

# CRETA: un sistema di produzione di percorsi di simulazione per l'apprendimento

Laura Spinsanti, Fosca Giannotti, KDDlab, ISTI CNR, Pisa  
Francesco Varanini, Direttore Scientifico Area e-Business ISTUD - Istituto Studi  
Direzionali

Dino Pedreschi, Vincenzo Gervasi, Salvatore Ruggieri, Antonio Cisternino,  
KDDlab, Dipartimento di Informatica, Università di Pisa

[laura.spinsanti@virgilio.it](mailto:laura.spinsanti@virgilio.it)

[f.giannotti@isti.cnr.it](mailto:f.giannotti@isti.cnr.it)

[fvaranini@iol.it](mailto:fvaranini@iol.it)

[{pedre, gervasi, ruggieri, cisternino}@di.unipi.it](mailto:{pedre,gervasi,ruggieri,cisternino}@di.unipi.it)

## Sommario

CRETA è un sistema di produzione di percorsi di simulazione, utilizzabili in ambito *e-learning*, che consente la scrittura della simulazione direttamente in linguaggio naturale strutturato. L'ambiente è costituito in livelli di astrazione: dal più basso, il motore di simulazione, al più alto, la produzione di *Storyboard* in linguaggio naturale. In questo lavoro, per illustrare CRETA, si utilizza un frammento di storia ambientata nel mondo della Pubblica Amministrazione, utilizzabile per l'apprendimento di metodi del *knowledge management*, ponendo particolare attenzione allo sviluppo di *soft skill* quali la capacità di negoziazione, la capacità di bilanciare tra cooperazione e competizione, l'intuizione del massimo sub-ottimale.

## 1. Introduzione

Questo lavoro costituisce una prima, breve presentazione di CRETA (*CREazione di Tecniche avanzate per l'Apprendimento*) [CRETA], un sistema di produzione di percorsi di simulazione per l'apprendimento di *soft skill*, ed è strutturato come segue. Nella sezione 2 si analizza la necessità di sviluppare percorsi formativi che permettano di acquisire conoscenze non formalizzate e non formalizzabili, ottenibili a nostro parere tramite il mezzo dell'analogia supportata dal *computer game*. Nella sezione 3 si analizzano i livelli in cui è strutturato il progetto CRETA, con particolare attenzione ai due più bassi: il motore di simulazione e il linguaggio di annotazione. La sezione 4 analizza lo sviluppo di un primo frammento di storia e il contesto in cui è stato inserito. La sezione 5 tratta dei prossimi sviluppi e la 6 le conclusioni.

Riteniamo che il progetto sia innovativo da svariati punti di vista: si tratta di un percorso di formazione flessibile e, nella versione definitiva, altamente riutilizzabile in differenti contesti; apre un collegamento tra il mondo dell'*e-learning* e le cosiddette *soft skill*; il percorso formativo può essere "giocato" più volte con esiti differenti intervenendo su alcune variabili del "mondo"; l'analisi metacognitiva dell'esperienza vissuta può, a sua volta, costituire un percorso di apprendimento.

Vi è inoltre l'obiettivo di realizzare un sistema autore ad alto livello per un sistema di scrittura creativa delle storie pensato appositamente per persone che non abbiano speciali competenze informatiche e realizzato il più possibile attraverso uno

Lasciare tre righe vuote.

Lasciare tre righe vuote.

Lasciare tre righe vuote.

*storyboard* guidato in linguaggio naturale.

## 2. L'analogia come percorso formativo

Ciò che appare centrale oggi è un coerente lavoro di acquisizione e scambio di conoscenze (*learning*), e di conservazione e socializzazione delle stesse (*Knowledge Management*) che tenga conto di come il mondo è costruito da visioni soggettive.

Dobbiamo accettare e comprendere la complessità. Questa è la sfida formativa che motiva il ricorso all'analogia [Varanini e Bocchi, 2004].

