



“Learning bricks”: oggetti riusabili per simulazioni efficaci

Vindice Deplano

Consulente di e-learning

v.deplano@tin.it

Parole chiave: simulation, learning object, system dynamics, serious games

Abstract

Le simulazioni costituiscono un metodo formativo particolarmente efficace, ma la loro penetrazione è fortemente limitata, oltre che da vincoli culturali, da fattori legati ai costi elevati che il mercato stenta ad accettare.

I Learning bricks sono un'architettura che punta a ingegnerizzare lo sviluppo di simulazioni interattive, abbassandone drasticamente costi e tempi di sviluppo senza limitarne la qualità. L'intera interfaccia della simulazione (personaggi, oggetti animati, testi, prove di autovalutazione, dialoghi, ecc.) viene costruita con un'operazione di montaggio non troppo dissimile da quella con cui si usano i mattoncini Lego. L'architettura è composta da un modulo base, che contiene le specifiche del motore della simulazione, e da un certo numero di bricks: moduli con funzioni specializzate, completamente parametrizzabili per quanto riguarda il contenuto informativo (testo, immagini, audio, video, animazioni), il collegamento con il motore della simulazione, il comportamento.

1 La simulazione: prospettive e problemi

Oggi quell'insieme di metodi, tecnologie e attività che chiamiamo e-learning ha conquistato un posto stabile nel mondo della formazione. Aziende, pubblica amministrazione e scuola hanno capito che è una via per raggiungere esperienze di apprendimento realmente efficaci, che non si potrebbero ottenere in presenza. E non solo per tagliare i costi.

Dove i progetti di e-learning si sono appiattiti imitando vecchi modelli hanno prodotto risultati deludenti e utenti (e committenti) insoddisfatti. È accaduto fin troppo spesso con i sistemi autodidattici, dove l'utente dovrebbe imparare qualcosa interagendo con un software: la stragrande maggioranza di questi WBT (Web Based Training) si riducono a sfogliapagine elettronici con il solo conforto di un po' di multimedialità (e, quando il budget lo consente, animazioni e filmati). In molti casi si rimpiange la straordinaria facilità d'uso di un buon libro.

Per realizzare qualcosa di davvero utile bisogna fare di più. È qui che entrano in gioco le simulazioni interattive (o serious games) che permettono di integrare l'esperienza diretta nel processo di apprendimento. Vale la pena di ribadire che una simulazione computerizzata non è un pallido surrogato virtuale della realtà vera, ma produce esperienze di apprendimento per molti versi migliori perché aiuta a giocare col mondo, interpretandone le dinamiche, ripetendo più volte gli stessi passaggi, riflettendo per tutto il tempo necessario sulle proprie azioni. E senza correre rischi (meglio, molto meglio, far fallire aziende e distruggere aerei virtuali).

Ci si potrebbe chiedere perché queste meraviglie didattiche sono ancora così poco diffuse nei progetti di e-learning e perché il mercato non le reclama a gran voce invece di ricorrere a modesti sfogliapagine elettronici.

I motivi a mio avviso sono due:

- manca una diffusa cultura della simulazione nei clienti (che spesso la vedono come un gioco) e anche nei produttori;
- le simulazioni hanno un costo molto elevato che il mercato, abituato da molti anni a prodotti spesso mediocri ma economici, non riconosce.

Oggi però la domanda di materiali didattici di qualità sta crescendo. Una crescita che va aiutata producendo vere simulazioni interattive a un prezzo non molto diverso da quello dei comuni WBT sequenziali.

2 Ottimizzare lo sviluppo delle simulazioni

I Learning bricks vogliono essere un contributo in questa direzione. Costituiscono nello stesso tempo un'architettura software e un modo di

pensare alle simulazioni che mira a ridurre drasticamente i costi di sviluppo senza porre alcun limite al livello qualitativo.

Dobbiamo chiederci quali sono gli elementi costitutivi (e, quindi, le voci di costo) di una generica simulazione.

