodotto finale. Tale scomposizione offre notevoli vantaggi anche sul fronte della validazione del prodotto. Infatti, in molti casi, è possibile procedere alla valutazione della validità tecnica e dell'efficacia didattica di singole componenti del courseware. Per fare un esempio, la qualità tecnica del software può, anzi deve, essere verificata dai suoi autori secondo i principi informatici del "test topologico e funzionale"². Successivamente esso può essere usato (sotto il diretto controllo degli autori) da un piccolo gruppo di studenti allo scopo di individuare problemi di interfaccia e di progettazione. In alcuni casi è anche possibile che moduli autoconsistenti del materiale (per esempio un singolo capitolo di dispense, una particolare funzionalità del software) siano realizzati prima del resto e possano essere "provati" con un campione ridotto della popolazione obiettivo. Questo consente di individuare precocemente difetti del materiale e di minimizzare quindi i costi di riparazione. Ad esempio, la scelta di uno stile comunicativo poco efficace nella realizzazione di un testo è un errore che può essere recuperato riscrivendo soltanto un capitolo campione piuttosto che l'intero testo.

La validazione

Come accennato sopra, la validazione costituisce una fase dello sviluppo di courseware soprattutto dal punto di vista concettuale, anche se dal punto di vista temporale essa si distribuisce lungo tutto il processo, pur comprendendo una consistente attività di sperimentazione del prodotto finale. D'altro canto, l'intero modello a cascata non va interpretato in maniera rigidamente sequenziale. Infatti le attività di validazione evidenziano, per definizione, carenze dell'approccio adottato a cui in generale si pone rimedio risalendo la "cascata" fino al punto in cui si in-

¹ Con il termine "ingegnere della conoscenza" si identifica l'esperto di metodologie e formalismi di rappresentazione della conoscenza, indipendentemente dal dominio.

² Tali principi prevedono, rispettivamente, che tutti i percorsi possibili all'interno del diagramma di flusso siano esercitati correttamente almeno una volta, e che tutte le funzionalità previste in fase di specifica del software siano esercitate con successo, sia con valori "limite" dei dati di ingresso sia con valori intermedi.

dividua la causa del problema, eliminandola e ripercorrendo il ciclo fino in fondo compresa la verifica del fatto che la correzione apportata abbia avuto successo. In generale, recuperare un errore commesso in fase di progetto costa quindi (in termini di tempi e di lavoro), ben più che porre rimedio a carenze realizzative. Per lo stesso motivo, la precocità della individuazione di eventuali inadeguatezze del sistema in corso di sviluppo è essenziale ai fini di minimizzare l'investimento necessario a porvi rimedio.

Ma veniamo ora ad una analisi un po' più approfondita di quelli che sono i metodi di validazione più diffusi. Tanto per cominciare, occorre introdurre un po' di terminologia³. Il processo di validazione (fig.2) generalmente comprende una prima attività di controllo (o valutazione) che produce delle informazioni (o dati) circa l'oggetto da validare. I dati così ottenuti devono essere analizzati per produrre una interpretazione degli stessi che funga da base per un processo compensativo volto a correggere o migliorare l'oggetto di partenza. Le tre attività tipiche della validazione sono dunque la valutazione, l'analisi dei dati e la compensazione.