La complessità pone al centro dell'attenzione, come capacità chiave da alimentare attraverso la formazione, l'attitudine a leggere conoscenza latente, non formalizzata. La capacità di cogliere i 'segnali deboli' all'interno di un discorso apparentemente equilibrato e monocorde. L'abilità di individuare biforcazioni nel cammino, sulla base di labili tracce. La capacità di intercettare i trend: di fronte a diversi eventi, comprendere quali saranno più gravidi di conseguenze.

Di fronte alla complessità non serve un approccio cognitivo – orientato ad elaborare una descrizione articolata del mondo, in base ad un modello di lettura. Serve invece un approccio costruttivo, orientato a creare una passabile ed efficace immagine nel nuovo che sta venendo alla luce.

La rappresentazione del mondo 'ottima' (e quindi 'ottimistica'), che già Voltaire prendeva in giro [Voltaire, 1759], appare, di fronte al mondo che ci troviamo nella necessità di descrivere e comprendere, uno spreco, in termini di tempo e di risorse, ed anzi una pericolosa illusione. E' impossibile realizzare una descrizione esaustiva di un sistema complesso. L'unica mappa esaustiva sarà paradossalmente, come notava Borges, una analogia del mondo, un mappa delle stesse dimensioni del mondo, e capace di accogliere tutti i dettagli del suo aspetto [Borges, 1983]. Ed allo stesso tempo, trattandosi di un sistema vivente ogni descrizione, nel momento della sua formulazione codificata, sarà già e sempre obsoleta.

Non si tratterà perciò di 'apprendere' e quindi applicare tecniche, *praxis*, ma di scoprire, rispecchiandoci in comportamenti altrui il nostro originale modo di rapportarci alla *poiesis*, creatività, produzione.

Proprio per questo, la 'formazione analogica' appare oggi particolarmente efficace. Perché l'apprendimento oggi più utile, o necessario, non risiede nel migliorare la capacità di muoversi in mondi già noti. Oggi, di fronte ai rapidi cambiamenti di scenario imposti dalla ricerca scientifica, dalle nuove tecnologie, dalle dinamiche dei mercati e della finanza; di fronte all'instabilità del sistema ecologico, oggi l'apprendimento più utile, crediamo, riguarda la capacità di esplorare nuovi mondi [Bocchi, 1993].

### 2.1. Computer game e knowledge management

Il *Computer Game* è un mondo con il quale il soggetto interagisce fino ad immergersi, rete di connessioni e di collegamenti fra soggetti, saperi, esperienze, sentimenti, emozioni. Spazio multiforme nel quale il valore aggiunto non è dato dal sapere contenutistico, ma dal saper sapere, da una forma di sapere quadratico dove ciò che è centrale è la capacità dei singoli di ipotizzare sentieri e percorsi del conoscere che non c'erano prima di essere scoperti. Qui la scoperta è creazione, *inventio*.

Il *Computer Game* dunque come esempio estremo di macchina analogica. Non si tratta di apprendere 'contenuti', che sono comunque dati a costi sempre più bassi, resi disponibili dal Web. Ciò che conta è la capacità di attivare nuove connessioni all'interno di una rete del sapere sempre mobile e instabile, ma già data ed accessibile: immaginare un mondo possibile. Ciò che conta, soprattutto, è la capacità di cogliere dal caos il senso emergente: edificare mondi, accrescerne e garantirne l'equilibrio,

accettarne la morte.

Pasquale Gagliardi sottolinea l'esigenza di "programmi nuovi", che esponano manager di diverso livello "ad esperienze educative insolite": "speculazioni squisitamente filosofiche, esperienze estetiche, lezioni di storia". Chiunque vi abbia partecipato resta colpito dallo "straordinario potenziale di estrapolazione analogica, di riflessione critica e di re-invenzione creativa che queste esperienze consentono".