2.1 Struttura di una simulazione

Una simulazione è un sistema software che consente di interagire con un mondo virtuale che astrae alcuni aspetti significativi della realtà. Significa che questo mondo virtuale:

- comunica con l'utente attraverso un'interfaccia grafica/sonora e alcune periferiche (mouse, tastiera e altri dispositivi di input);
- gli consente di effettuare determinate scelte (selezionare opzioni, inserire valori, ecc.);
- reagisce alle scelte in un modo che è sovrapponibile al mondo reale.

Non è certo la definizione più esauriente di simulazione, ma è utile per metterne in evidenza i due elementi costitutivi, implementati nel software:

- Il "motore", che comprende:
 - Gli aspetti della realtà che si vogliono prendere in considerazione (trasformati in un insieme di variabili).
 - Le regole di funzionamento del mondo virtuale (in concreto, è il modo con cui le variabili sono messe in relazione tra loro).
 - Il modello di simulazione, ovvero il tipo di logica che sottende alla definizione delle regole di funzionamento. Dipende dalle teorie a cui il progettista fa riferimento e dalle tecnologie disponibili. Da questo punto di vista, distinguiamo alberi decisionali, automi cellulari, reti bayesiane, dinamica dei sistemi, reti neurali, ecc.
- L'interfaccia, che comprende:
 - Il set di comandi e di dati che il sistema è in grado di riconoscere e interpretare.
 - La metafora comunicativa, quell'insieme di elementi grafici e funzionali che definiscono il modo con cui la realtà "virtuale" della simulazione viene presentata: cruscotto, scrivania, ambiente a due o tre dimensioni, ecc.
 - La forma con cui, all'interno della metafora, viene rappresentato lo stato del sistema: stringhe alfanumeriche, immagini, parole, suoni, filmati, animazioni più o meno complesse.



Fig. 1 . Una simulazione realizzata con i Learning bricks. Gli oggetti cerchiati corrispondono ad altrettanti bricks.

Non esiste necessariamente una corrispondenza diretta tra complessità del motore e interfaccia della simulazione. Tipicamente le simulazioni costruite a scopo di ricerca tendono ad avere motori molto raffinati (perché cercano una corrispondenza molto forte con il segmento di realtà che viene simulato) e interfacce spartane. Al contrario quelle usate per la formazione possono avere motori semplici, ma l'interfaccia deve essere sofisticata. Magari come un videogioco, risorse permettendo.

2.2 Una questione di costi e tempi

Un corollario di questa situazione è che nelle simulazioni formative il costo di sviluppo di un'interfaccia di qualità (e, in generale, del software) è molto più alto di quello necessario per progettare la logica del motore.

Visto da chi giorno per giorno si scontra con le esigenze di aziende e pubblica amministrazione, il problema ha due facce:

- Il costo delle simulazioni di buon livello, vale la pena di ripeterlo, è molto più alto di quanto i clienti sono generalmente disposti a spendere.
- Le risorse impiegate per creare l'interfaccia accattivante richiesta dal cliente non aggiungono nulla alla qualità della simulazione, cioè alla sua capacità di far apprendere con esperienze credibili e coinvolgenti.

I Learning bricks puntano a ridurre drasticamente i costi e i tempi di sviluppo dell'interfaccia. Con tre obiettivi:

- mercato: migliorare la penetrazione delle simulazioni, riducendone i costi in termini assoluti;
- qualità: liberare risorse per la progettazione del "motore";
- gestione dei progetti: facilitare la prototipazione e quindi la comunicazione con il cliente.

3 Architettura dei Learning bricks

L'idea di base è concettualmente semplice: realizzare simulazioni a partire da oggetti riusabili e parametrizzabili (i bricks), facilmente combinabili tra loro, che gestiscano l'interfaccia con grafica, multimedialità, animazioni. Come i mattoncini Lego con cui produciamo velocemente costruzioni complesse montando un numero finito di oggetti semplici.

L'architettura dei Learning bricks si compone di due elementi:

- il modulo base;
- i bricks.