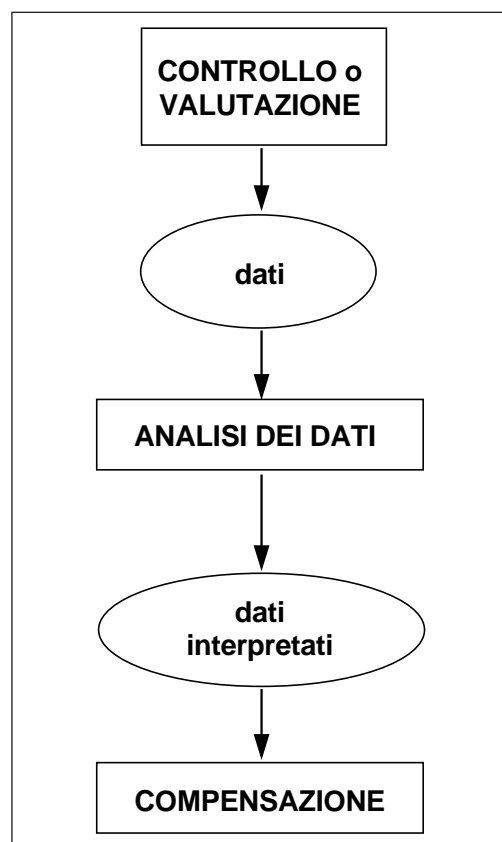
La valutazione

A seconda delle finalità con cui viene effettuata, si parla in genere di valutazione formativa o sommativa. Nel primo caso, lo scopo è ottenere informazioni sia puntuali sia generali che consentano di migliorare il prodotto. Di conseguenza, la valutazione formativa si effettua il più possibile nel corso dello sviluppo, piuttosto che al suo termine. La valutazione sommativa invece è finalizzata a raggiungere un giudizio globale sull'oggetto, come potrebbe essere il parere di un esperto imparziale che debba deciderne o meno la distribuzione o l'uso in un determinato ambito. È chiaro quindi che parlando di sviluppo di courseware l'interesse prevalente riguarda le metodologie di valutazione formativa. In ogni caso, tra i metodi di cui parleremo, molti sono utilizzabili sia in un'ottica formativa sia in un'ottica sommativa.

Un primo metodo di valutazione utilizzabile nel contesto dello sviluppo di courseware consiste nella valutazione soggettiva, ossia nel sottoporre un primo prototipo del prodotto, o addirittura i documenti prodotti in fase di progettazione, a degli esperti imparziali, diversi dagli autori. In generale, in questi casi, ci si rivolge ad esperti dei contenuti che ne certifichino la correttezza da quel punto di vista, a docenti della disciplina in esame che ne valutino, in base alla loro esperienza, le potenzialità dal punto di vista della efficacia didattica, a degli esperti di informatica e/o interfacce utente, a degli esperti di tecnologie didattiche, etc, ciascuno per il suo ambito di competenze. La valutazione soggettiva fornisce spesso utili "feedback"⁴ anche prima che sia pronta una versione utilizzabile del prodotto. Certamente però, di per sé, non è sufficiente a garantirne la qualità finale. Anzi, in alcuni casi può essere conveniente rimandare eventuali azioni compensative e attendere una conferma sperimentale del parere degli esperti, utilizzando se mai quest'ultimo come indicazione di aspetti "a rischio" da tener presente nelle successive attività di valutazione.

Un secondo importante metodo di valutazione sono le cosiddette "prove tutoriali", o anche "prove in casa", ossia l'uso del courseware con un campione ridotto (statisticamente non significativo) della popolazione obiettivo, in ambiente controllato (ossia in un ambiente in cui sia particolarmente agevole il rilevamento dei dati sull'uso, indipendentemente dal fatto che esso sia o meno

Fig. 2
Validazione



³ Le accezioni dei termini qui introdotti sono abbastanza diffuse ma non generalmente accettate. In questo contesto sono proposte soltanto ai fini di sciogliere possibili ambiguità.

⁴ O informazioni di ritorno, ma il termine inglese è ormai molto usato anche in italiano!

l'ambiente per cui il prodotto è stato progettato). Questo tipo di sperimentazione si effettua in generale sotto il controllo diretto degli autori. I dati rilevati in queste prove devono essere analizzati tenendo conto delle variabili ambientali non realistiche in cui sono state effettuate. Molto spesso per esempio, determinate informazioni positive non possono essere considerate del tutto attendibili. Per contro, in generale, le lacune del materiale che emergono in questa fase vanno prese molto sul serio, in quanto si sono verificate *nonostante* la presenza degli autori, *nonostante* le condizioni al contorno particolarmente favorevoli, etc. In ogni caso, le prove tutoriali hanno il grande vantaggio di poter essere effettuate con una prima bozza del materiale a stampa, un prototipo anche imperfetto del software, consentendone una prima sgrossatura che può risparmiare molto lavoro successivo, per non parlare di eventuali effetti negativi su un numero elevato di studenti.

