

UNIVERSITA' DEGLI STUDI DI SALERNO
FACOLTA' DI SCIENZE MATEMATICHE FISICHE E NATURALI

CORSO DI LAUREA IN SCIENZE DELL'INFORMAZIONE

TESI DI LAUREA

**ANALISI DEI SISTEMI DI TUTORING INTELLIGENTI CHE
UTILIZZANO MODELLI FUZZY**

Relatore:
Chiar.mo Prof.
Luigi Di Lascio

Candidato:
Salvatore Lega
Matr. 53/5143

Anno Accademico 2003/2004

A Ramona

INDICE

Elenco delle figure	IV
Introduzione	1
Riferimenti	4
Capitolo I	
Architettura di base di un ITS e logica Fuzzy	5
1.0 Introduzione	5
2.0 La struttura modulare di un ITS	6
2.1 Il modello esperto	6
2.2 Il modello studente	7
2.3 Il modello pedagogico	7
2.4 L'interfaccia utente	8
3.0 Il modello studente, aspetti teorici e pratici	8
3.1 Classificazione dei modelli studente	8
3.2 La rappresentazione del modello studente	10
4.0 La logica Fuzzy costruttrice del modello	12
4.1 Concetti di base della logica Fuzzy	13
4.2 Logica Fuzzy e linguaggio parlato	15
Riferimenti	17

Capitolo II	
Il Sistema di tutoring I.T.W.	18
1.0 Introduzione	18
2.0 IWT: Intelligent Web Teacher	19
3.0 Il modello della conoscenza	21
4.0 Il modello studente	23
5.0 Il modello didattico	25
6.0 Conclusioni e sviluppi futuri	26
Riferimenti	28
Capitolo III	
Un ITS fondato sul binomio Corba/WWW	29
1.0 Introduzione	29
2.0 Cosa è C.O.R.B.A.?	30
2.1 Quali sono i componenti di C.O.R.B.A.?	31
3.0 L'architettura modulare del sistema	33
4.0 Il modulo esperto	34
5.0 Il modulo studente	35
6.0 Conclusioni	37
Riferimenti	39
Capitolo IV	
Il sistema adattivo ipermediale Alice	40
1.0 Introduzione	40
2.0 Descrizione del sistema Alice	41
2.1 Il modello utente	42
3.0 Le tecnologie adattive	44
Riferimenti	45
Capitolo V	
La Piattaforma InterMediActor	46
1.0 Introduzione	46
2.0 La piattaforma InterMediActor	48
3.0 Il modello studente	50
3.1 Il livello d'importanza	51
3.2 Il livello di difficoltà	51
3.3 I voti	53

3.4 I requisiti iniziali (pre-requisiti) della conoscenza	54
3.5 La defuzzificazione	54
4.0 Conclusioni	54
Riferimenti	55
Capitolo VI	
Il Sistema INSPIRE	56
1.0 Introduzione	56
2.0 L'architettura di INSPIRE	57
3.0 Il problema del processo di diagnosi dello studente	59
4.0 Il processo di diagnosi dello studente in INSPIRE	60
4.1 La creazione del modello della conoscenza esperta	62
4.2 Il processo di diagnosi	65
5.0 Conclusioni	67
Riferimenti	68
Capitolo VII	
I Sistemi Witness e Sherlock II	69
1.0 Introduzione	69
2.0 Il sistema Witness	70
2.1 Le componenti del sistema	70
3.0 Le relazioni tra le diverse componenti del sistema	71
4.0 Il sistema Sherlock II	74
Riferimenti	76
Bibliografia Generale	77
Bibliografia divisa per argomenti	
Parte A: Riferimenti ad alcuni ITS fuzzy	85
Parte B: Riferimenti ad alcuni ITS non fuzzy	87

Elenco delle figure

Fig.1 Architettura di un I.T.S. : Interazione tra i modelli.....	pag.06
Fig.2 Il modello studente Overlay ed Overlay-Buggy.....	pag.11
Fig.3 Funzione d'appartenenza di un generico insieme fuzzy	pag.12
Fig.4 I tre livelli della conoscenza I.W.T.....	pag.22
Fig.5 La configurazione del sistema Corba-based.....	pag.33
Fig.6 La rappresentazione delle reti IUN e CN.....	pag.35
Fig.7 Il grafo della funzione D.V.....	pag.37
Fig.8 La struttura di un sistema adattivo ipermediale.....	pag.41
Fig.9 Le funzioni del server e del client in Alice.....	pag.42
Fig.10 Il dominio della conoscenza esperta e il dominio utente.....	pag.43
Fig.11 Analisi top-down degli obiettivi d'apprendimento.....	pag.48
Fig.12 La trasformazione in competenza atomica.....	pag.49
Fig.13 Le funzioni μ per la var. ling. "livello di difficoltà.....	pag.52
Fig.14 La struttura di base del sistema Inspire.....	pag.58
Fig.15 Le componenti funzionali di WITnESS.....	pag.71
Fig.17 Le relazioni tra le componenti di WITnESS.....	pag.72

Introduzione

Lo sviluppo delle nuove tecnologie e la crescita inarrestabile di modelli di insegnamento–apprendimento a distanza, sempre più flessibili ed efficaci, basati sull'utilizzo delle tecnologie stesse, stanno determinando la nascita di una nuova società che possiamo definire “società cognitiva”.

E' una società che richiede un profondo rinnovamento delle istituzioni formative e dei modi di trasmettere ed acquisire il sapere; le scuole, ed in particolare le Università, devono essere in grado di formare i professionisti del futuro, capaci di acquisire sempre nuove conoscenze e competenze e di far fronte ad un mercato del lavoro sempre più aperto e flessibile.

Il ruolo dei docenti e dei discenti, all'interno del processo formativo, muta radicalmente poiché, i primi, si trovano ad acquisire nuove e complesse competenze relative all'insegnamento a distanza, oltre che il ruolo di guida del processo di apprendimento degli studenti; i secondi, invece, si rivestono di un ruolo attivo che permette loro di diventare veri e propri protagonisti nella creazione di un nuovo sapere e di una nuova conoscenza.

In questo contesto, assume un'importanza rilevante la figura del sistema di tutoring, il cui compito è quello di sostenere, stimolare, accompagnare i discenti nel loro percorso di formazione e aiutarli a sviluppare capacità cognitive superiori, grazie anche all'utilizzo di quegli strumenti che Donald A. Norman definisce “*artefatti cognitivi*”: ovvero le nuove tecnologie.

I sistemi tutoriali intelligenti (ITS: Intelligent Tutoring System) sono sistemi software progettati, principalmente, per supportare l'attività d'apprendimento di tipo specializzato e individualizzato.

Questi sistemi possono essere utilizzati nei normali processi educativi, nei corsi d'apprendimento a distanza, sotto forma di cd-rom oppure come applicazioni per la divulgazione delle conoscenze via internet.

I primi sviluppi dell'introduzione di programmi software, come mezzi ausiliari nell'apprendimento educativo, furono formalizzati nell'ambito del progetto CAI (Computer Aided Instruction). Questi programmi utilizzavano un albero di decisione per guidare lo studente da una sessione all'altra a seconda delle risposte formulate [Urban Lurain 1996]. Tuttavia, non prendevano in considerazione la diversità degli studenti e le loro specifiche necessità, background o storia.

I programmi CAI, quindi, non erano in grado di adattarsi al modo specifico di acquisire le conoscenze che lo studente aveva e non erano in grado di fornire un'attenzione individualizzata che invece il tutore umano assicurava [Bennett 1997].

I recenti sviluppi della ricerca nel campo dell'intelligenza artificiale hanno portato ad un nuovo campo d'applicazione, che prende il nome di Intelligent Tutoring System [Burns & Capps 1998]. La caratteristica di base di questi sistemi, è di considerare ogni studente come unico, creando un modello studente capace di registrarne le preferenze e i progressi nel corso del processo cognitivo [VanLehn 1998]. In più ciò che ne incrementa l'efficacia nell'insegnamento, è la propria capacità di adattarsi alle caratteristiche dello studente. Per far questo, il sistema deve cercare di avvicinarsi il più possibile al modo di ragionare dello studente. Deve, in altri termini, gestire i differenti aspetti della vaghezza, presenti nei dati espressi con parole reali.

L'intelligenza artificiale ha risolto questo problema utilizzando, tra l'altro, la logica fuzzy, nel tentativo di fornire una base robusta e sistematica tesa a gestire le imprecisioni linguistiche, che sono dovute all'assenza di confini ben definiti, entro i quali i costrutti linguistici possono trovarsi.

Inizieremo con il dare uno sguardo all'architettura di base di un sistema tutoriale intelligente e alla relativa logica fuzzy, per poi descrivere in dettaglio gli aspetti di alcuni authoring ITS che utilizzano modelli fuzzy. In particolare si inizierà con una trattazione dei Sistemi di Tutoring Intelligente su base web.

Alla fine, presenteremo nella parte B della bibliografia un elenco di authoring ITS, incontrati durante la stesura della tesi, che non usano modelli di tipo fuzzy.

Riferimenti

Bennett, F. 1997. *'Computers as Tutors: Solving the Crisis in Education'*. First Monday. <http://www.firstmonday.dk/issues /issue6/intro/index.html>.

Burns, H. L., and Capps, C. G. (1988). *'Foundations of Intelligent Tutoring Systems'*. In Polson, M. C., and Richardson, J. J., eds., *'Foundations of Intelligent Tutoring Systems'*. Lawrence Erlbaum Associates. capitoli 3, 55–78.

Urban-Lurain, M. (1996). *'Intelligent tutoring systems, an historic review in the context of the development of artificial intelligence and educational psychology. Technical report'*. Department of Computer Science and Engineering, Michigan State University
<http://aral.cse.msu.edu/ Publications/ITS/its.htm>.

VanLehn, K. (1988). *'Student Modeling'*. In Polson, M. C., and Richardson, J. J., eds., *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates. capitoli 3, 55–78.

CAPITOLO I

Architettura di base di un ITS e logica Fuzzy

1.0 Introduzione

La personalizzazione dell'insegnamento attraverso l'uso di sistemi educativi, può in modo significativo, migliorare il processo d'apprendimento dello studente. Per cui un ITS svolge funzioni analoghe a quelle di un tutore umano, emulandone il processo decisionale. Esso è in grado di gestire una grande conoscenza su di una specifica area, inoltre instaura, tramite un'interfaccia in linguaggio naturale (o molto vicino ad esso) un colloquio con l'utente: esegue deduzioni relative al caso di studio, secondo le conoscenze del dominio precedentemente memorizzate.

Per creare un sistema che si adatti ad ogni singolo studente, dovremo essere in grado di memorizzare ogni azione e da questa dedurre l'evoluzione del suo "stato mentale". La rappresentazione dello stato mentale corrente dello studente è chiamata "modello studente".

Per poter soddisfare i suoi scopi un ITS dovrà, quindi, avere una struttura di tipo modulare.

La struttura di base di un Intelligent Tutoring System sarà trattata all'interno del paragrafo 2. Mentre una classificazione dei modelli studenti e loro rappresentazione sarà data nel paragrafo 3, infine all'interno del quarto si tratterà brevemente della logica fuzzy quale costruttrice dei modelli studenti.

2.0 La struttura modulare di un ITS

Gli ITS possono apparire, a prima vista, dei sistemi monolitici, tuttavia per gli obiettivi che si prefiggono e per una più facile implementazione, consistono di diverse componenti. Secondo McTaggart [2001] gli ITS sono composti da quattro componenti che interagiscono tra di loro in modo tale da mettere lo studente in relazione con la conoscenza di un soggetto predominante. Tale interazione fa sì che lo studente assimili nuove conoscenze e le inserisca nel suo schema mentale. Le componenti in questione sono:

- Il modello della conoscenza esperta.
- Il modello studente.
- Il modello pedagogico.
- L'interfaccia utente.

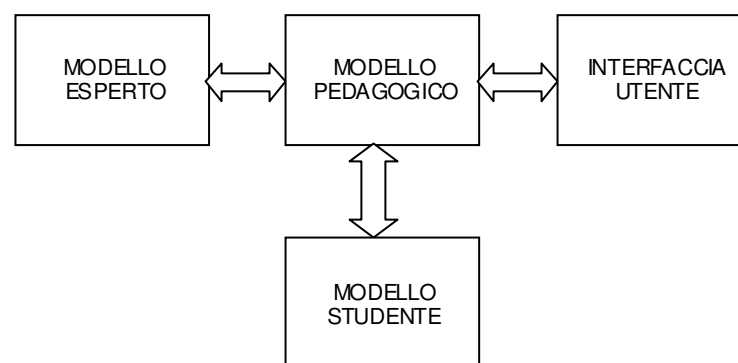


Figura 1. Architettura di un ITS: interazione tra i modelli.

2.1 Il modello esperto

Questo modulo contiene la rappresentazione esplicita della conoscenza da fornire allo studente espressa sotto forma di un modello (di

conoscenza) in grado sia di comunicare all'esterno i concetti e le proprietà del dominio di applicazione, sia di dotare il sistema di capacità dinamiche per l'elaborazione della conoscenza esperta.

Il modello consente perciò all'esperto di dominio di generare automaticamente le soluzioni ai problemi formulati nel corso del processo di insegnamento, di spiegare i passi del ragionamento seguito e di confrontare i processi di inferenza dello studente stesso con quelli corretti del sistema. Ne segue che è potenzialmente possibile valutare sia il livello di apprendimento sia i progressi nello studio del dominio in esame da parte dello studente.

2.2 Il modello studente

Interpreta le risposte fornite dallo studente formulando ipotesi sulle conoscenze che hanno permesso di produrre tali risposte. Tale modulo deve perciò mantenere un modello individuale dello specifico studente in esame. In alcuni sistemi l'uso di conoscenze diagnostiche esplicitamente rappresentate nel sistema permette di spiegare la genesi degli errori e delle distorsioni relative al dominio di applicazione.

Si parlerà in maniera più dettagliata del modello studente nel prossimo capitolo.

2.3 Il modello pedagogico

Il modello contiene le conoscenze specialistiche sulle strategie didattiche e le tecniche di insegnamento. E' il modulo deputato alla pianificazione della presentazione degli argomenti, dei problemi da risolvere sulla base della strategia didattica scelta e delle informazioni contenute nel modello studente e nel modello esperto.

Le strategie didattiche permettono al modulo di decidere, ad esempio, quando e come fornire spiegazioni, riassunti, esempi, ed analogie; quale tipo di aiuto, suggerimento o feedback fornire allo studente ecc. Nei sistemi più avanzati il modulo contiene anche un repertorio di "meta

strategie” didattiche, che vengono usate nelle varie situazioni di apprendimento che si presentano. Il modulo pedagogico stabilisce anche la modalità di interazione e di presentazione dei materiali allo studente.

2.4 L’interfaccia utente

Infine questa componente controlla il flusso delle comunicazioni da e verso l’utente traducendo le informazioni dal linguaggio di rappresentazione interno al sistema ad un linguaggio più facilmente comprensibile allo studente.

3.0 Il modello studente, aspetti teorici e pratici

Il modello studente, come si è accennato in precedenza, è la componente di un sistema tutoriale intelligente, che memorizza lo stato corrente della conoscenza acquisita.

Uno dei primi tentativi di costruzione del modello è stato quello sperimentato da Carbonell [1970], che utilizzò una rete semantica per rappresentare il dominio della conoscenza.

Tuttavia il termine “modeling student” scaturisce dalla ricerca CAI. I primi tentativi, da parte del CAI, furono fatti per incoraggiare l’insegnamento individualizzato attraverso l’uso di modelli studente.

I primi sistemi CAI, descritti dal Self [1974], erano caratterizzati dal fatto che commenti e problemi non erano pre-memorizzati, ma erano generati dinamicamente. Egli suddivise i sistemi CAI in due categorie a seconda che la conoscenza fosse acquisita da un insegnamento implicito o esplicito.

3.1 Classificazione dei modelli studenti

I modelli degli attuali ITS possono essere classificati a seconda delle funzioni che essi eseguono oppure, in modo più significativo per la

letteratura corrente, per mezzo dei rispettivi modi di interpretazione dell'informazione: modelli di elaborazione o statici [Cancey,1986].

Self [1988] descrive sei funzioni principali cui devono soddisfare i modelli studente:

- Funzione correttiva
- Funzione elaborativa
- Funzione strategica
- Funzione diagnostica
- Funzione predittiva
- Funzione valutativa.

La funzione correttiva

In questo caso il modello deve essere in grado di identificare la differenza tra ciò che ha compreso lo studente e il concetto reale, per poi essere in grado di porvi rimedio.

La funzione elaborativa

Qui il modello deve allargare la conoscenza dello studente, cioè deve identificare le nuove aree nelle quali il discente può essere introdotto, oppure affinare la sua conoscenza corrente.

La funzione strategica

Questa funzione cambia l'approccio d'insegnamento, modificando con un più alto livello di docenza l'attuale insegnamento.

La funzione diagnostica

Tale funzione descrive l'analisi dello stato cognitivo dello studente. Se, ad esempio, il tutore desidera introdurre un nuovo argomento e se il modello studente non è in grado di stabilire l'adeguatezza del livello di conoscenza raggiunto, allora dovrà essere in grado di generare esempi di tipo diagnostico da far svolgere allo studente stesso.

La funzione predittiva

Qui, il modello usa questa funzione per anticipare l'effetto di una azione sullo studente. In altri termini il modello simula il comportamento dello studente.

La funzione valutativa

Questa funzione fornisce una valutazione del livello d'apprendimento del discente. Ciò richiede al sistema una sorte di aggregazione delle informazioni possedute.

3.2 La rappresentazione del modello studente

I primi sistemi CAI supportavano le funzioni correttive ed elaborative ma erano carenti nelle procedure di diagnostica, strategia e di predizione. I modelli di elaborazione, chiamati di esecuzione, sono in grado di simulare il processo per mezzo del quale il discente risolve un problema e possono, pertanto, eseguire la funzione predittiva.