3.1 Modulo base

Il modulo base è il direttore d'orchestra in quanto incorpora:

- La funzione di calcolo che implementa il motore della simulazione. Il motore può essere di diverso tipo, poiché qualunque modello di simulazione è riducibile a una serie di calcoli che hanno per oggetto variabili e relazioni tra variabili. La funzione è richiamata ogni volta che lo stato del mondo simulato deve essere ricalcolato in seguito a un evento: scorrere del tempo, decisione dell'utente, esito di una prova di verifica, ecc.
- Il caricamento dei bricks necessari alla gestione dell'interfaccia, con il relativo passaggio di parametri.
- Il caricamento delle immagini di sfondo che consentono di realizzare simulazioni con più ambienti (per passare da un ambiente all'altro si usa un particolare brick "Porta"- vedi oltre).
- Alcune funzioni di utilità generale, richiamabili da tutti i bricks.

Le operazioni di sviluppo della simulazione che comportano la scrittura di righe di codice si limitano a inserire nel modulo base.

- le istruzioni contenute nella funzione di calcolo.
- i parametri dei bricks utilizzati.

Le operazioni di sviluppo della simulazione che comportano la scrittura di righe di codice si limitano a inserire nel modulo base.

- le istruzioni contenute nella funzione di calcolo.
- i parametri dei bricks utilizzati.

Il modulo base e i bricks sono sviluppati in Flash, esclusivamente tramite ActionScript. Chi aprisse i file sorgente troverebbe un solo fotogramma sulla linea temporale e nessun elemento precaricato sullo stage e in libreria.

3.2 I bricks

I bricks sono moduli software che hanno:

- una funzione specifica nell'interfaccia della simulazione.
- un collegamento con una qualunque delle variabili che determina il funzionamento del motore (variabile di riferimento).
- un contenuto comunicativo (testo, immagini, audio, video, animazioni) personalizzabile tramite parametri di configurazione.
- una comportamento personalizzabile nei limiti previsti dalla funzione specifica.

Per descrivere meglio lo scopo e il funzionamento di un qualunque brick, usiamo come esempio il Personaggio:

- Funzione. Parlare (o cantare, se si vuole), con un duplice movimento animato: occhi aperti e chiusi (sempre), bocca aperta e chiusa (in concomitanza con il parlato).
- Collegamento con la variabile di riferimento. In funzione del valore della variabile (o di un intervallo di valori), può pronunciare diversi commenti, assumere una determinata postura o cambiare posizione nello schermo.
- Contenuto comunicativo. I parametri di configurazione comprendono un elenco di file sonori e un elenco di file grafici (posture del personaggio).
- Comportamento. A seconda del valore della variabile di riferimento il personaggio può essere attivato o disattivato, comparire o scomparire (anche con effetti di dissolvenza).

Se sono necessari più personaggi, il brick "Personaggio" può essere chiamato più volte con parametri di configurazione diversi.

La Tabella 1 presenta un elenco dei bricks già realizzati.

4 Uso dei Learning bricks

È importante notare che i bricks sono moduli totalmente indipendenti l'uno dall'altro perché si collegano esclusivamente con le variabili che fanno parte della funzione di calcolo (cioè del motore). Così è possibile inserire un numero illimitato di oggetti animati per immettere dati, visualizzare lo stato delle variabili, effettuare test di autovalutazione, presentare informazioni sotto forma di schede o mini WBT animati, ecc.