Il terzo metodo di valutazione è la sperimentazione su scala pilota (o "prove sul campo"), ossia l'uso di una versione completa e autoconsistente del prodotto con un campione significativo della popolazione obiettivo. Questo tipo di esperimenti va effettuato in ambiente reale minimizzando le interferenze con le dinamiche del processo, in quanto le informazioni che ne dovrebbero emergere riguardano un po' tutti gli aspetti legati all'efficacia didattica del courseware: piacevolezza e adeguatezza dell'approccio proposto, fattibilità in condizioni reali, tempi e risorse effettivamente necessari, etc. Alcuni autori [Rowntree, 1986] distinguono tra le "prove su scala pilota" e le "prove sul campo", differenziandole in base agli obiettivi, ma poichè nella pratica è spesso possibile condurle in parallelo, nel corso di una unica sperimentazione, la maggior parte degli autori tende a confondere i due termini e ad utilizzarli come sinonimi.

Nel seguito faremo riferimento a due approcci distinti alla valutazione di sistemi didattici: l'approccio qualitativo e quello quantitativo. [Rowntree, 1992] ne sintetizza efficacemente la differenza, in termini di obiettivi: "Mentre nell'approccio quantitativo, in sostanza, si cerca una risposta alla domanda 'In che misura gli obiettivi sono stati raggiunti?', in quello qualitativo ci si chiede, più in generale 'Cosa è successo?'"

In altre parole, l'approccio qualitativo si

basa sull'osservazione delle dinamiche formative e sullo studio di tutto ciò che avviene durante il processo didattico, delle variabili in gioco e dei risultati ottenuti, senza cercare di misurare i cambiamenti, ma piuttosto studiando il sistema nel suo complesso. Al contrario, l'approccio quantitativo tende a privilegiare i dati oggettivi e quantificabili. Nonostante spesso i sostenitori di un approccio siano anche detrattori dell'altro, i due modi di pensare sono tutt'altro che incompatibili, nel senso che, come spesso succede, un buon compromesso tra i due orientamenti può forse costare più fatica, ma spesso fornisce un quadro più completo e attendibile. L'approccio qualitativo consente spesso di comprendere meglio il processo didattico, cogliendo tra l'altro eventuali dinamiche impreviste, mentre a quello quantitativo va riconosciuto soprattutto il vantaggio di una maggior oggettività.

L'analisi dei dati

A seconda della natura dei dati disponibili (qualitativi o quantitativi, o entrambi) il compito di analizzare ed interpretare i dati è metodologicamente differente, ma comunque complesso, e decisamente critico. È infatti noto come sulla base degli stessi dati, anche numerici, siano possibili le deduzioni più disparate. Ad esempio, supponiamo che in una qualche fase della fruizione di un courseware i risultati conseguiti dagli studenti siano mediamente peggiori che nelle altre. Ciò può significare sia che il materiale didattico relativo a quella fase è meno efficace, oppure meno chiaro, o meno interessante, o più difficile, o ancora che qualche fattore "esterno" ne ha disturbato la fruizione. Sarà solo considerando i dati nella loro globalità che si potrà cercare di selezionare una o più interpretazioni ragionevoli sulla base delle quali stabilire il processo di compensazione più adeguato. Quanto ai metodi di analisi, se nell'approccio quantitativo la statistica fornisce dei metodi abbastanza formalizzati, in quello qualitativo si procede in genere in modo euristico. In entrambi i casi comunque, la rappresentazione dei dati e la loro interpretazione richiedono esperienza e sensibilità, oltre a una buona dose di buon senso. Pur non esistendo metodi generali o procedure comunque applicabili, la letteratura tende a suggerire l'adozione di un approccio sistematico, basato su possibili modi di organizzare i propri dati, di rappresentarli

(graficamente o meno), di identificare punte o ricorrenze, di cercare conferma alle proprie interpretazioni e valutarne il grado di generalizzabilità [Riley, 1990].