"Problemi quotidiani di gestione affrontati solitamente secondo routine stereotipate" verranno così visti in una luce nuova. Per via analogica, ripensando il nostro agire alla luce di esperienze diverse e lontane da noi, apprenderemo a non limitarci "ad applicare algoritmi", attribuendo invece "importanza alla passione e alla ragione, alla competenza e alla saggezza." [Gagliardi, 2002]. Siamo stati preparati per agire 'secondo routine stereotipate'. E potremmo invece, con molto maggiore profitto individuale e sociale, agire re-inventando creativamente il mondo, apprendere in un luogo non ovvio, non già previsto nei manuali.

L'attenzione è quindi rivolta agli scenari complessi in cui, come nella situazione esperenziale, non conosco il mondo che sto esplorando, ma ho a disposizione solo pochi elementi attraverso i quali devo produrre delle ipotesi e in base a queste compiere delle scelte. Si delinea quindi, per il manager, ma non meno per la formazione continua ad alto livello in genere, l'importanza di un 'allenamento' alla formulazione di inferenze di tipo abducente. In questa direzione l'uso del computer game come generatore di simulazioni diventa una scelta naturale, in particolare quando sia possibile, variando pochi parametri, ottenere un cambiamento nel mondo che si rifletta sulla necessità di riconsiderare il processo abducente fin lì compiuto.

### 3. Il progetto CRETA

L'obiettivo finale del sistema CRETA di produzione di percorsi di simulazione è quello di supportare direttamente la scrittura creativa della simulazione in linguaggio naturale strutturato, cioè tale da guidare l'autore a descrivere i diversi elementi che costituiranno la simulazione. La tabella sottostante illustra i diversi strati del sistema che si va costruendo. Lo strato di scrittura creativa si basa sulla istanziazione di *template* (schemi) predefiniti. Questi sono realizzati al livello sottostante mediante documenti XML con uno specifico linguaggio di marcatura (*tag*). I documenti XML sono infine compilati su regole di tipo condizione-azione, direttamente eseguibili da parte del motore di simulazione.

	<b>Sistema/linguaggio</b>	<b>Prodotto</b>
<b>1</b>	Scrittura creativa	Storyboard in linguaggio naturale
<b>2</b>	Linguaggio autore basato su template	Storyboard strutturato mediante template predefiniti
<b>3</b>	Linguaggio di annotazione basato su XML	Testo annotato con tag XML
<b>4</b>	Motore di simulazione basato su regole di produzione	Collezione di regole <b>If</b> condizione <b>then</b> azione

Tabella 1. Livelli del sistema CRETA

L'autore (passo 1) progetta la simulazione in riferimento alla metodologia descritta in questo documento, e quindi (passo 2) ne dà una descrizione di dettaglio completando i template relativi a tutti gli elementi previsti, utilizzando uno strumento di editing tradizionale (ad esempio lo stesso Word) sui template necessari. Il sistema (passo 3) traduce automaticamente l'output del passo 2 in documenti XML, che vengono successivamente (passo 4) compilati nella collezione di regole per il motore di

simulazione.

### 3.1. Architettura del motore di simulazione

Gli script di simulazione definiti dall'autore vengono trasformati, attraverso una serie di compilazioni successive descritte nella sezione 3.2, in un insieme di regole per il sistema ad agenti CLIPS [CLIPS, 1994]. Le regole CLIPS sono costituite da una condizione, detta *testa*, e da una serie di azioni, detta *corpo*; le condizioni predicano su una raccolta di *fatti* che codificano, fra l'altro, lo stato corrente del mondo simulato e, riflessivamente, quello della stessa simulazione. I fatti sono codificati tramite *tuple* arbitrarie, che costituiscono un meccanismo di codifica universale. Ogni volta che la condizione descritta nella testa di una regola diventa *vera*, la regola è attivata, e vengono eseguite le azioni definite nel corpo. L'esecuzione delle azioni può naturalmente influenzare lo stato corrente della simulazione, e causare in cascata l'attivazione di ulteriori regole. L'intera computazione è organizzata in sequenze di attivazioni determinate da un apposito algoritmo di ordinamento. Quando non ci sono più regole che possono essere attivate (ovvero, regole la cui testa è verificata), la simulazione viene sospesa, in attesa che a cambiare lo stato del mondo simulato intervenga un'azione del discente/giocatore (o lo scorrere di un tempo prefissato). Non appena ciò si verifica, il sistema di regole reagisce al cambiamento, realizzando così una co-evoluzione del mondo simulato in cui il discente e gli altri agenti descritti dall'autore interagiscono secondo schemi non prefissati.