TABELLA 1
I Bricks

| Nome | Funzione | Particolarità |
|-------------------|--|--|
| PERSONAGGIO | PARLARE O EMETTERE SUONI (ATTIVANDO FILE MP3) DIVERSIFICATI IN FUNZIONE DEL VALORE DELLA VARIABILE DI RIFERIMENTO. | POSSIBILITÀ DI CAMBIARE POSTURA O POSIZIONE NELLO SCHERMO. |
| TUTORIALE | PRESENTARE SCHEDE INFORMATIVE CON TESTO E IMMAGINI IN MOVIMENTO E AUDIO SINCRONIZZATO. | SI ATTIVA CLICCANDO SU UN OGGETTO SENSIBILE. |
| TUTORIALE ESTERNO | PRESENTARE WBT (CON NAVIGAZIONE AVANTI/INDIETRO) CON PAGINE COSTITUITE DA ANIMAZIONI FLASH (SWF). | SI ATTIVA CLICCANDO SU UN OGGETTO SENSIBILE. |
| INPUT | IMMETTERE DA TASTIERA VALORI DA ASSEGNARE A UNA VARIABILE. | SI ATTIVA CLICCANDO SU UN OGGETTO SENSIBILE. |
| TEST | SOMMINISTRARE DOMANDE DI AUTOVALUTAZIONE CON FEEDBACK E ASSEGNAZIONE DI PUNTEGGI. | SI ATTIVA CLICCANDO SU UN OGGETTO SENSIBILE. |
| TESTO | PRESENTARE TESTI DIVERSIFICATI IN FUNZIONE DEL VALORE DELLA VARIABILE DI RIFERIMENTO. | SI ATTIVA CLICCANDO SU UN OGGETTO SENSIBILE. |
| SIPARIETTO | PRESENTARE BREVI TESTI ANIMATI DIVERSIFICATI IN FUNZIONE DEL VALORE DELLA VARIABILE DI RIFERIMENTO. | SI ATTIVA AUTOMATICAMENTE. |
| VIDEO | PRESENTARE FILMATI (FORMATO FLV) DIVERSIFICATI IN FUNZIONE DEL VALORE DELLA VARIABILE DI RIFERIMENTO. | SI ATTIVA CLICCANDO SU UN OGGETTO SENSIBILE. |
| PERSONAGGIO VIDEO | PRESENTARE OGGETTI ANIMATI 3D O FILMATI CON AUDIO DIVERSIFICATI IN FUNZIONE DEL VALORE DELLA VARIABILE DI RIFERIMENTO. | UTILIZZANDO FILMATI IN CHROMA KEY È POSSIBILE INSERIRE PERSONAGGI UMANI NELL'AMBIENTE. |
| DIALOGO | PRESENTARE UN DIALOGO TRA PIÙ PERSONAGGI, CON AUDIO E DIDASCALIE. | È POSSIBILE CAMBIARE INQUADRATURA A OGNI BATTUTA DEL DIALOGO. I PERSONAGGI PRESENTANO ANIMAZIONI NEGLI OCCHI E, QUANDO PARLANO, NELLA BOCCA. |
| CALENDARIO | PRESENTARE LA DATA (GIORNO, MESE, ANNO). | |
| PORTA | CONSENTIRE IL PASSAGGIO DA UN AMBIENTE ALL'ALTRO CLICCANDO SU UN'IMMAGINE | CONSENTE DI REALIZZARE SIMULAZIONI CON PIÙ AMBIENTI. |

4.1 Sviluppo di simulazioni

Usando i Learning bricks lo sviluppo di una simulazione, di qualunque complessità, prevede quattro tipi di operazioni (vedi lo schema di figura 2):

- Definizione di un sistema logico-matematico (variabili e loro relazioni) che ricalchino il funzionamento di alcuni aspetti del mondo reale, con criteri che variano in funzione del modello di simulazione prescelto.
- Traduzione del sistema in codice ActionScript all'interno della funzione di calcolo.
- Sviluppo o selezione da una libreria di testi, elementi grafici (sfondi, oggetti, pulsanti) e componenti multimediali (audio, clip video, ecc.).
- Richiamo dei bricks necessari all'interfaccia utente e definizione dei relativi parametri, inserendoli direttamente in un file ActionScript e/o in file XML esterni.