La compensazione

Il processo di compensazione segue logicamente dalla interpretazione dei dati e dalle conclusioni tratte in fase di analisi. A seguito di qualunque modifica è opportuno verificare, con appositi esperimenti, che le modifiche abbiano prodotto gli effetti desiderati. Il compimento di questo ciclo funge ovviamente da verifica della validità dell'intero processo.

Importanza della validazione

I metodi di validazione appena descritti, sia pure schematicamente, forniscono informazioni complementari o di conferma reciproca, e di conseguenza non sono da utilizzarsi in maniera mutuamente esclusiva. Purtroppo però un intervento di validazione svolto in maniera completa richiede un notevole investimento in termini di risorse umane ed economiche e può causare considerevoli dilatazioni dei tempi di sviluppo. Tale investimento viene spesso considerato inaccettabile dai produttori di courseware, e questo li porta a trascurare o minimizzare la fase di validazione. Ma la causa di fondo di questa trascuratezza è, a parere dell'autrice, l'ignoranza o almeno la sottovalutazione dei vantaggi (a breve e lungo termine) che la validazione può comportare. Tale ignoranza, nell'ambito professionale dei produttori di courseware, è al tempo stesso causa ed effetto dell'esclusione di persone con competenze specifiche dai gruppi che si dedicano allo sviluppo (competenze, si noti, non così facili da reperire). Per quanto riguarda il mercato del courseware, d'altro canto, la stessa carenza di informazioni fa sì che la richiesta non tenga debitamente conto dell'importanza del fatto che un prodotto sia stato validato.

STRUMENTI PER LO SVILUPPO DI MATERIALE DIDATTICO

All'inizio degli anni '80 gli strumenti per lo sviluppo di courseware consistevano per lo più nei cosiddetti "linguaggi autore" e "sistemi autore", sviluppati allo scopo di alleviare le fatiche di chi implementa software didattico, o addirittura di affrancarlo dalla necessità di conoscere un linguaggio di pro-

grammazione. Col tempo, per ciascuna delle fasi del processo di sviluppo del courseware (progetto, realizzazione e validazione) sono stati studiati strumenti concettuali o metodologici utilizzabili dagli attori del processo in una o più delle attività sopra menzionate. In alcuni casi gli strumenti concettuali hanno anche ispirato la realizzazione di strumenti software che costituiscono di fatto dei veri e propri "personal productivity tools" per la produzione di courseware. Numerosi esempi di questi strumenti sono riportati in letteratura [Persico et al, 1985, Van Marcke, 1992, Chiocciariello et al, 1990, Schmidt, 1993]. Nella stessa direzione si sono mossi anche alcuni progetti di ricerca internazionali, finanziati nell'ambito del programma DELTA della comunità Europea: il progetto OSCAR (Open System for Collaborative Authoring and Reuse, [Ulloa, 1994]) e il progetto DISCourse (Design and Interactive Specification of Courseware [Mispelkamp et al, 1994]). Obiettivo di questi progetti è la costruzione di veri e propri sistemi integrati costituiti appunto da applicazioni capaci di supportare sia le attività di progettazione di courseware sia quelle di realizzazione. Ciò che gli autori producono con l'ausilio di tali applicazioni entra così a far parte di una base di conoscenze (Common Information Space) comune all'intero progetto di sviluppo, garantendo la coerenza e la sincronizzazione nell'operato dei vari attori del processo. Per fare un esempio, qualora si desideri sviluppare un sistema "intelligente" per l'addestramento di tecnici nel campo dell'aeronautica (settore di riferimento del progetto OSCAR), la piattaforma congiunta OSCAR+DISCourse consente all'esperto del dominio di produrre una rappresentazione di contenuti, a un altro autore di definire o scegliere da un archivio già esistente le opportune strategie didattiche, a un terzo di progettare e costruire un opportuno modello dello studente, a un quarto di realizzare delle sequenze didattiche multimediali, il tutto fruibile sotto forma di un unico sistema capace di selezionare strategie e materiali didattici sulla base del comportamento dell'utente/studente. La collaborazione tra i vari attori del processo di sviluppo è controllata e gestita da una applicazione di "management" di progetto e realizzata tramite un software per la comunicazione e di supporto alle decisioni.