I modelli eseguibili sono anche chiamati procedurali [Brusilovsky 1994], in quanto gli elementi della conoscenza di quest'ultimo sono, di solito, rappresentati per mezzo di procedure.

Ci sono diversi modi per rappresentare l'informazione relativa allo studente.

Le tecniche comunemente usate sono:

- Modelli Overlay
- Modelli Overlay con estensione buggy
- Reti Bayesiane
- Logica Fuzzy

Il modello Overlay e la conoscenza buggy

I modelli Overlay sono caratterizzati dal fatto che considerano la conoscenza dello studente come un sottoinsieme della conoscenza del modello esperto. Con questa rappresentazione, un ITS porgerà allo studente sempre più materiale, fin quando il suo sapere non corrisponderà a quello del modello esperto.

Lo svantaggio di questo tipo d'approccio, sta nel fatto di non ammettere che gli studenti possano essere portatori di concetti non appartenenti alla base esperta della conoscenza.

Per cui un'estensione del modello overlay rappresenta esplicitamente la conoscenza buggy (errata) che lo studente potrebbe avere.

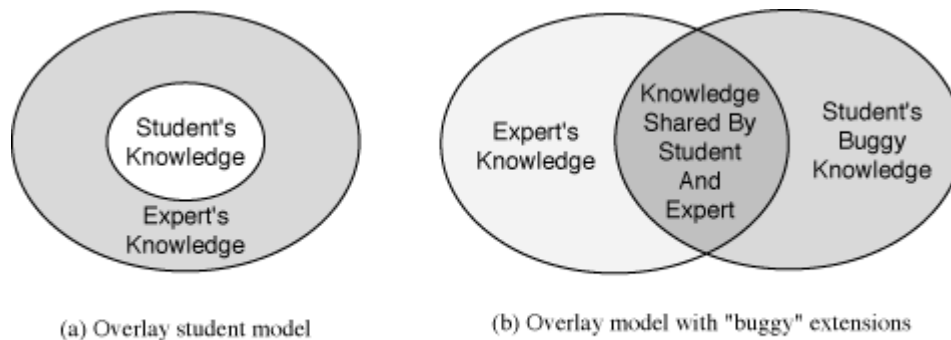


Figura 2 (a) Il modello studente Overlay; (b) Il modello studente "buggy"

Il modello esteso permette una migliore gestione degli errori compiuti dallo studente, tenendo presente che il fatto che egli possa avere un concetto errato, è pedagogicamente errato.

Le reti Bayesiane

Un altro meccanismo teso a memorizzare la conoscenza dello studente è rappresentato dalle reti Bayesiane. Queste ragionano in termini di probabilità sullo stato della conoscenza dello studente fornita dalla sua interazione con il tutore. Ciascun nodo della rete contiene una probabilità, che indica il grado di verosimiglianza tra ciò che lo studente sta conoscendo e la conoscenza effettiva.

4.0 La logica_Fuzzy costruttrice del modello

La caratteristica fondamentale della logica Fuzzy è la ridefinizione del concetto di appartenenza ad un insieme, o meglio la generalizzazione del criterio di definizione di un insieme.

Il padre della logica, colui che per primo ha messo le basi teoriche definendo in modo formale questi concetti è stato il prof. Lofti Zadeh. Egli fece notare come spesso in natura non ci si trovi di fronte a insiemi nettamente separati, cui si possono applicare i principi dell'insiemistica classica, come quello della non contraddizione o quello del terzo escluso. Ci sono moltissimi casi in cui esistono intrinsecamente le ambiguità e gli insiemi non sono, quindi, definibile in modo strettamente matematico.

La generalizzazione del concetto di insieme viene effettuata ridefinendo in maniera quantitativa il concetto di appartenenza, associando ad ogni elemento una coppia: la classe ed il grado di appartenenza a quella classe.

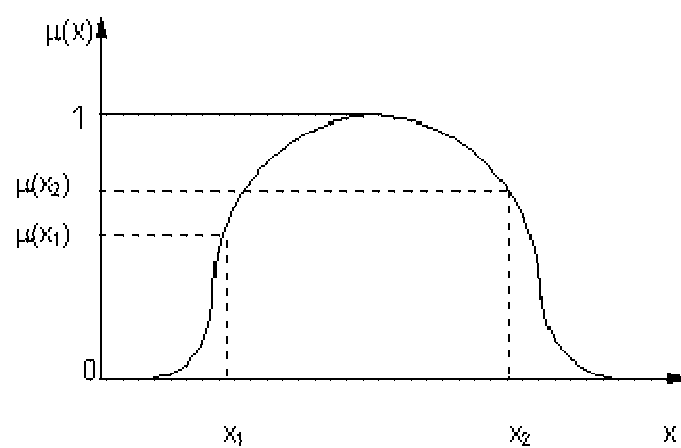


Figura 3.0 Funzione di appartenenza di un generico insieme Fuzzy

Il passo successivo, dopo la definizione degli insiemi Fuzzy, è l'approdo ad una teoria applicativa rispetto al concetto innovativo

dell'appartenenza graduata. Si arriva, quindi, alla logica Fuzzy, formalizzazione matematica di metodi di ragionamento basati su di una graduale separazione tra vero e falso e su informazioni consistenti, ma non definite con precisione: procedimenti molto simili ai processi decisionali umani

La logica Fuzzy può essere vista come un'estensione della logica multivalori, ma i suoi campi applicativi sono differenti. Perciò, il fatto che questo tipo di logica abbia a che fare con metodi di ragionamento "approssimato", più che con metodi precisi, implica che in generale il rigore non gioca, in questo ambito, un ruolo così importante come nella logica classica.

Si può affermare, quindi, che nella logica Fuzzy, tutto, incluso la verità, è qualcosa di graduato. Il grande potere espressivo della logica Fuzzy deriva dal fatto che essa contiene, come casi particolari, non solo le logiche classiche a due valori e multivalori, ma anche la teoria della probabilità. Un insieme Fuzzy sarà quindi un insieme il cui confine non è "crisp", cioè non è netto. In altre parole, quando si parla di appartenenza di un insieme fuzzy, si può parlare al tempo stesso di grado di appartenenza.

4.1 Concetti di base della logica Fuzzy

Nella logica a due valori, una proposizione p è vera o falsa. Nei sistemi multivalori una proposizione p , può essere vera, falsa o assumere un valore intermedio, che può essere un elemento di un insieme T finito o infinito. Nella logica Fuzzy, invece, i valori veri sono tutti quelli permessi in un intervallo determinato dai sottoinsiemi fuzzy di T . In altre parole, se T è un intervallo $[0;1]$, allora un valore vero in logica fuzzy, per esempio molto vero, può essere interpretato come un sottoinsieme fuzzy dell'intervallo T . Allo stesso modo, poco vero o altri. Si pensi, ad esempio, di dover dare un giudizio sull'altezza di una persona. Si potrà dire, che questa persona è molto alta, poco alta, abbastanza alta, bassa. Tutti questi sono sottoinsiemi

fuzzy del più grande insieme “altezza”. In questo senso, un valore fuzzy, può essere visto come una caratterizzazione imprecisa di un valore vero numerico.

I predicati in una logica a due valori devono essere precisi, nel senso che gli aggettivi utilizzati, possono essere del tipo “vero o falso”. Nella logica fuzzy, invece, si possono utilizzare attributi come “mortale”, “sempre”, ma più in generale aggettivi come “malato”, “alto”, “largo”, “stanco”.

La logica a due valori e multivalori permette l’uso di due soli quantificatori: “tutto” e “qualcuno”. In contrapposizione, la logica fuzzy permette l’uso di quantificatori tipo “parecchi”, “molti”, “frequentemente”.

Questi quantificatori, possono essere interpretati come numeri fuzzy che forniscono una caratterizzazione imprecisa della cardinalità di uno o più insiemi fuzzy o non fuzzy.

Nella logica a due valori una proposizione “P” può essere qualificata con un attributo: vero o falso. Quella fuzzy, invece, permette l’uso di tre principi di qualificatori:

Qualificatori di verità come “molto vero”, “vero”, “poco vero”, oltre che “falso”.

Qualificatori di probabilità come “probabilmente” o “improbabile”.

Qualificatori di possibilità come “possibilmente”.

Uno degli elementi fondamentali della logica fuzzy è il concetto di variabile linguistica, ossia di una variabile i cui valori siano parole più che numeri. E’ per questo, che tale logica ha molto in comune con l’intuizione e il pensiero umano. E’ chiaro che, se nella risoluzione di un problema posso tollerare maggiormente le imprecisioni, riuscirò ad abbassare la complessità dei metodi utilizzati per raggiungere la soluzione.

Un altro elemento fondamentale, è il concetto di regola fuzzy. In una regola del tipo:

*if (X_1 is A_1) and (X_2 is A_2) and ... (X_i is A_i) and ... (X_n is A_n)
then*

$Y_1 = B_1, Y_2 = B_2, Y_i = B_i, Y_m = B_m$

si chiamano antecedenti della regola ognuno dei termini $(X_i \text{ is } A_i)$ e si chiamano conseguenti quelli del tipo $(Y_i = B_i)$.

Sebbene esistano altri sistemi basati su regole, appartenenti ad un più vasto insieme denominato intelligenze artificiali, come la fuzzy, nessuno di questi possiede il motore che lavori con gli antecedenti ed i conseguenti della regola. Nella logica fuzzy, questa macchina è fornita dal calcolo delle regole fuzzy.

Vedremo nel prossimo paragrafo alcuni riscontri che questo tipo di logica ha nel linguaggio parlato.

4.2 Logica fuzzy e il linguaggio parlato

La logica è la scienza che sancisce i principi formali della ragione. In questo senso, la logica fuzzy, ha a che fare con i principi formali del pensiero “approssimato” e considera il ragionamento preciso come caso limite. In altre parole, il fulcro della logica fuzzy è che, diversamente dai sistemi logici classici, essa mira a modellare un modo impreciso di ragionare. Quest’ultimo gioca invece un ruolo essenziale nella abilità umana di prendere decisioni razionali in un ambiente di incertezze e imprecisioni. Questa abilità dipende dalla nostra capacità di “inferire” una risposta appropriata da una domanda che è inesatta, incompleta o totalmente inaffidabile.

L’inferenza può essere pensata come una deduzione intesa a provare o sottolineare una conseguenza logica. Altresì per inferenza si può intendere un procedimento di generalizzazione dei risultati ottenuti attraverso una rilevazione parziale dei campioni.

Di recente, Zadeh ha dimostrato che il principale contributo della logica fuzzy consiste nella metodologia dell'eseguire calcoli con le parole, che non può essere fatto ugualmente bene con altri metodi [Zadeh 1996].

Pertanto, le tecnologie che si fondano sulla logica fuzzy, con la loro capacità di rappresentare naturalmente l'esplicitazione dei concetti umani, possono realizzare molte applicazioni utili a modellare il sistema intelligente simil-umano.

Ad esempio, Anderson ha presentato un modello fuzzy che combina l'essenza della psicologia costruttiva personale con la teoria basata sulla logica fuzzy [Anderson 1998]. Yager, invece, ha rappresentato un esempio interessante di come utilizzare i modelli di logica fuzzy per la costruzione di sistemi intelligenti. Il processo di costruzione del modello è suddiviso in due parti [Yager 2000]. La prima fase consiste nel ripartire le variabili in termini di espressioni linguistiche del linguaggio naturale. Detta suddivisione, una caratteristica precipua di ciò che Lofti Zadeh definisce "esecuzione di calcoli mediante parole", semplifica enormemente la costruzione dei modelli. La fase successiva di questo processo consiste nel rappresentare questi concetti linguistici in termini di sottoinsiemi fuzzy. Ad esempio, il concetto "giovane" è rappresentato come sottoinsieme fuzzy "Young" su un insieme di età.

Nella costruzione dei modelli, supportati dalla logica fuzzy, un termine fuzzy può essere dipendente anche dal contesto corrente. Ad esempio, il termine "vecchiaia" potrebbe aver significati diversi per l'insieme degli studenti universitari e per l'insieme dei docenti. A 28 anni, uno studente universitario potrebbe essere considerato vecchio, ma un professore ritenuto giovane. Pertanto, il modello fondato sulla logica fuzzy sarebbe il miglior mezzo nella descrizione del ragionamento basato sul contesto corrente.

Riferimenti

Anderson J.R. (1998), in Alastair A., 'A Model for Fuzzy Personal Construct Psychology', in Fuzzy Systems Design (Studies in Fuzziness and Soft Computing)", ed. By Reznik L., Dimitrov V. and Kacprzyk J., Springer-Verlag, pp 153-170.

Brusilovskiy, P.L. (1994). 'The Construction and Application of Student Models in Intelligent Tutoring Systems'. Journal of Computer and Systems Sciences International 32 (1), pp.70-89.

Clancey,W.J. (1986). 'Qualitative Student Models'. Annual Review of Computer Science 1, pp.381-450.

Carbonell,J.R. (1970). AI in CAI: 'An Artificial-Intelligence Approach to Computer-Assisted.' Instruction. IEEE Transactions on Man-Machine Systems MMS-11, December 1970, 4.

McTaggart, J. (2001) 'Intelligent Tutoring Systems and Education for the Future.' 512X Literature Review, April 30 2001, pp. 2

Self, J.A. (1988). 'Student Models: What use are they?' in: Ercoli,P., Lewis,R., (Eds.), Artificial Intelligence Tools in Education. Elsevier Science Publishers B.V., North Holland, pp. 73-86.

Yager, R. R.(2000). 'Target on E-commerce marketing using Fuzzy intelligent agents'. IEEE Intelligent System.

Zadeh, L. A. (1996). 'Fuzzy logic = Computing with words'. IEEE Transactions on Fuzzy System, 4(2): 103-111.

CAPITOLO II

Il Sistema di Tutoring I.T.W.

1.0 Introduzione

L'e-learning sta acquisendo importanza sempre maggiore negli ambienti didattico-formativi moderni grazie ai suoi innegabili vantaggi rispetto alla tradizionale formazione in aula. Purtroppo le piattaforme di e-learning attualmente esistenti sulla scena tendono a sfruttare la tecnologia solo come veicolo dell'esperienza formativa piuttosto che come regista della stessa.

Negli ultimi anni il radicale cambiamento nel modo di intendere il binomio insegnamento/apprendimento ha portato, a livello mondiale, numerose scuole ed università ad integrare nel loro modo di fare didattica le più moderne ed avanzate tecnologie basate principalmente sul Web. La stessa tendenza si è avvertita nell'ambito della formazione professionale dato che i sistemi tradizionali basati su lezioni frontali in aula hanno da sempre rappresentato, per le aziende, costi elevati sia per l'implementazione che in termini di perdita di produzione.

Al contrario, la formazione a distanza attraverso le nuove tecnologie assicura vantaggi notevoli, primo tra tutti un'estrema flessibilità di tempo e di spazio: il discente non è più costretto ad essere presente nel medesimo luogo dell'insegnante e può studiare anche da casa quando e quanto vuole. Se a questo aggiungiamo il miglioramento dell'accesso all'istruzione, l'aumento della qualità del contenuto formativo, una sua gestione più flessibile, la possibilità di misurare facilmente i risultati e la diminuzione dei costi, capiamo perché la formazione a distanza è al giorno d'oggi molto appetita in tutti gli ambienti didattico/formativi.

Purtroppo gli attuali sistemi di didattica a distanza non sono privi di difetti. La principale pecca dei sistemi attualmente in commercio è che essi non sfruttano appieno le potenzialità del mezzo che hanno a disposizione utilizzandolo come mero veicolo di informazione e non come strumento capace di elaborare tale informazione in maniera intelligente e personalizzata.

Il presente capitolo descrive IWT, una piattaforma di e-learning che si propone di superare questo limite mirando a personalizzare l'apprendimento sulle reali esigenze e preferenze dell'utente ed a garantire estensibilità e flessibilità non solo al livello dei contenuti ma anche nelle funzionalità e soprattutto, a livello più alto, nelle strategie e nei modelli. Dopo un'introduzione sui limiti degli attuali sistemi di e-learning, verrà data una panoramica di IWT e verranno di seguito descritte le caratteristiche "intelligenti" ed i modelli che hanno permesso la loro implementazione. Infine si concluderà facendo il punto sullo stato attuale della ricerca su IWT ed ipotizzando possibili scenari futuri.

2.0 IWT: Intelligent Web Teacher

IWT è una piattaforma per l'apprendimento a distanza realizzata con il preciso intento di gettare le basi per l'e-learning di futura generazione. IWT mira, infatti, a personalizzare l'apprendimento sulle reali esigenze e preferenze dell'utente ed a garantire estensibilità e flessibilità non solo al livello dei contenuti ma anche nelle funzionalità e soprattutto, a livello più alto, nelle strategie e nei modelli.

IWT è scaturito dalla collaborazione tra una serie di soggetti di ricerca tra cui il Centro di Ricerca in Matematica Pura ed Applicata , il Dipartimento di Ingegneria dell'Informazione e Matematica Applicata dell'Università di Salerno , il Centro di Eccellenza in Metodi e Sistemi per l'Apprendimento e la Conoscenza ed il Dipartimento di Informatica ed Automazione dell'Università degli Studi "Roma Tre". Il know-how che ne

ha consentito la realizzazione nasce da una lunga esperienza dei soggetti interessati nel campo del distance learning intelligente acquisita nell'ambito di vari progetti di ricerca nazionali ed europei tra cui Diogene [sito web], InTraServ [sito web] e m-learning [sito web].

Completamente basato su Web, le caratteristiche innovative di IWT rispetto alle altre soluzioni di e-learning attualmente in commercio sono:

- possibilità di generazione automatica o assistita dei percorsi didattici a partire dagli obiettivi di apprendimento;
- possibilità di personalizzazione automatica dei corsi sulla base delle conoscenze pregresse dei singoli discenti e delle loro preferenze di apprendimento;
- supporto al monitoraggio ed alla valutazione automatica dei discenti sia in relazione alle conoscenze acquisite che alle abilità cognitive e capacità percettive mostrate;
- possibilità di gestione dei contenuti ad un alto livello di astrazione tramite ontologie mantenute in conformità con i maggiori standard per la rappresentazione della conoscenza;
- possibilità di estendere la piattaforma tramite Plug-In che consentono l'aggiunta di nuovi servizi e tramite Driver che consentono la gestione di nuove tipologie e formati di contenuto.