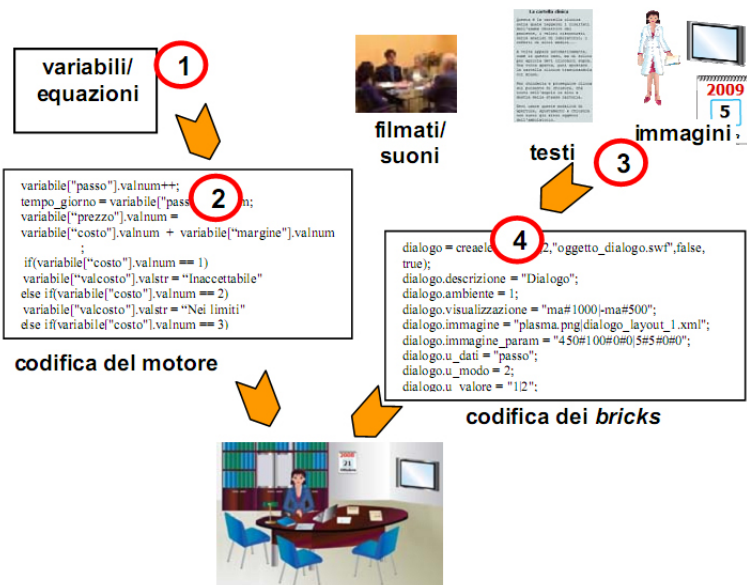


Fig. 2 . Schema dello sviluppo di una simulazione

In concreto le operazioni 1 e 3 si svolgono con modalità tradizionali, perché precedono la fase di implementazione del modello e del montaggio. Il tempo di realizzazione può essere anche molto lungo quando vengono progettate simulazioni complesse con decine o centinaia di variabili.

L'operazione 2 consiste nella traduzione del modello logico matematico in istruzioni ActionScript. Richiede un operatore con conoscenze di base del linguaggio e un tempo variabile da qualche decina di minuti ad alcune ore. Si tratta di un'operazione che necessita di attenzione, ma non risulta particolarmente complessa.

L'operazione 4 è determinante per l'interfaccia, l'interattività e quindi l'esperienza del fruitore. Ciascun brick è inserito attraverso una serie di istru-

zioni del tipo:

oggetto.parametro = valore.

A titolo di esempio, la tabella 2 elenca i parametri necessari per l'oggetto Video, che può gestire uno o più videoclip.

TABELLA 2
Parametri dell'oggetto "video"

| Parametro | Funzione | Tipo di dato |
|---------------------|--|---------------------|
| RICALCOLO | RICHIAMO AUTOMATICO DELLA FUNZIONE DI CALCOLO ALLA CHIUSURA DELL'OGGETTO. | TRUE/FALSE |
| INIBIZIONE OGGETTI | INIBISCE L'ATTIVAZIONE DI ALTRI OGGETTI FINO ALLA CHIUSURA DEL VIDEO. | TRUE/FALSE |
| DESCRIZIONE | TESTO DEL TOOLTIP CHE COMPARE SFIORANDO L'OGGETTO COL MOUSE. | TESTO |
| AMBIENTE | "STANZA" IN CUI L'OGGETTO È PRESENTE | CODICE NUMERICO |
| VISUALIZZAZIONE | Effetti (DISSOLVENZA, MOVIMENTO, ECC.) CON CUI L'OGGETTO SI APRE E SI CHIUDE. | CODICI ALFANUMERICI |
| IMMAGINI | FILE RELATIVI ALL'OGGETTO CLICCABILE E ALLO SFONDO DEL VIDEO APERTO. | NOMI DI FILE |
| PARAMETRI IMMAGINI | POSIZIONE SULLO SCHERMO E DIMENSIONI DELLE DUE IMMAGINI. | CODICI NUMERICI |
| PULSANTI | FILE IMMAGINE RELATIVI AI PULSANTI: CHIUDI, PLAY, PAUSA, STOP, ECC. | NOMI DI FILE |
| PARAMETRI PULSANTI | POSIZIONE E DIMENSIONI DEI FILE RELATIVI AI PULSANTI. | CODICI NUMERICI |
| VARIABILE | VARIABILE DI RIFERIMENTO CHE DETERMINA IL COMPORTAMENTO DELL'OGGETTO | NOME DI VARIABILE |
| VALORE | VALORI SIGNIFICATIVI DELLA VARIABILE DI RIFERIMENTO CHE DETERMINANO LA COMPARSA DI UN PARTICOLARE FILMATO E/O L'ATTIVAZIONE AUTOMATICA DEL VIDEO | DATI ALFANUMERICI |
| CONTENUTO | ELENCO DI FILE RELATIVI AI FILMATI. | NOMI DI FILE |
| PARAMETRI CONTENUTO | POSIZIONE E DIMENSIONI DEL FILMATO IN RELAZIONE ALLO SFONDO DEL VIDEO. | CODICI NUMERICI |
| TITOLI | ELENCO DEI TITOLI DEI FILMATI | TESTO |
| PARAMETRI TITOLO | STILE DEL TITOLO: CARATTERE, DIMENSIONI, GRASSETTO, ECC. | CODICI ALFANUMERICI |