Anche i sistemi di supporto alla realizzazio-

ne hanno subito una notevole evoluzione. Iper-testi, multimedia, sistemi per la realizzazione di interfacce grafiche, sono solo alcune parole chiave indicative sia di un notevole miglioramento qualitativo (dal punto di vista tecnico) del software sviluppabile, sia di una considerevole semplificazione del compito realizzativo. In maniera del tutto analoga, anche la realizzazione di materiale a stampa è diventata più semplice e veloce grazie alla sempre maggior diffusione e alle aumentate potenzialità dei sistemi per l'editoria elettronica.

La validazione è, per ovvi motivi, la fase più difficilmente automatizzabile. Gli strumenti utilizzabili variano in base al metodo e all'approccio che si intende utilizzare, e vengono di volta in volta messi a punto dai conduttori del processo. La valutazione soggettiva, ad esempio, può essere supportata con questionari o con liste di aspetti da valutare. L'apprendimento da parte degli studenti può essere verificato nel corso di interventi sperimentali con test di apprendimento (automatici o meno). Per verificare l'impatto del courseware nel dominio "affettivo" (le opinioni, gli atteggiamenti, il gradimento) possono essere usati questionari ad hoc. L'osservazione delle dinamiche di apprendimento nel corso delle sperimentazioni è anche uno strumento fondamentale di verifica: può avvenire in maniera diretta (i valutatori presenti sul luogo in cui l'apprendimento avviene) oppure indiretta (i docenti possono scrivere rapporti, insegnanti e studenti possono essere intervistati o anche ripresi con un filmato da esaminare successivamente). Per valutare il software, si possono infine mettere a punto delle "spie di percorso" che consentano di riesaminare gli itinerari seguiti dagli studenti nell'interazione col programma.

La pianificazione della valutazione e la predisposizione degli strumenti necessari sono attività estremamente delicate in quanto occorre prestare attenzione a numerosi fattori quali il rischio di interferenze del processo valutativo con quello da valutare, la difficoltà di formulare quesiti non ambigui e adeguati ad appurare univocamente le informazioni desiderate, che non favoriscano o suggeriscano risposte insincere [Schofield, 1978; Moore et al, 1973].

Nonostante la notevole evoluzione degli strumenti per lo sviluppo di courseware e la parziale automazione subita dal processo, la produzione di materiale didattico è ancora

una attività estremamente costosa, soprattutto dal punto di vista delle risorse umane e del tempo necessario. Di conseguenza, è più che mai fondamentale evitare che gruppi di lavoro diversi "re-inventino la ruota" in circostanze diverse. Al contrario, sarebbe auspicabile far sì che ogni progetto di sviluppo possa sfruttare e riutilizzare al meglio materiali ed esperienze altrui. È proprio in questa direzione che sono orientate le ricerche sulla cosiddetta "riusabilità del courseware"⁵. Il Common Information Space a cui abbiamo già accennato sopra consiste ad esempio in un archivio di materiale (tipicamente documenti di progetto e realizzativi, nonché materiale didattico di tipo testuale o multimediale) "interrogabile" attraverso opportune applicazioni di interfaccia utilizzando come chiavi di accesso i contenuti, le strategie didattiche, i media impiegati, e in generale le caratteristiche fisiche e pedagogiche dell'oggetto cercato. Ovviamente, la reperibilità è condizione necessaria ma non sufficiente ad ottenere la riusabilità del materiale. Una volta identificato un insieme di materiali ai fini del riuso, sarà necessario adattarli al nuovo contesto didattico. Questa operazione di decontestualizzazione ed inserimento di un oggetto in un nuovo ambito può senza dubbio essere facilitata da sofisticati strumenti di editing, ma dipende anche dalle modalità con cui tale oggetto è stato progettato. In altre parole, il courseware può essere progettato con criteri che favoriscano la riusabilità dei vari prodotti del processo di sviluppo [Persico et al, 1993].