IWT è stato realizzato completamente in ambiente Microsoft.NET e fa ampio uso dei maggiori standard in circolazione per la rappresentazione delle strutture dati relative ai Metadata (standard IMS-LOM), ai Test (standard IMS-QTI), alle Ontologie (standard SHOE e DAML+OIL), ai Corsi (standard SCORM e IMS-CP) e dalle Informazioni sugli Studenti (standard IMS-LIP).

In questo capitolo ci soffermeremo, in particolare, sulla descrizione della terna di modelli (e delle relative regole di evoluzione) che costituiscono la base delle caratteristiche "intelligenti" di IWT: il modello

della conoscenza, il modello studente ed il modello didattico che saranno oggetto dei prossimi paragrafi.

3.0 Il Modello della Conoscenza

Il Modello della Conoscenza di IWT consta di tre livelli di astrazione. Il livello più basso è costituito dai *Learning Object* ovvero dai moduli didattici elementari che possono essere usati nel corso della formazione. Il secondo livello è costituito dai *Metadata* il cui compito è di descrivere in maniera formale i Learning Object attraverso un insieme standard di attributi. IWT offre, inoltre, il supporto opzionale alla gestione della conoscenza ad un livello più alto di astrazione (terzo livello) orientato ai *Concetti* piuttosto che ai Learning Object attraverso le *Ontologie* [Fensel 2001].

La figura 4 schematizza i tre livelli del *Modello della Conoscenza* di IWT. In particolare, per la rappresentazione del secondo livello (*Metadata*), IWT adotta lo standard IMS-LOM che prevede la descrizione del Learning Object attraverso 47 elementi raggruppati in 9 categorie.

Purtroppo i *Metadata* si limitano a fornire informazioni circa le singole risorse ma nulla dicono circa le relazioni che intercorrono tra di esse o, meglio, tra i concetti coinvolti in tali risorse. Tali informazioni sono, del resto, necessarie per offrire le funzionalità “intelligenti” di valutazione automatica dello studente e di generazione automatica dei percorsi didattici. Per superare questo limite IWT consente di lagare ad ogni Learning Object uno o più concetti appartenenti ad un dominio didattico e di organizzare separatamente i domini didattici attraverso le *Ontologie*.

Le *Ontologie*, in IWT, sono strutture a grafo che consentono di descrivere formalmente un dominio didattico attraverso la specificazione di un vocabolario di concetti e l'identificazione delle relazioni intercorrenti tra essi.

Le ontologie di IWT rispettano gli standard SHOE e DAML+OIL e supportano le seguenti relazioni:

- Relazione B (Belongs to) per implementare una gerarchia di concetti. $c B d$ significa che il concetto c è parte del concetto d . Sia $C = \{c \mid c B d\}$ è possibile dire che, per apprendere d , è necessario e sufficiente apprendere tutti i concetti appartenenti a C .
- Relazione R (Requires) per implementare la propedeuticità. $c R d$ significa che d è pre-requisito per c . Sia $D = \{d \mid c R d\}$ è possibile affermare che, per apprendere c , è necessario aver appreso preliminarmente tutti i concetti appartenenti a D .
- Relazione SO (Suggested Order) per implementare una propedeuticità più lasca. $c SO d$ significa che, se occorre apprendere c e d , è conveniente apprendere c dopo d .

Attraverso la relazione implicita E (Explained by) mantenuta nei Metadata è possibile, infine, collegare ciascun Concetto ai Learning Object che spiegano tale concetto. $c E l$ significa che il concetto c è spiegato nel Learning Object l e $L = \{l \mid c E l\}$ è l'insieme di tutti i Learning Object che spiegano il concetto c mentre $C = \{c \mid c E l\}$ è l'insieme di tutti i concetti spiegati dal Learning Object l .

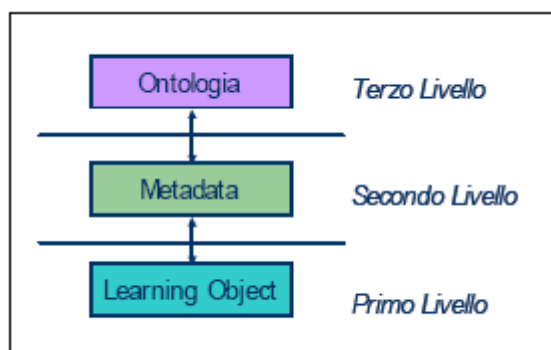


Figura 4 I tre livelli della conoscenza IWT

4.0 Il Modello Studente

Il Modello Studente di IWT è in grado di catturare (in maniera automatica) le conoscenze acquisite dai discenti durante l'esperienza formativa e le preferenze di apprendimento mostrate rispetto a importanti parametri pedagogici quali: media, approccio didattico, livello di interazione, densità semantica, ecc. Il modello è composto da tre elementi: uno *Stato Cognitivo*, un insieme di *Preferenze di Apprendimento* ed un insieme di *Regole di Evoluzione* ed è mantenuto in conformità con lo standard IMS-LIP.

Lo *Stato Cognitivo* si interessa di rappresentare, per ciascuno studente, le conoscenze possedute in un determinato istante sotto forma di insieme di coppie concetto-valutazione. Più formalmente, lo *Stato Cognitivo* dello studente s al tempo t può essere espresso come $CS_{s,t} = \{B_1, B_2, \dots, B_n\}$ dove ogni B_i (belief) rappresenta la coppia $B_i = \langle O_i(c_i), e_i \rangle$ dove $O_i(c_i)$ è il c_i -esimo concetto dell'ontologia O_i mentre e_i è un valore fuzzy, che rappresenta il grado di conoscenza del concetto da parte dello studente.

Si sceglie un valore fuzzy per e_i per tener conto anche dell'affidabilità della stima. Tale valore è, a sua volta, rappresentato dalla coppia $e_i = \langle d_i, r_i \rangle$ dove $0 \leq d_i \leq 1$ (degree) è il grado di conoscenza stimato dal sistema del concetto $O_i(c_i)$ mentre $0 \leq r_i \leq 1$ (reliability) è il grado di affidabilità di tale valutazione. Tale grado tende ad 1 con l'aumentare dei momenti di verifica superati dallo studente nel caso di risposte non contraddittorie.

Ad esempio, il belief: $\langle \text{limiti}, \langle 0.2, 0.9 \rangle \rangle \in CS_{s,t}$

significa che la conoscenza dello studente s al tempo t , relativa al concetto *limiti*, è solo del 20% ed il sistema è sicuro al 90% di questa valutazione.

Le Preferenze di Apprendimento si riferiscono ai campi della categoria Educational del metadata secondo lo standard IMS-LOM ovvero: Interactivity Type, Learning Resource Type, Interactivity Level, Semantic

Density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Language.

LePreferenze di Apprendimento dello studente s al tempo t possono essere espresse come $LP_{s,t} = \{P_1, P_2, \dots, P_m\}$ dove $P_i = \langle s_i, e_i \rangle$ è la singola preferenza composta da una affermazione s_i (statement) circa uno dei campi sopra elencati ed una valutazione fuzzy e_i (evaluation) della veridicità di tale affermazione.

La sintassi di s_i è la seguente: “ $f_i = v_i$ ” dove f_i (field) appartiene all’insieme di campi succitati mentre v_i è uno dei valori ammissibili per f_i secondo lo standard IMS-LOM (ad esempio i valori ammissibili per il campo “learning_resource_type” sono “text”, “slide”, “exercise”, “simulation”, ecc; i valori ammissibili per “semantic_density” sono, invece, “low”, “medium” e “high”). Ad esempio, la preferenza:

$$\langle \text{“learning_resource_type = text”}, \langle 0.7, 0.2 \rangle \rangle \in LP_{s,t}$$

significa che lo studente s al tempo t ha una preferenza del 70% relativa alla tipologia di risorsa “text” ma l’affermazione ha solo il 20% di affidabilità.

Il *Modello Studente* viene dedotto e continuamente aggiornato da IWT attraverso le *Regole di Evoluzione*. In particolare, l’aggiornamento dello *Stato Cognitivo* avviene al termine di ogni attività di verifica dello studente considerando i risultati ottenuti ai test (che, essendo Learning Object, sono essi stessi legati ai concetti del dominio) e mediandoli con i risultati dei test precedenti relativi agli stessi concetti. Per l’aggiornamento delle *Preferenze di Apprendimento*, invece, si ricorre all’osservazione congiunta del materiale didattico utilizzato e delle conoscenze acquisite al fine di determinare il grado di ricettività dello studente ai vari tipi di stimoli derivanti dalle varie tipologie di materiale.

5.0 Il Modello Didattico

Il *Modello Didattico* di IWT definisce le modalità ottimali di trasferimento della conoscenza del dominio agli studenti in base alla

disciplina (formalizzata nel *Modello della Conoscenza*) ed alle caratteristiche dello studente coinvolto (formalizzate nel *Modello Studente*). Tramite questo modello IWT è in grado di personalizzare l'esperienza didattica sulla base delle conoscenze pregresse dei singoli discenti e delle loro preferenze di apprendimento.

La struttura di base del *Modello Didattico* è il *Corso* rappresentato in conformità con gli standard IMS-CP e SCORM e composto dalle seguenti strutture:

- l'insieme di *Concetti Obiettivo* del corso (tra quelli che lo studente conoscerà alla fine del corso, quelli che si trovano al punto più alto nella gerarchia *Belongs To* dell'ontologia di riferimento);
- il *Learning Path* ovvero la sequenza di concetti che si dovrà trasferire allo studente affinché egli apprenda, al termine del corso, i *Concetti Obiettivo* (in tale sequenza di concetti, i punti di verifica saranno identificati attraverso delle *Milestone*);
- la *Presentazione* ovvero la sequenza di Learning Object corrispondente al *Learning Path* concettuale che permetterà l'effettivo trasferimento di conoscenza allo studente dei *Concetti Obiettivo* del corso;
- un *Metadata* conforme ai già citati standard IMS-CP e SCORM.

Dato un insieme di *Concetti Obiettivo* definiti dal docente o dallo stesso studente, IWT sarà in grado di generare, in primo luogo, il migliore *Learning Path* per un dato studente a partire dal suo *Stato Cognitivo* (eliminando cose già conosciute ed aggiungendo eventuali pre-requisiti mancanti) e, a partire dal *Learning Path* così generato, sarà in grado di generare la migliore *Presentazione* per un dato studente a partire dalle sue

Preferenze di Apprendimento (scegliendo, dunque, i Learning Object più congeniali).

Una volta generata, la *Presentazione* potrà essere fruita dallo studente. In generale, la fruizione inizia con il primo Learning Object (o l'ultimo Learning Object attivato nel caso in cui lo studente abbia sospeso la fruizione del corso) e prosegue fino a che non viene raggiunta la fine di un momento di verifica composto da uno o più test (*Milestone*). In questo ultimo caso avviene l'*Adattamento del Corso* e l'*Evoluzione del Modello Studente*.

L'Adattamento del Corso, prevede la modifica della parte di Presentazione non ancora visionata dallo studente (e del relativo Learning Path) per rispondere ad eventuali lacune riscontrate attraverso la somministrazione di materiale di recupero.

6.0 Conclusioni e Sviluppi Futuri

In questo capitolo abbiamo presentato una piattaforma di e-learning "intelligente" capace di personalizzare l'apprendimento sulle reali esigenze e preferenze dei singoli, compatibile con i maggiori standard presenti sulla scena dell'e-learning e con precise caratteristiche di estensibilità e flessibilità. IWT è disponibile come prodotto nella versione 1.0. Al momento esiste una versione personalizzata di IWT per la formazione dei manager d'impresa che include quattro corsi (Business Decision, Marketing, Ricerche di Mercato e Controllo di Gestione) in tre lingue (italiano, inglese, spagnolo) e due business games che trattano il processo decisionale di business attraverso un approccio di tipo "what...if" (Decisioni Strategiche e Controllo di Gestione). Tale versione è in corso di sperimentazione con un consorzio di imprese italiane, inglesi e spagnole nell'ambito del progetto InTraServ finanziato dalla EC.

Un'ulteriore versione di IWT per la didattica in ambito universitario è già pronta e tra breve comincerà la sperimentazione all'Università di

Salerno, di Roma³ e del Molise. Una versione di IWT personalizzata per l'ECM (Educazione Continua in Medicina), inoltre, è in corso di realizzazione e verrà sperimentata nel progetto GeCoSan.

Parallelamente a questi sforzi di verticalizzazione e sperimentazione, il lavoro di ricerca scientifica su IWT continua. Il progetto Diogene attualmente in corso ha come fine la realizzazione di caratteristiche aggiuntive per IWT. Tra queste citiamo: strategie di apprendimento dinamiche, apertura al Semantic Web, servizi Web per la gestione dei Learning Object, servizi per la ricerca e la creazione di curriculum vitae, supporto agli insegnanti free-lance e definizione assistita di Obiettivi Didattici.

Riferimenti

Capuano, N., Gaeta, M., Micarelli (2003) “*IWT: Una Piattaforma Innovativa per la Didattica Intelligente su Web*”.

http://www.cmpa.it/intraserv/documents/AIXIA_2003.pdf

Fensel, D., (2001) ‘Ontologies: a Silver Bullet for Knowledge Management and Electronic Commerce’. Springer.

Sito Web del progetto FP5 ‘*Diogene: a Training Web Broker for ICT Professionals*’ (IST-2001-33358): <http://www.diogene.org>.

Sito Web progetto FP5 ‘*InTraServ: Intelligent Training Service for Management Training in SMEs*’. (IST- 2000-29377): <http://www.intraserv.org>.

Sito Web progetto FP5 ‘*m-Learning: Mobile Communication Technologies for Young Adults Learning and Skills Development*’ (IST-2000-25270): <http://www.m-learning.org>.

CAPITOLO III

Un ITS fondato sul binomio CORBA/WWW

1.0 Introduzione

In questo capitolo verrà proposto un sistema di tutoring intelligente e individualizzato, implementato sul World Wide Web e capace di utilizzare applicazioni CGI (Common Gateway Interface) secondo le specifiche CORBA (Common Object Request Broker Architecture) [OMG 95] al fine di agevolare le interazioni tra l'utente ed il sistema.

Questo ITS può supportare, per ogni studente, un ambiente concorrente di apprendimento e una pianificazione dinamica delle lezioni utilizzando variabili linguistiche fuzzy [Yoon & Wang 94].

Il comportamento dello studente verrà esaminato all'interno di strutture di apprendimento chiamate unità didattiche, capaci di valutarne i progressi ottenuti e di conseguenza fornire allo studente lezioni adeguate al suo grado di preparazione.

Il sistema è stato progettato ed implementato, presso il dipartimento di ingegneria informatica e di scienze dell'informazione dell'università di Inha, sud Corea, dai professori Lee S. (seihoon@true.inhatc.ac.kr) e Wang C. (cjwangse@dragon.inha.ac.kr).

2.0 Cosa è CORBA?

Il Common Object Request Broker Architecture (CORBA), è la risposta dell'Object Management Group al bisogno di interoperabilità tra il numero crescente di prodotti hardware e software oggi disponibili. In parole semplici CORBA consente alle applicazioni di comunicare tra di loro senza tenere conto di dove siano poste o di chi le abbia progettate(...). L'ORB è il mediatore che stabilisce le relazioni client-server tra gli oggetti. Tramite l'utilizzo di un ORB un client può invocare in maniera trasparente un metodo su un oggetto del server, che può trovarsi sulla stessa macchina o in rete. L'ORB intercetta la chiamata e si prende l'incarico di cercare un oggetto che possa realizzare la richiesta, gli passa i parametri, ne invoca i metodi e restituisce il risultato. Il client non deve essere al corrente di dove sia posto l'oggetto, del suo linguaggio di programmazione, sistema operativo o di ogni altro aspetto di sistema che non sia parte dell'interfaccia di un oggetto. In questa maniera, l'ORB realizza l'interoperabilità tra applicazioni su macchine differenti in ambienti eterogeneamente distribuiti e connette in maniera trasparente sistemi formati da molti oggetti.

Nella scrittura di applicazioni client/server gli sviluppatori utilizzano una loro ideazione o uno standard affermato per definire il protocollo da usare tra i dispositivi. La definizione del protocollo dipende dall'implementazione del linguaggio, dal trasporto di rete e da una dozzina di altri fattori. Gli ORB semplificano questo processo. Grazie all'ORB il protocollo viene definito attraverso le interfacce delle applicazioni con delle specifiche indipendenti dal linguaggio di unica implementazione, IDL (Interface Description Language). E gli ORB forniscono flessibilità. Essi permettono ai programmatori di scegliere i sistemi operativi più appropriati, gli ambienti di esecuzione ed i linguaggi di programmazione da utilizzare per ogni componente di un sistema in costruzione. Ancora più importante essi permettono l'integrazione di componenti già esistenti. In una soluzione

basata su ORB gli sviluppatori semplicemente modellano i vecchi componenti utilizzando lo stesso IDL che viene usato per creare nuovi oggetti, quindi scrivono del codice "wrapper" che fa da traduttore tra il bus standard e le vecchie interfacce.

2.1 Quali sono i componenti di CORBA?

In questo paragrafo saranno inclusi le definizioni dei termini più usati nel lessico CORBA, in modo tale che si possa capire cosa si intende per un determinato componente CORBA.