| Parametro | Funzione | Tipo di dato |
|-----------|---|---------------------|
| comandi | AZIONI ASSOCIATE A VALORI DELLA VARIABILE DI RIFERIMENTO: ATTIVAZIONE DEL FILMATO, SCOMPARSA O DISABILITAZIONE DELL'OGGETTO CLICCABILE. | codici alfanumerici |
| SUONI | Effetti sonori associati all'avvio del film, alla sua chiusura, ecc. | CODICI NUMERICI |

La maggior parte dei parametri, quelli che descrivono gli aspetti estetici e i comportamenti di base, valgono per tutti gli oggetti Video della simulazione (o di una collana di simulazioni con interfaccia grafica simile).

Gli oggetti più complessi presentano meno parametri perché gli aspetti dinamici del comportamento sono descritti da file XML esterni. Per esempio l'oggetto Test, che presenta una batteria di domande, richiede per ciascuna di esse un file con:

- testo della domanda;
- opzioni di risposta;
- risposta esatta;
- testo della soluzione;
- testi del feedback per la risposta esatta e per quelle errate.

Un programmatore è in grado di costruire uno scenario nuovo con una decina di bricks in un tempo non superiore alle due ore. Il tempo si riduce notevolmente (anche della metà) se lo scenario presenta oggetti già usati in precedenza.

4.2 Competenza

Al momento l'uso dei Learning bricks richiede la competenza di un programmatore junior o comunque un addestramento sufficiente a lavorare direttamente nel codice.

Per i non programmatori (e per gli stessi progettisti delle simulazioni) sarebbe utile un editor con un'interfaccia amichevole che permetta di implementare il motore e richiamare i bricks senza programmare.

Ovviamente se una particolare simulazione richiede funzioni non presenti in nessuno dei bricks disponibili è necessario un programmatore senior per svilupparne uno ex novo. Brick che da quel momento sarà disponibile per i progetti futuri facilitando il recupero dell'investimento. La libreria di "mattoni" disponibili è, quindi, destinata a crescere nel tempo.

4.3 Simulazioni e altri scenari

I Learning bricks sono stati pensati per simulazioni il cui motore è costituito da equazioni sul modello della dinamica dei sistemi (Senge, 1990). Un esempio di questo genere di simulazioni, realizzato con metodi tradizionali, è Impresa oltre i confini¹.

Tuttavia, questa architettura si può applicare anche a:

- Simulazioni con motori basati su alberi decisionali o di tipo probabilistico (al momento non sembra adattabile facilmente ad automi cellulari e reti neurali).
- Learning objects più semplici, costruiti attorno alla narrazione di una storia (qui la variabile guida del comportamento degli oggetti è il tempo).
- Learning objects dalla classica struttura sequenziale. Qui, ovviamente, i Learning bricks non aggiungono nulla a una metodologia didattica non particolarmente evoluta ma permettono di ottenere prodotti più articolati e gradevoli per l'utente.

5 Learning bricks e sistemi autore

Escludendo i moltissimi editor di WBT e/o di test di valutazione è possibile creare velocemente simulazioni usando alcune categorie di strumenti disponibili sul mercato:

- Registratori di schermo, che memorizzano schermate, movimenti di cursore, digitazioni, ecc. con l'aggiunta di audio, testi, prove di valutazione, ecc. Sono utilissimi per simulare i software applicativi. Il più famoso è Captivate di Adobe.
- Strumenti destinati a simulazioni settoriali: geometria, circuiti elettronici, ecc. Un esempio è l'ottimo Geogebra².
- Editor di simulazioni all'interno di schemi predefiniti (per esempio dialoghi con un interlocutore virtuale su determinati argomenti). Un esempio è stato realizzato all'interno del progetto europeo SiSiNe³.