Lo stato della simulazione è rappresentato come un albero, in cui ciascun nodo, a qualunque livello della gerarchia, ha un nome e può avere associato un valore. Come avviene per gli *stem* del linguaggio REXX [REXX, 1996], a cui la struttura dello stato è ispirata, l'albero può essere usato per rappresentare in maniera uniforme strutture dati diverse come i record, gli array, gli array associativi (e dunque, dizionari e *hash table*). Questa uniformità consente poi di esporre verso l'autore un meccanismo unico per il mantenimento di informazioni sullo stato "interno" della simulazione così come sullo stato del mondo simulato.

Un secondo elemento fondamentale della simulazione è anch'esso mantenuto, sebbene solo virtualmente, sotto forma di albero: si tratta dell'*albero dei comportamenti*, che specifica i comportamenti dei vari agenti che compaiono nella simulazione sotto forma di alberi in cui ciascun nodo intermedio è una condizione, mentre i nodi foglia sono delle azioni. Durante la compilazione della simulazione, i cammini dalla radice a ciascuna foglia vengono linearizzati, le condizioni incontrate lungo il cammino composte (tipicamente per congiunzione), e il risultato usato per generare una singola regola CLIPS. Sebbene al momento dell'esecuzione i comportamenti risultino così "appiattiti", la loro strutturazione ad albero semplifica il processo di scrittura dello script di simulazione, che può indicare facilmente comportamenti generici (corrispondenti a nodi alti nell'albero) che gli agenti devono tenere in ogni caso, e comportamenti più specialistici (corrispondenti a nodi con elevato livello di annidamento) che descrivono più dettagliatamente situazioni specifiche.

È importante rimarcare qui che l'effettivo comportamento della simulazione in fasi di esecuzione è dato dalla *libera combinazione* di tutti i comportamenti possibili per tutti gli agenti, in base allo stato del mondo simulato (incluso lo stato epistemico degli agenti stessi). Non è infrequente che la simulazione esibisca comportamenti inattesi dall'autore, il quale non è chiamato a specificare singolarmente ogni possibile accadimento, come avviene in altri sistemi, bensì a descrivere isolatamente i comportamenti possibili per i singoli agenti, lasciando poi alla natura combinatoria del motore di esecuzione il controllo sull'effettivo dipanarsi degli avvenimenti.

Le nostre esperienze iniziali indicano che il sistema CRETA costituisce un buon compromesso fra il controllo che viene dato all'autore della simulazione e la quantità di lavoro richiesta per la stesura di uno script di ragionevole complessità. In altre parole, riteniamo che il sistema CRETA consenta di realizzare con uno sforzo di authoring moderato proprio quell'ambiente simulato dal comportamento complesso e a tratti indefinito che si prospetta come il più proficuo ambiente di apprendimento per le abilità analogiche di cui abbiamo parlato.