- *Object Request Broker* - (o ORB) un ORB è un pezzo di "middleware", come si dice, che risiede tra i client ed i server e rende possibile una semplice comunicazione tra di loro. L'ORB è un'entità concettuale, che a volte prende la forma di una libreria condivisa ed altre volte la forma di un programma esterno. L'ORB è responsabile di stabilire e distruggere sessioni tra client e server, del controllo delle azioni, del trasporto dei messaggi fra di essi durante una sessione.
- *Object Adaptor* - (o OA) un Object Adaptor mette a disposizione il canale attraverso il quale un oggetto server (come il pannello gnome) comunica con l'Object Request Broker (ORBit).
- *GIOP/IOP* - CORBA prende tutti questi linguaggi orientati agli oggetti e li trasporta attraverso componenti software orientati agli oggetti (marshaling, trasporto e demarshaling). Se due componenti software sono in locazioni differenti, verrà usato il protocollo GIOP per spostare le informazioni fra di loro. IOP è una specifica di GIOP per la suite del protocollo Internet; teoricamente si potrebbe implementare

GIOP anche su qualcos'altro che IP nel quale caso verrebbe chiamato in maniera differente.

3.0 L'architettura modulare del sistema

La rappresentazione di un sistema di tutoring ipermediale ed intelligente è proposta in figura 5.

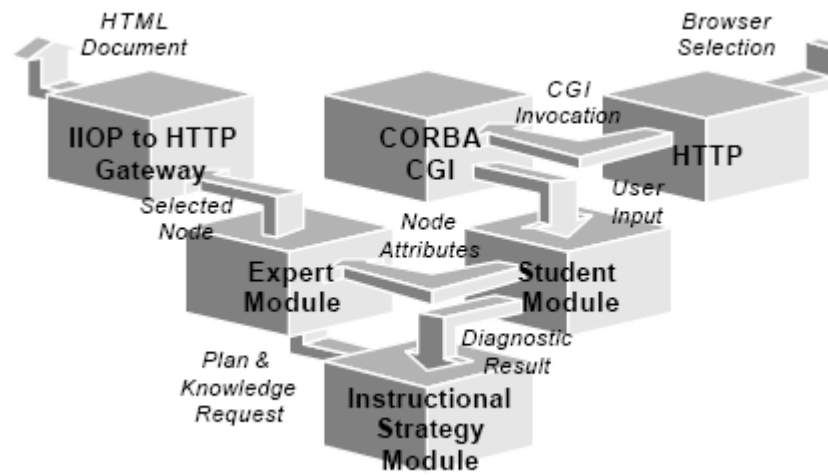


Figura 5. La configurazione del sistema Corba-based

Questa configurazione, simile ad un ITS stand-alone, è composta da un modulo esperto, un modulo studente ed infine un modulo pedagogico.

In aggiunta, il sistema al fine di consentire l'interazione degli studenti con il modulo studente, utilizza il protocollo HTTP (HyperText Transfer Protocol) ed un insieme di applicazioni CGI scritte secondo le specifiche Corba. Mentre, per poter effettuare la trasmissione dei documenti in formato html allo studente, utilizza un Gateway da IOP (Internet Inter Orb Protocol) verso HTTP.

Per quanto riguarda la struttura modulare, si può dire che il modulo studente è capace di diagnosticare il livello corrente dello stato d'apprendimento dello studente, deducendolo dall'informazioni, relative a ciascun studente, presenti all'interno di un database.

A seconda del livello di apprendimento raggiunto, il modulo pedagogico decide il tipo di lezione successiva, trasmettendola all'interno di una unità didattica sotto forma di documento html.

Questa trasmissione, avviene attraverso il gateway iiop to http, in modo che lo studente possa prenderne visione usando qualsiasi browser www.

Infine il modulo studente userà gli applicativi CGI per poter ricevere i dati dell'apprendimento raggiunto dallo studente, questo perché lo standard http non è in grado di riceverlo.

Tutto l'input dello studente passa attraverso le applicazioni CGI per poi elaborarlo per poter decidere una strategia migliore d'insegnamento.

4.0 Il modulo esperto

Il dominio della conoscenza sarà usato nella fase di insegnamento, nel senso che esso sarà capace di riconoscere l'eventuale errore concettuale dello studente e di fornire i mezzi adeguati per porvi rimedio.

Il dominio ha una rappresentazione a rete duale, la IUN (Instructional Unit Network) e la CN (Conceptual Network).

La IUN è una rete di conoscenza che include i mezzi forniti allo studente e il livello di apprendimento raggiunto.

Essa è configurata come una rete gerarchica, dove ciascun nodo rappresenta un'unità didattica, la cui conoscenza è propedeutica ad un'altra unità didattica subordinata, ad essa collegata. In realtà sono presenti due tipi di unità didattiche: quella dedita alla spiegazione e quella che costruisce i problemi da risolvere.

La prima, possiede il materiale didattico di vario tipo, grafici, immagini, suoni e filmati.

La seconda, utilizza i concetti di difficoltà e importanza per diagnosticare lo stato d'apprendimento dello studente.

La figura 6 evidenzia la costruzione della IUN e della CN.

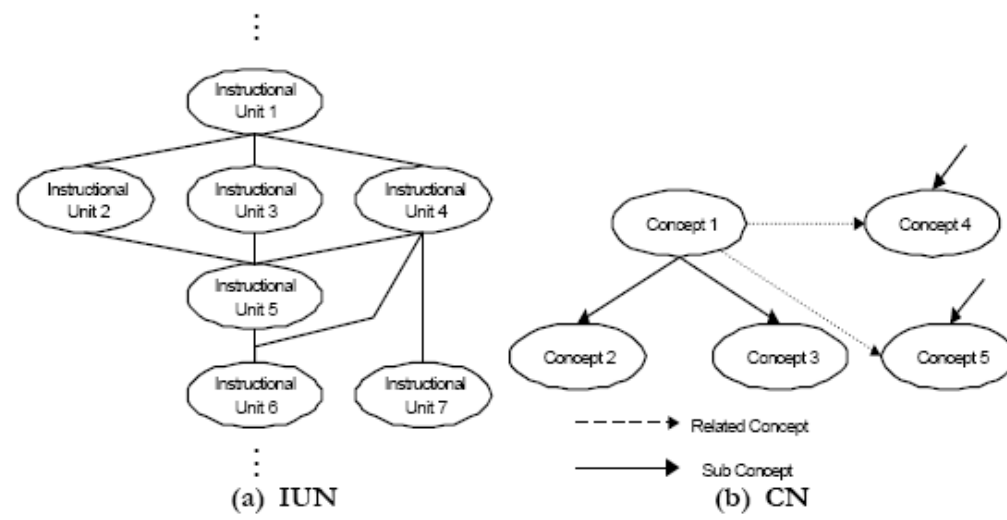


Figura 6. La rappresentazione delle reti IUN (a) e CN (b)

La CN è una rete di conoscenza che raggruppa la conoscenza da impartire allo studente. Ciascun nodo rappresenta un concetto didattico e i collegamenti tra i concetti indicano la loro interdipendenza. Usando queste relazioni, possibile generare problemi dinamicamente e diagnosticare eventuali errori dello studente.

5.0 Il modulo studente

Il modulo studente è essenzialmente un modello di tipo overlay sul dominio esperto. La conoscenza acquisita dallo studente è salvata all'interno del modulo studente, ed è molto simile alla rete IUN del modulo esperto. Infatti, con il progredire dell'apprendimento, la conoscenza dello studente, presente nel database, prende la forma di una rete simile alla IUN della conoscenza esperta. I risultati ottenuti dallo studente (compresi gli errori), i problemi e gli esempi sono salvati nei nodi della rete della conoscenza dello studente, che in altri termini mi descrive il percorso di apprendimento dello studente. Il modello studente, principalmente, valuta il risultato ottenuto dallo studente all'interno di una unità didattica, usufruendo delle variabili linguistiche.

La tabella successiva mostra le variabili linguistiche ed i relativi valori, associate sia ad una singola unità didattica che ad un risultato ottenuto:

Tabella 1 Le variabili Linguistiche

	Unità didattica	Diagnosi del risultato
Difficoltà	D,M,E	G,M,B
Importanza	I,M,Ni	
modificatore	[Very] , [Less]	

D=Difficile, M=Medium, E=Easy

I=Important, Ni=Not Important, G=Good, B=Bad

Il concetto di variabile linguistica è il kernel della teoria Fuzzy che permetterà di rappresentare il pensiero umano usando il computer.

Usando la concentrazione e la dilatazione presenti nella teoria fuzzy, si può definire la funzione che valuta la conoscenza corrente dello studente.

Stabilito che, le variabili linguistiche “difficoltà” e “importanza” sono entrambe fissate in termini di dilatazione del valore fuzzy “medio”, così come i valori “facile” e “non importante” in termini di concentrazione di “medio”, allora si può definire la funzione DV(Diagnostic Value) come segue:

$$DV (R_{n-1}) = norm \left(\sum_{i=1}^n (R_c / R_n)^k \right)$$

norm = è una funzione di normalizzazione che porta il risultato nell'intervallo [0,1].

R_{n-1} è il risultato del problema n-1 esimo

R_c è il numero di risposte esatte secondo le variabili linguistiche

R_n è il numero totale di problemi sottoposti allo studente

K è l'esponente collegato alle variabili linguistiche definito come segue:

$$I \text{ o } D = \frac{1}{2}, \quad M=1, \quad N_i \text{ oppure } E = 2$$

Per poter usare diversi valori associati alle variabili linguistiche “importanza” e “difficoltà” si possono introdurre dei modificatori fuzzy “very” o “less”. Il risultato della funzione DV è un numero reale all’interno dell’intervallo [0,1].

Per poter eseguire un’inferenza fuzzy usando le variabili linguistiche, il risultato ottenuto della funzione DV deve essere fuzzificato. Il risultato ottenuto dalla fuzzificazione è una variabile linguistica il cui valore sarà “good”, “medium” “bad” abbinato ai modificatori very o less.

La figura successiva mostra la rappresentazione grafica della fuzzificazione del valore diagnostico.

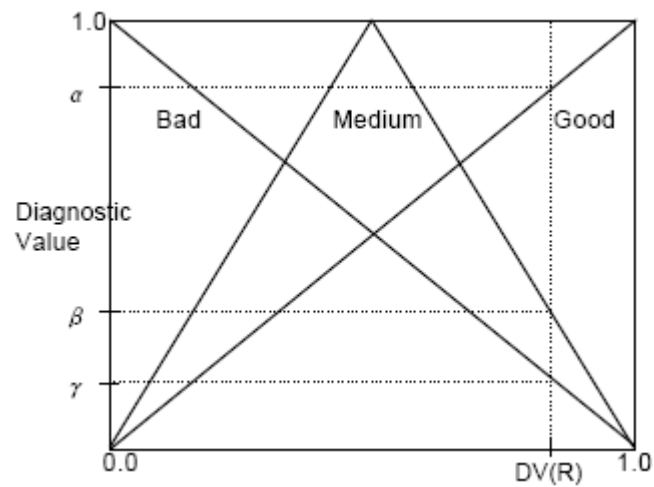


Figura 7. Il grafo della funzione DV

6.0 Conclusioni

In questo capitolo, si è progettato ed implementato un sistema di insegnamento a distanza che supporta l’interazione con gli studenti, e capace di creare strategie di insegnamento dinamiche. Il sistema proposto consta di un modulo esperto, un modulo studente e un modulo di tipo pedagogico. Il modello studente può supportare l’interazione con gli studenti attraverso le applicazioni CGI implementate sul WWW,

diagnosticarne la conoscenza corrente usando le informazioni personali e le risposte fornite.

Riferimenti

Lee S., Wang C., (1997) “*Intelligent Hypermedia Learning System on the Distributed Environment.*” <http://media.inhatc.ac.kr/papers/edmedia97.pdf>

Natabayashi K., Koike Y. and Maruyama M. (1995) ‘*An Intelligent Tutoring System on world wide web*’ in ‘An integrated learning Enviroment on a Distributed Hypermedia, Educational Multimedia and Hypermedia’, Association for the Advancement of Computing in Education, Charottesville, VA pagg. 488-493

Omg (1995) Corba 2.0 <http://omg.org/corba/iiop/html> .

Yonn K.S. and Wang C.J. (1994) ‘*Authoring System for the Development of ITS*’ IEEE TENCON, Singapore, 97-101

CAPITOLO IV

Il sistema adattivo ipermediale Alice

1.0 Introduzione

I sistemi adattivi, per definizione, combinano la tradizionale facilità d'uso degli ipermedia con le tecniche avanzate di tutoring degli ITS adattando dinamicamente una sequenza di sessioni educazionali ad un particolare livello di conoscenza dell'utente. Pertanto, il sistema adattivo ipermediale deve avere le seguenti caratteristiche:

- Deve basarsi sull'ipertesto o sugli ipermedia.
- Deve includere un modello dominio composto da un insieme di strati elementari di conoscenza con le relative relazioni.
- Deve possedere un esplicito modello studente che racchiuda tutte le caratteristiche individuali.
- Deve, infine, essere in grado di variare alcune parti visuali o funzionali del sistema in base alle informazioni che provengono dal modello studente.

In altre parole, tali sistemi possono essere considerati alla stregua di sistemi ipermediali che includono degli approcci intelligenti che li mettano in grado di adattarsi alle specifiche dell'utente. Il sistema tipico di ipermedia adattivi consiste di tre componenti: gli ipermedia, il tutoring intelligente e il supervisore [Kavcic 2001].

Il modulo ipermedia contiene l'organizzazione dell'iperspazio del dominio, nonché la rappresentazione e la navigazione. La componente del tutoring intelligente, che è simile alla struttura di un sistema di tutoring intelligente [Wenger 1987], si prende cura di tutte le azioni intelligenti del sistema. Infine, il modulo supervisore regola il funzionamento di entrambe

le componenti anzidette. La fig.8 mostra schematicamente la struttura nel suo insieme.

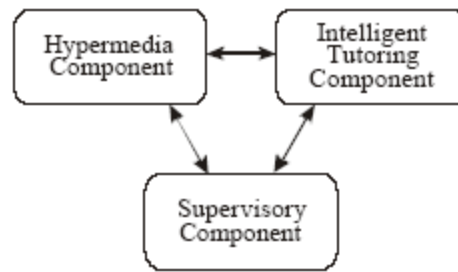


Figura 8 La struttura di un sistema adattivo ipermediale

Affinché, sia in grado di adattarsi ad un particolare utente, il sistema, deve conoscere il dominio della conoscenza, il grado di cognizione e i progressi ottenuti durante la fase di apprendimento. Di conseguenza, per ottenere un adattamento utile ed efficace, il modello del dominio e quello dell'utente hanno un ruolo chiave nella costruzione del sistema [Wu, De Bra, Aerts, Houben 2000]. In particolare, il modello studente, raccoglie informazioni specifiche per ogni singolo utente. Rappresenta la base dell'adattamento del sistema in quanto recupera tutte le informazioni utili riguardanti l'utente. Senza il modello studente, il sistema non sarebbe in grado di distinguere i singoli utenti e li tratterebbe tutti alla stessa stregua.

Un perfetto modello deve includere tutte le caratteristiche dell'utenza, il comportamento e il grado di conoscenza, che possono influenzare sia l'apprendimento che l'efficienza [Wenger 1998].

2.0 Descrizione del sistema Alice

Qui descriveremo un sistema adattivo ipermediale denominato Alice, progettato presso la facoltà di scienze dell'informazione e di calcolo dell'università di Ljubljana [Kavcic 2001] e implementato nella costruzione di un sistema di web-tutoring per l'introduzione della programmazione in linguaggio Java. Trattasi, quindi, di un sistema interamente basato sul web che consiste di applicazioni di tipo client and server. La parte client calcola,

aggiorna e preserva il modello d'utente in uso e fornisce il supporto adattivo alla navigazione. L'applicazione server, implementata come servlets java, fornisce la procedura di login e memorizza i dati del modello utente sul server. All'inizio essa viene impiegata per la procedura di login e, alla fine, per la memorizzazione del modello.

Nel corso dell'uso del sistema da parte dell'utente, il server fornisce soltanto le pagine web richieste. In tal modo, il carico del server e il traffico sulla rete vengono ridotti sensibilmente, mentre le risorse del client sono sfruttate per l'aggiornamento del modello studente.

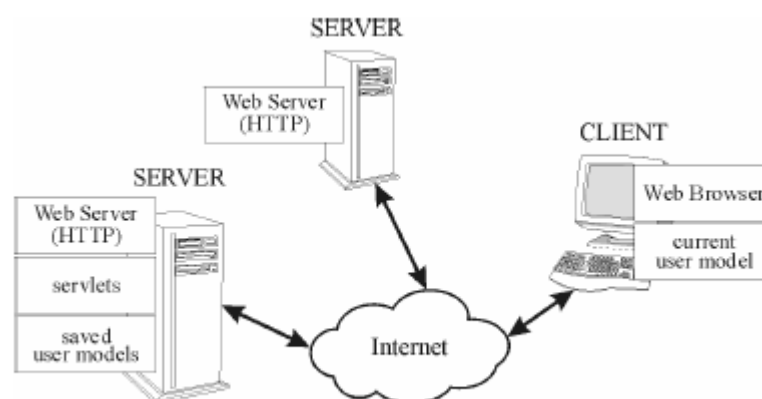


Figura 9 Le funzioni del server e del client nel sistema Alice

Il sistema descritto consiste in una shell di tutoring system ed è pertanto progettato per qualsivoglia ITS web-based. L'adattamento avviene per mezzo delle tecnologie di supporto alla navigazione.

2.1 Il modello utente

Innanzitutto, diciamo che il dominio della conoscenza è partizionato in piccole parti che vengono chiamate concetti. Per cui ogni dominio può essere rappresentato da un insieme finito di concetti (vedi figura 10).