IL MODELLO A SPIRALE APPLICATO AL COURSEWARE

Se il modello a cascata per lo sviluppo di courseware si ispirava ad una metodologia, allora generalmente accettata, di sviluppo di software, è ragionevole chiedersi se l'evoluzione avvenuta nel settore del software engineering e i nuovi modelli proposti in questo ambito possono ancora adattarsi allo sviluppo di courseware. Prendiamo ad esempio in considerazione il metodo cosiddetto "a spirale", orientato alla produzione rapida di prototipi che possono essere sottoposti a validazione da parte dell'utente finale e conseguentemente modificati in modo estremamente flessibile. L'esistenza di sistemi sofisticati per la produzione di ipermedia ha in qualche modo accorciato le distanze tra spe-

⁵ Per ulteriori dettagli vedi scheda a pagina 14.

cifiche ed implementazione, consentendo di sviluppare “specifiche animate”, o addirittura prototipi funzionanti, che rendono l’idea del risultato atteso più di un tradizionale documento di specifica o di progetto. Contemporaneamente, la tendenza a riusare materiale sviluppato in altri contesti stimola talvolta a percorrere il ciclo di vita del courseware tenendo un occhio al risultato finale, cercando di convergere su un progetto facilmente integrabile con quanto già disponibile.

Alla luce di quanto sopra dunque, il modello a spirale può essere applicato allo sviluppo di courseware, purché non venga inteso come un modo di “saltare” le fasi di analisi e progetto del sistema didattico, per passare direttamente alla sua implementazione, ma piuttosto come un modo per rendere lo sviluppo più flessibile ed economico, integrandolo più naturalmente con le attività di valutazione. Sulla base dello stesso modello, si può procedere alla messa a punto di singole componenti del sistema, attraverso attività parallele di autori diversi. Si pensi alla realizzazione di un courseware basato su materiale a stampa e su un ambiente software. Una volta definiti requisiti e specifiche nel loro insieme, un gruppo di autori può affrontare la stesura di una dispensa per lo studente, magari a partire da un capitolo campione, mentre altri possono procedere alla implementazione di un embrione del software, eventualmente incompleto, ma in qualche modo “utilizzabile”. Il materiale così prodotto può essere provato con un campione, anche numericamente ridotto, della popolazione obiettivo, e messo a punto sulla base del feedback così ricevuto. In quest’ottica, il modello a spirale altro non è che un ripercorrere la cascata del modello originario più volte, in parte per tutto il sistema, e talora per alcune sue componenti.

CONCLUSIONI

Da quanto detto emerge chiaramente il fatto che la produzione di materiale didattico, in generale, richiede una molteplicità di competenze e un impiego di risorse tale da non potersi ridurre all’attività di un singolo individuo. Ciò non significa ovviamente che non sia possibile, per un insegnante, produrre artigianalmente una dispensa o un ipertesto per i propri studenti, ma semplicemente che lo sviluppo di courseware destinato ad avere

un certo peso e una buona diffusione nel contesto per cui è stato progettato richiede un approccio sistematico che preveda tra l’altro la collaborazione tra diversi attori, quale che sia la natura del materiale che si intende sviluppare.

Gestire lo sviluppo di courseware in modo tale da valorizzare al massimo le competenze di diversi autori e integrarne armonicamente i contributi favorisce il mantenimento di un rapporto costi/benefici accettabile, evitando ai singoli di scontrarsi con problemi che esulano dalle loro competenze e incanalando produttivamente fantasia e creatività.

Le attività tipiche delle fasi alte dello sviluppo (analisi del contesto, identificazione e strutturazione dei contenuti, ideazione e scelta delle strategie didattiche più adeguate, etc) sono estremamente complesse e delicate, richiedono il contributo di competenze didattiche, disciplinari e metodologiche e sono difficilmente supportabili con sistemi automatici. Nonostante la difficoltà del compito, è proprio in questa direzione che si sta muovendo un consistente filone di ricerca in Europa, con l’obiettivo di creare sistemi che facilitino la collaborazione di attori diversi e li aiutino a svolgere anche attività afferenti alle fasi alte del ciclo di vita del courseware. Per quanto riguarda il software, in particolare, una esigenza sempre più frequente è produrre sistemi capaci di effettuare scelte strategico/didattiche a seconda del comportamento e delle caratteristiche intellettuali dell’utente e degli obiettivi cognitivi. La realizzazione di sistemi di questo tipo accentua la complessità del processo di sviluppo e comporta a maggior ragione la necessità di strumenti di supporto alle fasi di analisi e progetto di moduli che sono dei veri e propri sistemi esperti nella didattica dei contenuti, nel dominio di contenuti da insegnare e nella modellizzazione dinamica dell’utente. È chiaro che la realizzazione di un sistema didattico di questa complessità non è impresa di tutti i giorni, e comporta investimenti in risorse umane ed economiche decisamente significativi. Un’altra direzione in cui la ricerca si sta muovendo è appunto quella di contenere i costi dello sviluppo di courseware grazie al principio del “riuso”. Anche il concetto di riuso è stato preso a prestito dal mondo dell’ingegneria del software, e si basa sull’idea di progettare componenti di sistemi didattici così generali e astratti da po-