### **3.2. Il linguaggio di annotazione**

Il sistema CRETA consente l'authoring della storia su più livelli, che possono anche essere mescolati in maniera arbitraria all'interno dello stesso script. Come descritto nella sezione precedente, l'esecuzione avviene tramite l'applicazione di regole condizione/azione (livello 4 in Tabella 1) ottenute dalla compilazione di annotazioni XML (livello 3 nella stessa tabella). Quest'ultimo è organizzato in più sotto-linguaggi: quello di livello più basso (che indicheremo come "3.0") consente di esprimere condizioni tramite l'applicazione di operatori relazionali sui valori contenuti in nodi dell'albero dello stato (tag <test>), e azioni costituenti calcoli e modifiche agli stessi valori (tag <set>). Il livello 3.0 offre inoltre costrutti di scoping (attraverso la definizione di un "namespace corrente") e alcune altre operazioni di servizio. Si tratta come si vede di un livello assai basilico: la specifica di una simulazione attraverso costrutti così elementari sarebbe improponibile. Il livello immediatamente successivo ("3.1") fornisce tag per la definizione di una mappa (insieme di locazioni, connessioni fra di esse e tempo di attraversamento, tag <Mappa>, <Locazione> e altri), per la definizione di un agente (tag <Agente>, che indica fra l'altro il tipo di agente fra animato e inanimato, la sua locazione iniziale, le sue proprietà e – di fondamentale importanza – i suoi comportamenti), per la definizione di dialoghi, spostamenti, e altre interazioni fra gli agenti (per esempio, tramite i tag <Dice>, <SeAscolta>, <SeEntra>, ecc.). Tutti questi tag vengono tradotti in tag del livello 3.0: per esempio, mappa e attributi degli agenti vengono memorizzati in ramo dell'albero dello stato identificati da nomi predefiniti in cima al sottoalbero. L'authoring di una simulazione a questo livello è realistico, benché ancora molto dettagliato. Costrutti di livello ancora più alto sono definiti al livello 3.2, detto anche "livello delle macro". In questo livello sono definiti tag che riassumono schemi di interazione paradigmatici, quali la conversazione casuale (interazione senza scambio di informazione, data dal tag <Chiacchera>) e il tentativo di convincimento, ovvero il tentativo da parte di un agente di influenzare lo stato epistemico di un altro agente (tag <Convinci>).

### **4. Sviluppo di un ambientazione e di un frammento di storia**

L'ambiente Creta, per la sua generalità, si presta allo sviluppo di storie in qualunque ambito. In questo caso si è scelto di lavorare allo sviluppo di una simulazione nell'ambito di un ente della Pubblica Amministrazione.

La scelta dell'ambientazione permette di definire alcuni aspetti concreti relativi alla descrizione dei personaggi e alle loro caratteristiche che sono utili in questa specifica simulazione. Una successiva scelta relativa all'analisi di un caso ha portato allo studio di una situazione complessa in cui sperimentare e validare l'uso di alcune *soft skills*.

Si è scelto di limitare l'indagine ad una scena denominata "Scelta e negoziazione di un progetto complesso" in cui ogni personaggio è chiamato a svolgere un incarico del tipo: «Valutare il migliore tra cinque progetti e proporre una decisione alla riunione del dipartimento che si terrà tra X giorni».

La complessità della situazione sociale scelta ci permette di indagare le seguenti *soft skills*: la capacità di negoziazione; la capacità di bilanciare tra cooperazione e competizione; l'intuizione di soluzioni sub-ottimali, ovvero il saper riconoscere punti di ottimo locale senza necessariamente puntare all'ottimo globale, spesso irraggiungibile o inesistente.

#### **4.1. Ambiente, personaggi e caratteristiche**

È definita una mappa del mondo in cui si svolge la simulazione, il dipartimento Sviluppo E Imprese, in termini di luoghi diversi interconnessi. In ogni momento ciascun personaggio ed oggetto risiede in un luogo. I personaggi hanno luoghi preferenziali dove lavorano da soli o interagiscono con gli altri. I personaggi, in generale, possono muoversi. Il movimento del protagonista è pilotato dal discente, il movimento degli altri personaggi dal motore di simulazione, in modo random o in base a comportamenti tipici descritti da regole. Il movimento è funzionale ad incontri, che possono essere fortuiti o forzati al verificarsi di determinate circostanze.

Attraverso un'approfondita indagine degli ambienti della Pubblica Amministrazione si sono dettagliate le caratteristiche dei personaggi suddividendole in *Stili* e *Abilità*.

Lo *Stile* è il modo di affrontare gli eventi e di relazionarsi con le altre persone. Dà vita a degli archetipi di carattere e viene usato come un peso che agisce sui meccanismi di variazione delle *Abilità* e sulle regole di interazione dei personaggi.