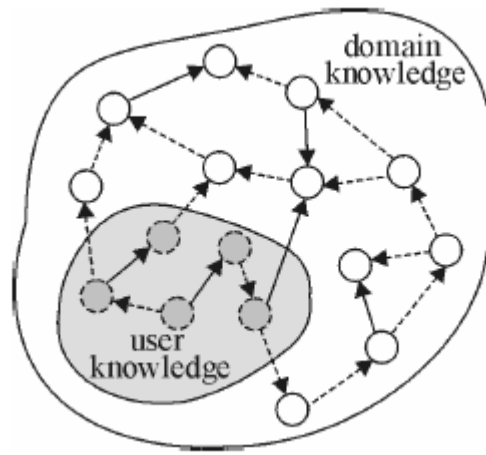


Figura 10...Il dominio della conoscenza esperta e il dominio utente

L'apprendimento viene effettuato stabilendo delle relazioni tra questi pezzi di conoscenza. Nel senso che, due concetti del dominio sono in relazione se la conoscenza del primo è richiesta per una migliore comprensione del secondo. In particolare queste relazioni sono suddivise in essenziali e di supporto. Una relazione viene detta essenziale se la conoscenza di un concetto è strettamente collegata alla conoscenza di un secondo concetto. Nel caso in cui la conoscenza di un secondo concetto non è necessaria ma utile allora la relazione viene detta di supporto. Il modello studente è sostanzialmente una sovrapposizione del modello del dominio della conoscenza, il quale si fonda sugli insiemi fuzzy e sulle regole fuzzy. Per cui, la conoscenza dell'utente, di un concetto del dominio, è rappresentata dalle funzioni di appartenenza relative a tre insiemi fuzzy (concetto ignoto, noto, appreso).

L'aggiornamento del modello si basa sulle regole fuzzy che vengono applicate per dedurre i concetti, essenziali o di supporto, dalla conoscenza del concetto dimostrato (algoritmo della propagazione del valore della conoscenza). Il principio cardine di questo meccanismo di ragionamento è del tutto simile a quello delle reti Bayesiane, salvo per il fatto che i valori sono propagati, in base alle regole fuzzy, operanti su di un grafo composto

dalla connessione dei concetti che ne rappresentano i nodi. Il modello viene inizializzato mediante un pre-test, quando ciascun concetto o è conosciuto oppure è ignoto. Durante la interazione con il sistema, il modello è aggiornato in base alle unità visitate, risultati ottenuti e alla propagazione della conoscenza.

3.0 Le tecnologie adattive

Nei sistemi ipermediali adattivi sono utilizzate sette differenti tecnologie adattive. Esse possono essere divise in due gruppi: il supporto alla presentazione e quello dedicato alla navigazione. La presentazione adattiva o adattamento al livello del contenuto include la presentazione sotto forma di testo e sotto forma di contenuti multimediali.

Qui, l'estetica o il contenuto della pagina ipermediale viene alterata in modo dinamico e può variare nei dettagli, nella spiegazione, nell'uso dei media o nel numero di collegamenti .

D'altra parte, il supporto adattivo alla navigazione, comprende tre tecniche differenti, che forniscono un'assistenza diretta all'utente. In più includono anche il riordinamento adattivo dei collegamenti, l'oscuramento adattivo dei collegamenti e le annotazioni adattive dei links. Il loro scopo principale, è quello di aiutare l'utente ad orientarsi, restringendo l'iperspazio di esplorazione, fornendo aiuti guide e supporti d'orientamento.

Riferimenti

Kavcic A.,(2001) '*Adaptation in Web-based Educational Hypermedia Considering the Uncertainty of User Knowledge*'. Ph.D. Dissertation, Faculty of Computer and Information Science, University of Ljubljana.

Kavcic A., Privosmik M., Marolt M., Divjak S., (2001)" *Educational Hypermedia System ALICE:an Evaluation of Adaptive Features.*"
<http://lgm.fri.uni-lj.si/~alenska/papers/Calie2001.pdf>

Wenger E., (1987) '*Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*'. Morgan Kaufmann Publishers, Inc., Los Altos.

Wu H., De Bra P., Aerts A., and G. Houben G.J.,(2000) '*Adaptation Control in Adaptive Hypermedia Systems, Adaptive Hypermedia and Adaptive Web-Based Systems*'. Proceedings of the International Conference AH, Trento, Italy, August 2000, Lecture Notes in Computer Science 1892, Springer Verlag, 2000. pp 250-259

CAPITOLO V

La Piattaforma InterMediActor

2.0 Introduzione

Come si è visto, in precedenza, la capacità di adattare l'istruzione alle esigenze di un singolo utente è una caratteristica essenziale dei sistemi educativi moderni.

Per offrire l'adattamento del sistema si possono adottare diverse tecniche, una di queste è rappresentata dal supporto alla navigazione. Questa tecnica aiuta lo studente nella ricerca di un percorso ottimale individualizzato attraverso il materiale d'apprendimento. La navigazione personalizzata, avviene tramite il modello studente che raccoglie i dati vitali riguardanti ciascun singolo studente.

Il sistema proposto, InterMediActor, nato dalla collaborazione dei dipartimenti di ingegneria delle comunicazioni e di informatica delle università di Madrid e di Ljubljana rispettivamente, è un sistema di tipo educativo, che fornisce, non solo, un ambiente per la progettazione, la produzione, e il riutilizzo di contenuti istruttivi, ma anche una valutazione del comportamento dello studente. Esso, sostanzialmente, si basa su un modello predittivo, costruito sui dati raccolti dalla precedente iterazione dello studente con il sistema.

Il grafico della navigazione (costituito per ogni singolo utente) viene aggiornato con l'avanzamento dello studente all'interno del corso.

Questo capitolo, si soffermerà sulla costruzione del modello studente di tipo fuzzy in un sistema educativo basato sulla competenza.

Inizialmente, verrà presentata una piattaforma, InterMediActor, di insegnamento a distanza, implementata su computers, all'interno della quale è sviluppato un modello studente.

Successivamente, si descriverà il meccanismo di inferenza fuzzy che rappresenta il cuore del modello. Tale meccanismo userà le variabili linguistiche per la deduzione delle informazioni che sono di maggiore interesse per un tipo di studente.

2.0 La piattaforma InterMediActor

Il modello educativo, in InterMediActor, [Pedraza & altri 2002] [Valverde & altri 2003], si realizza con la decomposizione degli obiettivi d'apprendimento in una struttura gerarchica di competenze corrispondenti. Il metodo proposto per lo sviluppo di un contenuto educativo si suddivide in due fasi:

- l'analisi top-down degli obiettivi d'apprendimento
- la sintesi bottom-up delle competenze.

L'analisi top-down, che è concettualmente mostrata in figura 11 procede partizionando e refinando gli obiettivi d'apprendimento in modo da renderli più dettagliati e concreti.

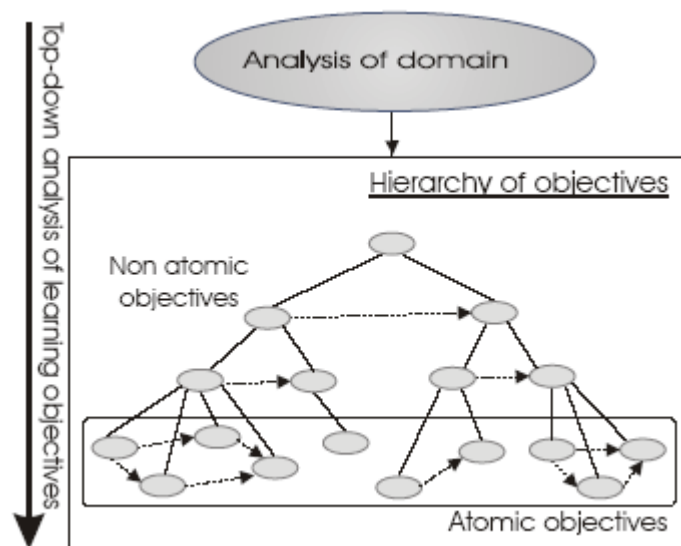


FIGURA 11 Analisi top-down degli obiettivi d'apprendimento

Il processo termina quando gli obiettivi atomici risultano identificati.

Il risultato di quest'ultima analisi, è un grafo eterogeneo delle dipendenze tra gli obiettivi d'apprendimento, con due tipi di relazioni: la relazione di "appartenenza" (part-of relationship) tra un obiettivo e i suoi sotto-obiettivi e una relazione di "dipendenza" (depends-on relationship) tra

gli obiettivi che sono parte dello stesso obiettivo (più in generale). La prima è descritta come un albero gerarchico (grafo gerarchico) e il secondo come un grafo finito delle dipendenze.

Dopo aver eseguito l'analisi top down, degli obiettivi d'apprendimento, tutti gli obiettivi atomici sono trasformati in competenze atomiche (vedi figura 12).

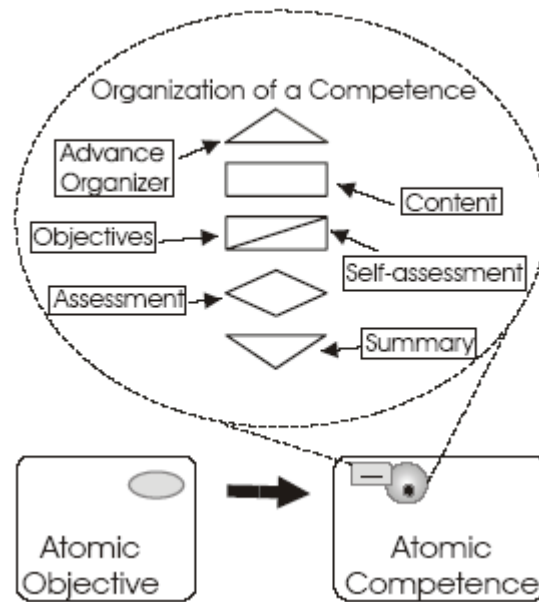


FIGURA 12 La trasformazione di un obiettivo atomico in una competenza atomica

Queste competenze consistono di :

- un organizzatore all'avanzamento (introduce l'argomento da apprendere)
- un contenuto da apprendere (che può essere un oggetto complesso multimediali)
- un riepilogo (che definisce quale deve essere il risultato dell'apprendimento)
- un insieme di tests (self-assessment , final-assessment) basati sul contenuto introdotto

- i prerequisiti di competenza (che devono essere assimilati prima di cercare di comprendere la competenza specifica; servono a stabilire la relazione di dipendenza tra le competenze della stessa specie).

Una volta specificate le competenze atomiche, si possono sintetizzare (in modo bottom-up) più competenze complesse seguendo la relazione di appartenenza (part-of) indicata del grafo gerarchico.

3.0 Il modello studente

L'obiettivo, principale, del modello studente nella piattaforma InterMediActor è di fornire una personalizzazione della navigazione attraverso il contenuto del corso. Ciò può avvenire, costruendo un grafo della navigazione personalizzato per ogni studente. Il grafo è rappresentato da nodi, che corrispondono alle competenze, di differenti colori; grigio per le competenze già acquisite, rosso per quelle non ancora consentite, e verde per le competenze raccomandate (dal verde chiaro, a quello più scuro a seconda del peso delle competenze). Gli archi del grafo di navigazione stanno a indicare tutti i possibili percorsi che lo studente può intraprendere, svariando dalle competenze studiate al momento a tutte quelle (più o meno) raccomandate (punti verdi).

Il grafo della navigazione è basato sul grafico delle dipendenze che viene fuori dall'analisi degli obiettivi d'apprendimento. Infine, il modello studente proposto consiste di un insieme di coppie (competenza,voto) e può essere considerato come un overlay su l'insieme delle competenze.

Non vi è alcun assegnamento di competenze prima si completi l'apprendimento fornito dal sistema. Pertanto, il modello studente, inizia senza alcun voto (mark) per tutte le competenze (sia esse atomiche che aggregate).

Nella costruzione del grafo della navigazione, si utilizza il modello studente al fine di assegnare tipi di categorie (categorizzare) alle

competenze, a seconda dei livelli di importanza. Nel processo di deduzione dei voti sono presenti alcune imprecisioni, per cui viene usata la teoria degli insiemi fuzzy come mezzo matematico per la gestione dell'incertezza.

Per ciascuna competenza si calcola un livello numerico di importanza e si assegna una categoria corrispondente, che determina il colore del nodo del grafo di navigazione.

Quando lo studente esegue la prova di accertamento finale per una competenza, possono cambiare anche le categorie delle altre competenze. Pertanto, per la componente testata, vengono calcolate nuove categorie e tutte le competenze ad essa collegata (tutte quelle competenze presenti nel grafo delle dipendenze alle quali si accede direttamente partendo dalla competenza nota).

3.1 Il livello d'importanza

A ciascuna competenza, (per quanto riguarda un particolare studente) è assegnata una delle seguenti cinque categorie: “appresa”, “molto importante”, “importante”, “meno importante” oppure “vietata”.

Queste categorie, corrispondono ai termini linguistici (fuzzy) della variabile linguistica *livello di importanza*. Il valore di importanza calcolato, per quella competenza, dipende dal livello di difficoltà della competenza stessa, dal voto (mark) ottenuto dallo studente nella prova d'accertamento finale (per quella competenza), e da quanto bene lo studente conosca o abbia imparato le nozioni fondamentali della competenza.

3.2 Il livello di difficoltà

Quando lo studente inizia il corso, *il livello di difficoltà* è calcolato per ciascuna competenza del corso. Esso dipende dal livello iniziale, specificato dal provider del corso, e dai voti che gli altri studenti hanno ottenuto per questa competenza. I livelli calcolati non variano durante la sessione.

La variabile linguistica *livello di difficoltà* è descritta da una funzione il cui valore è la terna $(\mu_{\text{easy}}, \mu_{\text{norm}}, \mu_{\text{diff}})$. Tale funzione può assumere i seguenti tre valori linguistici: *easy*, *normal*, *difficult*. Il livello iniziale di difficoltà l per una competenza c varia all'interno dell'intervallo $[1,10]$, dove 1 significa che molto facile, mentre 10 significa molto difficile. Questo valore è fuzzificato usando le funzioni d'appartenenza (vedi figura 13) per determinare il valore della terna.

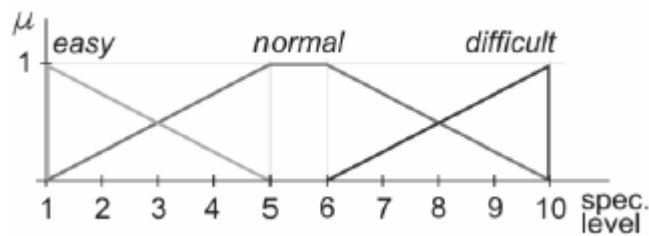


Figura 13 Le funzioni μ per la variabile linguistica "livello di difficoltà"

I valori saranno calcolati per mezzo delle seguenti equazioni:

$$\mu_{\text{easy}}(c) = \max [(5-l)/4, 0]$$

$$\mu_{\text{norm}}(c) = \min [(l-1)/4, (10-l)/4]$$

$$\mu_{\text{diff}}(c) = \max [(l-6)/4, 0].$$

Se il livello iniziale non è specificato allora verrà assegnato il valore di default $(0,1,0)$. Dopo di che i valori delle funzioni μ saranno modificati a seconda dei voti che gli altri studenti hanno ottenuto alla prova di valutazione finale. Se i voti sono alti, la competenza è considerata facile. Nel caso di voti bassi, la competenza risulta difficile. Per ciascun voto si applica quanto segue:

- Per voto 9 oppure 10 si diminuisce il grado di difficoltà e si applicano le seguenti equazioni
- $\mu_{\text{easy}}(c) = \mu_{\text{easy}}(c) + \mu_{\text{norm}}(c) * r$
- $\mu_{\text{norm}}(c) = \mu_{\text{norm}}(c) - \mu_{\text{norm}}(c) * r + \mu_{\text{diff}}(c) * r$
- $\mu_{\text{diff}}(c) = \mu_{\text{diff}}(c) - \mu_{\text{diff}}(c) * r$

- Per voto 7 oppure 8 si incrementa mediamente la difficoltà e si applicano le seguenti equazioni

$$\mu_{\text{easy}}(c) = \mu_{\text{easy}}(c) - \mu_{\text{easy}}(c) * r$$

$$\mu_{\text{norm}}(c) = \mu_{\text{norm}}(c) + \mu_{\text{easy}}(c) * r + \mu_{\text{diff}}(c) * r$$

$$\mu_{\text{diff}}(c) = \mu_{\text{diff}}(c) - \mu_{\text{diff}}(c) * r$$

- Per voto 6 oppure negativo si incrementa la difficoltà e si applicano le seguenti equazioni

$$\mu_{\text{easy}}(c) = \mu_{\text{easy}}(c) - \mu_{\text{easy}}(c) * r$$

$$\mu_{\text{norm}}(c) = \mu_{\text{norm}}(c) - \mu_{\text{norm}}(c) * r + \mu_{\text{easy}}(c) * r$$

$$\mu_{\text{diff}}(c) = \mu_{\text{diff}}(c) + \mu_{\text{norm}}(c) * r$$

La costante r nelle equazioni controlla l'andamento dei cambiamenti.

Il valore proposto di r è:

$$r = \min [1/n, 1/10]$$

dove n è il numero totale di tutti i voti (positivo o negativo) ricevuti dalla competenza c .

3.3 I voti

Quando gli studenti portano a termine la prova finale di accertamento, ottengono un voto che varia da 1 a 10. I voti, per la variabile linguistica *voti* (marks), possono assumere i seguenti valori: *negativo*, *positivo* oppure *nessun voto*. La variabile è pienamente descritta per mezzo della terna $(\mu_{\text{no}}, \mu_{\text{neg}}, \mu_{\text{pos}})$. Per un dato mark m , i valori delle funzioni μ possono essere calcolati usando le seguenti equazioni:

$$\mu_{\text{no}}(c) = \max [1-m, 0]$$

$$\mu_{\text{neg}}(c) = \max [\min [1-m, 6-m], 0]$$

$$\mu_{\text{pos}}(c) = \min [\max [m-5, 0], 1].$$

3.4 I requisiti iniziali (pre-requisiti) della conoscenza

Qualsiasi competenza ha dei pre-requisiti (nel grafo delle dipendenze), il cui livello di conoscenza è calcolato, ed il loro grado d'apprendimento è descritto da cinque termini linguistici: *not*, *little*, *enough*, *well*, e *very well*. La stima del grado di conoscenza dei pre-requisiti relativi ad una specifica conoscenza si basa sui voti ottenuti dallo studente alle prove finali d'accertamento. Se una competenza non ha alcun pre.requisito, la si tratta come se tutti i suoi pre-requisiti fossero conosciuti *very well*.