ter essere riutilizzati più volte in contesti diversi. Tali componenti, siano essi porzioni fisiche di courseware fruibili da uno studente, oppure semi-lavorati prodotti nelle fasi alte del ciclo di vita (come ad esempio una

rappresentazione dei contenuti) possono essere immagazzinati in un database accessibile a tutti gli attori del processo di sviluppo e resi facilmente reperibili ai fini di essere utilizzati più volte al prezzo di lievi modifiche.

Riferimenti Bibliografici

Chiocciariello A., De Michele M., Maryni P. G., Sarti L. (1990), *X-Graph. Studio e realizzazione di un ambiente per la manipolazione di grafici*, Rapporto Inter-no/ITD 3/90

Dick W., Carey L. (1990), *The systematic design of instruction* (3rd edn) Harper Collins, Glenview, Illinois

Farquhar J.D., Surry D.W. (1994), *Adoptional analysis: an additional tool for instructional developers*, ETTI, Vol.31, n.1

Ferraris M., Midoro V., Olimpo G. (1984 a), *Instructional system design and software system design: a unifying approach*, Journal of Structured Learning, Vol. 8

Ferraris M., Midoro V., Olimpo G. (1984

b), *Petri Nets as a modelling tool in the development of CAL Courseware*, Computers and Education, Vol. 8, n. 1

Ferraris M., Midoro V., Olimpo G. (1985) *Il computer nella didattica*, S.E.I., Torino

Hannafin M.J., Peck K.L. (1988), *The design, development, and evaluation of Instructional software*, Macmillan, New York

Mispelkamp H., Sarti L., (1994), *DISCourse: Tools for the design of learning material*, in de Jong T., Sarti L. (eds) *Design and production of multimedia and simulation-based learning material*, Kluwer Academic Publishers

Moore B., Nuttal D., and Willmott A. (1973), *Data Collection*, The Open University Press, Walton

Hall Bletchley Buckinghamshire, Great Britain

Persico D. et al (1993), *Specifications of the Reusability Services*, Deliverable 10, DELTA Project D2006 OSCAR

Persico D., Ferraris M., Midoro V., Olimpo G., Sissa G. (1985) *Diagnostic evaluation in the learning process*, in Duncan K. and Harris D. (eds) *Proc. of 4th WCCE*, Norfolk, VA, North Hollan, Amsterdam

Riley J. (1990), *Getting the most from your data*, Billings and Sons Ltd, Worcester

Romiszowski A.J. (1984), *Producing instructional systems*, Kogan Page, London

Rowntree D. (1986), *Teaching through self-instruction*, Kogan Page

Rowntree D. (1992), *Exploring Open and Distance Learning*, Kogan Page, London

Schmidt C., Windmueller W. (1993), *Domain Authoring Tool Specification*, Deliverable 20, DELTA Project D2008 DISCourse

Schofield H. (1978) *Valutazione e uso dei test*, La Nuova Italia Editrice, Firenze

Ulloa A., (1994) *Open System for Collaborative Authoring and Reuse*, in de Jong T., Sarti L. (eds) *Design and production of multimedia and simulation-based learning material*, Kluwer Academic Publishers

Van Marcke K. (1992), *Instructional expertise*, Proc. of Second International Conference, ITS '92, Springer-Verlag, Berlino