Le *Abilità* vengono utilizzate per simulare i comportamenti della vita reale. Sono state scelte quelle più significative ai fini del contesto di riferimento: Fiducia verso..., Paziente, Formale/Informale, Ambizioso, Conoscitore di ambienti esterni, Conoscitore di ambienti interni, Disciplinato, *Leadership*, Disponibile- *friendly*, Puntuale, Lento/veloce.

Alcune abilità sono fisse, cioè vengono individuate in fase di costruzione del personaggio e non variano nel corso della storia, mentre il valore delle altre può sia crescere che decrescere. Le abilità fisse possono essere considerate come tratti caratteriali consolidati e sono: paziente, formale, ambizioso, disciplinato, leadership.

Per acquisire un incremento di un punto relativamente ad un'abilità è necessario compiere un certo numero di azioni 'positive' con un meccanismo per cui passare da un livello basso ad uno medio è relativamente facile mentre arrivare ad eccellere in un campo richiede notevoli sforzi.

Alcune abilità possono decrementare in seguito ad azioni "sbagliate" (es. fiducia) oppure solo in casi di eventi straordinari (es. conoscitore di ambienti può scendere solo se avviene un cambiamento negli uffici o nella giunta, ...).

#### **4.2. Frammento di storia denominata "Scelta e negoziazione di un progetto complesso" e relative soft skills**

Si tratta di assolvere ad un incarico del tipo: «Valutare il migliore tra cinque progetti e proporre una decisione alla riunione del dipartimento che si terrà tra X giorni».

Le *variabili* che concorrono nella simulazione della complessità del mondo sono le seguenti. I *progetti* che devono essere valutati rispetto ad una griglia di 5 possibili caratteristiche (numero-soggetti-beneficiari, costi/benefici-attesi, sostenibilità, visibilità, facilità-di-realizzazione) rispetto alle quali ogni progetto eccelle al massimo in una o due: non vi è quindi un progetto palesemente migliore e la scelta dovrà tenere in considerazione vari fattori ambientali; inserimento di *imprevisti*: il peso delle singole

caratteristiche può variare a causa di aumento/diminuzione di fondi, coinvolgimento/allontanamento di un esperto in un settore, cambiamento del capo in corso d'opera, ecc.; lo *stile* del capo (manager/proceduralista) che può apprezzare maggiormente lo spirito collaborativo/individuale; i punteggi nelle *abilità* individuali degli altri agenti/personaggi che possono differenziare i risultati delle singole negoziazioni; l'inserimento di regole per la *sub-ottimalità* (come nel caso del dilemma del prigioniero).

Il protagonista/giocatore non è in grado da solo di ottenere tutte le informazioni necessarie sui singoli progetti e si trova a dover fare una scelta nello spazio di possibilità: alleanza = collaborare con altri per lo scambio di informazioni; negoziare informazioni = proporre una scelta comune vs. non condividere le informazioni e proporre la propria posizione individuale; tempo = scelta del tempo in cui annunciare la propria decisione.

### 4.3. Regole di interazione dei personaggi

Ai fini della nostra simulazione ci interessa indagare più un 'saper essere' che un 'saper fare' e in questa direzione una particolare attenzione si deve porre sui meccanismi di interazione 'sociale': per questo sono state definite una serie di regole per l'interazione dei personaggi. Ne descriviamo una a titolo esemplificativo.

Incontro a due: il protagonista PP incontra faccia a faccia un altro personaggio A. Lo scopo dell'incontro è di influenzare l'atteggiamento dell'altro su uno specifico argomento X. Entrano in gioco: l'abilità di leadership di PP contrapposta con quella di A, la relazione interpersonale tra A e PP (abilità: Fiducia verso).