3.5 La defuzzificazione

Alla fine, i valori linguistici della variabile linguistica *livello di importanza* sono defuzzificati in valori crisp (netti). Sarà usato il metodo Centre Of Maximum (CoM), che inizialmente determina i principali valori tipici per ogni termine linguistico, per poi computarne il valore crisp come il miglior compromesso tra le due logiche.

4.0 Conclusioni

In questo capitolo si è descritto un modo per introdurre il supporto della navigazione nella piattaforma di e-learning InterMediActor. Si è proposto un modello studente fuzzy, che è stato sviluppato e testato nella simulazione dei dati utente.

Riferimenti

Pedraza-Jimenez R., Valverde-Albacete F.J., Cid-Sueiro J., Molina-Bulla H., Navia- Vazquez A. (2004). "Fuzzy Student Model in InterMediActor Platform" <http://lgm.fri.uni-lj.si/~alenka/papers/Iti2004.pdf>

Pedraza-Jimenez R., Valverde-Albacete F.J., Cid-Sueiro J., Molina-Bulla H., Navia- Vazquez A. (2002) '*Hierarchy-Based Methodology for Producing Educational Contents with Maximal Reutilization*'. In: Proceedings of the Learning' 02 Conference; Oct 2002; Madrid, Spain.

Valverde-Albacete F.J., Cid-Sueiro J., Diaz- Perez P., Pedraza-Jimenez R., Molina-Bulla H., Navia-Vazquez A. (2003) '*InterMediActor: an Environment for Instructional Content Design Based on Competences*'. In: Proceedings of the 2003 ACM Symposium on Applied Computing SAC'03; Mar 2003; Melbourne, USA; 2003.

CAPITOLO VI

Il Sistema INSPIRE

3.0 Introduzione

In questo capitolo sarà presentato un prototipo di ipermedia adattivo di tipo educativo chiamato INSPIRE (Intelligent System for Personalized Instruction in a Remote Enviroment).

Il sistema è stato realizzato presso il dipartimento di informatica e telecomunicazione dell'università di Atene, in collaborazione con il

dipartimento di scienze dell'informazione della Brunel University di Uxbridge (UK).

Esso enfatizza il fatto che gli allievi percepiscano ed elaborano i dati in modi diversi e di conseguenza, attraverso l'interazione con gli studenti, genera dinamicamente le lezioni, con l'obiettivo di condurre lo studente alla conoscenza di un determinato argomento.

I sistemi ipermediali adattivi ed educativi (Adaptive Education Hypermedia, AEH), [Brusilovsky, 2001], rappresentano un risultato recente della ricerca nel campo dei sistemi di apprendimento. In più, questi sistemi sono capaci di combinare insieme i due "opposti" approcci per ciò che riguarda i sistemi d'insegnamento via computer. In altre parole, possono essere considerati come l'unione strutturata, dei sistemi tradizionali di tutoring intelligente con quelli più flessibili di tipo ipermediale.

L'adattamento in questo contesto, è definito come la possibilità di ricostruire l'ambiente educativo al fine di rimodellarlo a seconda delle abilità e dei bisogni degli allievi. Si possono distinguere diversi livelli di adattamento, a seconda di chi (sistema o studente) modifichi l'ambiente educativo ai bisogni dello studente. In altri termini, queste due forme di adattamento possono essere classificate come:

- Comportamento adattivo, se il sistema adatta i suoi output a seconda dei dati in suo possesso sulla conoscenza corrente acquisita dallo studente.
- Comportamento adattabile se il sistema supporta i mezzi che consentano allo studente, con l'uso di appropriate funzioni, di modificare l'ambiente d'apprendimento.

Sia l'adattamento che il comportamento adattabile di INSPIRE sono gestiti all'interno del modulo studente. In esso sono memorizzate tutte le informazioni relative agli studenti, come, ad esempio, il livello di conoscenza di un determinato dominio, oppure lo stile d'apprendimento dello studente. Il modello studente è utilizzato in tanti modi: per creare una

successione di curriculum script (cioè una successione di tutti i problemi mostrati allo studente con le relative risposte), un supporto adattivo alla navigazione, una presentazione adattiva oppure per fornire al sistema mezzi che possano consentirgli un comportamento adattabile.

Inizieremo con il dare uno sguardo all'architettura del sistema INSPIRE, per poi descrivere nei paragrafi terzo e quarto il processo di diagnosi dello studente (il processo di deduzione delle caratteristiche intellettive degli studenti per mezzo dei loro comportamenti visibili) e le principali tecniche (fondate sulla logica fuzzy) capaci di implementarlo.

2.0 L'architettura di INSPIRE

Il sistema INSPIRE [Papanikolaou & altri, 2001] è un web-based AEHS (Adaptive Educational Hypermedia System), costruito con l'obiettivo di fornire un'assistenza a distanza agli studenti durante i loro studi.

Il sistema costruisce e presenta una serie di lezioni, secondo precise direttive d'insegnamento, cercando di adattarsi al livello di conoscenza dello studente e al suo stile d'apprendimento.

Questo processo di personalizzazione del contenuto, richiede una completa conoscenza del livello (grado) d'apprendimento dello studente. Questo tipo di conoscenza viene acquisita dal sistema per mezzo di una serie di prove-tests, cui sottopone lo studente, su argomenti inerenti il dominio di studio. Dopo di che, in base ai risultati ottenuti dallo studente nei tests, INSPIRE produce delle stime sul livello di conoscenza raggiunto, usando il processo di diagnosi dello studente. Perciò, la diagnosi delle caratteristiche dello studente può essere vista come quel processo di deduzione del livello di conoscenza raggiunto su ciascun argomento del dominio, a partire dalle risposte fornite all'interno delle prove (comportamento osservabile).

INSPIRE, è composto da quattro moduli, più un dispositivo di memorizzazione dei dati (vedi figura 14):

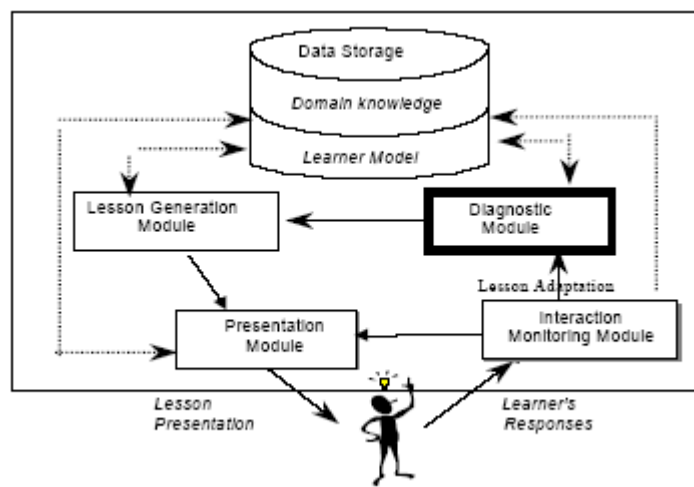


Figura 14 La struttura del sistema INSPIRE

- Il modulo che cattura e manipola l'interazione dello studente con il sistema (IMM)
- Il modulo che effettua la diagnosi dello studente, elabora i dati memorizzati e decide come classificare la conoscenza raggiunta dallo studente ed il suo stile d'apprendimento (SDM)
- Il modulo di generazione delle lezioni personalizzate, in base alla conoscenza corrente dello studente (LGM)
- Il modulo di presentazione del contenuto (argomento), la cui funzione è quella di inviare allo studente le lezioni create dal LGM.

Il dispositivo di memorizzazione dei dati contiene il dominio della conoscenza ed il modello studente, che contiene tutte le informazioni che il sistema ha raggruppato sullo studente. Questa informazione, include il numero e il tipo di domande cui lo studente ha tentato di dare una risposta, i tentativi fatti per rispondere a ciascuna domanda, il tempo impiegato dallo studente a rispondere su ognuna, la percentuale di tempo dedicato allo

studio (tempo dedicato agli esempi, alla simulazione e così via) ed altre quantità simili e misurabili.

Questo capitolo sarà dedicato, principalmente, sul modulo diagnostico, capace di fornire una stima del livello di conoscenza dello studente nell'ambito del dominio di interesse. Il modulo LGM userà poi questa informazione, per generare il "contenuto" personalizzato, che verrà poi assegnato allo studente.

3.0 Il problema del processo di diagnosi dello studente

La presenza dell'incertezza, è un fattore importante che spesso può condurre all'errore quando si analizza il comportamento dello studente. Essa sembra dovuta, in parte agli errori e alle approssimazioni che si presentano allorquando si cerca di stimare i dati dell'interazione studente-sistema, ed in parte alla natura astratta del pensiero umano. La perdita di informazione prodotta da questa incertezza, conduce ad una perdita dell'adattamento del sistema. In altri termini, quando si tenta di esplicitare i processi mentali (ma anche gli stati) ed emotivi, si deve tener conto di uno strato di approssimazione. Quindi, tenendo conto di tutti i problemi menzionati, sembra ovvio che lo sviluppo di un buon metodo di diagnosi dello studente è fondato su di una corretta gestione dell'incertezza.

Comunque, l'obiettivo è quello di creare un processo diagnostico simile al modo con cui un tutor umano accerta ed esplicita, mediante l'uso dei voti, il livello di conoscenza dello studente su di uno specifico argomento. Nel nostro caso, in particolare, il processo diagnostico è fondato sulla valutazione della conoscenza in base a specifici criteri, definiti in termini di obiettivi e competenze (criterion-referenced assessment). In questo modo, il sistema educativo è fornito continuamente di informazioni relative alle performance degli studenti, al fine di essere capace di adattare, di conseguenza, i suoi output.

Come metodo per gestire l'incertezza, delle risposte fornite dallo studente, ed incorporare l'abilità e la flessibilità dei docenti umani, sarà usata la logica fuzzy combinata ad un approccio decisionale che prende decisioni in base a diversi criteri (che in seguito vedremo coincidere con tre fissati livelli di abilità).

4.0 Il processo di diagnosi dello studente in INSPIRE

Le lezioni, in INSPIRE; sono generate in modo da condurre lo studente alla conoscenza di una specifica materia (obiettivo d'apprendimento). Ciascuno di questi obiettivi, è associato ad un sottoinsieme di argomenti. Per cui, se lo studente vuole raggiungere l'obiettivo finale, non può prescindere dalla conoscenza di questo sottoinsieme di argomenti. Conoscenza, che viene accertata dal sistema mediante l'uso di prove-tests, fornite allo studente durante la fase di studio dell'argomento.

I tests sono suddivisi in diverse categorie ciascuna delle quali corrisponde ad una specifica abilità. Le domande, cui i tests fanno riferimento, possono essere di tre tipi:

- Tipo remember (Remember level): sono quelle domande che verificano l'abilità degli studenti a ricordare precedenti informazioni
- Tipo use (Use level): sono quei quesiti che testano l'abilità degli studenti ad applicare le nozioni ricevute a singoli casi
- Tipo find (Find level): sono quelle domande che verificano la capacità degli studenti a proporre ed a risolvere problemi pratici.

Quando lo studente seleziona un test di verifica, le domande gli appariranno una dopo l'altra con difficoltà crescente. In altre parole, i quesiti più facili del livello Remember si presentano per primi, poi appariranno quelli di tipo Use ed infine quelli di tipo Find.

In qualsiasi istante della fase di verifica, lo studente può interrompere la prova. In base alle risposte fornite, il sistema proporrà una stima del livello di conoscenza raggiunto. Lo scopo del sistema è di produrre una stima precisa come quella che effettua l'insegnante umano. Per poter far questo, la conoscenza relativa ad un argomento sarà descritta in funzione di quattro livelli di abilità: {Insufficiente, Piuttosto Insufficiente, Piuttosto Sufficiente, Sufficiente}. In definitiva, l'obiettivo del processo di diagnosi è di ottenere informazioni sulla conoscenza dello studente su ciascun argomento in funzione delle quattro caratteristiche precedenti. Per poter far questo, dobbiamo creare un modello della conoscenza esperta e modellare il processo di inferenza usato da un esperto (insegnamento esperto) nella gestione della valutazione degli studenti.

4.1 La creazione del modello della conoscenza esperta

Sono considerate risorse preziose, nella fase di costruzione del modello di insegnamento esperto in quella dell'accertamento della conoscenza degli studenti e nella costruzione di un insegnamento personalizzato:

- a) I criteri che l'insegnante definisce per accertare il livello di conoscenza dello studente.
- b) Le stime degli insegnanti sull'importanza dei differenti tipi di quesiti di accertamento (che corrispondono ai criteri visti prima), rispetto al livello di conoscenza dello studente, al tempo impiegato e dal tipo di argomento preso in considerazione.
- c) Le stime degli insegnanti sulla relazione tra le corrette risposte date dagli studenti e la loro abilità dell'argomento.

La definizione dei criteri. Il processo di accertamento della conoscenza dello studente è facilitato dall'introduzione di particolari criteri, formulati in termini di obiettivi e competenze, le quali stabiliscono cosa lo

studente debba portare a termine per avere una competenza piena dell'argomento. Nel nostro approccio, il processo di analisi del livello di conoscenza usa tre criteri qualitativi corrispondenti a tre tipi di abilità: Remember, Use e Find (visti in precedenza).

L'importanza dei diversi tipi di domande. Come si è detto in precedenza, i tests di accertamento della conoscenza consistono, in INSPIRE, di domande suddivise in tre differenti categorie (Remember, Use, Find). L'importanza che possono assumere le domande riferite ad una categoria, piuttosto che ad un'altra, dipende dall'argomento che si vuole verificare e dalla quantità di tempo che lo studente impiega a risolverle.

Per esempio, per quanto riguarda l'argomento "La funzione della memoria cache": la conoscenza teorica (Remember) è più importante rispetto alle altre. Mentre per l'argomento "Le tecniche di mapping" è più importante che lo studente sia capace di risolvere problemi pratici (Use). L'importanza che assume ciascuna domanda a seconda della sua appartenenza ad una delle tre categorie, è uno degli aspetti della conoscenza del docente, che dovrebbe essere rappresentata (modellata) durante la fase di analisi delle capacità dello studente. Al fine di trasportare questo tipo di conoscenza dall'insegnante al computer, useremo il metodo AHP (Analytical Hierarchy Process) di Saaty [Saaty,1980] per assegnare dei pesi, per ciascuna domanda, ai tre tipi di abilità (Remember, Use Find).

I pesi associati a ciascuna domanda, su cui il sistema si basa per valutare il livello di conoscenza dello studente variano a seconda

- a) Del grado di conoscenza che ha lo studente nel momento in cui chiede di essere esaminato. Per esempio, quando lo studente si trova ad imparare un nuovo argomento, i pesi sono specificati in modo che le domande che valutano la comprensione teorica (1° criterio) hanno un peso maggiore rispetto a quelle che accertano la conoscenza applicativa (2° criterio). In altre parole, si assume che lo studente debba

inizialmente studiare la parte teorica (Remember level), per poi continuare con l'applicazione della teoria. Infatti, con il progredire dell'insegnamento, mentre il suo livello di conoscenza cambia da {Insufficiente} a {Piuttosto Insufficiente} (ha finito la parte teorica e dovrebbe cominciare quella applicativa), i pesi relativi ai livelli di abilità cambiano, ed il peso delle domande del livello (Use) verranno incrementati. Dopo questo cambiamento, al fine di raggiungere il livello {Piuttosto Sufficiente} di conoscenza, lo studente dovrebbe rispondere alle domande di tipo applicativo e così via.

- b) Del tipo di argomento preso in considerazione. Per esempio, se un argomento è di tipo procedurale, allora lo studente dovrebbe imparare principalmente come applicarla ed in quali casi. Quindi, in questo caso, il peso relativo alla conoscenza applicativa sarà più grande di quello associato alla conoscenza teorica.

La relazione tra le risposte corrette ed il grado di abilità. Nel sistema INSPIRE, si tenta di simulare il comportamento di un insegnante, che formuli domande allo studente su di uno specifico argomento e ne calcoli il livello di preparazione (conoscenza) raggiunto. Per far questo, il sistema usa la teoria degli insiemi fuzzy, che hanno il pregio di poter manipolare le caratteristiche qualitative (il livello di conoscenza) della conoscenza dello studente.

Con il termine insieme fuzzy, qui intenderemo una funzione del tipo $f : U \rightarrow [0,1]$, dove U è l'insieme di definizione della funzione. Il valore $f(x)$ della funzione per un input x , rappresenta il grado di appartenenza di x all'insieme fuzzy. Per esempio, supponiamo d'avere l'insieme fuzzy "Conoscenza insufficiente relativa al livello Remember". Supponiamo ancora che i valori della funzione f siano pari a $f(x) = \{1, 0.6, 0.3, 0.1, 0, 0, 0,$

$0, 0, 0\}$, con $x \in \{0, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$ che mi indica la percentuale delle domande riferite al livello Remember cui lo studente ha risposto correttamente. Allora, se per $x=10$ il valore della f è $f(10)=0.6$, significa che la conoscenza dello studente per quanto riguarda le domande del livello Remember (le cui risposte corrette sono pari sono solo al 10% del totale) è da considerarsi “Insufficiente”. Infatti il grado di appartenenza risulta essere pari a 0.6.

I gradi di appartenenza possono essere formulati dall’insegnante ponendo delle domande del tipo “Quanto è considerato insufficiente il livello di preparazione dello studente, limitatamente al livello Remember, se lo studente ha risposto correttamente per il 10% delle domande?”.

In particolare notiamo che l’insegnante deve fornirci di 12 insiemi fuzzy. Infatti, un insieme fuzzy è richiesto per ciascuno dei tre criteri {Remember, Use, Find} e per ognuno dei 4 livelli di abilità {Insufficiente, Piuttosto Insufficiente, Piuttosto Sufficiente, Sufficiente}. Così per esempio, avremo degli insiemi fuzzy che descrivono “Insufficiente Conoscenza del livello Remember”, “Piuttosto Insufficiente conoscenza del livello Remember”, “Sufficiente Conoscenza del livello Use” etc. Chiameremo questi insiemi fuzzy $f_{L,P}$ dove $L \in \{R, U, F\}$ e $P \in \{I, PI, PS, S\}$. L’insieme fuzzy $f_{L,P}$ rappresenterà il grado di abilità (competenza) pari a “P” per quanto riguarda il livello L del tipo di conoscenza. In altre parole la f mi descriverà il grado di preparazione associato ad un particolare tipo di conoscenza.