I Learning bricks, tuttavia, non sono nati da un modello già noto. Per questo non è possibile fare confronti con questi sistemi autore.

Conclusioni

I Learning bricks hanno l'obiettivo immediato di abbassare i costi delle simulazioni interattive. Alcuni progetti in corso stanno dimostrando la possibilità

¹ http://www.vindice.it/demo/impresa_oltre_confini/main.htm

² <http://www.geogebra.org>

³ <http://www.nac.unina.it/sisine/>

di ridurre drasticamente la complessità del lavoro. Ma il fine ultimo è un altro: facilitare quel salto metodologico che porti a fare delle simulazioni interattive una componente essenziale di qualunque sistema autodidattico realmente efficace.

Per questo, sarà necessario che il sistema evolva continuamente:

- realizzando nuovi bricks e ottimizzando di quelli esistenti con l'aggiunta di funzioni evolute;
- sviluppando un editor per non programmatori;
- creando di simulazioni accessibili, in base alla legge 4/2004.

Questa è probabilmente la sfida più complessa, che però è un passaggio obbligato per poter impiegare vere simulazioni nella pubblica amministrazione e nella scuola.

BIBLIOGRAFIA

- Bonaiuti G. (a cura di), (2006), *E-learning 2.0*, Trento, Erickson.
- Cross J. (2006), *Informal Learning: Rediscovering the Natural Pathways That Inspire Innovation and Performance*, Essential Knowledge Resource, Pfeiffer.
- Downes S. (2005), *E-learning 2.0*, URL: <http://www.elearnmag.org/subpage.cfm?section=articles&article=29-1> (verificato 20 Agosto 2008).
- Downes S. (2006), *Learning Networks and Connective Knowledge*, URL: <http://it.coe.uga.edu/itforum/paper92/paper92.html> (verificato 20 Agosto 2008).
- Kozlowski T. (2007), *E-learning 1.0 and e-learning 2.0 - Two Sides of the Coin, and how they can blend together*, International e-learning Conference at the Rajabhat Suan Dusit University.
- Landow G. P. (1993), *Ipertesto. Il futuro della scrittura. La convergenza tra teoria letteraria e tecnologia informatica*, Bologna, Baskerville, p. 275.
- Lèvy P. (1996), *L'intelligenza collettiva. Per un'antropologia del cyberspazio*, Milano, Feltrinelli.
- Lubensky R. (2006), *The present and future of Personal Learning Environments (PLE)*, URL: <http://members.optusnet.com.au/rlubensky/2006/12/present-and-future-of-personal-learning.html> (verificato 20 Agosto 2008).
- Mathes A. (2004), *Folksonomies - Cooperative Classification and Communication Through Shared Metadata*.
- Ninck A. (2003), *E-learning: imbuto di Norimberga o illusoria preconfezione digitale?*, Formazione Professionale Svizzera 01-2003.
- O'Reilly, T. (2005), *What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software*. URL: <http://oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> (verificato 20 Agosto 2008).
- Prensky M. (2001), *Digital Natives, Digital Immigrants in On the Horizon*, NCB University Press, Vol. 9 No. 5.

- Trentin G. (2005), *From "formal" to "informal" e-learning through knowledge management and sharing*, Journal of e-learning and Knowledge Society, 1(2), 209-217.
- Vander Wal T. (2007), *Folksonomy Coinage and Definition*, URL: <http://www.vanderwal.net/folksonomy.html> (verificato 20 Agosto 2008).
- Wilson S., Johnson M., Beauvoir P., Sharples P. & Milligan C. (2006), *Personal Learning Environments: Challenging the dominant design of educational systems*, URL: <http://hdl.handle.net/1820/727> (verificato 20 Agosto 2008).
- Deplano V. (2003), *Come valutare i materiali didattici nei progetti di e-learning?*, Roma, Isfol, La qualità dell'e-learning nella formazione continua. Collana I libri del FSE.
- Deplano V. (2004), *Come con Lara Croft*, E-learning & Knowledge Management, 2. 60-63.
- Parisi D. (2001), *Simulazioni*, Bologna, Il Mulino.
- Senge P. (1990), *The Fifth Discipline*, New York, Doubleday.