Si definisce un confronto tra la *leadership* di PP da una parte e la *leadership* di A meno il valore di Fiducia verso PP dall'altra (in questo modo se c'è molta fiducia sarà più facile che PP abbia credito presso A). Viene inoltre inserito un numero a generazione casuale (lancio di un dado) tale che anche personaggi con *leadership* bassa possano avere successo e quelli con *leadership* alta non convincano sempre gli altri. I due valori vengono confrontati e se quello del personaggio che cerca di convincere (nel nostro caso PP) è più alto il tentativo ha avuto successo e lo stato interno di A relativo all'argomento X cambia.

Esempio: il punteggio di *Leadership* di PP è 6, il punteggio di *Leadership* di A è 5, la Fiducia di A verso PP è 4, il lancio del dado è 3. Il confronto avviene tra 6 e  $5 - 4 + 3 = 4$ ; poiché 6 è maggiore di 4 PP è riuscito a convincere A.

Con meccanismi simili vengono definite tutte le interazioni di interesse per il frammento di storia considerata.

## 5. Stato dell'arte e sviluppi futuri

Allo stato attuale dei lavori del gruppo di sviluppo (Antonio Cisternino, Vincenzo Gervasi e Salvatore Ruggieri) il motore di simulazione è virtualmente completo, ed esiste una versione preliminare del linguaggio basato su XML, da estendere via via che aumenta la nostra esperienza collettiva nella stesura delle simulazioni.

E' in corso di completamento una demo del sistema (essenziale nell'interfaccia) descritta nella sezione 4.2 realizzata in XML.

Dallo sviluppo di questo prototipo si procederà con la prima definizione delle regole per il linguaggio autore tali da permettere l'implementazione di uno storyboard strutturato mediante template predefiniti.

In parallelo si lavorerà al test del prototipo sia in maniera manuale che in manie-

ra automatica, cioè analizzando e valutando i diversi alberi dello stato della simulazione che si generano variando i parametri. Si intende inoltre validare il frammento di storia attraverso dei test realizzati da esperti del settore.

## 6. Conclusioni

La complessità pone al centro dell'attenzione, come capacità chiave da alimentare attraverso la formazione, l'attitudine a leggere conoscenza latente, non formalizzata che si traduce nella necessità di un 'allenamento' alla formulazione di inferenze di tipo abduttivo: non conosco il mondo che sto esplorando, ma i pochi elementi di cui dispongo mi permettono di ipotizzare che... ecc. ecc.

Non si tratta solamente di apprendere 'contenuti', ciò che conta è la capacità di attivare nuove connessioni all'interno di una rete del sapere sempre mobile e instabile, ma già data ed accessibile: immaginare un mondo possibile. In questo scenario il *Computer Game* diventa un esempio estremo di macchina per la costruzione e l'immersione in un'esperienza che serva come generatore di analogie.

In tal senso, CRETA, come sistema di produzione di percorsi di simulazione, si propone come strumento flessibile per la creazione di storie per l'apprendimento realizzate in linguaggio 'naturale' attraverso *storyboards* e per la fruizione delle stesse storie come percorsi di *e-learning*.

## Riferimenti bibliografici

Bocchi, Gianluca, *La formazione come costruzione di nuovi mondi*, Roma, Formez Censis, 1993.

Borges, Jorge Luis, *Obras Completas*, Buenos Aires, Emecé, 1983.

CLIPS, <http://www.ghg.net/clips/download/documentation/3CCP.pdf>.

CRETA, [www.cretaonline.org](http://www.cretaonline.org)

Gagliardi, Pasquale, "The Role of Humanities in the Formation of New European Elites", Keynote Speech for the Annual Meeting of the European Academy of Management, Stockholm, May 2002, <http://www.sses.com/public/events/euram>.

REXX, "Programming Language - REXX, ANSI Standard X3.274-1996, 1996." <http://www.rexxla.org/Standards/standards.html>

Varanini, Francesco e Bocchi, Gianluca. *La macchina analogica*. In Raoul C. D. Nacamulli e Daniele Boldizzoni (a cura di), *Oltre l'aula*, Apogeo, 2004.

Voltaire, *Candide, ou L'Optimisme*, 1759; trad. it. *Candido, ovvero l'Ottimismo*, Milano, Mondadori, 1993.