4.2 Il processo di diagnosi

Il processo di diagnosi inizia dopo che lo studente ha risposto alle domande del test di verifica. La diagnosi mira a stimare il livello di conoscenza dello studente su di un argomento specifico.

Innanzitutto, per ogni categoria (R,U,F), cerchiamo di dividere il numero delle risposte esatte dal numero totale di domande, al fine di

calcolare la percentuale di risposte esatte, relativa alla categoria in esame. Dopo di che, il valore trovato deve essere arrotondato alla decina intera più vicina, poiché gli insiemi fuzzy hanno 10 elementi con valori {0, 10, 20, 30, 40, 50, 60, 70, 80, 90}. Chiameremo in seguito x_r , x_u , x_f le percentuali di risposte esatte relative a ciascuna delle tre categorie Remember, Use, Find. Poi, usando questi tre valori e i 12 valori fuzzy, che l'insegnante esperto ha precedentemente specificato, costruiamo la matrice D, che contiene i vari gradi (di appartenenza) del livello di conoscenza dello studente per ciascuno dei 12 insiemi fuzzy.

$$D = \begin{bmatrix} f_R^I(x_R) & f_R^{RI}(x_R) & f_R^{RS}(x_R) & f_R^S(x_R) \\ f_U^I(x_U) & f_U^{RI}(x_U) & f_U^{RS}(x_U) & f_U^S(x_U) \\ f_F^I(x_F) & f_F^{RI}(x_F) & f_F^{RS}(x_F) & f_F^S(x_F) \end{bmatrix}$$

Nel momento in cui lo studente chiede di essere esaminato, verrà selezionato il vettore $W=[w_r, w_u, w_f]$. Dove w_r è il peso dei quesiti che accertano la conoscenza del 1° livello (Remember), w_u delle domande che accertano il 2° livello (Use) e w_f dei quesiti che accertano la conoscenza del 3° livello (Find). Il valore di ciascun w_i , come detto in precedenza, viene assegnato prima che inizi il test da un docente esperto.

Successivamente, moltiplicando $W*D$, calcoliamo il vettore P ($P=W*D$), che risulta essere il grado di appartenenza della conoscenza dello studente relativamente ai 4 gradi di abilità (I, PI, PS, S). In altre parole, il vettore $P=[P_1, P_2, P_3, P_4]$, dove P_1 è il livello {Insufficiente} della conoscenza dello studente, P_2 è il grado {Piuttosto Insufficiente} della sua conoscenza, e così di seguito. In altre parole, il vettore P conterrà una stima del livello di conoscenza raggiunto rispetto a ciascuno dei 4 livelli {I, PI, PS, S}.

Infine, così come si è calcolato il vettore P, è possibile dare una stima finale sul livello di conoscenza, su di un argomento specifico, dello studente. Quindi, per raggiungere un risultato definitivo, si deve tentare di

combinare i 4 valori del vettore P in un unico valore, al fine di avere un valore unico del livello di conoscenza dello studente. Per poter far questo si userà il metodo del Centro di Gravità [Lin & Lee, 1996], che produrrà un valore v tale che:

$$v = \frac{\sum_{i=1}^4 p_i i}{\sum_{i=1}^4 p_i}$$

Il valore di quest'espressione è arrotondato al valore intero più vicino. Così alla fine, se abbiamo $\text{round}(v)=1$, stimiamo la conoscenza, di un argomento specifico, dello studente essere pari a "Insufficiente". Se invece $\text{round}(v)=2$, allora la conoscenza sarà Piuttosto Insufficiente, e così di seguito.

5.0 Conclusioni

In questo capitolo abbiamo trattato del sistema adattivo ipermediale ed educativo INSPIRE. Abbiamo mostrato la sua architettura e ci siamo soffermati, in particolare, sul processo di diagnosi della conoscenza dello studente. abbiamo mostrato, poi, come questo processo sia la chiave di volta per una corretta elaborazione del fase di valutazione cui deve essere sottoposto lo studente.

Riferimenti

Brusilovsky, P.: (2001), '*Adaptive Hypermedia*'. In User Modeling and User-Adapted Interaction (UMAI) **11** (1/2), 111-127.

Lin, C.T., Lee, C.S.G. (1996): *Neural Fuzzy Systems: "A Neuro-Fuzzy Synergism to Intelligent Systems"*. Prentice Hall P T R Upper Saddle River, New Jersey

Papanikolaou, K., Grigoriadou, M., Kornilakis, H., Magoulas, G.: (2001) '*INSPIRE: An Intelligent System for Personalized Instruction in a Remote Environment*'. In: Reich, S., Tzagarakis, M.M., De Bra, P. M.E. (eds.): "Hypermedia: Openness, Structural Awareness and Adaptivity". Lecture Notes in Computer Science Vol. 2266. Springer-Verlag, Berlin.

http://hermes.di.uoa.gr/lab/CVs/papers%5Cpapanikolaou%5Cgkpm_setn02.pdf

Saaty T.: (1980), '*The Analytic Hierarchy Process*'. McGraw-Hill, New York.

CAPITOLO VII

I Sistemi Witness e Sherlock II

4.0 Introduzione

In questo capitolo tratteremo di due sistemi tutoriali ed intelligenti: il sistema WITNeSS [Negoita & Pritchard; 2004, 2003a/b] e quello SHERLOCK II.

Il primo è un sistema intelligente ed ibrido, che usa tecniche fuzzy-neurali per ottimizzare la presentazione del materiale d'apprendimento da fornire allo studente. Il modello studente, costruito facendo uso delle tecnologie fuzzy, valido per qualsiasi tipo di tutoring intelligente, fornisce un'ottima strategia d'apprendimento adattando il sistema ai bisogni "unici" di ciascun studente.

Il secondo è un ambiente pratico d'insegnamento, sviluppato per addestrare i tecnici dell'aviazione alla diagnosi dei guasti che si verificano in sistema elettronico complesso [Katz & altri, 1993].

2.0 Il sistema Witness

Il sistema, al fine di adattarsi alle necessità degli studenti, deve svolgere i seguenti compiti:

- Presentare agli studenti qualsiasi contenuto o abilità che desiderano apprendere, tale da soddisfare il loro modo di imparare e le loro caratteristiche psicologiche. Questo compito cruciale viene portato a compimento mediante tecniche di ricerca e sviluppo sull'apprendimento e sfruttando le caratteristiche psicologiche del singolo studente.
- Suggerire allo studente, quando richiesto, come apprendere al meglio il contenuto o le abilità.
- Lavorare con lo studente al monitoraggio del processo d'apprendimento. Il monitoraggio di un programma d'apprendimento deve tenere in conto le attività di altri studenti e tutors. Devono essere sviluppati sistemi ibridi differenti a supporto dell'integrazione delle attività dello studente con quelle di altri studenti. Questa tecnologia multi-struttura di intelligenza ibrida deve rilevare eventuali conflitti e fornire suggerimenti agli studenti e ai tutors in base alle loro attività e alle loro esigenze
- Fornire un'analisi intelligente ed interattiva su quel che sta elaborando lo studente e un aiuto diagnostico in tempo reale. L'aiuto potrà provenire direttamente dal docente o da altre fonti (anche on-line) o addirittura anche da altri studenti.

2.1 Le componenti del sistema.

WITNeSS (Wellington Institute of Technology Neural Expert Student Support) è stato pensato come un sistema neuro-fuzzy di soft computing per l'ottimizzazione del materiale d'apprendimento da fornire

allo studente. Le sue componenti funzionali sono sostanzialmente (vedi figura 15):

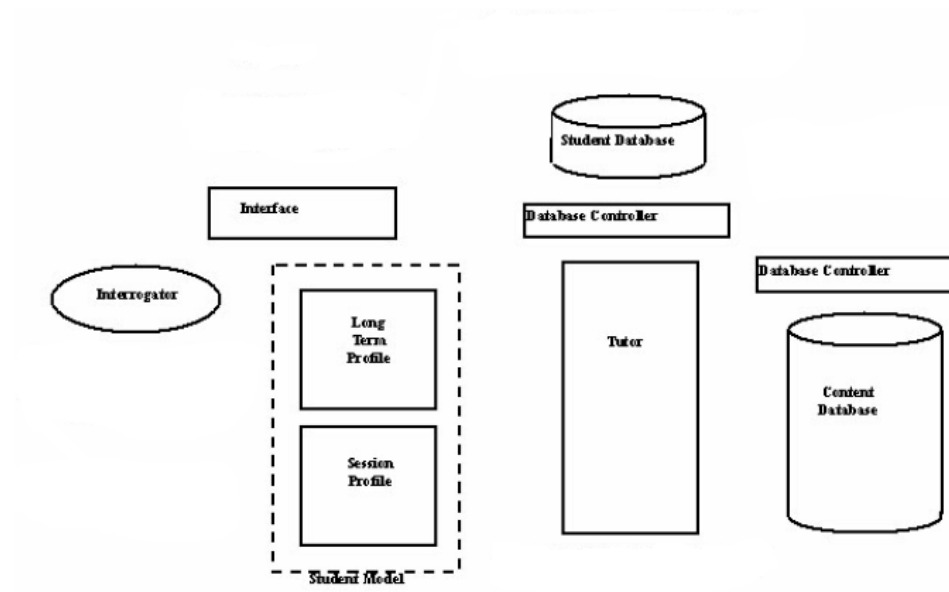


Figura 15 Le componenti funzionali del sistema WITNeSS

- Un'interfaccia utente, che fornisce allo studente un ambiente interattivo di comunicazione con il sistema Witness.
- Un database studente;
- Un modulo tutor (pedagogico);
- Un database dei contenuti (della conoscenza)
- Un modello studente diviso in due profili: quello della sessione corrente e quello a lungo termine che contiene il suo background d'apprendimento all'interno del sistema.
- Un modulo "interrogatore" in grado di interfacciare il comportamento dello studente con il modello studente.

3.0 Le relazioni tra le diverse componenti del sistema

La figura 16 mostra le relazioni tra i moduli del sistema. I numeri tra parentesi sulle frecce, si riferiscono ai corrispondenti box descrittivi.

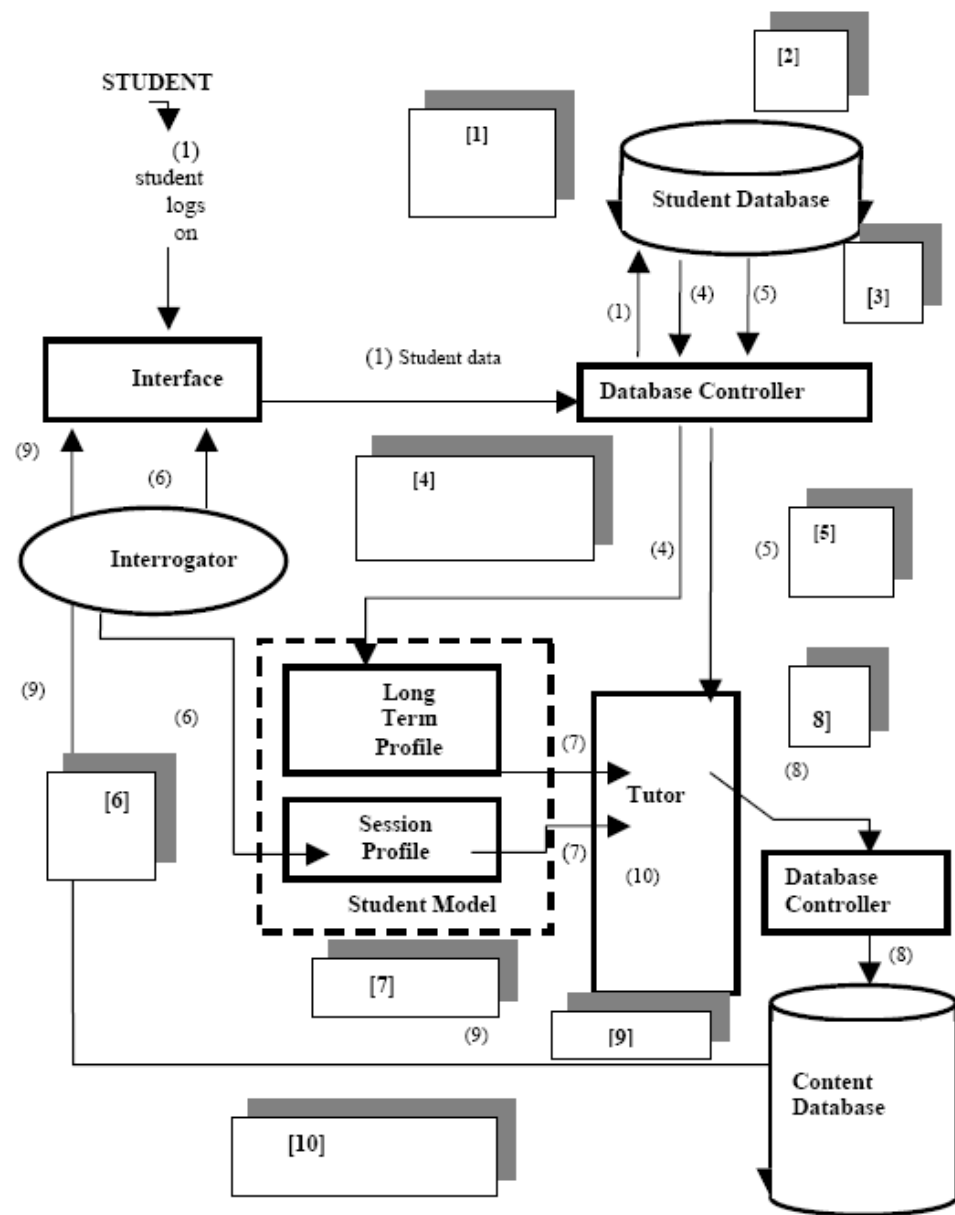


Figura 16 Le Relazioni tra le componenti di WITNeSS

- 1) Quando lo studente effettua l'operazione di log on tutta l'informazione (il profilo a lungo termine, il profilo della sessione corrente, le regole decisionali fuzzy) viene caricata all'interno del database studente.
- 2) Sebbene lo studente inizi con un profilo standardizzato, con il progredire della sua interazione, il sistema, usando le

tecnologie intelligenti ed ibride, sarà capace di raffinare i profili iniziali.

- 3) In altri termini, ciò che accade è che il sistema apprenderà come fornire il miglior insegnamento ad un particolare studente, che terminerà la sessione d'apprendimento con un proprio profilo individualizzato
- 4) L'informazione corrente, sullo studente, più il proprio profilo a lungo termine è caricato nel modulo studente.
- 5) Il profilo studente, realizzato con regole fuzzy, usato per la selezione dei contenuti viene memorizzato nel modulo pedagogico.
- 6) Il modulo interrogatore inizia a raccogliere, tramite l'interfaccia utente, l'informazione sul comportamento corrente dello studente. Questa viene poi inserita nel modello studente all'interno del profilo corrente.
- 7) L'informazione corrente sul comportamento dello studente, il suo profilo storico, i dati della sessione, sono inseriti modulo pedagogico.
- 8) Usando un insieme di regole decisionali di tipo fuzzy, il modulo pedagogico individua quali contenuti presentare, successivamente, allo studente.
- 9) Questi contenuti sono ordinati dal controller del database per poi essere inviati all'interfaccia studente.
- 10) Quando i dati finali relativi ad una sessione d'apprendimento giungono al modulo tutor (tramite l'interfaccia passando attraverso l'interrogatore e il modello studente) sarà utilizzata la componente fuzzy-neurale, presente all'interno del modulo, al fine di mettere a punto un insieme di regole fuzzy che possano descrivere nel modo migliore il livello

d'apprendimento dello studente. Il processo poi riprende dal passo (8) fino al successivo (10).

4.0 Il Sistema Sherlock II

Sherlock II fornisce un ambiente d'apprendimento consistente di una serie di esercizi di difficoltà sempre crescente. Ciascuna prova è suddivisa in due parti: la soluzione del problema e la sua versione corretta. Nel senso che durante la fase di soluzione del problema lo studente esegue, come mezzo d'aiuto, delle procedure per la scoperta degli errori.

Qui il modello studente è composto da una rete gerarchica di variabili fuzzy. Dove ciascun nodo, della struttura, è un indicatore di alcune capacità dello studente. In particolare, le variabili al livello inferiore indicano diversi tipi di conoscenza:

- Conoscenza di tipo concettuale (cioè il sapere come funzione un particolare attrezzo)
- Abilità (cioè la capacità di usare un determinato attrezzo come per esempio un oscilloscopio)
- Disposizioni comportamentali (cioè la tendenza a sostituire i componenti difettosi, piuttosto che ripararli)
- Fattori affettivi (ad esempio l'autostima).

Le variabili al livello più alto, della struttura, rappresentano una sorta di valutazione di tipo fuzzy su gruppi di quelle al livello inferiore (l'abilità di usare strumenti diagnostici, l'abilità di usare un oscilloscopio, l'abilità di usare un tester per la misurazione della tensione e così via).

Possono assumere cinque valori fuzzy:

- Nessuna conoscenza
- Conoscenza limitata
- Conoscenza non automatizzata
- Conoscenza parzialmente automatizzata

- Conoscenza pienamente sviluppata

Esse assumono un valore tra quelli descritti sopra a seconda dell'azioni svolte dallo studente in una fase determinata del processo d'apprendimento.

Il sistema, quindi, usa variabili differenti per rappresentare capacità distinte. Utilizza la logica fuzzy per rappresentare ed aggiornare l'imprecisione presente nei modelli studente.

Riferimenti

Katz, S., Lesgold, A., Eggan, G., Gordin, M.,(1993) '*Modeling the Student in Sherlock II*'. Journal of Artificial Intelligence and Education (Special Issue on Student Modeling), vol.3 pp. 496-518.

Negoita, M. Gh. and Pritchard, D. (2004) “ *Using a virtual student model for testing intelligent tutoring systems* “. Interactive Technology & Smart Education. http://www.troubador.co.uk/itse/journal_info.asp.

Negoita, M. Gh. and Pritchard, D. (2003a) '*Testing Intelligent Tutoring Systems by Virtual Students*'. Proceedings of International Conference on Machine-Learning and Applications (ICMLA '03), Los Angeles USA, pp 98–104.

Negoita, M. Gh. and Pritchard, D. (2003b) '*Some Test Problems Regarding Intelligent Tutoring Systems*'. Knowledge-Based Intelligent Information and Engineering Systems, V. Palade, J. R. Howlett, L. Jain (eds.). Springer-Verlag, Part II, pp 986–992

Bibliografia Generale

Anderson J.R. (1998), in Alastair A., 'A Model for Fuzzy Personal Construct Psychology, in Fuzzy Systems Design (Studies in Fuzziness and Soft Computing)', ed. By Reznik L., Dimitrov V. and Kacprzyk J., Springer-Verlag, pp 153-170.

Bennett, F. 1997. 'Computers as Tutors: Solving the Crisis in Education'. First Monday. <http://www.firstmonday.dk/issues /issue6/intro/index.html>.

Brusilovskiy, P.L. (1994). 'The Construction and Application of Student Models in Intelligent Tutoring Systems'. Journal of Computer and Systems Sciences International 32 (1), pp.70-89.

Brusilovsky, P.: (1996), 'Methods and Techniques of Adaptive Hypermedia'. User Modeling and User-Adapted Interaction (UMAI) 6 (2/3), 87-129.

Brusilovsky P., (1998) 'Methods and Techniques of Adaptive Hypermedia'. Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, pp. 1-43.

Brusilovsky, P.: (1999), 'Adaptive and Intelligent Technologies for Web-based Education'. In: C.Rollinger and C.Peylo (eds.): *Kunstliche Intelligenz*, Special Issue on Intelligent Systems and Teleteaching, pp. 19-25.

Brusilovsky P. (2000) Adaptive hypermedia: 'From intelligent tutoring systems to Web-based education. *Intelligent Tutoring Systems*'. Lecture

Notes in Computer Science, Vol.1839; Proceedings of 5th International Conference on Intelligent Tutoring Systems, ITS 2000; Jun 2000; Montreal, Canada. Berlin: Springer Verlag.

Brusilovsky, P.: (2001), '*Adaptive Hypermedia*'. In User Modeling and User-Adapted Interaction (UMAI) **11**: 87-110

Brusilovsky P., Eklund J., (1998) '*A Study of User Model Based Link Annotation in Educational Hypermedia*', Journal of Universal Computer Science, Volume 4, Issue 4 pp. 429-448.

Burns, H. L., and Capps, C. G. (1988). '*Foundations of Intelligent Tutoring Systems*'. In Polson, M. C., and Richardson, J. J., eds., '*Foundations of Intelligent Tutoring Systems*'. Lawrence Erlbaum Associates. capitoli 3, 55-78.

Capuano, N., Marsella, M., Salerno S. (2000) '*ABITS: An Agent Based Intelligent Tutoring System for Distance Learning*'. Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems. ITS.

Capuano, N., De Santo, M., Marsella, M., Molinara, M., Salerno S. (2001) '*Personalised Intelligent Training on the Web: A Multi Agent Approach. Electronic Business and Education, Recent Advances in Internet Infrastructures*'. Kluwer: Multimedia Systems And Applications Series, vol. 20, chap. 5.

Capuano, N., Gaeta, M., Micarelli, A., Sangineto, E. (2002) '*An Integrated Architecture for Automatic Course Generation*.' Proceedings of the IEEE

International Conference on Advanced Learning Technologies. ICALT, Kazan, Russia

Capuano, N., Gaeta, M., Micarelli (2003) “*IWT: Una Piattaforma Innovativa per la Didattica Intelligente su Web*”.

http://www.crimpa.it/intraserv/documents/AIXIA_2003.pdf

Carbonell, J.R. (1970). AI in CAI: ‘*An Artificial-Intelligence Approach to Computer-Assisted Instruction*’. IEEE Transactions on Man-Machine Systems MMS-11, December 1970, 4.

Clancey, W.J. (1986). ‘*Qualitative Student Models*’. Annual Review of Computer Science 1, pp.381-450.

Diogene (Sito web del progetto) “Diogene: a Training Web Broker for ICT Professionals” (IST-2001-33358): <http://www.diogene.org>.

Dubois, D., Prade, H. (1980) ‘*Fuzzy Sets and Systems – Theory and Applications*’. Academic Press.

Eklund J., Brusilovsky P., (1998) ‘*Individualising Interaction in Web-based Instructional Systems in Higher Education*’. AUC Academic Conference 98, University of Melbourne, Melbourne, Australia.

Eklund J., Zeiliger R., (1996) ‘*Navigating the Web: Possibilities and Practicalities for Adaptive Navigational Support*’. Proceedings of Ausweb’96, The Second Australian World Wide Web Conference, Southern Cross University Press.

Fensel, D., (2001) *'Ontologies: a Silver Bullet for Knowledge Management and Electronic Commerce'*. Springer.

Gürer DW, desJardins M, Schlager M. (1995) *'Representing a Student's Learning States and Transitions'*. AAAI Spring Symposium on Representing Mental States and Mechanisms. Stanford, CA, USA. AAAI technical report, SRI International.

Heflin, J., Hendler, J., (2000) *'Searching the Web with SHOE'*. Workshop on AI for Web Search. AAAI
<http://www.cs.umd.edu/projects/plus/SHOE/pubs/aiweb2000.pdf>.

InTraServ (sito web del progetto) *'InTraServ: Intelligent Training Service for Management Training in SMEs'* (IST- 2000-29377):
<http://www.intraserv.org>.

Katz, S., Lesgold, A., Eggan, G., Gordin, M.,(1992) *'Approaches to student modeling in the Sherlock Tutor'* Eds. Proceedings of the third International Workshop on User Modeling, pp.205-230. Dagstuhl Castle, Germany : International Conference and Research Center for Computer Science

Katz, S., Lesgold, A., Eggan, G., Gordin, M.,(1993) *'Modeling the Student in Sherlock II'*. Journal of Artificial Intelligence and Education (Special Issue on Student Modeling), vol.3 pp. 496-518.

Kavcic A.,(2001) *'Adaptation in Web-based Educational Hypermedia Considering the Uncertainty of User Knowledge'*. Ph.D. Dissertation, Faculty of Computer and Information Science, University of Ljubljana.

Kavcic A., Privosmik M., Marolt M., Divjak S., (2001) " *Educational Hypermedia System ALICE: an Evaluation of Adaptive Features.* "

<http://lgm.fri.uni-lj.si/~alenka/papers/Calie2001.pdf>

Kay, J.: 2001, ' *Learner control*'. (UMAI) User Modeling and User-Adapted Interaction **11**(1/2), 111-127.

Lee S., Wang C., (1997) " *Intelligent Hypermedia Learning System on the Distributed Environment.* " <http://media.inhatc.ac.kr/papers/edmedia97.pdf>

Lin, C.T., Lee, C.S.G. (1996) : " *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*". Prentice Hall P T R Upper Saddle River, New Jersey.

Lesgold, A., Katz, S., Greenberg, L., Hughes, E., Eggan, G., (1992) ' *Extensions of Intelligent Tutoring Paradigms To Support Collaborative Learning*'. Dijkstra, S., Kramer, H., and Van Merriënboer, J., Eds. Instructional Design Models in Computer Based Learning Enviroments. Berlin: Sprinter-Verlag, pp.291-311.

M-Learning (sito web del progetto) 'm-Learning: Mobile Communication Technologies for Young Adults Learning and Skills Development'. (IST-2000-25270): <http://www.m-learning.org>.

Merrill, M.D.: (1983), 'Component Display Theory'. In: C.M. Reigeluth (ed.): Instructional design theories and models: An overview of their current status. Lawrence Erlbaum Association, Hillsdale, New Jersey, pp. 279-333

McTaggart, J. (2001) ' *Intelligent Tutoring Systems and Education for the Future.* ' *512X Literature Review*, April 30 2001, pp. 2

Murray, D.: (1991), *'Modelling for Adaptivity'*. In: M.J.Tauber and D.Ackermann (eds.): *Mental Models and Human Computer Interaction* Vol. 2. Elsevier Science Publishers BV, Amsterdam, pp. 81-95.

Natabayashi K., Koike Y. and Maruyama M. (1995) *'An Intelligent Tutoring System on world wide web'* in *'An integrated learning Enviroment on a Distributed Hypermedia, Educational Multimedia and Hypermedia'*, Association for the Advancement of Computing in Education, Charottesville, VA pagg. 488-493

Negoita, M. Gh. and Pritchard, D. (2004) *" Using a virtual student model for testing intelligent tutoring systems "*. Interactive Technology & Smart Education. http://www.troubador.co.uk/itse/journal_info.asp.

Negoita, M. Gh. and Pritchard, D. (2003a) *'Testing Intelligent Tutoring Systems by Virtual Students'*. Proceedings of International Conference on Machine-Learning and Applications (ICMLA '03), Los Angeles USA, pp 98–104.

Negoita, M. Gh. and Pritchard, D. (2003b) *'Some Test Problems Regarding Intelligent Tutoring Systems'*. Knowledge-Based Intelligent Information and Engineering Systems, V. Palade, J. R. Howlett, L. Jain (eds.). Springer-Verlag, Part II, pp 986–992.

Omg (Object Managment Group, 1995) Corba 2.0 sito web di riferimento <http://www.omg.org/corba/iiop.html>

Papanikolaou, K., Grigoriadou, M., Kornilakis, H., Magoulas, G.: (2001) *'INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment'*. In: Reich, S., Tzagarakis, M.M., De Bra, P. M.E. (eds.):

“Hypermedia: Openness, Structural Awareness and Adaptivity”. Lecture Notes in Computer Science Vol. 2266. Springer-Verlag, Berlin.

Pedraza-Jimenez R., Valverde-Albacete F.J., Cid-Sueiro J., Molina-Bulla H., Navia- Vazquez A. (2002) *'Hierarchy-Based Methodology for Producing Educational Contents with Maximal Reutilization'*. In: Proceedings of the Learning' 02 Conference; Oct 2002; Madrid, Spain.

Pedraza-Jimenez R., Valverde-Albacete F.J., Cid-Sueiro J., Molina-Bulla H., Navia- Vazquez A. (2004). “Fuzzy Student Model in InterMediActor Platform“ <http://lgm.fri.uni-lj.si/~alenka/papers/Iti2004.pdf>

Saaty T.: (1980), *'The Analytic Hierarchy Process'*. McGraw-Hill, New York.

Self, J.A. (1974). *'Student Models in Computer-Aided Instruction.'* International Journal Man-Machine Studies 6, pp.261-276.

Self, J.A. (1988). *'Student Models: What use are they?'* in: Ercoli,P., Lewis,R., (Eds.), *'Artificial Intelligence Tools in Education'*. Elsevier Science Publishers B.V., North Holland, pp. 73-86.

Smyth, B. (2003)*'Intelligent Navigation on the Mobile Internet'*. Knowledge-Based Intelligent Information and Engineering Systems, V. Palade, J. R. Howlett, L. Jain (eds.). Springer-Verlag, Part I, pp 17 19.

Urban-Lurain, M. (1996). *'Intelligent tutoring systems, an historic review in the context of the development of artificial intelligence and educational psychology'*. Technical report, Department of Computer Science and

Engineering, MichiganState University. <http://aral.cse.msu.edu/Publications/ITS/its.htm>.

Valverde-Albacete F.J., Cid-Sueiro J., Diaz- Perez P., Pedraza-Jimenez R., Molina-Bulla H., Navia-Vazquez A. (2003) '*InterMediActor: an Environment for Instructional Content Design Based on Competences*'. In: Proceedings of the 2003 ACM Symposium on Applied Computing SAC'03; Mar 2003; Melbourne, USA; 2003.

VanLehn, K. (1988). '*Student Modeling*'. In Polson, M. C., and Richardson, J. J., eds., '*Foundations of Intelligent Tutoring Systems*'. Lawrence Erlbaum Associates. capitoli 3, pp. 55–78.

Wenger E., (1987) '*Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*'. Morgan Kaufmann Publishers, Inc., Los Altos.

Wu H., De Bra P., Aerts A., and G. Houben G.J.,(2000) '*Adaptation Control in Adaptive Hypermedia Systems, Adaptive Hypermedia and Adaptive Web-Based Systems*'. Proceedings of the International Conference AH, Trento, Italy, August 2000, Lecture Notes in Computer Science 1892

Yager, R. R.(2000). '*Target on E-commerce marketing using Fuzzy intelligent agents*'. IEEE Intelligent System.

Zadeh, L. A. (1996). '*Fuzzy logic = Computing with words*', IEEE Transactions on Fuzzy System, 4(2): 103-111.

Parte A

Per chi volesse, in futuro, approfondire gli argomenti trattati, segnalo i seguenti articoli facilmente reperibili sul Web:

Rothenhöfer D., Herzog C., '*SYPROS : An Intelligent Tutoring System for Parallel Programming*'. In Tak-Wan Chan, editor, Proc. 1993 International Conference on Computers in Education, Taiwan, 1993, 300–305.

<http://www.iasted.org/conferences/roor/austria/workshop/351-602.pdf>

Capuano, N.; Marsella, M., e Salerno S., '*ABITS: An Agent Based Intelligent Tutoring System for Distance Learning*', proceedings of the International Conference Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, L'Aquila, Jul. 31 – Aug. 06 2000.

http://www.capuano.biz/Papers/ITS_2000.pdf

Graesser, A.C., Olney, A., Haynes, B.C., & Chipman, P. (2005). '*AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue*'. In C. Forsythe, M.L. Bernard, and T.E. Goldsmith (Eds.), *Cognitive systems: Human cognitive models in systems design*. Mahwah, NJ: Erlbaum

<http://www.autotutor.org/>

Sykes, E. R., & Franek, F. (2003). '*An Intelligent Tutoring System Prototype for Learning to Program Java*'. Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, Athens, Greece, (pp. 485-486). <http://www.cas.mcmaster.ca/~franek/proceedings/syfra2.pdf>

Gütl, C., Pivec, M., ' *Virtual Tutor : A Multimedia Knowledge Module* *Virtual Tutors fosters Interactive Learning*' J. of Interactive Learning Research 14(2) pp. 209-236. http://www.iicm.edu/iicm_papers/virtual_tutor

Kinshuk, Nikov A. & Patel A. (2001). ' *Adaptive tutoring in business education using fuzzy backpropagation approach*'. In M. J. Smith, G. Salvendy, D. Harris & R. J. Koubek (Eds.) Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents and Virtual Reality, Proceedings of the 9th International Conference on Human-Computer Interaction (August 5-10, 2001, New Orleans, USA), Mahwah, NJ: Lawrence Erlbaum, 465-468 (ISBN 0-8058-3607-1, <http://infosys.massey.ac.nz/~kinshuk/papers/hcii2001.pdf>

Xu, D., Wang, H., Su, K., (2002) ' *Intelligent Student Profiling with Fuzzy Models* '. Proceedings of 35th Hawaii International Conference on System Sciences. <http://csdl.computer.org/comp/proceedings/hicss/2002/1435/03/14350081b.pdf>

Kosba, E., M., ' *Generative Computer-Based Advice in Web-Based Distance Education Environments*'. School of Computing-University of Leeds. <http://www.informatics.susx.ac.uk/research/hct/hctw2002/papers/Kosba.pdf>

Parte B

Di seguito, sono riportati i riferimenti ad alcuni sistemi adattivi ipermediali ed intelligenti non fuzzy, incontrati durante la stesura della tesi:

- *Calat*

Kiyama, M., Ishiuchi, S., Ikeda, K., Tsujimoto, M. & Fukuhara, Y. (1997). 'Authoring Methods for the Web-Based Intelligent CAI System CALAT and its Application to Telecommunications Service.' In the Proceedings of AAAI-97, Providence, RI.

http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Nakabayashi/Nakabayashi.html

- *ECSAIWEB*

Sanrach, C. & Grandbasien, M. (2000). 'ECSAIWeb: A Web-based authoring system to create adaptive learning systems'. In Proceedings of Adaptive Hypermedia 2000

www.springerlink.com/index/KLC2QDNQLHPAGFUK.pdf

- *GETMAS*

Wong, W.K. & Chan, T.W. (1997). 'A Multimedia authoring system for crafting topic hierarchy, learning strategies, and intelligent models'. International J. of Artificial Intelligence in Education, Vol. 8, No 1, pp. 71-96

http://aied.inf.ed.ac.uk/abstracts/Vol_8/wong.html

- *InterBook (Elm-Art + NetCoach)*

Brusilovsky, P., Schwartz, E., & Weber, G. (1996). 'A Tool for Developing Adaptive Electronic Textbooks on WWW'. *Proc. of WebNet-96*, AACE.

Brusilovsky, P., Schwartz, E. & Weber, G. (1996). 'ELM -ART: An Intelligent Tutoring System on the Work Wide Web'. In *Proceedings of ITS-96*, Frasson, Gauthier, Lesgold (Eds.), Springer: Berlin, 1996. pp. 261-269.

<http://www.contrib.andrew.cmu.edu/~plb/InterBook.html>

- *MetaLinks*

Murray, T., Condit, C., & Haaugsjaa, E. (1998). 'MetaLinks: A Preliminary Framework for Concept-based Adaptive Hypermedia'. *Workshop Proceedings from ITS-98 WWW-Based Tutoring Workshop*, San Antonio, Texas, 1998.

<http://www.aml.cs.umass.edu/~stern/webits/itsworkshop/murray.htm>

- *TANGOW*

Carro, R.M., Pulido, E., Rodriquies, P. (2002). 'An authoring tool that automates the process of developing task-based adaptive courses on the web'. *Journal of AI and Education*.

<http://wwwis.win.tue.nl/asum99/carro/carro.html>